Investigating the Use of Chronological Splitting to Compare Software Cross-company and Single-company Effort Predictions: A Replicated Study

Emilia Mendes¹, Chris Lokan²

¹ Computer Science Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand
emilia@cs.auckland.ac.nz

² School of IT&EE, UNSW@ADFA, Canberra ACT 2600, Australia
c.lokan@adfa.edu.au

CONTEXT: Three previous studies have investigated the use of chronological split to compare cross- to single-company effort predictions, where all used the ISBSG dataset release 10. Therefore there is a need for these studies to be replicated using different datasets such that the patterns previously observed can be compared and contrasted, and a better understanding with regard to the use of chronological splitting can be reached.

OBJECTIVE: The aim of this study is to replicate [17] using the same chronological splitting; however a different database – the Finnish dataset.

METHOD: Chronological splitting was compared with two forms of cross-validation. The chronological splitting used was the project-by-project chronological split, in which a validation set contains a single project, and a regression model is built from scratch using as training set the set of projects completed before the validation project's start date. We used 201 single-company projects and 593 cross-company projects from the Finnish dataset.

RESULTS: Single-company models presented significantly better prediction than cross-company models. Chronological splitting provided significantly worse accuracy than leave-one and leave-two out cross-validations when based on single-company data; and provided similar accuracy when based on cross-company data.

CONCLUSIONS: Results did not seem promising when using project-by-project splitting; however in a real scenario companies that use their own data can only apply some sort of chronological splitting when obtaining effort estimates for their new projects. Therefore we urge the use of chronological splitting in effort estimation studies such that more realistic results can be provided to inform industry.

Keywords: *chronological split, effort estimation, software projects, cross-company estimation models, single-company estimation models, regression-based estimation models.*

1. INTRODUCTION

Numerous software companies find it difficult to accumulate data on their past finished projects, and yet, to remain competitive, they must provide project effort estimates that are accurate. This situation has motivated researchers to investigate the use of cross-company project data to estimate effort for projects that belong to a single-company, where this company's projects are omitted from model construction [11]. Numerous studies have been conducted to date; however without converging results.

Some have found that cross-company models did not present significantly worse prediction accuracy than single-company models [1],[5],[24],[27],[19],[17]. Some have found that cross-company models did present significantly worse prediction accuracy than single-company models [8],[9],[11],[23],[27]. Others were inconclusive for various reasons.

The normal approach in these studies – and indeed in almost all work in software engineering that builds effort estimation models from historical data – involves separating the data into a training set (from which the model is built) and a validation set (used to assess a model's accuracy). When using leave-one-out cross-validation, each of the N projects in the data set is estimated in turn using a model built from the other $N-1$ points. Alternatively, the data can be allocated randomly or in a stratified way into a single training set (usually of about two thirds of the data) and a single validation set. Bootstrap and k -fold validation fall somewhere in between, involving multiple repetitions of model building and validation on different training and validation sets.

Almost invariably, the assignment of projects to training and validation sets is done without regard to the completion date of the projects. This makes it very likely (in leave-one-out cross-validation it is certain) that the data used to build a model to estimate effort for a given project p include projects that were completed after p was finished.

However, in a real scenario, a developer building a model to estimate effort for a new project could only consider a set of projects already completed. In other words, future projects could not be considered. To reflect this reality when building effort estimation models from historical data, the allocation of projects to training and validation sets should be based on a chronological split. Looking at this issue, three previous studies recently investigated the use of different types of chronological splitting for software effort estimation. Lokan and Mendes (2008a) (CS1) [17], and Lokan and Mendes (CS2) [19] investigated chronological splitting when comparing cross- to single-company effort models. CS1 used a project-by-project chronological split, whereas CS2 used a date-based chronological split. A date-based chronological split consists of choosing a splitting date d , which is used such that both cross- and single-company data sets only contain projects completed prior to d , and the validation set only contains projects started at or after d . Later, Lokan and Mendes (2008b) compared project-by-project splitting to date-based chronological split; however without also taking into account the issue of cross- versus single-company predictions. All of these three studies used the ISBSG dataset release 10, and unanimously found no differences in prediction accuracy between the use of chronological splitting and other commonly used techniques (e.g. leave-one-out cross-validation). Given that these three studies used a single dataset, we conjectured as to whether results would differ if employing a different dataset. Therefore, the aim of this paper is to replicate CS1 using the Finnish dataset.

As in CS1, the research questions addressed by this study are as follows:

1. Using a project-by-project chronological splitting for both cross- and single-company projects:
 - a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?
 - b) How successful is a cross-company model, compared to a single-company model?
2. Using a leave-one-out cross-validation approach:
 - a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?
 - b) How successful is a cross-company model, compared to a single-company model?
3. Using a leave-two-out cross-validation approach:
 - a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?
 - b) How successful is a cross-company model, compared to a single-company model?
4. Does the use of a project-by-project chronological split, instead of a leave-one-out cross-validation, affect the accuracy of the models?
5. Does the use of a project-by-project chronological split, instead of a leave-two-out cross-validation, affect the accuracy of the models?
6. Does the use of a leave-two-out cross-validation, instead of a leave-one-out cross-validation, affect the accuracy of the models?

The main contribution of this paper is therefore to replicate CS1 using a different dataset.

The remainder of the paper is organised as follows: Section 2 briefly summarizes related work. Section 3 describes the research method employed in this study. Results are presented in Section 4, and research questions are addressed in Section 5. A comparison between our results and those from the original study are discussed in Section 6, followed by a Section on threats to the validity, and finally conclusions and directions for future work in Section 7.

2. RELATED WORK

2.1 Cross- versus Single-company Models

Two systematic reviews have been published on how the accuracy of cross-company models compares with that of single-company models [12][21]. Both found ten studies each. Five further papers have since addressed the question [16],[19],[24],[27],[29].

The conclusions of the studies vary, and it is not clear what characteristics of the data sets and analysis methods affect the outcome. Kitchenham et al. [12] note that there are no consistent patterns concerning quality controls on data collection, the quality of the overall study, the size metrics used, the procedure used to build the single-company model, or the strength of the cross-company relationship.

If there is a trend emerging, it is to do with the homogeneity of the data. Kitchenham et al. [12] also note that all studies in which single-company models were significantly better than cross-company models used small single-company data sets; they conjecture that large data sets from a single company may be an indication of the size of the company and the homogeneity of the data set. The range of effort values in the single-company data set, compared to the range in the cross-company data set, also seems important: the greater the difference, the less likely it is that the cross-company model will be accurate for single-company projects.

Premraj and Zimmerman [29] studied the issue of homogeneity by grouping projects by business sector before building cross-company models. While their results varied for different companies, they concluded that it is better to train models using only homogeneous data rather than all data available.

Lokan and Mendes [16] applied Mendes et al.'s (S2a) [25] experimental procedure to the ISBSG database version used in Jeffery et al. (S1a) [9], to assess if differences in experimental procedure would have contributed towards

the different results of [9] and [25]. This work was later extended [24] by applying the experimental procedure of S1a to the data set used in S2a. By investigating the effect of all the variations between S1a and S2a, they concluded that differences in data preparation and analysis procedures did not affect the outcome of the analysis. Thus, the different results of S1a and S2a were due to fundamental differences in the data sets.

Mendes et al. [27] replicated Mendes and Kitchenham [23] study using data on Web projects from the Tukutuku database, volunteered after [23] was conducted. Their results corroborated most of those in [23], except for one of their regression-based cross-company models, which showed significantly similar predictions to the single-company model, thus contradicting the findings from [23].

As previously mentioned in the Introduction, Lokan and Mendes (2008a) [17], and Lokan and Mendes [19] investigated chronological splitting when comparing cross- to single-company effort models. The former used a project-by-project chronological split, whereas the latter used a date-based chronological split. Both found no differences in prediction accuracy between cross- and single-company predictions.

2.2 Use of Chronological Data

Although chronological splitting has been applied as a data-splitting approach in several other domains, few papers in software engineering have used chronological splitting.

Two studies - Lefley and Shepperd [15], and Sentas et al. [30], used a chronological splitting where a set date was chosen and used to split their data; however they did not investigate chronological splitting as a research question in its own right. Lokan and Mendes (2008a) [17], and Lokan and Mendes (2009) [19] investigated as a research question in its own right the use of chronological splitting to compare cross- and single-company predictions. They found that the use of chronological split, and the use of two different dates to split the data chronologically, did not seem to improve the accuracy of either single- or cross-company estimates. Lokan and Mendes (2008b) also investigated the use of chronological splitting as a research question in its own right; however they did not compare cross- to single-company predictions.

Auer and Biffi [2] and Auer et al. [3] considered chronology directly in their research into estimation by analogy. They tracked changes in accuracy as the portfolio of completed projects grew. However, they did not consider chronological splitting vs. other methods as a separate research question.

We are also aware of one study, by Kitchenham et al. [13], which considered a growing data set and whether a sliding window should be used. They found that when they ordered their projects by start date and then divided them into four equal-sized subsets, the regression models changed between the subsets. As a result they argued that old projects should be removed from the data set as new ones were added, so that the size of the data set remained constant. They recommended that the estimate for project n should be based on projects $n-30$ to $n-1$: a sliding window of 30 projects.

Another two studies have addressed a slightly different problem, using information from previous phases within a project to estimate effort in later tasks in the same project. MacDonell and Shepperd [20] studied effort distribution in major waterfall phases in 16 projects. They found the patterns of effort distribution between phases varied too much for this approach to work by itself, though it was helpful in conjunction with expert estimation. Abrahamsson et al. [1] were more successful using two projects developed using an extreme programming environment, finding that data from early iterations produced increasingly accurate effort estimates for later iterations.

The chronology of projects was also important in [28], which investigated changes in software development productivity over time. The focus in that paper was on characterising productivity in different years, rather than using past data to estimate future projects.

3. RESEARCH METHOD

3.1 Data set Description

The analysis presented in this paper was based on software projects from the Finnish dataset (the version of that data set available as at May 2008). This dataset contains data on 856 projects, including 201 from the single company that we studied. Rules of confidentiality enabled the single company's identity to be unknown.

To form a data set suitable for our analysis, we removed projects according to the following criteria:

- Remove projects if they were assigned a low data quality rating (X).
- Remove any duplicate projects.
- Remove projects if their size is measured in COSMIC, rather than FiSMA FPs.

We finished with a set of 794 software projects, where 201 projects came from a single company, and 593 came from other companies. All have high data quality, and comparable definitions for size and effort. The project selection criteria we applied were carried out carefully in order to maximise comparability between projects.

The Finnish dataset provides data on many variables. The fundamental variables are size, effort, and four basic project classifiers: development type, hardware platform, development language, and business sector. Other variables include several situation analysis variables, analogous to general system characteristics in IFPUG function points, or to COCOMO cost drivers. In this study we restricted our attention to the six fundamental variables. Over 50 development languages are represented; we consolidated these into a broad classification by language type. Our final variables are presented in Table 1.

Summary statistics for the ratio-scale variables are presented in Table 2. We also include the project delivery rate (PDR), calculated as Effort/Size, to provide readers an additional way to compare cross- to single-company

projects. This measure is often used to measure productivity, where high values indicate low productivity. Table 2 suggests that size, effort and PDR are all slightly smaller in the single-company data than in the cross-company data.

TABLE 1: Variables used in this study

Variable	Scale	Description
Effort	Ratio	Project effort in person hours
Size	Ratio	Application size in FiSMA function points
LangType	Nominal	Language type: 3GL, 4GL, application generator
DevType	Nominal	Development type: New development, enhancement (including integration and conversion), maintenance, other
Platform	Nominal	The type of hardware the system was developed for: mainframe, midrange, multi-platform, PC-network, standalone PC, other
Sector	Nominal	The business sector the system was developed for: banking, communication, insurance, manufacturing, public administration, wholesale and retail trade, other

TABLE 2: Project summary statistics for the ratio-scaled variables

SC data - 201 projects						
Variable	N	Mean	Median	StDev	Min	Max
Size	201	379	261	401	9	3375
Effort	201	2528	1470	3850	86	41643
PDR	201	7.01	6.33	4.60	0.72	46.47
CC data - 593 projects						
Variable	N	Mean	Median	StDev	Min	Max
Size	593	550	292	939	6	13682
Effort	593	4017	1688	6531	42	67576
PDR	593	10.44	6.67	44.21	0.38	1067.0

We do not present group statistics for the nominal variables because their description would use a large amount of space. To summarise, the single-company projects include fewer new developments (56% compared to 70% for cross-company projects), more mainframe projects (62% compared to 41%), and more 3GL projects (82% compared to 70%). They are almost entirely from the insurance sector, whereas the cross-company projects are varied (though insurance is still the most common sector).

The single-company and cross-company projects differ notably in age: although the overall range of dates is similar, only 8 of 201 single-company projects started before 1998, compared to 232 of 593 cross-company projects.

3.2 Project-by-project Chronological Splitting

Project-by-project chronological splitting was used both with the single- and cross-company data sets.

When employed with the single-company data set to obtain predictions for the single-company projects, the project-by-project chronological splitting was used as follows:

- A project p in the single-company data set was selected as the target project, for which estimated effort was to be obtained.
- The starting date (sd) for p was used to split the remaining single-company projects into two groups:
 - Completed*: projects that had finished prior to sd
 - Active/future*: projects that were active or had not yet started at sd
- The set of *completed* projects was used as the training set in order to build a regression model R
- Cook's distance [7] was used to determine whether any highly influential *completed* projects should be removed; if any were removed, R was then refitted using the reduced data set
- R was applied to p 's data in order to obtain an effort estimate for p
- Project p was returned to the single-company data set.
- Steps a to f were repeated until effort estimates were obtained for all the projects in the single-company data set.

When employed with the cross-company data set to obtain predictions for the single-company projects, the procedure was the same, except that cross-company projects replaced single-company projects as the training set.

3.3 Modelling Techniques

All the models used in this investigation were built using an automated process programmed in the statistical programming language R.

The procedure followed in the automated process was as follows:

- The first step in building every regression model was to ensure numerical variables were normally distributed. We used the Shapiro-Wilk test on both the training and validation sets to check if Effort and Size were normally distributed. Statistical significance was set at $\alpha = 0.05$. In every case, Size and Effort were not normally distributed, and were therefore transformed to a natural logarithmic scale. The transformed variables' names were preceded by 'L'; so Effort became LEffort, and Size became LSize.

- Models were built using backward stepwise multivariate regression.
- The size of the training set was always considered: no model was investigated involving more than N/10 independent variables, for a training set of N projects.
- To verify the stability of the effort model, the automatic process used the following step [11]:
 - Calculate Cook's distance values for all projects to identify influential data points. Any projects with distances higher than $3 \times (4/N)$, where N represents the total number of projects, are immediately removed from the data analysis. Those with distances higher than $4/N$ but smaller than $(3 \times (4/N))$ are removed temporarily in order to test the model stability, by observing the effect of their removal on the model. If the model coefficients remain stable and the goodness of fit improves, the highly influential projects are retained in the data analysis [16].

3.3 Prediction Accuracy Measures

To date the four measures most commonly used in software engineering to compare different effort estimation techniques have been [6]:

- Magnitude of Relative Error (MRE).
- Mean Magnitude of Relative Error (MMRE).
- Median Magnitude of Relative Error (MdMRE).
- Prediction at level I ($\text{Pred}(I)$).

MRE is defined as:

$$\text{MRE} = \frac{|e - \hat{e}|}{e} \quad (1)$$

where e represents actual effort and $\hat{e}$ estimated effort. MMRE is the mean of all MREs.

An alternative to the mean is the median, which also represents a measure of central tendency; however it is less sensitive to extreme values. The median of MRE values is called the MdMRE.

The Prediction at level I , also known as $\text{Pred}(I)$, measures the fraction of estimates that are within I % of the actual values. Suggestions have been made [6] that I should be set at 25% and that a good prediction system should offer this accuracy level 75% of the time.

Although MMRE, MdMRE and $\text{Pred}(I)$ are often used as evaluation criteria to compare different effort estimation techniques, Kitchenham et al. [12] showed that MMRE and $\text{Pred}(I)$ are respectively measures of the spread and kurtosis of z , where $(z = \hat{e} / e)$. They suggested the use of box plots of z and box plots of the residuals $(\hat{e} - e)$ as useful alternatives to simple summary measures since they can give a good indication of the distribution of residuals and z and can help explain summary statistics such as MMRE and $\text{Pred}(.25)$.

We use MMRE, MdMRE, z and $\text{Pred}(.25)$ to compare the effort models built in this study, and test for significant differences in absolute residuals.

To compare the statistical significance of predictions we used the paired-samples t-test, and set the statistical significance at 5%. All calculations were carried out using the statistical language R.

4. RESULTS

The single-company data set contained data on 201 projects. However, with project-by-project chronological splitting we only built regression models whenever there were at least 12 training projects that had been completed prior to start of the single-company project for which effort was to be estimated. Using single-company training data we were only able to obtain predictions for 183 of the 201 single-company projects; using cross-company training data we could obtain predictions for 199 of the 201 single-company projects. For comparability, all models were evaluated using the 183 projects that could be estimated using project-by-project chronological splitting with single-company training data.

4.1 Cross-company Model

The cross-company models used as part of the project-by-project chronological splitting procedure will not be described in this Section as there were 199 different models that were automatically fit using the statistical language R.

Therefore, the only cross-company model that will be detailed here is the one to be compared with predictions obtained from applying leave-one-out and leave-two-out cross-validations in the single-company data set.

In the cross-company data set, the categorical variables Platform, DevType, LangType and Sector had six, three, three and seven levels respectively. They were each replaced by five, two, two, and six dummy variables respectively (the dummy variable corresponding to the most common level was omitted). In addition, Effort and Size were transformed to a natural logarithmic scale to approximate a normal distribution. Table 3 presents the final set of variables used to build the cross-company model.

199 of 201 single-company projects provided data for every variable. The best cross-company model, based on the 577 out of 593 cross-company projects which also had data for all variables), included all of size, development type, platform, language type, and sector. Its adjusted R^2 was 0.754 (see Equations 2 and 3).

TABLE 3: Variables used to build the cross-company model

LEffort	Natural logarithm of Effort, measured in hours.
LSize	Natural logarithm of Size, measure in FiSMA FPs.
DevTypeEnh	Dummy variable where 'enhancement', 'conversion' and 'integration' are coded as 1 and others are coded as 0
DevTypeMaint	Dummy variable where 'annual maintenance' and 'maintenance redo' are coded as 1 and others are coded as 0
DevTypeOther	Dummy variable where development type 'other' is coded as 1 and others are coded as 0
PlatformMulti	Dummy variable where 'multi-platform' is coded as 1 and others are coded as 0
PlatformPC	Dummy variable where 'standalone pc' platform is coded as 1 and others are coded as 0
PlatformPCNet	Dummy variable where 'PC-network' is coded as 1 and others are coded as 0
PlatformMR	Dummy variable where 'Midrange' platform is coded as 1 and others are coded as 0
PlatformOther	Dummy variable where 'Other' platform is coded as 1 and others are coded as 0
LangTypeApg	Dummy variable where 'Application generator' language type is coded as 1 and others are coded as 0
LangType4GL	Dummy variable where '4GL' language type is coded as 1 and others are coded as 0
SectorBank	Dummy variable where 'Banking' is coded as 1 and others are coded as 0
SectorComms	Dummy variable where 'Communication' and 'Telecommunication' are coded as 1 and others are coded as 0
SectorManufacturing	Dummy variable where 'Manufacturing' is coded as 1 and others are coded as 0
SectorPublic	Dummy variable where 'Public administration' is coded as 1 and others are coded as 0
SectorRetail	Dummy variable where 'Wholesale & retail trade' is coded as 1 and others are coded as 0
SectorOther	Dummy variable where 'Insurance' or any of the values specified above is coded as 0 and others are coded as 1

$$LEffort = 2.473 + 0.950 LSize - 0.207 LangType4GL - 0.209 DevTypeMaint - 0.516 PlatformMR - 0.721 SectorManufacturing - 0.549 SectorPublic - 0.598 SectorRetail - 1.008 SectorOther \quad (2)$$

which, when transformed back to the raw data scale, gives the Equation:

$$Effort = 11.852 Size^{0.950} e^{-0.207 LangType4GL} e^{-0.209 DevTypeMaint} e^{-0.516 PlatformMR} e^{-0.721 SectorManufacturing} e^{-0.549 SectorPublic} e^{-0.598 SectorRetail} e^{-1.008 SectorOther} \quad (3)$$

One single-company project did not provide data about the platform. The best cross-company model, based on the 588 cross-company projects that had data for the same set of variables, was (after transformation back to the raw scale):

$$Effort = 13.630 Size^{0.924} e^{-0.261 DevTypeMaint} e^{-0.783 SectorManufacturing} e^{-0.604 SectorPublic} e^{-0.621 SectorRetail} e^{-0.976 SectorOther} \quad (4)$$

Finally, one single-company project did not provide data about the development language. The best cross-company model, based on the 582 cross-company projects that had data for the same set of variables, was (after transformation back to the raw scale):

$$Effort = 10.375 Size^{0.962} e^{0.625 PlatformMR} e^{-0.732 SectorManufacturing} e^{-0.581 SectorPublic} e^{-0.592 SectorRetail} e^{-0.925 SectorOther} \quad (5)$$

4.2 Cross-company Models applied to Single-company projects

Two different sets of prediction accuracy statistics are presented in Table 4:

1. Predictions based on the application of project-by-project chronological split to each of the automatically fit cross-company models (CC1), applied to each of the 201 single-company projects. Each of the cross-company models was built from scratch for each single-company project, using a procedure automated in the statistical language R.
2. Predictions based on the application of the cross-company model represented by Equation 4 (CC2) to the same 201 single-company projects. (The absolute residuals and z values resulting from CC2 will later be compared to those obtained from leave-one-out and leave-two-out cross-validations.)

TABLE 4: Prediction accuracy Statistics for cross-company models applied to single-company projects

Accuracy statistics	Cross-company Effort prediction models	
	CC1	CC2
Mean absolute residuals	1184	991.4
Median absolute residuals	895	693
MMRE	0.952	0.8271
MdMRE	0.5191	0.4376
Mean z	1.86	1.718
Median z	1.444	1.354
Pred(.25)	0.2896	0.3388

Overall both CC1 and CC2 presented poor prediction accuracy (high mean and median MREs, and low Pred(.25) values). However, CC2's median MRE and Pred(.25) seemed better than the values obtained for CC1. This trend was confirmed by the Paired T-Test, where CC2 was found to provide significantly superior predictions than CC1.

4.3 Single-company Models

Three different sets of prediction accuracy statistics are presented in Table 5:

1. Predictions based on project-by-project chronological splitting, using single-company data, for each of the same 201 single-company projects. Each of the single-company models (SC1) was built from scratch, using a procedure automated in the statistical language R.
2. Predictions based on the use of leave-one-out cross-validation applied to the 201 single-company projects, where each of the single-company regression models (SC2) was built from scratch, using a procedure automated in the statistical language R.
3. Predictions based on the use of leave-two-out cross-validation applied to the 201 single-company projects, where each of the single-company regression models (SC3) was built from scratch, using a procedure automated in the statistical language R.

The training set was small for early projects in the chronological sequence, but by the end of the sequence it was almost the complete data set.

TABLE 5: Prediction accuracy Statistics for single-company models applied to single-company projects

Accuracy statistics	Single-company effort prediction models		
	SC1	SC2	SC3
Mean absolute residuals	924.3	808.5	809.4
Median absolute residuals	501.7	486.3	486.3
MMRE	0.606	0.5573	0.5584
MdMRE	0.3227	0.2967	0.2967
Mean z	1.28	1.273	1.274
Median z	0.9546	0.9973	0.9974
Pred(.25)	0.3989	0.4372	0.4372

Overall all single-company models presented poor prediction accuracy (high mean and median MREs, and low Pred(.25) values). However, accuracy statistics for SC2 and SC3 were in general very similar and better than the accuracy presented by SC1. This trend was confirmed by the Paired T-Test, when applied to absolute residuals: SC1 was found to present significantly worse accuracy than either SC2 or SC3, and SC2 and SC3 were found to present similar accuracy.

4.4 Comparing Single- to Cross-company Predictions

Table 6 shows the results for the statistical significance tests comparing predictions between single- and cross-company predictions. All results were obtained using the Paired T-test ($\alpha = 0.05$), and absolute residuals. Absolute residuals unanimously indicate significant differences in prediction accuracy between cross- and single-company models, where superior predictions were presented by all three single-company models.

TABLE 6: Comparing prediction accuracy between single- and cross-company models

Cross-company model(s)	Single-company model(s)	Statistically significantly different based on absolute residuals?
CC1	SC1	Yes, SC1 is superior to CC1
CC2	SC2	Yes, SC2 is superior to CC2
CC2	SC3	Yes, SC3 is superior to CC2

5. ANSWERING OUR RESEARCH QUESTIONS

Research questions 1a), 2a) and 3a) from Section 1 ask how accurate cross-company models are for single-company data. These questions are addressed in Table 4. None of the estimates obtained for the single-company projects using cross-company models indicate good prediction accuracy. MMRE values range between 0.952 and 0.827, which are considered poor (0.25 is considered "good" [6]). The same applies to Pred(.25), which ranges from 0.2896 to 0.3388 (0.75 indicates a good prediction model).

Research questions 1b), 2b) and 3b) compare cross- and single-company models applied to single-company data. These questions are addressed in Table 6. Absolute residuals show significant differences between all the cross- and single-company predictions, when used to estimate effort for single-company projects. Single-company models unanimously present significantly superior predictions than cross-company models when estimating effort for single-company data.

To answer the fourth research question, comparing project-by-project chronological splitting with leave-one-out cross-validation, we used the Paired T-test to compare absolute residuals between models CC1 and CC2 and models SC1 and SC2. No significant differences were found when comparing single-company models; however

CC2 showed significantly better predictions than CC1. So the answer to question four is 'yes' if based on single-company models and 'no' otherwise.

Question five (comparing leave-two-out cross-validation, rather than leave-one-out cross-validation, with project-by-project chronological splitting) had the same answer as question four: the answer to question five is 'yes' if based on single-company models and 'no' otherwise.

To answer question six, comparing the two cross-validation approaches, we also used the Paired T-test to compare absolute residuals between models SC2 and SC3. No significant differences were found, therefore the answer to this question is 'no'.

In terms of the comparison between cross- and single-company predictions, results showed that single-company models significantly outperformed cross-company models. The two clear differences between the single-company and cross-company projects are that the single-company projects are more homogeneous in both business sector and age. In other respects the two data sets are both large and diverse; our other studies have found single- and cross-company models based on data with these characteristics to be similar in accuracy. This suggests that similar business sector, or age, or both, are particularly important when considering the use of cross-company data for estimation.

In relation to the use of project-by-project chronological splitting, our results showed that in the single-company data set it either showed similar accuracy to, or significantly worse than, both leave-one and leave-two out cross-validations. Also, project-by-project chronological estimation using cross-company data showed significantly worse accuracy than regression models based on the entire cross-company data set. In both situations, estimates that use the whole data set as training data are more accurate than estimates that can only use the data available so far for each individual project. In other words, research methods that draw on the entire data set to build estimation models are likely to over-estimate the accuracy that could be obtained in practice by estimators who can only make use of data available on a project-by-project basis. This is the most important finding in this paper: it suggests that it is important for researchers in this field to consider chronology, if the research results are to be relevant to industry.

6. COMPARISON WITH PREVIOUS RESULTS

In this Section we compare the results presented in this paper to those from [17], using as basis the same six research questions addressed in both studies.

1. Using a project-by-project chronological splitting for both cross- and single-company projects:

- a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?

Both studies showed the same trend: the estimates obtained for the single-company projects using cross-company models indicated poor prediction accuracy. MMRE values ranged between 0.952 and 0.982, considered poor (0.25 is considered "good" [6]); Pred(.25) ranged between 0.184 and 0.290 (0.75 indicates a good prediction model), also considered poor. Therefore, the results of the replicated study corroborated those from the original study.

- b) How successful is a cross-company model, compared to a single-company model?

Lokan and Mendes [17] showed no significant differences between the predictions obtained for single company projects using cross-company data and predictions obtained using single-company data; however these results were not corroborated by those reported herein, where predictions obtained for single company projects using single-company data presented significantly superior accuracy than predictions using cross-company data. Therefore, the results of the replicated study did not corroborate those from the original study.

2. Using a leave-one-out cross-validation approach:

- a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?

Both studies presented a similar trend, where the prediction accuracy using cross-company data to estimate effort for single company projects was poor. MMRE values ranged between 0.827 and 1.477, and Pred(.25) ranged between 0.180 and 0.339. Therefore, the results of the replicated study corroborated those from the original study.

- b) How successful is a cross-company model, compared to a single-company model?

The same pattern observed for 1.b) occurred here too. Therefore, the results of the replicated study did not corroborate those from the original study.

3. Using a leave-two-out cross-validation approach:

- a) How successful is a cross-company model at estimating effort for projects from a single company, when the model is built from a data set that does not include projects from that company?

Once again both studies presented a similar trend, where the prediction accuracy using cross-company data to estimate effort for single company projects was poor. MMRE values ranged between 0.8272 and 1.477, and Pred(.25) ranged between 0.180 and 0.339. Therefore, the results of the replicated study corroborated those from the original study.

- b) How successful is a cross-company model, compared to a single-company model?

The same pattern observed for 1.b) and 2.b) was repeated here. Therefore, the results of the replicated study did not corroborate those from the original study.

4. Does the use of a project-by-project chronological split, instead of a leave-one-out cross-validation, affect the accuracy of the models?

Lokan and Mendes [17] reported no significant differences between predictions obtained using project-by-project chronological split and leave-one-out cross-validation. The same pattern was observed in this study when comparing SC1 to SC2; however the pattern differed when comparing cross-company models given that CC2 showed significantly better predictions than CC1. Therefore, the results of the replicated study partially corroborated those from the original study.

5. Does the use of a project-by-project chronological split, instead of a leave-two-out cross-validation, affect the accuracy of the models?

The same results as for Research question 4. Therefore, the results of the replicated study partially corroborated the results of the original study.

6. Does the use of a leave-two-out cross-validation, instead of a leave-one-out cross-validation, affect the accuracy of the models?

Both studies showed no significant differences between predictions obtained using leave-one or leave-two out cross-validation. Therefore, the replicated study corroborated the findings from the original study.

An important difference between the single-company and cross-company datasets used in [17] and herein relates to the project delivery rate (PDR). The single-company projects from the ISBSG dataset presented similar PDR to the cross-company projects; conversely, the single-company projects from the Finnish dataset presented superior PDR when compared to the cross-company projects. This may help explain the differences observed between both studies for this research questions 1.b), 2.b) and 3.b).

7. THREATS TO THE VALIDITY

This study has some limitations and threats to validity. First, the Finnish dataset is a convenience sample, and does not represent a random sample of projects. Therefore these results are only applicable to those companies that volunteered data to the Finnish dataset, and companies that manage software projects similar to those used in this study.

Second, the models 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, log 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 adjusted R² is assumed to be preferred; multi-collinearity between independent variables is handled automatically in the stepwise procedure used to build the models. Based on our past experience with manual model building (from ISBSG data), we believe that these assumptions are acceptable. One would not want to base significant decisions on a single model built automatically, without at least doing some serious manual checking. But for calculations such as leave-one-out/leave-two-out cross-validation, or project-by-project chronological estimation across a substantial data set, we believe that the process here is reasonable.

8. CONCLUSIONS

This paper replicated a previous study and investigated the use of project-by-project chronological splitting when comparing cross- and single-company prediction models, and compared it with two cross-validation approaches. Data from the Finnish dataset was used to answer our research questions.

When comparing cross- and single-company predictions, we found that single-company models presented significantly better prediction accuracy than cross-company models. These results contradict those obtained using the ISBSG dataset, where cross-company predictions showed similar accuracy to single-company predictions.

When using project-by-project chronological splitting, these presented either similar accuracy or significantly worse accuracy than other two commonly used techniques (leave-one-out and leave-two-out cross-validation). Except for the comparison between CC1 and CC2, all remaining results (SC1 significantly worse than SC2 & SC3; SC2 presented similar accuracy to SC3) were common between this paper and the previous study it replicated. However, in a real setting, the only realistic option available for a company that employs its own data is to use some form of chronological splitting. If, however, a company relies on cross-company data, the results presented herein suggest that the use of common techniques such as leave-one-out cross-validation for effort estimation may be more beneficial than to apply a project-by-project chronological split.

We found no difference between the results from leave-one-out and leave-two-out cross-validation, which corroborates previous results. Leave-two-out cross-validation has the advantage of providing a distribution of estimates for each project, rather than a single estimate; it has the disadvantage that it involves a great deal more computation, proportional to the square of the size of the data set instead of linear in the size of the data set. Our results suggest that the computational cost of leave-two-out cross-validation is not worthwhile.

Future work includes repeating this experimental approach using other data sets; particularly, more homogeneous data sets, including Web projects. We are also interested in tracing the evolution of prediction models and their accuracy as a training set grows. Another research question to be investigated is whether it is best to use the

entire history of past projects for project-by-project estimation, or whether it is more appropriate in a rapidly-changing world to use a window of recent projects.

REFERENCES

- [1] Abrahamsson, P., R. Moser, W. Pedrycz, A. Sillitti and G. Succi (2007) Effort Prediction in Iterative Software Development Processes – Incremental Versus Global Prediction Models. Proceedings of ESEM 2007, Madrid, IEEE Computer Society, pp 344-353, September.
- [2] Auer, M. and S. Biffl (2004): Increasing the Accuracy and Reliability of Analogy-based Cost Estimation with Extensive Project Feature Dimension Weighting, Proceedings of ISESE'04, pp 147-155, August.
- [3] Auer, M., A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl (2006): Optimal Project Feature Weights in Analogy-based Cost Estimation: Improvement and Limitations, *IEEE Trans. Software Engineering*, 32(2), pp 83-92, February.
- [4] Briand, L.C., K. El-Emam, K. Maxwell, D. Surmann and I. Wieczorek (1999) An assessment and comparison of common cost estimation models. Proceedings 21st International Conference on Software Engineering, ICSE'99, pp 313-322, May.
- [5] Briand, L.C., T. Langley and I. Wieczorek (2000) A replicated assessment of common software cost estimation techniques. Proceedings of 22nd International Conference on Software Engineering, ICSE'00, pp 377-386, June.
- [6] Conte, S. D., H.E. Dunsmore and V.Y. Shen (1986) *Software Engineering Metrics and Models*, Benjamin-Cummins.
- [7] Cook, R.D. Detection of influential observations in linear regression (1977) *Technometrics*, 19, pp 15-18.
- [8] Jeffery, R., M. Ruhe and I. Wieczorek (2000) A Comparative Study of Two Software Development Cost Modeling Techniques using Multi-organizational and Company-specific Data. *Information and Software Technology*, 42, pp 1009-1016.
- [9] Jeffery, R., M. Ruhe and I. Wieczorek (2001) Using public domain metrics to estimate software development effort. Proceedings of 7th IEEE International Symposium on Software Metrics (Metrics'01), London, pp 16-27, April.
- [10] Kitchenham, B.A. A procedure for analysing unbalanced data sets (2004) *IEEE Trans. Software Engineering*, 24(4), pp 278-301, April.
- [11] Kitchenham, B.A. and E. Mendes (2004) A Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications, Proceedings EASE 2004, pp 47-55.
- [12] Kitchenham, B.A., E. Mendes and G. Travassos (2007) Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *IEEE Trans. Software Engineering*, 33(5), pp 316-329, May.
- [13] Kitchenham, B., S.L. Pfleeger, B. McColl and S. Eagan (2002) An empirical study of maintenance and development estimation accuracy. *Journal of Systems and Software* 64(1), pp 57-77.
- [14] Kitchenham, B.A., L.M. Pickard, S.G. MacDonell and M.J. Shepperd (2001) What accuracy statistics really measure. *IEE Proceedings - Software*, 148(3), pp 81-85, June.
- [15] Lefley, M. and M.J. Shepperd (2003) Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets, Proceedings of GECCO 2003, LNCS 2724, Springer-Verlag, pp 2477-2487, July.
- [16] Lokan, C. and E. Mendes (2006) Cross-company and Single-company Effort Models Using the ISBSG Database: a Further Replicated Study, Proceedings of ISESE'06, pp 75-84, September.
- [17] Lokan, C. and E. Mendes (2008a) Cross-company and Single-company Effort Models Using Chronological Splitting, Proceedings of EASE 2008, pp. 151-160.
- [18] Lokan, C. and E. Mendes (2008b) Investigating the use of chronological split for software effort estimation. Submitted journal paper.
- [19] Lokan, C. and E. Mendes (2009) Using Chronological Splitting to compare Cross- and Single-company Effort Models: Further Investigation, Proceedings of ACSC'09, Wellington, New Zealand, January.
- [20] MacDonell, S.G. and M.J. Shepperd (2003) Using Prior-Phase Effort Records for Re-estimation During Software Projects. Proceedings of 9th IEEE International Symposium on Software Metrics (Metrics'03), Sydney, September.
- [21] MacDonell, S.G. and M.J. Shepperd (2007) Comparing Local and Global Software Effort Estimation Models -- Reflections on a Systematic Review. Proceedings of First International Symposium on Empirical Software Engineering and Measurement, pp 401-409.
- [22] Maxwell, K. (2002). *Applied Statistics for Software Managers*. Software Quality Institute Series, Prentice Hall.
- [23] Mendes, E. and B.A. Kitchenham (2004) Further Comparison of Cross-Company and Within Company Effort Estimation Models for Web Applications. Proceedings 10th IEEE International Symposium on Software Metrics (Metrics'04), Chicago, Illinois, IEEE Computer Society, pp 348-357, September.
- [24] Mendes, E. and C. Lokan (2008) Replicating Studies on Cross- vs. Single-company Effort Models using the ISBSG Database. *Empirical Software Engineering*, 13(1): 3-37. February.
- [25] Mendes, E., C. Lokan, R. Harrison and C. Triggs (2005) A Replicated Comparison of Cross-company and Within-company Effort Estimation models using the ISBSG Database. Proceedings of 11th IEEE International Symposium on Software Metrics (Metrics'05), Como, Italy, September.
- [26] Mendes, E., N. Mosley and S. Counsell (2003) A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation, Proceedings of ISESE'2003, Rome, pp 100-109, September.
- [27] Mendes, E., Di Martino, S, Ferrucci, F., Gravino (2008) C. Cross-company vs. Single-company Web effort models using the Tukutuku Database: an Extended Study, *Journal of Systems and Software*, 81, pp. 673-690.
- [28] Premraj, R., M.J. Shepperd, B.A. Kitchenham and P. Forselius (2005) An Empirical Analysis of Software Productivity Over Time. Proceedings of 11th IEEE International Symposium on Software Metrics (Metrics'05), Como, Italy, September.
- [29] Premraj, R and T. Zimmerman (2007) Building Software Cost Estimation Models using Homogenous Data. Proceedings of ESEM 2007, Madrid, IEEE Computer Society, pp 393-400, September.
- [30] Sentas, P., Angelis, L., Stamelos, I., Bleris, G. (2005) Software productivity and effort prediction with ordinal regression, *Information and Software Technology*, 47, pp 17-29.
- [31] Wieczorek, I. and M. Ruhe (2002) How valuable is company-specific data compared to multi-company data for software cost estimation? In Proceedings of 8th IEEE International Symposium on Software Metrics (Metrics'02), Ottawa, pp 237-246, June.