Generation of Mimic Software Project Data Sets for Software Engineering Research

Maohua Gan
Graduate School of Natural Science and Technology
Okayama University
Okayama, Japan
pa2i5772@s.okayama-u.ac.jp

Kentaro Sasaki
Previously at Faculty of Engineering
Okayama University
Okayama, Japan
ken.default.0828@gmail.com

Akito Monden
Graduate School of Natural Science and Technology
Okayama University
Okayama, Japan
monden@okayama-u.ac.jp

Zeynep Yucel
Graduate School of Natural Science and Technology
Okayama University
Okayama, Japan
zeynep@okayama-u.ac.jp

***Abstract*—To conduct empirical research on industry software development, it is necessary to obtain data of real software projects from industry. However, only few such industry data sets are publicly available; and unfortunately, most of them are very old. In addition, most of today's software companies cannot make their data open, because software development involves many stakeholders, and thus, its data confidentiality must be strongly preserved. This paper proposes a method to artificially generate a "mimic" software project data set whose characteristics (such as average, standard deviation and correlation coefficients) are very similar to a given confidential data set. The proposed method uses the Box–Muller method for generating normally distributed random numbers, then, exponential transformation and number reordering are used for data mimicry. Instead of using the original (confidential) data set, researchers are expected to use the mimic data set to produce similar results as the original data set. To evaluate the usefulness of the proposed method, effort estimation models were built from an industry data set and its mimic data set. We confirmed that two models are very similar to each other, which suggests the usefulness of our proposal.**

***Keywords*— empirical software engineering, data confidentiality, software effort estimation, data mining**

I. Introduction

In the research field of empirical software engineering, researchers demand for data of real software development projects from industry. However, only few industry data sets are publicly available. Also, these data sets are quite old, which becomes a great problem in ensuring the validity and reliability of the research. For example, tera-Promise repository [13] provides several industry data sets such as Desharnais [7], COCOMO '81 [4], Kemerer [9], Albrecht [1], but these data were recorded in the 1980's; thus, the development environments and processes may greatly differ from modern software development. In addition, the sample size is often very small, e.g. Kemerer has only 15 projects and Albrecht has only 24 projects. Surprisingly, these old and small data sets are still actively used in recent research papers in top journals (e.g. [2][11][16]) due to lack of new industry data sets.

Meanwhile, although many companies measure and accumulate data of recent software development projects, it becomes more and more difficult for university researchers to use them for the research because the legal compliance to various data protection regulations has become extremely important for todays' companies. Moreover, since software development involves many stakeholders, their data confidentiality must be strongly preserved; thus, it became more difficult to take the data out of the company. In addition, although there are some studies performed using the latest software development data, only their analysis results are disclosed and the data itself is not disclosed. For example, the white paper on software development data in 2016-2017 [17] provides various analysis results of 4046 software development projects held in 31 Japanese software development companies; however, the data set itself is not disclosed.

In this paper, to make it possible for academic researchers to use the confidential software project data set of a company, we propose a method to artificially create a mimic data set that has very similar characteristics to a given confidential data set. Instead of using the original (confidential) data set, researchers are expected to use the mimic data set to produce similar results as the original data set. For example, researchers can use the mimic data set for the purpose of evaluation of software effort estimation methods, because many industry data sets are required for the evaluation of the stability assessment of the methods [16]. Moreover, such a mimic data set is also useful to practitioners because many companies want to compare their software development performance (such as productivity and defect density) with other companies.

As a basic idea of our proposal, we measure statistics of each variable as well as correlation coefficients between all pairs of variables in a confidential data set. Next, to produce a mimic variable, we use the Box–Muller method [5] for generating normally distributed random numbers; then, exponential transformation is applied to the generated values to mimic the value distribution of the original variable. After generating all mimic variables, number reordering is applied to the generated values to mimic the correlation coefficients between all pairs of original variables.

Interestingly, our method can freely determine the number of data points to generate. For example, we could produce a data set of sample size n = 1000, which means 1000 projects, from an original data set with much smaller samples, e.g. n = 30. This

TABLE I. AN EXAMPLE OF SOFTWARE PROJECT DATA SET (EXCERPT FROM DESHARNAIS DATA SET [7])

PM experience	Team experience	Language A	FP	Duration	Effort (person-hours)
1	1	0	140	5	2520
7	4	0	113	13	1603
3	1	1	291	8	3626
3	1	1	67	10	1267
4	0	0	99	6	546
5	4	1	645	26	9100

also means that there is no one-to-one mapping of projects between the original data set and the mimic data set. Therefore, data privacy and confidentiality are effectively protected even if the mimic data set is made open.

In contrast, conventional data anonymizing methods for software engineering data employ *data mutation* techniques to gain data privacy [14][15]. Since data mutation keeps the one-to-one mapping of data points between the anonymized data set and the original data set, threats of breaking the anonymity cannot be perfectly prevented. Moreover, since strong data mutation yields change of data characteristics, balancing privacy and utility is a big challenge in this approach [15]. On the other hand, our mimic data set is composed of randomly generated data points without keeping the one-to-one mapping to the original data set, data anonymization is much more effectively achieved. We believe that companies are more confident in using our method than using the data mutation to comply with various data protection regulations.

To evaluate the utility of the proposed method, this paper presents a case study of producing a mimic data set from Deshanais data set [7], which is one of the most frequently used data sets in software effort estimation study [10]. In the case study, we built effort estimation models from both the original data set and the mimic data set to see whether we could obtain similar results from both data sets.

II. RELATED WORK

Peters et. al [14] proposed a data anonymization method called MORPH to solve privacy issues in software development organizations. They target defect prediction research and try to anonymize the defect data set that consists of various software metrics measured for each source file of a software product. They use data mutation techniques, which add small amount of changes to each value to make it difficult to identify a specific source file in a data set. They further propose a method called CLIFF, which allows to eliminate some data points that are not necessary for the defect prediction. Combining CLIFF with MORPH, they try to balance privacy and utility of defect data sets [15].

Since their approach is specially proposed for two group classification problem (i.e., distinguishing defect-prone files and not defect-prone files in a defect data set), it cannot be applied to general purpose data sets such as software project data sets that we target in this paper. In addition, since data mutation keeps the one-to-one mapping of data points between the anonymized data set and the original data set except for eliminated ones, threats of breaking the anonymity cannot be perfectly prevented. In contrast, we try to produce a completely artificial data set from given characteristics of a confidential data set.

III. THE PROPOSED METHOD

A. Basic Idea and Procedure

In this paper, a confidential data set that needs to be kept secret is called a “source data set” or a simply “source data.” And, the artificially generated data to mimic the source data is called a “mimic data set” or “mimic data.”

As source data, we target software project data sets. Table I shows a part of Desharnais data set [7], which is one of the commonly used software project data sets for effort estimation studies. In Table I, “PM” stands for “project manager” and “FP” stands for “function point.” Many software companies record similar data sets that consist of various project features. In this paper we assume that there is no missing value in a data set.

Typically, software project data sets contain software size metrics such as Function Point (FP) and Source Lines of Code (SLOC), as well as the project length (often denoted as “duration”), and the development effort. It has been known that the probability distribution of these variables roughly follows log-normal distribution [10]. Therefore, this paper approximates the value distribution of quantitative variables by the log-normal distribution.

After setting the number of cases n to be generated in the mimic data, the procedure to generate mimic data from source data is described as follows:

- Step 1: For each ratio scale or interval scale variable in the source data, generate a set of artificial values whose distribution is similar to the source data.
- Step 2: For each ordinal scale or nominal scale variable in the source data, generate a set of artificial values whose distribution is similar to the source data.
- Step 3: For all variables in the mimic data, repeat swapping of values so that the correlation coefficient matrix of the mimic data becomes similar to that of the source data.

In the next section, details of these steps are described.

B. Step 1. Generation of Ratio/Interval Scale Variables

This paper employs the Box–Muller method [5] to generate quantitative variables. The Box–Muller method, also called the Box-Muller transform, is an algorithm for generating a pairs of normally distributed random numbers $N(\mu, \sigma^2)$ from given uniformly distributed random numbers. Its mathematical expression is as follows:

$$N_1 = \sigma\sqrt{-2logR_1}cos2\pi R_2 + \mu$$

$$N_2 = \sigma\sqrt{-2logR_1}sin2\pi R_2 + \mu$$

where R_1 and R_2 are independent samples from the uniformly distributed random numbers on the interval $(0,1)$. These R_1 and R_2 are easily generated in many programming languages (e.g. by using rand() function in C language). N_1 and N_2 are independent random variables with a normal distribution. In this paper we use N_1 only.

As mentioned above, we assume quantitative variables follow log-normal distribution. To generate log-normally distributed random numbers, we apply exponential transformation, which is inverse transformation of the logarithmic transformation, to values obtained by the Box-Muller method.

As an example, Fig. 1 shows the value distribution of “effort” in Desharnais data set, which we consider as source data. Fig. 2 shows its log-transformed value distribution. We see in Fig. 2 that log-transformed effort values roughly follow the normal distribution. We can use the standard deviation σ and the mean value μ of Fig. 2 to generate the mimic data by the Box-Muller method. Fig. 3 shows the result of the Box-Muller method, which is the mimic data of Fig. 2. Finally, Fig. 4 shows the result of its exponential transformation, which is a mimic data of Fig. 1. Although values in Fig. 4 are all artificially generated ones, we see that Fig 4 well resembles Fig. 1.

In addition, by the following equation, we can directly obtain the standard deviation σ and the mean value μ of log-transformed source data from the standard deviation σ' and the mean value μ' of original source data.

$$\sigma^2 = \ln\{1 + (\sigma'/\mu')^2\}$$

$$\mu = \ln(\mu') - \sigma^2/2$$

This means that, a company who own a (secret) source data only needs to provide σ' and μ' directly computed from the source data.

C. Step 2. Generation of Ordinal/Nominal Scale Variables

For each ordinal scale or nominal scale variable in the source data, we generate a set of artificial values so that the percentage of cases in each bin is same as the source data. For example, assuming that we have an ordinal scale variable “requirement clarity,” which has four ranks or bins (“1. very clear”, “2. clear”, “3. unclear”, “4. very unclear”). As also assume that the percentage of values belonging to these bins are 20% for “1. very clear”, 25% for “2. clear”, 35% for “3. unclear” and 10% for “5. very unclear” respectively. Then, to generate a mimic data, we simply generate an artificial mimic sample whose percentage of cases in each bin is same as that of the source data.

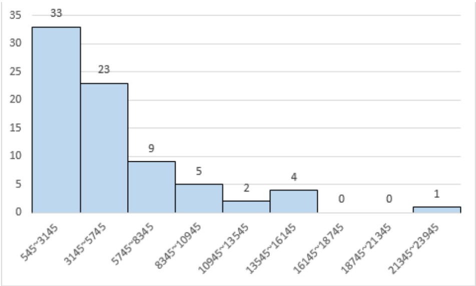

Fig. 1 Histogram of software development effort of Desharnais data set.

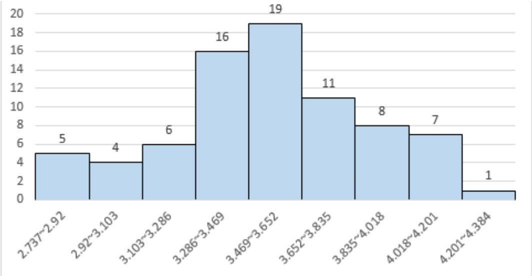

Fig. 2 Histogram of log-transformed software development effort of Desharnais data set.

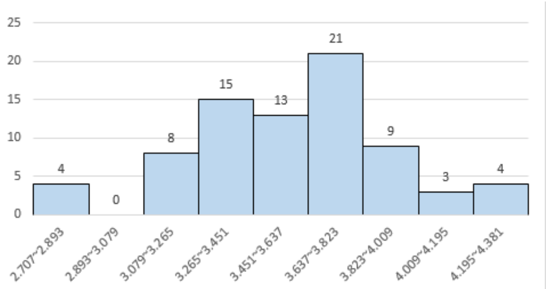

Fig. 3 Histogram of mimic data of log-transformed software development effort of Desharnais data set.

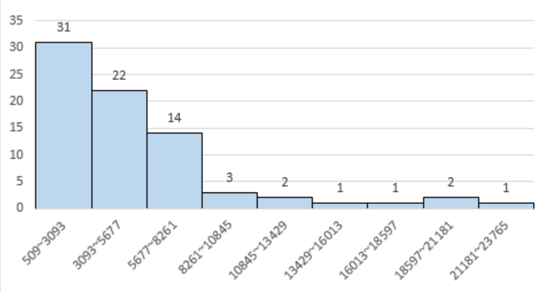

Fig. 4 Histogram of mimic data of software development effort of Desharnais data set.

D. Step 3. Mimicking the Relationship among Variables

For all pairs of variables in the source data, there may exists some sort of relationship. This paper captures such relationships via the correlation coefficient matrix of the source data; and, the proposed method tries to make the correlation coefficient matrix of the mimic data close to that of the source data. This can be done by swapping values within a variable, which does not break the value distribution of that variable. In this study, we assume there is some outliers in the source data; therefore, we decided to use Spearman’s rank correlation coefficient instead of the Pearson correlation coefficient to capture the relationships among variables.

We propose the following procedure to mimic the relationship among variables in source data.

1. Compute the correlation coefficient matrix of the source data.
2. Randomly select one variable in the mimic data. Then, randomly select two values from this variable, and swap them.
3. If the correlation coefficient matrix of the mimic data becomes more similar to that of the source data, we consider that the value swapping is successful, and go back to Step 2. Otherwise, we consider that the swapping is unsuccessful, cancel swapping and go back to Step 2. To evaluate the similarity of the correlation coefficient matrices, we use the sum of squared differences ($\sum(a_i - b_i)^2$) between the set of rank correlation coefficients of the mimic data and those of the source data. If the sum of squared differences becomes smaller after the swapping, then we consider that the correlation coefficient matrices get more similar.
4. When the sum of squared differences converges, swapping is completed (i.e. stop repeating Step 2.)

E. *Rounding Off Generated Values*

This is an additional step to make the mimic data visually more similar to the source data. Since the values of quantitative variables are generated from random numbers, their significant figures are different from that of source data. For this reason, each value should be rounded off to an appropriate precision according to the significant figure of source data. For example, Function Point is an integer in source data, so it should be rounded off to integer.

IV. CASE STUDY

To evaluate the effectiveness of the proposed method, this section presents a case study to generate a mimic data from Desharnais data set [7]. In the case study, we built effort estimation models from both the source data and the mimic data to investigate their similarity.

A. *Source data set*

The Desharnais data set is one of the most frequently used data sets in software effort estimation research [10]. It contains 77 projects without missing values. This case study generated mimic data of same sample size (n=77.) Quantitative variables used in this paper are Duration, Transactions, Entities, PointsAdjust, and Effort. And, qualitative variables used are TeamExp, ManagerExp, and Lang. TeamExp and ManagerExp are ordinal scale variables, the TeamExp ranges from 0 to 4, and the ManagerExp ranges from 0 to 7. The variable Lang is divided into two binary variables Lang2 and Lang3.

B. *Characteristics of Generated Ratio/Interval Scale Variables*

TABLE II. STATISTICS OF SOURCE DATA

	Mean value	Standard deviation	Maximum value	Minimum value
Duration	11.299	6.742	36	1
Transactions	177.468	145.129	886	9
Entities	120.545	85.547	387	7
PointsAdjust	298.013	181.076	1127	73
Effort	4833.909	4160.9	23940	546

TABLE III. STATISTICS OF MIMIC DATA

	Mean value	Standard deviation	Maximum value	Minimum value
Duration	11.571	7.172	42	3
Transactions	180.078	139.485	822	39
Entities	123.208	91.018	534	29
PointsAdjust	300.299	168.783	986	99
Effort	4913.26	4246.176	25365	893

TABLE IV. RELATIVE DIFFERENCE OF STATISTICS BETWEEN SOURCE DATA AND MIMIC DATA

	Mean value	Standard deviation	Maximum value	Minimum value
Duration	0.024	0.06	0.143	0.667
Transactions	0.014	0.04	0.078	0.769
Entities	0.022	0.06	0.275	0.759
PointsAdjust	0.008	0.073	0.143	0.263
Effort	0.016	0.02	0.056	0.389

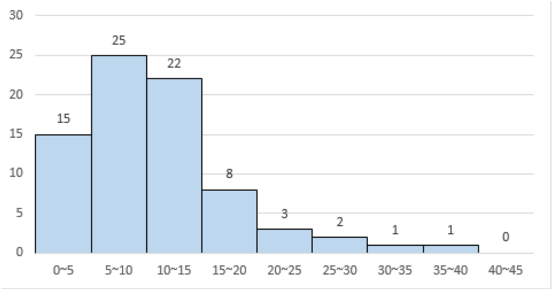

Fig. 5.1 Histogram of Duration of source data.

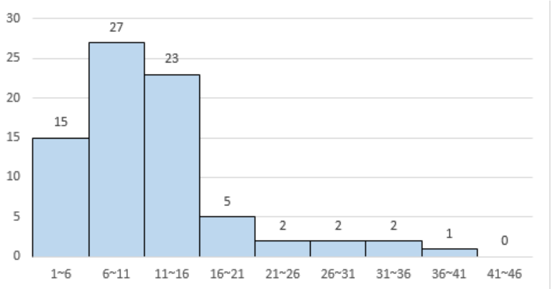

Fig. 5.2 Histogram of Duration of mimic data.

The mean value, standard deviation, maximum value and minimum value of quantitative variables of source data and mimic data are shown in Table II and Table III respectively. Their relative differences are shown in Table IV. From these results, we see that the difference of mean value, standard deviation and minimum value between two data sets are very small, which indicates effectiveness of the proposed method. On the other hand, the maximum values are turned out to be not very similar. This is because source data contain outliers. Mimicking the outliers are our important future work.

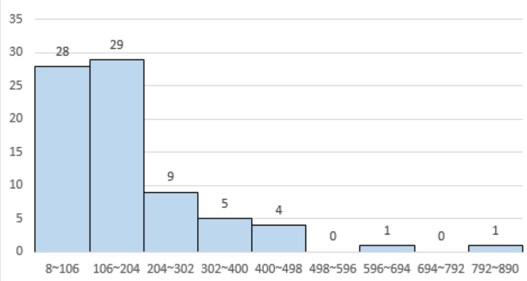

Fig. 5.3 Histogram of Transactions of source data.

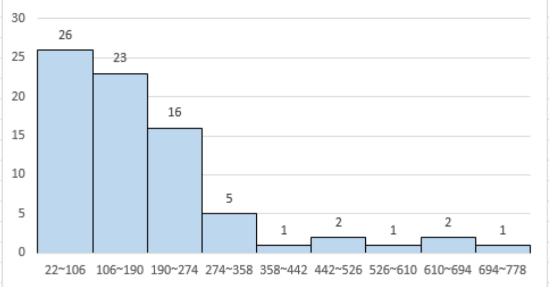

Fig. 5.4 Histogram of Transactions of mimic data.

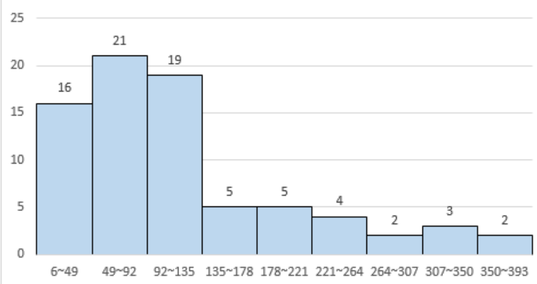

Fig. 5.5 Histogram of Entities of source data.

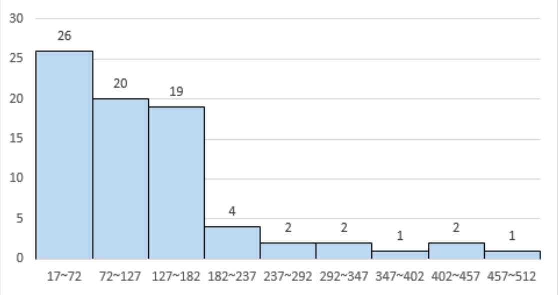

Fig. 5.6 Histogram of Entities of mimic data.

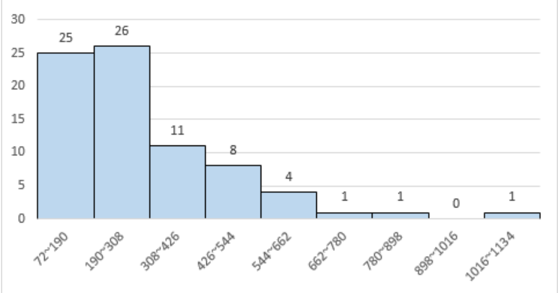

Fig. 5.7 Histogram of PointsAdjust of source data.

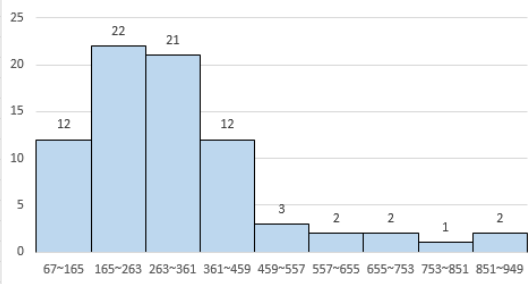

Fig. 5.8 Histogram of PointsAdjust of mimic data.

For more details of the generated variables, the distribution of source data and mimic data of the four quantitative variables are shown in Figure 5.1 to Figure 5.8. From these figures we can also visually see the similarity between two data sets. (For the variable "Effort", we have already shown the histograms in Fig. 1 and Fig. 4.)

C. Rank Correlation Coefficient Matrix

Fig. 6 shows the convergence of the sum of squared differences of rank correlation coefficients when increasing the number of updates (i.e. successful swapping) of variables. As shown in the figure, the sum squared differences becomes very close to zero (0.000069) as the number of updates increases.

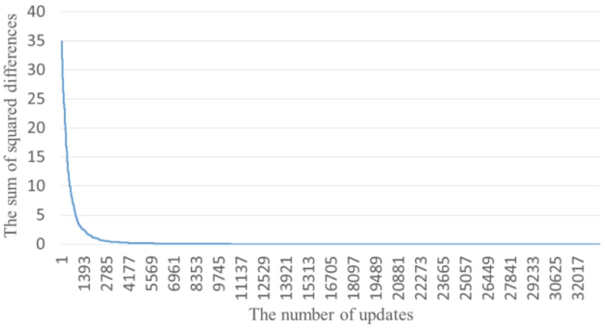

Fig. 6 Convergence of the sum of squared differences of rank correlation coefficients.

TABLE V. RANK CORRELATION COEFFICIENT MATRIX OF SOURCE DATA

	TeamExp	Manager Exp	Duration	Transactions	Entities	PointsAdjust
TeamExp	1.000					
ManagerExp	0.388	1.000				
Duration	0.365	0.233	1.000			
Transactions	0.087	0.109	0.382	1.000		
Entities	0.319	0.170	0.533	0.265	1.000	
PointsAdjust	0.266	0.190	0.592	0.744	0.778	1.000
Lang2	-0.073	0.157	0.147	-0.129	0.045	-0.039
Lang3	-0.075	0.180	-0.106	0.248	-0.120	0.077
Effort	0.252	0.086	0.572	0.467	0.647	0.688

TABLE VI. RANK CORRELATION COEFFICIENT MATRIX OF MIMIC DATA

	TeamExp	Manager Exp	Duration	Transactions	Entities	PointsAdjust
TeamExp	1.000					
ManagerExp	0.389	1.000				
Duration	0.365	0.235	1.000			
Transactions	0.088	0.109	0.381	1.000		
Entities	0.319	0.170	0.532	0.265	1.000	
PointsAdjust	0.266	0.190	0.591	0.742	0.776	1.000
Lang2	-0.071	0.165	0.146	-0.128	0.045	-0.039
Lang3	-0.067	0.187	-0.106	0.248	-0.120	0.077
Effort	0.252	0.086	0.572	0.466	0.647	0.690

TABLE VII. EFFORT ESTIMATION MODEL FOR SOURCE DATA

	Coefficient	p-value
Intercept	1.373	0.000
TeamExp	-0.006	0.727
ManagerExp	0.010	0.550
LOG(Length)	0.254	0.016
LOG(Transactions)	0.204	0.127
LOG(Entities)	0.184	0.173
LOG(PointsAdjust)	0.504	0.046
Lang2	-0.065	0.195
Lang3	-0.604	0.000

TABLE VIII. EFFORT ESTIMATION MODEL FOR MIMIC DATA

	Coefficient	p-value
Intercept	1.502	0.000
TeamExp	-0.020	0.329
ManagerExp	0.011	0.554
LOG(Length)	0.162	0.180
LOG(Transactions)	0.259	0.082
LOG(Entities)	0.217	0.159
LOG(PointsAdjust)	0.424	0.127
Lang2	-0.073	0.205
Lang3	-0.602	0.000

A part of rank correlation coefficient matrix for each data set is shown in Table 5 and Table 6. From Table 5 and Table 6, we can see that the maximum of the difference is 0.008, which is sufficiently small. So it is considered that the relationship between any two variables is sufficiently reproduced.

D. The Comparison of Predictive Model about Man-hour

Assuming the effort estimation research using mimic data. we conduct log-log regression modeling on both source data and mimic data respectively, and investigate their similarity. The objective variable is "Effort" and other variables are predictor variables. The log-log regression model is a linear regression model with logarithmic transformation applied to both predictor variables and the objective variable before model construction. Kitchenham and Mendes [10] pointed out the necessity of logarithmic transformation to improve the prediction performance of effort estimation models.

The result of log-log regression for source data and mimic data are shown in Table VII and Table VIII. From these tables, we see constant (intercept) and coefficients of predictor variables are similar. The R^2 values of these models are 0.882 for source data and 0.820 for mimic data, which are also similar. Looking at p-values, for some variable, p-value is not very similar. One of the possible reason is that outliers might affected the p-value. We need further investigation in our future study. Also, in future we will evaluate the prediction performance of the models.

V. SUMMARY

In this paper we proposed a method for artificially generating a mimic data set from a given (confidential) source data set. From a case study with a software project data set, our main findings are as follows.

- The standard deviation and the mean value of quantitative variables of mimic data are very similar to that of source data.
- The rank correlation coefficient matrix of mimic data is very similar to that of source data.
- Effort estimation models using log-log regression modeling built from source data and mimic data are similar in their coefficients.

In future, we will evaluate the prediction performance of the built models. Also, we will apply various data analysis techniques such as clustering and association rule mining for mimic data to evaluate the utility of the proposed method. In addition, we will try to improve our method by mimicking more aspects in source data, such as outliers, skewness and kurtosis of variables.

REFERENCES

[1] A. J. Albrecht, J. Gaffney, "Software function, source lines of code, and development effort prediction," IEEE Transactions on Software Engineering, vol. 9, pp.639-648, 1983.

[2] M. Azzeh, M., "A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation," Empirical Software Engineering, vol.17, no.1-2, pp.90-127, 2012.

[3] B. Baskeles, B. Turhan, and A. Bener, "Software effort estimation using machine learning methods," Proc. 22nd International Symposium on Computer and Information Sciences (ISCIS2007), pp.126-131, Dec. 2007.

[4] B. Boehm, "Software engineering economics," Prentice-Hall, NY, 1981.

[5] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," The Annals of Mathematical Statistics, vol. 29, no. 2 pp. 610–611, 1958.

[6] L. Briand, T. Langley, and I. Wieczorrek, "A replicated assessment and comparison of common software cost modeling techniques," Proc. 22nd International Conference on Software Engineering (ICSE2000), pp.377-386, 2000.

[7] J.-M. Desharnais, "Analyse statistique de la productivitie des projects informatique a partie de la technique des point des function," Master's Thesis, University of Montreal, 1989.

[8] M. C. Jones, and A. Pewsey, "Sinh-arcsinh distributions," Biometrika, vol.96, no.4, pp.761-780, Dec. 2009.

[9] C. F. Kemerer, "An empirical validation of software cost estimation models," Communications of the ACM, vol. 30, no. 5, pp. 416-429, 1987.

[10] B. Kitchenham, and E. Mendes, "Why comparative effort prediction studies may be invalid," Proc. 5th International Conference on Predictor Models in Software Engineering, Article no.4, May 2009.

[11] E. Kocaguneli, T. Menzies, J. Keung, "On the value of ensemble effort estimation", IEEE Transactions on Software Engineering, vol. 38, no. 6, pp. 1403-1416, 2012.

[12] K. Maxwell, "Applied statistics for software managers," Englewood Cliffs, NJ, Prentice-Hall, 2002.

[13] T. Menzies, R. Krishna, and D. Pryor, "The promise repository of empirical software engineering data," http://openscience.us/repo, North Carolina State University, Department of Computer Science, 2015.

[14] F. Peters and T. Menzies, "Privacy and utility for defect prediction: experiments with MORPH," Proc. International Conference on Software Engineering, pp.189-199, 2012.

[15] F. Peters, T. Menzies, L. Gong, and H. Zhang, "Balancing privacy and utility in cross-company defect prediction," IEEE Transactions on Software Engineering, vol. 39, no. 8, pp. 1054-1068, 2013.

[16] P. Phannachitta, J. Keung, A. Monden, and K. Matsumoto, "A stability assessment of solution adaptation techniques for analogy-based software effort estimation," Empirical Software Engineering, vol.22, no.1, pp.474-504, 2017.

[17] Software Reliability Enhancement Center, Information-technology Promotion Agency, "White paper on software development data in 2016-2017," SEC Books, 2016