RESEARCH ARTICLE

Improving Effort Estimation Accuracy in Software Development Projects Using Multiple Imputation Techniques for Missing Data Handling

Shahida Hayat[1], **Wajahat Akbar**[2], **Tariq Hussain**[3,4,*], **Muhammad Inam Ul Haq**[5], **Altaf Hussian**[5], **Irshad Khalil**[6,7], **Muhammad Nawaz Khan**[8] **and Samsonova Diana**[9,*]

[1] Department of Computer Science, University of Peshawar, Pakistan
[2] School of Electronic and Control Engineering, Chang'an University, Xián 710064, China
[3] School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou 310018, China
[4] School of Mathematics and Statistics, Zhejiang Gongshang University, Hangzhou 310018, China
[5] Department of Computer Science and Bioinformatics, Khushal Khan Khattak University Karak, Pakistan
[6] Department of Health Science and Technology, Gachon University, Incheon 21936, Republic of Korea
[7] Gachon Advanced Institute for Health Sciences and Technology, Gachon University, Incheon 21936, Republic of Korea
[8] Department of Computer Science and Information Technology, University of Malakand, Chakdara, Pakistan
[9] School of International Education, Zhejiang Gongshang University, Hangzhou 310018, China

Abstract

The challenge of accurately estimating effort for software development projects is critical for project managers (PM) and researchers. A common issue they encounter is missing data values in datasets, which complicates effort estimation (EE). While several models have been introduced to address this issue, none have proven entirely effective. The Analogy-Based Effort Estimation (ABEE) model is the most widely used approach, relying on historical data for estimation. However, the common practice of deleting cases or cells with missing observations results in a reduction of statistical power and negatively impacts the performance of ABEE, leading to inefficiencies and biases. This study employs the Multiple Imputation (MI) technique to address missing data by filling in incomplete cases. A comparison is conducted between the original and imputed ISBSG datasets for both small- and large-scale projects, using other imputation techniques to identify the most effective method for ABEE. The results demonstrate that the MI technique enhances effort estimation, providing more accurate and efficient outcomes while preserving valuable information throughout the project estimation process.

Keywords: analogy-based effort estimation, multiple imputation, software development effort estimation.

Academic Editor:
Jing Na

Submitted: 01 October 2024
Accepted: 08 November 2024
Published: 12 November 2024

Vol. 1, **No.** 3, 2024.
10.62762/TIS.2024.751418

***Corresponding authors:**
Tariq Hussain
uom.tariq@gmail.com
Samsonova Diana
dina.samsonova.2000@bk.ru

Citation
Hayat, S., Akbar, W., Hussain, T., Haq, M. I. U., Hussian, A., Khalil, I., Khan, M. M., & Diana, S. (2024). Improving Effort Estimation Accuracy in Software Development Projects Using Multiple Imputation Techniques for Missing Data Handling. *ICCK Transactions on Intelligent Systematics*, 1(3), 190–202.

1 Introduction

Software effort estimation in software development projects is not just a technical necessity but a cornerstone of successful project management and overall software quality. The significance of this topic is underscored by the multifaceted challenges that project managers (PMs) encounter when predicting the resources required for software projects, which are typically quantified in terms of calendar months and man-hours. Accurate effort estimations are essential for planning, resource allocation, budgeting, and ultimately ensuring that projects meet their deadlines and stay within budget constraints. Inaccurate estimations can have far-reaching consequences, including project delays, spiraling costs, and dissatisfaction among stakeholders and end-users. Consequently, addressing the complexities of effort estimation is vital not only for the success of individual projects but also for the long-term sustainability and competitiveness of the software development industry as a whole [1–3].

Despite the importance of accurate effort estimation, numerous methods and techniques have been proposed to enhance this process, yet many existing approaches have failed to achieve definitive success in practice. Research in this area has produced a variety of models, including parametric methods, which rely on mathematical relationships between variables; non-parametric techniques, which do not assume a specific distribution; and statistical approaches that incorporate historical data [4–6]. However, significant challenges remain in achieving precise and reliable project effort estimations. These challenges are compounded by the dynamic and complex nature of software development, where variables can change rapidly and unpredictably.

Among the various estimation techniques, analogy-based estimation approaches, such as Analogy-Based Effort Estimation (ABEE), have gained considerable popularity due to their inherent simplicity and effectiveness in utilizing historical project data to inform current estimations [7–9]. The ABEE approach allows PMs to leverage past experiences, drawing parallels between similar projects to predict the effort required for new undertakings. However, despite the many advantages of ABEE, limitations exist—particularly regarding its ability to handle missing data (MD), which is a pervasive issue in software project datasets such as the International Software Benchmarking Standards Group (ISBSG) [10, 11]. Missing data can arise from various sources, including incomplete records, human error, and inconsistencies in data collection practices.

The reliance on potentially irrelevant or insufficient historical data can significantly hinder the effectiveness of effort estimation models, leading to ineffective and inaccurate estimates. For instance, when projects are compared based on historical data that lacks critical variables or contains irrelevant information, the resulting estimations may fail to reflect the actual effort required [4, 12]. This, in turn, complicates the effort estimation process and may result in increased project risks, missed deadlines, and heightened costs. The impact of these inaccuracies extends beyond the immediate project, potentially affecting the reputation of the development team and the organization as a whole.

In light of these challenges, it is crucial to explore further and refine the ABEE approach to improve its robustness in the face of missing data [13]. Understanding how to effectively manage missing values and enhance the applicability and efficiency of effort estimation techniques is essential for advancing the field of software project management. By addressing these gaps in the current literature and practice, this research seeks to provide valuable insights that can lead to improved estimation accuracy, better resource management, and ultimately more successful software development projects [14]. To guide this exploration and provide a structured approach to addressing these issues, the following research questions will be formulated:

1. How do different imputation techniques improve effort estimation accuracy for software development projects?
2. How does the MI technique compare with traditional imputation methods, such as listwise deletion and mean imputation, in terms of preserving dataset integrity?
3. To what extent does MI reduce data loss compared to other imputation methods, ensuring that valuable project information is retained?
4. Can MI significantly enhance the accuracy of effort estimation models, especially for large datasets with high percentages of missing data?

Building upon the insights gained from addressing the research questions, the following subsection aims to describe our main research contributions.

The main contributions of this paper are as follows:

- A detailed comparison of six imputation techniques is provided, offering insights into their effectiveness for effort estimation.
- MI is applied to the ISBSG dataset with 5,052 project records, demonstrating that MI preserves data integrity by minimizing data loss.
- The results show that MI significantly improves the accuracy of effort estimation compared to listwise deletion, mean imputation and regression imputation.
- The study highlights the importance of robust imputation techniques to avoid biases and maintain the reliability of effort estimation models, ultimately improving software project planning.

The paper is organized in the following sections: Section 2 shows the background of MD and ABEE. Section 3 explains related work. Section 4 presents the methodology, which addresses the framework and research problem. Section 5 presents the results and comparison of different techniques. Section 6 concludes the overall study with future directions discussed.

2 Background about Missing Data

2.1 Concept of Missing Data

Missing data (MD) refers to the absence of values in datasets, which can occur due to various factors such as human error, technical issues, or incomplete data collection. In software development projects, MD often affects the accuracy of effort estimation models, resource allocation, and project timelines. Left unaddressed, MD can introduce bias, reduce statistical power, and compromise the quality of predictions. Therefore, understanding the nature of MD is essential to apply the appropriate imputation techniques effectively.

2.2 Different Missing Data Mechanisms

Mechanisms of missing data help determine why data is missing and influence the choice of imputation techniques. According to Little and Rubin [15], MD is typically classified into three main types: Missing Completely at Random (MCAR) Missing at Random (MAR) Missing Not at Random (MNAR) Understanding these mechanisms is crucial to applying the correct techniques for handling missing values and avoiding biased results.

Missing Completely at Random (MCAR). The MCAR mechanism occurs when the missingness is unrelated to either observed or unobserved variables. In this case, the absence of data is entirely random and does not systematically affect the dataset.

Missing at Random (MAR). The MAR mechanism occurs when the reason for missing data is related to other observed variables in the dataset but not the missing values themselves. MAR implies that the missingness can be predicted using the observed data.

Missing Not at Random (MNAR). The MNAR mechanism occurs when the reason for the missingness is related to the value of the missing data itself. This type of missingness is the most challenging to address.

3 Related Work

ABEE is considered the most attractive and popular method for estimating effort. This method depends on historical datasets to develop new endeavors in the future. Consequently, much work from previous decades has been done on EE. Issues with datasets persist when several projects with similar features but differing sizes and numbers of software project modules—typically with fewer MDs—lead to incorrect project estimations. Several research studies demonstrate that both overestimation and underestimation occur collectively in ABEE and MD approaches. These problems impact software quality, leading to biased and inconsequential outcomes. To deal with MD values in datasets, the majority of researchers employed a variety of MD techniques, some of which include KNN imputation, Naive Baise Classifiers, Fuzzy Analogy (FA), Regression Analysis (RA), and Classical Analogy (CA) [5, 10, 14]. Nonetheless, many MD strategies have been used to enhance and determine the level of accuracy in effort estimation for SD projects. Shah et al. [4] Used simple median to impute Desherneis and Deshmiss dataset having less amount MD values to find the nearest neighbor between projects for Analogy Based Estimation (ABE) termed as Median Imputation Nearest Neighbor (MINN). However, the study revealed that MINN, compared to NC and KNNI, gives more proper values regarding datasets with missing values. It is also discovered that, in contrast to other large datasets, the MINN approach is only utilized to impute tiny datasets, such as Desherneis and Deshmiss, with fewer MDs. Furthermore, when using a simple median with static values to compare the MINN approach with NC and KNNI, there are significant problems in predicting the MMRE values for EE, according to Abnane and Idri [10]—concentrated on a few of the most popular imputation techniques. It demonstrates a noteworthy enhancement over support vector regression (SVR) and demonstrates that EBA works best when combined with KNN Imputation. According to related studies, the toleration strategy outperforms the deletion method in improving EBA accuracy. Cartwright et al. [16] address the MAR and MNAR mechanisms by assessing the issues brought up by MD in a tolerable manner. KNN and Toleration techniques were used to find the mechanisms of MD studied and reviewed by Song et al. [17], who found that MD in datasets affects the performance measure of toleration and KNN. They demonstrated that if a dataset has more than 40% missing data. MD has a more detrimental effect on analogy-based estimation. This study aims to characterize and investigate the disparities and inconsistencies in MI outcomes obtained from disparate software packages, even when the same quantity of data is employed under identical Imputation-based models in SAS and SPSS [14]. Because the naive Baise classifier is insensitive when assessing the efficiency and presence of MD in software projects, it was employed in this investigation. This investigation, in particular, reveals 30 studies and 280 candidate articles for missing values in data extraction and experimentation [18].

Several preprocessing data techniques and an empirical investigation about MD were done and described by Hosni and Idri [3] collected 35 publications were the subject of the study, 19 of which were examined to enhance the SDEE process using Case-Based Reasoning (CBR) techniques for feature weighting calculation. According to the study, 16 out of 19 papers were chosen to demonstrate how CBR features were chosen and weighted to attain an exceptional performance. According to related studies, ABEE's outstanding performance in finishing the datasets makes it the most straightforward and efficient solution for SD projects in SE. It demonstrates how important it is for SDEE to finish projects without losing any MD. This process motivates the practitioners and research community to work more efficiently to improve the imputation techniques by comparing them with other methods. Get better solutions for estimating SD projects with the help of the ABEE method, which helps project managers in the field of SE [19].

3.1 Different Approaches to Handle Missing Data

There are various useful MD handling techniques used for excluding missing values from different datasets that cause several problems when developing software projects. Earlier, this was discussed by researchers in literature from diverse sub-fields of SE. Some of them are discussed one by one and are stated as follows:

3.2 Missing Data Deletion Technique

In this technique, most field researchers deleted all case observations containing MD to complete the tasks. It is considered the most practical and widely used method for MD management. Another name for this process is Complete Case Analysis (CCA). Although the deleting process is straightforward, significant project data that contains priceless EE knowledge is lost. MD elimination could be justified when a large dataset has few missing values. Conversely, a data set with a very high percentage of missing values causes bias and inefficiency for SD projects, reducing statistical power and preventing the data set from being used to its full potential. Consequently, it is no longer suitable to delete all cases with MD values using this process [4, 10, 20, 21].

3.3 Pair Wise Deletion

When a dataset has a modest amount of MD, this strategy uses a pairwise deletion process. This strategy does not remove the entire set of project cases included in the data set. This approach uses cases and variables after the analysis with non-missing values for projects with MD when a specific variable has missing values. This approach works well for tiny datasets with few MD values. However, this technique is unsuitable for large datasets due to its complicated nature. Pairwise deletion does not compare the complete cases with other data variables due to their size, and the size of the data set changes for different estimations of parameters with missing values [10, 20, 21].

3.4 Missing Data Toleration Technique

This method bases its analysis on internal treatment performed directly on MD-containing data sets. Toleration is a simple strategy, but it is not a flexible and reliable way to effectively handle data [4, 10]. The toleration technique strategy provides inefficiency effort values and produces biased estimated results for SD projects instead of the deletion technique [10, 20–22].

3.5 Single Imputation Technique

Collecting many other Single Imputation (SI) techniques consumes minimal data processing that replaces MD with a single value. SI is only helpful in situations where the dataset has minimal missing data values. The SI approach is ineffective when dealing with data with a sizable fraction of MD values. The SI technique arises because it repeatedly imputes a fixed number of values using the mean approach added to the data set to replace the missing values [15, 22]. The single imputed value is treated as equal to the data using the SI technique. This is not attributed. Additionally, this technique introduces errors in the analysis. This method consists of a wide range of other available techniques, named arithmetic mean imputation, regression imputation (RI), stochastic regression (SR), etc. [23–25].

3.6 Mean Imputation

This method involves the arithmetic mean being gathered and used to replace any missing values in the dataset observed values present in that variable with fixed data each time. Even though mean imputation is fast in execution and generally a simple method performs well in the normal distribution, it also maintains the original dataset sample size. Besides, this method produces biased estimates, including extreme values close to the mean that decrease and underestimate the variances [25, 27].

3.7 Regression Imputation

In this technique, RI uses mean imputation and replaces all the data with missing case observations with the help of multiple regression process equations to estimate the effort values for SD projects [26]. This method is useful and easy when the number of units that respond to impute missing values is smaller. Some of the shortcomings that restrict this method are that RI creates overlapping for single values, needs a standalone model every time for each missing variable with MD in the dataset projects, and underestimates standard errors [25].

3.8 Stochastic Regression Imputation

In this technique, SR doesn't replace missing values with the help of the mean. This method uses the RI method and a random selection of different values to draw suitable and normal distributions to predict incomplete cases. This particular technique works best for large surveys with a complete set of units in the data with large samples. Besides all that, the limitation of the SR creates a problem: the underestimation of standard error gives fixed values multiple times due to random draws with zero mean and input of equal variance results [25].

3.9 Multiple Imputation

Out of several other imputation techniques, MI is considered an efficient and attractive technique when the bulk of missing values frequently occur in small and large dataset projects. MI is used as a proposed technique and a good choice where the values are imputed multiple times, for example, m = 5 iterations or more, according to the individual researcher. In addition, create a collection of several unique duplicates of the original data set for each case observation. After that, the projects with a significant number of missing values in each case observation are collected or pooled using the data taken from the MI technique to provide better estimates and improved MMRE effort values that contain valid results with maintained data quality. MI is compared with all the above-mentioned techniques to analyze the similarity between incomplete projects, and the necessary effort must be made to improve project estimating. When data is missing from historical datasets, the MI approach performs admirably in most cases, regardless of the size of the dataset. which is most likely evaluated through Manhattan and Euclidean similarity [4, 22, 28, 29].

3.10 Euclidean Distance

The distance D is measured between both the cases Pi (new project for estimation) and Pi' (old completed project) with the help of finding the summation and taking the square root represented by the value of distinct number 'n' projects with the i^{th} feature or attribute shown in Equation 1 [4, 19].

$$\text{Distance}(P, P') = \sum_{i=1}^{n} (p_i - p_i')^2 \tag{1}$$

3.11 Manhattan Distance

To find the actual aggregate by calculating both the projects absolute differences between point P and P' with the summation of i^{th} project features/attributes followed by 'n' present in Equation (2) [20, 22, 30].

$$\text{Distance}(P, P') = \sum_{i=1}^{n} |p_i - p_i'| \tag{2}$$

This paper's primary goal is to apply the MI approach to preserve the original dataset without erasing the data from various projects to preserve and protect significant and priceless information in dataset projects.

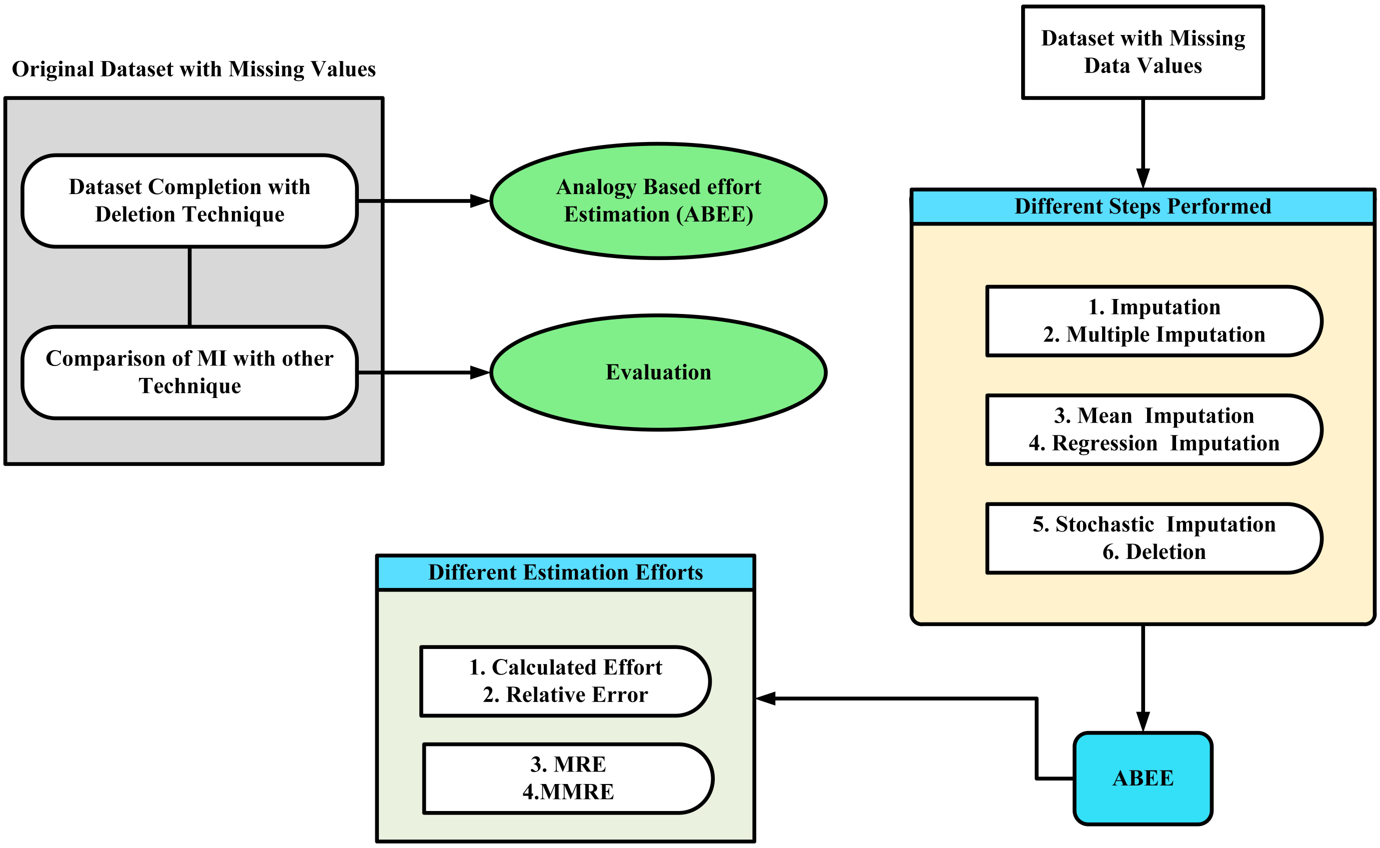

Figure 1. Block diagram of proposed methodology.

4 Methodology

The proposed MI technique is applied and tested over the ISBSG software effort estimation dataset, which was first introduced in Australia in 1997 with different project attributes collected from 24 countries. Therefore, the comparison between MI and other EE processes is shown in Figure 1.

The ABEE procedure involves estimating the effort from previous projects and aiming new dataset projects to impute MD values. The study aimed to fill in the projects with likely missing values from previous datasets using the MI approach. The main focus is to identify the effects of MD on the ABEE model and calculate the effort to estimate SD projects more efficiently and accurately without any biases. The comparison of MI with RI, mean imputation, deletion technique, and SR was used to complete projects and determine the most useful and suitable imputation method for ABEE. Moreover, using the ABEE model enhances and estimates the effort for the dataset projects. The entire procedure described in Figure 1, which shows above where Step 3 emphasizes the actual effort, RE, MRE, and MMRE are used to evaluate the performance measure of estimation accuracy and identify effort values for new projects in the dataset for better estimation with unbiased results and without any loss of important project information or reduction in statistical power. In contrast, the original dataset with missing values compares with the imputed dataset and the MI technique with the other techniques presented in Figure 1.

4.1 ISBSG Dataset Description

ISBSG is considered one of the popular data repositories, stands for International Software Benchmarking Standard Groups, and contains SD projects for multiple purposes collected from Developmental organizations throughout different countries. To evaluate and estimate numerous datasets with small and large SE projects [20]. The projects in the dataset are most probably affected by the size, missing values, cost, and duration of the projects [5]. The Joint Software Association Group established the ISBSG dataset projects in Australia in 1997, and this study makes experimental use of them. This dataset primarily aims to expand the IT industry through high-quality software solutions.

ISBSG is the most promising dataset researchers use in the SE field [4]. The global collection and upkeep of software project data repositories are the accomplishments of this dataset. The work needed to

Table 1. Descriptive statistics of the ISBSG dataset.

	N	Minimum	Maximum	Mean	St. Deviation
Functional Size	3573	3	19050	431.26	897.501
Project Elapsed Time	4229	.03	1316.90	9.7441	42.37633
Max Team Size	1541	.50	468.00	9.2821	17.06536
Normalized |Work Effort Level 1	4616	0	230514	4327.56	9755.250
Organization Type	3831	1	157	64.09	40.360
Application Type	3678	1	297	163.36	75.617
Client Server	2250	1	4	3.02	1.206
Development Techniques	2529	1	409	292.95	127.065
1st Operating System	2876	1	423	223.26	116.468
Valid N (Listwise)	217				

create software using ISBSG datasets [31]. To improve the estimation, various models are utilized to examine the ISBSG dataset using project team size, productivity rate, calendar months, and cost [32]. Additionally, the ISBSG dataset identifies the suggested MI approach as a proof of concept with various businesses to test the model's effectiveness and modify the framework's deployment and maintenance [33]. The study utilized the ISBSG dataset, which included 5052 unique software projects, to construct and estimate project effort. [5, 19]. Nine significant and related attributes/features out of the total 100 features in the dataset were owned by ISBSG, according to the demand of the project. Each feature is shown given down below. Overall statistics for 9 different features start from Functional Size (FS) and end on the first Operating System (OS) shown in Table 1. Each feature shows the total average values calculated by mean 431.26 with different numbers of function point values describing the size of the projects. The symbol "N" Checks the total number of missing values (3573) for the 9.7-month-long single dataset project and the normalized work effort level (1 4327.58 man-hours). The ISBSG dataset, considering different types of software development projects, indicates that (3) is the smallest, and this number (19050) shows the highest calculated size and FP of the project. On the other hand, the project completion time shown in decimal numbers took less time (0.03) along with (1316) months, and its standard deviation calculated in total shows the values (42.3). Meanwhile, the project efficiency with effort values ranges from minimum range zero to (230514) hours, with a standard deviation of 9755.250. List-wise deletion left (217) projects out of 5052 records in the dataset. This shows the overall technique's outcome with large data loss. The FS of project numbers 3,4 and 7 with maximum team size feature values for SD projects 2, 3, 4, and so forth are displayed, along with the MD values for the remaining 8 features in the 5052 projects in the ISBSG dataset described in Table 2.

4.2 An Overview and Strategic Approach of ABEE

An effective and popular effort estimation (EE) approach is ABEE [5]. Shepperd and Schofield introduced this method as a non-parametric empirical estimation model based on self-computing. This concept leverages modern computational techniques designed to handle imprecision, uncertainty, and other complexities, providing both speed and cost efficiency. Furthermore, soft computing techniques—used in ABEE—are well-suited to addressing complex and nonlinear problems, where traditional methods often fall short [10, 13, 34]. Due to its exceptional performance in estimating efforts for new software development (SD) projects and forecasting missing values, ABEE has gained considerable popularity [4, 10, 12].

4.3 Architecture of ABEE

Data collected from both new and old projects is compared to the essence and growing use of the ABEE approach. ABEE easily handles both qualitative and quantitative data. Figure 2 shown Analogy-Based Effort Estimation Framework. The process starts with collecting data from previous projects to compare it with new estimated dataset projects with its known effort. Further, this framework is based on four rules stated down below:

4.4 Basic Four Rules of ABEE

Proper selection of projects using historical datasets

1. Proper selection of projects using historical datasets

Table 2. Original ISBSG dataset with missing values.

Imputation	Functional Size	Project Elapsed Time	Max Team Size	Normalized Work Effort	Org_new	Apptype new	Clientserver new	Dev tech new	Os_new
0	237	6.00	5.00	1850	127	277	2	132	273
0	443	2.60	N/A	856	35	259	4	408	346
0	76	N/A	N/A	1100	16	12	5	410	424
0	3	N/A	N/A	28	158	298	4	410	424
0	382	3.00	N/A	N/A	153	170	4	410	83
0	620	7.00	N/A	18160	83	228	1	408	245
0	297	N/A	N/A	8186	14	111	5	410	73
0	113	2.60	N/A	596	14	111	4	408	372
0	183	2.80	N/A	N/A	55	172	1	410	411
0	N/A	4.00	N/A	271	139	277	5	410	424

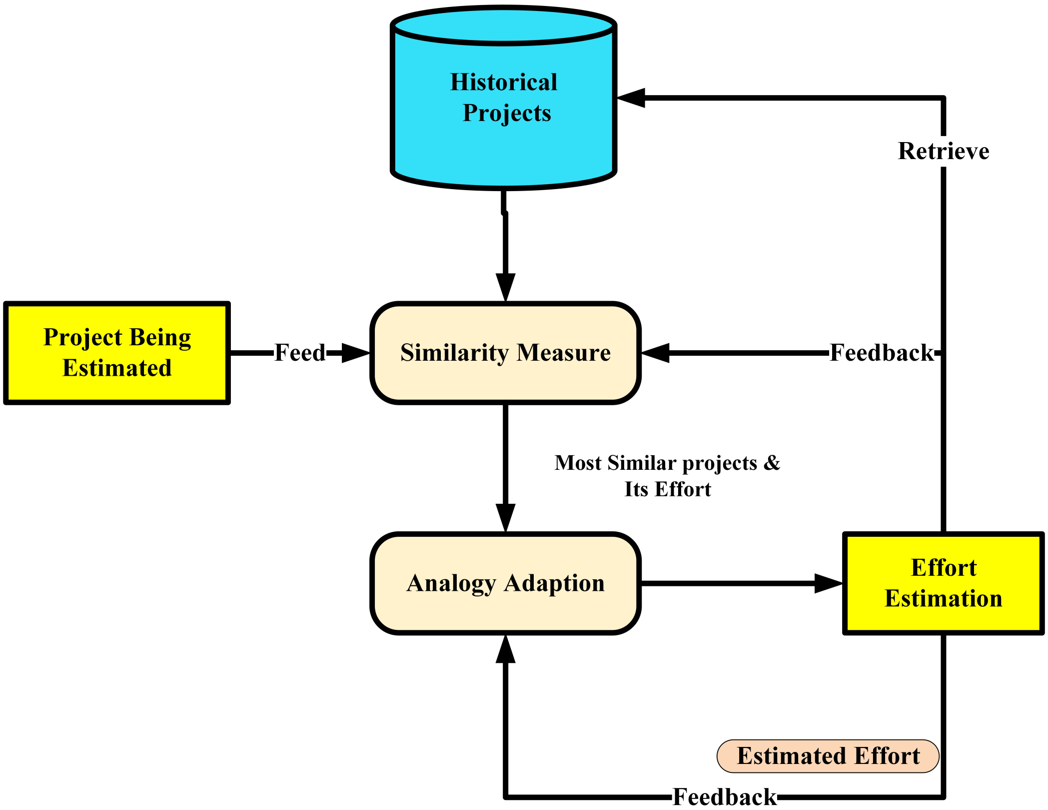

Figure 2. Analogy based effort estimation framework.

2. Select several project attributes to calculate the similarity function.

3. Using the solution function, choose and identify the projects closest to each other.

4. Follow the adopted effort of similar projects by associated rule to generate effort estimation.

5 ABEE Evaluation Performance Accuracy Metric

Several performance accuracy metrics are employed to assess the EE model's correctness and ascertain whether the model's predictions have resulted in accurate effort. Consequently, compute the real effort of the most comparable projects to assess the impact of EE approaches on performance., through the terms R.E to MMRE. One of the most commonly used evaluation performance metrics that MMRE finds near similarity to the project's effort to estimate the models of ABEE [4]. Here, each unit is defined as:

5.1 Relative Error:

The comparison of the object's actual measurement with its absolute inaccuracy. The formula for R.E. is derived from absolute error, a precise measurement error, shown in Equation 3.

The formula for $R.E$ is given by:

$$R.E = \frac{\text{Estimated} - \text{Actual}}{\text{Actual}} \tag{3}$$

5.2 Mean Relative Error (MRE)

The accuracy measured by statistical analysis technique to find the new estimated projects subtracted from the actual effort of old projects divided by the actual absolute values shown in Equation 4.

$$MRE = \frac{|\text{Estimated} - \text{Actual}|}{\text{Actual}} \tag{4}$$

5.3 Mean Magnitude of Relative Error (MMRE)

It is used for the performance measurement of SDEE projects to take out MRE values by aggregating mean, which is shown in Equation 5. The standard significance of MMRE is less or equal to the value 0.25, producing improved effort for better quality software projects to estimate the accuracy of ABEE model [4, 19].

$$\text{MMRE} = \frac{\sum_{i=1}^{N} \text{MRE}_i}{N} \tag{5}$$

The overall summary of missing values is shown in Figure 3. The form of a graph represents 9 variables out of 100% data in the ISBSG dataset with different SE-related projects having complete cases with no MD that is 217 (4.295%) out of incomplete data 4.835 (95.70%) with MD values 16,345 (35.95%) and CCA with no missing values in the projects 29,123 (64.05%).

The ISBSG dataset has chunks of MD with highlighted areas shown as monotone and the rest shows as arbitrary patterns that can be seen in Figure 3.

6 Experimental Results

This section presents the comprehensive experimental findings for each method utilized to assess the accuracy of ABEE with MD in the ISBSG dataset.

6.1 Multiple Imputation Technique.

The primary goal is to assess and quantify the often-occurring records with MD values that can be utilized in the ABEE process. Missing values give the dataset more context and enhance the estimation of SD projects without wasting any crucial data. MI is the proposed technique used in this study because of its simple and efficient nature in filling the gap of MD present in the form of bulk. It raises several other problems, such as projects with a loss of necessary information, producing biased results, reducing statistical power, etc. This technique handles MD and works quite productively for different mechanisms of MD, such as MAR, and for little MCAR missing mechanisms. MI is divided into three steps while making different imputed datasets.

1. Make imputed datasets and recode the features.
2. Imputed method of dataset analysis
3. Sorted or combined the findings of the analysis

After the analysis phase, m=5 or m=10, different iterations are performed on the ISBSG dataset to get five other different completed datasets after the execution process of the MI technique for better estimation of SDEE projects.

6.2 Comparison of Different Techniques

The MI technique is compared and tested to find different MMRE values from different ISBSG datasets. The original ISBSG dataset is incomplete, and a huge amount of MD is present in small and large SDEE projects. After the recording and pooling process to make the dataset complete through the MI technique to get different sets of imputations with complete data and different values for each single project is shown in Table 3.

Applying various techniques to ABEE using the ISBSG dataset, such as listwise deletion, mean imputation, RI, SR, and MI, produced distinct results. The MMRE values for the simpler techniques were: list-wise deletion (0.0293), mean imputation (0.0286), and regression imputation (0.0293), as summarized in Table 5. As shown in Table 4, the performance of the more advanced Stochastic Regression (SR) and Multiple Imputation (MI) techniques was evaluated across five imputed datasets. The SR technique produced MMRE values of 0.0529, 0.0583, 0.0571, 0.0612, and 0.0598, respectively, with an average of 0.0579. In comparison, the MI technique yielded more consistent and slightly lower MMRE values:

Table 3. Imputed ISBSG dataset without MD for MI.

Imputation	Functional Size	Project Elapsed Time	Max Team Size	Normalized Work Effort Level1
1	237	6.00	5.00	1850
1	443	2.60	17.90	856
1	76	-16.01	-5.94	1100
1	3	35.40	19.73	28
1	382	3.00	26.51	19267
1	620	7.00	-26.52	18160
1	297	27.92	6.33	8186
1	113	2.60	21.62	596
1	183	2.80	19.84	16418
1	-314	4.00	2.20	271

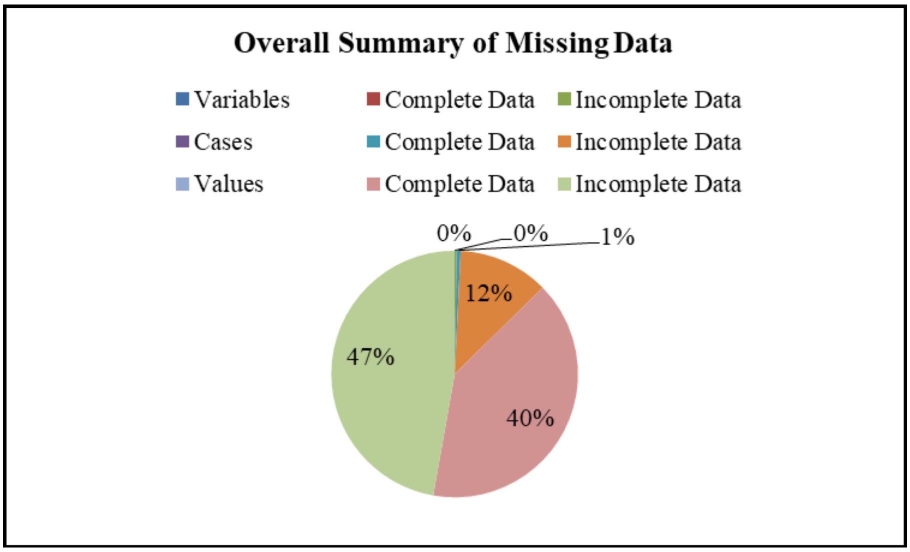

Figure 3. Original ISBSG dataset with missing values.

Table 4. Comparison of MMRE values between MI and stochastic regression techniques.

Imputation Dataset	MI Technique (MMRE)	Stochastic Regression (MMRE)
Imputation 1	0.0516	0.0529
Imputation 2	0.0535	0.0583
Imputation 3	0.0537	0.0571
Imputation 4	0.0535	0.0612
Imputation 5	0.0553	0.0598
Average	**0.0535**	**0.0579**

Table 5. Comparison of imputation techniques.

DATASET	MMRE
Listwise Deletion	0.0293
Mean Imputation	0.0286
Regression Imputation	0.0293

0.0516, 0.0535, 0.0537, 0.0535, and 0.0553, averaging 0.0535. This comparison demonstrates that while both advanced imputation methods provide acceptable accuracy (all MMRE < 0.25), the MI technique offers a marginal improvement in estimation precision and greater stability across different imputed datasets. To determine the practical impact, it is important to consider not only the accuracy metric but also the data preservation capability. While the simpler techniques (deletion, mean imputation) yielded slightly lower

MMRE values in this specific test, they achieve this at the cost of significant data loss or bias, as indicated by the drastic reduction in usable cases (from 5052 to 217 projects for listwise deletion). In contrast, the MI technique maintains the dataset's integrity by utilizing all available information, thereby increasing statistical power without discarding valuable project data. This comprehensive approach ultimately supports more reliable effort estimation throughout the software project lifecycle [4].

7 Conclusion

The handling of missing software metrics data is an important research topic in the SE community. Most researchers and practitioners handle MD by using the deletion technique to complete the dataset with a relatively major loss of valuable information from software projects. On the other hand, successful software PM is strongly associated with the accuracy of SDEE, as it can directly affect the whole cycle of any developmental project, be it planning, analysis, or scheduling. ABEE is one of the most attractive and widely used techniques for better estimating SD projects to improve effort. One more model is used, but the MI method is the one that is compared to mean imputation, SI, the deletion technique, RI, and SR to deal with MD in the ISBSG dataset without deleting any data. Numerous models have been applied to calculate the problems resulting from MD across various datasets to improve the quality of past dataset projects. Accurately and effectively estimating project effort becomes more challenging as software expands in size and complexity. As a result, individual researchers mostly use a variety of distinctive imputation strategies to address missing information in one way or another. Consequently, MI, the most straightforward and appealing technique, was employed in this work to assess and quantify the records with missing values that commonly appeared in the ISBSG dataset. The results suggested that using the proposed MI technique to handle the bulk of MD present in small and large datasets gives improved MMRE values similar to the effort values, which is an advantage compared to the rest of the techniques. The MI technique is practically implemented and compared with the original and imputed ISBSG dataset, resulting in improved results for SDEE projects for impact evaluation. The result shows that the MI technique added meaning to the dataset, which helped the PM estimate and analyze the SD project effort for ABEE models. In the future, different experiments will be practically performed on numerous distinct numbers of small and large datasets, along with evaluation performance measure metrics to improve efforts to estimate SD projects better.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Kelkar, B. A. (2022). Missing data imputation: a survey. *International Journal of Decision Support System Technology (IJDSST)*, 14(1), 1-20.

[2] Bardsiri, A. K., & Hashemi, S. M. (2014). Software effort estimation: a survey of well-known approaches. *International Journal of Computer Science Engineering (IJCSE)*, 3(1), 46-50.

[3] Hosni, M., & Idri, A. (2018). Software development effort estimation using feature selection techniques. In *New trends in intelligent software methodologies, tools and techniques* (pp. 439-452). IOS Press.

[4] Shah, M. A., Jawawi, D. N., Isa, M. A., Wakil, K., Younas, M., & Ahmed, M. (2019). MINN: A missing data imputation technique for Analogy-Based Effort Estimation. *International Journal of Advanced Computer Science and Applications*, 10(2).

[5] Idri, A., & Abnane, I. (2017, August). Fuzzy analogy based effort estimation: An empirical comparative study. In *2017 IEEE International Conference on Computer and Information Technology (CIT)* (pp. 114-121). IEEE. [CrossRef]

[6] Song, L., Minku, L. L., & Yao, X. (2018, October). A novel automated approach for software effort estimation based on data augmentation. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 468-479). [CrossRef]

[7] El Bajta, M. (2015, July). Analogy-based software development effort estimation in global software development. In *2015 IEEE 10th International Conference on Global Software Engineering Workshops* (pp. 51-54). IEEE. [CrossRef]

[8] Sharma, P., & Singh, J. (2017, December). Systematic literature review on software effort estimation

using machine learning approaches. In *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)* (pp. 43-47). IEEE. [CrossRef]

[9] Jones, T. C. (2007). *Estimating software costs*. McGraw-Hill, Inc.. [CrossRef]

[10] Abnane, I., & Idri, A. (2018, September). Improved analogy-based effort estimation with incomplete mixed data. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 1015-1024). IEEE.

[11] Kim, Y., & Lee, K. (2005). A comparison of techniques for software development effort estimating. *SYSTEM*, 407.

[12] Azzeh, M., Elsheikh, Y., & Alseid, M. (2017). An optimized analogy-based project effort estimation. *arXiv preprint arXiv:1703.04563*. [CrossRef]

[13] Shepperd, M., & Schofield, C. (1997). Estimating software project effort using analogies. *IEEE Transactions on software engineering*, 23(11), 736-743. [CrossRef]

[14] Wang, J., & Johnson, D. E. (2019). An examination of discrepancies in multiple imputation procedures between SAS® and SPSS®. *The American Statistician*, 73(1), 80-88. [CrossRef]

[15] Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

[16] Cartwright, M. H., Shepperd, M. J., & Song, Q. (2004, September). Dealing with missing software project data. In *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No. 03EX717)* (pp. 154-165). IEEE. [CrossRef]

[17] Song, Q., Shepperd, M., Chen, X., & Liu, J. (2008). Can k-NN imputation improve the performance of C4. 5 with small software project data sets? A comparative evaluation. *Journal of Systems and software*, 81(12), 2361-2370. [CrossRef]

[18] Kumar, K. H., & Srinivas, K. (2023). Preliminary performance study of a brief review on machine learning techniques for analogy based software effort estimation. *Journal of Ambient Intelligence and Humanized Computing, 14*(3), 2141-2165. [CrossRef]

[19] Mardhia, M. M., & Handayaningsih, S. (2018). Analogy-based model for software project effort estimation. International Journal of Advances in Intelligent Informatics, 4(3).

[20] Bala, A., & Abran, A. (2016). Use of the multiple imputation strategy to deal with missing data in the ISBSG repository. *Journal of Information Technology & Software Engineering*, 6, 171.

[21] Bala, A., & Abran, A. (2018). Impact analysis of multiple imputation on effort estimation models with the ISBSG repository of software projects. Softw. Meas. News, 23(1), 17-34.

[22] Idri, A., Abnane, I., & Abran, A. (2016). Missing data techniques in analogy-based software development effort estimation. *Journal of Systems and Software*, 117, 595-611. [CrossRef]

[23] Holman, C. D. A. J., Bass, J. A., Rosman, D. L., Smith, M. B., Semmens, J. B., Glasson, E. J., ... & Stanley, F. J. (2008). A decade of data linkage in Western Australia: strategic design, applications and benefits of the WA data linkage system. *Australian health review*, 32(4), 766-777.

[24] Tamura, K., Kakimoto, T., Toda, K., Tsunoda, M., Monden, A., & Matsumoto, K. I. (2008). Empirical Evaluation of Missing Data Techniques for Effort Estimation. *n3n*.

[25] Read, S. H. (2015). Applying missing data methods to routine data using the example of a population-based register of patients with diabetes.

[26] Sentas, P., & Angelis, L. (2006). Categorical missing data imputation for software cost estimation by multinomial logistic regression. *Journal of Systems and Software*, 79(3), 404-414. [CrossRef]

[27] Zhu, X. (2014). Comparison of four methods for handing missing data in longitudinal data analysis through a simulation study. *Open Journal of Statistics*, 4(11), 933.[CrossRef]

[28] González-Ladrón-de-Guevara, F., Fernández-Diego, M., & Lokan, C. (2016). The usage of ISBSG data fields in software effort estimation: A systematic mapping study. *Journal of Systems and Software*, 113, 188-215. [CrossRef]

[29] Mohanty, S. K., & Bisoi, A. K. (2012). Software effort estimation approaches-a review. *International Journal of Internet Computing*, 1(3), 82-88.

[30] Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology*, 17, 1-10.

[31] Shukla, S., & Kumar, S. (2021). An Extreme Learning Machine based Approach for Software Effort Estimation. In *ENASE* (pp. 47-57).

[32] Papageorgiou, G., Grant, S. W., Takkenberg, J. J., & Mokhles, M. M. (2018). Statistical primer: how to deal with missing data in scientific research?. *Interactive cardiovascular and thoracic surgery*, 27(2), 153-158. [CrossRef]

[33] Mahdi, M. N., Mohamed Zabil, M. H., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., ... & Happala Naidu, H. (2021). Software project management using machine learning technique—A Review. *Applied Sciences*, 11(11), 5183. [CrossRef]

[34] Živadinović, J., Medić, Z., Maksimović, D., Damnjanović, A., & Vujčić, S. (2011, June). Methods of effort estimation in software engineering. In *Proc. Int. Symposium Engineering Management and Competitiveness (EMC)* (pp. 417-422).

Shahida Hayat is a PhD student in the Department of Computer Science at the University of Peshawar, Pakistan. She completed her MS degree in Computer Science from the same department in 2017 and received her Bachelor's degree in Information Technology from the University of Peshawar. She is currently working as a lecturer at the Agriculture University, Pakistan. She was the winner of the All-Pakistan Competition at GIKI, Swabi, Pakistan. Her areas of specialization include Software Engineering, Artificial Intelligence, and Deep learning.

Wajahat Akbar is a PhD student in the School of Electronic and Control Engineering at Chang'an University Xi'an, China. He received his BS degree in Computer Science from Khushal Khan Khattak University Karak in 2019. He further pursued his academic journey at the same university and received his MS degree in Computer Science (Gold Medalist), specializing in Artificial Intelligence in 2023. He was honored with the Youth Talent Award and held a merit scholarship during his academic pursuits. His research interests span a diverse range, encompassing Artificial Intelligence, Deep Learning, Natural Language Processing (NLP), Computer Vision, Computer Networks, and Network Security, with a focus on healthcare applications.

Tariq Hussain received his B.S. and M.S. degrees in Information Technology from the University of Malakand, Pakistan (2015) and the Institute of Computer Sciences and Information Technology at the University of Agriculture Peshawar, Pakistan (2019), respectively. He has published many research papers in the area of Computer Networks. He is currently a doctoral candidate at the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, China, and the School of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou, China. He has over 35 research publications, two scientific book chapters, and a technical review committee for several international journals. He is also a review editor for Frontier in Big Data, Data Science, and Drone Technology Journals. His research interests are the Internet of Things, Big Data, data analytics, 3D Point Cloud, and Artificial Intelligence.

Muhammad Inam Ul Haq received his MS-IT from the Institute of Management Sciences, University of Peshawar, Pakistan, and his Ph.D. from Jean Monnet University, Saint-Etienne, France. He works as an Assistant Professor in the Department of Computer Science and Bioinformatics at Khushal Khan Khattak University, Karak, Pakistan. He has published several research papers in computer science and is a member of the technical review committee for several international journals. His research interests include computer vision, image processing, networks, optonumeric security, deep learning, and NLP.

Altaf Hussain holds a Bachelor of Science in Computer Science from the University of Agriculture-Peshawar, Pakistan (2016) and a Master of Science in Computer Science from Khushal Khan Khattak University, Karak-Pakistan (2023). He has actively contributed to research in blockchain, differential privacy, and Federated Learning, particularly focusing on data privacy and security innovations. He has published research articles, showcasing his pioneering work in these areas. His interests span a broad spectrum, including Data Security, Network Security, Artificial Intelligence, Computer Networks, Machine Learning, and Deep Learning. His work is driven for enhancing secure and accurate digital ecosystems through cutting-edge technologies.

Irshad Khalil was born in Malakand, Pakistan in 1992. He received his bachelor's degree in computer science from University of Malakand, Pakistan, in 2013. He received MS degree in Image Processing from University of Malakand, Pakistan, in 2016. Now he is a Ph.D. scholar at Gachon University, Incheon 21936, Korea, and his current research interests include image processing, medical imaging, Internet of Medical Things and Deep learning, AI.

Muhammad Nawaz Khan is working as an Assistant Professor in the Higher Education, Archives Libraries Department at KPK, Pakistan. He received a Silver Medal in Computer Science for his B.S. (Hons) degree from the University of Malakand in 2008. In 2010, he worked as a Research Assistant on a project related to Distributed Computing supported by the Higher Education Commission of Pakistan. He completed his MS in Computer Science from SZABIST Islamabad and PhD from the University of Malakand in Pakistan in 2023. His interest areas are Wireless Sensor Networks, Internet of Things, and Information Security. In addition, he reviews various publications, including IEEE ACCESS, Array (Elsevier), Computational and Mathematical Methods in Medicine, Scientific Programming, and others.

Samsonova Diana received his B.S at the School of Computer Science and Technology, Zhejiang Gongshang University, Hangzhou, China, and the School of International Education, Zhejiang Gongshang University, Hangzhou, China. Her research interests are the E-commerce, Big Data, Data Analytics, International Business.