Imputation of Missing Values through Profiling Metadata

Bernardo Breve, Loredana Caruccio, Vincenzo Deufemia and Giuseppe Polese

University of Salerno, via Giovanni Paolo II, 132, Fisciano (SA), 84084, Italy

Abstract

Among the several problems related to the management of database instances, missing values represents a crucial factor that could severely compromise the integrity and the meaningfulness of such data representations. Thus, the data imputation research field focuses its efforts on solutions for filling missing values by means of plausible candidates, while still preserving the overall semantic integrity the database instance is characterized by. To keep imputation times low while still keeping high accuracy, the employment of metadata has made its way through research proposals. This discussion paper presents our effort in the definition of RENUVER, a novel data imputation algorithm relying on Relaxed Functional Dependencies (RFDs) for identifying value candidates best guaranteeing the semantic integrity of data. Experimental results on real-world datasets highlighted the effectiveness of RENUVER in terms of both filling accuracy and imputation times, also compared to other well-known approaches.

Keywords

Data imputation, Profiling metadata, Relaxed Functional Dependencies, Data quality

1. Introduction

With the advent of big data, the presence of missing values inside database instances has been widely recognized as a complex problem to handle, especially for Relational Database Management Systems [1]. Moreover, several application contexts might require the absence of this data quality issue inside their datasets. For instance, machine learning processes could not provide good accuracy scores if trained on data with many missing values. In general, it is not possible to infer reliable knowledge using datasets with incomplete information [2].

The identification of the best values in a dataset to impute the missing ones is an extremely complex task, since it entails the evaluation of all possible combinations in the value distribution. Most of the approaches proposed in the literature focus on maximizing the number of imputed values, overshadowing the accuracy of single imputations. This discussion paper presents the data imputation algorithm proposed in [3], namely RENUVER (RFD basEd NUll ValuE Repairer), which relies on Relaxed Functional Dependencies (RFDs) for imputing missing values within a relational database instance. By adopting the concept of RFDs as metadata for supporting the imputation process, we are able to perform a broader analysis of the correlations among

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy
bbreve@unisa.it (B. Breve); lcaruccio@unisa.it (L. Caruccio); deufemia@unisa.it (V. Deufemia); gpolese@unisa.it (G. Polese)
0000-0002-3898-7512 (B. Breve); 0000-0002-2418-1606 (L. Caruccio); 0000-0002-6711-3590 (V. Deufemia); 0000-0002-8496-2658 (G. Polese)

attributes, yielding an accurate and somewhat fast solution for the imputation of missing values within relational database instances. In fact, RFDs are still widely considered for detecting and repairing many types of errors, such as duplicates, outliers, and constraint violations [4]. Thus, we made use them for identifying suitable candidate values for replacing missing ones in the data imputation process. RENUVER exploits RFDs for: i) identifying the candidate tuples useful for the imputation of missing values, ii) ranking candidate tuples based on their similarity with respect to the tuples containing missing values, and iii) evaluating each imputation to guarantee the semantic consistency of the whole dataset.

In particular, RENUVER generates candidate tuples and rank them, according to RFDs implying the attribute on which a value is missing. Moreover, the imputation strategy of RENUVER does not alter value consistency with respect to the ones in the original dataset. Finally, RENUVER exploits RFDs to also judge whether it is possible to impute a missing value, in order to preserve the integrity of data and to avoid the insertion of inconsistent information.

The effectiveness of RENUVER has been evaluated on real-world datasets[1] in terms of accuracy, and execution time. In order to extract RFDs, we relied on an existing RFD discovery algorithm [5], since the problem of discovering RFDs is out of the scope of this paper. Moreover, we introduce a novel method for the automatic evaluation of data imputation results, which permits to judge the imputed values even with different syntactical representations. Evaluation results demonstrate that RENUVER outperforms other data imputation approaches [6, 7, 8].

The paper is organized as follows: Section 2 provides preliminary notions on RFDs. Section 3 introduces RENUVER's logic through the employment of the RFDs in the data imputation problem. An experimental evaluation measuring the effectiveness RENUVER is presented in Section 4. Finally, conclusions and further research are reported in Section 5.

2. Preliminaries

Before describing how we approached the imputation problem through the employment of RFDs, let us introduce some propaedeutics notions to our methodology.

Functional Dependency. Given a relational database schema $\mathcal{R}$, and $R = \{A_1, \ldots, A_m\}$ one of its relation schemas, and a tuple $t \in r$, we use $t[A_i]$, with $0 \leq i \leq m$, to denote the projection of t onto A_i; similarly, for a set of attributes $X = \{A_{i_1}, \ldots, A_{i_k}\}$, with $1 \leq k \leq m$, $t[X] \in dom(A_{i_1}) \times \ldots \times dom(A_{i_k})$ represents the projection of t onto X, also denoted with $\Pi_X(t)$. An FD on $\mathcal{R}$ is a statement $X \to Y$ (X implies Y), with $X, Y \subseteq attr(R)$, such that, given an instance r of R, $X \to Y$ is satisfied in r if and only if for each pair of tuples (t_1, t_2) in r, whenever $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$. The sets of attributes X and Y are named Left-Hand-Side (LHS) and Right-Hand-Side (RHS) of the FD, respectively.

With respect to FD definition, the RFD generalizes the comparison paradigm, by including similarity/distance-based comparisons between tuple projections, also admitting the possibility for a dependency to hold only on a subset of tuples. The latter can be defined through either a *coverage measure*, quantifying the portion of the dataset on which a dependency holds or a *condition* restricting the domain on which a dependency can hold [9]. Since the proposed

[1] https://github.com/DastLab/RENUVER-evaluation-datasets

Table 1
A sample of the Restaurant dataset.

	Name	City	Phone	Type	Class
t_1	Granita	Malibu	310/456-0488	Californian	6
t_2	Chinois Main	LA	310-392-9025	French	5
t_3	Citrus	Los Angeles	213/857-0034	Californian	6
t_4	Citrus	Los Angeles	_	Californian	6
t_5	Fenix	Hollywood	213/848-6677	_	5
t_6	Fenix Argyle	_	213/848-6677	French (new)	5
t_7	C. Main	Los Angeles	_	French	5

approach exploits only RFDs relying on a similarity/distance-based tuple comparison method, in what follows we provide only the definition of this type of RFDs, known as RFD_c. For a more general definition of RFD, see [9].

RFD_c. Given a relational database schema $\mathcal{R}$, and $R = \{A_1, \ldots, A_m\}$ one of its relation schemas, an RFD_c φ on $\mathcal{R}$

$$X_{\Phi_1} \rightarrow Y_{\Phi_2} \tag{1}$$

where

- $X, Y \subseteq attr(R)$;
- Φ_1 contains (for each attribute $X_i \in X$) a constraint $\phi_i[X_i]$ that can be used to determine whether pair of tuples with values in $dom(X_i)$ are "similar" enough (likewise for each attribute $Y_j \in Y$ with $\phi_j[Y_j] \in \Phi_2$). More specifically, each $\phi_i[X_i]$ ($\phi_j[Y_j]$ resp.) requires the specification of a similarity/distance function defined on the domain of X_i (Y_j, resp.), an operator, and a threshold setting the boundaries for the satisfaction of the constraint.

holds on a relation instance r (denoted by $r \models \varphi$) if and only if for each pair of tuples (t_1, t_2) $\in r$ for which $t_1[X]$ and $t_2[X]$ satisfy the constraint $\phi_i[X_i]$ for each $X_i \in X$, then $t_1[Y]$ and $t_2[Y]$ satisfy the constraint $\phi_i[Y_i]$ for each $Y_i \in Y$.

For sake of simplicity, in the following, we apply a more compact notation for the constraints, showing only the operator and the numeric threshold associated with each attribute.

Example. Let us consider the sample relation shown in Table 1, derived from a database of restaurants in USA. Within this database, each tuple represents a restaurant providing information about its name, address, city, phone number, type of cuisine, and class. The latter is a numeric id associated to the type of cuisine. On such dataset, the following RFD_c holds: $\text{Name}_{(\leq 4)} \rightarrow \text{Phone}_{(\leq 1)}$ which states that, if two restaurants have a similar name, then they also have a similar phone number. This should be true despite the names and/or the phone numbers of restaurants being written in different ways or using different abbreviations.

From a theoretical point of view, RFD_cs permit to use any type of similarity/distance functions, e.g., edit distance, abs differences, and so forth. However, they are usually inherited from the functions involved in the automatic RFD_c discovery process [5]. For the scope of this proposal, without loss of generality, we can consider RFD_cs with a single attribute on the RHS, and the associated constraint ϕ_2. In particular, we considered ϕ_2 composed of a distance function, the operator $\leq$, and a distance threshold.

A particular type of RFD_c is the *key*-RFD_c, which is defined in the following.

Key RFD_c. Given a relation schema R, and an instance r of R, an RFD_c $\varphi : X_{\Phi_1} \rightarrow A_{\phi_2}$ is said to be *key* if and only if φ holds on r ($r \models \varphi$), but there is no pair of distinct tuples $(t_1, t_2) \in r$, for which $t_1[X]$ and $t_2[X]$ satisfy all the constraints in $\Phi_1[X]$.

3. The RENUVER imputation approach

In this section, we formalize the data imputation problem by defining some of its underlying concepts, then describing the basics of the proposed imputation approach. Let us start defining the concept of missing value.

Missing value. Given a relation schema R, defined over a set of attributes $attr(R)$, an instance r of R, an attribute $A \in attr(R)$, and a tuple $t \in r$, a *missing value* of tuple t on the attribute A, denoted as $t[A] = \_$, is such that $t[A]$ is null.

Here, r is said to be an *incomplete instance*, and $\hat{r} \subseteq r$ contains only *incomplete tuples*.

The general missing value imputation problem is formally defined as follows.

Missing value imputation problem. Given a relation schema R, and an instance r of R, for every tuple $t \in r$ and every attribute $A \in attr(R)$ for which $t[A] = \_$, the imputation problem consists of finding a plausible value $a \in dom(A)$, such that the database instance r' resulting from the imputation process does not contain inconsistent values.

A missing value imputation approach also requires the application of constraints for evaluating the consistency of values at the end of the imputation process. The proposed approach exploits RFDs to both guarantee the verification of the semantic consistency, and to drive the searching of meaningful candidates for all missing values.

Semantically consistent imputation. Given a relation schema R, defined over a set of attributes $attr(R)$, an instance r of R,

and a set of RFD_cs, Σ, holding on r ($r \models \Sigma$), an instance r' of R resulting from an imputation process I over the instance r, denoted as $r' = I(r)$, is *semantically consistent* iff $r' \models \Sigma$. One of the possible strategies that could guarantee the semantic consistency of the imputation process is to find candidate values for $t[A] = \_$ by considering a set $T_{candidate} \subseteq r$ of *plausible* candidate tuples for imputing $t[A]$, such that $\forall t_k \in T_{candidate}$, $t_k[A] \neq \_$ and t_k *is similar* to t on some attributes beyond A.

In what follows we define the criteria used by RENUVER for deciding when a tuple can be considered as a plausible candidate, which is based on RFD_cs.

Plausible candidate tuple. Given a missing value $t[A]$=_ over a database instance r of a relation schema R, and an RFD_c $\varphi : X_{\Phi_1} \rightarrow A_{\phi_2}$ holding on r, a tuple $t' \in r$ can be considered as a *plausible candidate tuple* for imputing $t[A]$ according to φ iff t and t', are similar according to the constraints in Φ_1.

The candidate tuple generation process performed according to the definition presented above, has to be generalized in order to perform the imputation process on tuples containing more than one missing value, and for each $t \in \hat{r}$.

Missing value imputation for a tuple. Let R be a relational schema defined over a set of attributes $attr(R)$, r an instance of R, t a tuple of r, $Z \subset attr(R)$ a set of attributes such that for each $A \in Z$ $t[A] = \_$, and Σ a set of RFD_cs holding on r. An imputation process for t consists of selecting a plausible candidate tuple t_j for each $A \in Z$ such that $t[A] = \_$, so that $t[A]$ can be set equal to $t_j[A]$. However, when for a $t[A] = \_$ it is not possible to identify a plausible candidate tuple guaranteeing a semantic consistent imputation, it is better to leave $t[A]$ unimputed. Although this strategy has been widely applied in other approaches [7], it

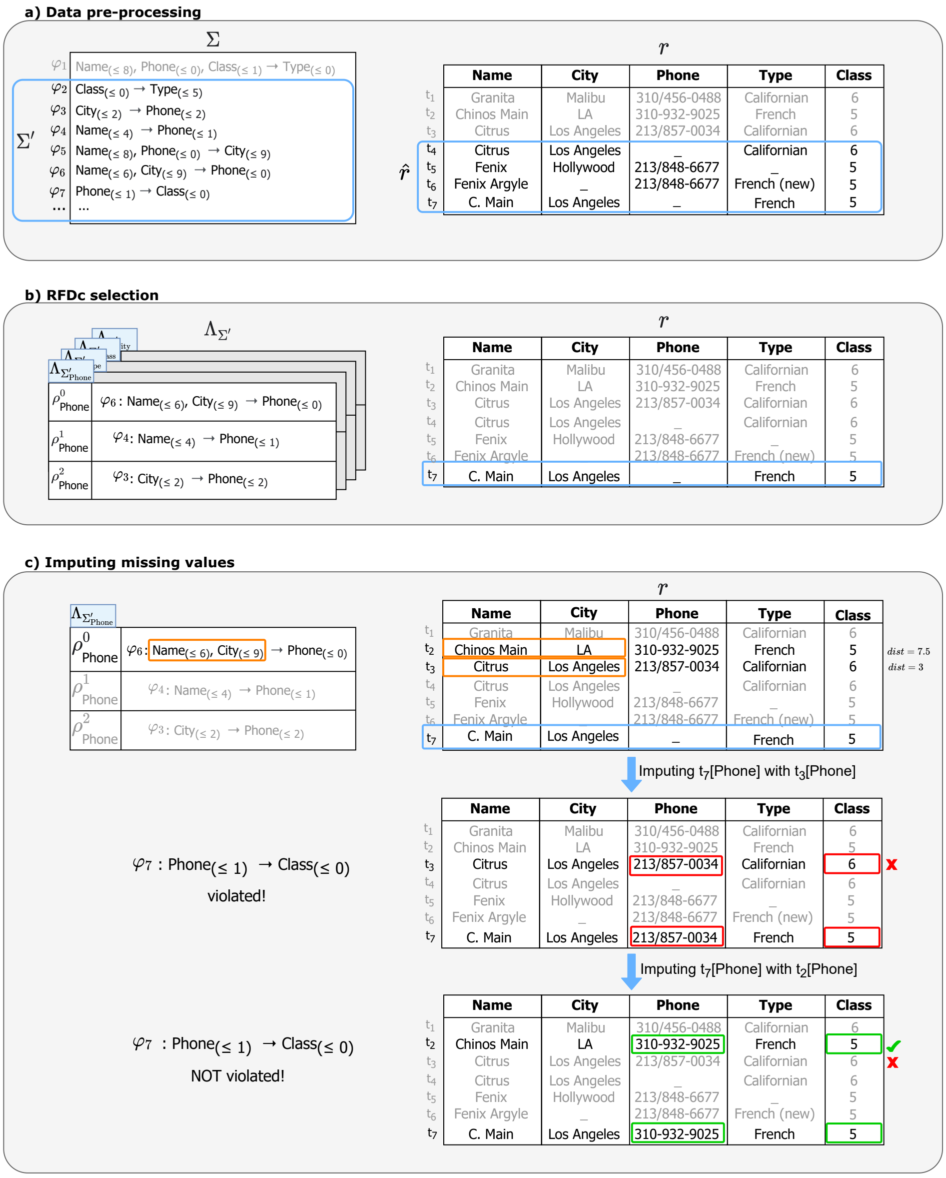

Figure 1: An example of RENUVER imputation on the Restaurant dataset of Table 1.

yields to another important issue that RENUVER deals with, i.e., minimizing the number of non-imputed values.

Figure 1 summarizes the imputation logic of RENUVER[2] through an example. In particular, we show how the aforesaid definitions empower the imputation of a missing value in the Restaurant dataset, previously introduced. In details, we can identify three major phases yielding the imputation of certain missing value, that are:

- **Pre-processing**: during this phase, missing values within a database instance are identi-

[2] A deep overview of RENUVER, together with a more exhaustive evaluation has been carried out in [3].

fied and isolated. Furthermore, RENUVER excludes all key-RFD_cs from the set of the RFD_cs which can be employed for the imputation of any missing value (see Figure 1.a).

- **RFD_c selection**: following the selection of a missing value to impute, during this phase RENUVER identifies all the RFD_cs that can be useful for its imputation. RFD_cs are then organized in a set of clusters according to their threshold on the RHS (see Figure 1.b).
- **Imputing missing values**: during this phase, RENUVER performs a series of operations leading to the imputation of a missing value by retrieving the value from a set of plausible candidate tuples relying on the same database instance (see Figure 1.c). In particular, RENUVER iteratively performs the following operations:
  - generates a set of plausible candidate tuples that satisfy the LHS constraints of an RFD_cs belonging to one of the clusters previously generated.
  - computes a *distance value* for each plausible candidate tuple with respect to the tuple having the missing value. The evaluation is performed by considering the LHS attributes of the RFD_cs selected. Finally the candidate tuple having the minimum distance is the exploited for the imputation of the missing value.
  - verifies whether the imputed value causes a violation of holding RFD_cs. In this case, RENUVER selects the next plausible candidate tuple with the lowest distance value.

  These operations are repeated for each cluster as long as the imputation is not successful.

4. Experimental Evaluation

In this section, we present a comparative evaluation of RENUVER w.r.t. other approaches exploiting different imputation strategies. In particular, we benchmarked RENUVER against an holistic-machine learning-based approach, namely Holoclean [6], (considering its attention-based expansion module AimNet [10]) and a differential dependencies guided approach [7] named Derand, for which we employed the same RFD_cs as RENUVER. All evaluations were performed under the same conditions on an iMac Pro with an 8-core CPU and 32GB RAM.

Datasets. The considered algorithms have been evaluated on two real-world datasets [2] in order to perform a stress test on RENUVER and all compared imputation approaches, aiming to determine their time and memory requirements. To this end, we stopped the executions exceeding 48 hours of execution time and/or 30GB of memory consumption, respectively.

Furthermore, in order to obtain an accurate comparison between the imputed values and the expected ones, missing values have been artificially injected in a random manner. Moreover, to avoid an arrangement of missing values over one algorithm, for each missing injection we produced five different datasets, yielding a total of twenty-five variants of the same dataset. The metrics adopted for the comparison are then averaged over each missing rate.

Evaluation metrics. The effectiveness of the data imputation approaches have been evaluated by considering three different metrics: *precision*, *recall*, *F1-measure*. Which can be formally defined as:

$$precision = \frac{|\text{true} \bigcap \text{imputed}|}{|\text{imputed}|} \qquad recall = \frac{|\text{true} \bigcap \text{missing}|}{|\text{missing}|} \qquad F1\text{-}measure = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where *true* represents the correctly imputed missing values at the end of the imputation process, *imputed* represents all the imputed missing values, and *missing* the missing values in the dataset.

Table 2
Comparative evaluation of RENUVER on the Restaurants and Physician datasets.

Dataset	#Tuples	#Attributes	#Missing val.	#RFD$_c$s	#DCs
Restaurant	864	6	259 (**5%**) 518 (**10%**) 1037 (**20%**) 1555 (**30%**) 2074 (**40%**)	1961	9

Dataset	Approach	Recall	Precision	F1-Meas.	Time	Mem.
Restaurant **(varying the missing rate)**	RENUVER	0.329	0.864	0.476	14m 29s	1.38 GB
		0.296	0.832	0.437	23m 21s	1.31 GB
		0.294	0.845	0.436	33m 20s	1.36 GB
		0.258	0.828	0.394	36m 37s	1.37 GB
		0.232	0.726	0.349	30m 23s	1.38 GB
	Derand	0.295	0.419	0.345	47h 13m	7.21 GB
		-	-	-	TL	-
		-	-	-	TL	-
		-	-	-	TL	-
		-	-	-	TL	-
	Holoclean	0.275	0.544	0.362	14s	0.99 GB
		0.099	0.218	0.131	15s	0.99 GB
		0.071	0.153	0.095	14s	0.99 GB
		0.064	0.192	0.095	11s	0.78 GB
		0.165	0.419	0.237	10s	0.79 GB

TL: time limit of 48 hours exceeded — ML: memory limit of 30 GB exceeded

Dataset	#Tuples	#Attributes	#Missing val.	#RFD$_c$s	#DCs
Physician	104 (**0.05%**) 208 (**0.1%**) 1036 (**0.5%**) 2072 (**1%**) 10359 (**5%**)	13	13 (1%) 27 (1%) 135 (1%) 269 (1%) 1319 (1%)	1430 2553 3895 5708 6137	74

Dataset	Approach	Recall	Precision	F1-Meas.	Time	Mem.
Physician **(varying the number of tuples)**	RENUVER	0.338	1	0.505	470ms	1.48 GB
		0.328	0.547	0.410	3s	1.79 GB
		0.326	0.607	0.424	1m 19s	0.71 GB
		0.254	0.483	0.333	15m 1s	1.30 GB
		-	-	-	TL	-
	Derand	0.121	0.210	0.151	1h 10s	1.25 GB
		0.125	0.190	0.150	9h 49m	3.32 GB
		0.110	0.121	0.115	25h 40m	8.21 GB
		-	-	-	TL	-
		-	-	-	TL	-
	Holoclean	0.230	0.300	0.599	7s	3.95 GB
		0.115	0.120	0.117	12s	5.15 GB
		0.097	0.114	0.104	1m 8s	6.16 GB
		0.156	0.167	0.161	8m 21s	26.89 GB
		-	-	-	-	ML

TL: time limit of 48 hours exceeded — ML: memory limit of 30 GB exceeded

Results. The first evaluation session is focused on the Restaurant dataset by considering the following missing rates: $[5\%, 10\%, 20\%, 30\%, 40\%]$ (see Table 2). We can notice that the fastest approach is Holoclean, whereas Derand registered severely higher execution times, exceeding the 48h time limit starting from the 10% of missing rate. The faster execution times of Holoclean can be justified by the conspicuously lower number of metadata to be processed during the imputation process, i.e., 9 Denial of Constraints, compared to 1961 RFD$_c$s. Nevertheless, RENUVER registered the best performances on all the considered qualitative metrics.

The second evaluation session is focused on the Physician dataset, by fixing the missing rate and by varying the number of tuples to be considered. This dataset is particularly complex to analyze, since it also contains a high number of attributes (i.e., 13 attributes). In fact, this dataset allowed us to catch a time and/or memory limit for all considered approaches (i.e., RENUVER, Derand, and Holoclean), as shown in Table 2. In particular, we can notice that, on average, both RENUVER and Holoclean registered faster execution times than Derand. In fact, the latter exceeds the time limit of 48h on the datasets having 2072 and 10359 tuples, respectively. On the other hand, Holoclean manages to achieve reasonable executions times, but the huge amount of consumed memory makes it exceed the 30GB memory limit on the dataset having 10359 tuples. Finally, RENUVER also exceeds the time limit on the largest dataset, despite a more reasonable memory consumption. This evaluation session proved the capability of RENUVER to outperform the compared approaches on the considered qualitative metrics. It also emphasized that Derand's execution times are strongly dependent on the number of missing values, whereas although Holoclean provided overall faster execution times, it resulted heavily memory-consuming.

5. Conclusion

In this paper, we proposed RENUVER, a data imputation algorithm that exploits relaxed functional dependencies. The latter enables RENUVER to select and evaluate tuple candidates to be used during the imputation process. The whole imputation process preserves the semantic consistency of the data, by guaranteeing that no imputation can violate any RFD_c. Evaluation results demonstrated that RENUVER outperforms recent approaches using different imputation strategies: machine learning-based (Holoclean) and dependency-based (Derand).

In the future, we would like to extend RENUVER with the possibility of selecting plausible candidate tuples among multiple datasets. Finally, we would like to study the applicability of RENUVER over incremental scenarios, like for example those related to the imputation of time series [11], which would require the usage of incremental RFD_c discovery algorithms [12, 13].

References

[1] M. V. Martinez, C. Molinaro, J. Grant, V. Subrahmanian, Customized policies for handling partial information in relational databases, IEEE Transactions on Knowledge and Data Engineering 25 (2012) 1254–1271.

[2] B. Montesdeoca, J. Luengo, J. Maillo, D. García-Gil, S. García, F. Herrera, A first approach on big data missing values imputation, in: Proceedings of 5th International Conference on Internet of Things, Big Data and Security (IoTBDS), SciTePress, 2019, pp. 315–323.

[3] B. Breve, L. Caruccio, V. Deufemia, G. Polese, RENUVER: A missing value imputation algorithm based on relaxed functional dependencies, in: To appear in Proceedings of the 25th International Conference on Extending Database Technology, (EDBT), OpenProceedings.org, 2022.

[4] I. F. Ilyas, X. Chu, et al., Trends in cleaning relational data: consistency and deduplication, Foundations and Trends® in Databases 5 (2015) 281–393.

[5] L. Caruccio, V. Deufemia, F. Naumann, G. Polese, Discovering relaxed functional dependencies based on multi-attribute dominance, IEEE Transactions on Knowledge and Data Engineering 33 (2021) 3212–3228.

[6] T. Rekatsinas, X. Chu, I. F. Ilyas, C. Ré, Holoclean: holistic data repairs with probabilistic inference, Proceedings of VLDB Endowment 10 (2017) 1190–1201.

[7] S. Song, Y. Sun, A. Zhang, L. Chen, J. Wang, Enriching data imputation under similarity rule constraints, IEEE Transactions on Knowledge and Data Engineering 32 (2020) 275–287.

[8] C.-C. Huang, H.-M. Lee, A grey-based nearest neighbor approach for missing attribute value prediction, Applied Intelligence 20 (2004) 239–252.

[9] L. Caruccio, V. Deufemia, G. Polese, Relaxed functional dependencies—A survey of approaches, IEEE Transactions on Knowledge and Data Engineering 28 (2016) 147–165.

[10] R. Wu, A. Zhang, I. Ilyas, T. Rekatsinas, Attention-based learning for missing data imputation in holoclean, Proceedings of Machine Learning and Systems 2 (2020) 307–325.

[11] M. Khayati, A. Lerner, Z. Tymchenko, P. Cudré-Mauroux, Mind the gap: An experimental evaluation of imputation of missing values techniques in time series, Proceedings VLDB Endowment 13 (2020) 768–782.

[12] L. Caruccio, S. Cirillo, V. Deufemia, G. Polese, Incremental discovery of functional dependencies with a bit-vector algorithm, in: Proceedings of Italian Symposium on Advanced Database Systems, volume 2400 of *SEBD '19*, CEUR-WS.org, 2019, pp. 1–12.

[13] L. Caruccio, S. Cirillo, Incremental discovery of imprecise functional dependencies, Journal of Data and Information Quality (JDIQ) 12 (2020) 1–25.