Evaluating Fuzzy Analogy on Incomplete Software Projects data

Ibtissam Abnane and Ali Idri
Software Project Management Research Team
ENSIAS, University Mohammed V in Rabat, Morocco
{ali.idri, ibtissam_abnane}@um5.ac.ma

***Abstract*— Missing Data (MD) is a widespread problem that can affect the ability to use data to construct effective software development effort prediction systems. This paper investigates the use of missing data (MD) techniques with Fuzzy Analogy. More specifically, this study analyze the predictive performance of this analogy-based technique when using toleration, deletion or k-nearest neighbors (KNN) imputation techniques using the Pred(0.25) accuracy criterion and thereafter compares the results with the findings when using the Standardized Accuracy (SA) measure.**

A total of 756 experiments were conducted involving seven data sets, three MD techniques (toleration, deletion and KNN imputation), three missingness mechanisms (MCAR: missing completely at random, MAR: missing at random, NIM: non-ignorable missing), and MD percentages from 10 percent to 90 percent. The results of accuracy measured in terms of Pred(0.25) confirm the findings of a study which used the SA measure. Moreover, we found that SA and Pred(0.25) measure different aspects of technique performance. Hence, SA is not sufficient to conclude about the technique accuracy and it should be used with other metrics, especially Pred(0.25).

Keywords*— *Analogy-based Software Development Effort Estimation, Missing Data, Imputation , Fuzzy Analogy.

I. Introduction

Software development effort/cost estimation (SDEE) is the process of predicting the effort required to develop a software system[1]. Estimating development effort accurately in the early stage of software life cycle plays a crucial role in effective project management. Hence, delivering an accurate estimate of effort remains a challenge for software community [2].

Attempts to estimate the effort and time involved in the development of a software product usually involve the construction/use of one or more effort estimation techniques [3]. However, a major problem in building estimation models arises when missing data (MD) are encountered in historical data sets [4]. The lack of data values in several important project attributes is a common phenomenon that may cause misleading results regarding the model's accuracy and prediction ability [5]. Most of the software databases suffer from this problem, which is the result of several reasons related to the demanding process of collecting adequate data. Indeed, the data collection requirements for a company include consistency, experience, time, cost and methodology [5].

Wen et al. carried out a systematic literature review in which they identified eight types of machine learning techniques used in SDEE [1]. Among them, analogy-based software effort estimation (ASEE) and artificial neural networks (ANN) were the most frequently used techniques for effort estimation, accounting for 37% and 26%, respectively. This confirms the results of the study by Jorgensen and Shepperd (2007): ASEE is gaining in popularity over other ML techniques (10% compared to 7% for ANN and 5% for Classification and Regression Trees up to 2004).

Idri et al conducted a systematic literature review (SLR) to gain more knowledge about the use of ASEE [6]. Their results suggest that ASEE tends to yield acceptable estimates. More specifically, the mean of the prediction accuracy values is 49.8% for MMRE, 29.4% for MdMRE, and 51.2% for Pred (0.25)). Moreover they found that ASEE outperforms the other prediction models. This conclusion is supported by most of the authors' selected papers. In contrast, their SLR showed that some data set properties still present serious challenges for ASEE: 1) it cannot adequately handle categorical data other than binary valued attributes [7]; 2) it is sensitive to irrelevant features and the degree of attribute influence on effort estimates; and 3) it cannot be directly applied to historical data sets containing missing data [8].

To examine the research on the use of MD techniques in SE (Software Engineering) data sets, Idri et al. have conducted a mapping study in which 35 papers concerning MD treatments of SE data were selected [9]. They found that the topic of MD techniques in SE data sets is taken seriously by researchers, as observed by the number of papers presented in conferences/symposiums and published in journals. They also found that most papers that investigated all mechanisms of missingness insist on taking the missingness mechanisms into account while investigating MD techniques and showing how these mechanisms can significantly affect their performance. Moreover, they found that the main motivation behind investigating MD techniques in SE data sets was to predict software development effort.

In a previous work, we presented an empirical comparative study of two ASEE techniques (classical and fuzzy analogy) using three MD techniques [10]: toleration, deletion and KNN imputation with different percentages (from 10% to 90%) and missingness mechanisms (MCAR, MAR and NIM) on seven data sets: ISBSG R8, COCOMO81, Desharnais, China, Maxwell, Miyazaki and Albrecht. We investigated the evaluation of the performance of classical/fuzzy analogy techniques when dealing with MD in software development effort estimation using an unbiased Standardized Accuracy measure (SA) proposed by Shepperd and MacDonell [2].

However, SA values would only confirm if a technique is reasonable (e.g. is actually predicting and how much is it better than random guessing) which is not sufficient to conclude about the technique accuracy [11]–[13]. In fact, although SA results would confirm the ability of the SDEE technique to produce more meaningful prediction than random guessing, SA is not sufficient to provide a final decision about the technique accuracy as stated by Azzeh et al [11]. Therefore, SA was always used in conjunction with other metrics.

Within this context, we extend the evaluation of Fuzzy Analogy in the presence of MD by using a well-known accuracy metric, Pred(0.25). Pred(0.25) evaluates the accuracy of a technique (e. g. count the number of cases where the technique gave close values to the actual effort). It is the most used measure to assess the accuracy of SDEE [1], [6]. Pred(0.25) is widely used by the SDEE community since it is easily interpretable, robust and nearly immune to outliers [14]. Hence, this study investigates whether the results when using Pred(0.25) would confirm the findings of our prior work which used the SA measure.

The structure of the paper is the following: Section 2 briefly introduces the different missingness mechanisms of MD and the most common techniques for handling them. Section 3 presents an overview of Fuzzy Analogy. Section 4 describes the data sets used. Section 5 presents the experiment design, which includes the criteria for evaluating estimation accuracy, the process of introducing MD, and the experiment process itself. The results are presented and discussed in Section 6. Section 7 concludes by discussing the findings as well as suggesting some directions for future work.

II. Concepts of Missing Data

This section discusses the different missingness mechanisms (i.e., different ways in which data can be missing) and presents the MD techniques used in this study.

A. Missingness Mechanisms

Missingness mechanisms are assumptions about the nature and types of MD. Little and Rubin defined three such mechanisms[15]: (1) MCAR (missing completely at random) means that the MD are independent of any variable observed in the data set; (2) MAR (missing at random) means that the MD may depend on variables observed in the data set, but not on the MD themselves; (3) NIM (non-ignorable missing) means that the MD depend on the MD themselves, and not on any other observed variable. In general, a missingness mechanism concerns whether the missingness is related to the study variables or not. This is very significant as it determines how difficult it may be to handle the MD and, at the same time, how risky it is to ignore them [15].

B. MD Techniques

The MD problem has been studied by researchers in many subfields of software engineering for more than 30 years [16–[18]. There are three approaches to deal with this problem [19], [20]:

- MD ignoring technique which simply deletes the cases that contain MD. Because of its simplicity, it is widely used; however, this may not lead to the most efficient utilization of the data and incurs a bias in the data unless the values are missing completely at random. Consequently it may be used only in situations where the level of MD is very low [20].
- MD toleration techniques are internal treatment strategies in which analysis is performed directly, using the data sets with MD. In order to tolerate MD in the data sets of this study, a special value NULL is used to replace MD in the data sets. A similar approach was used in Li et al.[19].
- MD imputation techniques refer to any strategy for filling in the MD of a data set so that standard methods can be applied to analyze the completed data set. This study uses KNN imputation, a popular technique that has proven to yield good results in software effort estimation [20]–[22]. Moreover, it has no explicit missingness mechanism assumption, which makes it highly practical. This technique uses other complete cases within the data set as donors to impute an incomplete case. The MD of an incomplete case are replaced by aggregating the values of its k nearest neighbor cases. To determine the k nearest neighbors, the similarity between the incomplete/complete cases is measured by means of a given distance. The following distances have been used to find nearest cases [23]:

  Euclidean Distance: It measures the distance between two cases x and y described by n attributes using Equation (1).

$$\boldsymbol{d}(\boldsymbol{x},\boldsymbol{y})= \sqrt{\sum_{j=1}^{n}(x_j - y_j)^{\,2}} \quad (1)$$

  Manhattan Distance: It is the sum of the absolute differences between two points x and y described by n attributes using Equation (2).

$$\boldsymbol{d}(\boldsymbol{x},\boldsymbol{y}) \;=\; \sum_{j=1}^{n}\left|x_j - y_j\right| \quad (2)$$

III. Fuzzy Analogy Description

Analogy-based software effort estimation (ASEE) is based on the principle that actual values achieved in an earlier and similar project are better indicators to predict the future project performance [24]. The three steps of Fuzzy Analogy are detailed below.

A. Identification of Cases

In this step, each project is described by a set of relevant and independent attributes. These attributes can be measured by either numerical or linguistic values. Unlike traditional

ASEE approaches in which numerical values are represented by classical intervals, in fuzzy analogy numerical values are transformed into linguistic ones. Let us suppose that a project P is described by M numerical/linguistic variables (V_j). Then, for each numerical/linguistic variable V_j, a measure with linguistic values is defined (A_k^j). Each linguistic value A_k^j is represented by a fuzzy set with a membership function$\mu_{A_k^j}$. These fuzzy sets and their membership functions can be built (1) empirically, using expert knowledge, or (2) automatically, using clustering techniques [25], [26]. In particular, when the descriptions of software attributes are insufficient to empirically build their fuzzy representations, fuzzy analogy uses an automated process to build fuzzy sets and their membership functions [25], [26]. The proposed fuzzy set generation process is based on the fuzzy C-means clustering technique (FCM) and a real coded genetic algorithm (RCGA). This process consists of two main steps. First, the FCM algorithm and the Xie-Beni validity criterion are used to decide on the number of clusters (fuzzy sets)[27], [28]. For each software project attribute, several experiments are conducted with the FCM algorithm with different number of clusters (c) and different values of the parameter m. For each attribute, we choose the number of clusters and the parameter m that minimize the value of the Xie-Beni criterion [26]. Second, an RCGA is used to build membership functions for these fuzzy sets [29]. Membership functions can be trapezoidal, triangular, or Gaussian.

B. Retrieval of Similar Cases

To measure the similarity between two software projects described by linguistic values such as 'low' or 'high', Fuzzy Analogy proposes a set of new measures based on fuzzy logic [30]. Two steps are required for evaluating similarity using these measures:

Individual similarities: Assessing the similarity between two projects P_1 and P_2, according to each individual attribute V_j describing P_1 andP_2 by means of Equation (3).

$$S_{V_j}(P_1, P_2) = \begin{cases} max - \min aggregation \\ max_k \min(\mu_{A_K^I}(P_1), \mu_{A_K^I}(P_2)) \\ sum - product\ aggregation \\ \sum_k \mu_{A_K^I}(P_1) * \mu_{A_K^I}(P_2) \end{cases} \tag{3}$$

Global similarities: Evaluating overall similarity S (P_1, P_2) by aggregating the individual similarities S (P_1,P_2) using Regular Increasing Monotone (RIM) linguistic quantifiers, such as 'all', 'most', 'many', or 'some'. The choice of the appropriate RIM linguistic quantifier, Q, depends on the characteristics and needs of each environment. *Q*(Equation (4)) indicates the proportion of individual distances that we feel is necessary for a good evaluation of the overall distance. The overall similarity of P_1 and P_2, S (P_1,P_2) is given by one of the formulas of Equation (5).

$$Q = r^{\alpha};\ \alpha > 0 \tag{4}$$

$$S(P_1, P_2) = \begin{cases} \text{All of } \left(S_{V_j}(P_1, P_2)\right) \\ \text{Most of } \left(S_{V_j}(P_1, P_2)\right) \\ \text{Many of } \left(S_{V_j}(P_1, P_2)\right) \\ \cdots \\ \text{There exists of } \left(S_{V_j}(P_1, P_2)\right) \end{cases} \tag{5}$$

C. Case Adaptation

The objective of this step is to derive an estimate for the new project by using the known effort values of similar projects. The first task is to decide how many similar projects will be used in the adaptation phase to estimate the effort of the new project. Fuzzy Analogy has proposed a strategy based on similarity measures S (P,P_i) and the definition adopted in the studied environment for the proposition 'P_i is a project that is highly similar to P'. Intuitively, $P_i$is highly similar to P if S (P,P_i) is in the vicinity of 1. To represent the value 'vicinity of 1', we use a fuzzy set defined in the interval [0, 1]. The second task is how to adapt the chosen analogies in order to generate an estimate for the new project. Fuzzy Analogy uses the weighted mean of all known effort projects in the dataset; the weights being the similarity distance. The formula is given by Equation (6).

$$Effort(P) = \frac{\sum_{i=1}^{N} \mu_{vincity\ of\ 1}(S\ (P,P_i)) * Effort\ (P_i)}{\sum_{i=1}^{N} \mu_{vincity\ of\ 1}\ (S\ (P,P_i))} \tag{6}$$

where $\mu_{vincityof\ 1}$is the membership function representing the linguistic value 'vicinity of 1'.

IV. Data description

SDEE techniques evaluation is highly affected by the characteristics of the chosen datasets such as size, number of attributes describing a software project, missing data and outliers [31]. In this study, we used seven datasets from two sources: 1) ISBSG (The International Software Benchmarking Standards Group data repository) [32]; and 2) PROMISE (The PRedictOr Models In Software Engineering data repository) [33]. Note that the aim of this study is to deal only with numerical attributes. Thus, for the selected datasets from PROMISE, all projects and numerical features were used in the evaluations.

However, for ISBSG dataset and according to our previous studies, only 148 projects and 9 numerical attributes were used [25], [34]. Table 1 gives the description of the seven selected datasets, including the number of attributes, the number of historical projects, the effort units, and the minimum, maximum, mean, median, skewness and kurtosis of efforts. These datasets are diverse in terms of their fields, their sources and theirs sizes.

TABLE I. DESCRIPTION STATISTICS OF THE SELECTED DATASETS

Dataset	#of Projects	Effort Unit	#of features	Effort					
				Min	Max	Mean	Median	Skewness	Kurtosis
Albrecht	24	Man/Months	7	0.5	105.2	21.87	11.45	2.30	4.66
COCOMO	252	Man/Months	13	6	11400	683.44	98	4.39	20.5
China	499	Man/Hours	18	26	54620	3921.04	1829	3.92	19.3
Desharnais	77	Man/Hours	12	546	23940	4833.9	3542	2.03	5.3
ISBSG	148	Man/Hours	10	24	60270	6242,6	2461	3.05	11.34
Kemerer	15	Man/Months	7	23.2	1107.31	219.24	130.3	3.07	10.59
Miyazaki	48	Man/Months	8	5.6	1586	87.47	38.1	6.26	41.35

V. EXPERIMENT DESIGN

This section describes the experimental process used in this study. It consists of four main steps: data removal, complete data set generation, accuracy evaluation and significance testing. The study was designed to apply Fuzzy Analogy with nine percentages of incomplete data, three different MD mechanisms (MCAR, MAR and NIM), and four techniques for handling MD (toleration, deletion and KNN imputation using Euclidean/Manhattan distances) on seven data sets. Hence, the experimental design consisted in evaluating 9×4×3×7= 756 different effort estimation experiments.

A. Step 1: Data Removal

The first step in the experimental process requires a complete data set to work with. For this purpose, we first preprocessed the seven data sets by deleting cases with MD to obtain the corresponding seven complete data sets. We then used the complete data sets to artificially generate MD by mimicking the different mechanisms. A similar approach was followed in [4], [10], [20]: 1) MCAR mechanism was simulated by inducing MD completely at random for each variable. 2) Implementation of the MAR mechanism was based on an attribute of each data set, namely Interface Count for ISBSG, Software Size for COCOMO81, Size for Maxwell, SCRN for Miyazaki, Input for China, Transactions for Desharnais and Input for Albrecht respectively. Specifically, we first sorted the cases in ascending order of the selected attribute. We then divided the sorted data set into three subsets of equal size. If the total percentage of MD is p, the percentage of MD in the three subsets will be 0.6×p, 0.4×p and 0, so that the MD will be induced with bias related to the selected attribute. The MD in each subset was assigned randomly. 3) NIM mechanism was implemented in a very similar way to MAR. We sorted the data sets according to the same attributes mentioned in the MAR process, except that the MD were not induced to all attributes but only to the attribute according to which the projects were sorted.

Nine MD percentages (from 10% to 90%) were simulated. By combining the seven data sets, three missingness mechanisms and nine percentages, we obtained 189=7×3×9 incomplete data sets at this stage.

B. Step 2: Complete Data Set Generation

This step involves four MD techniques: toleration, deletion, and KNN imputation with Euclidean and Manhattan distances, which were used to generate complete data sets from those in Step 1. As mentioned in Section 1.2.2, the toleration technique uses a special value *NULL* to replace a missing value in a data set. Hence, the local and global similarity measures can be calculated in the presence of *NULL* values. A similar approach was used in [19], [10]. Under the deletion technique, cases with missing values of any variable are omitted in the analysis. Under the imputation techniques, the two KNN methods (KNN with Euclidean distance and KNN with Manhattan distance) were applied to each missing data value in the data removal step; we used the median value of the *K* nearest neighbors to replace a missing value. Applying these four techniques to the 189 data sets resulting from the data removal step, we obtained 756 complete data sets.

C. Step 3: Accuracy Evaluation

In our previous work [10], the performance of Fuzzy Analogy was evaluated using the SA measure proposed by Shepperd and MacDonell [2], which is based on the Mean Absolute Residual (MAR). The SA is defined by Equation (7).

$$\mathrm{SA}_{\mathrm{P_i}} = 1 - \frac{\mathrm{MAR}_{\mathrm{P_i}}}{\mathrm{MAR}_{\mathrm{P_0}}} \times 100 \qquad (7)$$

where MAR_{P_i} is defined as the MAR of the estimation method P_i and MAR_{P_0} is the mean of a large number of random guesses (in our case 1000). SA gives us an idea of how good an estimation method is in comparison to random guessing. Since the term MAR_{P_i} is in the nominator, the higher the SA values, the better the estimation method.

In addition to SA, this study uses the Pred(0.25) criterion to evaluate the accuracy of the estimates of Fuzzy Analogy. The evaluation method used is the Jackknife method (also called leave-one-out cross-validation). It consists on excluding the target project from the historical data set and its effort calculation is performed using the actual effort values of the remaining projects. The Pred(0.25) measure is based on the MRE measure. It is one of the most well-known measures used to assess the accuracy of SDEE techniques [1], [6]:

- MRE is calculated for each project in the dataset following Equation (8):

$$\mathrm{MRE} = \left|\frac{\mathrm{Effort_{actual}} - \mathrm{Effort_{estimated}}}{\mathrm{Effort_{actual}}}\right| \times 100 \qquad (8)$$

- Pred(p) is defined as the percentage of successful predictions falling within p% of the actual values, and is calculated by Equation (9):

$$\text{Pred}(p) = \frac{k}{N} \tag{9}$$

where N is the total number of observations, and k is the number of observations with an MRE less than or equal to p.

D. Significance Testing

Although Pred(0.25) can show if any of the MD techniques are better than others in a descriptive and graphical manner, the remaining question is whether the observed differences are statistically significant [35]. We used the Wilcoxon statistical test which is a non-parametric procedure used to test whether there is sufficient evidence that the median of two probability distributions differ in location [36]. Being a non-parametric test, it does not make any assumptions about the form of the underlying probability distribution of the sampled population. All statistical tests were two-sided and performed at α=0.05 significance level. To adjust for multiple testing, we used the Holm-Bonferroni method [37].

VI. Results

This section presents and discusses the experimental results regarding the effects of MD techniques on the predictive accuracy of Fuzzy Analogy. First, the impact of these techniques is explored for different MD percentages and missingness mechanisms to analyze their influence on the accuracy values of Fuzzy Analogy in terms of Pred(0.25). Then, we investigated whether the results when using Pred(0.25) would confirm the findings of a prior work which used the SA measure.

a) Accuracy of Fuzzy Analogy in terms of Pred(0.25)

To analyze the effect of the missingness mechanism on the accuracy of Fuzzy Analogy in terms of Pred(0.25), Table 2 shows the median Pred(0.25) values for Fuzzy Analogy applied to the seven data sets with three mechanisms of missingness, different MD percentages and four MD techniques. Moreover, to test if the accuracy of Fuzzy Analogy is significantly affected by the MD mechanisms and MD techniques, we compared the median of Pred(0.25) values across datasets for each MD percentage using the Wilcoxon t-test. For example, to compare the effect of MCAR and MAR mechanisms when using toleration technique, we compared their median Pred(0.25) values across datasets for nine MD percentages. Consequently, for each mechanism, nine values are provided and each one is the median Pred(0.25) values across datasets for a MD percentage. Therefore, the following hypotheses were drawn in order to test if Fuzzy Analogy is significantly affected by the MD techniques and mechanisms:

1) Null Hypothesis 1 (NH1): The prediction accuracy of Fuzzy Analogy in terms of Pred(0.25) is not affected by the MD technique used.

2) Null Hypothesis 2 (NH2): The prediction accuracy of Fuzzy Analogy in terms of Pred(0.25) is not affected by the mechanism of missingness.

TABLE II. Median Pred(0.25) and SA of Fuzzy Analogy applied to seven data sets with three mechanisms of missingness, different MD percentages and four MD techniques

		MCAR		**MAR**		**NIM**	
		Pred (0.25)	**SA**	**Pred (0.25)**	**SA**	**Pred (0.25)**	**SA**
KN N-Euclidean	**10**	0.75	0.75	0.72	0.70	0.62	0.60
	20	0.70	0.71	0.66	0.67	0.6	0.57
	30	0.66	0.65	0.61	0.62	0.54	0.52
	40	0.64	0.61	0.54	0.53	0.48	0.47
	50	0.53	0.57	0.45	0.49	0.44	0.43
	60	0.40	0.52	0.41	0.46	0.38	0.39
	70	0.37	0.48	0.38	0.41	0.34	0.35
	80	0.34	0.46	0.38	0.38	0.33	0.3
	90	0.32	0.43	0.25	0.34	0.25	0.27
KNN-Manhattan	**10**	0.63	0.72	0.65	0.66	0.55	0.62
	20	0.62	0.69	0.58	0.63	0.54	0.58
	30	0.62	0.66	0.56	0.59	0.51	0.53
	40	0.58	0.61	0.5	0.54	0.44	0.5
	50	0.47	0.58	0.42	0.51	0.44	0.47
	60	0.42	0.51	0.39	0.49	0.41	0.41
	70	0.35	0.49	0.35	0.45	0.33	0.36
	80	0.33	0.47	0.31	0.4	0.32	0.34
	90	0.32	0.45	0.26	0.39	0.26	0.28
Deletion	**10**	0.59	0.65	0.53	0.56	0.48	0.53
	20	0.54	0.61	0.45	0.52	0.41	0.46
	30	0.54	0.53	0.4	0.47	0.36	0.39
	40	0.47	0.45	0.3	0.4	0.22	0.35
	50	0.39	0.39	0.29	0.36	0.21	0.3
	60	0.37	0.34	0.25	0.29	0.17	0.25
	70	0.21	0.29	0.21	0.22	0.16	0.21
	80	0.2	0.26	0.13	0.17	0.13	0.14
	90	0.2	0.15	0.11	0.13	0.08	0.12
Toleration	**10**	0.58	0.66	0.49	0.59	0.38	0.54
	20	0.48	0.61	0.42	0.56	0.33	0.51
	30	0.43	0.58	0.38	0.49	0.31	0.47
	40	0.43	0.53	0.38	0.44	0.29	0.43
	50	0.39	0.49	0.36	0.41	0.27	0.37
	60	0.36	0.42	0.3	0.34	0.22	0.32
	70	0.35	0.37	0.26	0.29	0.16	0.27
	80	0.27	0.28	0.25	0.25	0.15	0.2
	90	0.22	0.24	0.24	0.2	0.16	0.19

Each null hypothesis was tested separately for each MD technique and each missingness mechanism. Tables 3 and 4 summarize the results of the Wilcoxon t-test conducted to assess the NH1 and NH2 respectively, where p(α) denotes the p-value of the Wilcoxon test and p(α’) denotes the p-value corrected by the Holm-Bonferroni method.

Under the MCAR mechanism, we notice that Fuzzy Analogy using KNN-Euclidean gave the best accuracy, followed by Fuzzy Analogy using KNN-Manhattan and then toleration and deletion. (Pred(0.25) values at 10% MD: 76%, 63%, 58% and 59% for Fuzzy Analogy using KNN-Euclidean, KNN-Manhattan, toleration and deletion, respectively). Moreover, Table 3 shows that: 1) Fuzzy Analogy using KNN significantly outperformed Fuzzy Analogy when using deletion (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') = 0.0167$; for KNN-Manhattan: $p(\alpha) = 0.0077$ and $p(\alpha') = 0.0125$) or toleration (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') = 0.0083$; for KNN-Manhattan ($p(\alpha) = 0.0077$ and $p(\alpha') = 0.01$); 2) Fuzzy Analogy using toleration and deletion techniques behaved almost the same ($p(\alpha) = 0.767$ and $p(\alpha') =0.05$); and 3) Fuzzy Analogy using KNN-Euclidean significantly outperformed Fuzzy Analogy using KNN-Manhattan ($p(\alpha) = 0.0077$ and $p(\alpha') = 0.025$). Note that Fuzzy Analogy using deletion outperformed toleration at small percentages of MD (Pred(0.25) values at 40% MD: 43% and 47% for Fuzzy Analogy using toleration and deletion respectively). However, when the MD percentage exceeds 70%, Fuzzy Analogy using toleration gave better accuracy than deletion (Pred(0.25) values at 70% MD: 35% and 21% for Fuzzy Analogy using toleration and deletion respectively). We notice from Table 2 that the accuracy of Fuzzy Analogy decreases as the MD percentage increases for the four MD techniques.

Regarding the MAR mechanism, we notice that Fuzzy Analogy using KNN-Euclidean gave the best accuracy, followed by Fuzzy Analogy using KNN-Manhattan, toleration and deletion (Pred(0.25) values at 10% MD: 72%, 65%, 49% and 53% for Fuzzy Analogy using KNN-Euclidean, KNN-Manhattan, toleration and deletion, respectively). Moreover, Table 3 shows that: 1) Fuzzy Analogy using KNN significantly outperformed Fuzzy Analogy when using deletion (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') =0.0125$; for KNN-Manhattan: $p(\alpha) = 0.0077$ and $p(\alpha') =0.0167$) or toleration (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') = 0.0083$; for KNN-Manhattan ($p(\alpha) = 0.0077$ and $p(\alpha') = 0.01$); 2) Fuzzy Analogy using toleration and deletion techniques behaved almost the same ($p(\alpha) = 0.051$ and $p(\alpha') = 0.05$); and 3) Fuzzy Analogy using KNN-Euclidean significantly outperformed Fuzzy Analogy using KNN-Manhattan ($p(\alpha) = 0.021$ and $p(\alpha') = 0.025$).Moreover, we notice that Fuzzy Analogy using deletion outperforms toleration at small percentages of MD (Pred(0.25) values at 20% MD: 41% and 45% for Fuzzy Analogy using toleration and deletion respectively). However, as the MD exceeds 40% Fuzzy Analogy using toleration gave better results than when using deletion (Pred(0.25) values at 40% MD: 38% and 30% for Fuzzy Analogy using toleration and deletion respectively).

Under the NIM mechanism, we notice that Fuzzy Analogy using imputation gave better accuracy than when using toleration or deletion (Pred(0.25) values at 10% MD: 62%, 55%, 38% and 48% for Fuzzy Analogy using KNN-Euclidean, KNN-Manhattan, toleration and deletion, respectively).

TABLE III. COMPARISON OF P-VALUES OBTAINED IN THE WILCOXON TEST OF NH1

		KNN-E	KNN-M	Deletion
		p(α)/ p(α’)	p(α)/ p(α’)	p(α)/ p(α’)
MCAR	Toleration	0.0077/0.0083	0.0077/0.01	0.767/0.05
	KNN-E		0.0077/0.025	0.0077/0.0167
	KNN-M			0.0077/0.0125
MAR	Toleration	0.0077/0.0083	0.0076/0.01	0.051/0.05
	KNN-E		0.021/0.025	0.0077/0.0125
	KNN-M			0.0077/0.0167
NIM	Toleration	0.0077/0.0083	0.00770/0.01	0.859/0.005
	KNN-E		0.411/0.025	0.0077/0.0125
	KNN-M			0.0077/0.0167

Moreover, Table 3 shows that: 1) Fuzzy Analogy using KNN significantly outperformed Fuzzy Analogy when using deletion (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') =0.0125$; for KNN-Manhattan: $p(\alpha) = 0.0077$ and $p(\alpha') =0.0167$) or toleration (For KNN-Euclidean: $p(\alpha) = 0.0077$ and $p(\alpha') = 0.0083$; for KNN-Manhattan ($p(\alpha) = 0.0077$ and $p(\alpha') = 0.01$); 2) Fuzzy Analogy using toleration and deletion techniques behaved almost the same ($p(\alpha) = 859$ and $p(\alpha') = 0.05$); and 3) Fuzzy Analogy using KNN-Euclidean and KNN-Manhattan behaved almost the same ($p(\alpha) = 0.411$ and $p(\alpha') = 0.025$). Table 2 shows that Fuzzy Analogy using deletion outperforms toleration at small percentages of MD (Pred(0.25) values at 30% MD: 31% and 36% for Fuzzy Analogy using toleration and deletion respectively). However, when the MD percentage exceeds 40%, we notice that Fuzzy Analogy performs better using toleration technique (Pred(0.25) values at 40% MD: 29% and 22% for Fuzzy Analogy using toleration and deletion respectively).

To compare the impact of the missingness mechanisms on the predictive accuracy of Fuzzy Analogy, Table 4 presents the results of the statistical tests of NH2. Table 4 shows that: 1) Fuzzy Analogy with MCAR achieved significantly better Pred(0.25) over Fuzzy Analogy with MAR when using toleration, KNN-Euclidean or deletion (For toleration: $p(\alpha) = 0.011$ and $p(\alpha') = 0.05$; KNN-Euclidean: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.0167$ and Deletion: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.0167$); 2) The performance of Fuzzy Analogy under MCAR and MAR is not significantly different when using KNN-Manhattan ($p(\alpha) = 0.017$ and $p(\alpha') = 0.0167$); 3) Fuzzy Analogy with MCAR achieved significantly better Pred(0.25) over Fuzzy Analogy with NIM regardless of the MD technique used (For toleration: $p(\alpha) = 0.076$ and $p(\alpha') = 0.0250$; KNN-Euclidean: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.0250$, KNN-Manhattan: $p(\alpha) = 0.021$ and $p(\alpha') = 0.0250$ and Deletion: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.0250$; 4) Fuzzy Analogy with MAR also achieved significantly better Pred(0.25) over Fuzzy Analogy with NIM when using toleration, KNN-Euclidean or deletion (For toleration: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.0167$; KNN-Euclidean: $p(\alpha) = 0.011$ and $p(\alpha') = 0.05$ and Deletion: $p(\alpha) = 0.0076$ and $p(\alpha') = 0.05$); and 5) The performance of Fuzzy Analogy under MAR and NIM is not significantly different when using KNN-Manhattan ($p(\alpha) = 0.155$ and $p(\alpha') = 0.05$).

TABLE IV. COMPARISON OF P-VALUES OBTAINED IN THE WILCOXON TEST OF NH2.

	Toleration		KNN-Euclidean		KNN-Manhattan		Deletion	
	MAR p(α)/ p(α’)	NIM p(α)/ p(α’)	MAR p(α)/ p(α’)	NIM p(α)/ p(α’)	MAR p(α)/ p(α’)	NIM p(α)/ p(α’)	MAR p(α)/ p(α’)	NIM p(α)/ p(α’)
MCAR	0.011/0.05	0.0076/0.0250	0.0076/0.0167	0.0076/0.0250	0.017/0.0167	0.021/0.0250	0.0076/0.0167	0.0076/0.0250
MAR		0.0076/0.0167		0.011/0.0500		0.155/0.0500		0.0076/0.0500

To conclude on the hypotheses NH1 and NH2, from Tables 2-4, we notice that:

- Fuzzy Analogy with KNN techniques achieved significantly better Pred(0.25) over Fuzzy Analogy with toleration or deletion technique regardless of the mechanism of missingness.
- Fuzzy Analogy with toleration did not provide significant improvement over Fuzzy Analogy with deletion regardless of the mechanism of missingness.
- Fuzzy Analogy with MCAR achieved significantly better Pred(0.25) over Fuzzy Analogy with MAR when using toleration, KNN-Euclidean or deletion. However, when using KNN-Manhattan, the improvement provided by MCAR over MAR is not significant.
- Fuzzy Analogy with MCAR achieved significantly better Pred(0.25) over Fuzzy Analogy with NIM regardless of the MD technique used.
- Fuzzy Analogy with MAR achieved significantly better Pred(0.25) over Fuzzy Analogy with NIM when using toleration, KNN-Euclidean or deletion. However, when using KNN-Manhattan, the improvement provided by MAR over NIM is not significant.

Those observations are confirmed by the p-value of the Holm-Bonferroni correction, which gives us more confidence in the results obtained. So, at $\alpha = 0.05$ level of significance, there is enough evidence to conclude that the missingness mechanism (MD techniques respectively) significantly affects the accuracy of Fuzzy Analogy. Therefore NH1 and NH2 are rejected.

b) Comparison of Fuzzy Analogy accuracy in terms of Pred(0.25) and SA

Table 2 shows the performance of Fuzzy Analogy when using MD with Pred(0.25) and SA. The SA values are the result of our prior work [10]. Table 2 shows that the findings of the evaluation of Fuzzy Analogy using Pred(0.25) confirm the results of our study [10] which used SA. Form Table 2, we notice that the performance of Fuzzy Analogy compared with random guessing (SA values) as well as its accuracy (Pred(0.25) values) increase when : 1) KNN imputation is used instead of deletion or toleration, 2) The MD percentage is small or 3) the mechanism of missigness is MCAR rather than MAR or NIM.

However, even though the evaluation of Fuzzy Analogy using Pred(0.25) and SA gave the same results, we noticed that there were some differences. For example, from the study [10] and Table 2, we notice that Fuzzy Analogy using KNN yields acceptable values of SA even at high percentages of MD (at 60% MD under MCAR, SA = 52% for Fuzzy Analogy using KNN-Euclidean and SA = 51% for Fuzzy Analogy using KNN-Manhattan). From Table 2, we notice that its accuracy in terms of Pred(0.25) rapidly decreases as the MD percentage increases (at 60% MD under MCAR, Pred(0.25) = 40% for Fuzzy Analogy using KNN-Euclidean and Pred(0.25) = 42% for Fuzzy Analogy using KNN-Manhattan). Besides, when using toleration under NIM mechanism, we notice that even if Fuzzy Analogy gave acceptable values of SA, it yield low values of Pred(0.25) (at 10% MD under NIM, Pred(0.25) = 38% and SA= 54% for Fuzzy Analogy using toleration).

Those findings suggest that it is possible for Fuzzy Analogy to be reasonable (high values of SA) but this not necessary implies that it is accurate (high values of Pred(0.25)). Hence, we confirm the findings of Azzeh et al [11] claiming that SA alone is not sufficient to conclude about the technique accuracy. We conclude that SA and Pred(0.25) are complementary performance measures which may be used to evaluate SDEE techniques. SA measures to what extent the technique is reasonable while Pred(0.25) measures its estimation accuracy.

VII. CONCLUSION AND FUTURE WORK

This study investigated the impacts of MD techniques on the accuracy of Fuzzy Analogy. We evaluated its performance on seven data sets (in terms of Pred(0.25) values) when used in conjunction with three MD techniques (toleration, deletion and KNN imputation method), different missingness mechanisms (MCAR, MAR and NIM) and percentages of MD (from 10% to 90%). We investigated whether the results would confirm the findings of our prior work which used the SA measure. The results of accuracy measured in terms of Pred(0.25) confirm the findings of study [10] which used the SA measure. In fact, we conclude that Fuzzy analogy is more reasonable and more accurate when: 1) it uses imputation technique rather than toleration and deletion, 2) missigness mechanism is MCAR rather than MAR or NIM and 3) the MD percentage is small.

Moreover, this study claimed that Pred(0.25) and SA measure different aspects of SDEE technique performance. SA determines if a technique is reasonable (i.e., is actually predicting, and how much better is it than random guessing) while Pred(0.25) evaluates the accuracy of a technique (i.e., counts the number of cases where the technique gave values that were close to the actual effort). Hence, SA is not sufficient to conclude about the technique accuracy and it should be used with other metrics, especially Pred(0.25).

This study provides practical and substantiated guidelines for researchers and practitioners constructing effort estimation models when their data sets have MD. We encourage the replication of this study on alternative data sets. If further replications confirm our findings, it will be of important practical significance to researchers building analogy-based effort estimation techniques. Furthermore, future work ought to assess alternative imputation techniques, as these may provide even better performance than the ones we studied. Moreover, the results presented in this work are only preliminary, since we investigated only numerical data. Consequently, further investigation may be required on mixed data (numerical and categorical) to confirm our findings.

Aknowledgement

This work was conducted within the research project MPHR-PPR1-2015-2018. The authors would like to thank the Moroccan MESRSFC and CNRST for their support.

References

[1] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, 2012.

[2] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 820–827, 2012.

[3] M. Jorgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, 2007.

[4] K. Strike, K. El Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Trans. Softw. Eng.*, vol. 27, no. 10, pp. 890–908, 2001.

[5] P. Sentas and L. Angelis, "Categorical missing data imputation for software cost estimation by multinomial logistic regression," *J. Syst. Softw.*, vol. 79, no. 3, pp. 404–414, 2006.

[6] A. Idri, F. A. Amazal, and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," *Inf. Softw. Technol.*, vol. 58, pp. 206–230, 2014.

[7] F. A. Amazal, A. Idri, and A. Abran, "An Analogy-based Approach to Estimation of Software Development Effort Using Categorical Data," in *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 2014, pp. 252–262.

[8] M. Azzeh, D. Neagu, and P. I. Cowling, "Analogy-based software effort estimation using Fuzzy numbers," *J. Syst. Softw.*, vol. 84, no. 2, pp. 270–284, 2011.

[9] A. Idri, I. Abnane, and A. Abran, "Systematic mapping study of missing values techniques in software engineering data," in *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS*, 2015, pp. 1 – 8.

[10] A. Idri, I. Abnane, and A. Abran, "Missing data techniques in analogy-based software development effort estimation," *J. Syst. Softw.*, vol. 117, pp. 595–611, 2016.

[11] M. Azzeh and A. B. Nassif, "A hybrid model for estimating software project effort from Use Case Points," *Appl. Soft Comput. J.*, pp. 1–9, 2016.

[12] L. L. Minku and X. Yao, "Can Cross-company Data Improve Performance in Software Effort Estimation?," *Proc. 8th Int. Conf. Predict. Model. Softw. Eng.*, pp. 69–78, 2012.

[13] E. Kocaguneli and T. Menzies, "Transfer Learning in Effort Estimation," *Empir. Softw. Eng.*, vol. 20, no. 3, pp. 813–843, 2015.

[14] M. Korte and D. Port, "Confidence in software cost estimation results based on MMRE and PRED," *Proc. 4th Int. Work. Predict. Model. Softw. Eng. - PROMISE '08*, pp. 63–70, 2008.

[15] R. J. A. Little and D. . Rubin, "Statistical Analysis with Missing Data," *Wiley, New York.*, 1987.

[16] K. El Emam and A. Birk, "Validating the ISO/IEC 15504 measures of software development process capability," *J. Syst. Softw.*, vol. 51, no. 2, pp. 119–149, 2000.

[17] K. El Emam and A. Birk, "Validating the ISO/IEC 15504 measure of software requirements analysis process capability," *IEEE Trans. Softw. Eng.*, vol. 26, no. 6, pp. 541–566, 2000.

[18] Q. Liu, W. Qian, and A. Atanas, "Application of missing data approaches in software testing research," *2011 Int. Conf. Electron. Commun. Control. ICECC 2011 - Proc.*, no. November 2009, pp. 4187–4191, 2011.

[19] J. Li, A. Al-Emran, and G. Ruhe, "Impact Analysis of Missing Values on the Prediction Accuracy of Analogy-based Software Effort Estimation Method AQUA," *First Int. Symp. Empir. Softw. Eng. Meas. (ESEM 2007)*, pp. 126–135, 2007.

[20] Q. Song, M. Shepperd, X. Chen, and J. Liu, "Can k-NN imputation improve the performance of C4.5 with small software project data sets? A comparative evaluation," *J. Syst. Softw.*, vol. 81, no. 12, pp. 2361–2370, 2008.

[21] M. H. Cartwright, M. J. Shepperd, and Q. Song, "Dealing with missing software project data," in *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717)*, 2003.

[22] B. Twala, M. Cartwright, and M. Shepperd, "Ensemble of Missing Data Techniques to Improve Software Prediction Accuracy," *Proc. 28th Int. Conf. Softw. Eng.*, pp. 909–912, 2006.

[23] S. Yenduri, "an Empirical Study of Imputation Techniques for Software Data Sets," Louisiana State, 2005.

[24] J. Keung, "Software Development Cost Estimation Using Analogy: A Review," *2009 Aust. Softw. Eng. Conf.*, 2009.

[25] F. A. Amazal, A. Idri, and A. Abran, "Software Development Effort Estimation Using Classical and Fuzzy Analogy: a Cross-Validation Comparative Study," *Int. J. Comput. Intell. Appl.*, vol. 13, no. 03, pp. 1-19, 2014.

[26] A. Idri, A. Abran, and S. Mbarki, "An Experiment on the Design of Radial Basis Function Neural Networks for Software Cost Estimation," *2nd Int. Conf. Inf. Commun. Technol.*, vol. 1, pp. 1612–1617, 2006.

[27] J. C. Bezdek, "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 1, pp. 1–8, 1980.

[28] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, 1991.

[29] A. Idri, A. Zahi, and A. Abran, "Generating Fuzzy Term Sets for Software Project Attributes using Fuzzy C-Means and Real Coded Genetic Algorithms," in *Information and Communication Technology in Muslim World*, 2006, no. 2.

[30] A. Idri and A. Abran, "A fuzzy logic based set of measures for software project similarity: validation and possible improvements," *Proc. Seventh Int. Softw. Metrics Symp.*, 2001.

[31] A. Bakır, B. Turhan, and A. B. Bener, "A new perspective on data homogeneity in software cost estimation: a study in the embedded systems domain," *Softw. Qual. J.*, vol. 18, no. 1, pp. 57–80, 2010.

[32] C. Lokan, T. Wright, P. Hill, and M. Stringer, "Organizational benchmarking using the ISBSG Data Repository," *Software, IEEE*, vol. 18, no. 5, pp. 26 –32, 2001.

[33] T. Menzies, B. Caglayan, and E. et al. Kocaguneli, "The promise repository of empirical software engineering data," 2012. .

[34] A. Idri, A. Abran, and S. Robert, "Investigating Soft Computing in Case-Based Reasoning for Software Cost Estimation," *Int. J. Eng. Intell. Syst.*, pp. 1–18, 2002.

[35] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[36] D. Sheskin, *Handbook of Parametric and Non-parametric Procedures*. CRC Press, 1997.

[37] H. Abdi, "1 Overview 2 Preliminary : The different meanings of alpha," *Encycl. Res. Des.*, pp. 1–8, 2010.