HAL open science

# IoT Data Repairing with Incremental Multiple Linear Regression

Tao Peng, Sana Sellami, Omar Boucelma

**To cite this version:**

Tao Peng, Sana Sellami, Omar Boucelma. IoT Data Repairing with Incremental Multiple Linear Regression. BDA 2018 34ème Conférence sur la Gestion de Données – Principes, Technologies et Applications, Oct 2018, Bucarest, Romania. hal-02092757

**HAL Id: hal-02092757**

**https://hal.science/hal-02092757v1**

Submitted on 16 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# IoT Data Repairing with Incremental Multiple Linear Regression*

†

Tao Peng
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
tao.peng@etu.univ-amu.fr

Sana Sellami
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
sana.sellami@univ-amu.fr

Omar Boucelma
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
omar.boucelma@univ-amu.fr

## ABSTRACT

In this paper we address the problem related to data completeness in the IoT domain. More specifically, we propose an Incremental Space-Time-based (ISTM) model for fast repairing missing values in an IoT real-time data stream. ISTM is based on Incremental Multiple Linear Regression, which processes data as follows: upon arrival of new data, ISTM updates quickly the model after reading again an intermediary matrix instead of accessing historical data. If a missing value is detected, ISTM will provide an estimation for the missing value based on historical data and the observation of sensors surrounding the one responsible for missing value(s). The paper also presents the performance studies in comparing ISTM with existing techniques using real traffic data.

## CCS CONCEPTS

• **Information systems** → **Sensor networks**; **Data streaming**;
• **Computing methodologies** → **Learning linear models**;

## KEYWORDS

IoT, Data Quality, Data Repairing, Linear Regression

**ACM Reference Format:**
Tao Peng, Sana Sellami, and Omar Boucelma. 2018. IoT Data Repairing with Incremental Multiple Linear Regression: . In *Proceedings of 34éme Conférence sur la Gestion de Données âĂŞ Principes, Technologies et Applications (BDA2018).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

With the advent of the Internet of Things (IoT), we are witnessing a proliferation of sensors, with a large scale deployment in cities. Data emitted by sensors enable the inference of Knowledge [7] in various domains such as traffic conditions or the management of natural resources in urban environments.

IoT data in general, and sensor data in particular, need to be handled with quality-awareness, i.e., data quality issues must be addressed in order to infer reliable knowledge. For sensor data, incompleteness (characterized by means of missing values) is an important quality dimension to deal with. Machine learning algorithms using datasets with missing values may generate bias because most of them make the assumption that input data is curated (e.g., with no missing value) [19].

As described in [19], there exist four approaches that address missing values problem: 1) delete incomplete observations; 2) manually repairing; 3) Substitute by a constant/mean; 4) get most probable value to fill in the missing values. Deleting incomplete observations is, obviously, the easiest way for handling missing values [22] but at a heavy cost since one may loose useful information. Manually repairing may be hard to achieve and may contradict IoT's philosophy that is, extracting meaning without human intervention [7, 19]. Replacing missing values with the average/last-observation is the most common method but leads to biased estimates [5]. The last approach known as imputation, which is the most popular one [5, 11, 16, 17, 19, 20], uses as more information as possible from the observed data to predict missing values [8].

There are three major types of Imputation methods: (1) k-nearest neighbors algorithm (k-NN)[17, 21], (2) Regression imputation and (3) Multiple imputation [6]. Regression imputation can be linear, logistic, Poisson, or a combination of the three methods [18]. Another method, Multiple Linear Regression (MLR) is widely used for forecasting. In general, the whole set of data is used as a training set of MLR, and missing values are estimated by MLR with its independents variables. For example, model proposed in [15] improves prediction accuracy by establishing a MLR model for both spatial and temporal data.

In the IoT context, one may manipulate data streams: hence, continuously training of MLR with new data may improve the accuracy of forecasting. Meanwhile, without modifying the original model, we need to read all new and old data at one time to completely retrain the model. Usually the volume of historical data may be very huge, for example, on Internet every day around 2.5 quintillion bytes of data are created [2]. It is obviously expensive (slow) to read all historical data into memory at one time once new data arrive. If the data streams are not permanently stored [11], it is impossible to read all historical data needed by the model. So updating the model of repairing missing value with data stream in real time is a new challenge.

In this article, we make the MACR assumption (Missing Completely at Random), that means the underlying reasons for missing values are independent of known (or unknown) data characteristics [5]. We propose to repair missing values (of data stream) in real time via incremental MLR (IMLR) [24], which can significantly increase

*

†acmart.pdf document

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
*BDA2018, 22-26 octobre 2018, Bucarest*
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

learning efficiency without sacrificing accuracy. Incremental learning is considered as an effective method [25]. Finally, we propose the incremental Space-Time-based model (ISTM) approach which can rapidly update the model with new data. ISTM re-reads one intermediary matrix instead of all historical data.

Our findings show that ISTM has a good performance in repairing accuracy and overall, performs better than existing techniques.

The remainder of this paper is organized as follows: In section 2, we formalize the problem and present our approach. In section 3 we describe the results of our solution and compare it with existing methods. In section 4, we describe some related work on IoT data repairing. Finally, in section 5, we conclude and highlight some future directions.

## 2 INCREMENTAL SPACE-TIME MODEL APPROACH

In this section, we describe ISTM, an incremental Space-Time-based model that we proposed for repairing missing values. ISTM extends STM, the Space Time Model proposed in [15], in using incremental Multiple Linear Regression.

### 2.1 Modeling missing values with ISTM

The problem of repairing missing values can be represented as matrix $Z$ as illustrated in Figure 1. Given $p$ sensors, each element $z_{i,t}$ represents a data value generated by a sensor $i$ at time $t$. If the value emitted by sensor $i$, at time $t$ is lost, then a value that we denote $\hat{z}_{i,t}$ will be generated in place of $z_{i,t}$. Estimation of $\hat{z}_{i,t}$ is the solution to the problem of minimizing $|z_{i,t} - \hat{z}_{i,t}|$.

The main feature of ISTM is to avoid reading all the historical data. Indeed, STM [15] adapts the multiple linear regression model to estimate missing data both along the temporal dimension and the spatial one, and get the weighted average of the two values as the final estimation.

If the value $z_{p',t'}$ of a sensor $p'$ at time $t'$ is missing, STM needs to get an estimation $\hat{z}^S_{p',t'}$ depending on the observation (called SM) of its neighbors and also calculates another estimation $\hat{z}^S_{p',t'}$ depending on its observation of nearly time points (called TM) as illustrated in Figure 1. Then a weighted sum of the two is given as the final estimation as illustrated by the following formula:

$$\begin{aligned} \hat{z}^{ST}_{p',t'} &= w_S * \hat{z}^S_{p',t'} + w_T * \hat{z}^T_{p',t'} \\ 0 \le w_S, w_T &\le 1, w_S + w_T = 1 \end{aligned} \tag{1}$$

### 2.2 ISTM vs. STM

It is demonstrated that, for repairing values, STM performs better than SM and TM [15]. However, STM is difficult to adapt to incremental because the weights $w_S$ and $w_T$ are determined by their performances in historical data. It is clear that once the SM and TM models are established, $w_S$ and $w_T$ are also determined. If some new data arrive, SM and TM will be modified (maybe by one 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.

Compared to STM, ISTM has the following features:

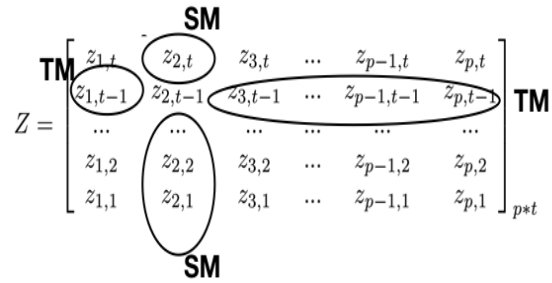


**Figure 1: STM Repairing Illustrated ($z_{2,t-1}$ missing value)**

*Replacement of $t'$ by $t'-1$.* Considering the correlation between the observation of one sensor (link) at time $t'$ and those of its neighbors at the immediate past time, *SM* can be estimated with the value of observations of the neighbors at time $t'$ rather than those at $t'-1$. There is an additional benefit: avoid the problem of "*The missing value in its surrounding sensors*", because all the missing values at the previous time points have been already repaired.

*One computing pass.* ISTM does not calculate separately $\hat{z}^S_{p',t'}$ and $\hat{z}^T_{p',t'}$. The values of the neighbors at time $t'-1$ and its historical data observed between from time $t'-1$ to $t'-g$ are both provided as input to incremental MLR. In doing so, we avoid to calculate $w_S$ and $w_T$ which requires reading all historical data.

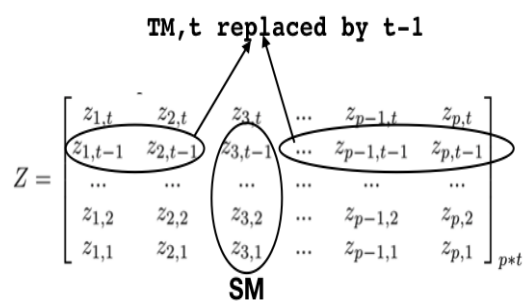


**Figure 2: ISTM Repairing Illustrated (missing value $z_{3,t}$)**

### 2.3 ISTM Repairing Function

As discussed above, the repairing process consists mainly on the replacement of the missing value by an estimated one. The estimation function of ISTM can be represented as follows:

$$\begin{aligned} \hat{z}^{ST}_{p',t'} &= w_0 + w_1 z_{p',t'-1} + w_2 z_{p',t'-2} + \ldots + w_g z_{t'-g} \\ &+ w_{g+1} z_{k1,t'-1} + w_{g+2} z_{k_2,t'-1} + \ldots + w_{g+q} z_{k_q,t'-1} \\ &= w_0 + \sum_{i=1}^{g} w_i * z_{p',t'-i} + \sum_{j=g+1}^{q+g} w_i * z_{j,t'-1} \end{aligned} \tag{2}$$

If we note $U_{p',t'}$ as:

$$U_{p',t'} = \begin{bmatrix} 1 & z_{p',t'-1} & \dots & z_{p',t'-g} & z_{k1,t'-1} & \dots & z_{kg,t'-1} \end{bmatrix}_{1*(1+g+q)} \quad (3)$$

and $W = \begin{bmatrix} w_0 & \dots & w_{g+q} \end{bmatrix}^T_{1*(1+g+q)}$, where $q$ is the number of neighbors.

So, ISTM function can also be represented as:

$$\hat{z}^{ST}_{p',t'} = W^T U_{p',t'} \quad (4)$$

Figure 3 illustrates the repairing process. If ISTM receives the right message in a timely manner (considered as a normal/good message), then ISTM uses the message to train/update the model in real-time. If this message is not received in time (which is the case of a missing value), ISTM will generate an estimation that substitutes this missing value. The buffer in Figure 3 represents the historical data.

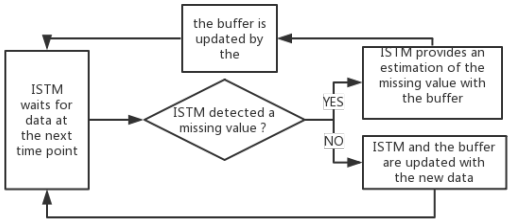


**Figure 3: ISTM Processes**

## 3 EVALUATION

In this section, we evaluate the complexity of our incremental model and we present the experimental results on a real dataset. The evaluation is made according the following assumptions made in [15, 17, 19, 20]:

(1) $n \gg P, m \gg P$,
(2) $P \gg n, P \gg m$,
(3) $n \gg P, n \gg m$

where $n$ denotes the old sample size, $P$ is number of feature values and $m$ is new sample size of one unit time.

Table 1 below summarizes those assumptions.

### 3.1 Time and Space Complexity

We calculate the complexity of IMLR and we compare it with MLR. It is worth emphasizing that we only care about the complexity with respect to updating the model.

If we assume that $n$ is much larger than $P$ and the $m$ like in the article [15, 17, 19, 20], This situation is recorded as $n \gg P, n \gg m$ in Table 1.

If we assume that the number of sensor is large enough, the $n$ and $m$ may be much larger than $P$ too, like in the article [15, 17], which is represented as $n \gg P, m \gg P$ in Table 1.

| | | Original | | Incremental |
|---|---|---|---|---|
| | * | $O\left((n+m)P^2 + P^3\right)$ | | $O\left(mP^2 + P^3\right)$ |
| | ** | $O\left(2(n+m)P + 2P^2\right)$ | | $O\left(2mP + 4P^2\right)$ |
| $n \gg P, m \gg P$ | * | $O\left((n+m)P^2\right)$ | $>$ | $O\left(mP^2\right)$ |
| | ** | $O\left(2(n+m)P\right)$ | $>$ | $O\left(2mP\right)$ |
| $P \gg n, P \gg m$ | * | $O\left(P^3\right)$ | $=$ | $O\left(P^3\right)$ |
| | ** | $O\left(2P^2\right)$ | $<$ | $O\left(4P^2\right)$ |
| $n \gg P, n \gg m$ | * | $O\left(nP^2\right)$ | $>$ | $O\left(mP + P^2\right)$ |
| | ** | $O\left(2nP\right)$ | $>$ | $O\left(2mP + 4P^2\right)$ |

* this is the Time Complexity
** this is the Space Complexity

**Table 1: Time Complexity, Space Complexity of IMLR and MLR in 3 cases**

We are also interested in the other extreme opposite case, because it can depict when there are not many sensors and not much historical data (possibly because the system just started or only stores very little history data), but with more features value. This case is represented as $P \gg n, P \gg m$ in Table 1.

In the first case $n \gg P, m \gg P$ and in third case $n \gg P, n \gg m$, the time complexity and space complexity of IMLR are generally better than its traditional version. But, in the second case ($P \gg n, P \gg m$), the incremental does not outperform MLR. When the number of features is too large, and the number of samples is too small, the incremental version does not improve the time complexity; moreover, it consumes more memory.

### 3.2 Dataset Description

In order to evaluate ISTM, we use CityPulse [1] data consisting of average speed of cars. CityPulse dataset covers seven different domains: Road Traffic, Parking, Pollution, Weather, Cultural, Social and Library Events Data of Aarhus, Denmark and Brasov, Romania for years 2014 and 2015. Road Traffic Data is the most important part.

Road Traffic Data save real world data of travel information during "2/2014 - 6/2014","8/2014 - 9/2014", "10/2014 - 11/2014"," 07/2015 - 10/2015", in City of Aarhus, Denmark. The total number of monitors is 449 (assume that one sensor in one area). The volume of the data in format CSV is 747.2 MB.

Traffic Data are collected by a number of sensors installed in the road. Every five minutes, each sensor will send a bunch of information (one line of table of Traffic Data) to a central computer center. If one line needs 60 Bytes, every 5 minutes the center will receive 29940 Bytes (0.029MB).

Figure 4 illustrates a sample of data for one sensor with one missing value at "2014-02-13T11:50:00" (between "2014-02-13T11:45:00" and "2014-02-13T11:55:00"). The frequency of missing value is close to 9%.

### 3.3 Implementation and Results

We set that the model can read data from the last 6 time points noted as $g = 6$. In fact, the bigger g is, the greater is the effectiveness of the model, but $g$ is limited by the computing center capacities.

| status | avgMeasuredTime | avgSpeed | extID | medianMeasuredTime | TIMESTAMP |
|---|---|---|---|---|---|
| OK | 89 | 44 | 891 | 89 | 2014-02-13T11:35:00 |
| OK | 90 | 43 | 891 | 90 | 2014-02-13T11:40:00 |
| OK | 90 | 43 | 891 | 90 | 2014-02-13T11:45:00 |
| OK | 89 | 44 | 891 | 89 | 2014-02-13T11:55:00 |
| OK | 98 | 40 | 891 | 98 | 2014-02-13T12:00:00 |

**Figure 4: CityPulse Missing Values**

Given a sensor, devices (sensors) within in 1 km around are considered as its neighbors, and hence are related to it.

So our application of ISTM falls into the following case ($n \gg P$,$n \gg m$): we have a large amount of old data (big $n$), $P$ is a small constant ($P = q + g$) where $q$ is the number of neighbours and $g$ is number of time points recently that our model can review, and the model is updated immediately with each new data arrival.

For measuring the effect of the reparation, we have to know some ground truth, i.e., the original value. So we randomly simulate some missing values from the good ones as illustrated in Figure 5, called Simulated Missing Values(SMV).

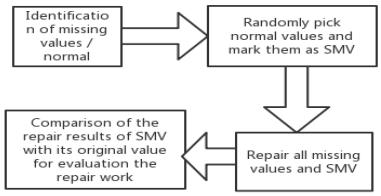


**Figure 5: Process of simulation of Missing Values and evaluation of repair work**

The Sum Squared Residual (SSE) of SMV is used to evaluate the performance of accuracy of ISTM and we compare it with some existing models for repairing missing values.

$$SSE = \frac{1}{n}\sum_{i=0}^{n-1}(\hat{z_i} - z_i)^2 \,; z_0, ..., z_{n-1} \in SMV \tag{5}$$

*Accuracy Discussion.* As shown in Figure 6, we compare ISTM with three traditional MLR models: Static Space-Based, Static Time-Based and Static Space-Time-Based, in terms of accuracy. We do not consider the Dynamic Space-Time-Based Model because it has the same accuracy as our model.

There exist different methods for data replacement:

- *Mean of All Historical(MAH).* Means of all Historical data can directly replace missing values.
- *Previous Data (PD).* If we assume that the values does not change drastically over a certain period of time, the value of a previous timestamp can replace the missing value (of current timestamp).
- *Static Space-Based Model(SSBM).* A subset of values of the surrounding sensors are used to train the model(MLR), but when new data arrive, it won't re-train/update the model.
- *Static Time-Based Model(STBM).* A part of historical data of one sensor is used to train the model(MLR, non incremental), but when new data arrive, it won't re-train/update the model.
- *Static Space-Time-Based Model(SSTBM).* A part of historical data of one sensor and values of the surrounding sensors are used to train the model(MLR, non incremental), but when new data arrive, it won't re-train/update the model.
- *Dynamic Space-Time-Based Model(DSTBM).* The model (MLR, non incremental) is updated by means of new data (considered as dependent variable), a subset of historical data and the data of surrounding sensors (considered as as independent variables).

MAH and PD are two simple approaches but widely used, in some case they can get an acceptable results [5, 11, 21].

For each sensor, the data of first 174 time points are used to initialize the models(if needed). Respectively 5%, 10%, 15%, 20%, 25% of the remaining data will be selected randomly and be marked as a Simulated Missing Value(SMV).

As the proportion of missing values increases, the SSE value of all algorithms increases. This is because all of the above methods rely on historical data, and 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 6 shows that, when the ratio of SMV is equals to 5%, the SSE of SSBM is much higher than the other methods. Although its SSE is improved at its subsequent point, but it is still higher than other methods.

When ratio of SMV is equal to 5% and 10%, the performance of STBM is close to PD and ISTM (which are the two best methods according to our findings). But, when the proportion of missing values increases, the growth rate of SSE of STBM is significantly higher than MAH, PD, ISTM. That is to say, STBM is more suitable for situations where the ratio of missing values is low than high.

SSE of MAH relatively gently grows with the ratio of SMV, is better (lower) than STBM in the ratio of SMV 15%, 20% and 25%, and is always not as good as PD or ISTM.

As 2nd best method, compared to others, PD's performance is close to ISTM. But there are still clear differences between PD and ISTM as shown in Figure 6.B: when the ratio of SMV is equal to 5% ,10% ,15% ,20%, PD's curve is relatively flat; in the case where ratio of SMV grows 5 times, SSE only increases by 50%. And when SMV ratio is equal to 25%, the accuracy of PD is better than all methods involved in comparison.

In general, our new method ISTM achieves the best performance. ISTM wins in terms of accuracy when ratio of SMV is equal to 5% ,10% ,15% ,20%. Until a ratio of SMV equal to 25%, SSE of PD (99) performs slightly better than ISTM (100). The speed limit of the Dutch highway is 120, we can think that these two methods are very close at the point where ratio of SMV is equal to 25%. ISTM and PD are two robust methods, their performance is stable with the different ratio of SMV.

In summary, when the ratio of SMV is relatively low (5-20%), ISTM gains overall advantage. When the ratio of SMV reaches 25%, PD and ISTM perform similarly.

*Comparison of Efficiency.* We have shown that ISTM has better accuracy than MAH, PD, SSBM, STBM, SSTNM. According to [24], we know that DSTBM has the same accuracy performance as ISTM.

In order to compare efficiencies of DBSTM and ISTM, we ran a set of experiments on a computer (Intel i5, 2.60 GHz, 8GB RAM, macOS, Python), with five different data samples sizes: 10%, 20%, 30%, 40%, 50% of all data (the simulated missing value ratio is 10%, which is close to 9% mentioned in section 3.2).

Figure 7 illustrates the speed of DSTBM and ISTM. Obviously, as the number of samples increases, the running time of ISTM increases: with 10% of all data, it takes 497 seconds; with 50%, it takes 2587 seconds; 5 times the amount of data leads to about 5 times running time. In comparison, DSTBM running time increases much faster than ISTM: with 10% of all data, it takes 3994 seconds; with 50%, it takes 77241 seconds; hence, 5 times the amount of data brings about 19 times the running time.

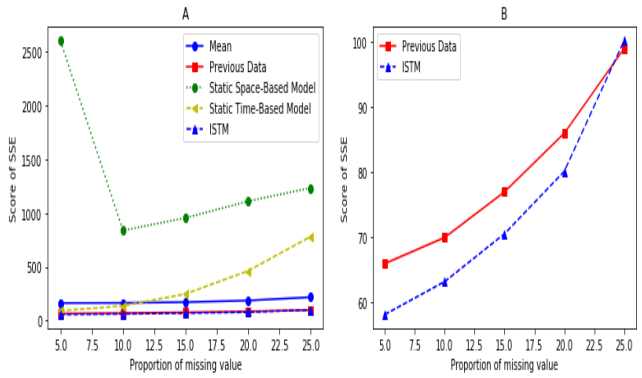


**Figure 6: Comparison of SSE on ISTM and other methods**

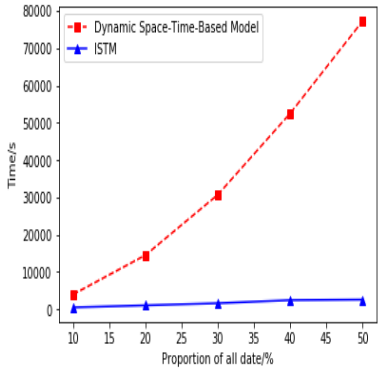


**Figure 7: Comparison of speed between ISTM and DSTBM**

## 4 RELATED WORK

According to [23], "*real-time analytics of massive IoT data is in its infancy*". The paper also describes a set of use cases of real-time analytics together with their network requirements.

To mitigate data quality issues of IoT in real time, some approaches try to build a more reliable network that has better fault tolerance, including missing values. For instance, in [9] the tree topology is considered to have good fault-prone because of the single point of failure. Standby hardware, which replaces the failed physical network components in real-time [23], is another way of reducing missing values. The hardware-based fault tolerance is trustworthy but expensive, so software-based network fault tolerance approaches needs to be carefully investigated [23].

In [10], some approaches based on principal component analysis (PCA) are employed for imputing missing values, in order to get the real-time crash likelihood prediction. In order to improve the accuracy, [27] propose a Singular Value Decomposition algorithm which can be updated with the current reconstructed frame adaptively.

In [5, 26] a typology of IoT missing values is proposed: (1) missing completely at random (MCAR), (2) missing at random (MAR), (3) not missing at random (NMAR). According to different types of missing values, corresponding methods should be used instead of using a generic method. Among the models proposed are: MCL, a model based on context and linear mean; MBS, a model based on binary search; and MGI, a Gaussian mixture model (MGI).

In [3], authors propose an imputation technique based on spatio-temporal and association rule mining (STARM). Firstly STARM uses space-time data to determine the Pearson coefficient between two sensors, then the missing value is replaced by a weighted average of values for sensors whose Pearson exceeds a threshold. A new moving-neighborhood interpolation algorithm based on Delaunay triangulation technique has been proposed in order to find out the neighbor's sensors set having a strong spatial correlation.

Nearest Neighbor (NN) imputation is proposed in [12] which uses k-d tree and spatial and temporal data to determine which nodes are spatially and temporally correlated with each other. According to authors, NN is suitable for use in resource-constrained WSNs.

In [4], authors propose TKCM (Top-k Case Matching) to impute missing values in streams of time series data. For the imputation, TKCM exploits both a set of reference time series (for each time series) and a similar historical situations in the reference time series.

## 5 CONCLUSION AND FUTURE WORK

In this article, we described a method for repairing missing values in an IoT context, using incremental MLR (IMLR), depending on the spatial and temporal features. By analyzing time and space complexity of IMLR, we have showed that IMLR is suitable for our application scenario. Our method, called ISTM relies on a Space-Time model adapted to incremental MLR. Our method has been tested on repairing traffic data drawn from CityPulse; moreover experimental results show that ISTM outperforms some traditional methods in terms of accuracy and efficiency.

After surveying some related work, to the best of our knowledge, there is no work on repairing missing value of IoT data stream with incremental Multiple Linear Regression, i.e.; a method that (on-line) updates dynamically the model.

Future directions of this work are based on the following observations: MLR is a convenient linear modeling tool, which is easy to adapt for incremental without loss of accuracy; Non-linear models, such as Ridge Regression, SVM may be more powerful [14] than linear models. Using these incremental tools in order to detect/repair missing values in the IoT world deserves further investigation.

## REFERENCES


[1] Muhammad Intizar Ali, Feng Gao, and Alessandra Mileo. 2015. CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets. In *In proceedings of ISWC 2015 - 14th International Semantic Web Conference*. W3C, Bethlehem, PA, USA, 374–389.

[2] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. 2012. Semantics for the Internet of Things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)* 8, 1 (2012), 1–21.

[3] Li Bin, Lin Yaping, Zhou Siwang, Luo Qing, and Yin Bo. 2012. An interpolation algorithm based on sliding neighborhood in wireless sensor networks. *Journal of Computer Research and Development* 49, 6 (2012), 1196–1203.

[4] Kevin Wellenzohn Michael H Böhlen, Anton Dignös Johann Gamper, and Hannes Mitterer. 2017. Continuous Imputation of Missing Values in Streams of Pattern-Determining Time Series. (2017).

[5] Moniek CM de Goeij, Merel van Diepen, Kitty J Jager, Giovanni Tripepi, Carmine Zoccali, and Friedo W Dekker. 2013. Multiple imputation: dealing with missing data. *Nephrology Dialysis Transplantation* 28, 10 (2013), 2415–2420.

[6] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. 2006. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology* 59, 10 (2006), 1087–1091.

[7] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.

[8] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.

[9] Christoforos Kachris, Konstantinos Kanonakis, and Ioannis Tomkos. 2013. Optical interconnection networks in data centers: recent trends and future challenges. *IEEE Communications Magazine* 51, 9 (2013), 39–45.

[10] Jintao Ke, Shuaichao Zhang, Hai Yang, and Xiqun Chen. 2018. PCA-Based Missing Information Imputation for Real-Time Crash Likelihood Prediction Under Imbalanced Data. *arXiv preprint arXiv:1802.03699* (2018).

[11] Mihail Halatchev Le Gruenwald. 2005. Estimating missing values in related sensor data streams. In *COMAD*.

[12] YuanYuan Li and Lynne E Parker. 2014. Nearest neighbor imputation using spatial–temporal correlations in wireless sensor networks. *Information Fusion* 15 (2014), 64–79.

[13] W Min, L Wynter Transportation Research Part C Emerging, and 2011. 2011. Real-time road traffic prediction with spatio-temporal correlations. *Elsevier* 19, 4 (Aug. 2011), 606–616.

[14] Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara, and Shin Ishii. 2003. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19, 16 (2003), 2088–2096.

[15] LUO Jizhou PAN Liqiang, LI Jianzhong et al. 2010. A Multiple-Regression-Model-Based Missing Values Imputation Algorithm in Wireless Sensor Network. *Journal of Computer Research and Development* 33, 1 (2010), 1–11.

[16] Dan Puiu, Payam Barnaghi, Ralf Tönjes, Daniel Kümper, Muhammad Intizar Ali, Alessandra Mileo, Josiane Xavier Parreira, Marten Fischer, Sefki Kolozali, Nazli Farajidavar, et al. 2016. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access* 4 (2016), 1086–1108.

[17] Ma Q, Gu Y, Li FF, and Yu G. 2016. Order-Sensitive multi-source sensory missing value imputation technology. *Ruan Jian Xue Bao/Journal of Software* 27, 9 (2016), 2332âĹŠ2347.

[18] Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk, and Peter Solenberger. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey methodology* 27, 1 (2001), 85–96.

[19] RS Somasundaram and R Nedunchezhian. 2011. Evaluation of three simple imputation methods for enhancing preprocessing of data with missing values. *International Journal of Computer Applications, Vol21* 21, 10 (2011).

[20] Antti Sorjamaa, Amaury Lendasse, et al. 2010. Fast missing value imputation using ensemble of SOMs. (2010).

[21] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* 17, 6 (2001), 520–525.

[22] Joost R van Ginkel, L Andries van der Ark, et al. 2005. SPSS syntax for missing value imputation in test and questionnaire data. *Applied Psychological Measurement* 29, 2 (2005), 152–153.

[23] Shikhar Verma, Yuichi Kawamoto, Zubair Md Fadlullah, Hiroki Nishiyama, and Nei Kato. 2017. A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1457–1477.

[24] Huang Lele Wang Huiwen, Wei Yuan. 2014. Incremental algorithm of multiple linear regression model. *Journal of Beijing University of Aeronautics and Astronsutics* 40, 11 (2014), 1487–1491.

[25] Jie Xu, Chen Xu, Bin Zou, Yuan Yan Tang, Jiangtao Peng, and Xinge You. 2018. New Incremental Learning Algorithm With Support Vector Machines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018).

[26] Xiaobo Yan, Weiqing Xiong, Liang Hu, Feng Wang, and Kuo Zhao. 2015. Missing value imputation based on Gaussian mixture model for the internet of things. *Mathematical Problems in Engineering* 2015 (2015).

[27] Liang Zhao and Fang Zheng. 2017. Missing Data Reconstruction Using Adaptively Updated Dictionary in Wireless Sensor Networks. In *Proceedings of the 2017 The 7th International Conference on Computer Engineering and Networks. 22-23 July, 2017 Shanghai, China (CENet2017) Online at href=" https://pos. sissa. it/cgi-bin/reader/conf. cgi? confid= 299"> https://pos. sissa. it/cgi-bin/reader/conf. cgi? confid= 299, id. 40.*