IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 June 2022, accepted 28 June 2022, date of publication 4 July 2022, date of current version 8 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188246

RESEARCH ARTICLE

Improvement and Evaluation of Data Consistency Metric CIL for Software Engineering Data Sets

MAOHUA GAN, ZEYNEP YÜCEL, (Member, IEEE), AND AKITO MONDEN, (Member, IEEE)
Graduate School of Natural Science and Technology, Okayama University, Kita-ku, Okayama-shi 700-8530, Japan

Corresponding author: Maohua Gan (pa2i5772@s.okayama-u.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant JP20K11749 and Grant JP20H05706.

ABSTRACT Software data sets derived from actual software products and their development processes are widely used for project planning, management, quality assurance and process improvement, etc. Although it is demonstrated that certain data sets are not fit for these purposes, the data quality of data sets is often not assessed before using them. The principal reason for this is that there are not many metrics quantifying fitness of software development data. In that respect, this study makes an effort to fill in the void in literature by devising a new and efficient assessment method of data quality. To that end, we start as a reference from Case Inconsistency Level (CIL), which counts the number of inconsistent project pairs in a data set to evaluate its consistency. Based on a follow-up evaluation with a large sample set, we depict that CIL is not effective in evaluating the quality of certain data sets. By studying the problems associated with CIL and eliminating them, we propose an improved metric called Similar Case Inconsistency Level (SCIL). Our empirical evaluation with 54 data samples derived from six large project data sets shows that SCIL can distinguish between consistent and inconsistent data sets, and that prediction models for software development effort and productivity built from consistent data sets achieve indeed a relatively higher accuracy.

INDEX TERMS Data quality metric, data inconsistency, software project data analysis, software effort estimation, software productivity estimation.

I. INTRODUCTION AND MOTIVATION

In early stages of a software development project, various target values, such as development effort, software productivity, defect density, etc. need to be estimated, typically by referring to the data from other past projects using machine learning techniques [1], [7], [34]. However, if the quality of such data is low, these estimation techniques may not work efficiently or may behave in unexpected ways [18].

Although the benefits of assessment of data quality are self-evident, according to a systematic review by Liebchen and Shepperd, only 23 out of hundreds of articles explicitly addressed this issue [22]. In that respect, Liebchen and Shepperd emphasize that researchers should pay more attention to the quality of data, before deploying it.

According to the taxonomy proposed by Bosu *et al.* [6], [7], data quality challenges in empirical software engineering (ESE) include specifically outliers [21], [27], noise [11], [15], [19], data incompleteness [8], [19], [31], data inconsistency [9], [18], [32], and redundancy [13].

To the best of our knowledge, Case Inconsistency Level (CIL) is the only data inconsistency metric proposed to date [28]. According to [28], inconsistency is characterized by project cases with *conflicting feature values*. Table 1 illustrates such a conflict on an excerpt from the China data set [24]. Specifically, each row of Table 1 represents a software project and each column represents a project feature [14], [24]. When we look closely at projects ID 2 and ID 5, we see that they appear to have very similar characteristics, but the development effort of project ID 5 is more than five times greater than that of project ID 2. CIL evaluates the quality of a data set based on the number of such inconsistent pairs.

In this study, we first conduct a follow-up evaluation of CIL using a large sample set, and show that CIL is not effective

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

TABLE 1. Example of software project data set (excerpt from China data set [24].

ID	AFP (FP)	Input	Output	Enquiry	File	Interface	Duration	Effort (person-hours)
1	919	458	236	56	58	76	6	4815
2	66	27	12	6	14	0	2	246
3	54	36	0	16	0	0	1	686
4	151	41	21	29	41	19	3	391
5	66	27	14	6	17	0	2	1817
6	70	24	0	3	42	5	3	540
7	754	301	130	204	90	0	8	7443

in assessing the quality of certain data sets. Based on our analysis of ineffective cases, we propose an improved metric called Similar Case Inconsistency Level (SCIL).

Incorporating SCIL with various distance metrics and pre-processing methods, we empirically determine an efficient execution mode.[1] Applying it on 18 sets derived from six large project data sets from the SeaCraft repository of ESE data [24],[2] we distinguish between consistent and inconsistent data sets. Finally, we show that effort/productivity estimation models built from data sets identified as consistent by SCIL achieve indeed a relatively high accuracy.

The rest of the article is organized as follows: We elaborate on the definition of CIL in Section III and discuss its issues in Section IV through a follow-up evaluation. In Section V, we propose a new improved metric SCIL and evaluate it in detail and compare it to CIL in Section VI. Section VII provides a discussion on experimental validation. Finally, we conclude the article and discuss future prospects in Section VIII.

II. BACKGROUND AND RELATED WORK

While there are many different definitions and views on data quality, the most widely accepted definition is based on ''fitness for purpose'' [6], [20], [22], [29]. Specifically, Mocnik *et al.* define ''fitness for purpose'' as the affordance of data to be interpreted and used in a context that renders a certain usage, that is, the purpose, possible [26].

Adopting the point of view, this study handles data quality in relation to a specific ''purpose'', namely ''construction of effort/productivity estimation models''. The reason for choosing this purpose is (i) the importance of effort/productivity estimation in software project planning and (ii) the large number of attempts in effort/productivity estimation from *empirical data* (i.e. based on actual historical data sets) [1], [2], [12], [30], [31], [36].

Bosu *et al.* [6] point out that data quality issues in ESE can be broken-down into three main classes as (i) accuracy, (ii) relevance and (iii) provenance. Specifically, accuracy refers to the correctness of the data,[3] whereas relevance refers to the appropriateness of data for developing a model and provenance refers to the accessibility and trustworthiness of data.[4]

The data quality assessment framework of this study excludes the use of non-relevant and inaccessible data, which can be associated also with the flaws in scientific procedure rather than issues with the data set. In that respect, we focus mainly on the accuracy aspect in determining data quality. Note also that from the point of view of our purpose (i.e. effort/productivity estimation), accuracy is particularly important [6], since high estimation accuracy is essential in efficient project planning and control.

According to [6], accuracy issue can further be broken-down into five sub-issues as outliers, noise, inconsistency, incompleteness and redundancy. We recognize that the aspects like outliers, noise, incompleteness, and redundancy are important factors that determine accuracy and, in turn, data quality. Nevertheless, we let them remain beyond the scope of this investigation and focus on data inconsistency in determining quality of a data set.[5]

Liebchen and Shepperd [19] state that inconsistent data is the one that cannot be easily explained. More specifically, Bosu *et al.* [6] state that inconsistency means a lack of harmony between different parts or elements in a data set (i.e. instances conflicting within themselves or between each other).

According to the survey by Phannachitta *et al.*, at the time of their study there was no existing software metric that could directly quantify the level of consistency in ESE data sets [28]. This lack of procedure motivated Phannachitta *et al.* to develop a software metric for evaluating the inconsistency level of a data set. In that respect, they proposed a metric called Case Inconsistency Level (CIL) [28]. However, as explained in Sections IV and IV-F, CIL has some serious problems, which this study identifies and offers a solution for.

[1]Here, by ''efficient execution mode'' we refer to the incorporation of SCIL with the best performing combination of distance metric and pre-processing methods (among those that we experimented with).

[2]SeaCraft stands for ''Software Engineering Artifacts Can Really Assist Future Tasks'', is an open access repository accumulating various software development project data sets provided to software engineering researchers and practitioners for analyzing/tackling real-life challenges.

[3]Namely, absence of noise.

[4]In other words, provenance is related to the possibility of experimental replication.

[5]In that respect, when we mention ''quality of a data set'' in the rest of text, we refer specifically ''its data consistency''.

III. REVISITING CIL

In this section, we will first define the variables and concepts essential for computing CIL. We will then explain the basic idea underlying CIL and provide its formal definition.

Let D be a data set of N software projects,

$$D = \{\mathbf{p}_i,\ 1 \leq i \leq N\}, \tag{1}$$

where $\mathbf{p}_i$ stands for the i^{th} project (or equivalently its feature vector).

Let the feature f_{m^*} represent the estimation target and consider that all other features, i.e. f_m such that $m \neq m^*$, are disposable for estimating its value. Henceforth, we refer to f_{m^*} as "estimation target variable" (or simply as "target variable") and f_m as "estimator variables".

Suppose that $\mathbf{P}_D$ stands for the set which contains all possible pairs of different projects belonging to the data set D,

$$\mathbf{P}_D = \left\{\mathbf{p}_{ij} \mid \mathbf{p}_i, \mathbf{p}_j \in D,\ i \leq j\right\}, \tag{2}$$

where $\mathbf{p}_{ij}$ is simply an unordered pair $(\mathbf{p}_i, \mathbf{p}_j)$.[6]

A. CONCEPTS ESSENTIAL IN COMPUTING CIL

In this section, we will explain two concepts which are essential in computing CIL, namely the Interpolated Value Difference Metric (IVDM) and normalized rank of relative similarity.

1) INTERPOLATED VALUE DIFFERENCE METRIC

Let $Q(v)$ denote a function, which discretizes continuous input values v into C intervals with an equal width of δ. In addition, suppose that $\bar{Q}(v)$ is the mid-point of the discretization bin corresponding to input v.

Let $P'(Q(\mathbf{p}[f_{m^*}]) = c|\mathbf{p}[f_m])$ denote the conditional probability that the estimation target variable f_{m^*} is mapped to the discretization bin c given the value of the estimator variable f_m. This conditional probability can be expressed as in Equation 3, shown at the bottom of the page.

Interpolated Value Difference Metric (IVDM) is defined in terms of the difference in such conditional probabilities,

$$\mathbf{IVDM}\left(\mathbf{p}_{ij}, f_{m^*}\right) = \sum_{f_m}\sum_{c=1}^{C}\left(P'\Big(Q\big(\mathbf{p}_i[f_{m^*}] = c \mid \mathbf{p}_i[f_m]\big)\Big) - P'\Big(Q\big(\mathbf{p}_j[f_{m^*}] = c \mid \mathbf{p}_j[f_m]\big)\Big)\right)^2 \tag{4}$$

[6]Note that $(\mathbf{p}_i, \mathbf{p}_j)$ and $(\mathbf{p}_j, \mathbf{p}_i)$ are essentially the same. In such, we opt for setting $i \leq j$ in Equation 2 so as to eliminate the redundancy in notation.

2) NORMALIZED RANK OF RELATIVE SIMILARITY

The normalized rank of relative similarity is denoted with d_{NR} and computed based on the probabilistic similarity measure IVDM.

Let S be a set of real numbers and suppose that $\mathbf{rank}(s, S)$ returns the index of one of its elements s, when S is sorted in ascending order. Namely,

$$\mathbf{rank}(s, S) = \#\left(\left\{s' | s > s', s' \in S\right\}\right), \tag{5}$$

where $\#(\cdot)$ returns the number of elements of a set.

In order to compute d_{NR} relating to target variable f_{m^*}, firstly IVDM values concerning all pairs of projects in $\mathbf{P}_D$ (i.e. $\mathbf{IVDM}\ (\mathbf{P}_D)$) are computed. These values are then ranked using Equation 5 and normalized as in Equation 6,

$$d_{NR}\left(\mathbf{p}_{ij}, f_{m^*}\right) = \frac{\mathbf{rank}\left(\mathbf{IVDM}\left(\mathbf{p}_{ij}, f_{m^*}\right),\ \mathbf{IVDM}\left(\mathbf{P}_D, f_{m^*}\right)\right)}{\#(\mathbf{P}_D) - 1}. \tag{6}$$

Note that here the rank value in the numerator is inherently between 0 and $\#(\mathbf{P}_D) - 1$, which means that $d_{NR} \in [0, 1]$.[7]

The normalized rank of relative similarity is often considered to be a better distance function than the conventional Euclidean distance when used in predictive models [3], [4].

B. DEFINITION OF CIL

According to [28], a project pair $\mathbf{p}_{ij}$ is regarded to be inconsistent, if at least one of the following conditions is satisfied:

- (R1) $\mathbf{p}_i$ and $\mathbf{p}_j$ are dissimilar in terms of the target variable f_{m^*}, although they are very similar in terms of the estimator variables f_m.
- (R2) $\mathbf{p}_i$ and $\mathbf{p}_j$ are similar in terms of the target variable f_{m^*}, although they are dissimilar in terms of the estimator variables f_m.

In addition, a "consistent data set" is considered to be a data set *free of inconsistent pairs*, i.e. involving no cases of (R1) or (R2). On the other hand, an "inconsistent data set" is a data set with non-zero cases of (R1) or (R2), where the level of its inconsistency can simply be evaluated in terms of the rate of inconsistent pairs to all pairs.

In that respect, for assessing the level of inconsistency of a data set with CIL, it is necessary to find the number of cases associated with (R1) and (R2). To that end, the similarity or dissimilarity of each project pair $\mathbf{p}_{ij}$ concerning (1) the target variable f_{m^*} and (2) estimator variables f_m need to be judged.

[7]In other words, Equation 6 applies a MinMax normalization on the rank given by Equation 5.

$$P'\Big(Q\big(\mathbf{p}[f_{m^*}]\big) = c \mid \mathbf{p}[f_m]\Big) = P\Big(Q\big(\mathbf{p}[f_{m^*}]\big) = c \mid Q\big(\mathbf{p}[f_m]\big)\Big) + \frac{\mathbf{p}[f_m] - \bar{Q}\big(\mathbf{p}[f_m]\big)}{\delta} \cdot \left(P\Big(Q\big(\mathbf{p}[f_{m^*}]\big) = c \mid Q\big(\mathbf{p}[f_m]\big)\Big) - P\Big(Q\big(\mathbf{p}[f_{m^*}]\big) = c \mid Q\big(\mathbf{p}[f_m]\big) + 1\Big)\right) \tag{3}$$

1) JUDGING SIMILARITY OF THE TARGET VARIABLE

The similarity of a project pair $\mathbf{p}_{ij}$ in terms of the target variable f_{m^*} is assessed based on their *relative distance*. Specifically, we denote the relative distance of a project pair $\mathbf{p}_{ij}$ in terms of the target variable f_{m^*} with $d_R\left(\mathbf{p}_{ij}, f_{m^*}\right)$,

$$d_R\left(\mathbf{p}_{ij}, f_{m^*}\right) = \frac{\left|\mathbf{p}_i[f_{m^*}] - \mathbf{p}_j[f_{m^*}]\right|}{\frac{\mathbf{p}_i[f_{m^*}]+\mathbf{p}_j[f_{m^*}]}{2}}. \tag{7}$$

If $d_R\left(\mathbf{p}_{ij}, f_{m^*}\right)$ is smaller than 1, i.e.

$$d_R\left(\mathbf{p}_{ij},\ f_{m^*}\right) < 1, \tag{8}$$

then the project pair $\mathbf{p}_{ij}$ is considered to have *similar* target variables. Otherwise (i.e. $d_R\left(\mathbf{p}_{ij},\ f_{m^*}\right) \geq 1$), it is regarded to have *dissimilar* target variables.

Let $\overset{\approx}{\mathbf{P}}_{D,m^*}$ denote the set of project pairs in D with similar target variable f_{m^*}:

$$\overset{\approx}{\mathbf{P}}_{D,m^*} = \left\{\mathbf{p}_{ij} \mid d_R\left(\mathbf{p}_{ij},\ f_{m^*}\right) < 1,\ \mathbf{p}_{ij} \in \mathbf{P}_D\right\}. \tag{9}$$

Moreover, let $\overset{\not\approx}{\mathbf{P}}_{D,m^*}$ denote the set of project pairs with dissimilar target variables f_{m^*}:

$$\overset{\not\approx}{\mathbf{P}}_{D,m^*} = \left\{\mathbf{p}_{ij} \mid d_R\left(\mathbf{p}_{ij}, m^*\right) \geq 1,\ \mathbf{p}_{ij} \in \mathbf{P}_D\right\}. \tag{10}$$

2) JUDGING SIMILARITY OF ESTIMATOR VARIABLES

The similarity of a project pair $\mathbf{p}_{ij}$ in terms of the estimator variables f_m is assessed based on their normalized rank of relative similarity d_{NR}.

Let $\overset{\approx}{\mathbf{P}}_{D,m}$ denote the set of project pairs with similar estimator variables f_m.

$$\overset{\approx}{\mathbf{P}}_{D,m} = \left\{\mathbf{p}_{ij} \mid d_{NR}\left(\mathbf{p}_{ij}, f_m\right) < \alpha,\ \mathbf{p}_{ij} \in \mathbf{P}_D\right\}, \tag{11}$$

where α is a threshold in the interval between 0 and 1. In addition, suppose that $\overset{\not\approx}{\mathbf{P}}_{D,m}$ is the set of project pairs with dissimilar estimator variables f_m.

$$\overset{\not\approx}{\mathbf{P}}_{D,m} = \left\{\mathbf{p}_{ij} \mid d_{NR}\left(\mathbf{p}_{ij}, f_m\right) \geq 1-\alpha,\ \mathbf{p}_{ij} \in \mathbf{P}_D\right\}. \tag{12}$$

Note that when $\alpha \leq d_{NR}\left(\mathbf{p}_{ij}, f_m\right) < 1-\alpha$ we do not regard the project pair $\mathbf{p}_{ij}$ to be neither similar nor dissimilar in terms of the estimator variables.

3) EXPLICIT FORMULATION OF CIL

Let $\mathbf{P}_{D,R1}$ denote the set of inconsistent project pairs of D, which satisfy (R1). Then, $\mathbf{P}_{D,R1}$ can be written as

$$\mathbf{P}_{D,R1} = \left\{\mathbf{p}_{ij} \middle| \ \mathbf{p}_{ij} \in \overset{\not\approx}{\mathbf{P}}_{D,m^*},\ \mathbf{p}_{ij} \in \overset{\approx}{\mathbf{P}}_{D,m}\right\}. \tag{13}$$

Let $\mathbf{P}_{D,R2}$ denote the set of inconsistent project pairs of D, which satisfy (R2). $\mathbf{P}_{D,R2}$ is simply

$$\mathbf{P}_{D,R2} = \left\{\mathbf{p}_{ij} \middle| \ \mathbf{p}_{ij} \in \overset{\approx}{\mathbf{P}}_{D,m^*},\ \mathbf{p}_{ij} \in \overset{\not\approx}{\mathbf{P}}_{D,m}\right\}. \tag{14}$$

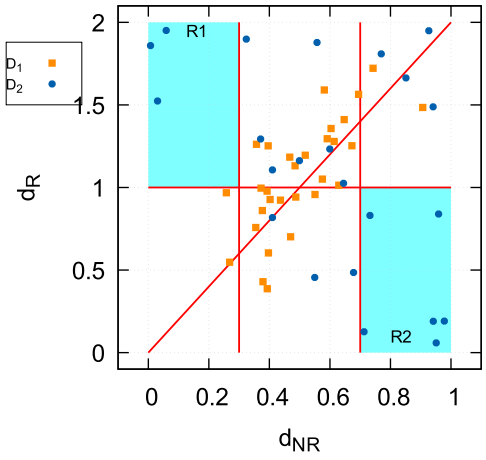


FIGURE 1. Distribution of relative distance d_R of the target variable f_{m^*} and normed rank distance d_{NR} of estimator variables f_m concerning two hypothetical data sets D_1 and D_2. The solid curve shows the linear fit for both data sets. The vertical dashed lines illustrate $d_{NR} = \alpha$ and $d_{NR} = 1-\alpha$ for $\alpha = 0.3$ and the horizontal dashed line marks $d_R = 1$.

Putting together the information on similarity/dissimilarity of the target variable and estimator variables, CIL concerning the data set D can be expressed as

$$\mathbf{CIL}(D) = \frac{\#\left(\mathbf{P}_{D,R1} \bigcup \mathbf{P}_{D,R2}\right)}{\#(\mathbf{P}_D)}. \tag{15}$$

For two hypothetical data sets D_1 and D_2, Figure 1 illustrates the distribution of relative distance d_R of target variables f_{m^*} and normed rank distance d_{NR} of estimator variables f_m. Here, the shaded region in the upper left corner corresponds to (R1) and the region in the lower right corner corresponds to (R2) mentioned in Section III-B.

One may see in Figure 1 that there are no project pairs in D_1, which meet the conditions stated in (R1) and (R2). Thus, D_1 is free of inconsistencies and has high data quality. On the other hand, there are a non-zero number of cases in D_2 corresponding to (R1) and (R2). Such inconsistencies indicate that D_2 has a lower data quality than D_1.

IV. FOLLOW-UP EVALUATION OF CIL

In the follow-up evaluation of CIL, we keep certain properties of evaluation same as Phannachitta *et al.* [28] and change certain others to get a better insight into the performance of CIL. In particular, the estimation target variable, estimator variables and performance evaluation metrics are kept the same to have a fair comparison with the original study [28].[8] On the other hand, the number and variety of data sets are increased to provide a more comprehensive evaluation. In addition, the experiment procedure is modified by diversifying experimental runs with cross-validation and testing with various values of the threshold α. In this section, we first elaborate on each of these experimental factors and then present experimental results.

[8]Note that in the experiments, we use $C = 5$ as recommended in the original IVDM study [35].

TABLE 2. Reference data sets employed in this study.

Data set	# of projects N	# of project features
China [24]	499	12
Coc81dem [5]	63	18
Desharnais [10]	77	9
Maxwell [23]	62	26
Miyazaki94 [25]	48	8
Nasa93 [24]	93	26

A. DATA SETS AND PRE-PROCESSING

In [28], four data sets are employed in assessing the efficiency of CIL, where one particular data set, i.e. Kemerer [37], is a very small one, containing only 15 projects.

Here, in this follow-up evaluation of CIL, we conduct more extensive experiments using six reference data sets shown in Table 2. Note that these data sets are published as part of the SeaCraft repository [24] and are relatively large, containing at least 48 projects. And China data set contains projects from many Chinese companies, Coc81dem and Nasa93 data sets contain NASA projects, Desharnais data set contains projects from a Canadian company, Maxwell data set contains projects from banks in Finland, Miyazaki data set contains projects developed in COBOL language.

As for pre-processing, no particular operation is carried out following the same strategy of [28]. Note that this helps us make a direct and fair comparison with the results reported in [28]. In addition, it helps us to confine our analysis to the evaluation of CIL and especially to the cases where it fails more frequently, rather than digressing the discussion towards what pre-processing operations are necessary or how they should be tuned etc.

Nevertheless, we of course recognize that there may be certain data treatment techniques, which may reflect as an improvement on the efficacy of CIL. In order to address such aspects, we choose in Section VI certain pre-processing operations and apply them on both CIL and the proposed metric SCIL. In this way, we point out how much improvement can be obtained in CIL due to pre-processing, and how much the proposed metric can improve further over that.

B. TARGET VARIABLE AND ESTIMATION MODEL

Similar to [28], in the follow-up evaluation, we consider ''effort'' to be target variable and use all other variables in the data sets to estimate its value.

As for the estimation method, we used Classification and Regression Trees (CART) [17] with tree pruning based on the error rates in cross-validation, since it is discussed by Phannachitta *et al.* to be the most efficient estimator [28].[9]

C. EXPERIMENT PROCEDURE

In the experiments, we conducted 3 repetitions of 3-fold cross-validation for each data set, as illustrated in Figure 2. Specifically, we randomly split a source data set into 3 subsets. Then, we conduct three rounds of model construction and model evaluation, each of which employs two subsets in model construction. In each case, the remaining subset is used for evaluation. Eventually, we used 54 data samples derived from the six data sets shown in Table 2. The CIL values are calculated based on the fit subset in each case and their correlation with estimation error is examined.

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 a better understanding of stability of performance.

[9]Note that in Section VI-B we reassess this claim.

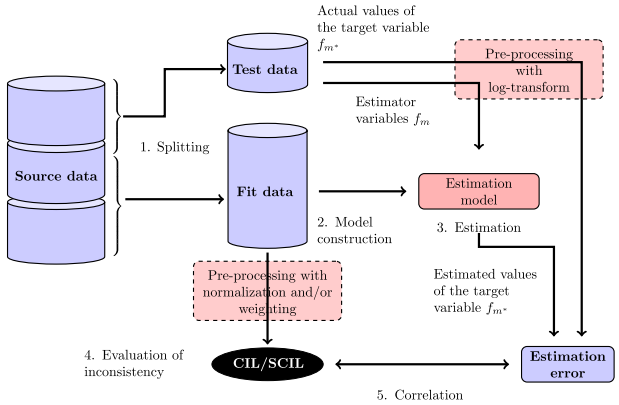


FIGURE 2. Experiment procedure. The blue blocks are common in all experiments, whereas the solid pink block (i.e. estimation model) varies between experiments. The dashed pink blocks are not used in the follow-up evaluation of CIL as in [28]. But they are used in used in comparing CIL and SCIL in Section VI-F.

D. ESTIMATION ERROR

For measuring estimation error, similar to Gupta *et al.* we use Mean Magnitude of Relative Error (MMRE) [12], which is commonly used in effort estimation studies. In particular, we denote the Magnitude of Relative Error (MRE) of project $\mathbf{p}_i$ concerning an estimation target variable of f_{m^*} with $\mathbf{MRE}(\mathbf{p}_i[f_{m^*}])$ and compute it as,

$$\mathbf{MRE}(\mathbf{p}_i[f_{m^*}]) = \frac{|\mathbf{p}_i[f_{m^*}] - \widehat{\mathbf{p}}_i[f_{m^*}]|}{\mathbf{p}_i[f_{m^*}]}, \tag{16}$$

where $\widehat{\mathbf{p}}_i[f_{m^*}]$ denotes the estimated value of f_{m^*} for project $\mathbf{p}_i$. Then, MMRE concerning a data set D is computed as the mean value of MREs concerning all projects in D,

$$\mathbf{MMRE}(D) = \frac{1}{\#(D)} \left(\sum_{\mathbf{p}_i \in D} \mathbf{MRE}\left(\mathbf{p}_i[f_{m^*}]\right) \right). \tag{17}$$

E. ASSESSMENT OF EFFICACY OF CIL

The efficacy of CIL is assessed based on its correlation with MMRE. Namely, if a data set is of high quality, then the rate of data points satisfying (R1) and (R2) should be small, yielding a low CIL. Similarly, for a data set of high quality, the estimation error (i.e. MMRE) should be small. On the contrary, a low quality data set is expected to have more data points satisfying (R1) and (R2), thus to have a higher CIL, and also to suffer from high estimation error (i.e. high MMRE).

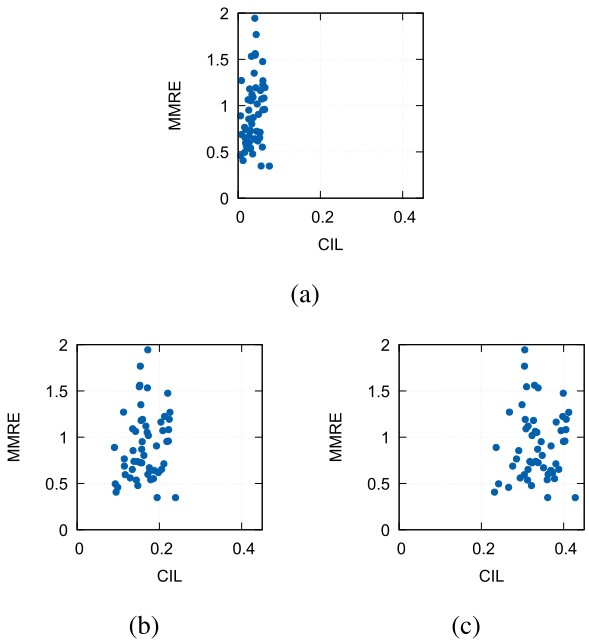


FIGURE 3. Relationship between CIL and MMRE in effort estimation with thresholds of (a) $\alpha = 0.1$, (b) $\alpha = 0.3$ and (c) $\alpha = 0.5$.

TABLE 3. Correlation coefficients R concerning MMRE and CIL values in the follow-up evaluation (without pre-processing).

Data set	R		
	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
China [24]	-0.170	0.106	0.058
Coc81dem [5]	-0.159	-0.120	-0.439
Desharnais [10]	-0.768	-0.578	-0.640
Maxwell [23]	-0.737	-0.431	-0.265
Miyazaki94 [25]	0.194	0.387	0.261
Nasa93 [24]	-0.308	-0.011	-0.110

Therefore, CIL and MMRE are expected to be positively correlated. Specifically, the correlation between CIL and MMRE is represented with R and computed as

$$R = \frac{\mathbf{Cov(CIL, MMRE)}}{\sigma_{\mathbf{CIL}}\sigma_{\mathbf{MMRE}}}. \tag{18}$$

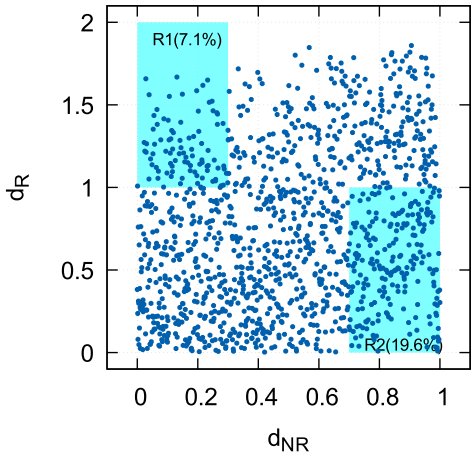
F. RESULTS OF THE FOLLOW-UP EVALUATION

Figure 3 and Table 3 show the results of our follow-up evaluation. By examining the distribution of CIL and MMRE values in Figure 3, one may judge in a qualitative way that there is no strong correlation between them for any of the α values.

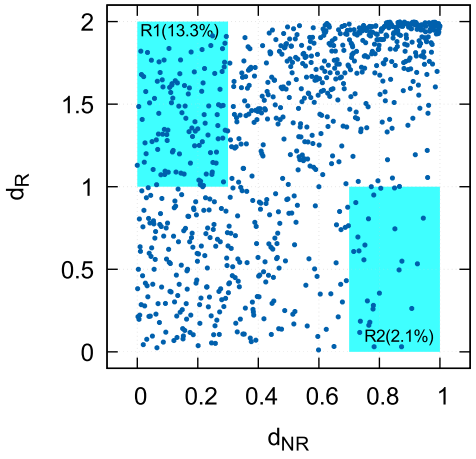
In addition, Table 3 proves in a quantitative way that there is very little correlation between CIL and MMRE ($R < 0.4$) in all cases. In that respect, the current form of CIL is shown not to be effective in assessing data quality of software project data sets.

V. DEFINITION OF SCIL

In order to improve the efficacy of CIL, we first contemplate on the reasons for its poor performance in the follow-up analysis. To that end, we focus on two particular data sets, namely Desharnais [10] and Coc81dem [5], and take a closer look at their properties.



(a) Desharnais data set (CIL = 0.239, MMRE = 0.347).



(b) Coc81dem data set (CIL = 0.154, MMRE = 1.768).

FIGURE 4. Distribution of relative distance d_R of the target variable and normed rank distance d_{NR} of estimator variables for (a) Desharnais and (b) Coc81dem data sets. These values are obtained with 3-fold cross-validation and $\alpha = 0.3$.

The distribution of relative distance d_R of the target variable and normed rank distance d_{NR} of estimator variables for Desharnais data set is presented in Figure 4-(a). For the threshold value $\alpha = 0.3$, the relating CIL value is found to be 0.239, whereas MMRE is found to be 0.347.

The distribution of relative distance d_R of the target variable and normed rank distance d_{NR} of estimator variables for Coc81dem data set is presented in Figure 4-(b). Here, the CIL and MMRE values are found to be 0.154 and 1.768, respectively, for the same threshold value $\alpha = 0.3$.

Based on CIL values, Coc81dem seems to have better data quality than Deshairnais. However, based on the MMRE

values, Deshairnais seems to have a superior data quality over Coc81dem, a contradicting conclusion to CIL.

The reason for this contraction is considered to be the assumption of CIL that the contributions of the data points lying in the two regions of (R1) and (R2) to deterioration of data quality are virtually the same. We claim that the data in (R1) and (R2) contribute to data inconsistency in different ways and elaborate on our reasoning based on Figure 4.

The (R1) region, which corresponds to the projects with similar target variables and dissimilar estimator variables, accommodates 7.1% of the data points in Figure 4-(a), whereas it accommodates 13.3% of the data points in Figure 4-(b). On the other hand, the (R2) region, which corresponds to the projects with similar target values but dissimilar estimator variables, accommodates 16.9% of the data in Figure 4-(a), whereas it accommodates 2.1% of the data in Figure 4-(b). In other words, the percentage of data points in (R2) of Figure 4-(a) (16.9%) is much larger than that of Figure 4-(b) (2.1%).

As a matter of fact, concerning effort, it is not uncommon that dissimilar estimator variables have very similar effort values. This indicates that it is not appropriate to focus on (R2) region in evaluating data inconsistency and that it is better to pay regard rather to (R1) region.

Based on such contemplation, we propose improving CIL by focusing on inconsistencies due to data points in the (R1) region. We call the improved metric Similar Case Inconsistency Level (SCIL) and define it explicitly as follows:

$$\mathbf{SCIL}(D) = \frac{\#\left(\mathbf{P}_{D,R1}\right)}{\#(\mathbf{P}_D)}. \tag{19}$$

VI. PERFORMANCE COMPARISON OF CIL AND SCIL

In this section, we carry out a new performance of assessment for CIL taking in consideration the impact of several data pre-processing operations. In addition, we test SCIL with exactly the same conditions concerning this secondary re-assessment of CIL and ensure an objective (unbiased) comparison.

A. DATA SETS AND PRE-PROCESSING

In evaluating the efficacy of the proposed metric SCIL and comparing it to that of CIL, we used the data sets reported in Table 2 and the experiment procedure with 3-fold cross-validation defined in Section IV-C (see also Figure 2).

However, unlike [28] and Section IV,we considered three kinds of pre-processing operations and applied them on the data set before computing CIL as well as SCIL. Specifically, the pre-processing operations are normalization, weighting and log-transformation, which are detailed in Sections VI-A1, VI-A2 and VI-A3, respectively.

1) NORMALIZATION OF FEATURES

Although normalization is a crucial part of data pre-processing in software analytics, Phannachitta *et al.* do not consider any normalization in [28]. However, the data sets under investigation contain variables from significantly different value ranges. In that respect, normalization is necessary to make sure that the project variables have a balanced influence on the calculation of distance metrics. In this study, we consider two alternatives for normalization, i.e. MinMax normalization and Z-score normalization (also known as standardization, see Appendix II) for determining the more effective normalization scheme for the target variable in focus.

2) WEIGHTING

Weighting is another common data pre-processing tool in the analysis of software industry data and is omitted in [28]. Yet, it is plausible that different estimator variables are likely to have different influence on the target variable, which can be dealt with by asserting different weights on them through the correlation coefficient. In our study, we opt for weighting the estimator variables as follows:

$$\mathbf{weighted}(\mathbf{p}_i[f_m]) = \mathbf{p}_i[f_m] \times \underset{D}{\mathbf{Corr}}(f_{m^*}, f_m), \tag{20}$$

where $\underset{D}{\mathbf{Corr}}(f_{m^*}, f_m)$ is Pearson's correlation coefficient between the estimation target f_{m^*} and estimator variable f_m in data set D.

Note that in analyzing the effect of weighting in Section VI-F, we report the results of the experiments with and without weighting and point out the benefits and drawbacks.

3) LOG TRANSFORM

Kitchenham and Mendes empirically showed that logarithmic transformation helps in improving effort estimation accuracy [16]. Thus, we decided to perform experiments involving also a logarithmic transformation preceding estimation of target variables. Note that when we employ logarithmic transformation, it is applied on both the target variable and the estimator variables prior to model construction.[10] In Section VI-B we compare the performance of several estimation schemes with and without log transformation, and select the best one to be deployed in the experiments.

B. TARGET VARIABLES AND ESTIMATION MODELS

As mentioned in Section II, it is common in ESE to use ''effort'' and/or ''productivity'' as target variables in analyzing software project data sets. In that respect, in investigating the efficacy of SCIL and comparing it to that of CIL, we diversify target variables by considering both ''effort'' and ''productivity''.

While effort refers to the amount of the professional activity in man-hours to complete a project (e.g. by developers, testers etc.), productivity is generally defined as the size of development per unit effort. Specifically, we employ the following definition for productivity,

$$\text{Productivity} = \frac{FP}{\text{Effort}}, \tag{21}$$

where FP stands for the ''function point''.

[10]For variables containing 0, the offset value 1.0 is added before the transformation.

TABLE 4. Ranking of effort estimation models with respect to ascending values of MMRE.

Rank	Log transform	Estimation model	Average of MMRE
1	+	Random Forest	**0.660**
2	+	Linear regression	0.687
3	+	CART + Tree pruning	0.912
4	-	Random Forest	1.439
5	-	CART + Tree pruning	2.228
6	-	Linear regression	4.638

Note that not all data sets in Table 2 involve *FP* as a project feature. However, these sets involve ''lines of code'' (*LOC*), which can be used as a replacement in Equation 21 (for the purpose of this study). In that respect, if the data set under investigation involves *FP* as a project feature, we compute productivity as in Equation 21, and otherwise we replace *FP* with *LOC*.

In effort estimation studies, linear regression technique is commonly used [14], [16]. Besides, CART with tree pruning based on the error rates in cross-validation [17] (hereafter denoted as CART + Tree pruning) is shown to be the best method in the original CIL study [28]. Furthermore, Random Forest technique is shown to be a promising method in recent effort estimation studies [1], [2], [33]. Therefore, we employ all these models and select the model with the smallest estimation error to evaluate and compare CIL and SCIL.

Table 4 shows the result of effort estimation of the above models with/without logarithmic transformation.[11] According to Table 4, the minimum estimation error is obtained by the Random Forest technique with logarithmic transformation. Note that as mentioned in Section IV-B, Phannachitta *et al.* claim the best performing estimator to be CART + Tree pruning in [28]. However, based on the results presented in Table 4, we observe that Random Forest with logarithmic transformation performs even better. Therefore, in Section VI-F, we carry out performance evaluation and comparison of CIL and SCIL based on Random Forest technique with logarithmic transformation.

Finally, we note that the integration of pre-processing operations and the improvement of the estimator model are expected to enhance the performance of CIL reported in Section VI-F. In order to provide an insight how much these two modifications contribute to it, in Appendix III, we first integrate a pre-processing module to the original framework of CIL and then change the former estimator (CART+Tree pruning) with a more competent one (Random Forest). By this means, we assess the improvement that can be expected on CIL by applying such extras/fine-tuning.

C. EXPERIMENT PROCEDURE

Similar to Section IV, we repeat 3-fold cross-validation 3 times yielding 54 different experimental runs (see also Section IV-C). In addition, we employ different combinations of distance functions, normalization techniques and weighting or not to find which combination works best, and ensure a comprehensive assessment.

As explained in Section III-B, Phannachitta *et al.* employ IVDM in computing $d_{NR}(\mathbf{p}_{ij}, f_{m^*})$. However, it is not clear whether IVDM is the best distance function to evaluate relative difference of projects or whether alternative metrics can perform better. Therefore, we compute the proposed SCIL metric as well as the previously proposed CIL metric using three different distance functions: Euclidean Distance d_E, cosine distance d_C (see Appendix I), and IVDM. We contrast the performance values for figuring out which distance function is most appropriate.

In addition, at each run three different threshold values $\alpha \in \{0.1, 0.3, 0, 5\}$ are used in computing d_{NR}. Namely, the similarity or dissimilarity of estimator variables are judged at varying degrees. Namely, higher values of α imply a broad range for inconsistency (for that matter also consistency), whereas lower values of α leave more space for ambiguity (i.e. regarding the estimator variables as neither similar nor dissimilar).

D. ESTIMATION ERROR

In assessing the performance of CIL and SCIL, we use mean MMRE[12] as the estimation error similar to the follow-up evaluation of CIL reported in Section IV.

E. ASSESSMENT OF EFFICACY OF SCIL

Based on the apprehension that a more consistent data set is likely to contribute to a more accurate estimation model, we evaluate CIL and SCIL metrics by observing the correlation between them and estimation errors concerning the models built from real project data sets. We expect that the data sets with higher data quality (i.e. characterized by lower values of CIL or SCIL) will on average have a lower estimation error than that of the data sets with lower data quality.[13]

F. RESULTS ON COMPARISON OF CIL AND SCIL

In this section, we assess the performance of CIL and SCIL metrics from the point of view of the ''fitness for purpose'' [22], where the ''purpose'' is determined as ''effort estimation'' and ''productivity estimation'' within the scope of this study. To that end, we analyze the correlation R between the estimation error (of effort and productivity) quantified in terms of MMRE and the values of two data inconsistency metrics CIL and SCIL. Section VI-F1 presents the results relating to effort estimation, and Section VI-F2 illustrates the results for productivity estimation.

[11] Note that + and - denote an experiment with and without log-transform, respectively.

[12] Note that here ''mean'' refers to the average over all data sets illustrated in Table 2.

[13] In other words, CIL/SCIL values are expected to be positively correlated with estimation error, where a higher correlation indicates a better assessment of data quality.

TABLE 5. Correlation between MMRE and CIL concerning target variable of *effort*.

Distance	Norm.	Weigh.	R		
			$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
d_E	Z-score	+	**0.459**	0.439	0.320
d_E	Z-score	-	0.090	**0.285**	0.213
d_E	MinMax	+	0.450	0.479	**0.494**
d_E	MinMax	-	0.142	0.204	**0.265**
d_C	Z-score	+	-0.127	**0.049**	0.048
d_C	Z-score	-	-0.117	-0.010	**0.046**
d_C	MinMax	+	**-0.315**	-0.368	-0.338
d_C	MinMax	-	**-0.157**	-0.192	-0.162
IVDM	*	*	**0.381**	0.372	0.290

TABLE 6. Correlation between MMRE and SCIL concerning target variable of *effort*.

Distance	Norm.	Weigh.	R		
			$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
d_E	Z-score	+	0.604	**0.615**	0.570
d_E	Z-score	-	0.531	**0.534**	0.485
d_E	MinMax	+	**0.578**	0.575	**0.578**
d_E	MinMax	-	0.527	**0.530**	0.524
d_C	Z-score	+	0.391	**0.472**	0.456
d_C	Z-score	-	0.486	**0.522**	0.515
d_C	MinMax	+	0.079	0.218	**0.340**
d_C	MinMax	-	0.361	0.392	**0.424**
IVDM	*	*	**0.629**	0.624	0.583

1) EFFORT ESTIMATION

Concerning effort estimation, Tables 5 and 6 present the correlation of MMRE with CIL and SCIL, respectively.[14] Note that in these tables as pre-processing scheme we consider different combinations of distance metrics, normalization and weighting schemes.

From Tables 5 and 6, we can see that the correlation of SCIL and MMRE is considerably better than that of CIL and MMRE for any data pre-processing scheme. In other words, between the corresponding values of CIL and SCIL in the two tables, the one of SCIL is always larger.

In addition to serving for performance comparison, Tables 5 and 6 help us also identify an efficient pre-processing scheme for CIL and SCIL.[15] In that respect, we start by taking a closer look at Table 6, since the main focus of our study is SCIL. The values in bold are highest in their corresponding rows. In addition, the highlighted row corresponding to IVDM attains the highest correlation overall ($R = 0.629$) with $\alpha = 0.1$. Actually, IVDM attains the highest value for any threshold value of α (i.e. in every column of Table 6). In addition, Euclidean distance d_E coupled with Z-score normalization and weighting is observed to attain comparable results to those of IVDM concerning all α values.

[14]Note that correlation is computed in the same way as in Equation 18, but by replacing CIL with SCIL. Note also that + and - denote an experiment with and without weighting, respectively, whereas * denotes that normalization or weighting does not apply to IVDM.

[15]Since virtually there are an infinite number of pre-processing possibilities, we can not claim that we identified the *optimal* scheme. Nevertheless, we can say that we have found one with a fairly good performance.

TABLE 7. Correlation between MMRE and CIL concerning target variable of *productivity*.

Distance	Norm.	Weight.	R		
			$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
d_E	Z-score	+	**0.509**	0.388	0.356
d_E	Z-score	-	0.354	0.331	**0.437**
d_E	MinMax	+	**0.518**	0.384	0.338
d_E	MinMax	-	0.333	0.376	**0.398**
d_C	Z-score	+	-0.394	-0.374	**-0.337**
d_C	Z-score	-	-0.116	-0.169	**-0.095**
d_C	MinMax	+	**-0.139**	-0.140	-0.156
d_C	MinMax	-	**0.015**	-0.127	-0.126
IVDM	*	*	**0.094**	-0.257	-0.313

TABLE 8. Correlation between MMRE and SCIL concerning target variable of *productivity*.

Distance	Norm.	Weight.	R		
			$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
d_E	Z-score	+	0.773	0.805	**0.814**
d_E	Z-score	-	0.721	0.790	**0.804**
d_E	MinMax	+	0.781	0.802	**0.814**
d_E	MinMax	-	0.731	0.808	**0.819**
d_C	Z-score	+	0.094	0.294	**0.564**
d_C	Z-score	-	0.463	0.592	**0.670**
d_C	MinMax	+	0.174	0.358	**0.558**
d_C	MinMax	-	0.391	0.518	**0.635**
IVDM	*	*	0.673	0.688	**0.698**

Regarding the variation on MMRE due to variations on α, we can say that 0.1 and 0.3 are superior to 0.5, since both IVDM and Euclidean distance d_E coupled with Z-score normalization and weighting attain $R > 0.6$, indicating a relatively high correlation between SCIL and MMRE (see Table 6). Overall, to compute SCIL metric for effort estimation purposes, IVDM or the Euclidean distance d_E coupled with Z-score normalization and weighting and $\alpha = 0.1$ or 0.3 can be recommended.

On the other hand, cosine distance d_C coupled with MinMax normalization and weighting shows the lowest correlation for any threshold value α. Note also that cosine distance d_C shows in general lower correlation (i.e. regardless of the pre-processing techniques) and therefore, it is not recommended to be used for the purpose of effort estimation.

Next, we also take a closer look at the results concerning CIL in Table 5. We can see that the values at the highlighted row are quite high in their corresponding columns (actually, the highest for $\alpha = 0.3$ and $\alpha = 0.5$ and very close to the highest for $\alpha = 0.1$). In that respect, the best performing combination of distance, normalization and weighting scheme is found to be Euclidean distance d_E coupled with MinMax normalization and weighting. Also, replacing MinMax normalization with Z-score normalization in this combination yields somewhat comparable results.

2) PRODUCTIVITY ESTIMATION

Concerning productivity estimation, Tables 7 and 8 present the correlation of MMRE with CIL and SCIL, respectively. We can see that the relation between Tables 7 and 8 is similar to the relation between Tables 5 and 6. Namely, similar

to effort, also for productivity estimation, the correlation between SCIL and MMRE is considerably better than that of CIL and MMRE for any data pre-processing scheme. Specifically, CIL attains the highest value of $R = 0.518$, whereas SCIL gets a maximum of $R = 0.814$. Moreover, the highest performance concerning different combinations of pre-processing schemes (i.e. bold values in Table 7) varies considerably for CIL (i.e. between -0.337 and 0.518), whereas for SCIL they are somewhat more stable (i.e. 0.564 and 0.819). In addition, between the corresponding values of CIL and SCIL in Tables 7 and 8, the one of SCIL is always larger.

Next, we take a closer look at Table 8 for identifying the best performing combination of pre-processing operations concerning SCIL. Unlike the results for effort estimation, the Euclidean distance d_E attains the highest correlation values regarding all three thresholds α, regardless of the normalization technique or weighting. The IVDM performs second best, and cosine distance d_C performs worst regardless of pre-processing.

Regarding the variation on MMRE due to variations on α, 0.3 or 0.5 are recommended in computing SCIL, since in all cases with Euclidean distance d_E, $R > 0.8$ is attained, which indicates to a high correlation between SCIL and MMRE. Taking in consideration also the results of effort estimation reported in Table 6, Euclidean distance function coupled with Z-score normalization and weighting with $\alpha = 0.3$ is highly recommended to compute SCIL metric for both effort and productivity estimation purposes.

Subsequently, we take a closer look at the results concerning CIL. In Table 7, we can see that Euclidean distance d_E coupled with MinMax normalization and weighting attains the highest value of correlation ($R = 0.518$). In that respect, the best performing pre-processing combination is exactly the same as the one for estimating effort (see Table 5). Moreover, similar to effort estimation, replacing MinMax normalization with Z-score normalization again yields somewhat comparable results.

Finally, we would like to draw the attention of the reader to the similarity of the distribution of the highest values in each row of Tables 5 and 7 and also Tables 6 and 8. One may see that the pattern is quite similar between Tables 5 and 7, whereas very different between Tables 6 and 8. We believe that this indicates the prominent effect of pre-processing on the performance of CIL. In other words, CIL is quite sensitive to the changes in experimental conditions and is likely to miss the actual role of input and/or target in estimation accuracy.

Based on the above discussion, we conclude that the proposed metric SCIL effectively quantifies the level of inconsistency of a data set, is considerably superior to CIL and appraises data inconsistency in a more resilient manner.

VII. 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 3 repetitions in the validation process to produce stable results. However, there is one possible issue of internal validity in this study. The issue is the single sampling method (3-fold cross-validation) we used. Our important future work is to employ other methods such as leave-one-out cross-validation to increase the validity of the result.

External validity refers to the generalization of the results. In this study, we address external validity by using 6 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. Our future work is to employ more data sets to increase the generalization of the results.

Construct validity refers to the relevance and capability of the observations and measurements in evaluating the posed hypothesis. In this study, we use single error measure MMRE to evaluate the target variable value estimation performance. It is our future work to employ other error measures to increase the validity of our work.

VIII. CONCLUSION AND FUTURE PROSPECTS

This study proposes an improved data quality metric SCIL that can quantify the level of inconsistency of a data set. Comparing the conventional CIL metric and the SCIL metric, we believe that the proposed SCIL metric is more suitable for the purpose of effort and productivity estimation as we have shown through experimental evaluation. Considering the experimental results, it is recommended to combine the Euclidean distance function be combined with the Z-score normalization and weighting (threshold $\alpha = 0.3$) to calculate the SCIL metric.

As future work, we will employ other methods such as leave-one-out cross-validation and other error measures to increase validity, and more data sets to increase generality.

APPENDIX I. DISTANCE METRICS

We denote Euclidean distance between projects $\mathbf{p}_i$ and $\mathbf{p}_j$ with $d_E(\mathbf{p}_{ij})$, where

$$d_E(\mathbf{p}_{ij}) = \sqrt{\sum_{f_m} \left(\mathbf{p}_i[f_m] - \mathbf{p}_j[f_m]\right)^2}. \tag{22}$$

In addition, we denote the cosine distance between the same pair with $d_C(\mathbf{p}_{ij})$, where

$$d_C(\mathbf{p}_{ij}) = \frac{\sum_{f_m} \mathbf{p}_i[f_m] \cdot \mathbf{p}_j[f_m]}{\sqrt{\sum_{f_m} \mathbf{p}_i^2[f_m]} + \sqrt{\sum_{f_m} \mathbf{p}_j^2[f_m]}}. \tag{23}$$

TABLE 9. Correlation coefficients R concerning MMRE and CIL values in the follow-up evaluation (with pre-processing and CART + Tree pruning).

Data set	R		
	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
China [24]	0.324	0.388	0.476
Coc81dem [5]	-0.435	-0.305	-0.106
Desharnais [10]	-0.592	-0.260	0.265
Maxwell [23]	-0.380	-0.277	-0.261
Miyazaki94 [25]	-0.230	-0.586	-0.593
Nasa93 [24]	0.196	0.382	0.459

TABLE 10. Correlation coefficients R concerning MMRE and CIL values in the follow-up evaluation (with pre-processing and Random Forest).

Data set	R		
	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$
China [24]	0.034	-0.130	-0.052
Coc81dem [5]	-0.242	-0.111	0.060
Desharnais [10]	-0.413	-0.005	0.608
Maxwell [23]	-0.395	-0.208	-0.357
Miyazaki94 [25]	-0.492	-0.227	-0.137
Nasa93 [24]	-0.277	0.102	0.208

APPENDIX II. NORMALIZATIONS

Let $\mathbf{p}_i$ be an arbitrary project from a data set D, f_m be an arbitrary feature and $\bar{\mathbf{p}}_i[f_m]$ be the MinMax normalized value of that feature relating project $\mathbf{p}_i$.

$$\bar{\mathbf{p}}_i[f_m] = \frac{\mathbf{p_i}[f_m] - f_{m,min}}{f_{m,max} - f_{m,min}} \tag{24}$$

where $f_{m,min}$ and $f_{m,max}$ are the minimum and maximum values of that feature over all projects in the data set.

$$f_{m,min} = \min_{\mathbf{p_i} \in D} (\mathbf{p}_i[f_m])$$
$$f_{m,max} = \max_{\mathbf{p_i} \in D} (\mathbf{p}_i[f_m])$$

Let $\bar{\bar{\mathbf{p}}}_i[f_m]$ be the z-normalized value of the feature f_m relating project $\mathbf{p}_i$.

$$\bar{\bar{\mathbf{p}}}_i[f_m] = \frac{\mathbf{p_i}[f_m] - \mu_i}{\sigma_i} \tag{25}$$

where μ_i is the mean value and σ_i is the standard deviation of that feature over projects in the data set.

APPENDIX III. THE EFFECT OF PRE-PROCESSING AND ESTIMATION METHOD ON THE PERFORMANCE OF CIL

In this section, we assess the improvement on CIL that can be expected by applying pre-processing operations and by improving the estimator performance.

As the target variable, we focus on effort. As pre-processing, we employ Euclidean distance d_E coupled with MinMax normalization and weighting, since this combination is determined to be the best for CIL in Section VI-F1. As for estimator model, we use first CART + Tree pruning [17], which is claimed to be the most efficient estimator by Phannachitta *et al.* [28], and then Random Forest, which is demonstrated empirically to perform better than CART + Tree pruning in Section VI-B.

Similar to Table 3, Table 9 presents the correlation between CIL and MMRE for the target variable of effort with the estimator model of CART + Tree pruning. However, unlike Table 3 the input data is pre-processed in Table 9. Since the absence/presence of pre-processing is the only difference, we can make a direct comparison between Tables 3 and 9 to assess the effect induced on CIL by pre-processing.

By examining these tables, we observe that R values are higher in Table 9 in most but not all cases. This indicates that the lack of a strong correlation between CIL and MMRE is partially due to the lack of pre-processing. Nevertheless, it can not be attributed solely to that. In addition, the positive values in Table 9 are quite small, and thus, CIL is very far

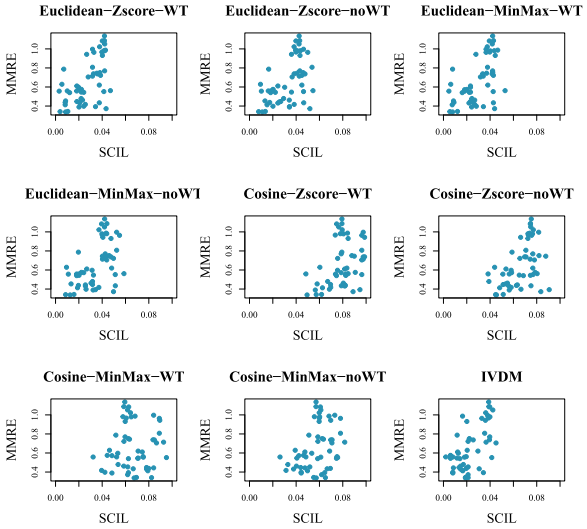


FIGURE 5. Scatter plot of SCIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.1 for all plots.

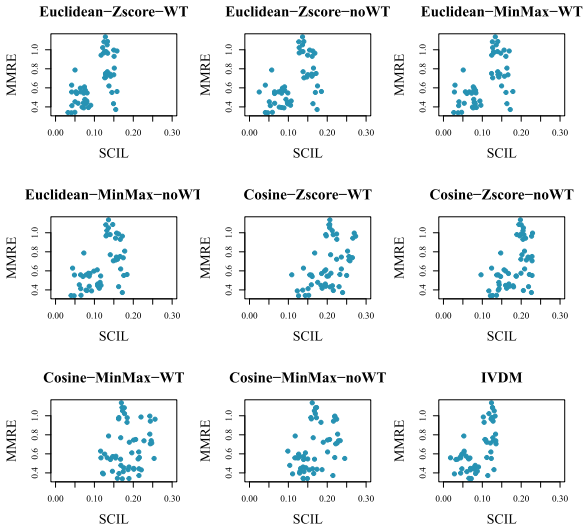


FIGURE 6. Scatter plot of SCIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.3 for all plots.

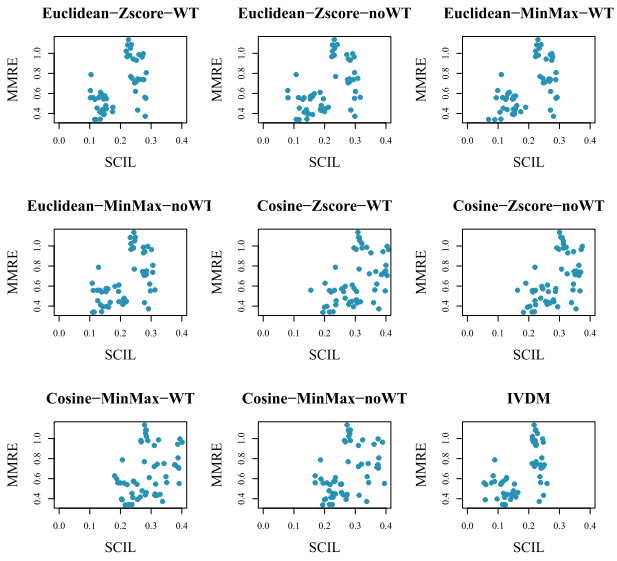


FIGURE 7. Scatter plot of SCIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.5 for all plots.

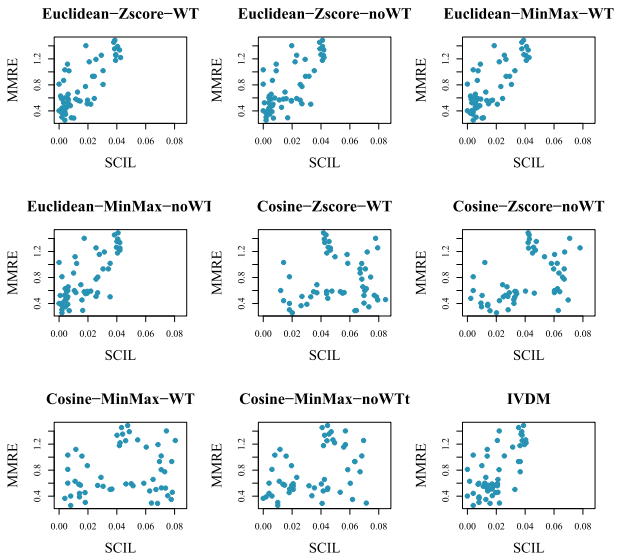


FIGURE 8. Scatter plot of SCIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.1 for all plots.

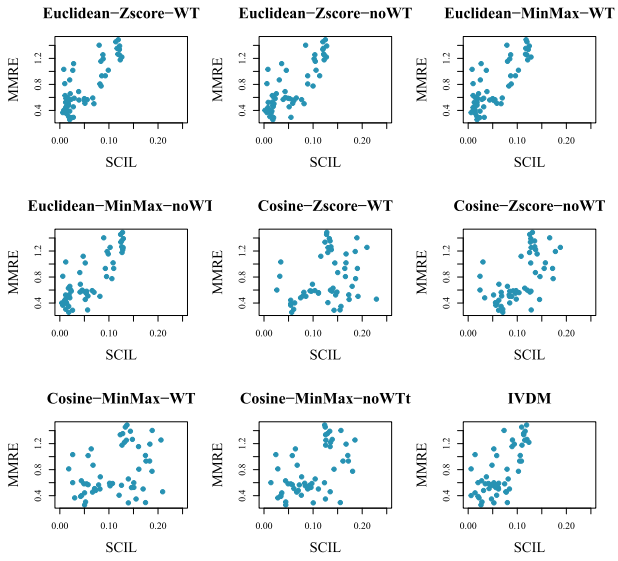


FIGURE 9. Scatter plot of SCIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.3 for all plots.

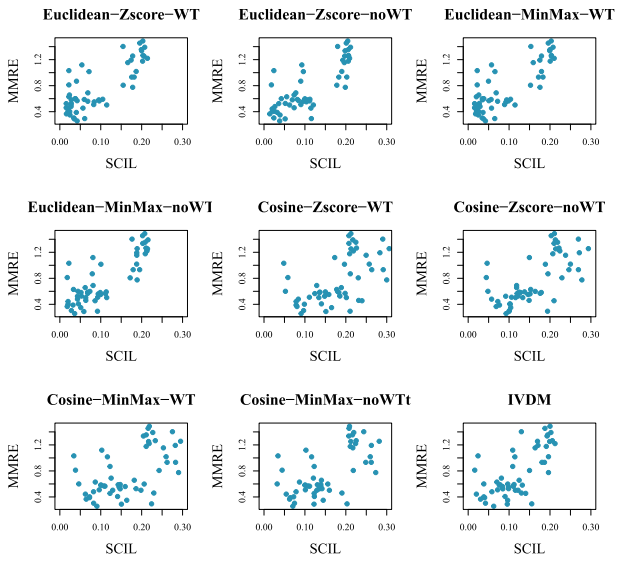


FIGURE 10. Scatter plot of SCIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.5 for all plots.

from satisfactory even with the most efficient pre-processing combination (among those addressed in this study).

Similar to Table 9, Table 10 presents the correlation between CIL and MMRE for the target variable of effort and with pre-processing. However, in Table 3 the estimator model is CART + tree pruning and in Table 9 it is Random Forest. Since the estimator model is the only difference, we can make a direct comparison between Tables 9 and 10 to assess the effect induced on CIL by improvement of the estimator model.

Note that in Table 10, we support CIL not only by applying the pre-processing operations but also by integrating it with a better estimator model (i.e. replacing CART + Tree pruning with Random Forest). In that respect, Table 10, gives an insight to the maximum improvement that we can expect on CIL by applying the best execution mode identified in this study.

However, comparing Tables 9 and 10, we see that improving the estimator does not necessarily lead to an increase in the correlation between CIL and MMRE. By comparing corresponding values in these tables, it is seen that there are more cases where R degrades than where it improves.

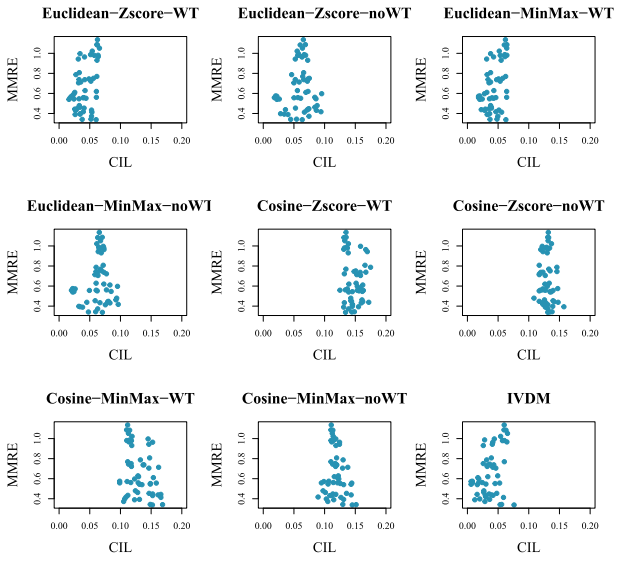


FIGURE 11. Scatter plot of CIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.1 for all plots.

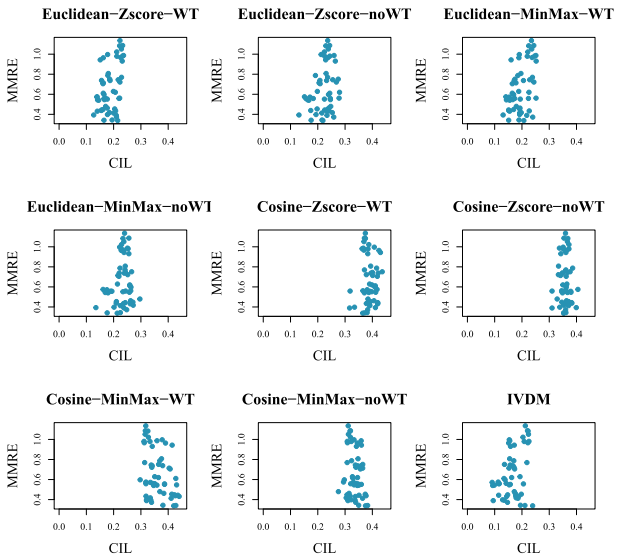


FIGURE 12. Scatter plot of CIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.3 for all plots.

Thus, we conclude that the low correlation between MMRE and CIL cannot be blamed on the poor performance of the estimation model either.

APPENDIX IV. RESULTS CONCERNING ALTERNATIVE THRESHOLD VALUES

Figures 5 ∼ 7 show scatter diagrams of SCIL and MMRE of effort estimation for different threshold α values (0.1, 0.3 and 0.5). Figures 8 ∼ 10 show scatter diagrams of SCIL and MMRE of productivity estimation for different threshold α values (0.1, 0.3 and 0.5).

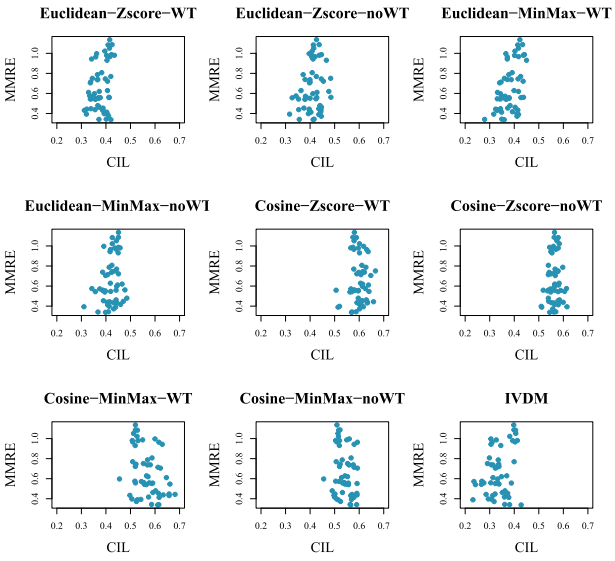


FIGURE 13. Scatter plot of CIL vs MMRE for the estimation target of effort concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.5 for all plots.

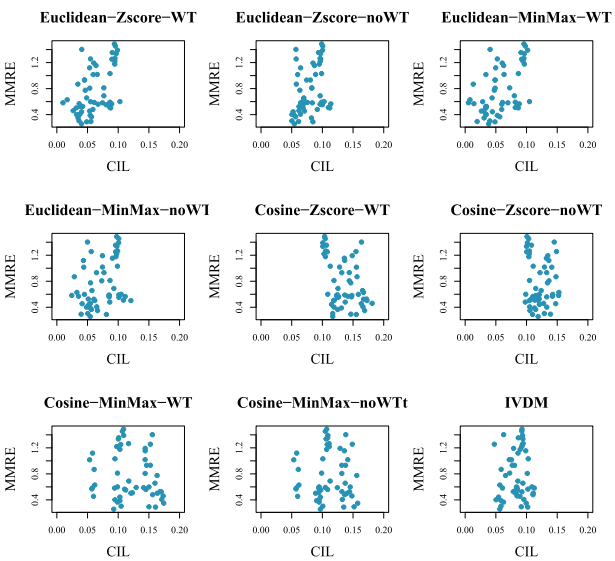


FIGURE 14. Scatter plot of CIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.1 for all plots.

Figures 11 ∼ 13 show scatter diagrams of CIL and MMRE of effort estimation for different threshold α values (0.1, 0.3 and 0.5). Figures 14 ∼ 16 show scatter diagrams of CIL and MMRE of productivity estimation for different threshold α values (0.1, 0.3 and 0.5).

In each diagram, the title shows the distance function used, normalization technique used, and weighting used or not (e.g. Cosine-MinMax-noWT means the cosine distance and MinMax normalization without weighting). Note that for IVDM distance function, we did not apply normalization and weighting because it already considers the relationship

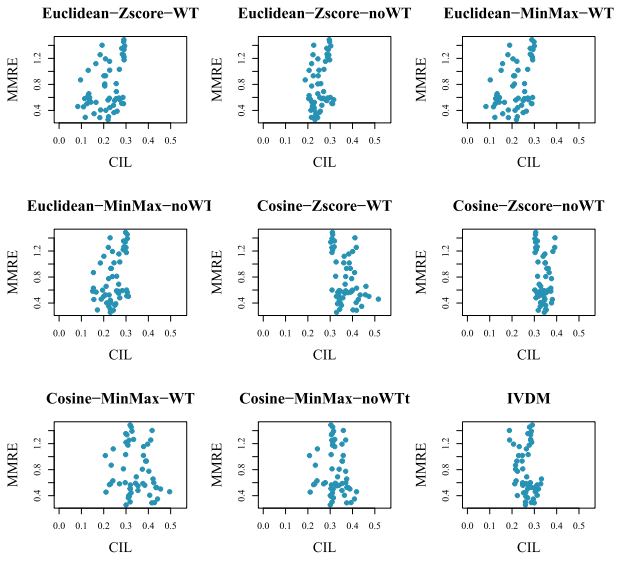


FIGURE 15. Scatter plot of CIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.3 for all plots.

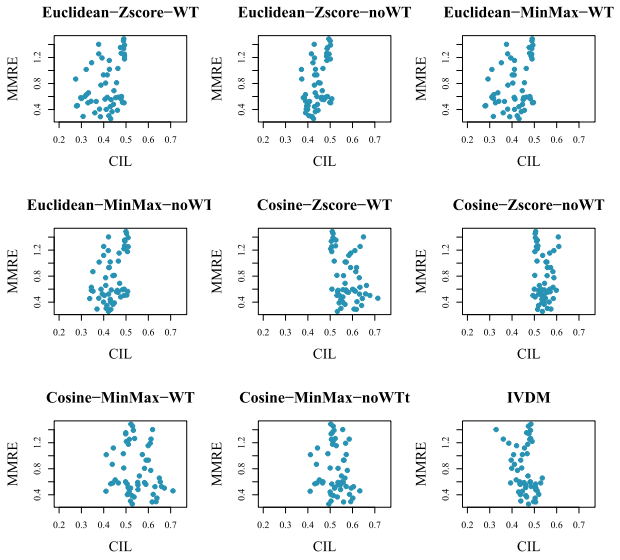


FIGURE 16. Scatter plot of CIL vs MMRE for the estimation target of productivity concerning Euclidean distance d_E, Cosine distance d_C and IVDM. Note that the threshold α is set to 0.5 for all plots.

between the target variable and feature variables in distance computation.

REFERENCES

[1] Z. Abdelali, H. Mustapha, and N. Abdelwahed, "Investigating the use of random forest in software effort estimation," *Proc. Comput. Sci.*, vol. 148, pp. 343–352, Jan. 2019.

[2] A. BaniMustafa, "Predicting software effort estimation using machine learning techniques," in *Proc. 8th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Jul. 2018, pp. 249–256.

[3] Y. Bao, N. Ishii, and X. Du, "Combining multiple K-nearest neighbor classifiers using difference distance functions," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn. (IDEAL)*, 2014, pp. 634–641.

[4] E. Blanzieri and F. Ricci, "Probability based metrics for nearest neighbor classification and case-based reasoning," in *Proc. Int. Conf. Case-Based Reasoning*, 1999, pp. 14–28.

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

[6] M. F. Bosu and S. G. MacDonell, "A taxonomy of data quality challenges in empirical software engineering," in *Proc. 22nd Austral. Softw. Eng. Conf.*, Jun. 2013, pp. 97–106.

[7] M. F. Bosu and S. G. Macdonell, "Experience: Quality benchmarking of datasets used in software effort estimation," *J. Data Inf. Qual.*, vol. 11, no. 4, pp. 1–38, Sep. 2019.

[8] M. L. Brown and J. F. Kros, "Data mining and the impact of missing data," *Ind. Manage. Data Syst.*, vol. 103, no. 8, pp. 611–621, Nov. 2003.

[9] J. S. Chen and C. H. Cheng, "Software diagnosis using fuzzified attribute base on modified MEPA," in *Proc. Int. Conf. Ind., Eng. Appl. Appl. Intell. Syst., Adv. Appl. Artif. Intell.*, 2006, pp. 1270–1279.

[10] J. M. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," M.S. thesis, Univ. Montreal, Montreal, QC, Canada, 1989.

[11] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Proc. Comput. Sci.*, vol. 161, pp. 466–474, Jan. 2019.

[12] S. Gupta, G. Sikka, and H. Verma, "Recent methods for software effort estimation by analogy," *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 4, pp. 1–5, Aug. 2011.

[13] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, "The misuse of the NASA metrics data program data sets for automated software defect prediction," in *Proc. 15th Annu. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2011, pp. 96–103.

[14] J. Keung, E. Kocaguneli, and T. Menzies, "Finding conclusion stability for selecting the best effort predictor in software effort estimation," *Automated Softw. Eng.*, vol. 20, no. 4, pp. 543–567, Dec. 2013.

[15] S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in *Proc. 33rd Int. Conf. Softw. Eng.*, May 2011, pp. 481–490.

[16] B. Kitchenham and E. Mendes, "Why comparative effort prediction studies may be invalid," in *Proc. 5th Int. Conf. Predictor Models Softw. Eng. (PROMISE)*, 2009, pp. 4:1–4:5.

[17] R. J. Lewis, "An introduction to classification and regression tree (CART) analysis," in *Proc. Annu. Meeting Soc. Academic Emergency Med.*, 2000, pp. 1–14.

[18] S. Liu, Z. Guo, Y. Li, C. Wang, L. Chen, Z. Sun, Y. Zhou, and B. Xu, "Inconsistent defect labels: Essence, causes, and influence," *IEEE Trans. Softw. Eng.*, early access, Mar. 7, 2022, doi: 10.1109/TSE.2022.3156787.

[19] G. A. Liebchen and M. Shepperd, "Software productivity analysis of a large data set and issues of confidentiality and data quality," in *Proc. 11th IEEE Int. Softw. Metrics Symp. (METRICS)*, Sep. 2005, pp. 46–48.

[20] B. Lauro and R. Traverso, "Data fitness for integration," Mimeo, New York, NY, USA, Tech. Rep., 2018.

[21] G. A. Liebchen, B. Twala, M. Shepperd, and M. Cartwright, "Assessing the quality and cleaning of a software project dataset: An experience report," in *Proc. Electron. Workshops Comput.*, Apr. 2006, pp. 122–128.

[22] G. A. Liebchen and M. Shepperd, "Data sets and data quality in software engineering," in *Proc. Int. Workshop Predictor Models Softw. Eng.*, 2008, pp. 39–44.

[23] K. D. Maxwell, *Applied Statistics for Software Managers*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[24] T. Menzies, R. Krishna, and D. Pryor. (2017). *The SeaCraft Repository of Empirical Software Engineering Data*. [Online]. Available: https://zenodo.org/communities/seacraft

[25] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust regression for developing software estimation models," *J. Syst. Softw.*, vol. 27, no. 1, pp. 3–16, Oct. 1994.

[26] F. Z. Mocnik, A. Zipf, and H. Fan, "Data quality and fitness for purpose," in *Proc. AGILE Conf. Geographic Inf. Sci.*, 2017, pp. 9–12.

[27] K. Ono, M. Tsunoda, A. Monden, and K. Matsumoto, "Influence of outliers on estimation accuracy of software development effort," *IEICE Trans. Inf. Syst.*, vol. 104, no. 1, pp. 91–105, 2021.

[28] P. Phannachitta, A. Monden, J. Keung, and K. Matsumoto, "Case consistency: A necessary data quality property for software engineering data sets," in *Proc. Int. Conf. Eval. Assessment Softw. Eng.*, Apr. 2015, pp. 1–10.

[29] M. W. Reynolds, A. Bourke, and N. A. Dreyer, "Considerations when evaluating real-world data quality in the context of fitness for purpose," *Pharmacoepidemiol. Drug Saf.*, vol. 29, no. 10, pp. 1316–1318, Oct. 2020.

[30] I. Stamelos, L. Angelis, P. Dimou, and E. Sakellaris, "On the use of Bayesian belief networks for the prediction of software productivity," *Inf. Softw. Technol.*, vol. 45, no. 1, pp. 51–60, Jan. 2003.
[31] K. Strike, K. E. Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Trans. Softw. Eng.*, vol. 27, no. 10, pp. 890–908, Oct. 2001.
[32] J. Van Hulse, T. M. Khoshgoftaar, C. Seiffert, and L. Zhao, "Noise correction using Bayesian multiple imputation," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Sep. 2006, pp. 478–483.
[33] A. G. P. Varshini, K. A. Kumari, D. Janani, and S. Soundariya, "Comparative analysis of machine learning and deep learning algorithms for software effort estimation," *J. Phys., Conf.*, vol. 1767, no. 1, Feb. 2021, Art. no. 012019.
[34] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012.
[35] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *J. Artif. Intell. Res.*, vol. 6, pp. 1–34, Jan. 1997.
[36] J. Wu and S. Gao, "Software productivity estimation by regression and Naive-Bayes classifier—An empirical research," in *Proc. Int. Conf. Promotion Inf. Technol.*, Aug. 2016, pp. 20–24.
[37] C. F. Kemerer, "An empirical validation of software cost estimation models," *Commun. ACM*, vol. 30, no. 5, pp. 416–429, May 1987.

MAOHUA GAN received the B.E. degree in software engineering from Northwestern Polytechnical University, in 2015, and the M.S. degree in information science from Okayama University, in 2020, where he is currently pursuing the Ph.D. degree with the Division of Industrial Innovation Sciences, Graduate School of Natural Science and Technology. His research interest includes software measurement and analytics.

ZEYNEP YÜCEL (Member, IEEE) received the B.S. degree from Boğaziçi University, Istanbul, Turkey, and the M.S. and Ph.D. degrees from Bilkent University, Ankara, Turkey, in 2005 and 2010, respectively, all in electrical engineering. She was a Postdoctoral Researcher at ATR Laboratories, Kyoto, Japan, for five years, before being awarded a JSPS Fellowship, in 2016. She is currently an Associate Professor with Okayama University, Japan. Her research interests include robotics, signal processing, computer vision, and pattern recognition.

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