PAPER

Empirical Evaluation of Mimic Software Project Data Sets for Software Effort Estimation

Maohua GAN[†], **Zeynep YÜCEL**[†], ***Nonmembers***, **Akito MONDEN**[†a)], ***Member***, ***and*** **Kentaro SASAKI**[††], ***Nonmember***

SUMMARY 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. To that end, this study proposes a method for artificially generating 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. 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. The proposed method uses the Box-Muller transform for generating normally distributed random numbers; and exponential transformation and number reordering for data mimicry. To evaluate the efficacy of the proposed method, effort estimation is considered as potential application domain for employing mimic data. Estimation models are built from 8 reference data sets and their concerning mimic data. Our experiments confirmed that models built from mimic data sets show similar effort estimation performance as the models built from original data sets, which indicate the capability of the proposed method in generating representative samples.

key words: *empirical software engineering, data confidentiality, data mining*

1. Introduction and Motivation

Empirical software engineering relies to a great extent on real software development data. Namely, it is highly desirable to use data collected from industry software development projects. However, there exist only very few industry data sets, which are publicly available [1]. In addition, these data sets are quite old and have a small sample size, which pose a great problem in ensuring the validity and reliability of the research [2]–[5].

As a matter of fact, most companies measure and accumulate data relating their own (recent) software development projects on an independent basis. However, companies cannot release any part of this (real) data due to two principal reasons. Namely, they need to comply to various data protection/privacy laws and standards. In addition, due to the large number of stakeholders involved, they are required to strictly preserve data confidentiality.

In this respect, this study proposes a method for artificially creating a data set with similar characteristics to a given industry data set. Namely, instead of releasing the original (confidential) data set, the companies may provide only several statistical values of their data, such that a completely anonymous data set is automatically generated with similar characteristics.

From a practical point of view, such a method is beneficial to several parties. First of all, academic researchers can work on recent and realistic information. For instance, for studies on software development effort (henceforth, referred simply as effort) estimation, the proposed method is expected to be very helpful in assessment of stability, which inherently requires numerous industry data sets [6]. Thereby, validity and reliability of new effort estimation methods can better be assured. The proposed method is potentially useful also for the practitioners. Namely, companies may want to compare their software development performance metrics (such as productivity and defect density) with other companies. The proposed method enables comparison of performance through artificially generated data sets replicating statistical features of authentic data. We expect the proposed approach to encourage the companies to share the statistics of their data, once the researchers release their findings, which potentially involve beneficial information ready to be transferred to industry applications.

The principals of the proposed method are as follows. Regarding a real industry software project data set, we consider certain (authentic) variables (e.g. software metrics) and measure their statistics as well as pairwise correlation relations. Next, to generate synthetic variables, we use the Box-Muller transform [7] and obtain a set of normally distributed random numbers. Subsequently, we apply exponential transformation on those and transfigure the resulting values such that their (value) distribution emulates that of the authentic variables. After obtaining all such synthetic variables, number reordering is applied to achieve similar pairwise correlation relation to that within the authentic variables. In this respect, we assume that certain statistical information regarding an industrial data set (i.e. mean, standard deviation and correlation coefficient matrix) are not confidential and we expect to receive them as inputs to generate a mimic data set. Nevertheless, this assumption does not jeopardize confidentiality of the -input- data set, since the proposed method prevents identification of *any specific project*

Manuscript received June 3, 2019.
Manuscript revised October 30, 2019.
Manuscript publicized July 3, 2020.
[†]The authors are with Okayama University, Okayama-shi, 700–8530 Japan.
[††]The author was with Okayama University, Okayama-shi, 700–8530 Japan.
a) E-mail: monden@okayama-u.ac.jp
DOI: 10.1587/transinf.2019EDP7150

in this set, as it is a common requirement in data anonymization studies.

This paper extends our previous work [8] with extensive empirical evaluation carried out on 8 industry data sets. In addition, we confirm the predictive ability of mimic data sets by illustrating the efficacy of synthetic variables for the particular purpose of effort estimation.

This paper is organized as follows. We elaborate on the background and relevant studies in Sect. 2. Section 3 first gives an outline of the proposed method and then details each stage, whereas Sect. 4 provides a demonstration of the procedure on a commonly used data set. Subsequently, Sect. 5 considers effort estimation as one of the potential many application domains of mimic data; and evaluates and compares estimation performance obtained from authentic data and mimic data. Section 6 provides a discussion on the experimental validation, whereas Sect. 7 concludes the paper summarizing our main results, contributions and future work.

2. Background and Related Work

Some of the most popular industry data sets employed in empirical software engineering studies such as Desharnais [2], Coc81-dem [5], Kemerer [3], and Albrecht [4], are available at [1]. These data sets are all recorded in the 1980's. In this respect, the development environments and processes greatly differ from modern software development. In addition, the sample size is often very small, e.g. Kemerer has only 15 projects, whereas Albrecht has 24 projects. Surprisingly, although these data sets are old and small, they are still actively used in various recent studies appearing in top journals (e.g. [6], [9], [10]) due to the lack of more recent industry data sets.

On the other hand, there exist also a few self-contained studies based on recent software development data. However, they only report the analysis results and do not disclose any of the data itself. For example, the white paper on software development data in 2018-2019 [11] provides various analysis results of 4564 software development projects carried out by 34 Japanese software development companies. But it does not release the data set.

To mitigate the problems due to use of outdated or small sets, it is proposed to apply *anonymization* on recent software data sets. Data anonymization aims removing any identifying information from the original data such that the source or its private characteristics cannot be determined. Conventional data anonymizing methods for software engineering data employ *data mutation* techniques to gain data privacy [12], [13]. 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 [13].

In [12], Peters and Menzies 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 [13].

Since their approach is specifically proposed for a binary classification problem (i.e. distinguishing defect-prone 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 as we target in this paper.

3. Proposed Method

Software project data sets typically involve various variables including software size metrics (e.g. function point, source lines of code), as well as project duration, and effort. As an example for a software project data set, Table 1 depicts an excerpt from the Desharnais data set [2], which is one of the commonly used software project data sets in effort estimation studies.

As it can clearly be seen in Table 1, the variables can be measured at varying scales. Namely, for the specific case of [1], *language* emerges as a nominal variable, whereas *team experience* and *project manager experience* are ordinal. On the other hand, quantitative variables involve such ratio scale variables as *function point*, *effort* and *duration*.

Many software companies record such data sets consisting of project features similar to those listed in Table 1[†]. Henceforth, we refer to such an authentic confidential data set as "source data set" or as simply "source data", whereas the synthetic data emulating the characteristics of the source data is referred as "mimic data set" or "mimic data".

Section 3.1 introduces the outline of the proposed method in a nutshell, whereas the details of the procedure are elaborated on in Sects. 3.2, 3.3 and 3.4.

3.1 Outline

For generating mimic variables, we exploit the fact that probability distributions of the quantitative variables in software project data sets roughly follow a log-normal distribution [14]. From this viewpoint, we approximate their (value) distribution with a log-normal distribution.

We initially generate a set of variables based on log-normal distribution assumption and obtain any number n of artificial values with a similar distribution to that of the corresponding values in source data. At this point, we are clearly required to follow a different approach for variables

[†]In this study, we assume that there is no missing value in a data set.

Table 1 An example of software project data set (excerpt from Desharnais data set [2].

TeamExp (years)	ManagerExp (years)	Duration (months)	Transactions	Entities	PointsAdjust	Lang2	Lang3	Effort (person-hours)
1	1	5	78	99	177	0	0	2520
4	7	13	69	74	143	0	0	1603
1	3	8	194	97	291	1	0	3626
1	3	10	42	31	73	1	0	1267
0	4	6	97	42	139	0	1	546
4	5	26	482	227	709	1	0	9100

of different scale (i.e. nominal, ordinal, ratio and interval scale), which will be distinguished in Sect. 3.2 and Sect. 3.3.

Subsequent to obtaining a set of randomly generated values, we compute the correlation between each pair of variables and build the correlation matrix. Comparing it to the correlation matrix relating the source data, we try to emulate similar pairwise relations. To that end, we keep swapping the positions of the variables such that the correlation matrix of the mimic data resembles enough to that of the source data.

In relation to the above procedure, we would like to point out to two particular advantages. First of all, the proposed method supports any number of data points to generate. For example, we can generate mimic data with a sample size of $n = 1000$ from a source data of much smaller sample size, e.g. $n = 30$. Relying on this property, our second advantage is recognized as the lack of one-to-one correspondence between the projects of the authentic data set and the values in the mimic data set. Thanks to this, data privacy and confidentiality are suggested to be effectively protected, even if the mimic data set is made open.

3.2 Generation of Ratio Scale Variables

Suppose that a certain quantitative variable in the source data set has a mean of m and a standard deviation of σ^2. Further, assume that after log-transforming it, the resulting distribution has a mean and standard deviation of $\widehat{m}$ and $\widehat{\sigma}^2$, respectively. Clearly, there exist the following relations between these pairs (see [15] for a detailed explanation),

$$\begin{aligned} m &= \ln\left(\frac{\widehat{m}}{\sqrt{\frac{\widehat{\sigma}^2}{\widehat{m}^2}+1}}\right), \\ \sigma^2 &= \ln\left(\frac{\widehat{\sigma}^2}{\widehat{m}^2}+1\right). \end{aligned} \tag{1}$$

For generating mimic values emulating the characteristics of the log-transformed quantitative (i.e. ratio scale and interval scale) variables, this paper employs the Box-Muller transform [7], which is a pseudo-random number sampling method.

Essentially, Box-Muller transform generates a pair of independent and normally distributed random numbers from a given set of uniformly distributed random numbers. Let R_1 and R_2 be two independent random variables drawn from a uniform distribution in the interval (0,1)[†]. Box-Muller method, transforms these values into independent random variables N_1 and N_2 as follows,

$$\begin{aligned} N_1 &= \sigma^* \sqrt{-2\log R_1}\cos 2\pi R_2 + m^*, \\ N_2 &= \sigma^* \sqrt{-2\log R_1}\sin 2\pi R_2 + m^*, \end{aligned} \tag{2}$$

where m^* and σ^* denote the desired mean and standard deviation of N_1 and N_2. Note that this procedure can be used also for interval scale variables.

In our specific application, since we target generating values emulating the log-transformed distribution, we use $m^* = \widehat{m}$ and $\sigma^* = \widehat{\sigma}$. Moreover, this study utilizes only N_1.

As mentioned in Sect. 3.1, we assume that quantitative variables follow a log-normal distribution. Therefore, we apply exponential transformation on N_1 and obtain the mimicking values of the variables.

Figure 1 gives a demonstrating example of this process depicted on the effort variable concerning Desharnais data set. The distribution is illustrated both in terms of a histogram and a kernel density estimate (KDE). Figure 1-(a) relates the source data, whereas Fig. 1-(b) illustrates its log-transform[††]. We use the mean value m and standard deviation σ of the authentic effort values to obtain the desired statistics of the log-transformed distribution using Eq. (1) and generate the mimic data using Eq. (2). Figure 1-(c) shows the outcome of this operation in terms of the histogram of the mimic variables and relating KDE. Finally, Fig. 1-(d) shows the result of exponential transformation applied on the mimic variables. Although values in Fig. 1-(d) are derived artificially, we see that the distributions are well in line with those of the source data presented in Fig. 1-(b).

For realizing the above procedure, a company that owns a (confidential) software development data set, needs to provide only m and σ, which are directly computed from the source data.

3.3 Generation of Ordinal and 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.

[†] R_1 and R_2 can easily be generated in many programming languages, e.g. by using `rand()` function in C programming language.

[††] Figure 1-(b) confirms that log-transformed values roughly follow the normal distribution.

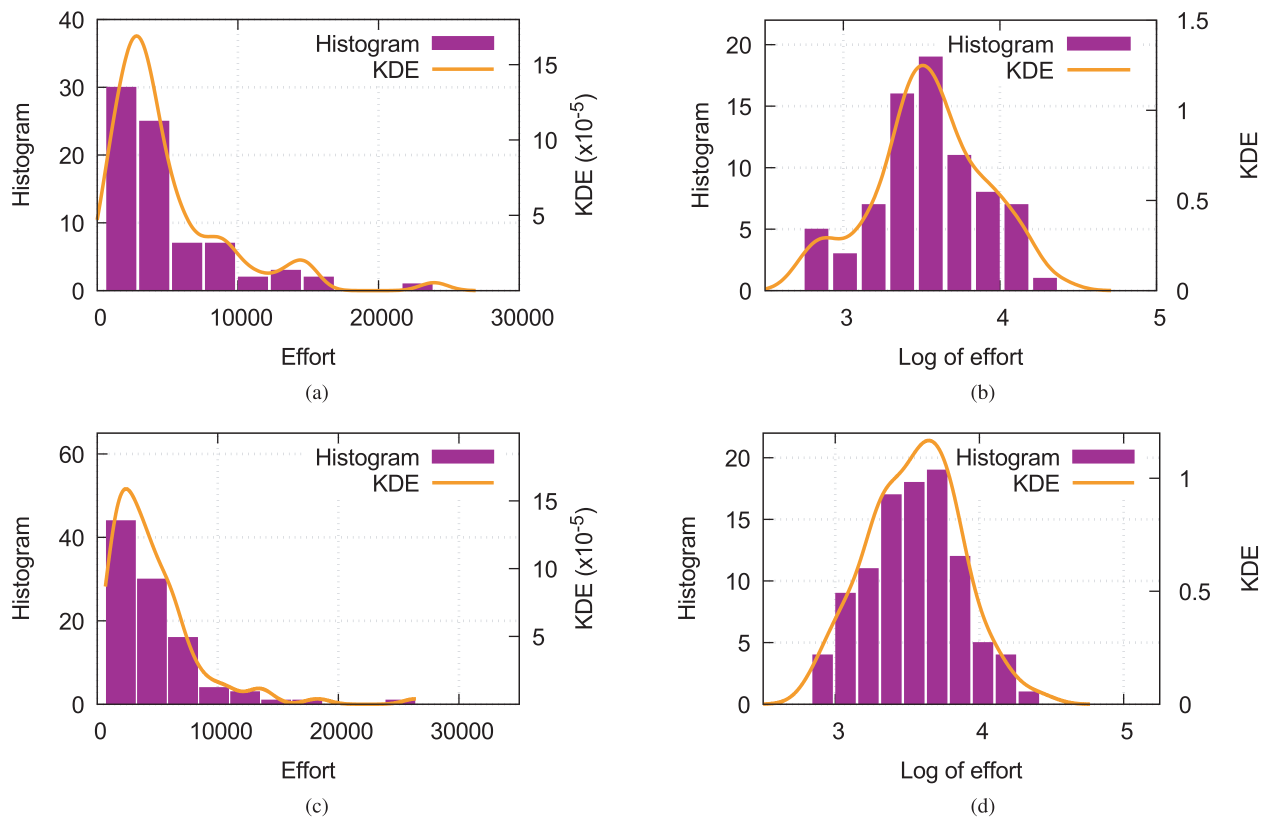

Fig. 1 Histogram and kernel density estimates regarding (a) raw and (b) log-transformed values of effort of Desharnais data set; and (c) mimic data after exponential transformation and (d) the mimic data.

For instance, consider that we have an ordinal scale variable as "requirement clarity", which has four ranks (or bins) as "1. very clear", "2. clear", "3. unclear", "4. very unclear".

Suppose that the percentage of values belonging to each bin are 20% for "1. very clear", 25% for "2. clear", 40% for "3. unclear" and 15% for "4. very unclear", respectively. In order to generate mimic data, which represents the characteristics of the authentic distribution, we simply generate an artificial mimic sample, whose percentage of cases corresponding to each bin is same as that of the source data.

3.4 Mimicking Pairwise Relationships of the Variables

Between every pair of variables in the source data, there exist a certain relationship, which we opt to capture via a correlation matrix χ. Specifically, we use Spearman's rank correlation coefficient instead of the common practice based on Pearson correlation coefficient (see Algorithm 1[†]). This choice is due to the existence of outliers in the source data.

Based on the correlation matrix, we emulate a similar pairwise relation between every possible mimic variable pair to that of the authentic variable pairs. To that end, we apply number reordering to the array of the mimic data.

Algorithm 1: Computation of correlation matrix based on Spearman's rank correlation coefficient.

```
Input: Values of project variables $\vec{V}_i$, $\forall i \in [1, T]$
Output: Correlation matrix $\chi$
1 for $i \leftarrow 1 : T$ do
2     Set $F$ to size of $\vec{V}_i$
3     Set $\vec{S}_i$ to ranked array of $\vec{V}_i$
4     for $i \leftarrow 1 : F$ do
5         for $j \leftarrow 1 : F$ do
6             $\chi(i, j) = \frac{\mathrm{cov}(S_i, S_j)}{\sigma_{S_i} \sigma_{S_j}}$
```

Namely, we swap random values, which obviously does not have any effect on the distribution of that variable. Specifically, we employ the procedure presented in Algorithm 2, which evaluates similarity of the correlation matrices χ and χ' concerning source and mimic data in terms of sum of squared differences ε[††].

Subsequently, we make mimic data visually more similar to source data by rounding-off to a suitable precision. Namely, mimicking values of the quantitative variables are generated from random numbers, and thus their significant

[†]In Algorithm 1, cov stands for covariance, σ stands for standard deviation, and T stands for number of project variables.

[††]Here, convergence is judged in terms of the number of successive iterations which do not lead to an improvement.

Algorithm 2: Mimicking pairwise relations.

```
   Input: Values of mimic variables $\overrightarrow{V}'_i$ $\forall i \in [1,T]$
   Correlation matrix of source data $\chi$              // See Alg. 1
   Output: Reordered values of mimic variables $\overrightarrow{V}'_i$, $\forall i \in [1,T]$
1  Set $S'$ to size of any array $\overrightarrow{V}'_i$
2  Set $\varepsilon_0 = \infty$                     // Previous value of $\varepsilon$
3  do
       /* Reorder by swapping arbitrary values */
4      Get a pair of arbitrary indices $1 < p,q < S'$ , $i \neq j$
5      $\overrightarrow{V}'_i(p) \leftrightarrow \overrightarrow{V}'_i(q)$                            // Swap
6      Get $\chi'$ relating reordered mimic data       // See Alg. 1
       /* Similarity of correlation matrices $\varepsilon$ is
          expressed in terms of sum of squared
          differences.  */
7      Set $\varepsilon = \sum_{i,j=1}^{T} (\chi(i,j) - \chi'(i,j))^2$
8      if $\varepsilon < \varepsilon_0$ then              // There is improvement
9
10         $\varepsilon_0 = \varepsilon$
11     else                        // There is no improvement
12
13         $\overrightarrow{V}'_i(p) \leftrightarrow \overrightarrow{V}'_i(q)$                   // Swap back
14 while $\varepsilon$ converging
```

figures are different from those in the source data. Therefore, each mimicking value should be rounded off to an appropriate precision according to the significant figure in the source data. For instance, Function Point variable is an integer in the source data, and as such, it is rounded off to integer.

4. Case Study on the Desharnais Data Set

In order to demonstrate the operation of the procedure introduced in Sect. 3, we present a case study carried out on the Desharnais data set [2], which is one of the most frequently used data sets in effort estimation research [14].

Desharnais data set contains 77 projects without missing values. Here, we generate mimic data with a sample size of $n = 100$. Besides, mimic data composes of 5 quantitative variables as Duration, Transactions, Entities, PointsAdjust, and Effort; and 3 qualitative variables as TeamExp, ManagerExp, and Lang. Here, TeamExp and ManagerExp are ordinal scale variables, where TeamExp ranges from 0 to 4, and the ManagerExp ranges from 0 to 7. In addition, the variable Lang is divided into two binary variables as Lang2 and Lang3.

4.1 Characteristics of Quantitative Variables

For an arbitrary quantitative (i.e. ratio or interval scale) variable r, we denote the mean value, standard deviation, minimum and maximum with m, σ, $r_{\min}$ and $r_{\max}$, respectively. Table 2 illustrates these values for the source data set, while Table 3 presents similar values relating mimic data, which are represented with $\widetilde{m}$, $\widetilde{\sigma}$, $\widetilde{r}_{\min}$ and $\widetilde{r}_{\max}$.

In addition, in Table 4 we present the relative differences of the statistics given in Tables 2 and 3. For instance for mean value, relative difference is defined as,

$$\Delta m = \frac{|m - \widetilde{m}|}{\max(m,\ \widetilde{m})}. \tag{3}$$

Table 2 Statistics of source data.

	m	σ	$r_{\min}$	$r_{\max}$
Duration	11.30	6.74	1	36
Transactions	177.47	145.13	9	886
Entities	120.55	85.55	7	387
PointsAdjust	298.01	181.08	73	1127
Effort	4833.91	4160.90	546	23940

Table 3 Statistics of mimic data.

	$\widetilde{m}$	$\widetilde{v}$	$\widetilde{r}_{\min}$	$\widetilde{r}_{\max}$
Duration	11.571	7.172	3	42
Transactions	180.078	139.485	39	822
Entities	123.208	91.018	29	534
PointsAdjust	300.299	168.783	99	986
Effort	4913.26	4246.176	893	25365

Table 4 Relative difference of statistics.

	Δm	$\Delta\sigma$	$\Delta r_{\min}$	$\Delta r_{\max}$
Duration	0.02	0.03	0.5	0.14
Transactions	0.03	0.16	0.44	0.37
Entities	0.02	0.07	0.41	0.04
PointsAdjust	0.03	0.14	0.32	0.34
Effort	0.04	0.13	0.39	0.30

From these results, we see that the absolute difference of mean value, standard deviation and minimum value between two data sets are very small, which indicates effectiveness of the proposed method. Note that, the relative difference values relating the minimum are higher than those relating the maximum. However, it is sufficient to check the values in Tables 2 and 3 to realize that they are inflated due to the inherently small values of $r_{\min}$ and the distributions still attain very similar minimums. On the other hand, the maximum values turn out to be not very similar, principally because source data set contains outliers.

4.2 Characteristics of the Correlation Matrix

Parts of the correlation matrices regarding source data and mimic data are shown in Table 5 and Table 6. In addition, Table 7 presents the absolute value of element-wise differences of Table 5 and Table 6. From Table 7, it can be observed that the maximum difference is attained for a particular pair of variables, namely of Lang2 and Lang3. This maximum difference of 0.023 is considered to be sufficiently small. Therefore, the relationship between any two variables is regarded to be effectively reproduced.

In addition, Fig. 2 shows the convergence of the sum of squared differences ε of rank correlation coefficients for increasing number of updates (i.e. successful swapping of variables). As shown in the figure, ε becomes very close to zero for growing number of iterations (e.g. 0.20495 at 1000 iterations and 0.000216 at 10000 iterations).

Table 5 Correlation matrix for the source data.

	TeamExp	ManagerExp	Duration	Transactions	Entities	PointsAdjust	Lang2	Lang3	Effort
TeamExp	1.000								
ManagerExp	0.388	1.000							
Duration	0.365	0.233	1.000						
Transactions	0.088	0.109	0.382	1.000					
Entities	0.319	0.170	0.533	0.265	1.000				
PointsAdjust	0.266	0.189	0.592	0.744	0.778	1.000			
Lang2	−0.072	0.157	0.147	−0.129	0.045	−0.039	1.000		
Lang3	−0.078	0.180	−0.106	0.248	−0.120	0.077	−0.247	1.000	
Effort	0.252	0.086	0.572	0.467	0.647	0.688	0.022	−0.428	1.000

Table 6 Correlation matrix for the mimic data.

	TeamExp	ManagerExp	Duration	Transactions	Entities	PointsAdjust	Lang2	Lang3	Effort
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	0.189	0.591	0.742	0.776	1.000			
Lang2	−0.071	0.165	0.145	−0.128	0.045	−0.039	1.000		
Lang3	−0.067	0.187	−0.106	0.248	−0.120	0.077	−0.224	1.000	
Effort	0.252	0.086	0.572	0.466	0.647	0.690	0.022	−0.427	1.000

Table 7 Absolute value of difference of correlation matrices.

	TeamExp	ManagerExp	Duration	Transactions	Entities	PointsAdjust	Lang2	Lang3	Effort
TeamExp	0								
ManagerExp	0.001	0							
Duration	0.002	0	0						
Transactions	0	0	0.001	0					
Entities	0	0	0.001	0	0				
PointsAdjust	0	0	0.001	0.002	0.002	0			
Lang2	0.001	0.008	0.002	0.001	0	0	0		
Lang3	0.011	0.007	0	0	0	0	**0.023**	0	
Effort	0	0	0	0.001	0	0.002	0	0.001	0

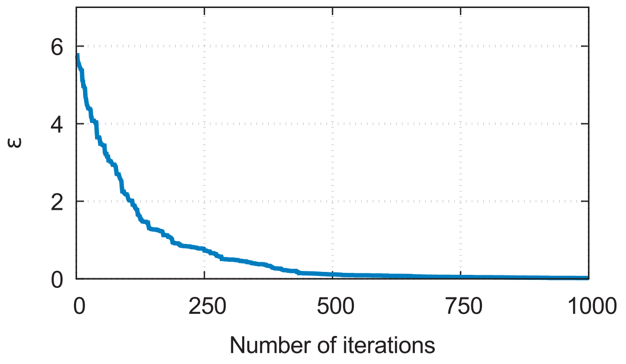

Fig. 2 Convergence of sum of squared differences ε of rank correlation coefficients.

5. Evaluation Based on Effort Estimation Performance

To evaluate utility of the mimic data set, we consider effort estimation as a representative application domain for employing mimic data. Namely, we consider a source data set with a set of variables composed of effort and several others. From this source data set, we generate mimic data regarding the variables other than effort. Based on this artificially generated set, we carry out effort estimation. On the other hand, we estimate effort based on the authentic variables in the source data set. We compare both estimations to the true values of effort. Effort estimation performance obtained using the authentic variables (i.e. source data set) is considered as benchmark performance. We compare this to the performance rates obtained using mimic data to investigate whether estimation performance based on mimic data is similar to the performance of the benchmark estimation obtained using authentic data.

To that end, in our experiments we employ as source data 8 data sets (henceforth, noted as reference data sets) introduced in Table 8. All data sets used in this study are publicly available [1].

5.1 Effort Estimation Model

As mentioned above, we consider effort estimation to be one of the many possible areas of deployment of mimic data. On this basis, we carry out the effort estimation method de-

Table 8 Reference data sets employed in the experiments.

Data set	Number of categorical variables	Number of continuous variables	Number of projects
Albrecht [4]	0	7	24
China [1]	1	11	499
Coc81-dem [5]	14	4	63
Desharnais [2]	4	5	77
Kemerer [3]	2	5	15
Maxwell [16]	22	4	62
Miyazaki94 [17]	0	8	48
Nasa93 [1]	24	2	93

scribed below.

Specifically, this study performs effort estimation (in person-months or person-hours) based on linear regression modeling. Generally speaking, a linear regression model is described as follows:

$$\widehat{Y} = \sum_{j=1}^{n} k_j N_j + C \tag{4}$$

$\widehat{Y}$	: Estimated value of the objective variable
N_j	: Predictor variables
k_j	: Partial regression coefficients
C	: A constant

For our specific case, the linear regression model employs effort as the objective variable and remaining variables as (potential) predictor variables.

As mentioned in Sect. 3.1 and illustrated in Fig. 1, logarithmic transformation of project variables yields a more similar distribution to normal distribution. Although linear regression does not explicitly require neither the objective variable nor the predictor variables to come from a normal distribution[†], as pointed out by Kitchenham and Mendes [14], their logarithmic transformation is empirically shown to help improving estimation results obtained by linear regression. Thus, it is a commonly used preprocessing operation in linear regression model construction, which also we opt to follow in this study.

In this respect, as a preprocessing operation, logarithmic transformation is applied on both the objective variable (i.e. effort) and the predictor variables (e.g. function point, duration etc) prior to model construction. In addition, for variables containing 0, an offset value (such as 0.5 or 1) is added before the transformation.

Thereby, the estimation model boils down to a log-log regression, which is expressed simply as follows:

$$\log \widehat{Y} = \sum_{j=1}^{n} k_j \log N_j + C. \tag{5}$$

It follows that, by applying exponential transformation on Eq. (5), the estimated value $\widehat{Y}$ can be obtained as:

[†]Linear regression assumes normality of residual errors.

Algorithm 3: Selection of predictor variables.

```
Input: Set of all project variables $V = \{V_1, \dots, V_T\}$
Output: Set of predictor variables $N, k, C$
Choose an arbitrary $i \in [1, T]$
Set $N = V_i$
while True do
    /* Insertion of a variable */
    Choose an arbitrary $j \in [1, T]$, $V_j \notin N$
    $N' = N \cup V_j$
    /* Removal of a variable */
    Choose an arbitrary $k \in [1, T]$, $V_k \in N$
    $N'' = N \setminus V_k$
    /* Difference in AIC */
    $\delta' = AIC(N') - AIC(N)$
    $\delta'' = AIC(N'') - AIC(N)$
    if $\delta' < 0 \,||\, \delta'' < 0$ then      // There is improvement
        if $\delta' < \delta''$ then
            $N = N'$
        else
            $N = N''$
    else                      // There is no improvement
        break
```

$$\widehat{Y} = \exp(C) \prod_{j=1}^{n} N_j^{k_j}. \tag{6}$$

5.2 Selection of Predictor Variables

As explained in Sect. 5.1, effort is the objective variable and remaining variables are potential predictor variables. In other words, we do not employ all available variables of a data set in effort estimation and instead eliminate any irrelevant or useless variables, which is one of the most crucial factors acting on estimation performance.

To that end, this study uses an iterative variable selection procedure based on Akaike's Information Criterion (AIC) [18] as depicted in Algorithm 3. Namely, we first build a simple model with a single predictor variable. At each iteration, we modify the model (i) by inserting an additional predictor variable and (ii) by removing a single predictor variable. This modification yields one extended model and one simplified model as compared to original one. Comparing the three AIC values concerning the original, extended and simplified models, we choose the best performing set of variables and update the model. We pursue this procedure until no extension or simplification brings any performance enhancement (in terms of AIC).

5.3 Evaluation Procedure

For each data set presented in Table 8, we conducted 10 repetitions of the 3-fold cross validation procedure illustrated in Fig. 3. We expect this series of experiments to mitigate the effect introduced by the random splitting (of the source data into test and fit data) on estimation results and provide an insight to the stability of performance.

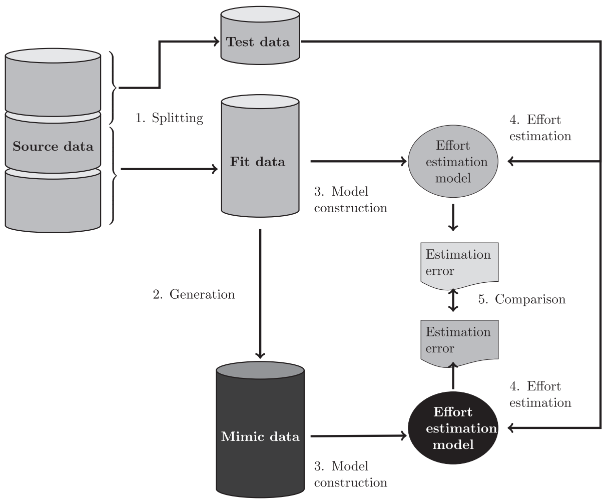

Fig. 3 The 3-fold cross validation scheme.

Specifically, we randomly split a source data set into 3 subsets. Then, we conduct three sets of model construction and model evaluation, each of which employs two subsets in model construction. In each case, the remaining subset is used in evaluation. We then construct an effort model for each original (source) fit subset and mimic data derived from fit subset.

Evaluation of both models are done using only the test subset. In other words, we do not produce mimic data for the test subset, since this experiment aims to evaluate how mimic data performs in effort estimation of real source data (i.e. not the mimic data).

To evaluate the effectiveness of our method, we apply this procedure on each reference data set and evaluate the performance using the metrics defined in Sect. 5.4.

5.4 Evaluation Criteria

As $\widehat{Y}$ denotes estimated value (see Eq. (6)) and Y denotes the true value of effort, let absolute error (AE), magnitude of relative error (MRE), magnitude of error relative to the estimate (MER), and balanced relative error (BRE) be defined as,

$$\begin{aligned} \mathbf{AE} &= |Y - \widehat{Y}|, \\ \mathbf{MRE} &= \frac{|Y - \widehat{Y}|}{Y}, \\ \mathbf{MER} &= \frac{|Y - \widehat{Y}|}{\widehat{Y}}, \\ \mathbf{BRE} &= \frac{|Y - \widehat{Y}|}{\max(Y, \widehat{Y})}. \end{aligned} \tag{7}$$

As for evaluation criteria, we use the above values, namely the relative difference of means between the source-based and mimic-based effort estimation; the difference of means between the source-based and mimic-based effort estimation relative to the mean of source data set, relative to the mean of mimic data set and relative the larger one of mean of source and mimic data sets. Let $\boldsymbol{m}(\cdot)$ and $\boldsymbol{\sigma}(\cdot)$ stand for functions returning mean and standard deviation, respectively. Consider relative difference is defined as in Eq. (3). Then, our evaluation criteria are expressed simply as $\Delta\boldsymbol{m}(\mathbf{AE})$, $\Delta\boldsymbol{m}(\mathbf{MRE})$, $\Delta\boldsymbol{m}(\mathbf{MER})$, and $\Delta\boldsymbol{m}(\mathbf{BRE})$.

For the two effort estimation procedures (one being based on source data and the other being based on mimic data), if the above-mentioned mean values are similar, the proposed method is concluded to be effective in effort estimation.

Table 9 Evaluation results.

Data set	$\Delta\boldsymbol{m}(\mathbf{AE})$	$\Delta\boldsymbol{m}(\mathbf{MRE})$	$\Delta\boldsymbol{m}(\mathbf{MER})$	$\Delta\boldsymbol{m}(\mathbf{BRE})$
Albrecht	0.63	0.02	0.30	0.01
China	0.02	0.01	0.03	0.01
Coc81-dem	0.10	0.05	0.17	0.08
Desharnais	0.02	0.05	0.02	0.02
Kemerer	0.33	0.53	0.44	0.18
Maxwell	0.04	0.03	0.08	0.02
miyazaki94	0.04	0.11	0.09	0.07
nasa93	0.04	0.02	0.30	0.08

5.5 Evaluation Results

Table 9 presents estimation performance based on the criteria presented in Sect. 5.4. Obviously, most of the values are quite low, indicating that similar **AE**, **MRE**, **MER** and **BRE** values are achieved by effort estimation using source data and mimic data. However, there are a few individual cases, which stand out. For instance, the values of $\Delta\boldsymbol{m}(\mathbf{AE})$ relating Kemerer and Albrecht data sets are quite high ($0.33 \sim 0.63$). This is primarily due to limited number of samples in these data sets (see Table 8). This disadvantage of Kemerer and Albrecht data sets can be observed also in terms of the other performance metrics. Namely, for Kemerer, **MRE**, **MER** and **BRE** are all significantly higher compared to other data sets, whereas for Albrecht **MER** sticks out as usually large.

In addition, we notice that in general **MER** is larger than **MRE**. This is possibly due to the fact that it is hard to mimic the projects, whose variables belong to the tails of the distributions. Therefore, since the proposed method has a tendency to produce mimic data resembling the bulk of the distribution, the estimated value $\widehat{Y}$ is likely to be lower, which in turn increases **MER** (see Eq. (7)). On the other hand, **BRE** introduces better (i.e. lower) rates, since it considers a larger value in the denominator (see Eq. (7)).

6. Threats to Validity

We provide a discussion on the validity of the proposed method in terms of three commonly adopted experimental validation approaches, i.e. internal validity, external validity and construct validity.

Internal validity refers to the extent by which the observed effect is a consequence of the presumed cause. In our case, internal validity questions whether or not different conclusions can be drawn with regard to the different settings in the experiment. To ensure internal validity, we conducted 10 repetitions in the validation process to produce stable results. However, there are several possible issues of internal validity in this study. One issue is the single sampling method (3-fold cross validation) we used. Our important future work is to employ other method such as leave-one-out cross validation to increase the validity of the result. Second issue is the single modeling method we used. We chose log-log regression model with stepwise variable selection, which is one of the most commonly used modeling techniques in software effort estimation. It is our future work to employ other modeling methods.

External validity refers to the generalization of the results. In this study, we address external validity by using 8 reference data sets with diverse characteristics. Namely, they vary in size (i.e. number of projects), and project variables, as well as origin (i.e. recording organization) and recording period. We believe that a diversity of data sets can produce generalized findings.

Construct validity refers to the relevance and capability of the observations and measurements in evaluating the posed hypothesis. In this study, we address construct validity by using both absolute error and relative error in the evaluating the effort estimation performance. However, since there are other measures of estimation error such as balanced relative error, it is our future work to employ such measures to increase the validity of our work.

7. Conclusions

This study proposes a method for building a mimic data set replicating the statistical properties of a (confidential) source data set. Software development companies, which cannot release data sets, can provide only a couple of data statistics (i.e. mean and standard deviation), which enable generating a mimic data set of any desired size with similar characteristics to the authentic data, by complying to legal requirements on privacy as well as meeting the demands of the stakeholders.

Based on our case study, we demonstrated that the mimic data set follows the characteristics of the source data set with considerable similarity. As for a potential application domain for utilizing mimic data, we consider effort estimation. By experimentally evaluating the effort estimation performance of mimic data sets derived from 8 different reference data sets, which are commonly used in effort estimation studies, we proved that performance rates achieved by mimic data are quite similar to those achieved by source data, provided that the data set has a sufficient number of samples (i.e. projects).

Based on these results, we claim that data anonymization is achieved effectively and the proposed method is a superior alternative to data mutation. Specifically, since all data points are artificially produced from randomly produced normal distribution values without referring to data points in the source data set, and the number of data points in the mimic data set can be set to any desired value, one cannot find one-to-one data mapping between source data and mimic data. We expect these results to be encouraging reasons for the companies to release statistics of their data for generation of mimic data sets; and both the research community and the industrial partners to profit from the outcomes of this study.

As future work, we consider applying various other 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 other characteristics of source data (in addition to mean and standard deviation), such as outliers, skewness and kurtosis.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant number 17K00102.

References

[1] T. Menzies, R. Krishna, and D. Pryor, "The seacraft repository of empirical software engineering data," https://zenodo.org/communities/seacraft, 2017.

[2] J. Desharnais, "Analyse statistique de la productivitie des projects informatique a partie de la technique des point des function," Masters Thesis University of Montreal, 1989.

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

[4] A.J. Albrecht and J.E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation," IEEE Trans. Softw. Eng., vol.SE-9, no.6, pp.639–648, 1983.

[5] B.W. Boehm, Software Engineering Economics, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1981.

[6] 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, Feb. 2017.

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

[8] M. Gan, K. Sasaki, A. Monden, and Z. Yucel, "Generation of mimic software project data sets for software engineering research," QuASoQ 2018, p.34, 2018.

[9] M. Azzeh, "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.

[10] E. Kocaguneli, T. Menzies, and J. Keung, "On the value of ensemble effort estimation," IEEE Trans. Softw. Eng., vol.38, no.6, pp.1403–1416, Nov. 2012.

[11] Software Reliability Enhancement Center, White paper on software development data in 2018-2019, SEC Books, 2018.

[12] F. Peters and T. Menzies, "Privacy and utility for defect prediction: Experiments with morph," Proceedings of International Conference on Software Engineering, ICSE, Piscataway, NJ, USA, pp.189–199, IEEE Press, 2012.

[13] F. Peters, T. Menzies, L. Gong, and H. Zhang, "Balancing privacy and utility in cross-company defect prediction," IEEE Trans. Softw. Eng., vol.39, no.8, pp.1054–1068, Aug. 2013.

[14] B. Kitchenham and E. Mendes, "Why comparative effort prediction

studies may be invalid," Proceedings of International Conference on Predictor Models in Software Engineering, PROMISE, New York, NY, USA, pp.4:1–4:5, ACM, 2009.

[15] Z. Yücel, "Implications of log-transformation on data statistics," https://yucelzeynep.github.io/pub/2019_10_appdx_ieice_gan.pdf, 2019.

[16] K. Maxwell and K. Maxwell, Applied statistics for software managers, Prentice Hall PTR Englewood Cliffs, 2002.

[17] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," Journal of Systems and Software, vol.27, no.1, pp.3–16, 1994.

[18] H. Akaike, "A new look at the statistical model identification," in Selected Papers of Hirotugu Akaike, pp.215–222, Springer, 1974.

Maohua Gan is a master course student in the Division of Electronic and Information Systems Engineering, Graduate School of Natural Science and Technology, Okayama University. He received the BE degree in software engineering from Northwestern Polytechnical University in 2015. His research interests include software measurement and analytics.

Zeynep Yücel is an assistant professor at Okayama University, Japan. She obtained her B.S. degree from Bogazici University, Istanbul, Turkey, and her M.S. and Ph.D. degrees from Bilkent University, Ankara, Turkey in 2005 and 2010, all in electrical engineering. She was a postdoctoral researcher at ATR labs in Kyoto, Japan for 5 years, before being awarded a JSPS fellowship in 2016. Her research interests include robotics, signal processing, computer vision, and pattern recognition.

Akito Monden is a professor in the Graduate School of Natural Science and Technology at Okayama University, Japan. He received the BE degree (1994) in electrical engineering from Nagoya University, and the M.E. and D.E. degrees in information science from Nara Institute of Science and Technology (NAIST) in 1996 and 1998, respectively. His research interests include software measurement and analytics, and software security and protection. He is a member of the IEEE, ACM, IEICE, IPSJ and JSSST.

Kentaro Sasaki is an engineer at a software company in Japan. He received the BE degree in Information Technology from Okayama University in 2016. His research interests include empirical software engineering.