

Missing values imputation in Arabic datasets using enhanced robust association rules

Awsan Salem¹, Nurul Akmar Emran¹, Azah Kamilah Muda¹, Zahriah Sahri¹, Abdulrazzak Ali²

¹Computational Intelligence Technologies (CIT) Research Group, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

²Faculty of Computer and Information technology, Aden University, Aden, Yemen

Article Info

Article history:

Received Mar 17, 2022

Revised Aug 17, 2022

Accepted Sep 6, 2022

Keywords:

Arabic dataset
Association rules
Data Imputation
Missing values
Morphology

ABSTRACT

Missing value (MV) is one form of data completeness problem in massive datasets. To deal with missing values, data imputation methods were proposed with the aim to improve the completeness of the datasets concerned. Data imputation's accuracy is a common indicator of a data imputation technique's efficiency. However, the efficiency of data imputation can be affected by the nature of the language in which the dataset is written. To overcome this problem, it is necessary to normalize the data, especially in non-Latin languages such as the Arabic language. This paper proposes a method that will address the challenge inherent in Arabic datasets by extending the enhanced robust association rules (ERAR) method with Arabic detection and correction functions. Iterative and Decision Tree methods were used to evaluate the proposed method in an experiment. Experiment results show that the proposed method offers a higher data imputation accuracy than the Iterative and decision tree methods.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Nurul Akmar Emran

Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka

Ayer Keroh, 76100 Melaka, Malaysia

Email: nurulakmar@utem.edu.my

1. INTRODUCTION

Data is usually dirty at its source. "Dirty" from data quality context means the data have impurities such as duplication, misspellings, and missing values (MV). The ratio of impurities in datasets varies, and factors such as failures of monitoring, a fault in data input process, equipment errors, disruption of communication between data collectors and the central management system, failure during the archiving system (hardware or software), or human errors contribute to the problem [1].

Dirty data must be cleaned before it can be useful in decision-making or analysis for an organization. In fact, the quality of the analysis is determined by the quality of data [2]. Missing values is an example of a data completeness problem that causes dirty data. The missing values have become a rising concern for many sectors such as business, industry, healthcare, and e-governance [3]. Missing values occur when no data values are stored for the variable in an observation [2]. Missing values is a typical problem in many applications, and the failure to deal with them can significantly affect the results drawn from the data. Missing values come in different patterns, for example, Univariate, Montana, and Arbitrary, that are caused by several mechanisms such as missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR) [4]. Most methods that deal with missing values usually seek to understand the nature of missing values before proposing a solution.

Researchers adopt at least two ways of dealing with missing values. The first is the "deletion technique" that will ignore the missing values present in a dataset. This approach is effective in the case

where the number of missing values is small, and the deletion will not cause bias in the datasets. Data imputation is another way to deal with missing values. The missing values are replaced by estimated values based on the information available in the dataset [5], [6].

The selection of the imputation method is always based on the missing values mechanism and the pattern of missing values (MV) [7]. The problem with these methods is that some may perform well in certain data types while others may not [5], [8]. According to Allison, no single imputation method handle all missing value types [9]. According to Suthar *et al.* [2], imputation methods are classified into two categories. The first category is single imputation, which refers to substituting a single value for each missing value. The methods under single imputation are mean imputation, regression imputation, hot-deck imputation, and advanced imputation methods such as (expectation maximisation (EM) approach and raw maximum likelihood (RML) [5], [8]. The second category is multiple imputations. Multiple imputations create several copies of the dataset, each containing different imputed values, and these datasets are subsequently combined into a single set of results. Association rules (AR) are popular methods adopted by many data imputation methods.

The early use of AR can be seen in data mining for finding interesting relations between variables in large databases [10]. Ragel, proposed a new approach to mine AR in relational databases containing missing values where the tuples with missing attributes were partially disabled instead of deleting it to ease the impact of lost rules [11]. In 1999 Ragel and Cremilleux proposed missing values completion (MVC) using AR [12]. Based on MVC methods, several algorithms were developed, such as recycle combined association rules (RCAR) [13], fast recycle combined association rules (FRCAR) [14], association rule mining from data with missing values (ARDM) [15], Iterative missing-value completion [16], and enhanced robust association rules (ERAR) [17]. AR is a concept based on complete matching between values during the process of computing frequent itemsets. Accurate matching in computing frequent items will fail with nominal datasets due to misspellings and typo errors [18].

Most of the imputation methods focus on English, which deals with similar challenges with other Latina and non-Latina languages, such as letters, rules, and neutral morphology. Nevertheless, the non-Latina language, such as the Arabic language, has additional challenges that researchers need to deal with. The Arabic language is based on 28 letters and short vowel signs known as diacritics which are positioned either above or below the letter. This mark indicates the phonetic information associated with each letter helps explain the context and sense of the word [19].

Inconsistent variations are an issue that needs to be handled in Arabic text. Different versions of Alif (أ, إ, ا, آ), for example, may be written interchangeably; another example is alif maqsurah and normal dotted Yaa' (ي, ي), which are commonly used interchangeably at the word's end. The same may be said for Taa-marbutah and Haa-marbutah (ة, هـ) such as (المعلمة, المعلمة) which means (Teacher). The two words come in two other forms, but it refers to one meaning. As a result, accurate matching is complex unless the values in the dataset can be unified in one form. Therefore, it is crucial to address the Arabic language issues and correct the misspelled before we can produce accurate frequent itemsets for imputation.

There are several techniques used to handle Arabic issues. typically involves two primary modules: detecting errors within a text and correcting those errors [20]. The simplest approach is the dictionary lookup technique, where the input word is compared with the words in the dictionary. If the input word is not found in the dictionary, the word is considered an incorrect word. Another approach is morphological analyzers that check whether a word is following the morphological language rules or not. These two methods are usually combined for better results. Error correction is complex, especially for a linguistically rich language such as Arabic due to morphology complexity [21]. There are some attempts at spelling correction for the Arabic language. The following are the latest developments in the work:

- i) Hamza *et al.*, created a separate spell-checking corpus with ill-formed words identified as a result of morphological analysis failure [22]. Although they use a stemming dictionary to shorten many Arabic words, the disadvantage of this approach is that as small lexical is used, many types of errors are not being covered.
- ii) Al-hagree *et al.*, proposed an algorithm that tests the accuracy and efficiency of Arabic name matching. The algorithm focuses on the unique qualities of the Arabic language and the multiple degrees of correspondence between Arabic letters, including keyboard similarities, letter shapes, and phonetic similarities [23]. Moreover, the suggested approach uses the Damerau-Levenshtein Distance algorithm to account for the transposition operation and the improved states of substitution, deletion, insertion, and transposition operations. The authors did not handle Arabic morphology's complexity and did not use the Arabic lexical. The technique depends on Levenshtein Distance algorithm and pattern to select the correct word, which is prone to the possibility of errors.
- iii) Alkhatib *et al.* proposed a system for detecting and correcting spelling errors [20]. They developed a systematic framework for spelling and grammar error detection and a correction at the word level, based

on a bidirectional long short-term memory mechanism and word embedding, in which a polynomial network classifier is at the top of the system. The experiment compared results with the output of two well-known tools: Ayaspell version 3.47 and Microsoft office 2013. Their system showed higher accuracy than (93.89%) compared with the two other tools. Nevertheless, they did not mention how to handle Arabic misspelling.

- iv) Atawy and Ahmed proposed a spelling checker (DYS-EnSC) which is based on the n-gram method, a lookup dictionary, and Damerau-Levenshtein. The detection and correction of misspellings made by Arabic second language learners with dyslexia are handled by creating a list of candidates and selecting the best appropriate candidate for each misspelled word. The system achieved 93% accuracy in detecting misspellings and corrected about 86% of words, and outperformed Microsoft Word (MW) and spell and language tools [24]. However, the system could not discover some errors such as synonyms for the wrong term, such as (home) instead of (house). Another issue with this method is that adding or removing a character result in an identical word for a word in the dictionary. Thus, the system recognized it as an error in some circumstances. The last issue is that the system did not use morphology analysis to handle the complexity of Arabic morphology.

All the methods presented deal with the Arabic language in different areas, such as analysis and imputation. Nevertheless, improvement is needed to deal with missing values in the Arabic datasets in terms of imputation. Imputation methods are limited in handling incomplete datasets in the Arabic language, especially in considering the complexity of morphology issues and the correction process in Arabic texts. With these limitations, further work is needed in order to evaluate the accuracy of missing values imputation within Arabic datasets.

In this paper, we present an enhanced method to impute the missing values in the Arabic dataset based on combining the enhanced robust association rules (ERAR) and Arabic language detection and correction techniques inspired by earlier works. The ERAR method uses the Arabic correction and detection techniques such as dictionary lookup and morphology analysis on Arabic datasets to correct the misspelling and unify the words in one form before selecting the frequent itemsets. In this way, high accuracy of frequent itemsets can be generated to calculate values with the same meaning and derive the rules for the imputation. The remainder of this paper is organized as follows. The proposed method will be presented in Section 2, while the results and discussion in Section 3. Finally, the conclusions are in Section 4.

2. METHOD

In this section, we focus on the data preparation stage to improve the efficiency of the ERAR method to impute missing values in the Arabic datasets. ERAR method was inspired by an Iterative algorithm [16]. ERAR method consists of five main steps: i) filter the candidates, ii) select the frequent itemsets, iii) generate the association rules, iv) fill in the missing values, and finally v) check the dataset. In fact, ERAR relies on simple data preparation processes that are inadequate for addressing the Arabic datasets issues. Therefore, an extension of the data preparation process by adding the Arabic language detection and correction functions is needed to overcome this problem.

2.1. Arabic dataset preparation

Arabic dataset preparation focuses on misspelling and morphology aspects that become the major concerns in the Arabic language. The proposed workflow for the Arabic language dataset preparation is shown in Figure 1. The module of Arabic dataset preparation is divided into three parts: normalization, error detection, and error correction.

2.1.1. Normalization

In this step, inconsistent variations are handled in raw Arabic text using various natural language processing (NLP) applications as the following: i) different forms of letter alif (ا , آ , إ , إ) are unified in one form which is aliff without Hamza or diacritical marks (ا); ii) change the taa marbutah and haa marbutah (ة , هـ) to one form which is (هـ); iii) remove all three forms of diacritics marks to make all words in one form. These steps attempt to unify all the letters into one form, which improves the matching process.

2.1.2. Error detection

In this step, misspelled words are detected. Many techniques, such as dictionary search and morphology analysis, are used to detect errors in Arabic languages [25], [26]. Dictionary lookup is the most common and the fastest method due to the size of the dictionary (corpus). The morphology process takes more time to complete due to the complexity of morphology in Arabic. In this study, we used both techniques, which are dictionary lookup and morphology rules. The following are the description of the processes:

- i) Dictionary: dictionary lookup technique is the most extensively used misspelling detection methodology in many applications. In this process, the input word is compared to the words within the dictionary. If the input word is not found in the dictionary, the word is considered incorrect. Unlike morphological analysis, this method ensures that the most often used words are covered [26]. The size of our proposed lexicon in this model is around 9 million unique Arabic words taken from Shaalan *et al.* [26].
- ii) Morphology analysis: in this process, the morphological generator is used to generate a sufficient list of potential words [26], [27]. As there is no corpus, the list will contain all possible word forms that reflect the richness and complexity of Arabic morphology. The morphological rules apply using the Light stemming methodology to cover words included in a generated dictionary [28]. The prefix added to begins of the root to generate words are as follow:
 (وا,ون,وه,ان,ات,أي,كم,ون,فن,بن,ي,بن,ب,ين,يه,ه,ي,ول,فل,فا,فلل,ولل,است,ست, وال,وكال,وقال,وبال,فبال,وت,فت,ون,فن,كت,م,وم,فم,بم,لم,ولم,فلم,وكم,وكالم,فكالم,وللم,ولم,فللم,وللم,وبالم,فبالم,وا,فاب,وب,وف,ب,ل.)
 and the suffix adding to the ends of root to generate a word as follow:
 (ك,ت,هم,تهما,تكنأ,تكمأ,ا,ي,ه,يه,ين,نا,تك,ي,همها,هن,كم,تم,ته,تي,ان,وه,ون,وات)
- iii) Selection of frequent words: this step is performed after the error detection stage. This step calculates the frequency of each incorrect word. Based on the frequency value, we selected the incorrect word as the correct word if the word had a frequency of more than ten times. Figure 2 shows some words in the Poems dataset not found in dictionary and morphology analysis. The words selected based on iterate are often adjectives or names.

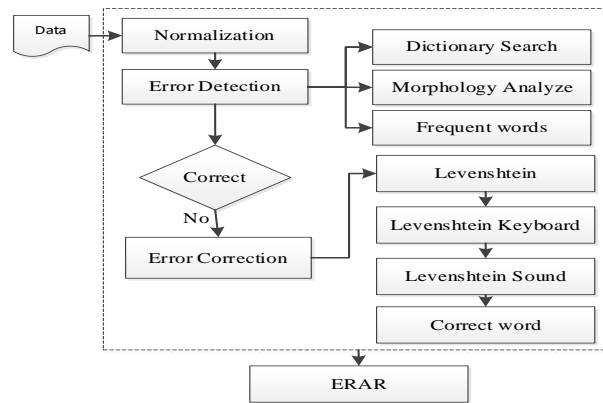


Figure 1. Arabic dataset preparation modules

NO	Word	Number of Iterate
1	الباخرزي (albakhirzi)	205
2	البلتسي (albilinsu)	147
3	الابوردي (alabyurdi)	384
4	كشاجم (kashajim)	360
5	الماغوط (almaghut)	112

Figure 2. Example of frequent words in dataset

2.1.3. Error correction

Correction of spelling errors is one of the most important areas of NLP. The majority of spelling errors are caused by a misplaced letter, an additional letter, or a single transposition between characters. Four operations are defined to rectify these errors: substitution, insertion, deletion, and transposition. This study used the Damerau Levenshtein technique to handle these types of errors as the following:

- i) Damerau Levenshtein algorithm: this algorithm generates the edit-distance metric, which was first introduced in 1974 by Wagner and Fischer [29].
- ii) Keyboard related: some spelling errors are the result of pressing the wrong keys of the keyboard. The majority of spelling errors are caused by pressing the erroneous or adjacent key on the keyboard. Figure 3 shows The Arabic letters and neighbor letters on the Arabic keyboard.
- iii) Sound Similarity: Some of the letters in Arabic has the same sound as the English language. We used the edit-distance method to deal with this issue by grouping similarly sounding letters as one letter.

Figure 4 shows the letters that have similar sounds in Arabic. The sound similarity technique can be used to rank suggested words that are selected from the Damerau Levenshtein algorithm in the previous steps.

- iv) Selection of correct word: After completing the previous three steps, we will have a list of potential words for each incorrect word. The selection of the correct word will be performed by replacing each potential word in a list and counting the iteration of values found after replacing the incorrect word. Hence, one of these potential words will show iteration value while the rest of the words will show zero iteration. Therefore, the potential word that has iteration value will be the replaceable word instead of the incorrect word. Figure 5 shows an example on how to select the correct word from a list of potential words that were taken from the poem dataset.

No	Arab. Letter	Neighbor keys	No	Arab. Letter	Neighbor keys
1	ا	ل, ر, ع, ش, ي	15	ض	ذ, ص, ش
2	ب	ي, ق, ف, ل, ر, و, ن	16	ط	ك, ج, د, ظ
3	ت	ا, و, ع, ه, ن, ق, ي	17	ظ	ز, ك, ط
4	ث	ص, ق, ي, س	18	ع	غ, ه, ت, ا
5	ج	ح, د, ط, ك	19	غ	ف, ع, ا, ل
6	خ	ه, ح, م, ن	20	ف	ق, غ, ل, ب
7	ح	خ, ج, ك, م	21	ق	ش, ف, ب, ي
8	د	ط	22	ك	م, ح, ج, ط, ظ, ز
9	ذ	ض	23	ل	ب, ف, غ, ا, ل, ر
10	ر	و, ب, ل	24	م	ن, خ, ح, ك, ز, و
11	ز	و, م, ك, ظ	25	ن	ت, ه, خ, م, و, ة
12	س	ش, ص, ث, ي, ع, ا	26	ه	ع, خ, ن, ت
13	ش	ض, ص, س, ع, ا	27	و	ق, ن, م, ز
14	ص	ض, ث, س, ش	28	ي	س, ث, ق, ب, و, ة

Figure 3. Arabic keyboard letters

No	Letters	Correct Arabic Word	English meaning	Sound Error Example
1	ا, و, ا, ي	(awahaa) اوحى	inspire	(awha) اوحا
2	ذ, ز	(zakah) زكاه	zakat	(dhakah) ذكاه
3	ض, ظ	(zalim) ظالم	unjust	(dalam) ضالم
4	س, ص	(mustaruh) مسطره	Ruler	(musataruh) مصطره
5	ي, ي	(muathiruh) مؤثره	impressive	(muthiruh) مؤثره

Figure 4. Similarity sound Arabic letters

Incorrect word	value	Potential words	(Damerau, Keyboard, Sound)	Frequent value
هانظي (hazi)	(ايليا ابو هانظي) (aylya abu hazi)	باهي (bahi) → (good)	2	0
		صافي (safi) → (clear)	2	0
		شادي (shadi) → (Shady) noun	2	0
		ناجي (naji) → (Survivor) noun	1	0
		سامي (sami) → (Sami) noun	2	0
		ماضي (madi) → (Past)	1	287

Figure 5. An example of the correct word selection from the potential words

2.2. Evaluation of the proposed method

In this study, we used two real datasets that are available from the Kaggle repository. The first one is the Arabic Poetry dataset. This dataset consists of six attributes and 58,000 records. Several types of Arabic problems such as spelling mistakes and diacritical marks are added to simulate the problems of the Arabic dataset. The second dataset is an English dataset which is the Zomato Restaurant dataset has 9,551

restaurants and 21 attributes. An arbitrary pattern is used to add the missing values within the 10%-70% range.

The Iterative and decision trees were used as benchmarks to evaluate ERAR's performance. We selected the Iterative method because ERAR's proposal was inspired by the Iterative method's paradigm and thus, comparing them is necessary. At the same time, the selection of the DT method as a benchmark is due to its similar nature to capture the relations between attributes and handle nominal values [30], involving those that contain numerical and nominal data commonly used in imputation experiments [31]. The confusion matrix is used to compute the performance measures, and the accuracy is determined by calculating the F-Score.

3. RESULTS AND DISCUSSION

This section presents the results obtained from implementing ERAR, Iterative, and DT methods on the dataset under study. Table 1 shows the confusion matrix that consists of true positive (TP), false positive (FP), and true negative (TN) discovered for three methods in different missing values (MV) rates, support value= 0.6 and Confidence= 0.7. Overall, ERAR outperforms Iterative and DT methods with the highest TP scores that can be seen for all MV rates.

Table 1. Experiments results on arabic dataset

MV Rate	ERAR			Iterative			DT		
	TP	FP	TN	TP	FP	TN	TP	FP	TN
10%	13329	11	19955	10760	12854	9681	6239	3	27053
20%	26055	44	43022	21720	21397	26004	11587	18	57516
30%	34733	76	65313	30686	43268	26168	12609	2803	84710
40%	40543	91	95660	39779	47985	48530	6079	380	129835
50%	40891	114	132144	42851	59450	70848	30	2912	170207
60%	34273	91	174395	37611	44826	126322	369	9077	199313
70%	21312	175	221303	24162	19279	199349	1393	4050	237347

Figure 6 illustrates the imputation accuracy using the Arabic Poetry dataset by the three methods under study within a range of missing values (10%-70%). The ERAR method consistently exhibits the highest accuracy as compared to other methods. The Iterative method shows lower accuracy than ERAR, with quite a consistent trend throughout the MV rates. The DT method initially performs very well for MV between 10 to 20%, but the accuracy suddenly drops after 30% of MV. The lowest accuracy for DT is at 50% of the MV rate. The accuracy of DT is expected to degrade as the number of missing values increases.

As it is worth discovering whether ERAR does not only behaves well in imputing missing values within the Arab dataset, it is necessary to examine the behavior of ERAR in dealing with the non-Arabic dataset. Thus, the same procedure was implemented in the experiment against an English dataset called Zomato restaurants. Table 2 shows the results of implementing ERAR, Iterative, and DT methods on the Zomato dataset.

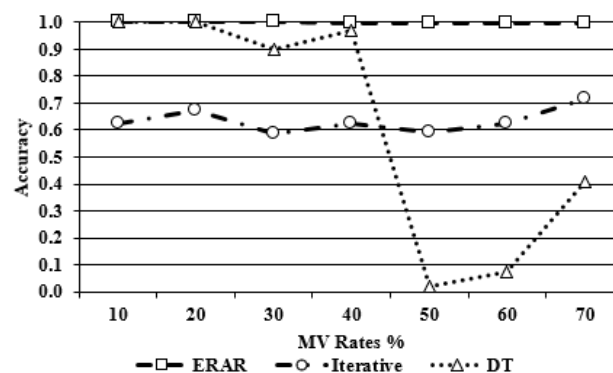


Figure 6. MV imputation accuracy for three methods using Arabic poetry dataset

Table 2. Implementation result on the non-Arabic dataset

MV Rate	ERAR			Iterative			DT		
	TP	FP	TN	TP	FP	TN	TP	FP	TN
10%	6901	1779	8057	4593	11669	475	5169	337	11231
20%	12873	3867	15552	11051	20805	436	8416	741	23135
30%	17754	5611	22707	17717	27511	844	33	26	46013
40%	22780	9240	31790	23495	38186	2129	50	38	63722
50%	25970	12838	39736	25620	50834	2090	53	44	78447
60%	27746	11567	58539	31903	54074	11875	33	26	97793
70%	27262	13188	71783	32474	52150	27609	34	25	112174

Figure 7 shows the imputation accuracy of ERAR, Iterative, and DT using the Zomato dataset for a range of MV rates. The results show that DT performed best within a low range of missing values (between 10-20) as compared to ERAR and Iterative methods. ERAR showed better performance of all from 20% of MV. Based on the experiment, we found several limitations of ERAR. The first limitation is that ERAR is not good in dealing with numerical data. AR used complete matching to generate the rules, and the rules were used in replacing the MV. Unfortunately, this process fails with numbers. For example, the aggregate rating column in the Zomato dataset that consists of decimal numbers is related to a column such as rating color. As such, the values within the range 4.5-5 refer to dark green in the rating color column. The second limitation of ERAR is in dealing with sequence values. For example, in the address column in the Zomato dataset, we can find the same value in different sequences such as "Vikasपुरi-New Delhi" and "New Delhi-Vikasपुरi" (both refer to the same address). Thus, a data cleaning step is necessary to standardize the sequence values.

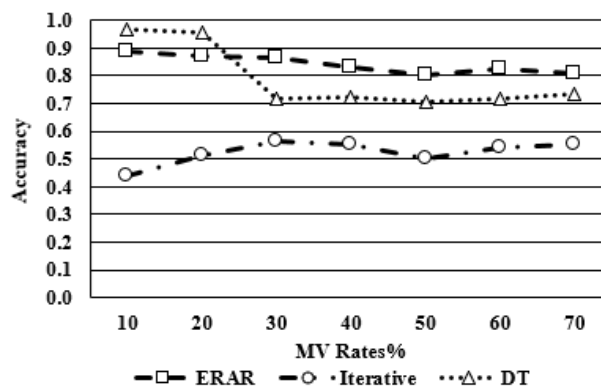


Figure 7. MV imputation accuracy for three methods using Zomato dataset

4. CONCLUSION

In conclusion, we investigated the missing values problem within Arabic datasets in this paper. We proposed a data imputation method that extends the ERAR method with the Arabic language preparation function. Arabic language preparation is introduced to handle Arabic language issues such as misspelling, letters forms, diacritical marks, and morphology. In particular, the Arabic preparation function reduces the impact of Arabic language issues in selecting the frequent itemsets. The aim of the method is to provide data imputation accuracy regardless of the challenges inherent in Arabic datasets. An experiment was conducted to evaluate ERAR's data imputation accuracy for the Arabic dataset by comparing it to the Iterative and DT methods. The results show that ERAR behaves better than Iterative and DT despite increasing missing values. The Arabic preparation function has successfully enabled the ERAR to achieve the highest number of true positive and the lowest false positive in most missing values rates compared to the iterative and DT methods. Similar results where ERAR outperformed the other two methods can also be observed for the English dataset. The findings presented in this paper contribute to understanding the implications of treating the problems inherent in a language for missing values imputation. It is worth studying how data imputation can be improved by addressing other types of spelling problems, such as hidden semantic errors. Other performance indicators such as the speed of imputation can also be considered in our future work.

ACKNOWLEDGEMENTS

We would like to thank the Fakulti Teknologi Maklumat Dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM) for supporting this research. This work is under research grant PJP/2020/FTMK/PP/S01772.




REFERENCES

- [1] A. Ali, N. A. Emran, S. A. Asmai, and A. R. Ismail, "An assessment of open data sets completeness," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, pp. 557-562, 2019, doi: 10.14569/ijacsa.2019.0100672.
- [2] B. Suthar, H. Patel, and A. Goswami, "A survey: classification of imputation methods in data mining," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 1, pp. 309-312, 2012.
- [3] N. A. Zainuri, A. A. Jemain, and N. Muda, "A comparison of various imputation methods for missing values in air quality data," *Sains Malaysiana*, vol. 44, no. 3, pp.449-456, 2015.
- [4] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, Third. New York: Wiley Series in Probability and Statistics, 2019.
- [5] B. Twala, M. Cartwright, and M. Shepperd, "Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases," *International Symposium on Empirical Software Engineering, 2005*, 2005, pp. 105-114, doi: 10.1109/ISESE.2005.1541819.
- [6] A. Ali, N. A. Emran, and S. A. Asmai, "Missing values compensation in duplicates detection using hot deck method," *Journal of Big Data*, vol. 8, no. 1, p. 112, 2021, doi: 10.1186/s40537-021-00502-1.
- [7] J. L. Peugh and C. K. Enders, "Missing data in educational research: a review of reporting practices and suggestions for improvement," *Review of Educational Research*, vol. 74, no. 4, pp. 525-556, 2004, doi: 10.3102/00346543074004525.
- [8] Md. G. Rahman and Md. Z. Islam, "A decision tree-based missing value imputation technique for data pre-processing," *Proceedings of the Ninth Australasian Data Mining Conference*, 2010, pp. 41-50.
- [9] P. D. Allison, *Missing data*, SAGE Publications, Inc, 2001.
- [10] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp.487-499.
- [11] A. Ragel, "Preprocessing of missing values using robust association rules," *European Symposium on Principles of Data Mining and Knowledge Discovery*, 1998, pp. 414-422, doi: 10.1007/BFb0094845.
- [12] A. Ragel and B. Cremilleux, "MVC — a preprocessing method to deal with missing values," in *Research and Development in Expert Systems XV.*, 1999, pp. 159-170, doi: 10.1007/978-1-4471-0835-1_11.
- [13] J. Shen and M.-T. Chen, "A recycle technique of association rule for missing value completion," *17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003*, 2003, pp. 526-529, doi: 10.1109/AINA.2003.1192936.
- [14] J. J. Shen, C. C. Chang, and Y. C. Li, "Combined association rules for dealing with missing values," *Journal of Information Science*, vol. 33, no. 4, pp. 468-480, 2007, doi: 10.1177/0165551506075329.
- [15] K. Rameshkumar, "A novel algorithm for association rule mining from data with incomplete and missing values," *ICTACT Journal on Soft Computing*, vol. 1, no. 4, 2011, doi: 10.21917/ijsc.2011.0026.
- [16] T.-P. Hong and C.-W. Wu, "Mining rules from an incomplete dataset with a high missing rate," *Expert Systems with Applications*, vol. 38, no. 4, pp.3931-3936, 2011, doi: 10.1016/j.eswa.2010.09.054.
- [17] A. Thabet, N. A. Emran, and A. K. Muda, "Enhanced robust association rules (ERAR) method for missing values imputation," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 6036-6042, 2020, doi: 10.30534/ijatcse/2020/272942020.
- [18] M. Priyanka and A. Baby, "A survey on various duplicate detection methods," *International Journal of Computer Science and Information Technologies*, vol. 8, no. 1, pp. 7-9, 2017.
- [19] R. Alnefaie and A. M. Azmi, "Automatic Minimal Diacritization of Arabic Texts," *Procedia Computer Science*, vol. 117, pp. 169-174, 2017, doi: 10.1016/j.procs.2017.10.106.
- [20] M. Alkhatib, A. A. Monem, and K. Shaalan, "Deep learning for Arabic error detection and correction," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 19, no. 5, pp. 1-13, 2020, doi: 10.1145/3373266.
- [21] H. Elzayady, M. S. Mohamed, K. M. Badran, and G. I. Salama, "Detecting Arabic textual threats in social media using artificial intelligence: An overview," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1712-1722, 2022, doi: 10.11591/ijeecs.v25.i3.pp1712-1722.
- [22] B. Hamza, Y. Abdellah, G. Hicham, and B. Mostafa, "For an independent spell-checking system from the Arabic language vocabulary," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 113-116, 2014, doi: 10.14569/ijacsa.2014.050115.
- [23] S. Al-hagree, M. Al-sanabani, K. M. A. Alalayah, and M. Hadwan, "Designing an accurate and efficient algorithm for matching Arabic names," *First International Conference of Intelligent Computing and Engineering*, 2019, pp. 1-12, doi: 10.1109/ICOICE48418.2019.9035184.
- [24] S. M. El Atawy and H. M. M. Ahmed, "Spelling Checker for Dyslexic Second Language Arab Learners," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 2, pp. 390-402, 2021.
- [25] M. A. Al-Hagery, L. A. Alreshoodi, M. A. Almutairi, S. I. Alsharekh, and E. S. Alkhowaiter, "A hybrid technique for cleaning missing and misspelling Arabic data in data warehouse," *International Journal of Information Technology and Computer Science*, vol. 11, no. 7, pp. 17-25, 2019, doi: 10.5815/ijitcs.2019.07.03.
- [26] K. Shaalan, Y. Samih, M. Attia, P. Pecina, and J. van Genabith, "Arabic word generation and modelling for spell checking," *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, 2012, pp. 719-725.
- [27] A. Chennoufi and A. Mazroui, "Morphological, syntactic and diacritics rules for automatic diacritization of Arabic sentences," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 2, pp. 156-163, 2017, doi: 10.1016/j.jksuci.2016.06.004.
- [28] S. Kheder, D. Sayed, and A. Hanafy, "Arabic light stemmer for better search accuracy," *International Journal of Cognitive and Language Sciences*, vol. 10, no. 11, pp. 3587-3595, 2016.
- [29] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168-173, 1974, doi: 10.1145/321796.321811.




- [30] B. Mouaz, C. Walid, B. H. Abderrahim, and E. Abdelmajid, "A new framework based on KNN and DT for speech identification through emphatic letters in Moroccan dialect," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1417-1423, 2021, doi: 10.11591/ijeecs.v21.i3.pp1417-1423.
- [31] J. You, X. Ma, D. Y. Ding, M. Kochenderfer, and J. Leskovec, "Handling missing data with graph representation learning," Oct. 2020, [Online]. Available: <http://arxiv.org/abs/2010.16418>.

BIOGRAPHIES OF AUTHORS






Mr. Awsan Salem    he holds a bachelor's degree from AL Balqa Applied University in Jordan and a master's degree from the Universiti Teknologi MARA (UiTM), Malaysia. He worked as a computer instructor at the College of Computer and Information Technology, the University of Aden from 2009 to 2017. Currently, he is a PhD candidate in computer science at Information Technology from the Universiti Teknikal Malaysia Melaka. His research interests include improving data quality by developing a new imputation method that meets high accuracy standards and overcoming the problem of non-Latin languages in the imputation approach. He can be contacted at e-mail: awsanthabet666@gmail.com.






Associate Professor Dr. Nurul Akmar Binti Emran    is an associate professor at the Universiti Teknikal Malaysia Melaka (UTeM). She received a bachelor's degree in Management Information System (MIS) from the International Islamic University Malaysia in 2001, an MSc in Internet and Database Systems from London South Bank University in 2003, and a Ph.D. degree in computer science from the University of Manchester, the UK in 2011. She begins her career in academia in 2002, as a tutor at the Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM). In 2004, she was appointed as a lecturer; in 2011, she was a senior lecturer before being promoted to Associate Professor in 2017. Her research interests include data quality (data completeness), database systems and security, storage space optimization, mobile analytics, and green computing. She can be contacted at email: nurulakmar@utem.deu.my.






Professor Dr. Azah Kamilah Muda    is a professor at Faculty of Information and Communication Technology (FTMK), Universiti Teknikal Malaysia Melaka (UTeM). She received her degree and master in Computer Science (Software Engineering) and her Ph.D in Computer Science from Universiti Teknologi Malaysia. Her research interests include soft computing, pattern recognition, image processing, machine learning, computational intelligence and hybrid systems. Dr Azah also serves as Editorial Boards of various international journal besides involved in organizing international conference as aorganizing chair, program committee chair, publication chair and reviewer for SoCPaR, HIS, ISDA, IAS, CaSON, NWeSP, WICT, CSNT, Graphite etc. She is a member of Technical Committee of Soft Computing for IEEE Systems, Man and Cybermatic Society. She can be contacted at e-mail: azah@utem.edu.my.



Dr. Zahriah Binti Sahri    she received the Diploma in Computer Science from Universiti Teknologi Mara, Malaysia in 1995; the Bachelor of Science (Computer) from Universiti Teknologi Malaysia, in 2003; the Master of Computer Science from Universiti Putra Malaysia, in 2006; and the Ph.D. degree at the Universiti Teknologi Malaysia, in 2016. She worked as an IT application developer in various of industries for a decade. She is currently a Senior Lecturer in the Department of Intelligent Computing and Analytics, Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka. Her research interests include power transformer fault diagnostic, missing data imputation, feature selection, and meta-heuristic algorithms. She can be contacted at szahriah@utem.edu.my.



Mr. Abdulrazzak Ali    Member of the Teaching Assistant at the College of Computer and Information Technology, University of Aden. He received a bachelor's degree in computer programming from the University of Mosul, College of Computer Sciences and Mathematics and a master's degree in computer science from the Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka. During his work in the Programming, Research and Development Department of the Computer Center, he contributed to the design and development of a number of computer systems for the University of Aden. He can be contacted at e-mail: dowsan1@yahoo.com.