A Spatio-Temporal Data Imputation Model for Supporting Analytics at the Edge

Christos Anagnostopoulos, Stathes Hadjiefthymiades, Kostas Kolomvatsos, Panagiota Papadopoulou

A Spatio-Temporal Data Imputation Model for Supporting Analytics at the Edge

Kostas Kolomvatsos[1], Panagiota Papadopoulou[1], Christos Anagnostopoulos[2], Stathes Hadjiefthymiades[1]

[1] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece
{kostasks, peggy, shadj}@di.uoa.gr
[2] School of Computing Science, University of Glasgow, UK
christos.anagnostopoulos@glasgow.ac.uk

Abstract. Current applications developed for the Internet of Things (IoT) usually involve the processing of collected data for delivering analytics and support efficient decision making. The basis for any processing mechanism is data analysis, usually having as an outcome responses in various analytics queries defined by end users or applications. However, as already noted in the respective literature, data analysis cannot be efficient when missing values are present. The research community has already proposed various missing data imputation methods paying more attention of the statistical aspect of the problem. In this paper, we study the problem and propose a method that combines machine learning and a consensus scheme. We focus on the clustering of the IoT devices assuming they observe the same phenomenon and report the collected data to the edge infrastructure. Through a sliding window approach, we try to detect IoT nodes that report similar contextual values to edge nodes and base on them to deliver the replacement value for missing data. We provide the description of our model together with results retrieved by an extensive set of simulations on top of real data. Our aim is to reveal the potentials of the proposed scheme and place it in the respective literature.

Keywords: Internet of Things, Edge Computing, Missing values Imputation, Clustering, Consensus

1 Introduction

Modern applications aiming at providing innovative services to end users are based on the management of responses in analytics queries. Such queries target to the provision of the results of data analysis that will facilitate knowledge extraction and efficient decision making. Any processing will be realized on top of data collected by various devices or produced by end users. If we focus on the Internet of Things (IoT), we can detect numerous devices capable of collecting data and interacting each other to support the aforementioned applications. IoT

devices can send the collected / observed data to the edge infrastructure, then, to the Cloud for further processing. The envisioned analytics can be provided either at the Cloud or at the edge of the network to reduce the latency in the provision of responses. With this architecture, we can support innovative business models that could create new roads for revenues offering novel applications in close proximity with end users.

Edge nodes can interact with a set of IoT devices to receive the collected data and perform the processing that analytics queries demand. IoT devices create streams of data towards the edge nodes, however, due to various reasons these streams can be characterized by missing values. Missing values can be a serious impediment for data analysis [14]. Various methodologies have been proposed for handling them [11]: data exclusion, missing indicator analysis, mean substitution, single imputation, multiple imputation techniques, replacement at random, etc. To the best of our knowledge, the majority of the research efforts mainly focus on the 'statistical' aspect of the problem trying to provide a methodology for finding the best values to replace the missing one with the assistance of statistical methodologies. Their aim is to identify the distribution of data under consideration and produce the replacements.

In this paper, we go a step forward and propose a missing value imputation method based not only on a statistical model but also on the dynamics of the environment where IoT devices act. We deliver a technique that deals with the group of nodes as they are distributed in the space and the temporal aspect of the data collection actions. When a missing value is present, we rely on the peer IoT devices located in close proximity to conclude the envisioned replacements. The proximity is detected not only in relation with the location of the devices but also in relation with the collected data. We propose the use of a two layered clustering scheme and a data processing model based on a sliding window approach. The first clustering process is applied on the IoT devices spatial information while the second is applied on top of the collected data. Our aim is to identify the devices reporting similar multidimensional data for the same phenomenon enhanced by the correlation of each individual dimension in a successive step. We are able to combine two different techniques, i.e., an unsupervised machine learning model with a consensus based strategy to conclude the final replacements for any observed missing value.

The remaining paper is organized as follows. Section 2 reports on the prior work in the domain while Section 3 presents the problem under consideration and gives insights into our model. Section 4 discusses the proposed solution and provides formulations and our solution. Section 5 describes our experimental evaluation efforts and gives numerical results for outlining the pros and cons of our model. Finally, in Section 6, we conclude our paper by presenting our future research plans.

2 Prior Work

Data management in the IoT has received significant attention in recent years. The interested reader can refer in [9] for a review of the domain. IoT based large

scale data storage in Cloud is studied by [4], where a review of acquisition, management, processing and mining of IoT big data is also presented. The authors of [8] discuss a comparison of Edge computing implementations, Fog computing, cloudlets and mobile Edge computing. The focus is also on a comparative analysis of the three implementations together with the necessary parameters that affect nodes communication (e.g., physical proximity, access mediums, context awareness, power consumption, computation time). Data storage and management is also the focus of [27]. The authors propose a model and a decision making scheme for storing the data in Cloud. The storage decision is delivered on top of a mathematical model that incorporates the view on the available resources and the cost for storing the envisioned data. Another storage framework is presented by [17]. The authors deal with structured and unstructured data combining multiple databases and Hadoop to manage the storage requirements. In [12], the authors propose a system to facilitate mobile devices and support a set of services at the Edge of the network. A controller is adopted to add the devices to the available clusters, thus, the system can have a view on how it can allocate the envisioned tasks. A storage model enhanced with a blockchain scheme is discussed in [30]. The proposed model aims at increasing the security levels for distributed access control and data management. In [10], the authors present a scheme for security management in an IoT data storage system. The proposed scheme incorporates a data pre-processing task realized at the edge of the network. Time-sensitive data are stored locally, while non-time-sensitive data are sent to the Cloud back end infrastructure. Another distributed data storage mechanism is provided by [35]. The authors propose a multiple factor replacement algorithm to manage the limited storage resources and data loss.

Missing data imputation is a widely studied subject in multiple application domains as it is a very important topic for supporting efficient applications. Moreover, imputation mechanisms can be applied over various types of values, e.g., over sensory data [16]. The simplest way to impute missing data is to adopt the mean of values; this technique cannot take into consideration the variance of data or their correlation [21] being also affected by extreme values. Hence, research community also focused on other statistical learning techniques to provide more robust models for missing data substitution. Statistical learning focuses on the detection of statistical dependencies of the collected data [19], [36]. One example is the imputation scheme based on Auto-Regressive Integrated Moving Average and feed forward prediction based method [7]. Any prediction model builds on top of historical values, thus, researchers have to take into consideration the prediction error and the demand for resources required for storing all the necessary historical observations. Usually, a sliding window approach is adopted to manage the most recent measurements, thus, to limit the demand for increased resources. When corrupted or missing data are identified, the calculated probability distribution is adopted for the final replacement [36]. Other efforts deal with the joint distribution on the entire data set. Such efforts assume a parametric density function (e.g., multivariate normal) on the data given with estimated parameters [15]. The technique of least squares provides individual univariate regressions to impute features with missing values on all of the

other dimensions based on the weighted average of the individual predictions [2], [25]. Extensions of the least squares method consist of the Predictive-Mean Matching method (PMM) where replacements are random samples drawn from a set of observed values close to regression predictions [3] and Support Vector Regression (SVR) [34]. Apart from linear regression models, other imputation models incorporate random forests [32], K-Nearest Neighbors (K-NN) [33], sequential K-NN [18], singular value decomposition and linear combination of a set of eigenvectors [33], [22] and Bayesian Principal Component Analysis (BPCA) [23], [24]. Probabilistic Principal Component Analysis (PPCA) and Mixed Probabilistic Principal Component Analysis (MPPCA) can be also adopted to impute data [36]. All the aforementioned techniques try to deal with data that are not linearly correlated providing a more 'generic' model. Formal optimization can be also adopted to impute missing data with mixed continuous and categorical variables [1]. The optimization model incorporates various predictive models and can be adapted for multiple imputations.

It becomes obvious that any data imputation process incorporates uncertainty related to the adopted decisions for substituting absent values. Fuzzy Logic (FL) and machine learning algorithms can contribute in the management of uncertainty and the provision of efficient schemes, especially when combined with other computational intelligence techniques. In [31], the authors proposes the use of a hybrid method having the Fuzzy C-means (FCM) algorithm combined with a Particle Swarm Optimization (PSO) model and a Support Vector Machine (SVM). Patterns of missing data are analysed and a matrix based structure is used to represent them. Other models involve Multi-layer Perceptrons (MLPs) [26], Self-Organizing Maps (SOMs) [6], and Adaptive Resonance Theory (ART) [5]. The advantages of using neural networks for this problem are that they can capture many kinds of relationships and they allow quick and easy modeling of the environment [20].

In our model, we aim to avoid the use of a scheme that requires a training process, thus, we target to save time and resources. The proposed approach is similar to the scheme presented in [19], however, we do not require a training process to build our model. We focus on the adoption of an unsupervised machine learning technique combined with a fast consensus model for the delivery of the replacement of a missing value. We aim to build on top of the spatio-temporal aspect of the collected data, i.e., the location where they are reported and the report time. We adopt a sliding window approach and use spatial clusters of the IoT devices. A second clustering process is realized on top of the collected data to detect the devices reporting similar information to the edge nodes. Based on this approach, we can handle a dynamic environment where nodes change their location. The data correlation between IoT devices is adopted to provide the basis for our consensus model in the proposed imputation method. Hence, any missing value is replaced on top of the 'opinion' of the IoT devices having the same 'view' on the phenomenon.

3 Preliminaries

Our scenario involves a set of Edge Nodes (ENs) where a number of IoT devices are connected to report the collected data. The proposed model aims to support the behaviour of ENs and provides a model for missing data imputation based on the data received by all the IoT nodes in the group. Without loss of generality, we focus on the behaviour of an EN and consider a set $\mathcal{N}$ of IoT devices i.e., $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$. IoT devices are capable of observing their environment, collect data and performing simple processing tasks. As their resources are limited, IoT devices should store only the necessary data. These data are updated while the remaining are sent to ENs or the Fog/Cloud for further processing. It is worth noticing that when IoT devices rely on the Fog/Cloud for the processing of data they enjoy increased latency [28].

We consider that data are received and stored in the form of multivariate vectors i.e., $\overrightarrow{x} = [x_1, x_2, \ldots, x_M]$ where M is the number of dimensions. Let D_i be the dataset stored in the ith EN. The EN should identify if the incoming data contain missing values and when this is true, it should apply the proposed imputation technique. We consider the discrete time $\mathbf{T}$. At $t \in \mathbf{T}$, the EN receives a set of multivariate vectors coming from the IoT devices, i.e., $\overrightarrow{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iM}], i = 1, 2, \ldots, N$. The missing data can refer in: (i) the whole vector; (ii) specific dimensions of the reported vectors. When a value x_{jk} is absent, the EN should replace it with the result of our imputation function, i.e., $x_{jk} = f\left(\overrightarrow{x}_i\right), \forall i$.

$f\left(\right)$ builds on top of a sliding window approach. The window W deals with the interval where data can be adopted to 'generate' the missing dimension(s). In addition, the EN maintains a set of clusters of nodes based on their spatial proximity. When nodes are static, our approach considers a 'static' clustering model. When IoT devices are mobile, we have to perform the clustering process at pre-defined intervals. In any case, this will add overhead in the performance of the system. We can reduce the overhead if we rely on an incremental clustering algorithm to save time and resources. The imputation function takes into consideration the location of nodes before it delivers the final result. This approach enhances the localized aspect of decision making adopted into our model. Afterwards, the imputation process is based on only the data coming from the devices located in close distance that are correlated with the data reported by the device where missing values are observed. The envisioned architecture is depicted by Figure 1. It should be noted that we do not focus on IoT devices with 'special' requirements, e.g., sensors that record images performing advanced processing models.

4 The Proposed Model

Data Clustering and Correlation. The proposed model performs a hierarchical clustering, i.e., it creates clusters based on the spatial proximity of IoT

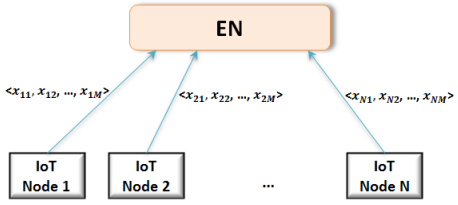

Fig. 1: The envisioned architecture.

devices and accordingly it delivers clusters based on the data proximity between the previously selected devices. For the clustering process based on the location of the devices, we can adopt any clustering algorithm (e.g., k-means or a subtractive method). Assume that this process returns the set $\mathcal{N}$ of the IoT devices (N IoT devices). The ith IoT device reports to the EN a data vector $\overrightarrow{x}_i^t = [x_{i1}, x_{i2}, \ldots, x_{iM}]$ at t. The EN performs the envisioned processing over the pre-defined window W. Hence, the EN has access to the $W \times M$ matrix $\overrightarrow{X} = \{\overrightarrow{x}_1^t, \overrightarrow{x}_2^t, \ldots, \overrightarrow{x}_N^t\}, \forall t \in [1, W]$. In each cell of this matrix, the EN stores the multidimensional vector reported by the corresponding IoT device at t. An additional vector $\mathbf{I}$ is adopted to store the ids of the involved devices. The discussed matrix can be characterized by 'gaps' in the collected values, i.e., the missing values that should be replaced.

We propose a second level of clustering as follows. For every $t \in [1, W]$, we perform clustering for N data vectors and store the corresponding ids. Figure 2 presents an indicative example. For the delivery of clusters, we adopt the Euclidean distance and the k-means algorithm. The distance between two vectors i and j will be delivered as follows: $\|\overrightarrow{x}_i - \overrightarrow{x}_j\| = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{iM} - x_{jM})^2}$. The k-means algorithm is simple and adopts a set of iterations for concluding the final clusters. After the initial generation of k random multidimensional centroids $m_1, m_2, \ldots, m_k$, at each iteration, the algorithm assigns every vector to the closest centroid. The objective is to find the $\arg\min_{\overrightarrow{X}_{row}} = \sum_{c=1}^{k} \sum_{\overrightarrow{x} \in \overrightarrow{X}_{row}} \|\overrightarrow{x} - m_c\|^2$. This is realized in the assignment step of the algorithm. In the update step, centroids are updated to depict the vectors participating in each cluster, i.e., $m_c = \frac{1}{|S_c|} \sum_{\overrightarrow{x} \in S_c}$ where S_i is the set of vectors participating in the cth cluster.

Let the ids of the IoT devices be annotated with n_i^{id}. At each t, we focus on the k clusters where ids are present. We consider that every cluster represents a 'transaction', thus, at each t we have to process k transactions. Every n_i^{id} is present in a cluster, thus, in a single transaction. In total, n_i^{id} will be present in W transactions. The ids present in each cluster vary. For instance, at $t = 1$, the 1st device (e.g., $n_1^{id} = XYZ$) can be present in the 2nd cluster together with two more peers, e.g., $n_5^{id} = YSZ$ and $n_3^{id} = BCD$, at $t = 2$, the 1st device can be present in the 3nd cluster, and so on and so forth. Figure 2 presents a clustering example.

Every transaction is an ID-set depicting the corresponding cluster. The presence of specific ids in an ID-set represents the correlation between the corresponding IoT devices as delivered by the clustering algorithm. When a missing value is present in a device, we consider the intersection of the ID-sets where the id of the device is present. The aim is to identify the devices that are in close data distance in W. Let $L^I_{n_i}$ be the intersection list for n_i. $L^I_{n_i}$ represents the intersection of W transactions; actually, we deliver nodes that are in the same cluster for αW transactions, $\alpha \in [0, 1]$. Together with $L^I_{n_i}$, we provide the list $L^C_{n_i}$ where the multidimensional correlation result between n_i and any other device present in $L^I_{n_i}$ is maintained. We detect the correlation between the corresponding dimensions in W. To produce $L^C_{n_i}$, we adopt the known Pearson Correlation Coefficient (PCC) for each dimension of vectors reported by two devices. The PCC is calculated for each device present in $L^I_{n_i}$ with the current device where a missing values is observed. Assume that we have to calculate the PCC for devices i and j. The final PCC is: $R_{PCC} = \sum_{l=1}^{M} r_{\vec{x}^t_{il}, \vec{x}^t_{jl}}, \forall t \in [1, W]$ with $r_{\vec{x}_{il}, \vec{x}_{jl}} = \frac{\sum_{t=1}^{W}(x_{il}-\overline{x_{il}})(x_{jl}-\overline{x_{jl}})}{\sqrt{\sum_{t=1}^{W}(x_{il}-\overline{x_{il}})^2}\sqrt{\sum_{t=1}^{W}(x_{jl}-\overline{x_{jl}})^2}}$ When applying the PCC in a single dimension, we get results in the interval [-1,+1]. In our case, due to the multiple dimensions, we get results in the interval $[-M, +M]$. Hence, the final format of $L^C_{n_i}$ is $L^C_{n_i} = \{R^{n_j}_{PCC}\}$ where j depicts the nodes present in $L^I_{n_i}$. When $R^{n_j}_{PCC} \to +M$ means that n_i and n_j exhibit a high positive correlation for all the envisioned dimensions while a strong negative correlation is depicted by $R^{n_j}_{PCC} \to +M$. The $L^C_{n_i}$ is sorted in a descending order and adopted to deliver the replacement of missing values as we report in the upcoming section.

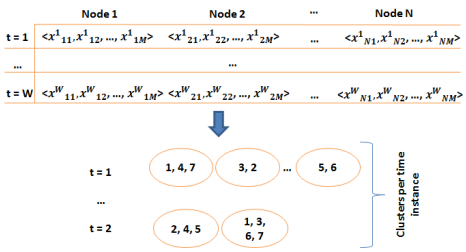

Fig. 2: An example of the envisioned clustering process.

Data Imputation. For substituting missing values, we rely on $L^I_{n_i}$ & $L^C_{n_i}$ and we adopt the linear opinion pool model. For each device present in $L^I_{n_i}$, we focus on the correlation with the device requiring the missing value imputation, say n_i. At first, we focus on the dimension where the missing value is present. If multiple dimensions suffer, we adopt and iterative approach over the entire set of the dimensions under consideration. For each peer device in $L^I_{n_i}$, we rely on devices exhibiting a strong positive correlation with n_i. Let us focus on the subset C of correlated devices and the lth dimension. Our model proposes the use of the linear opinion pool scheme for the lth dimension in W. At first, we focus on the time instance t^* where the missing value is observed. At t^*, we have

available $|C|$ values observed by the devices exhibiting a high correlation with n_i; each one has already observed a value for the lth dimension.

The linear opinion pool is a standard approach adapted to combine experts opinion (i.e., devices) through a weighted linear average of the adopted values. Our aim is to combine single experts opinions and produce the most representative value for the missing observation. We define a specific weight for each node in C to 'pay more attention' on its measurement, thus, to affect more the final aggregated result, i.e., the missing value substitution. Formally, $F(x_{1l}, \ldots, x_{|C|l})$ is the aggregation opinion operator (i.e., the weighted linear average), i.e., $y = F(x_{1l}, \ldots, x_{|C|l}) = \sum_{c=1}^{|C|} w_c x_{cl}$ where w_c is the weight associated with the measurement of the cth node such that $w_c \in [0, 1]$ and $\sum_{\forall c} w_c = 1$. Weights w_c are calculated based on the correlation with peer nodes depicted by $L_{n_i}^C$; $w_c = \frac{R_{PCC}^{n_j}}{\sum_{\forall n_j \in C} R_{PCC}^{n_j}}$. Weights are calculated on top of the correlation of all dimensions as we want to avoid any 'random' correlation events. Evidently, the mechanism assigns a high weight on the node that exhibits a high correlation with n_i. The final result y replaces the missing value observed at n_i.

5 Experimental Evaluation

Experimental Setup & Performance Metrics. We report on the performance of the proposed scheme aiming to reveal if it is capable of correctly substituting any missing value. Aiming at evaluating the 'proximity' of the replacement value with the real one, we adopt the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). MAE is defined as follows: $MAE = \frac{1}{|V|} \sum_{i=1}^{|V|} |v_i - \hat{v}_i|$ where $|V|$ is the number of missing values in our dataset (V denotes the set of the missing values), v_i is the actual and $\hat{v}_i$ is the proposed value. RMSE is defined as follows: $RMSE = \sqrt{\frac{1}{|V|} \sum_{i=1}^{|V|} (v_i - \hat{v}_i)^2}$. RMSE is similar to MAE, however, RMSE assigns a large weight on high errors. RMSE is more useful when high errors are undesirable.

We rely on three real datasets, i.e., (i) the GNFUV Unmanned Surface Vehicles Sensor Data Set [13]; (ii) the Intel Berkeley Research Lab dataset [3] and (iii) the Iris dataset [4]. The GNFUV dataset comprises values of mobile sensor readings (humidity, temperature) from four Unmanned Surface Vehicles (USVs). The swarm of the USVs is moving according to a GPS predefined trajectory. The Intel dataset contains millions of measurements (temperature, humidity, light) retrieved by 54 sensors deployed in a lab. From this dataset, we get 15,000 measurements such that 15 sensors produced 1,000 measurements. Finally, the Iris dataset involves the classification of flowers into specific categories based on their attributes (e.g., sepal length).

We present results for our Clustering Based Mechanism (CBM) compared with an Averaging Mechanism (AM) and the Last Value Mechanism (LVM).

[3] Intel Lab Data, http://db.csail.mit.edu/labdata/labdata.html
[4] http://archive.ics.uci.edu/ml/datasets/iris

The AM replaces any missing value with the mean of values reported by the peer devices at the same time interval. The LVM replaces missing values with the observation retrieved in the previous recording interval in the same device. At random time steps, we consider that a missing value is observed in a device selected randomly as well. We calculate the replacements for the considered schemes and compare them with the real ones to deliver the MAE and RMSE measurements. Our experiments deal with $W \in \{5, 10, 50\}$ and $M \in \{5, 50, 100\}$ trying to reveal the 'reaction' of our model to different window size and number of dimensions.

Performance Assessment. Our experimental evaluation involves a large set of experiments on top of the aforementioned datasets. In Figure 3, we present our results for the GNFUV dataset (Left: MAE results; Right: RMSE results). We observe that our CBM exhibits the best performance when $W = 5$. Actually, it outperforms the AM (for $W \in \{5, 10\}$) and exhibits worse performance than the LVM (MAE results). When the RMSE is the case, the CBM outperforms both models when $W = 5$. A short sliding window positively affects the performance of our model as the EN decides on top of a low number of the envisioned clusters delivered for each t. The error of CBM increases as W increases as well. This also exhibits the capability of the CBM to deliver good results on top of a limited amount of data.

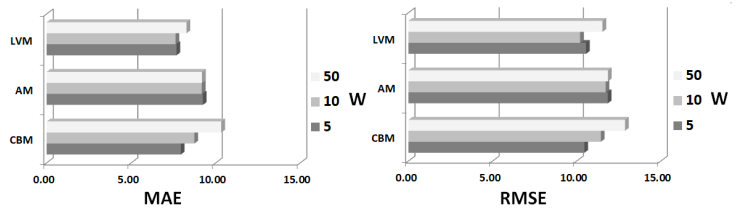

Fig. 3: MAE and RMSE for the GNFUV dataset.

In Figure 4, we present our results for the Intel dataset. We observe that the CBM, again, for a short W exhibits the best performance compared to the remaining models. The CBM 'produces' 16% (approx.) less MAE compared to the AM and 35% (approx.) less MAE compared to the LVM (for ($W = 5$). When $W \rightarrow 50$, the CBM leads to 23% (approx.) more MAE than the AM and 2% (approx.) less MAE than the LVM. Similar results are observed for the RMSE which support the conclusion that the proposed model is more efficient when the EN is 'forced' to take decisions on top of a limited list of historical values.

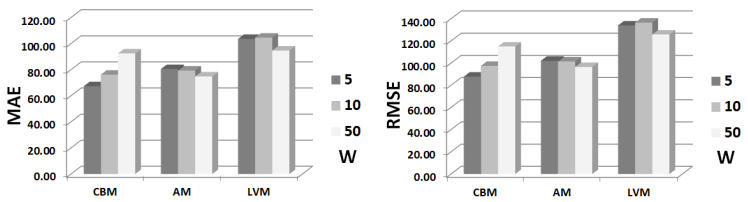

Fig. 4: MAE and RMSE for the Intel dataset.

In Figure 5, we see our results for the Iris dataset. The CBM exhibits better performance than the AM but worse performance than the LVM. However, the results for the discussed models are very close. In general, the MAE for our CBM is in [0.51, 0.57] and the RMSE is in [0.63, 0.74]. Comparing our model with other schemes proposed in the literature, we focus on the comparative assessment discussed in [29]. There, the authors adopt the Iris dataset and provide performance results for the following missing values imputation algorithms: Mean, K-nearest neighbors (KNN), Fuzzy K-means (FKM), Singular Value Decomposition (SVD), bayesian Principal Component Analysis (bPCA) and Multiple Imputations by Chained Equations (MICE). The provided results deal with an RMSE in [5, 20] which is worse than the performance of the proposed CBM. Finally, in Figure 6, we provide our results for the GNFUV dataset and for different M realizations. Concerning the MAE, the CBM performs better than the LVM for all M and the AM for $M = 5$. When $M > 5$, the AM exhibits the nest performance. A low number of dimensions lead to the best performance for the CBM. The CBM's MAE is around 0.5 while the RMSE is around 0.70.

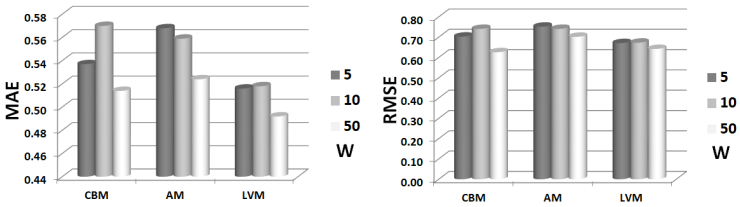

Fig. 5: MAE and RMSE for the Iris dataset.

Concluding the presentation of our results, we can note that in practical terms, the proposed CBM manages to efficiently replace the missing values when it deals with 'fresh' data and a low number of dimensions. The reason is that the CBM is affected by the clustering process which is applied on the multivariate data vectors. When the number of vectors and dimensions increase, there is a room for accumulating the distance of the vectors from the centroids, thus, we can meet vectors with high distance from centers affecting the final calculation of the substitution values.

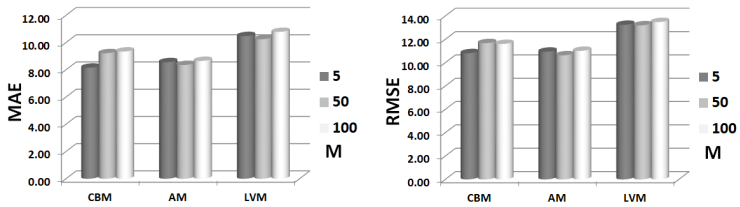

Fig. 6: MAE and RMSE for the GNFUV dataset and different dimensions.

6 Conclusions & Future Work

Missing values imputation is a significant task for supporting efficient data analysis, thus, efficient decision making. In the IoT, data can be collected by numerous

devices transferred to the available edge nodes and the Cloud for further processing. Edge nodes can host the data and process them to deliver analytics limiting the latency. However, due to various reasons, the reported data can contain missing values. We propose a model for enhancing edge nodes behaviour to be capable of handling possible missing values. Our contribution deals with the provision of a two layered clustering scheme and a consensus methodology for the substitution of any missing value. Edge nodes take into consideration the observations retrieved by peer devices in close proximity that report similar data. The replacement values are calculated on top of the data of 'similar' devices weighted by the correlation between the device reporting the missing data and its peers. We provide the results of extensive simulations on top of real data and reveal the strengths of the proposed model. Our future research plans involve the definition and adoption of a more complex methodology taking into consideration the uncertainty behind the adoption of specific peer devices in the envisioned processing.

Acknowledgment

This work is funded by the H2020 research project under the grant agreement no 833805 (ARESIBO).

References

1. Bertsimas, D. et al., 'From Predictive Methods to Missing Data Imputation: An Optimization Approach', JMLR, 18, 2018, 1–30.
2. Bo, T. et al., 'LSimpute: accurate estimation of missing values in microarray data with least squares methods', NAR, 32(3), 2004.
3. Buuren, S., Groothuis-Oudshoorn, K., MICE: Multivariate imputation by chained equations in R, JSS, vol. 45(3), 2011.
4. Cai H., et al., 'IoT-based big data storage systems in Cloud computing: Perspectives and challenges', IEEE IOT, 4(1), 2017, 75–87.
5. Carpenter, G., Grossberg. S., Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks, 1991, 759-771.
6. Catterall, et al., Self organization in ad hoc sensor networks: An empirical study, 8th ICSSL, 2002.
7. Chang, G., Ge, T., Comparison of Missing Data Imputation Methods for Traffic flow, ICTMEE, 2011.
8. Dolui, K. and Datta, K. S., 'Comparison of edge computing implementations: Fog computing, Cloudlet and mobile edge computing', IEEE GIoTS, 2017.
9. Escamilla-Ambrosio P.J., et al., 'Distributing Computing in the Internet of Things: Cloud, Fog and Edge Computing Overview', In: Maldonado Y., et al., (eds), Studies in Computational Intelligence, 731, 2018.
10. Fu, J.-S., et al., 'Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing', IEEE TII, 2018.
11. Guan, N. C., Yusoff, M. S. B., 'Missing Values in data Analysis: Ignore or Impute?', EMJ, 3(1), 2011.

12. Habak, K., et al., 'Femto Clouds: leveraging mobile devices to provide Cloud service at the edge, 8th IEEE CLOUD, 2015, 9-16.
13. Harth, N., Anagnostopoulos, C., 'Edge-centric Efficient Regression Analytics', IEEE EDGE, 2018.
14. He, Y., 'Missing Data Analysis Using Multiple Imputation: getting to the heart of the Matter', CCQO, 3(1), 2010, 98–105.
15. Honaker, J., et al., Amelia II: A program for missing data, JSS, 45(7), 2011, 1–47.
16. Jiang, N., 'A Data Imputation Model in Sensor Databases', ICHPCC, 2007, pp. 86–96.
17. Jiang, L., et al., 'An IoT-Oriented Data Storage Framework in Cloud Computing Platform', IEEE TII, 2015, 10(2), 1443–1451.
18. Kim, L., et al., Reuse of imputed data in microarray analysis increases imputation efficiency, BMC Bioinformatics, 5(1), 2004.
19. Ku, W., et al., A Clustering-Based Approach for Data-Driven Imputation of Missing Traffic Data, IEEE FISTA, 2016.
20. Li, Y., Parker, L., A spatial-temporal imputation technique for classification with missing data in a wireless sensor network, IEEE ICIRS, 2008.
21. Little, R., Rubin, D., Statistical Analysis with Missing Data, Wiley, 1987.
22. Mazumder, R., et al., Spectral regularization algorithms for learning large incomplete matrices, JMLR, 11, 2010, 2287–2322.
23. Mohamed, S., et al., Bayesian exponential family PCA, ANIPS, 2009, 1089–109.
24. Oba, S., et al., A Bayesian missing value estimation method for gene expression profile data, Bioinformatics, 19(16), 2003, 2088–2096.
25. Raghunathan, T., et al., A multivariate technique for multiply imputing missing values using a sequence of regression models, Survey Methodology, 27(1), 2001, 85–96.
26. Reznik, L., et al., Signal change detection in sensor networks with artificial neural network structure, IEEE ICCIHSPS, 2005, 4451.
27. Ruiz-Alvarez, A. and Humphrey, M. (2012). A Model and Decision Procedure for Data Storage in Cloud Computing. 12th IEEE/ACM CCGrid, 2012.
28. Satyanarayanan, M., 'A brief history of cloud offload: A personal journey from Odyssey through cyber foraging to cloudlets', MCC, 18(4), 2015, 19–23.
29. Schmitt, P., et al., 'A Comparison of Six Methods for Missing Data Imputation', Journal of Biometrics & Biostatistics, 6(1), 2015.
30. Shafagh, H., et al., 'Towards Blockchain-based Auditable Storage and Sharing of IoT Data', 9th ACM CCS Workshop, 2017.
31. Shang, B., et al., An Imputation Method for Missing Traffic Data Based on FCM Optimized by PSO-SVR, JAT, 2018.
32. Stekhoven, D., Buhlmann, P., Missforest: non-parametric missing value imputation for mixed-type data, Bioinformatics, 28(1), 2012, 112–118.
33. Troyanskaya, O., et al., Missing value estimation methods for DNA microarrays, Bioinformatics, 17(6), 2001, 520–525.
34. Wang, X., et al., Missing value estimation for DNA microarray gene Expression data by support vector regression imputation and orthogonal coding scheme, BMC Bioinformatics, 7(1), 2006.
35. Xing, J., et al., 'A distributed multi-level model with dynamic replacement for the storage of smart edge computing', JSA, 83, 2018, 1-11.
36. Zhao, N., et al., Improving the Traffic Data Imputation Accuracy Using Temporal and Spatial Information, ICICTA, 2014.