How to Treat Timing Information for Software Effort Estimation?

Masateru Tsunoda
Toyo University
2100 Kujirai, Kawagoe, Saitama
350-8585 Japan
tsunoda@ieee.org

Sousuke Amasaki
Okayama Prefectural University
111 Kuboki, Soja, Okayama
719-1197 Japan
amasaki@cse.oka-pu.ac.jp

Chris Lokan
School of Engineering and IT
UNSW@ADFA
Canberra, Australia
c.lokan@adfa.edu.au

ABSTRACT
Software development effort estimation is an essential aspect of software project management. An effort estimation model expresses relationships between effort and factors such as organizational and project features (e.g. software functional size, and the programming language used in a project). However, software development practices and tools change over time, to environmental changes. This can affect some relationships assumed in an effort estimation model. A moving windows method (a method for treating the timing information of projects), has thus been proposed for estimation models. The moving windows method uses data from a fixed number of the most recent projects data for model construction. However, it is not clear that moving windows is the best way to handle the timing information in an estimation model. The goal of our research is to determine how best to treat timing information in constructing effort estimation models. To achieve the goal, we compared six different methods (moving windows, dummy variable of moving windows, dummy variables of equal bins, dummy variables of year, year predictor, and serial number) for treating timing data, in terms of estimation accuracy. In the experiment, we use three software development project datasets. We found that moving windows is best when the number of projects included in the dataset is not small, and dummy variable of moving windows is the best when the number is small.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *Cost estimation*, K.6.1 [**Computing Milieux**]: Project and People Management – *Staffing*

General Terms
Management, Measurement, Economics, Experimentation.

Keywords
Model-based effort estimation, time series, process changes, moving windows, interaction

ICSSP'13, May 18–19, 2013, San Francisco, USA.

1. INTRODUCTION
An effort estimation model expresses a relationship between effort and project features, such as software functional size and the programming language used in a project. The estimation model is trained with data obtained from past projects. However, sustainable organizations change their software development process and software development tool in order to adapt to environmental changes. The change can affect the relationship assumed in an effort estimation model.

The following formula is a typical effort estimation model based on software size:

$$\text{Effort} = a\,\text{Size}^{b} + \varepsilon \quad (1)$$

Here, a and b are coefficients inferred with training data from the past projects, and ε is an error term. Because b is usually close to 1, a corresponds to productivity. However, productivity can vary during the data collection period. This suggests that consideration of timing information (i.e. when the past projects were performed) is relevant in constructing effort estimation models. Kitchenham et al. [13] demonstrated that productivity varies over time, which may affect the accuracy of an estimation model. This issue should not be ignored in organizations that address process improvement.

A moving windows method [16] is one way to treat timing information. The moving windows method uses data from a fixed number of the latest projects for model construction. For example, suppose that an organization has data from 8 projects, as shown in Table 1. Here, the project records are listed in chronological order. If the window size is 4, only the the latest four finished projects (from P005 to P008) would be considered when estimating a new project with this dataset. In the research [16], moving windows was effective on a project data subset collected by ISBSG (International Software Benchmarking Standards Group).

The moving windows method treats the timing information by stratification. The method divides a dataset by time, instead of by types of the past projects such as adopted programming language. However, stratification does not always contribute to estimation accuracy. Tsunoda et al. [22] demonstrated that using dummy variables sometimes works better than stratification in handling categorical variables. Furthermore the authors suggested using an interaction term, which allows more flexible estimation models, might be effective. Consequently the question arises whether treating timing information by other methods than moving windows may be better, in terms of estimation accuracy.

The goal of our research is to determine how best to treat timing information in effort estimation model construction. To achieve the goal, we set three research questions as follows:

- RQ1. Do different methods for treating timing information lead to different estimation accuracy?
- RQ2. (If the answer of RQ1 is "yes") Which methods are effective for constructing effort estimation model?
- RQ3. Is using timing information always effective for constructing effort estimation models?

To address these questions, we compared six methods for treating timing information (moving windows, dummy variable of moving windows, dummy variables of equal bins, dummy variables of year, year predictor, and serial number) in terms of estimation accuracy. In the experiment, we adopted linear regression models and used three software development project datasets (ISBSG dataset [9], Maxwell dataset [19], and Kitchenham dataset [13]).

The contributions of our research are as follows:

- Proposal of three new methods for treating timing information (dummy variable of moving windows, dummy variables of equal bins, and serial number predictor).
- Empirical evaluation of the methods for treating timing information.
- Empirical evaluation of the effects of interaction term [1] between the timing information and functional size.

2. TREATING TIMING INFORMATION

Timing information can be treated as a predictor variable, in other ways than stratification (i.e. moving windows). To treat it as a predictor variable, we apply dummy variable of moving windows, dummy variables of equal bins, dummy variables of year, year predictor, and serial number. The first three methods treat the timing information as dummy variables, since estimation models based on dummy variables (an alternative method to stratification) showed better accuracy than stratification on some datasets in our research [22]. The last two methods treat the timing information as a continuous predictor. Some studies [10][14][19] used it as a continuous predictor variable. Note that it assumes there is a linear relationship between effort and serial number predictor to some extent, while timing information treatment methods with dummy variables do not assume such a relationship.

2.1 Moving Windows

Moving windows [16] uses *n* latest projects in a dataset, instead of all project data. Table 1 presents some example project data. The project data is listed in chronological order. With moving windows of $n = 4$, for instance, an estimation model for a new project P009 is trained with the four most recent projects, P005 to P008. When the project P009 finished, its project data is entered in the dataset. Then a new estimation model for P010 is trained with project data from P006 to P009.

The window size *n* is arbitrary but affects estimation accuracy. In [16], they found the best window size was around 75 (about one to two years of data) on a project data subset collected by ISBSG (International Software Benchmarking Standards Group).

2.2 Dummy Variable of Moving Windows

The use of dummy variables is a common method for handling a categorical variable. A categorical variable having n levels is replaced by n - 1 dummy variables taking a binary value 0 or 1.

Table 1. Example of moving windows (window size is four)

Project ID	Start date	Size	Effort	
P001	2009/7/9	106	10	
P002	2009/11/12	1520	129	
P003	2010/3/25	641	58	
P004	2010/8/7	392	44	
P005	2010/12/11	1156	95	Used to build a model
P006	2011/2/4	228	22	
P007	2011/6/30	963	103	
P008	2011/10/24	463	37	

For example, suppose that a categorical variable represents an adopted programming language either "C" or "Java". In this case, one dummy variable "C" replaces the categorical variable. "C" is set to 1 if a project adopted C. Otherwise "C" is set to 0, which means a project adopted "Java".

Moving windows is a form of stratification. It can also be realized with one dummy variable. The dummy variable takes 1 if a project finished recently. Otherwise it takes 0.

While the stratification produces effort estimation models for every level in a categorical variable, the use of dummy variables allows an effort estimation model to be trained with all data points in a dataset.

2.3 Dummy Variables of Year

This method makes dummy variables, each of which corresponds to a start year of projects. This method assumes that change in productivity can be identifiable in granularity of year. Table 4 shows an example of dummy variables of start years. The dummy variables "2011" of P006, P007, and P008 are set to 1 because these projects started in 2011. Likewise, the dummy variable "2010" of P003, P004, and P005 are set to 1.

2.4 Dummy Variables of Equal Bins

This method segments a time range of the past projects into an arbitrary number of equal bins. In contrast to the dummy variables of year, this method can define an adequate number of projects belonging to a dummy variable. Table 2 is an example of dummy variables of equal bins. Here, the size of bins is 3. The dummy variable "MW1" is set to 1 on projects P006, P007, and P008. Likewise, the dummy variable "MV2" is set to 1 on projects P003, P004, and P005. Values of the dummy variables are reassigned as shown in Table 3 when new project data enters the dataset.

2.5 Year Predictor

Some studies [10][14][19] used start years of projects as an independent continuous variable. In [19], for instance, a time variable, which is a relative year to the earliest year, remained a final candidate predictor after model selection. We used start years of projects as an independent variable in order to treat the timing information.

This method assumes that a relationship between effort and project features varies according to start years of projects. It has the advantage over moving windows that the method does not need to set a window (bin) size and can use more data points in model construction.

Table 2. Example of the dummy variables of equal bins (bin size is three)

Project ID	Start date	MW1	MW2	Size	Effort
P001	2009/7/9	0	0	106	10
P002	2009/11/12	0	0	1520	129
P003	2010/3/25	0	1	641	58
P004	2010/8/7	0	1	392	44
P005	2010/12/11	0	1	1156	95
P006	2011/2/4	1	0	228	22
P007	2011/6/30	1	0	963	103
P008	2011/10/24	1	0	463	37

Table 3. Example of the dummy variables of equal bins when new data is added

Project ID	Start date	MW1	MW2	MW3	Size	Effort
P001	2009/7/9	0	0	0	106	10
P002	2009/11/12	0	0	1	1520	129
P003	2010/3/25	0	0	1	641	58
P004	2010/8/7	0	0	1	392	44
P005	2010/12/11	0	1	0	1156	95
P006	2011/2/4	0	1	0	228	22
P007	2011/6/30	0	1	0	963	103
P008	2011/10/24	1	0	0	463	37
P009	2012/1/9	1	0	0	1392	151
P010	2011/4/11	1	0	0	505	67

2.6 Serial Number Predictor

The serial number predictor method converts project start date to an interval scale variable, and uses it as an independent variable. The variable holds the difference between an arbitrary base date and a start date of a project (cf. UNIX time or serial numbers of dates in Microsoft Excel). For example, if the base date is April 1, 2000, and the start date of a project is April 2, 2000, the variable holds 1.

The serial number predictor method has the same advantage as the year predictor over the moving windows: it can use more project data and there is no need to define a window (bin) size. In contrast to the year predictor, the serial number can represent more gradual change of relationships between a dependent variable and independent variables. It can also be free from an assumption that the change occurs according to year of project start date.

3. EXPERIMENT

To evaluate the effects of the timing information treating methods, we made effort estimation models using log-transformed linear regression (applying logarithmic transformation to ratio scale variables, which has a log-normal distribution) with the methods. This is a standard method for building estimation models [4][11]. Dummy variable of moving windows, year, dummy variables of year, serial number were also applied with the interaction, in addition to applying them without the interaction.

Table 4. Example of the dummy variables of year

Project ID	Start date	2011	2010	Size	Effort
P001	2009/7/9	0	0	106	10
P002	2009/11/12	0	0	1520	129
P003	2010/3/25	0	1	641	58
P004	2010/8/7	0	1	392	44
P005	2010/12/11	0	1	1156	95
P006	2011/2/4	1	0	228	22
P007	2011/6/30	1	0	963	103
P008	2011/10/24	1	0	463	37

3.1 Model Formulation

We evaluated the effects of the timing information treating methods with effort estimation models based on linear regression, as in our past study [22]. We employed the following formulation:

$$\log(\text{Effort}) = \beta_0 + \beta_1 \log(\text{Size}) + \beta_2 y + \beta_3 z + \varepsilon. \quad (2)$$

Here, y denotes the timing information variable, z denotes an explanatory variable, and β_k are regression coefficients. Equation (2) retains the same relationship between effort and functional size as Eq. (1). Equation (2) can include other independent variables if those are in project data.

We also employed Eq. (2) with the interaction. The interaction [1] introduces a new variable made by multiplying independent variables. It may improve estimation accuracy [22]. We introduced the interaction between functional size and a timing information variable. The resultant model includes the new variable as follows:

$$\log(\text{Effort}) = \beta_0 + \beta_1 \log(\text{Size}) + \beta_2 y + \beta_3 z + \beta_4 \log(\text{Size}) y + \varepsilon. \quad (3)$$

Here, $\log(\text{Size})y$ is the new variable. Equation (3) can be transformed as:

$$\log(\text{Effort}) = \beta_0 + (\beta_1 + \beta_4 y)\log(\text{Size}) + \beta_2 y + \beta_3 z + \varepsilon. \quad (4)$$

The relationship between log(Effort) and log(Size) varies according to y. The averages of log(Size) is subtracted from log(Size) in order to avoid multicollinearity between a main effect (log(Size)) and an interaction (log(Size)y). If y is a continuous variable, its average is also subtracted from y.

3.2 Datasets

We used ISBSG dataset [9], Maxwell dataset [19], and Kitchenham dataset [13] because they are relatively large and record start dates of projects. We assumed an estimation point is at the end of the basic design phase. Therefore, an effort estimation model uses project features as independent variables which values were fixed at the point.

3.2.1 ISBSG Dataset

The ISBSG dataset (Release 10) was collected by ISBSG (International Software Benchmarking Standards Group). It includes 4106 project data and more than 100 variables collected from software development companies in various countries. We used projects collected from a single company, the same as in the past research [16]. This is because transitions of the relationships between effort and project features are different for each company, and using cross-company data might cancel effects of timing information methods.

Table 5. Estimation accuracy of the baseline models on the each dataset

Dataset	*AE*	*MdAE*	*MRE*	*MdMRE*	*MER*	*MdMER*	*BRE*	*MdBRE*
ISBSG	2301	1268	135.1%	72.5%	99.3%	57.8%	186.0%	107.0%
Maxwell	3202	1458	50.4%	29.0%	49.3%	23.8%	68.6%	29.9%
Kitchenham	1094	733	61.1%	38.7%	67.0%	39.9%	92.8%	52.3%

The single company dataset holds project data rated as A or B on data quality. These projects adopted IFPUG 4.0 or later for size measurement. We also removed projects having missing values by list-wise deletion. The remained project data has 217 projects carried out between June 1994 and December 2002. We used latest 76 projects started after July 2001 as test dataset. Note that the test dataset is a form of hold-out set.

Independent variables: unadjusted function point, development type, development platform, programming language, and industry sector. The industry sector was added in order to improve estimation accuracy in [2].

3.2.2 Maxwell Dataset

The Maxwell dataset has 62 software development projects of a bank in Finland, shown in Maxwell's book [19]. The projects were carried out between October 1985 and November 1993. We used the latest 20 projects, started after April 1991, as the test dataset. They have 25 project features. The number of candidate predictors was larger than some window (bin) sizes, and we could not perform variable selection by backward stepwise regression. We thus conducted a preliminary analysis to select possible predictors. The preliminary analysis did variable selection on the oldest 25 project in the dataset. Those projects were not used for a testing dataset in our experiment.

Independent variables: function point and three ordinal scale variables (customer participation, standards use, and tools use).

3.2.3 Kitchenham Dataset

The Kitchenham dataset [13] includes 145 projects from Computer Sciences Corporation (CSC). We selected 135 projects which do not include missing values. The projects were carried out between May 1994 and August 1998. We used the latest 63 projects, started after November 1996, as the test dataset.

Independent variables: adjusted function point and development type.

3.3 Experimental Procedure

This study evaluated the effects of the timing information methods by moving windows of several sizes along with a timeline of projects' history. This limit is due to the original moving windows method. We followed the below steps:

1. Sort all projects by start date.
2. Find the earliest project p_0 for which at least $w+1$ projects, where w is a window (bin) size, were completed prior to the start of p_0.
3. For every project p_i in chronological sequence, starting from p_0, make estimates with the timing information methods. For the original moving windowing method, the training set is the w most recent projects that finished before the start of p_i. For the other methods, the training set is all of the projects that finished before the start of p_i.
4. Evaluate estimation results.
5. Change window (bin) size w, and repeat step 2 to 4.

The estimation models perform variable selection based on AIC (Akaike's information criterion). Thus, all independent variables are just candidates. All categorical variables other than the timing information were converted into binary dummy variables (we gathered together minor categories into consolidated categories in order to lessen candidates of independent variables). We compared the performance with that of a baseline model. The baseline model did not consider any timing information and uses all project data. We evaluated the significance of differences in estimation accuracy using Wilcoxon signed rank test with significance level at 0.05.

We set minimum window size and bin size as 10, based on past studies [8][13] (Study [8] suggests that 3 may be useable if they show a reasonable correlation between size and effort, and study [13] suggests at least 30 training projects and no fewer than 20. We adopted almost the middle of 3 and 20). Maximum window size and bin size were set, considering the size of training dataset.

3.4 Performance Measures

To evaluate the accuracy of estimation models, we used average and median of *AE* (Absolute Error), *MRE* (Magnitude of Relative Error) [6], *MER* (Magnitude of Error Relative to the estimate) [12], and *BRE* (Balanced Relative Error) [20]. Especially, *MRE* is widely used to evaluate effort estimation accuracy [21].

A lower value of each criterion indicates higher estimation accuracy. Intuitively, *MRE* means error relative to actual effort, and *MER* means error relative to estimated effort. However, *MRE* and *MER* have biases for evaluating under and over estimation [5] [15]. Accordingly we adopted *BRE* whose evaluation is not biased as is both *MRE* and *MER* [21], and we evaluated the timing information treating methods based mainly on *BRE* (*MRE* and *MER* were adopted for reference). We did not use Pred(25) [6], which is sometimes used as an evaluation criterion, because Pred(25) is based on *MRE* and it has also a bias for evaluating under estimation.

4. RESULTS AND DISCUSSION

Table 5 shows estimation accuracy of the baseline models on each dataset. Figure 1, Figure 2, and Figure 3 depict the difference in mean *BRE* and median *BRE* against window sizes and bin sizes. The x-axis is the size of the window (bin), and the y-axis is the subtraction of the accuracy measure value with the timing information methods from that with the corresponding baseline models at the given x-value. A timing information method is advantageous where the line is beyond 0 because smaller *BRE* is better. Results of dummy variables of year, year predictor, and

Table 6. Difference of criteria from the baseline on ISBSG dataset

Timing information treating method	*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
Moving windows <54>	394	348	26.2%	14.5%	17.7%	4.1%	40.1%	10.9%
	(0.02)		**(0.01)**		(0.22)		**(0.02)**	
Dummy variable of moving windows <51>	285	282	24.5%	13.3%	*-2.1%*	3.6%	19.3%	16.7%
	(0.02)		**(0.00)**		(0.33)		**(0.01)**	
Dummy variable of moving windows <62>	354	190	18.0%	10.6%	3.5%	0.3%	19.8%	15.1%
+ interaction	**(0.08)**		**(0.03)**		(0.34)		**(0.05)**	
Dummy variables of equal bins <49>	315	301	26.3%	17.1%	*-2.0%*	3.8%	20.0%	16.1%
	(0.00)		**(0.00)**		(0.07)		**(0.00)**	
Dummy variables of equal bins <63>	355	182	18.2%	10.7%	2.9%	0.5%	19.2%	15.3%
+ interaction	**(0.05)**		**(0.01)**		(0.24)		**(0.03)**	
Dummy variables of year	*-198*	*-329*	*-7.7%*	*-10.8%*	*-7.9%*	*-6.4%*	*-12.0%*	*-15.3%*
	(0.03)		**(0.00)**		(0.13)		**(0.01)**	
Dummy variables of year + interaction	*-81*	*-129*	2.1%	*-12.8%*	*-8.5%*	*-0.2%*	*-4.2%*	*-15.8%*
	(0.25)		(0.16)		(0.29)		(0.09)	
Year predictor	238	304	22.8%	12.2%	*-14.3%*	4.2%	7.1%	14.1%
	(0.13)		**(0.00)**		(0.48)		**(0.04)**	
Year predictor + interaction	238	304	22.8%	12.2%	*-14.3%*	4.2%	7.1%	14.1%
	(0.13)		**(0.00)**		(0.48)		**(0.04)**	
Serial number predictor	232	273	23.1%	12.7%	*-4.7%*	4.1%	16.9%	15.7%
	(0.10)		**(0.00)**		(0.66)		**(0.03)**	
Serial number predictor + interaction	232	273	23.1%	12.7%	*-4.7%*	4.1%	16.9%	15.7%
	(0.10)		**(0.00)**		(0.66)		**(0.03)**	

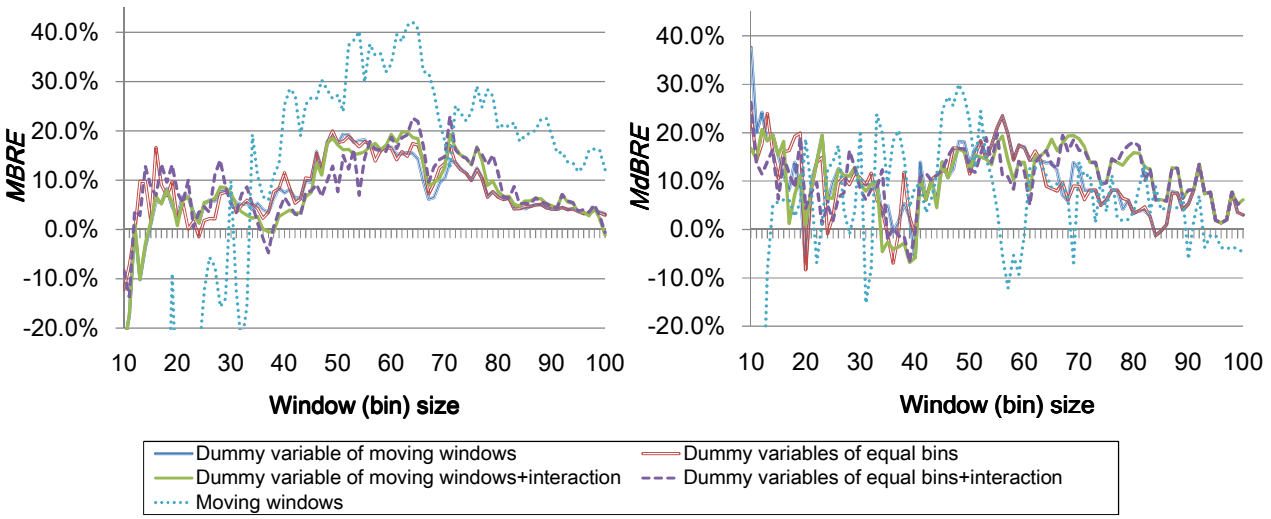

Figure 1. Relationship between window (bin) size and difference of criteria from the baseline on ISBSG dataset

serial number are not included in the figures, since they do not need to set window size or bin size.

Table 6, Table 7, and Table 8 show the difference of performance between models with the timing information treating methods and the corresponding baseline models. The tables show the results with the best window size and the best bin size for the corresponding methods. Numbers in angle brackets denote the best window size or the best bin size (The best size is settled based on *MBRE* and p-value). A positive value in a cell means superiority of a timing information method, while a negative value means inferiority (a negative value is italicized). Numbers in parentheses indicate p-values on the statistical tests of the differences between models with the timing information treating methods and without the methods (a bold number means the p-value is statistically significant, i.e. smaller than 0.05).

4.1 ISBSG Dataset

We observe from Figure 1 and Table 6 the following:

Moving windows: It showed the best improvement of *MAE*, *MdAE*, and *MBRE*, and they were significantly better than the model without timing information treatment methods. The improvement of *MdBRE* was also the highest when window size is close to 50.

Dummy variable of moving windows, and dummy variables of equal bins: They significantly improved the accuracy of the

Table 7. Difference of criteria from the baseline on Maxwell dataset

Timing information treating method	*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
Moving windows <20>	316	453	6.3%	9.8%	4.1%	2.6%	7.0%	8.7%
	(0.05)		(0.05)		(0.03)		(0.05)	
Dummy variable of moving windows <20>	200	261	6.7%	6.2%	1.7%	*-0.9%*	6.1%	1.8%
	(0.08)		(0.10)		(0.09)		(0.06)	
Dummy variable of moving windows <20>	269	386	6.7%	6.2%	4.7%	*-1.1%*	9.4%	1.4%
+ interaction	(0.22)		(0.10)		(0.11)		(0.06)	
Dummy variables of equal bins <20>	198	241	6.5%	5.7%	1.4%	*-0.9%*	5.9%	1.8%
	(0.08)		(0.11)		(0.08)		(0.06)	
Dummy variables of equal bins <22>	270	543	6.8%	9.8%	4.1%	1.3%	7.2%	6.0%
+ interaction	(0.22)		(0.19)		(0.23)		(0.26)	
Dummy variables of year	*-1,042*	129	*-1.0%*	*-18.2%*	*-50.4%*	*-31.0%*	*-42.1%*	*-44.0%*
	(0.11)		(0.62)		**(0.02)**		(0.09)	
Dummy variables of year + interaction	*-1,244*	*-380*	*-3.8%*	*-27.4%*	*-59.1%*	*-26.0%*	*-49.7%*	*-51.7%*
	(0.05)		(0.28)		**(0.00)**		**(0.04)**	
Year predictor	*-969*	*-7*	5.5%	*-14.7%*	*-54.4%*	*-22.8%*	*-38.4%*	*-34.7%*
	(0.14)		(0.57)		**(0.01)**		(0.11)	
Year predictor + interaction	*-991*	9	5.4%	*-17.0%*	*-52.5%*	*-33.2%*	*-36.6%*	*-34.2%*
	(0.08)		(0.60)		**(0.00)**		(0.10)	
Serial number predictor	*-918*	97	6.4%	*-13.6%*	*-49.7%*	*-25.2%*	*-33.6%*	*-29.5%*
	(0.15)		(0.55)		**(0.00)**		(0.10)	
Serial number predictor + interaction	*-846*	86	7.2%	*-14.1%*	*-43.9%*	*-30.2%*	*-27.8%*	*-27.7%*
	(0.26)		(0.67)		**(0.01)**		(0.14)	

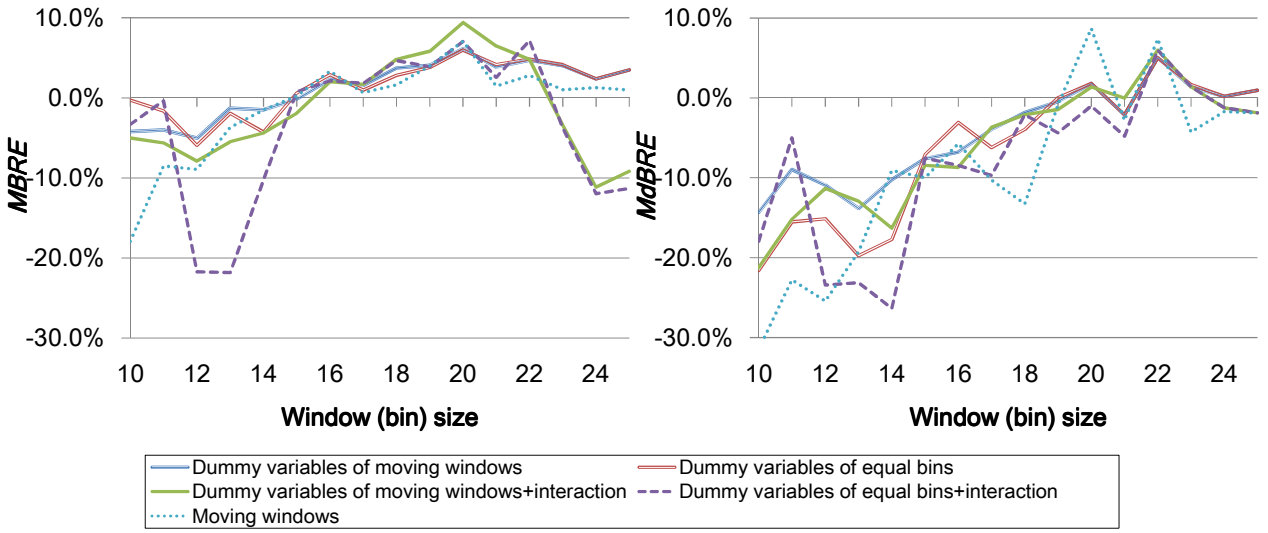

Figure 2. Relationship between window (bin) size and difference of criteria from the baseline on Maxwell dataset

estimation model, and the improvement of evaluation criteria was relatively higher. They had similar tendency in Figure 1.

Dummy variables of year: It significantly worsened (median of) *AE*, *MRE*, and *BRE*.

Year predictor: It significantly improved (median of) *MRE* and *BRE*. Compared with other timing information treating methods, the performance was not very high.

Serial number predictor: It significantly improved (median of) *MRE* and *BRE*. The improvement of performance was medium among timing information treating methods.

Interaction: It contributed to a slight improvement of evaluation criteria in some cases. However, the degree of effect was unstable when it was applied to dummy variable of moving windows and dummy variables of equal bins.

4.2 Maxwell Dataset

We observe from Figure 2 and Table 7 the followings:

Moving windows: It showed the best improvement of *MAE*, *MdAE*, *MdER* and *MdBRE*. However, the performance was not less stable than that of dummy variable of moving windows or dummy variables of equal bins.

Table 8. Difference of criteria from the baseline on Kitchenham dataset

Timing information treating method	*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
Moving windows <30>	21	197	2.7%	0.6%	2.8%	5.5%	4.7%	9.4%
	(0.56)		(0.42)		(0.46)		(0.37)	
Dummy variable of moving windows <30>	32	115	1.0%	0.5%	1.9%	4.5%	1.9%	8.7%
	(0.06)		(0.06)		(0.06)		(0.06)	
Dummy variable of moving windows <50>	*-7*	*-47*	3.8%	0.0%	*-0.5%*	*-3.7%*	4.2%	3.9%
+ interaction	(0.38)		(0.21)		(0.43)		(0.53)	
Dummy variables of equal bins <14>	38	154	1.8%	5.3%	3.6%	*-1.0%*	3.8%	8.9%
	(0.17)		(0.36)		**(0.03)**		(0.09)	
Dummy variables of equal bins <52>	*-8*	*-47*	3.8%	0.0%	*-0.6%*	*-3.7%*	4.1%	3.9%
+ interaction	(0.35)		(0.19)		(0.40)		(0.46)	
Dummy variables of year	6	36	2.6%	3.0%	*-2.7%*	*-0.4%*	*-0.2%*	8.2%
	(0.87)		(0.25)		(0.06)		(0.46)	
Dummy variables of year + interaction	*-237*	12	6.6%	*-3.7%*	*-109.8%*	*-16.3%*	*-97.2%*	*-14.2%*
	(0.03)		(0.05)		**(0.02)**		(0.07)	
Year predictor	*-36*	*-75*	6.0%	*-0.3%*	*-8.4%*	*-1.7%*	*-1.1%*	*-2.5%*
	(0.33)		(0.71)		**(0.02)**		(0.33)	
Year predictor + interaction	*-220*	163	7.2%	*-6.7%*	*-41.5%*	*-22.1%*	*-28.5%*	*-15.1%*
	(0.10)		(0.31)		**(0.03)**		(0.15)	
Serial number predictor	*-40*	68	5.7%	0.5%	*-9.6%*	*-3.7%*	*-2.4%*	*-1.4%*
	(0.40)		(0.82)		**(0.02)**		(0.31)	
Serial number predictor + interaction	*-269*	68	3.8%	*-12.5%*	*-54.6%*	*-20.6%*	*-43.2%*	*-21.3%*
	(0.05)		(0.16)		**(0.01)**		**(0.04)**	

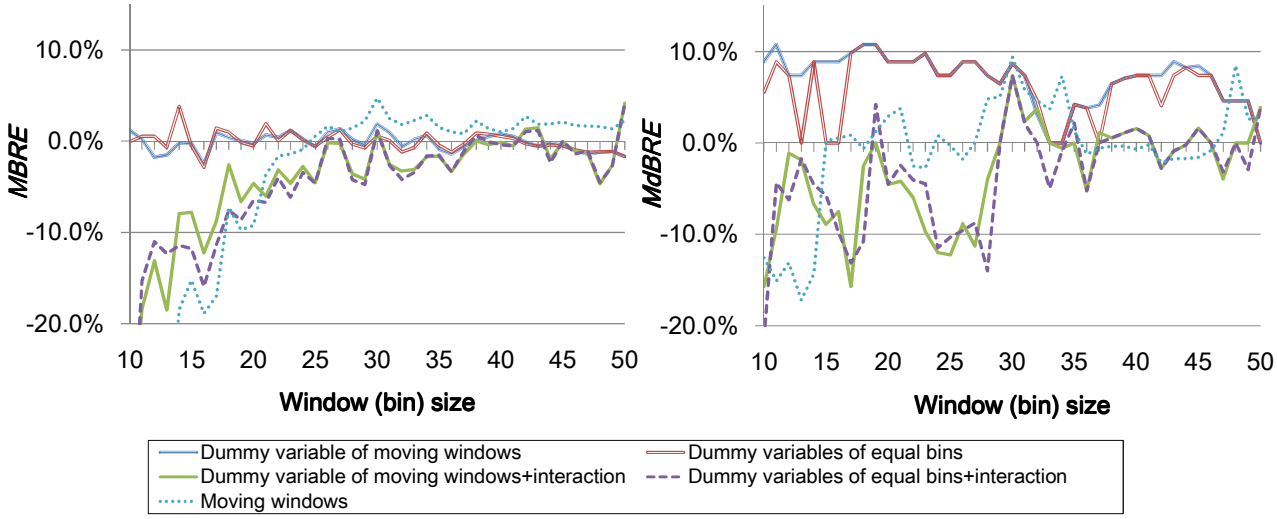

Figure 3. Relationship between window (bin) size and difference of criteria from the baseline on Kitchenham dataset

Dummy variable of moving windows, and dummy variables of equal bins: They improved evaluation criteria except for *MdMER*. Although the improvement was not large and not statistically significant, at least they did not worsen the estimation accuracy. They had similar tendency in Figure 2. The performances were more stable than those of the moving windows.

Dummy variables of year, year predictor, and serial number predictor: They worsened most of estimation criteria.

Interaction: It contributed to a slight improvement of evaluation criteria in some cases. However, the degree of effect was unstable when it was applied to dummy variable of moving windows and dummy variables of equal bins.

4.3 Kitchenham Dataset

We observe from Figure 3 and Table 8 the following:

Moving windows: It showed the best improvement of *MdAE*, *MdMER*, *MBRE*, and *MdBRE*.

Dummy variable of moving windows, and dummy variables of equal bins: They improved the accuracy of the estimation model in evaluation criteria except for *MdMER*. They had similar tendency in Figure 3. However, the degree of effects of dummy variables of equal bins was less stable than that of the dummy variable of moving windows.

Table 9. Best window (bin) size and estimation accuracy

(a) Moving windows

		Window size				
Dataset	**Criteria**	**46**	**47**	**48**	**49**	**50**
ISBSG	*MBRE*	26.5%	30.3%	28.3%	26.5%	27.2%
ISBSG	*MdBRE*	27.4%	25.6%	30.0%	27.6%	22.3%
Kitchenham	*MBRE*	1.7%	1.7%	1.6%	1.3%	2.3%
Kitchenham	*MdBRE*	*-0.8%*	1.1%	8.5%	2.6%	2.6%

		Window size				
Dataset	**Criteria**	**21**	**22**	**23**	**24**	**25**
Maxwell	*MBRE*	1.5%	2.8%	1.0%	1.3%	1.0%
Maxwell	*MdBRE*	*-2.8%*	7.4%	*-4.3%*	*-1.7%*	*-1.9%*

(b) Dummy variables of moving windows

		Window size				
Dataset	**Criteria**	**21**	**22**	**23**	**24**	**25**
ISBSG	*MBRE*	6.7%	6.7%	3.4%	1.3%	5.1%
ISBSG	*MdBRE*	10.0%	13.9%	19.4%	5.6%	10.0%
Maxwell	*MBRE*	3.8%	4.7%	4.0%	2.4%	3.5%
Maxwell	*MdBRE*	*-2.1%*	5.0%	1.4%	0.2%	0.9%
Kitchenham	*MBRE*	0.7%	0.4%	1.2%	0.3%	*-0.6%*
Kitchenham	*MdBRE*	8.9%	8.9%	9.8%	7.4%	7.4%

(c) Dummy variables of equal bins

		Bin size				
Dataset	**Criteria**	**21**	**22**	**23**	**24**	**25**
ISBSG	*MBRE*	5.4%	0.1%	1.7%	-1.5%	1.8%
ISBSG	*MdBRE*	8.3%	13.9%	14.9%	-0.9%	2.9%
Maxwell	*MBRE*	4.2%	4.8%	4.1%	2.4%	3.5%
Maxwell	*MdBRE*	*-2.1%*	5.0%	1.7%	0.2%	0.9%
Kitchenham	*MBRE*	1.9%	0.1%	1.2%	0.1%	*-0.6%*
Kitchenham	*MdBRE*	8.9%	8.9%	9.8%	7.4%	7.4%

Dummy variables of year: It improved *MAE*, *MdAE*, *MRE*, *MdMRE*, and *MdBRE*, but it worsened *MBRE* slightly.

Year predictor and serial number predictor: They worsened most of estimation criteria.

Interaction: It contributed to a slight improvement of evaluation criteria in some cases. However, the degree of effect was unstable when it was applied to dummy variable of moving windows and dummy variables of equal bins.

4.4 Discussion

We first summarize the results of the experiment and discuss methods for treating timing information based on the results.

Moving windows was effective in 'sweet spots'. The moving windows improved estimation accuracy in relatively wide range of window sizes. However, the degree of improvement was unstable and showed a negative effect in case of small windows. Small training datasets might cause the negative effect, since moving windows worked on the ISBSG dataset and the Kitchenham dataset, but not on the Maxwell dataset. We conclude that moving windows requires a careful attention to sample size of dataset for performance improvement.

Dummy variables of moving windows and equal bins were also effective. The effect of these methods was more modest than that of moving windows. However, these methods retained positive effects in a wider range than moving windows. These methods are similar in where they are effective. However, the use of dummy variables of moving windows was more preferable in stability.

Based on the results on the three datasets, we discuss setting of window size and bin size. Table 9 shows the best window (bin) size and estimation accuracy on the three datasets. When dummy variables of moving windows or dummy variables of equal bins are applied, and window (bin) size was 21 to 25, estimation accuracy was almost improved on all datasets. When moving windows are applied, and window size was 46 to 50, estimation accuracy was almost improved on two datasets. When the window size is 21 to 25 on the Maxwell dataset, the performance of moving windows was less stable than dummy variables of moving windows and dummy variables of equal bins. The best window (bin) sizes were the same among the three datasets. The coincidence might suggest how well an organization follows environmental changes. However, in practice, an organization should determine the best window (bin) size with an experiment on its own project data.

The dummy variables of the year showed a negative effect on Maxwell and Kitchenham datasets. Year predictor also showed a negative effect on those datasets. They showed positive but smaller effects on ISBSG dataset. We conclude that the use of year information may not be a good idea. The timing information with smoother and more flexible granularity is preferable.

Serial number predictor also showed a negative effect on the Maxwell dataset and the Kitchenham datasets. The serial number predictor showed better results than the year predictor on the ISBSG dataset though they are similar in the trend of effects. However, the serial number predictor was still worse than the dummy variable methods.

We examined the effects of interaction and found that the interaction made the effects of the timing information more subtle. The significance difference became insignificant in many cases. The interaction adds instability to an effort estimation model in many cases.

Based on the above discussion, we answered the research questions as follows:

The answer to RQ1 (do different methods give different accuracy?) is YES. The experiment revealed that an adequate timing information treating method significantly improved estimation accuracy in ISBSG dataset. In addition, the use of an inadequate method for treating timing information is not only ineffective but also harmful for estimation accuracy.

The answer to RQ2 (which timing information treating methods are effective for constructing effort estimation model?) is that moving windows is the best choice to enhance estimation accuracy of an effort estimation model once an organization has a moderate amount of project data. If an organization has only a small amount of project data, dummy variables of moving windows is the best choice. Although inappropriate application, i.e., small window (bin) size and interaction worsens estimation accuracy, appropriate application of them does not worsen

estimation accuracy, and it may be very effective to improve the accuracy for some datasets.

The answer to RQ3 (is it always effective to use timing information?) is NO. No timing information treating method ever improved estimation accuracy on Maxwell and Kitchenham datasets significantly. However, some adequate timing information treating methods never worsen estimation accuracy on any dataset. Generally timing information contributes to estimation accuracy more clearly for a larger dataset. For small datasets, the effect on estimation accuracy is also small. We think this is due to model complexity. Empirical software engineering datasets in practice are often small. So, practitioners should consider these findings carefully.

The results also suggest that there is no single best timing information treating method, and that the characteristics of each dataset is influential in the results. However, we think the results suggest that selecting an appropriate method for treating timing information does not worsen estimation accuracy, and sometimes enhances the accuracy.

5. THREATS TO VALIDITY

This study has some limitations and threats to validity.

First, the accuracy of the raw data may be questioned. A project data records a start and end date. The recorded start date can be assumed to be correct, but there is a small probability that the recorded end date may not be correct (e.g. projects can be officially closed off for compliance reasons, yet work on them continues). However, we think the uncertainty this introduces is small.

Second, the datasets used here are convenience samples and may not be representative of software projects in general. Thus, the results may not be generalized beyond these datasets; this is true of all studies based on convenience samples. We used various size and span of datasets. That is, the size of the Maxwell dataset is small, and that of ISBSG dataset is large. The time span of the data in the ISBSG dataset is long, and that of the Kitchenham dataset is short. So, we assume they are fair representations of typical organizations' projects. We trust that numerous potential sources of variation are small on those datasets, since single-company datasets were used. We note that data collected from large organizations may be as diverse as cross-company datasets. So, it may be good to stratify such datasets within departments, to make the datasets homogeneous.

Third, all the models employed in this study were built automatically. Automating the process necessarily involved making some assumptions, and the validity of our results depends on those assumptions being reasonable. For example, logarithmic transformation is assumed to be adequate to transform numeric data to an approximately normal distribution; residuals are assumed to be random and normally distributed without that being actually checked; when choosing between two models in which all independent variables were significant, the one with higher AIC is assumed to be preferred; multi-collinearity between independent variables is assumed to be handled automatically by the nature of the stepwise procedure. Based on our past experience building models manually, we believe that these assumptions are acceptable. One would not want to base important decisions on a single model built automatically, without at least doing some serious manual checking, but for calculations such as project-by-project chronological estimation across a substantial dataset we believe that the process here is reasonable.

6. RELATED WORK

Consideration of timing information is not common in effort estimation research, though some studies [13][19] emphasized its importance. Those studies that did timing did not consider different methods. For example, Auer and Biffl [3] evaluated dimension weighting for analogy-based effort estimation, considering the effect of a growing dataset. However, they did not consider any of the methods for treating timing information studied here.

Some analogy-based effort estimation research [10][14] uses year of project end date as an independent variable. This may be because the Desharnais dataset [7], which is often used in analogy-based effort estimation research, includes the year of project end date. However, their experiments did not consider the chronological order of projects (e.g. effort of some older projects was estimated by using a model based on a dataset including newer projects).

MacDonell and Shepperd [18] investigated moving windows as part of a study into how well data from prior phases in a project could be used to estimate later phases. They found that accuracy was better when a moving window of the 5 most recent projects was used as training data, rather than using all completed projects as training data.

Lokan and Mendes [17] investigated the effect on accuracy when using moving windows of various durations to form training sets on which to base effort estimates. They showed that the use of windows based on duration can affect the accuracy of estimates, but to a lesser extent than windows based on a fixed number of projects. Applying our methods for treating timing information, compared to duration-based windows, is a topic for our future work.

The prior literatures treated timing information by a single method, and did not treat it by various methods. In contrast, our paper handled it by six treating methods, to suggest how to treat timing information in the software effort estimation research area. Timing information treating methods can be applied to various estimation methods, and it is expected to improve the estimation accuracy when the application is adequate.

7. CONCLUSIONS

In this paper, we evaluated the performance of methods for treating timing information, in order to determine how best to treat the information in effort estimation models. We devised six treating methods (moving windows, dummy variable of moving windows, dummy variables of equal bins, dummy variables of year, year predictor, and serial number predictor) and evaluated their effects on estimation accuracy with linear regression, with and without an interaction term, on three datasets.

The experimental results showed that different timing information treating methods lead to different estimation accuracy, and sometimes worsen estimation accuracy when a timing information treating method was inadequate. However, appropriate methods for treating timing information often improved estimation accuracy and did not worsen it. We suggest dummy variable of moving windows is preferable when the size of a dataset is small.

Moving window is also preferable when the size is large. It is not necessary to apply other methods because their effects are small.

We believe this suggestion is useful for organizations which address process improvement. To improve accuracy of effort estimation, practitioners such as people in PMO (Project Management Office) should consider applying the methods when building estimation models. Although applying the methods requires practitioners to invest additional analysis effort, the effect of the method (enhancing estimation accuracy) is worthwhile.

As future work, we plan to examine the effectiveness of those methods considering timing windows based on duration.

8. REFERENCES

[1] Aiken, L., West, S. 1991. *Multiple Regression: Testing and Interpreting Interactions*. SAGE Publications.

[2] Amasaki, S., Lokan, C. 2012. The Effects of Moving Windows to Software Estimation: Comparative Study on Linear Regression and Estimation by Analogy. In *Proceedings of International Workshop on Software Measurement (IWSM) and International Conference on Software Process and Product Measurement (Mensura), Assisi, Italy,* 23-32.

[3] Auer, M., and Biffl, S. 2004. Increasing the accuracy and reliability of analogy-based cost estimation with extensive project feature dimension weighting. In *Proceedings of International Symposium on Empirical Software Engineering (ISESE)*. Redondo Beach, CA, 147–155.

[4] Briand, L., Langley, T., and Wieczorek, I. 2000. A replicated assessment and comparison of common software cost modeling techniques. *In Proceedings of international conference on Software engineering (ICSE)*. Limerick, Ireland, 377-386.

[5] Burgess, C., and Lefley, M. 2001. Can genetic programming improve software effort estimation? A comparative evaluation. *Journal of Information and Software Technology* 43, 14, 863-873.

[6] Conte, S., Dunsmore, H., and Shen, V. 1986. *Software Engineering, Metrics and Models*. Benjamin/Cummings.

[7] Desharnais, J. 1989. *Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function*. Master Thesis. University of Montreal.

[8] Humphrey, W. 1995. *A Discipline for Software Engineering*. Addison-Wesley Professional.

[9] International Software Benchmarking Standards Group (ISBSG). 2007. ISBSG Estimating, Benchmarking and Research Suite Release 10. ISBSG.

[10] Keung, J., Kitchenham, B., and Jeffery, R. 2008. Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation. *IEEE Trans. on Software Eng*. 34, 4, 471-484.

[11] Kitchenham, B., and Mendes, M. 2009. Why comparative effort prediction studies may be invalid. In *Proceedings of International Conference on Predictor Models in Software Engineering (PROMISE)*. Vancouver, Canada, Article 4, 5 pages.

[12] Kitchenham, B., MacDonell, S., Pickard, L., and Shepperd, M. 2001. What Accuracy Statistics Really Measure. In *Proceedings of IEE Software*. 148, 3, 81-85.

[13] Kitchenham, B., Pfleeger, S., McColl, B., and Eagan, S. 2002. An Empirical Study of Maintenance and Development Estimation Accuracy. *Journal of Systems and Software* 64, 1, 57-77.

[14] Li, J. and Ruhe., G. 2008. Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+. *Empirical Softw. Engg.* 13, 1 (February 2008), 63-96.

[15] Lokan, C. 2005. What Should You Optimize When Building an Estimation Model? In *Proceedings of International Software Metrics Symposium (METRICS)*. Como, Italy, 34.

[16] Lokan, C. and Mendes, E. 2009. Applying moving windows to software effort estimation. In *Proceedings of International Symposium on Empirical Software Engineering and Measurement (ESEM)*. Orlando, Florida, 111-122.

[17] Lokan, C. and Mendes, E. 2012. Investigating the Use of Duration-based Moving Windows to Improve Software Effort Prediction. In *Proceedings of 19th Asia-Pacific Software Engineering Conference (APSEC)*. Hong Kong, China, 818-827.

[18] MacDonell, S. G., and Sheppered, M., 2010. Data Accumulation and Software Effort Prediction, In *Proceedings of International Symposium on Empirical Software Engineering and Measurement*. Bolzano-Bozen, Italy, 31:1-31:4.

[19] Maxwell, K. 2002. *Applied Statistics for Software Managers*. Prentice Hall.

[20] Miyazaki, Y., Terakado, M., Ozaki, K., and Nozaki, H. 1994. Robust Regression for Developing Software Estimation Models. *Journal of Systems and Software* 27, 1, 3-16.

[21] Mølokken-Østvold, K., and Jørgensen, M. 2005. A Comparison of Software Project Overruns-Flexible versus Sequential Development Models. *IEEE Trans. on Software Eng*. 31, 9, 754-766.

[22] Tsunoda, M., Amasaki, S., and Monden, A. 2012. Handling categorical variables in effort estimation. In *Proceedings of international symposium on Empirical software engineering and measurement (ESEM)*. Lund, Sweden, 99-102.