An Empirical Analysis of Three-stage Data-Preprocessing for Analogy-based Software Effort Estimation on the ISBSG Data†

Jianglin Huang[1], Yan-Fu Li[2], Jacky Wai Keung[1], Y. T. Yu[1], and W. K. Chan[1]

[1]Department of Computer Science
City University of Hong Kong
Hong Kong
{jianhuang7, jacky.keung, csytyu, wkchan}@cityu.edu.hk

[2]Department of Industrial Engineering
Tsinghua University
Beijing, China
liyanfu@tsinghua.edu.cn

***Abstract*---Analogy-based software effort estimation is a method to estimate the project cost of an unseen project based on analogies against previous projects sharing selected features. The validity of the selected features depends on many factors, and one of most crucial factors is the effectiveness of the data-preprocessing techniques applied to the datasets of the previous projects. In this paper, we report the *first* controlled experiment that studies the class of three-stage data-preprocessing techniques with stages of missing data imputation, data normalization, and feature selection for analogy-based effort estimation. We conducted our investigation on the ISBSG data. The experimental results show that three-stage data-preprocessing techniques have significant impacts on the resultant effort estimation accuracy. The results also indicate that the combined use of Z-Score normalization, *k*NN imputation and mutual information based feature weighting can be an effective choice for analogy-based effort estimation.**

Keywords*---*analogy-based effort estimation; data-preprocessing; missing data imputation; data normalization; feature selection

ABBREVIATIONS

ABE	analogy-based estimation
DP	Data-preprocessing
DN	data normalization
FS	feature selection
GA	genetic algorithm
*k*NNI	*k* nearest neighbor imputation
ML	machine learning
MEI	mean/mode imputation
MDI	missing data imputation
MDT	missing data techniques
MI	mutual information
SA	standard accuracy
SBS	sequential backward selection
SEE	software effort estimation
SFS	sequential forward selection

† This work is supported in part by the General Research Fund of the Research Grants Council of Hong Kong (project numbers 125113, 11200015 and 11214116), and the research funds of City University of Hong Kong (project numbers 7004683 and 7004474).

I. INTRODUCTION

Estimating the software development effort is an essential task in the project management of any serious software development project [1]. Analogy-based estimation (ABE) [2] is a machine-learning (ML) method for predicting the most likely software development efforts of a given software project in the early stage of the project. The idea of ABE is to use the actual efforts of the selected analogies to estimate the effort of an unseen project with the assumption that software development projects with similar features require similar software development efforts [2]. Extrapolating the efforts of the selected analogies to a given project sharing same or similar features has the potential of providing a grounded estimate of the given project.

Data-preprocessing (DP), for example, feature selection and data normalization, is a fundamental stage of all ML methods in software effort estimation (SEE). Previous work [3] has shown that using different DP techniques can produce significantly different impacts on the accuracy of ABE techniques over the same projects with the references to the same corresponding sets of previous projects [4]. For examples, feature selection (FS) has shown to improve the accuracy of a wide spectrum of ABE techniques [3]; whereas, ABE techniques based on classification tree [5] or regression tree [6] are not significantly affected by data normalization as their DP techniques [7]. Many prior studies (e.g., [4]) have focused on studying individual DP techniques [7-9] because different DP techniques can handle different, specific data issues exhibiting in SEE. These issues include missingness, redundancy, and abnormality [10, 11]. However, to the best of our knowledge, only few studies investigate the effect of the ensembles over individual DP techniques on the accuracy of ABE [7]. Chen, et al. [8] and Liu, et al. [12] studied the effect of the use of feature selection followed by the case dimension reduction for software fault prediction. Huang, et al. [7] found that the overall effect of multi-stage DP on the accuracy of ABE techniques or SEE could not be adequately explained by each single DP technique. A holistic view on multi-stage DP is thus indispensable in understanding the use of DP techniques in chain rather than taking one or two DP techniques blindly.

978-1-5386-0592-9/17 $31.00 © 2017 IEEE
DOI 10.1109/QRS.2017.54

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

The present paper extends our previous work [7]. It extends the literature review in [7] and improves the experimental design in terms of the scope, dataset, and metrics. This study confines to the finding on the ISBSG data since we believe its cross-company characteristic could fairly help reflect the effect of multi-stage DP [7]. In particular, we investigate a kind of three-stage DP, which consists of data normalization, missing data imputation and feature selection in chain. After the data is processed via a three-stage DP, a ABE technique is applied to show the corresponding estimation accuracy. Compared to our previous work [7], the marginal effect of more DP techniques is also analysed. The ambiguity in terms of the multi-stage DP in our previous work [7] is further clarified. Apart from only using wrappers for feature selection in SEE, filters and optimization techniques are also frequently applied in recent analogy-based SEE (see Table I for more details). Our experiment includes MI based feature weighting scheme and genetic algorithm based feature selection. At the same time, our study excludes the investigation on case selection [7] and outlier deletion [10]. It is because applying an appropriate data cleansing process (Section III.A) has already deleted lower quality-rating cases for building SEE models [13], and the cleansed dataset could serve as a candidate dataset for subsequent analogy purpose, and different outlier detection methods often produce inconsistent results [10]. The experiment results show that the effectiveness of ABE could be significantly affected by an individual DP technique and the possible combinations. Multi-stage DP recommendation is also discussed in the end.

The main contribution of this work is as follows: To the best of our knowledge, this work is one of the pioneering work in studying recent and effective multi-stage DP techniques, including Z-Score normalization [14], sequential forward/backward feature selection [3], mutual information (MI) based feature weighting [15], and genetic algorithm based feature selection [6].

The rest of this paper is organized as follows. Section II presents the background of DP. Section III describes our experimental design. Section IV analyses the data and presents the experimental results and the data analyses. Section V discusses the threats to validity. Section VI concludes the work and presents the future work.

II. BACKGROUND

A. Data Preprocessing

ML requires the presence of data in a reasonable format, which is achieved through data-preprocessing (DP). DP techniques include reduction, projection, and missing data techniques (MDTs). Data reduction decreases the data size via, for example, feature selection (FS) or dimension reduction. Data projection transforms the appearance of the data to a different scale or dimension. For example, data normalization is a kind of data projection, which scales all the features to fit within a predefined range. MDT includes deleting instances with missing values [16] and/or replacing them with their estimates [17]. Table I summarizes 25 relevant representative works published from 2012 to 2016 and shows the types of DP techniques these works employed. We have found that techniques of *Missing Data Imputation* (MDI), *Data Normalization* (DN), and/or *Feature Selection* (FS) are commonly used in these works. Ten of these works ([18] [19] [20] [21] [22] [23] [24] [2] [25] and [26]) use at most one DP technique, and the rest use a combination of DP techniques.

To the best of our knowledge, there are very few previous works that investigate DP systematically. Huang, et al. [7] surveyed the use of DP techniques in the context of SEE. Their survey reviewed the literature for those DP techniques in four categories, including data normalization, MDT, FS, and case selection. Their finding shows that prior SEE studies often apply DP techniques having two stages, such as data normalization followed by feature selection (FS) [11, 27] or MDT followed by FS [28, 29], without a systematic study of the effect of the selected DP techniques on the resultant estimation effort.

Table I: DATA-PREPROCESSING TECHNIQUES IN ANALOGY-BASED SEE

Reference	FS[a]	DN[b]	MDT[c]
Kocaguneli *et al.* (2012) [2]	Filter		
Azzeh (2012) [30]	Wrapper	[0, 1]	LD
Borges and Menzies (2012) [27]	Dimension reduction	[0, 1]	
Ferrucci *et al.* (2012) [23]	Filters		
Kocaguneli *et al.* (2012) [31]	Filters		MDI
Kosti *et al.* (2012) [32]		[0, 1]	LD
Kocaguneli *et al.* (2013) [33]	Filters	[0, 1]	MDI
Menzies *et al.* (2013) [34]	Dimension reduction		
Mittas and Angelis (2013) [35]	Wrapper		LD
Bou *et al.* (2013) [36]	Filters		
Minku and Yao (2013) [37]	Filters	[0, 1]	MDI
Corazza *et al.* (2013) [29]	Filters		LD
Kocaguneli *et al.* (2013) [17]	Wrapper	[0, 1]	MDI
Seo and Bae (2013) [10]	Wrapper	[−1, 1]	LD
Khatibi Bardsiri *et al.* (2014) [38]	Optimization	[0, 1]	LD
Kocaguneli *et al.* (2014) [25]		[0, 1]	
Lee *et al.* (2014) [22]			MDI
Sigweni and Shepperd (2014) [24]	Optimization		
Azzeh *et al.* (2015) [21]		[0, 1]	
Mittas *et al.* (2015) [39]	Filters		LD
Di Martino *et al.* (2016) [20]		[0, 1]	
Idri *et al.* (2016) [40]		[0, 1]	MDI, LD
Phannachitta *et al.* (2016) [18]	Wrapper, Optimization		
Sarro *et al.* (2016) [26]			LD

[a] FS, feature selection, that is, what kinds of feature selection scheme are used, e.g., wrapper, filter, dimension reduction, or unknown
[b] DN, data normalization, that is, which data normalization method is used, [0, 1], [−1, 1], or unknown
[c] MDT: LD (listwise deletion), MDI (missing data imputation), or in blank (others)

However, their work [7] bears certain limitations. First, case selection, such as listwise deletion (LD), is increasingly unpopular to be used in SEE because it may damage the completeness of the data, rendering the application of the derived model less applicable. Second, the comparison between case-based LD and mean/mode imputation (MEI) is inappropriate because the set of dimensions of data after LD and that of MEI are not necessarily the same. Third, other commonly used DP techniques have not been included, such as Z-score normalization, *k*NN imputation and the search-based feature selection techniques. In our experience, many prior SEE studies had not reported their DP process in sufficient detail to allow replications [7].

B. Data Normalization

Data normalization (DN) transforms numeric values via a predefined rule so that all the features will have the same degree of influence. It is an indispensable step for most ML methods [41]. The typical targets of the transformation are the intervals of [0,1] or [−1,1].

$$x_k \text{ in } [0,1] = \frac{x_k - x_{\min}}{x_{\max} - x_{\min}},\ x_k \text{ in } [-1,1] = \frac{2x_k - x_{\min} - x_{\max}}{x_{\max} - x_{\min}} \quad (1).$$

Another common DN transformation is the Z-Score normalization:

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

$$Z-score\ x_k=\frac{x_k-\bar{x}}{std(x)} \quad (2),$$

in which x stands for a feature column in a data matrix, x_k is the k-th value of x, $x_{\min}$ and $x_{\max}$ are corresponding to the minimum and maximum values in x, and $\bar{x}$ and $std(x)$ are corresponding to the mean and standard deviation of x. To the best of our knowledge, the majority of prior studies prefer the use of the interval [0, 1] as the default DN (see Table I for example) instead of using Z-score. A notable exception is the work of Jing, et al. [42] which applied Z-Score instead of the interval [0, 1]. However, all the studies listed in Table I did not provide the underlying reasons on their preferences, and some of them did not mention DN in their work.

C. Missing Data Imputation

Mean/mode imputation (MEI) replaces each missing value with the mean (for numeric features) or mode (for categorical features) of observed values. However, it may lead to diminution of the variance of variables. As an alternative, *k*NN imputation (*k*NNI) has been specially studied in the SEE domain, but it is less commonly accepted than MEI.

D. Feature Selection Scheme

FS selects a feature subset that produces similar impacts on the estimation as if all the available features were selected. Most studies (such as [3]) support the use of FS when applying ML methods in SEE. Intelligent FS methods are summarized as wrappers or filters. In SEE, sequential forward selection (SFS) and sequential backward selection (SBS) [3] are some of the most commonly used wrappers. SFS is less computationally complex than SBS, but suffers from more severe information loss. Both can be used simultaneously.

Filters evaluate the preset criteria independently prior to the ML training phase. For example, the MI based feature weight indicates the highest relevant features by calculating the dependency, MI, among variables. MI could indicate the highest relevant features by calculating the dependency measure, the mutual information, among variables. Given two random variables X and Y, their corresponding MI I is quantified as $I(X;Y)=\sum_{x\in X}\sum_{y\in Y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$. For ease of presentation, MI based feature weighting is also treated as a strategy of FS. Besides, some prior work prefers search-based FS techniques such as genetic algorithm (GA). GA is a class of robust problem-solving techniques based on a population of solutions, which evolve through successive generations by means of the application of three genetic operators: selection, crossover, and mutation. GA is suitable to search for a solution in large search spaces, where other methods (local or gradient searches) may not provide satisfactory results. For instance, the clustering and feature selection problem that is encoded in a binary representation can be effectively solved by a GA method.

E. Research Questions

Based upon the above-mentioned introduction and background, this paper studies the following four research questions (RQs) in terms of the three DP techniques and their combination in the application of ABE. The research target is to verify the impact of the recent DP techniques and to determine effective multi-stage DP chains for ABE based on these recent DP techniques. The set of RQs to be answered is as follows:

RQ1 (DN): Does Z-Score outperform [0, 1] and [−1, 1]?

RQ2 (MDI): Is *k*NNI more preferable than MEI?

RQ3 (FS): Is FS more effective than ABE? Is GA or MI more effective than sequential feature selection?

RQ4 (three-stage DP): Is there any combination of data-preprocessing techniques effective to ABE?

It should be noted that all the four research questions are based on the special circumstance: analogy-based software effort estimation from the ISBSG data. The former studies have already shown that performing DP could be case-dependent and non-generalizable [7, 8]. For example, using either [0, 1] or [−1, 1] as the normalization choice would not make any difference on the tree models [7]. Owing to these reasons, our study on these recent DP techniques and their combinations adds new and important knowledge to the body of knowledge for SEE.

III. Experimental Design

A. ISBSG Repository

We used the ISBSG (International Software Benchmarking Standards Group) R10 repository for empirical tests, which contained 4106 projects. Owing to its heterogeneous nature and large size, ISBSG recommended extracting a suitable subset for any SEE practice [13]. We followed ISBSG's suggestion to select a subset with 14 features, including 7 numerical project size features (*input count, output count, enquiry count, file count, interface count, adjusted function point, and normalized work effort*) and the 7 categorical features listed in Table II [13]. Based on ANOVA, the original categories are merged into homogeneous groups [43].

Table II: Summary of the ISBSG Categorical Features

Feature Name	Description
Development Platform	1 = MF (Mainframe); 2 = MR (Mid-range); 3 = PC (Personal Computer); 4 = Multi
Development Type	1 = <Enhancement; Re-development>; 2 = New Development
Organization Type	1 = Banking; 2 = <Medical & Health Care; Public Administration; Government>; 3 = <Wholesale & Retail Trade; Financial, Property & Business Services>; 4 = <Manufacturing; Construction>; 5 = <Electricity, Gas, Water; Energy>; 6 = <Community Services; Service; Professional Services; IT Services; Communications; Computers & Software>; 7 = Aerospace/Automotive
Business Area Type	1 = Insurance; 2 = Banking; 3 = Financial; 4 = Accounting; 5 = Engineering; 6 = Personnel;
Application Type	1 = Transaction/Production System; 2 = Financial transaction process/accounting; 3 = Management Information System; 4 = Stock control & order processing; 5 = Billing; 6 = Management of License and Permits; 7 = Voice Provisioning; 8 = Document management; 9 = Electronic Data Interchange; 10 = Office Information System; Office Automation; 11 = Trading; 12 = E-Business; Internet Banking; 13 = Local
Primary Programming Language	1 = C; 2 = C++; 3 = COBOL; 4 = Java; 5 = NATURAL; 6 = ORACLE; 7 = SQL; 8 = Visual Basic; 9 = Access; 10 = Shell/Assembler

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

As suggested by Rodríguez et al. [44], the ISBSG repository data need to be properly prepared for effort estimation. Thus, we performed the following steps to remove the less significant instances: (1) select only the instances with rating in either A or B in *data quality* as well as the rating of A in *UFP* [16], (2) filter out the projects with normalized ratio larger than 1.2 [13], (3) select the projects with resource level '1', that is, only those with the effort of development team recorded (because our present study is only interested in the work effort of development teams) [10, 44], and finally, (4) select the projects sized using IFPUG V4/V4+ in *FP Standard All*. After the performing above four steps, 446 projects remained. Among all the features, '*NorEffort*' is the target. We noted from further statistical analysis that '*NorEffort*' in the ISBSG data were highly-skewed (7.1) with very large kurtosis (70.4), and if the missing values were excluded from the dataset, the skewness and kurtosis would be reduced.

B. Analogy-based Estimation

Generally, after data collection and cleaning, ABE requires the choice of the following three parameters:

- *Distance function*. This function computes the distances between the project x' being estimated and the i-th historical projects x_i. The Euclidean distance is $D(x_i,x') = \sqrt{\sum_{j=1}^{p} wDis(x_{ij},x'_j)}$ and the Manhattan distance is $D(x_i,x') = \sum_{j=1}^{p} wDis(x_{ij},x'_j)$. $Dis(x_{ij},x'_j)$ is defined as follows. We chose these two distance functions in our experiment.

$$Dis(x_{ij},x'_j) = \begin{cases} (x_{ij}-x'_j)^2, & \text{Euclidean distance, if feature } x_{ij} \text{ and } x'_j \text{ are numeric} \\ |x_{ij}-x'_j|, & \text{Manhattan distance, if feature } x_{ij} \text{ and } x'_j \text{ are numeric} \\ 0, & \text{if feature } x_{ij} \text{ and } x'_j \text{ are nominal and } x_{ij}=x'_j \\ 1, & \text{if feature } x_{ij} \text{ and } x'_j \text{ are nominal and } x_{ij}\neq x'_j \end{cases} \quad (3)$$

- *Value of k*. Bardsiri et al. [28] recommended locating the best k from 1 to 5. In our experiment, we chose $k = 3$ and $k = 5$.
- *Solution function*. We chose *mean* as the solution function, which was considered a reliable choice used in many ABE studies [38].

C. Performance Metrics

In our experiments, we adopted **standardized accuracy** (SA) as the measure for comparing different DP combinations. SA was proposed by Shepperd and MacDonell [45] as a superior metric for prediction model comparison than traditional relative error metrics. It was further modified by Langdon et al. [46] to be an absolute unbiased metric without random guessing.

SA is defined as $SA = \left(1 - \frac{MAR}{MAR_{p_0}}\right) \times 100\%$, where MAR is the mean absolute residual, which is computed as $MAR = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$, and MAR_{p_0} is the baseline MAR, which is computed as $MAR_{p_0} = \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{j<i} |y_i - y_j|$. It is noted that a larger value of SA indicates higher accuracy achieved by of a model. The ideal range of SA is 0 to 100 (%).

To the best of our knowledge, the present work is the *first* work that uses the exact mean absolute error of baseline predictor MAR_{p_0} in computing SA [46]. The mean squared error, $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, drives prediction via training and is used as to guide SFS/SBS and GA based feature selection on validating data.

We used a cross-validation scheme for parameter tuning. The dataset was split into three non-overlapping subsets: training, validating, and testing. The training and validating sets help select the best parameters. In our study, only the phase of feature selection needed cross-validation to search the optimal feature sets. The features that minimize *MSE* were selected for testing.

D. Experimental Setting

For each testing dataset, we generated 24 (2 MDI × 3 DN × 4 FS) combinations of DP techniques. Instances of each of the three kinds of techniques (FS, data normalization and MDI) presented in Section II were performed as the three stages in our experimental design to synthesize each DP technique chain. More specifically, we studied the following instances: For missingness, we included *k*NNI and MEI. For data normalization, we employed [0, 1], Z-Score, and [−1, 1]. For *k*NNI, our empirical analysis applies the incomplete-case based *k*NNI version proposed by van Hulse and Khoshgoftaar [47]. This version of *k*NNI uses both complete and incomplete cases to perform imputation, and it uses Euclidean distance and 5 neighbors as its *k*NN parameters. For FS, we adopted 3 FS strategies: SFS_SBS (combination of SFS and SBS), GA, MI together with a control strategy '*Null*', which did not perform any feature selection. For SFS_SBS, we only selected the features with the minimal prediction error in either SFS or SBS. For MI, our study uses *mRMR* [48]. The parameter of w, the feature weight, was defined as $w_i = \frac{I(f_i;f_{effort})}{\sum_{i=1}^{M} I(f_i;f_{effort})}$, where f_i was the ith feature. That is, no feature was excluded in MI based feature weighting unless the corresponding weight was zero. For GA, the weight parameter w in GA application was defined as 1 if the feature was selected, otherwise 0. Our experiment adopted the GA implementation (*ga* and *gaoptimseti*) in Matlab R2016a. The population size was 100 with 70 generations each time. Mutation followed the uniform distribution at the rate of 0.05.

Note that all the three DNs were applied to numeric features only. The effort feature (dependent variable) was normalized to [0, 1]. For parameter tuning, we used a cross-validation scheme. The entire dataset was randomly 10-1 split into two exclusive subsets using the hold-out method: training&validating and testing. The training&validating subset was used to train and validate the models to obtain the best feature subset. The validation was done for parameter tuning and preventing over-fitting. The testing subset was used to evaluate the estimation accuracy of ABE based on the metric SA. Only the sequential feature selection (SFS_SBS) and GA based feature selection needed further 10-fold cross-validation for searching the best features. MI-based feature weighting was also conducted on the training subset.

We included four versions of ABE in our experiment: (1) ABE with $k = 3$ and Euclidean distance (ABE3e), (2) ABE with $k = 3$ and

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

Manhattan distance (ABE3m), (3) ABE with k = 5 and Euclidean distance (ABE5e), and (4) ABE with k = 5 and Manhattan distance (ABE5m).

Figure 1 illustrates the experimental procedure. After MDI and normalization, the cross-validation scheme began. At each fold, FS was applied during training and validation, and then the parameters were tuned by validation. Finally, the ABE methods were tested on the testing set. After repeating 30 times, the testing outcomes across all data splits were aggregated to calculate SA.

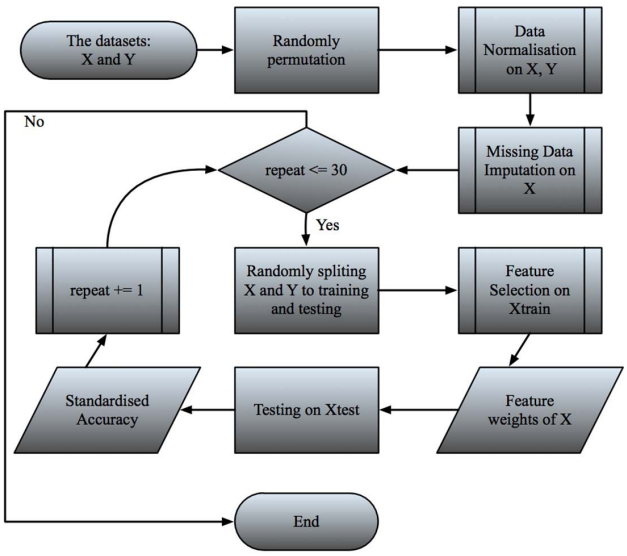

Fig 1: Experimental procedure.

IV. Data Analysis

The techniques used in three-stage DP were: FS (Null/SFS_SBS/MI/GA), data normalization (Z-Score/[0, 1]/[−1, 1]), and MDI (*k*NNI/MEI). First, the means and standard deviations of the SA were collected and analyzed. The mean values indicated the average accuracy, and the standard deviations of SA aimed to measure the robustness. Finally, the overall accuracy of ABE is presented.

A. Data Normalization Scheme

The marginal means, as well as the marginal standard deviations of the SA metric under each normalization scheme, are shown in Table III. From the table, the magnitude of SA is quite consistent across the four types of ABE methods. The data normalization schemes [0, 1] and [−1, 1] are less competitive than scheme Z-Score in terms of SA. The robustness of scheme Z-Score is worse if using 3 nearest neighbors in ABE. The SA values for the two data normalization schemes [0, 1] and [−1, 1] are similar. The difference is small, but [0, 1] appears to be slightly more robust. In the previous study [7], both schemes [0, 1] and [−1, 1] were generally beneficial to analogy-based SEE, compared to using no data normalization. Scheme [0, 1] slightly outperforms scheme [−1, 1]. For ABE with 5 nearest neighbors, scheme Z-Score could improve the overall estimation accuracy. The above discussion provides answer to RQ1.

Table III: Average Accuracy and Robustness of Different DN Schemes in terms of SA (%)

ABE	[0, 1]		Z-Score		[−1, 1]	
	Mean	Std	Mean	Std	Mean	Std
ABE3m	26.7	23.3	30.0	31.7	27.1	25.6
ABE5m	28.9	20.3	36.9	17.4	30.1	19.8
ABE3e	25.1	23.5	30.1	31.0	23.9	27.3
ABE5e	28.7	17.9	37.3	16.9	28.1	21.1

B. Missing Data Imputation Scheme

The marginal means and the standard deviations of the testing errors in terms of SA when using the two MDIs are shown in Table IV. From the table, MEI usually produces larger SA values and smaller standard deviations. The results indicate that MEI may be more effective than *k*NNI, the latter being only superior in accuracy when using ABE3m.

Table IV: Average Accuracy and Robustness of Different MDI Schemes in terms of SA (%)

ABE	*k*NNI		MEI	
	Mean	Std	Mean	Std
ABE3m	28.2	29.1	27.7	25.0
ABE5m	31.7	20.3	32.3	18.6
ABE3e	25.7	29.8	27.0	25.0
ABE5e	30.7	19.8	32.1	18.4

The issue of missing values in the ISBSG repository is severe. Our data analysis shows that *k*NNI may not provide a robust estimation due to missingness. *k*NNI is only slightly better under ABE3m (SA 28.2>SA 27.7) and slightly worse than MEI under each of ABE5m, ABE3e and ABE5e.

We can answer RQ2 that *k*NNI may not be a generally preferable option to MEI. The marginal effect in terms of missing data imputation shows that MEI is slightly better than *k*NNI. However, the relatively larger standard deviation of *k*NNI implies that the accuracy of *k*NNI could be better than that of MEI under certain circumstances. The impact of MDI is also influenced by the characteristics of the ISBSG data: high skewness and large kurtosis. To further explore the impact of MDI, ranking the data-preprocessing combinations for each ABE version is necessary.

C. Feature Selection Scheme

Table V shows the marginal means and standard deviations of SA on the testing subsets classified by Null/SFS_SBS/MI/GA. The result surprisingly shows that FS could not improve the accuracy of ABE methods. In addition, the combination of SFS_SBS is the least accurate and least robust. This finding shows that FS for ABE does not consistently improve the accuracy of effort estimation (especially when using sequential feature selection). Rather, using MI for calculating feature weights or using GA for subset selection could be potential candidates when appropriate parameters could be chosen.

The discussion above provides answers to RQ3. Scheme FS may not always be beneficial to ABE. Schemes GA and MI based FS could be superior to FS. We further analyzed the individual DP combinations

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

to examine how each different FS scheme performs. The 'Null' FS scheme and MI-based feature weighting scheme kept all the features (i.e., no features were excluded). Sequential feature selection, however, mostly selected 3-7 features from the ISBSG data, which ignored most of the data, while GA generally selected most of the features from the data. Therefore, the standard deviations of SA in SFS_SBS were found to be much larger than these of MI and GA based FS.

Table V: AVERAGE ACCURACY AND ROBUSTNESS OF DIFFERENT FS SCHEMES IN TERMS OF SA (%)

ABE	Null		SFS_SBS		MI		GA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ABE3m	35.1	17.0	10.9	37.3	34.2	19.1	31.6	22.7
ABE5m	38.0	13.4	17.0	24.7	36.9	13.7	36.1	15.7
ABE3e	33.0	18.1	9.9	38.9	30.1	21.6	32.4	19.3
ABE5e	36.6	14.3	18.2	20.6	34.4	18.2	36.4	16.6

D. Ranking of the Data Preprocessing Combinations

We draw the boxplots of SAs of the four ABE versions in Fig 2 to show the overall accuracy of each version. The accuracy of ABE5m is observed to be slightly higher than the others, and the ABE versions with $k = 5$ are more robust than those with $k = 3$.

We also ranked the data-preprocessing combinations according to the SA they achieved. We have 24 sets of data for SA. We only keep the top-5 ranked results, that is, the 5 best data-preprocessing combinations and their SA values, as shown in Table VI. All the repeated data-preprocessing combinations among ABE3e, ABE3m, ABE5e and ABE5m are highlighted in bold. Each single DP is coded with a corresponding number (shown in the footnote of Table VI). For example, the data-preprocessing combination of 2-1-3 (Z-Score, *k*NNI, with MI for FS) is highlighted in bold because it appears in all of ABE3e, ABE3m, ABE5e and ABE5m. One-tailed Wilcoxon signed-rank test (using significance levels $\alpha = 0.05$) is chosen to test if the best data-preprocessing combination is significantly different from the worst (with the smallest SA value in the top-5). The statistical tests results are marked with * in the first column of Table VI. For example, under ABE3e, the combination of 2-1-3 is significantly better than the worst combination of 2-2-1, and the two combinations of 2-1-1 and 2-2-4 are significantly better than the combination 2-2-1 as well.

Based on the findings in Table VI, we can see that the combinations 2-1-1 (Z-Score normalization, *k*NN imputation and Null FS) and 2-1-3 (Z-Score normalization, *k*NN imputation and MI-based feature weighting) are relatively robust choices. The accuracy of DP combinations with different ABE parameter settings may vary. Further observations from Table VI provides more useful knowledge of implementation of DP combinations. Combinations 2-1-1 and 2-1-3 are recommended for ABE. Although the marginal effect of *k*NNI is not better than MEI, the data-preprocessing ranking in Table VI presents a different story. The benefit of an individual DP technique to SEE could be presented via appropriate combinations. MEI may be a safe choice for ABE in general because its marginal effect outperforms *k*NNI. We note that the potential of *k*NNI is motivated only with other appropriate DPs, for example Z-score normalization and Null/MI-based FS in the present study. The above discussion gives us the answer to RQ4 in terms of selection of DP techniques: Most of the best DP technique combinations contain Z-Score as the DN scheme.

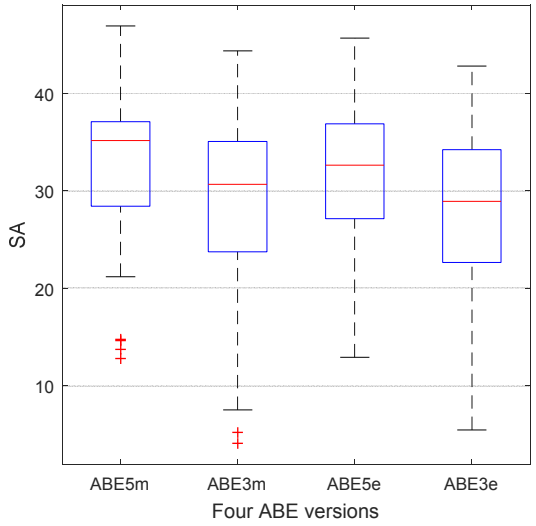

Fig 2: Boxplots of SA values

Table VI: SUMMARY OF RECOMMENDED DATA-PREPROCESSING COMBINATIONS

Mean SA	DN	MDI	FS	Mean SA	DN	MDI	FS
	ABE3e				ABE3m		
*42.8	**2**	**1**	**3**	*44.3	**2**	**1**	**3**
*40.1	**2**	**1**	**1**	*44.2	**2**	**1**	**1**
*36.7	2	2	4	*39.1	**2**	**2**	**1**
36.4	**2**	**2**	**3**	38.3	**2**	**2**	**3**
34.8	**2**	**2**	**1**	37.1	1	1	4
	ABE5e				**ABE5m**		
*45.7	**2**	**1**	**3**	*46.9	**2**	**1**	**1**
*45.2	**2**	**1**	**1**	*45.7	**2**	**2**	**1**
*43.8	**2**	**2**	**1**	*43.7	**2**	**1**	**3**
*41.9	**2**	**2**	**3**	39.0	**2**	**2**	**3**
39.1	2	2	4	37.6	1	1	4

DN, 1 – [0, 1]; 2 – Z-Score; 3 – [−1, 1]
MDI, 1 - kNNI; 2 – MEI
FS, 1 – Null; 2 – SFS_SBS, 3 – MI; 4 – GA
* The mean SA value in a cell is marked with an asterisk if the DP combination for the corresponding row is significantly better than that for the bottom row in terms of SA at the significance level of 0.05 using the one-tailed Wilcoxon signed-rank test

Both the marginal results on FS scheme and the ranking of DP combinations imply that excluding too many features from the ISBSG data may worsen the accuracy of ABE. Indeed, the collected features from the ISBSG data have already been recommended for SEE by the ISBSG organization. If the sequential feature selection leaves the data rather incomplete, it may lead to model under fitting. Moreover, another factor that may impact on the FS scheme is MDI. Since MDI is applied before FS in a DP chain, the imputed values certainly determine the relevance of the corresponding features. The performance between two different MDI schemes in Table V is quite close. Even though MEI is slightly more accurate than *k*NNI, but all the more accurate DP combinations in Table VII is associating with *k*NNI. Moreover, MDI is also related to the statistical characteristics of the ISBSG data. After using MEI, the mutual information between the feature with much missingness and the effort value could be weakened. It may be the reason that the more accurate DP combinations use MI for FS when MDI is *k*NNI. It appears to us that there is a significant interaction between individual DP techniques when they are applying together.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

V. THREATS TO VALIDITY

This section describes the threats to validity in terms of internal, construct and external aspects.

Conclusion validity is about the capacity to make significant correct conclusions. The Wilcoxon signed-rank test is used, which shows statistical significance. In the meantime, the large-sized ISBSG dataset mitigates the threats in terms of the number of observations.

As the work concentrates on the empirical investigation about the data-preprocessing impacts, the internal validity is analyzed on the experiment design. Two pre-defined parameters (distance measure and value of *k*) were considered in ABE in the study. Both of the adopted parameters have been widely used in the literature [2]. Moreover, the study uses SA as the model evaluation metric and MSE as the feature selection guidance, both of which are balanced and unbiased performance measures.

Construct validity questions the model performance assessment. Construct validity in this study is closely connected to the experimental design. To the best of our knowledge, SA is the most well-recognized performance measure in the SEE domain.

External validity represents the possibility of generalizing the research findings. Firstly, the data used in the study is collected from many different organizations. The data cleansing process is based upon clearly defined selection criteria. But we only used the data that comes from ISBSG. The data source is limited. Moreover, only a few DP techniques and estimation methods were studied in this experiment, which cause the difficulties to generalize our conclusions to other circumstances. Many cutting-edge missing data imputation approaches have not been considered in the experiment. While the results are promising, we cannot claim external validity.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have investigated three-stage data preprocessing techniques in the context of ABE. We have found that the effectiveness of ABE could be significantly affected by an individual data-preprocessing technique and their possible combinations. Compared with *k*NN imputation, we have found that applying mean/mode imputation (MEI) seems to be a potential candidate to be used if the datasets incur a severe problem of missingness. MEI is a safe choice to be used, but an appropriate use of *k*NN imputation may provide better data imputation. We have also found that the normalization scheme of Z-Score could be better than the scheme [0, 1] or [−1, 1]. The best data-preprocessing combination in our study for ABE on the ISBSG data is the combination of Z-Score data normalization, *k*NN imputation and mutual information based feature weighting. The second best one is the combination of Z-Score data normalization and *k*NN imputation without feature selection. In the experiment, when the second recommended data-preprocessing combination, analogy-based effort estimation with Manhattan distance measure with $k = 5$ produces the most accurate result.

As for future work, 1) experiments on more datasets will be considered to generalize the findings; 2) more advanced estimation methods can be involved; 3) more advanced missing data imputation approaches could be adopted to deal with missing data; 4) finding the strategy of selecting data-preprocessing techniques based on the characteristics of the data instead of picking the best data-preprocessing combination for certain ML method.

ACKNOWLEDGEMENT

We are grateful to the enormous support and encouragement from Hongyi Sun, Min Xie, and Federica Sarro.

REFERENCES

[1] R. D. A. Araújo, A. L. I. Oliveira, and S. Soares, "A shift-invariant morphological system for software development cost estimation," *Expert Systems with Applications,* vol. 38, pp. 4162-4168, 2011.

[2] E. Kocaguneli, S. Member, and T. Menzies, "Exploiting the essential assumptions of analogy-based effort estimation," *IEEE Transactions on Software Engineering,* vol. 38, no. 2, pp. 425-439, 2012.

[3] J. W. Keung, B. A. Kitchenham, and D. R. Jeffery, "Analogy-X: Providing statistical inference to analogy-based software cost estimation," *IEEE Transactions on Software Engineering,* vol. 34, no. 4, pp. 471-484, 2008.

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

[5] N. H. Chiu and S. J. Huang, "The adjusted analogy-based software effort estimation based on similarity distances," *Journal of Systems and Software,* vol. 80, no. 4, pp. 628-640, 2007.

[6] S. J. Huang and N. H. Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation," *Information and Software Technology,* vol. 48, no. 11, pp. 1034-1045, 2006.

[7] J. Huang, Y. F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and Software Technology,* vol. 67, pp. 108-127, 2015.

[8] J. Chen, S. Liu, W. Liu, X. Chen, Q. Gu, and D. Chen, "A two-stage data preprocessing approach for software fault prediction," in *the 8th International Conference on Software Security and Reliability (SERE'14)*, San Francisco, CA, USA, 2014, pp. 20-29.

[9] V. K. Bardsiri, D. N. Abang Jawawi, and E. Khatibi, "Towards improvement of analogy-based software development effort estimation: A review," *International Journal of Software Engineering and Knowledge Engineering,* vol. 24, no. 07, pp. 1065-1089, 2014.

[10] Y. S. Seo and D. H. Bae, "On the value of outlier elimination on software effort estimation research," *Empirical Software Engineering,* vol. 18, no. 4, pp. 659-698, 2013.

[11] M. Tsunoda, T. Kakimoto, A. Monden, and K. I. Matsumoto, "An empirical evaluation of outlier deletion methods for analogy-based cost estimation," in *the 7th International Conference on Predictive Models in Software Engineering (PROMISE '11)*, Banff, Canada, 2011, pp. 1-10.

[12] W. Liu, S. Liu, Q. Gu, J. Chen, X. Chen, and D. Chen, "Empirical studies of a two-stage data preprocessing approach for software fault prediction," *IEEE Transactions on Reliability,* vol. 65, no. 1, pp. 38-53, 2016.

[13] ISBSG, "The ISBSG Development & Enhancement project data ", ISBSG, Ed., R 10 ed, 2007.

[14] K. Strike, K. E. Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Transactions on Software Engineering,* vol. 27, no. 10, pp. 890-908, 2001.

[15] Y. F. Li, M. Xie, and T. N. Goh, "A study of mutual information based feature selection for case based reasoning in software cost estimation," *Expert Systems with Applications,* vol. 36, no. 3, pp. 5921-5931, 2009.

[16] N. Mittas and L. Angelis, "LSEbA: least squares regression and estimation by analogy in a semi-parametric model for software cost estimation," *Empirical Software Engineering,* vol. 15, no. 5, pp. 523-555, 2010.

[17] E. Kocaguneli, T. Menzies, and J. W. Keung, "Kernel methods for software effort estimation: Effects of different kernel functions and

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.

bandwidths on estimation accuracy," *Empirical Software Engineering,* vol. 18, no. 1, pp. 1-24, 2013.

[18] 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,* pp. 1-31, 2016.

[19] N. Ramasubbu and R. K. Balan, "Overcoming the challenges in cost estimation for distributed software projects," in *the 34th International Conference on Software Engineering (ICSE'12)*, Zurich, Switzerland, 2012, pp. 91-101.

[20] S. Di Martino, F. Ferrucci, C. Gravino, and F. Sarro, "Web Effort Estimation: Function Point Analysis vs. COSMIC," *Information and Software Technology,* vol. 72, pp. 90-109, 2016.

[21] M. Azzeh, A. B. Nassif, and L. L. Minku, "An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation," *Journal of Systems and Software,* vol. 103, pp. 36-52, 2015.

[22] T. Lee, T. Gu, and J. Baik, "MND-SCEMP: An empirical study of a software cost estimation modeling process in the defense domain," *Empirical Software Engineering,* vol. 19, no. 1, pp. 213-240, 2014.

[23] F. Ferrucci, E. Mendes, and F. Sarro, "Web effort estimation: the value of cross-company data set compared to single-company data set," in *the 8th International Conference on Predictive Models in Software Engineering (PROMISE'12)*, Lund, Sweden, 2012, pp. 29-38, 2365330.

[24] B. Sigweni and M. Shepperd, "Feature weighting techniques for CBR in software effort estimation studies: a review and empirical evaluation," in *the 10th International Conference on Predictive Models in Software Engineering (PROMISE'14)*, Torino, Italy, 2014, pp. 32-41: ACM.

[25] E. Kocaguneli, T. Menzies, and E. Mendes, "Transfer learning in effort estimation," *Empirical Software Engineering,* pp. 1-31, 2014.

[26] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," in *the 38th International Conference on Software Engineering (ICSE'16)*, Austin, TX, USA, 2016, pp. 619-630: ACM.

[27] R. Borges and T. Menzies, "Learning to change projects," in *the 8th International Conference on Predictive Models in Software Engineering (PROMISE'12)*, Lund, Sweden, 2012, pp. 11-18.

[28] V. Khatibi Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "A PSO-based model to increase the accuracy of software development effort estimation," *Software Quality Journal,* vol. 21, pp. 501-526, 2013.

[29] A. Corazza, S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, and E. Mendes, "Using tabu search to configure support vector regression for effort estimation," *Empirical Software Engineering,* vol. 18, no. 3, pp. 506-546, 2013.

[30] 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.

[31] E. Kocaguneli, T. Menzies, J. Hihn, and B. H. Kang, "Size doesn't matter? On the value of software size features for effort estimation," in *the 8th International Conference on Predictive Models in Software Engineering (PROMISE'12)*, Lund, Sweden, 2012, pp. 89-98.

[32] M. V. Kosti, N. Mittas, and L. Angelis, "Alternative methods using similarities in software effort estimation," in *the 8th International Conference on Predictive Models in Software Engineering (PROMISE'12)*, Lund, Sweden, 2012, pp. 59-68.

[33] E. Kocaguneli, T. Menzies, J. Keung, D. Cok, and R. Madachy, "Active learning and effort estimation: Finding the essential content of software effort estimation data," *IEEE Transactions on Software Engineering,* vol. 39, no. 8, pp. 1040-1053, 2013.

[34] T. Menzies *et al.*, "Local versus global lessons for defect prediction and effort estimation," *IEEE Transactions on Software Engineering,* vol. 39, no. 6, pp. 822-834, 2013.

[35] N. Mittas and L. Angelis, "Ranking and clustering software cost estimation models through a multiple comparisons algorithm," *IEEE Transactions on Software Engineering,* vol. 39, no. 4, pp. 537-551, 2013.

[36] A. Bou, D. Ho, and L. Fernando, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *Journal of Systems and Software,* vol. 86, no. 1, pp. 144-160, 2013.

[37] L. L. Minku and X. Yao, "Ensembles and locality: Insight on improving software effort estimation," *Information and Software Technology,* vol. 55, no. 8, pp. 1512-1528, 2013.

[38] V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, and E. Khatibi, "A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons," *Empirical Software Engineering,* vol. 19, no. 4, pp. 857-884, 2014.

[39] N. Mittas, E. Papatheocharous, L. Angelis, and A. S. Andreou, "Integrating non-parametric models with linear components for producing software cost estimations," *Journal of Systems and Software,* vol. 99, pp. 120-134, 2015.

[40] A. Idri, I. Abnane, and A. Abran, "Missing data techniques in analogy-based software development effort estimation," *Journal of Systems and Software,* vol. 117, pp. 595-611, 2016.

[41] S. García, J. Luengo, and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Systems,* vol. 98, pp. 1-29, 2016.

[42] X. Jing, F. Wu, X. Dong, F. Qi, and B. Xu, "Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning," in *the 10th Joint Meeting on Foundations of Software Engineering (FSE'15)*, Bergamo, Italy, 2015, pp. 496-507: ACM.

[43] P. Sentas and L. Angelis, "Categorical missing data imputation for software cost estimation by multinomial logistic regression," *Journal of Systems and Software,* vol. 79, no. 3, pp. 404-414, 2006.

[44] D. Rodríguez, M. A. Sicilia, E. García, and R. Harrison, "Empirical findings on team size and productivity in software development," *Journal of Systems and Software,* vol. 85, no. 3, pp. 562-570, 2012.

[45] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology,* vol. 54, pp. 820-827, 2012.

[46] W. B. Langdon, J. Dolado, F. Sarro, and M. Harman, "Exact mean absolute error of baseline predictor, MARP0," *Information and Software Technology,* vol. 73, pp. 16-18, 2016.

[47] J. Van Hulse and T. M. Khoshgoftaar, "Incomplete-case nearest neighbor imputation in software measurement data," *Information Sciences,* vol. 259, pp. 596-610, 2014.

[48] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 8, pp. 1226-1238, 2005.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on September 17,2026 at 02:31:47 UTC from IEEE Xplore. Restrictions apply.