IoT Data Imputation with Incremental Multiple Linear Regression

Tao Peng, Sana Sellami, Omar Boucelma

Research online Publishing
www.ronpub.com

Open Access

Open Journal of XXX (OJXX)
Volume X, Issue X, 20XX
http://www.ronpub.com/ojxx
ISSN XXXX-XXXX

IoT Data Imputation with Incremental Multiple Linear Regression

Tao Peng, Sana Sellami, Omar Boucelma

Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France, {tao.peng, sana.sellami, omar.boucelma}@lis-lab.fr

Abstract

In this paper, we address the problem related to missing data imputation in the IoT domain. More specifically, we propose an Incremental Space-Time-based model (ISTM) for repairing missing values in IoT real-time data streams. ISTM is based on Incremental Multiple Linear Regression, which processes data as follows: Upon data arrival, ISTM updates the model after reading again the intermediary data matrix instead of accessing all historical information. If a missing value is detected, ISTM will provide an estimation for the missing value based on nearly historical data and the observations of neighboring sensors of the default one. Experiments conducted with real traffic data show the performance of ISTM in comparison with known techniques.

Type of Paper and Keywords

Short communication: *IoT, Smart City, Missing value, Data Stream, Incremental Linear Regression*

1 Introduction

The Internet of Things (IoT) paradigm is still gaining attention among different stakeholders: practitioners, researchers or simply citizens due to the impact on their everyday life. An example of such impact is illustrated by the deployment of a network of sensors in a city for monitoring the city environment, particularly, the road traffic conditions, such as speed, congestion, pollution, accident, etc. Data emitted by sensors in real time are aggregated as a data stream, and serve as input for different IoT applications or services, such as traffic recommendation, urban planning etc.

Figure 1 illustrates a simplified version of a end-to-end data processing pipeline, that is from data collection to the ingestion by an application/service. An example of such service may be the IoT based traffic recommendation service (IoTTRS for short), which is one of the important services based on traffic sensor networks in smart cities [6]. IoTTRS covers tasks such as searching for parking, planning a trip, and finding a shared bike, leading to the improvement of the quality of life of urban residents, time and cost saving, etc.

Figure 1 also shows that some data may be unreachable (missing values), hence leading to incompleteness issues [11, 30]. The reasons can be various: 1) Sensor has failed, by a high winds, even by a traffic accident or by a quality problem ; 2) battery is exhausted; 3) wireless transmission is hacked, etc.

The missing IoT value can cause serious deviations in the IoTTRS system, giving false recommendations and negatively impacting the final decision.

According to [5, 30], there exist three categories of missing values : (a) missing completely at random (MCAR), (b) missing at random (MAR) and (c) not missing at random (NMAR).

Missing value repairing can be performed in adopting one of the four strategies [24]:

1. delete incomplete observations;

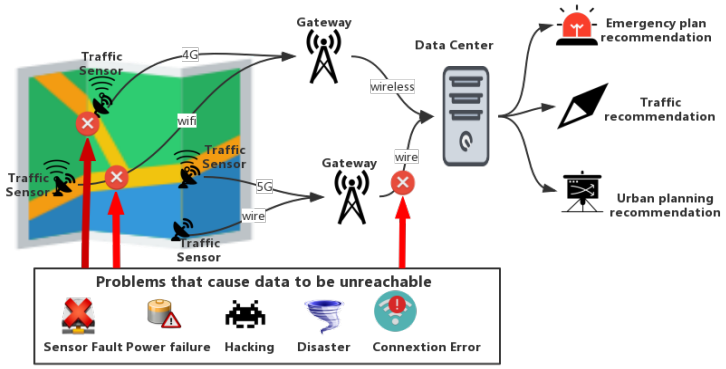


Figure 1: Overview of a Data Pipeline in a Sensor Network

2. manually repair;
3. substitute by a constant/last-observation/mean;
4. estimate the most probable value.

The first three strategies are not suitable for IoT data streams. Indeed, deleting incomplete information is the easiest way for repairing but may cause information loss. Manually repairing is too slow in the presence of a huge volume of data [10, 24]. Replacing missing values with a constant/last-observation/average is the most common method but it may lead to a biased estimation [5].

The last strategy, also called Imputation, does not need human intervention and is much more efficient than manual ones. Imputation retains incomplete information and uses as much information as possible from the gathered observations to repair missing values [24]. Imputation overcomes the limitations of the first three approaches and represents the driving direction of our research.

While reviewing several work that address the same problem [3, 8, 9, 14–16, 18, 20, 22, 25–28, 30, 31], we noticed that characteristics of IoT data streams (timeliness, non-stationary, etc.) are not well taken into account. Therefore, traditional data repairing methods cannot be directly applied to IoT data stream.

Hence, the work described in this paper addresses the missing data problem in the IoT context. More precisely, we describe ISTM, an Incremental Space-Time-based Model together with the imputation method that we developed for repairing missing values in an IoT real-time data stream, that is to say that we repair missing values at real-time.

ISTM repairing process can be summarized as follows: 1) Initialize the model with historical data; 2) Upon arrival of new data, update the model after reading two intermediary matrices instead of accessing all historical data; 3) If a missing data is detected, an estimation is calculated accordingly with a set of reference values related to the missing one.

The remainder of this paper is organized as follows: Section 2 reviews some related work devoted to missing IoT data repairing algorithms. In section 3, we describe our approach. Section 4 illustrates the experimental results obtained while comparing ISTM with existing methods. Finally, we conclude in section 5.

2 Related Work

In this section, we review the different approaches based on imputation methods that have been proposed in the literature to overcome the missing IoT data problem. Imputation methods can be based on static or incremental models.

2.1 Static Imputation

Classical missing data imputation methods are largely based on static models. The static model is trained by a fixed set of samples, and its parameters can not be changed afterwards. There is a plethora of static imputation methods devoted to IoT data which are based on K-nearest neighbors (K-NN) [15, 16, 30], Matrix Factorization [8], Regression [3, 18, 20], Neural Network [22, 26] and Multiple Imputation [2]. We review the related missing data imputation works for IoT according to these different models.

In [16], the authors propose to use time and geographic information to find the nearest neighbor sensor and validated this idea with a set of sensor data about air pollution. The work in [30] used Gaussian Mixture Model (MGI) and Expectation Maximization

(EM) for clustering the sensors. If a missing attribute of one data occurs, the proposed model will find the nearest neighbor in its corresponding cluster to replace this missing value.

The approach proposed in [8] splits the sensor into different clusters, each cluster associates a process of probabilistic matrix factorization (PMF) to recover the missing data. An experimental simulation has been made with sensor data and suggested that the proposed model outperformed support vector machine (SVM) and deep neural network (DNN). In [3], authors propose a Kernel Ridge Regression data imputation in the context of a sensor network, where new kernel function is used to enrich the dimensionality of training data.

In [18], a Spatial-Temporal model(STM) is proposed to repair data in a wireless sensor network and to improve prediction accuracy by establishing a Multiple Linear Regression (MLR) model for both spatial and temporal data.

In [2] authors created a set of candidates and used the data in a moving window to constantly adjust the component weight of candidates. However, it is clear that if every chunk of data serves as input for weights recalculation , the data center must ensure a strong computing power, especially as the number of sensors increases.

As mentioned above, static models have been widely used in the literature in order to deal with missing data. However, sensor data are often non-stationary and may change over time due to the concept drift [13] and the data evolution [32]. If concept drift or/and data evolution happens, the all mentioned static models may generate incorrect values [32], so they cannot be employed directly to deal non-stationary sensor data. In addition, [1] highlighted the importance of streaming analytic in real-time, because sensor data is time sensitive. Hence, we should adjust/adapt the imputation in real time [21], which is called Online Learning [32].

2.2 Incremental Imputation

Thus, the incremental model emerges [9, 14, 22, 27, 28], as one branch of Online Learning, which updates/adjusts the parameters of the existing model with the last incoming data instead of building a new model from scratch. Its advantages are obvious: 1) its prediction accuracy does not fall off the cliff; 2) it does not require backtracking historical data (or just recently historical data); 3) its computational complexity is small.

Unfortunately, the research on "repairing missing values of the sensor data stream by incremental model" is largely overlooked. According to a survey in [12], only the works [9, 14, 22, 27, 28] repair the missing data in a real-time sensor data stream in an incremental manner.

Then , we distinguish between two methods: "Kalman Filter" [4] and "Association Rule Mining" [17].

The works in [27] and [28] based on Kalman Filter (KF), tried to correct/impute the sensor data stream, by combining the observation measurements of different sources (e.g., by one sensor or by one model) in real time, so as to climate the noises. When a missing value occurs, KF can also give an estimate based on the user-specified underlying model (its parameters are always updated).

However, KF makes a strong assumption that the relationships between the previous state and the current state are known. Then, if we do not consider environmental noise and measurement errors, we can identify the current state from the last one.

In [14], authors proposed WARM (Window Association Rule Mining), which discovers that two sensors often generate the same data, by reviewing only the data in a window. When a missing value of one traffic light is detected, the actual observations of its similar sensors will be taken as reference data to estimate the missing one (e.g., by average). An extension of WARM has been proposed in [9] as Freshness Association Rule Mining (FARM), which record all the same behaviors in the data stream and provides the fresher data with a higher weight. This improvement is interesting because it provides FARM the ability to continuously update the traffic data.

FARM (and WARM) is only suitable for discrete values. In the case where the sensors produce continuous values, they have to be converted to discrete values. Such operation is called Discretization. Unfortunately, the optimization of discretization is a NP problem. We arbitrarily think that different discretizations will lead to different neighbors and different estimation.

It is also noteworthy that FARM makes use of the observations in the current round as reference data. This situation may raise two issues: 1) missing value may also appear within neighbors; 2) before estimating a value, we must wait for all neighbor's values.

As described above, incremental models are the most suitable methods to address the missing IoT data problem because they deal with the dynamic character of IoT data. In addition, regression imputation has several advantages such as: 1) it does not require user-defined parameters, 2) it can get the optimal global solution, 3) when the data obeys a linear relationship (e,g. temperature sensors in the same room), the effect will be excellent.

However, our study showed that there is no published work using the incremental multiple linear regression (IMLR) in the IoT missing data context which is the main contribution of this paper.

3 INCREMENTAL SPACE-TIME APPROACH

In this section, we describe ISTM, an incremental Space-Time-based model that we propose for repairing missing values. ISTM extends STM, the Space Time Model proposed in [18], in using incremental Multiple Linear Regression. So we will briefly introduce STM and we describe the process of the ISTM (offline and online).

3.1 Formal Description

In order to develop our imputation method, we provide a formal description of the missing data problem. Figure 2(a) depicts an IoT data stream which contain missing values: Given f sensors, each sensor will generate a value in each time point. Lost data is represented by ◯; and if one data reaches the computer center in time, it is marked as $\surd$.

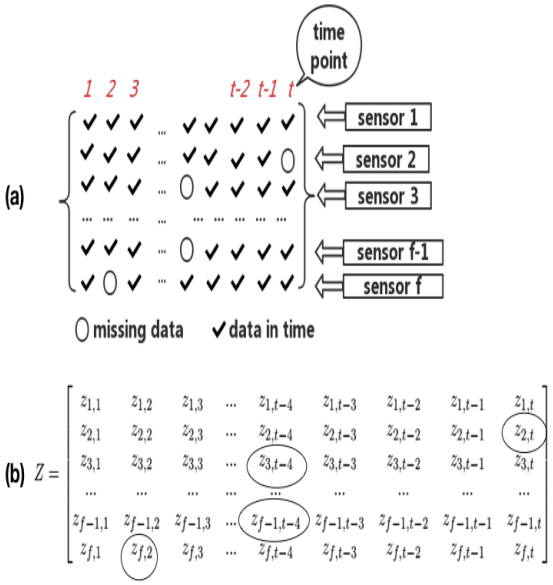


Figure 2: In (a), every sensor generates a data in each time point. If the data value is lost, it is marked as ◯; if not, it is marked as $\surd$; In (b), a matrix corresponding figure (a), if the data value is lost, it is placed in ◯

Missing values in IoT data stream can also be represented as a matrix Z (see figure 2(b)). Given f sensors, in last t time points, an element $z_{f',t'}$ in the matrix Z represents an expected data value generated by a sensor f' in the time point t'. The elements in the ◯ refer to the missing value.

Residual Minimization If a value emitted by a sensor f', at time t' is lost, (e.g. the data marked by ◯ in figure 2(a) or $z_{2,t}, z_{3,t-4}$... in figure 2(b)), an estimated value denoted $\hat{z}_{f',t'}$ will be generated, in place of $z_{f',t'}$. Estimation of $\hat{z}_{f',t'}$ is the solution to the problem of minimizing $|z_{f',t'} - \hat{z}_{f',t'}|$.

So the problem of repairing missing values can be represented as a 'Residual Minimization' problem.

3.2 STM

In [18], a Spatial-Temporal model(STM) is proposed to repair data in a wireless sensor network and to improve prediction accuracy by establishing a Multiple Linear Regression (MLR) model for both spatial and temporal data. The idea is to make two regressions: one based on the sensors neighborhood and the other one on the previous data of some data source, the weighted average of the two will being the final model.

If the value $z_{f',t'}$ of a sensor f' at time t' is missing, STM needs to:

- First, get $\hat{z}^{SM}_{f',t'}$ by Linear Regression with the observations of its neighbors(the greens, called S shown in figure 3).
- Second, get $\hat{z}^{TM}_{f',t'}$ by Linear Regression with observation of nearly time points (denoted T and circled in red in figure 3).
- Finally, compute a weighted sum($\hat{z}^{STM}_{f',t'}$) according to equation 1.

$$\begin{aligned} \hat{z}^{STM}_{f',t'} &= w_S * \hat{z}^{SM}_{f',t'} + w_T * \hat{z}^{TM}_{f',t'} \\ 0 \leq w_S, w_T \leq 1, w_S + w_T &= 1 \end{aligned} \tag{1}$$

However, in a scenario involving real-time traffic data, STM cannot be directly applied because of the following reasons:

- STM needs some reference data. In some cases, a user/application has to wait for this reference data, because it may arrive later than the detection time of a missing value.
- There is no guarantee about the availability of all required reference data. These data may also be lost.
- STM makes the assumption that data is stationary. And, it is not easy to update the model. More specifically, if some new data arrive, STM needs to access again all historical data to update all parameters, hence resulting in a costly process.

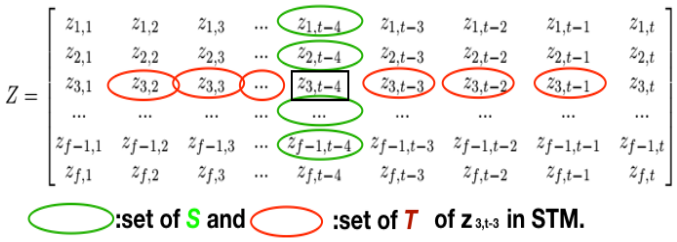


Figure 3: Dataset reference for STM (Repairing $z_{3,t-1}$ missing value) in the data matrix)

It is clear that once the $z^{SM}_{f',t'}$ and $z^{TM}_{f',t'}$ are known, so are w_S and w_T. If some new data arrive, $z^{SM}_{f',t'}$ and $z^{TM}_{f',t'}$ will be modified (maybe by an incremental method which does not consume a lot of resources), but they still have to read again all historical data in order to recalculate w_S and w_T, hence resulting in a costly process.

3.3 Reference Data in ISTM

Definition The reference dataset ($rd_{f',t'}$) of a missing value($z_{f',t'}$) consists of the last g observations of sensor ($f^{'}$) that causes the missing value at (current) time ($t^{'}$) and the observations at time $t^{'}-1$ of its neighbors sensors ($K_{f'}$ and $|K_{f'}| = r$) as equation 2 and figure 4. We note $P = 1 + g + r$.

The definition of the neighbors of one sensor can be based on spatial location or on other measures. Value g is predefined by the user.

Reference datasets of ISTM and STM (Fig. 4 and 3) differ as follows:

- Given one missing value of senor$f^{'}$ at time $t^{'}$, $ISTM$ does not take into account reference data after time point $t^{'}$, because those data are not available for a real-time process.
- $ISTM$ consider the observations of the neighbors at time $t' - 1$ as the reference dataset, instead of those at t'. In doing so, we avoid the problem of "*Missing value in the reference dataset*".
- $ISTM$ does not separate the reference data like S and T in STM. All the reference data are passed to the incremental MLR.

$$rd_{f',t'} = [1, z_{f',t'-1}...z_{f',t'-g}, z_{k_1,t'-1}, ..., z_{k_r,t'-1}]_{1*P} \quad (2)$$
$$where\ [k_1, k_2, ..., k_r] = K_{f'}$$

3.4 ISTM Processing Model

ISTM is based on Incremental Multiple Linear Regression (IMLR) [23, 29] and has three processes: Initialization (offline phase), Estimation (online phase) and Update (online phase).

Initialization Given one sensor f' with n, historic data as a matrix $Y_{f'}$, and its recording reference data as $X_{f'}$ are presented as follows:

$$Y_{f'} = \begin{bmatrix} rd_{f',t'_1} \\ rd_{f',t'_2} \\ ... \\ rd_{f',t'_{n-1}} \\ rd_{f',t'_n} \end{bmatrix}_{P*1} \quad X_{f'} = \begin{bmatrix} z_{f',t'_1} \\ z_{f',t'_2} \\ ... \\ z_{f',t'_{n-1}} \\ z_{f',t'_n} \end{bmatrix}_{n*P} \quad (3)$$

We can compute two intermediary matrix $X^T_{f'}X_{f'}$ and $X^T_{f'}Y_{f'}$ [23, 29].

The coefficients of a model for the sensor $f^{'}$ [23, 29] is:

$$B_{f'} = \left(X^T_{f'}X_{f'}\right)^{-1} X^T_{f'}Y_{f'} \quad (4)$$

Estimation If one missing value is detected, as depicted with the red path in Fig. 5, the model generates an estimation at real time referring to some data in the reference database. This estimation will be stored in the reference database like a real value. The oldest data may be removed for saving space

The formal description of this situation is as follows: when the value $z_{f',t'}$ is lost, and if we have its reference data $rd_{f',t'}$, we can calculate the estimation function of ISTM $\hat{z}_{f',t'}$ as defined in the equation 5:

$$\hat{z}_{f',t'} = rd_{f',t'} \cdot B_{f'} \quad (5)$$

Updating If some data issued by a sensor arrives in time at its computer center (corresponding to the blue path in Fig. 5, the dynamic model is updated with the new data. Then, it is stored in a reference database. The

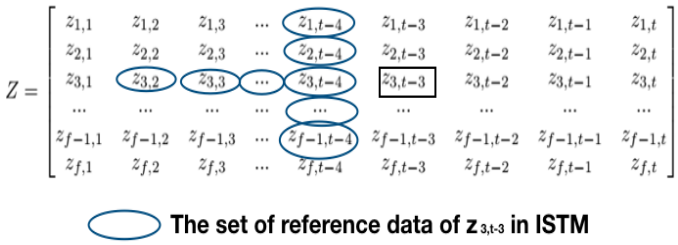


Figure 4: The reference data set of ISTM (Repairing $z_{3,t-1}$ missing value) in data matrix, corresponding figure 3)

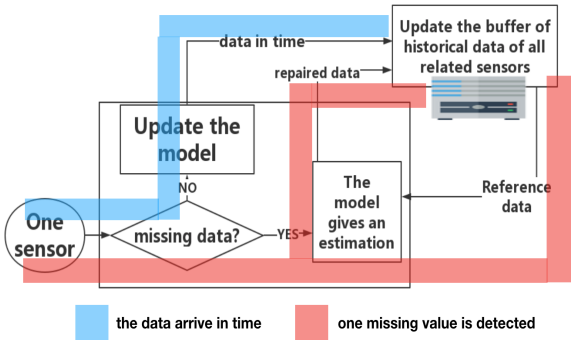


Figure 5: ISTM online processes: 1) Update: upon data arrival at (expected) time, update ISTM, 2) Estimation: one missing value is detected, one estimation is generated.

oldest data in the reference database may be deleted to save storage space.

The formal description is as follows: if $z_{f',t'}$ arrives in time, its reference data $rd_{f',t'}$ will be used to update the model. The updating functions (re-calculating the two intermediary matrix) are presented by equations 6, 7 and 8 [23, 29]:

$$X_{f'}^T X_{f'} \leftarrow X_{f'}^T X_{f'} + \left(rd_{f',t'}\right)^T rd_{f',t'} \quad (6)$$

$$X_{f'}^T Y_{f'} \leftarrow X_{f'}^T Y_{f'} + \left(rd_{f',t'}\right)^T z_{f',t'} \quad (7)$$

$$B_{f'} \leftarrow \left(X_{f'}^T X_{f'}\right)^{-1} X_{f'}^T Y_{f'} \quad (8)$$

4 Experiments

In this section, we present the experimental results that we obtained with real data provided by CityPulse [1].

[1] http://www.ict-citypulse.eu/

4.1 Experimental Data Description

To evaluate ISTM, we use CityPulse data consisting of the speed of cars. CityPulse data set covers seven different domains: namely, Road Traffic, Parking, Pollution, Weather, Cultural, Social and Library Events Data of Aarhus, Denmark and Brasov, Romania for years 2014 and 2015. Among all these parts, Road Traffic Data is of greatest importance.

Data Set Volume Road Traffic Data are real-world data about travel information of Aarhus (Danmark) during the following periods: "2/2014 - 6/2014","8/2014 - 9/2014", "10/2014 - 11/2014"," 07/2015 - 10/2015". There is a total of 449 monitors (assuming that one sensor was installed in one area). The volume of the data in format CSV is 747.2 MB.

One bunch every 5 minutes Traffic Data is collected by many sensors installed on the road. Every 5 minutes, each sensor will send a bunch of information (one line of table Traffic Data) to a central computer center. Every 5 minutes the center receives 29,940 Bytes (0.029MB).

Real Missing Value Fig. 6 illustrates a sample data for one sensor with one missing value at timestamp "2014-02-13T11:50:00" (between timestamps "2014-02-13T11:45:00" and "2014-02-13T11:55:00"). The total missing value rate is close to 9%.

Simulated Missing Value We simulate some values randomly (5%, 10%, 15%, 20%, 25%, 30%) and mark them as Simulated Missing Values(SMV). Thanks to SMV and its ground truth, we can measure the effectiveness of the reparation. The percentage rate of simulated missing values is borrowed from works [7, 9, 19] where the percentage of missing value or simulated missing value vary from 5% to 30%. There are both real missing values and simulated missing value in our test data.

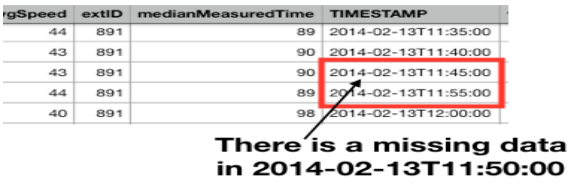


Figure 6: CityPulse Missing Values

4.2 Evaluation

We evaluate the performance of our model in terms of MSE accuracy:

$$MSE = \frac{1}{|SMV|} \sum_{z_{f',t'} \in SMV} \left(\hat{z}_{f',t'} - z_{f',t'}\right)^2 \quad (9)$$

We compare ISTM with some existing models for repairing missing values.

- The $Average$ and $PreviousValue$ are the naive methods, but they can be easily applied in a data stream context.
- $LinearRegression$ uses the same reference data set as ISTM, but without an no Update phase. By comparing with $LinearRegression$, we can show the adaption of ISTM in the case of concept drift.
- FARM, which is mentioned in section 2, which can also update the model with an incremental manner.

Not ethat we do not compare our model with Kalman Filter, because we can not satisfy the Kalman Filter assumption which is that the relationships between the previous state and the current state are known.

4.2.1 Configurations

The configurations of the methods are described as follows:

ISTM Given one sensor, neighbors' sensors within 1 km around are considered as its neighbors. The reference data set review 6 last historical data of self-sensor and previous data of its neighbors. 30% of instances are taken for initiation.

Average The missing value of one sensor is replaced by the average of its all historical data.

Previous Value The missing value of one sensor is replaced by its previous value.

Linear Regression Similar to ISTM, but without an update step.

FARM Continuous data will be converted to discrete values. The chosen discretization is the best discretization from 848 randomly generated. 30% of the data set are used for initiation. For a missing value at timepoint t', the weight function of a value in timepoint i is $0.999^{t'-i}$.

4.2.2 Results

In this subsection, we will discuss the accuracy and the time cost of all the mentioned methods.

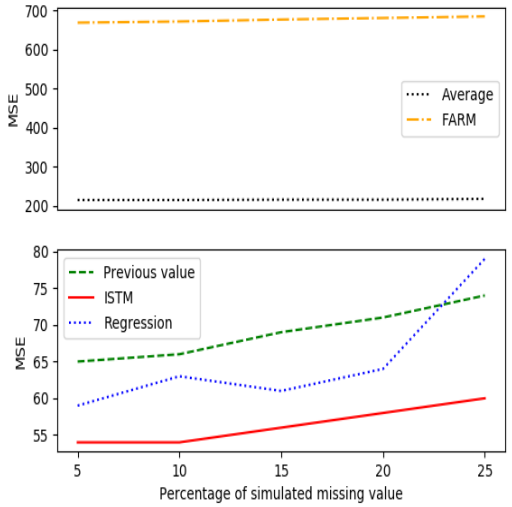


Figure 7: MSE Evaluation with different methods. ISTM red curve has the best MSE (the lower is better)

Accuracy As the proportion of missing values increases, the SSE value of all algorithms increases(worse). This is because all of the above methods rely on historical data. If the quality of the historical data decreases (although it has already been repaired), the quality of the prediction/repair will be reduced. It can be speculated that directly applying data with missing values (without being repaired) to commercial activities may bring intolerable deviations/losses.

Figure 7 shows that, when the ratio of simulated missing value (SMV) is equal to 5%, MSE of FARM is much higher than the other methods. Although, others growths are obvious, FARM growth is not obvious, but the MSE of FARM is still higher than other methods.

The curve(MSE) of $Average$ is like FARM in terms of stability. $Average$ is just better than FARM and worse than all others.

Figure 7 also shows that the curve (MSE) of ISTM is smaller than any other model at all missing value ratios. In other words, ISTM has the highest precision (its MSE is the smallest). Indeed, the MSE of ISTM is always lower than 60, which means that the expected error is under$\sqrt{60} \approx 7.5$. Considering that the speed of the car is between 0 and 120, we arbitrarily think that it's an outstanding performance. In short, ISTM has an excellent performance in both relative and absolute terms.

When the ratio of SMV is equal to 5% and 10%, the performance of $Regression$ and of $PreviousValue$ are close to ISTM. When the proportion of missing values increases, the difference between the best and the worst cases for ISTM is only around 6. On the other hand, the growth rate of MSE of $Regression$ and of $PreviousValue$ are significantly higher than ISTM. The difference between the worst and the best of $PreviousValue$ (respectively $Regression$) is equal to 19 (respectively 20).

To summarize, in terms of accuracy, 1) $Regression$ and $PreviousValue$ are more suitable for situations where the ratio of missing values is low rather than high. 2) The performance of ISTM is more stable than $Regression$ and $PreviousValue$.

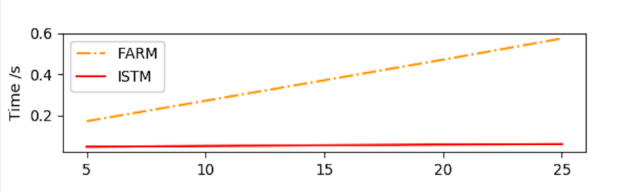


Figure 8: The time consumed of $ISTM$ and $FARM$ (one ground)

Time Evaluation In order to evaluate the performance in terms of time consumed, we compare ISTM and $FARM$ because $FARM$ is the only incremental system. So we do not discuss the performance of $Average$, $Regression$ and $PreviousValue$.

As illustrated in Fig. 8, ISTM and $FARM$ spend much smaller time (for one ground) than the interval of two grounds (5 minutes). When the ratio of SMV is equal to 5%, $FARM$ needs just 0.17 seconds for a ground. When the ratio of SMV is equal to 25%, the worst time cost is nearly 0.6 second. The reason for this is that the estimation function will be called many times as the proportion of missing values increases. ISTM performs better than $FARM$. When the ratio of SMV is equal to 5%, its time cost is close to 0.047 second for a ground. Furthermore, when the proportion of missing values increases, the time cost of ISTM increased slightly. Its worst time cost (with 25% SMV) is 0.06 s, which is much more stable than $FARM$. This means that ISTM is suitable for data with a high ratio for missing data.

5 Conclusion

In this article we described ISTM, an Incremental Spatio-Temporal regression method for repairing missing values in IoT data streams. ISTM is based on Incremental Multiple Linear Regression (IMLR) while taking into account spatial and temporal features. ISTM has been implemented and tested in using real traffic data provided by CityPulse. Experimental results show that ISTM outperforms some traditional methods in terms of accuracy and efficiency.

However, experiments show also that ISTM may need more computation time, mainly for updating the model. This is a "slight price to pay" that could be minimized if, for instance, we activate the *update function* only at mandatory stages.

For the future, we are considering other data quality dimensions such as real-time "outlier detection" in an IoT data stream. Improving ISTM in adapting other methods such as Incremental SVM or Incremantal Neural Network are definitely research directions to consider.

References

[1] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in Internet of Things," *Computer Networks*, 2017.

[2] I. Azimi, T. Pahikkala, A. M. Rahmani, H. Niela-Vilén, A. Axelin, and P. Liljeberg, "Missing data resilient decision-making for healthcare iot through personalization: A case study on maternal health," *Future Generation Computer Systems*, vol. 96, pp. 297–308, 2019.

[3] B.-W. Chen, S. Rho, L. T. Yang, and Y. Gu, "Privacy-preserved big data analysis based on asymmetric imputation kernels and multiside similarities," *Future Generation Computer Systems*, 2018.

[4] C. K. Chui, G. Chen *et al.*, *Kalman filtering*. Springer, 2017.

[5] M. C. de Goeij, M. van Diepen, K. J. Jager, G. Tripepi, C. Zoccali, and F. W. Dekker,

"Multiple imputation: dealing with missing data," *Nephrology Dialysis Transplantation*, vol. 28, no. 10, pp. 2415–2420, 2013.

[6] S. Di Martino and S. Rossi, "An Architecture for a Mobility Recommender System in Smart Cities," *Procedia Computer Science*, 2016.

[7] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[8] B. Fekade, T. Maksymyuk, M. Kyryk, and M. Jo, "Probabilistic Recovery of Incomplete Sensed Data in IoT," *IEEE Internet of Things Journal*, 2018.

[9] L. Gruenwald, H. Chok, and M. Aboukhamis, "Using Data Mining to Estimate Missing Sensor Data," in *2007 Seventh IEEE International Conference on Data Mining - Workshops (ICDM Workshops)*. IEEE, 2007, pp. 207–212.

[10] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[11] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in internet of things A state of the art survey," *Journal of Network and Computer Applications*, 2016.

[12] A. Kejariwal, S. Kulkarni, and K. Ramasamy, "Real Time Analytics: Algorithms and Systems," *Proceedings of the VLDB Endowment*, 2017.

[13] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, 2017.

[14] M. H. Le Gruenwald, "Estimating missing values in related sensor data streams," in *12th International Conference on Management of Data*, 2005.

[15] Q. Ma, Y. Gu, F. Li, and G. Yu, "Order-sensitive missing value imputation technology for multi-source sensory data," *Journal of Software*, vol. 27, no. 9, pp. 2332–2347, 2016.

[16] I. P. S. Mary and L. Arockiam, "Imputing the missing data in IoT based on the spatial and temporal correlation," in *2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, 2017.

[17] G. Modi, S. Bansal, and M. A. Patidar, "A survey on sequential rule mining techniques," *International Journal For Technological Research In Engineering*, vol. 6, no. 3, 2018.

[18] L. J. PAN Liqiang, LI Jianzhong *et al.*, "A multiple-regression-model-based missing values imputation algorithm in wireless sensor network," *Journal of Computer Research and Development*, vol. 33, no. 1, pp. 1–11, 2010.

[19] D. Puiu, P. Barnaghi, R. Tönjes, D. Kümper, M. I. Ali, A. Mileo, J. X. Parreira, M. Fischer, S. Kolozali, N. Farajidavar *et al.*, "Citypulse: Large scale data analytics framework for smart cities," *IEEE Access*, vol. 4, pp. 1086–1108, 2016.

[20] T. E. Raghunathan, J. M. Lepkowski, J. Van Hoewyk, and P. Solenberger, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Survey methodology*, vol. 27, no. 1, pp. 85–96, 2001.

[21] R. Ranjan, O. Rana, S. Nepal, and M. Y. I. Cloud, "The Next Grand Challenges: Integrating the Internet of Things and Data Science," *IEEE Cloud Computing*, 2018.

[22] P. P. Rodrigues and J. Gama, "Online prediction of streaming sensor data," in *Proceedings of the 3rd international workshop on knowledge discovery from data streams (IWKDDS 2006), in conjuntion with the 23rd international conference on machine learning*, 2006.

[23] M. Schleich, D. Olteanu, and R. Ciucanu, "Learning linear regression models over factorized joins," pp. 3–18, 2016.

[24] R. Somasundaram and R. Nedunchezhian, "Evaluation of three simple imputation methods for enhancing preprocessing of data with missing values," *International Journal of Computer Applications, Vol21*, vol. 21, no. 10, 2011.

[25] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[26] H. Turabieh, A. A. Salem, and N. Abu-El-Rub, "Dynamic L-RNN recovery of missing data in IoMT applications," *Future Generation Computer Systems*, 2018.

[27] N. Vijayakumar and B. Plale, "Missing Event Prediction in Sensor Data Streams Using Kalman Filters," in *Knowledge Discovery from Sensor Data*, 2009.

[28] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation,"

in *Symposium on Adaptive Systems for Signal Processing Communications and Control*, 2000.

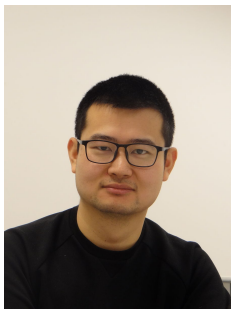
[29] H. L. Wang Huiwen, Wei Yuan, "Incremental algorithm of multiple linear regression model," *Journal of Beijing University of Aeronautics and Astronsutics*, vol. 40, no. 11, pp. 1487–1491, 2014.

[30] X. Yan, W. Xiong, L. Hu, F. Wang, and K. Zhao, "Missing value imputation based on gaussian mixture model for the internet of things," *Mathematical Problems in Engineering*, 2015.

[31] X. Zhou, X. Wang, and E. R. Dougherty, "Missing-value estimation using linear and non-linear regression with Bayesian gene selection," *Bioinformatics*, 2003.

[32] I. Zliobaite and B. G. I. t. o. k. and, "Adaptive preprocessing for streaming data," *IEEE Transactions on Knowledge and Data Engineering*, 2014.

AUTHOR BIOGRAPHIES



PENG Tao is a Ph.D. candidate in the Aix Marseille Univ, Marseille, France. He received his B.S. in Univ of Shanghai for Science and Technology, MBA in East China Normal Univ, M.S. in Aix Marseille Univ. His Research Interests include IoT, data quality,context-aware, recommendation. More specifically, Recommending IoT-based data and services, considering context and data quality.

Biographies with short text should use the latex command: *shortbio*{*photo_file*}{*text of bio here*}. However, a short biography should not be less than 70 words.

Biographies with short text should use the latex command: *shortbio*{*photo_file*}{*text of bio here*}. However, a short biography should not be less than 70 words.