Missing Data Estimation in Mobile Sensing Environments

Yuchao Zhou¹, Suparna De¹, Member, IEEE, Wei Wang², Ruili Wang³ and Klaus Moessner¹, Senior Member, IEEE
¹Institute for Communication Systems, University of Surrey, UK GU2 7XH
²Department of Computer Science and Software Engineering, Xi’an Jiaotong Liverpool University, China
³Institute of Natural and Mathematical Sciences, Massey University, Auckland, New Zealand

Corresponding author: Yuchao Zhou (e-mail: yz0003@surrey.ac.uk).

This paper describes work undertaken in the context of the TagItSmart! project (www.tagitsmart.eu). TagItSmart! is a collaborative project supported by the European Horizon 2020 programme, contract number: 688061. The first author also acknowledges funding support from the collaborative European Union and Ministry of Internal Affairs and Communication (MIC), Japan, Research and Innovation action, iKaaS under EU Grant number 643262.

ABSTRACT Mobile sensing techniques have been increasingly deployed in many Internet of Things based applications because of their cost efficiency, wide coverage and flexibility. However, these techniques are unreliable in many situations due to noise of different kinds, loss of communication, or insufficient energy. As such, datasets created from mobile sensing scenarios are likely to contain large amount of missing data, which makes further data analysis difficult, inaccurate, or even impossible. We find that the existing estimation models and techniques developed for static sensing do not work well in the mobile sensing scenarios. To address the problem, we propose a spatio-temporal method, which is specifically designed for answering queries in such applications. Experiments on a real-world, incomplete mobile sensing dataset show that the proposed method outperforms the state-of-the-art noticeably in terms of estimation errors. More importantly, the proposed model is tolerant to datasets with extremely high missing data rates. Training with the proposed model is also efficient, which makes it suitable for deployment on computationally constrained devices and platforms that need to process massive amounts of data in real time.

INDEX TERMS Missing Sensor Data, Data Estimation, Mobile Sensing, Support Vector Regression, Spatio-Temporal Model

I. INTRODUCTION

Sensors play important roles in developing various kinds of smart applications of different scales, e.g., smart home, smart building, and smart city. The rationale is that with the sensing data collected from different sources, we could better monitor and understand the physical world objects and their surroundings, and take pre-emptive actions to prevent undesirable events from happening in many circumstances. With intelligent analytics of the data, we could discover useful knowledge and insights, which allows us to develop truly smart applications [1-3]. In recent years, mobile sensing techniques have been increasingly employed due to cost efficiency, large coverage and flexibility, e.g., attaching different types of sensors to mobile objects such as human bodies, taxis, or public buses [4, 5]. However, mobile sensing might not function as expected in the presence of many unpredictable factors such as noise of different kinds, communication fault, or insufficient energy, which result in incomplete data. The incomplete values have to be processed and approximated in order for further data processing to be more faithful and reliable.

Given a dataset of large size with few missing data values, one could simply remove the whole set of records that include missing values [6]. However, if the number of missing values is large, doing so causes a significant loss of information. Some simple methods replace the missing values with “0”s or the mean of other values; however, this may introduce undesirable bias to the entire dataset [7] or even change the pattern of data, especially with high variance [8]. Therefore, more sophisticated estimation techniques, e.g., regression, are needed to tackle the problem.

In static sensing scenarios, estimation methods are designed based on the assumption that different sensors are installed at the same or nearby locations and are potentially
correlated to each other. The standard model in the existing works creates datasets by collecting all sensor observations and storing them in a matrix form. Each row consists of observations from all sensors collected at the same time or period and same or nearby locations. This model is referred to as the Observation Model (O model). The missing values can be estimated using techniques like matrix completion [9-11], which has been very successful in applications such as collaborative filtering for user rating prediction [12] and intelligent transportation systems for traffic speed estimation [13]. These methods do not require any training and can deal with large matrices of very high missing rates (70-80%). Performance of these methods is usually sensitive to the choice of rank; and the problem can be alleviated by using nuclear norm minimisation [14] or variants of Bayesian Principal Component Analysis [12]. Nevertheless, they may not be suitable for mobile sensing data as rows and columns of the matrix have to represent fixed objects (e.g., users and products in collaborative filtering, or road segments and time in traffic speed estimation). In mobile sensing applications where mobile sensors report observations at continuously changing locations and time, it is not easy to build such matrices. Where a matrix is built by fixing either the time or location, estimation performance remains unknown. Missing values can also be estimated using supervised learning methods, e.g., regression [15]. This approach requires enough training data, i.e., the number of missing values in a dataset should be sufficiently low. Furthermore, the assumption that two sensors physically close to each other are correlated seems simplistic and questionable.

In mobile sensing scenarios, it is usually hard to directly build a usable dataset for estimation because of the much larger number of missing data points compared to static sensing, as evidenced in the real-world mobile sensing dataset used in our experiments. As such, in many situations, standard regression models cannot be created based on the collected observation data. It should also be noted that this problem is different from time series analysis and forecasting, in which a model can be built to predict future values based on previously observed values, usually obtained at successive, equally-spaced time points.

In mobile sensing, observation data is normally attached with spatial (e.g., latitude, longitude and altitude) and temporal (e.g., day, hour, and minute) metadata. Intuitively, within a constrained situation, that is, a specific temporal range (e.g., 6 hours) and a specific spatial range (e.g., area within distance of 2 KMs from the current location), observation data is correlated to metadata. For instance, outdoor temperature values fluctuate as time elapses, and air pressure varies with different altitude values. Two observation values would be similar to each other if their temporal and spatial metadata were similar. This observation leads to the design of a new model for missing data estimation, referred to as the Spatio-Temporal Model (ST Model) in this paper. In the ST model, each observation data point is represented in terms of temporal and spatial features. We perform inference on the original dataset and only select observation data that falls under a constrained temporal and spatial range to build the training dataset. We then apply regression techniques to explore the unobvious correlations among the observation data and the metadata.

This work is the first attempt for missing data estimation in large-scale mobile sensing scenarios, which are becoming more and more prevalent with increasing smart city deployments [4]. The proposed model is reliable even with high missing rates and efficient enough to be executed on constrained devices. The contributions of the paper include: 1) design and implementation of the ST model and Support Vector Regression for missing sensor data estimation. The method aims to solve the problem of existing methods and to perform reliable estimation from incomplete mobile sensing datasets with extremely high missing rates. It allows performing spatial and temporal inference to approximate the regression input for the missing values; and 2) extensive experiments and evaluations based on a real-world dataset, and comparison to the state-of-the-art methods in terms of standard evaluation metrics, i.e., Root Mean Squared Error, Mean Absolute Percentage Error, and execution time. It shows that the proposed method outperforms the existing methods noticeably. Furthermore, it is tolerant to datasets with high missing rates and suitable for deployment on computationally constrained devices.

The rest of the paper is organised as follows. Section 2 provides a review of the related work and techniques for missing value estimation. The details of the proposed model and regression technique are described in Section 3. Section 4 presents the experiments performed on a dataset collected from a real-world smart city application, with different missing rates, from low, moderate to extremely high. Section 5 reports the evaluation results based on the standard metrics by comparing to existing methods and discusses the applicability of the proposed method. Section 6 concludes the paper and outlines the future research directions.

II. RELATED WORK

The O Model is the standard representation model used in many existing methods for missing value estimation. The fundamental assumption of this model is that all sensors are located in the same place or in close vicinity, and their observation data is measured at the same time or time period. The observation values can be aligned in a matrix-like form as shown in Table I, where each row contains a sensorID and the values observed by a number of co-located sensors at the same time. The cells marked with “N/A” represent the missing values and the ones in the last column marked with a “?” mark represent the values to be estimated or queries. As an example, when using techniques such as regression, missing values have to be pre-processed to make the training set complete. In Table I, the query (at Sensor 3033) asks for a value of Humidity (i.e., the dependent variable) based on
values of CO, Ozone+NO2, Particles, and Temperature (i.e., the explanatory or independent variables). To train a regression model, records of Sensor 3037 and 3080 have to be removed from the training set as the values for CO and Particles are missing. The record of Sensor 3066 also needs to be removed because the dependent variable, Humidity is unknown. In another query at Sensor 3000, a missing data for Particles is present; therefore, the whole column of Particles has to be removed. Furthermore, records of Sensor 3037 and 3066 also need to be removed due to missing values.

Our study shows that there are three main categories of methods for missing data estimation in literature, namely, interpolation/extrapolation-based, matrix completion-based, and regression-based methods.

**Interpolation/extrapolation-based** - Instead of removing rows and columns containing missing values, some methods in this category keep all the information in the original dataset, and replace missing values with “0”s or the mean of other values. Troyanskaya et al. [9] and Dixon et al. [16] propose to substitute missing values by the mean values of the k nearest neighbour records. One notable advantage is that no information is removed and therefore, there is enough training data to be used for estimation algorithms. However, different k values may lead to different results and varying performances during the estimations. Moreover, this may create a bias to the original dataset [7] or change the pattern of data with high variance [8]. Considering the uncertainty of missing values, multiple-step imputation based methods [17-19], first impute missing values multiple times (m times) to generate m datasets, and then average the imputed values at the same data point of all generated datasets to get a final integrated value. The generation of the multiple datasets can be done by randomly drawing m times missing values from the joint distributions of the variables containing the missing values and other variables in the whole dataset. These methods focus on generating unbiased datasets and provide confidence coefficient values to show the unreliability of imputed values. In multiple imputation, larger number of m usually leads to less variability produced [20], and how to select the parameter m is discussed in [21, 22]. However, the major limitations of multiple imputation are its complexity and production of non-deterministic results. In addition, to be more effective, multiple imputation models need to be congenial with the analysis model [20]. Kim et al. [23] propose a local least square estimation method based on similar genes selected using the k nearest neighbour method. In application in which data contains both spatial and temporal attributes, linear interpolation [24], kriging algorithm [25-27], and Inverse Distance Weighting (IDW) method [28] are the most popular techniques. Interpolation algorithms try to find a smooth way to fill missing data points between values. Nevertheless, they have limitations: for example, linear interpolation depends on the assumption that data follows a linear distribution, which is not true for most data; Kriging requires a large computation cost; and the IDW approach forces all the interpolated values to be between the maximum and minimum of the observed data.

**Matrix Completion-based** – Matrix completion-based methods attempt to derive a low-PCA matrix from a small number of samples [9-11]. Troyanskaya et al. [9] utilise the Singular Value Decomposition (SVD) to extract significant eigenvalues and the corresponding eigenvectors. These vectors are then used with linear regression to estimate the missing values. Gao et al. [29] make use of the channel correlations of data from phasor measurement unit to separate the data into blocks with low rank. For each block, the authors apply SVD to recover the missing data. To save computational cost and reduce storage requirement, Cai et al. [30] propose a Singular Value Thresholding (SVT) method to complete a low-rank matrix with minimum nuclear norm (e.g., sum of the singular values of a matrix). Genes et al. [31] exploit the Gaussian distribution of data and use SVT to recover missing data in electricity distribution systems. One common limitation of these methods is that they require the matrix derived from the incomplete dataset to be of low-rank, which is not necessarily true in all practical scenarios. With the increment of the matrix rank, the method becomes more and more computationally expensive. Matrix completion-based methods are also closely related to the concept of compressive sensing [32]. The original idea of compressive sensing is to under-sample high-dimensional signals and to accurately reconstruct them by exploiting hidden structures in the underlying data. It has been used in many applications for missing value estimation, such as urban traffic [33] and Internet traffic [34]. The main technique for compressive sensing is matrix completion with norm minimisation (e.g., nuclear norm or spectral norm). It has been reported that compressive sensing-based methods outperform conventional imputation-based ones when missing rates are extremely high, e.g., 80%. However, it also inherits the limitations for matrix completion based methods. Furthermore, in mobile sensing applications, mobile sensors report observation data at constantly changing locations and time. A matrix only can be built by fixing either the time or location, estimation performance remains unknown.

**Regression-based** - Liu et al. [35] propose a method to estimate missing values for hierarchical time series data. The method utilises the hierarchical relationship between time series data and applies a Locally Weighted Regression (LWR) technique to fit the data. However, hierarchical time series data is very different from mobile sensing data. Kurasawa et al. [15] consider missing sensor value estimation in a participatory sensing environment. Assuming that not all the sensor observations are correlated to each other, the authors apply LWR and only train the model with locally correlated sets of sensor records and ignore sensor observations that are not highly related. The selection of sensor observations is done by the sparse feature of coefficients trained by Least Absolute Shrinkage and Selection Operator (LASSO). The sensor observations whose
coefficients are close to zero are removed. For each missing value, the selection of locally correlated sensor records requires multiple steps for selection and validation, thus the process takes a long time to complete. Another limitation is that it is hard to choose the optimal number of nearest neighbours, \( k \). In their experiments, each dataset has one optimal \( k \) value varying from 10 to 50, which is likely to be dependent on each dataset and needs to be chosen from experience. As such, it might not be applicable in a mobile sensing environment as the situation keeps changing all the time. Zhu et al. [36] build a linear mixture kernel function based on the polynomial and radial basis functions to estimate both discrete and continuous missing values. The method considers the impact of each point in the dataset according to the mixture kernel model and estimates missing values one by one based on the previous estimated values. It iteratively refines the estimated values and the process stops when a convergence constraint is satisfied. The results show that mixture kernel function model outperforms other single function models. However, the method needs to consider the entire dataset and estimates a missing value based on all previously estimated ones, hence it is applicable in datasets with high missing rates. Therefore, the accuracy of the estimation cannot be guaranteed. Another relevant concept considered in this study is Random Sample Consensus (RANSAC) [37]. Though it should not be directly used for estimation of missing data, it is a widely used model fitting method. It works by iteratively selecting the inliers in the dataset and has been widely used for outlier/ anomaly detection [38, 39]. By fitting a model based on only the inliers, it is anticipated that better performance can be obtained compared to models learned from the entire dataset.

Investigation of the existing work shows that the missing data estimation problem in mobile sensing has not been sufficiently studied. Among the reviewed work, Kurasawa et al.’s work [15] based on regression analysis is similar to our method and is used for benchmarking in the experiments and evaluation. Additionally, we compare the proposed methods with the RANSAC, compressive sensing and K-nearest Neighbour (kNN) based methods.

### III. MODEL DESIGN

In this section, we first introduce some of the important terms used throughout the paper. Then we present the design of the ST Model and the algorithm to perform the regression analysis.

#### A. DEFINITIONS OF RELATED TERMS

We define some of the related terms and clarify their specific meanings in the context of this work.

**Definition 1 (Observation)** An Observation is a situation in which a sensing method has been used to estimate or calculate a value of a Property of a feature of interest [40].

Example: A PM10 particle sensor is installed on the top of a bus to measure the PM10 level at the centre of the city of Santander.

**Definition 2 (Observation Value)** An observation value is the value of the result of an Observation [40].

Example: At 01:53:00, on 12th, March, 2016, at the city centre (Latitude/Longitude/Altitude: 43.4519/-3.8322/7, the PM10 observation value is 0.68 mg/m\(^3\).

**Definition 3 (Query)** A query is an inquiry for estimation of a missing observation value of a sensor.

Example: Based on the observation values of co-located sensors, a missing observation value of the sensor of interest is estimated. In Table I, the query on Humidity can be answered based on the observation values from CO, Ozone+NO2, Particles and Temperature.

**Definition 4 (Record)** A record consists of an observation value and its corresponding features.

Example: A row in Table I and Table II is considered as a record. In Table I, the training set, which consists of a number of records, is used to build a regression model for missing data estimation. In Table I, the observation values of co-located sensors are used as features, while in Table II, the temporal and spatial metadata is used as features.

**Definition 5 (Missing Rate)** Missing rate is the ratio of the number of missing data points to the number of all the data points.

#### B. SPATIO-TEMPORAL MODEL

It is clear that by using the O model, the pre-processing steps are likely to reduce the size of the training data significantly and result in insufficient training data. In the worst case, the training dataset may be empty. Although the techniques such as LASSO can be used to alleviate the problem to a certain extent, as proposed in [15], they still have difficulties for datasets with very high missing rates. Statistics on the SmartSantander dataset (detail can be found in Section IV.A) used in our experiments show that the missing rate could reach 62\% (see Table III). In such cases, the O model tends to be not useful at all as can be seen from the experimental results.

Another notable characteristic of mobile sensing is that the data from mobile sensors is usually reported at different times and different locations. In most of the situations, one
might not be able to identify any co-located sensors for a particular sensor at a specific time or time period. This makes O model’s assumption that observation values from sensors can be collected at the same time and at the same location invalid.

We propose the Spatio-Temporal Model (ST Model) to solve these problems. The ST model exploits the spatial and temporal metadata associated with the observation values and helps design efficient methods for accurate estimation in datasets with high missing rates. The underlying assumption in this model is that two observation values tend to be similar to each other if they are measured and reported at the same/similar time or time period, and in the same location or vicinity. Observation values are correlated with spatial and temporal metadata, e.g., outdoor temperature values fluctuate as time elapses, and air pressure varies with different altitude values. Nevertheless, the correlation and the impact that different metadata has on the observation is extremely difficult to measure and quantify. Moreover, the probability distribution of the missing data is completely unknown; therefore, it is not possible to build a formal mathematical model for the estimation task. We resort to regression techniques to solve the problem: within a constrained situation, e.g., a specific temporal range and a specific spatial range, the correlations among the different metadata or features and the sensor observation can be captured by the regression model.

This leads to a new data representation, where each record consists of spatial (e.g., latitude, longitude and altitude) and temporal features (e.g., day, hour, and minute), as well as the observation value. Each record in Table I can be transformed into multiple records, which are stored in different datasets depending on their observation types. Each dataset contains only one type of observation and each record contains the observation value associated with spatial and temporal features. As an example, observation values of humidity from Table I are transformed to one dataset with spatial and temporal information, as illustrated in Table II.

Decoupling of the mobile sensor types and the new representation bring at least two advantages: 1) sufficient data for regression training can be obtained; and 2) training datasets can be built more efficiently without the need to search for co-located sensors. It should be noted that in mobile sensing, if an observation value is missing, then its associated spatial data is also missing, such as the record of Sensor 3066 in Table II. A query in this model is to ask an observation value of a sensor at a given time. Before a query can be answered, two steps need to be performed: location inference (Algorithm 1) and dataset preparation (Algorithm 2). This model can also be used to answer queries asking for an observation value at a given location and time, in which case the location inference is omitted.

In location inference, the spatial data of the query must be inferred first based on the available information. Algorithm 1 shows the algorithm LocationInference. As the exact location cannot be recovered, the algorithm searches the dataset (d) and tries to find the approximate location based on the SensorID (SID) that has the missing value and the time (t) at which the observation happened. The function getLocation(SID, t-1, d) searches the dataset and returns the latest location of the sensor before time t and stores it in location_BeforeT. The function getLocation(SID, t+1, d) returns the earliest location after the time t and stores it in location_AfterT. The function getLocation() returns the mean of the location_BeforeT and location_AfterT, in terms of longitude, latitude and altitude values.

In dataset preparation, the objective is to select appropriate data for training. The values for the two parameters, time offset At, and radius r, can be defined to specify the scope of the search. The offset At specifies the length of the time window before and after time t. The radius r specifies the

---

Algorithm 1. LocationInference

1. INPUT: Sensor ID (SID), time point t, and dataset d
2. OUTPUT: inferred location, inferred_Location
3. 4. location_BeforeT = getLocation(SID, t-1, d)
5. location_AfterT = getLocation(SID, t+1, d)
6. IF (location_BeforeT = null) AND (location_AfterT = null)
7. inferred_Location = getMean(location_BeforeT, location_AfterT)
8. END IF
9. 10. RETURN inferred_Location

Algorithm 2. DatasetPreparation

1. INPUT: query q, dataset d, time offset At, and radius r
2. OUTPUT: trainingSet, dataset for training regression models
3. 4. trainingSet = []
5. FOR each record, row, in the original dataset
6. IF row has no missing fields
7. IF [q-time] < At and distance(locationq, locationrow) < r
8. trainingSet.add(row)
9. END IF
10. END FOR
11. RETURN trainingSet

---

Table II: Example of Incomplete Dataset of ST Model

<table>
<thead>
<tr>
<th>SensorID</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Altitude</th>
<th>Day</th>
<th>Hour</th>
<th>Minute</th>
<th>Humidity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3033</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>12</td>
<td>7</td>
<td>56</td>
<td>?</td>
</tr>
<tr>
<td>3000</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>12</td>
<td>9</td>
<td>56</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>12</td>
<td>9</td>
<td>56</td>
<td>...</td>
</tr>
<tr>
<td>3103</td>
<td>43.4748</td>
<td>-3.79949</td>
<td>8</td>
<td>12</td>
<td>9</td>
<td>54</td>
<td>0.77</td>
</tr>
<tr>
<td>3037</td>
<td>43.4551</td>
<td>-3.8338</td>
<td>7</td>
<td>12</td>
<td>9</td>
<td>53</td>
<td>0.64</td>
</tr>
<tr>
<td>3080</td>
<td>43.4556</td>
<td>-3.8379</td>
<td>11</td>
<td>12</td>
<td>9</td>
<td>53</td>
<td>0.62</td>
</tr>
<tr>
<td>3066</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>12</td>
<td>9</td>
<td>56</td>
<td>N/A</td>
</tr>
<tr>
<td>3048</td>
<td>43.4792</td>
<td>-3.79249</td>
<td>2</td>
<td>12</td>
<td>9</td>
<td>54</td>
<td>0.59</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>12</td>
<td>9</td>
<td>54</td>
<td>...</td>
</tr>
</tbody>
</table>

Data is extracted from the SmartSantander dataset used in the experiments.
and utilises L2-norm, and
is a constant controlling the trade-off between
i; linear SVR tries to fit the set of data with a
+,

Analogous to the “soft margin” principle, some errors are
allowed and the above equation becomes

minimise \( \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i^- + \xi_i^+ ) \)

subject to

\[
\begin{align*}
    & y_i - f(\mathbf{x}_i) \leq \varepsilon + \xi_i^- , \\
    & f(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i^+ \\
    & \xi_i^- , \xi_i^+ \geq 0
\end{align*}
\]

where \( C > 0 \) is a constant controlling the trade-off between
flatness of the linear function (2) and the tolerance of the
number of points whose deviation is larger than \( \varepsilon \). \( \xi_i^- \), \( \xi_i^+ \) are
slack variables for each pair \((\mathbf{x}_i, y_i)\) \( \in \mathcal{D} \) to tolerant errors
up to these slack variables.

The linear \( \varepsilon \)-insensitive loss function is

\[
\text{Loss}_\varepsilon = \begin{cases} 
0 & \text{if } |y_i - f(\mathbf{x}_i)| \leq \varepsilon \\
|y_i - f(\mathbf{x}_i)| - \varepsilon & \text{otherwise} 
\end{cases}
\]

This optimisation problem could be solved by the
Lagrange dual formulation.

minimise \( \frac{1}{2} \sum_{i=1} l (a_i - a_i^*) (a_i - a_i^*) \mathbf{x}_i \mathbf{x}_i \)

\[- \sum_{i=1} a_i (y_i + \varepsilon \sum_{i=1} a_i + a_i^* ) \]

subject to

\[
\begin{align*}
    & \sum_{i=1} (a_i - a_i^*) \mathbf{x}_i \mathbf{x}_i = 0 \\
    & 0 \leq a_i, a_i^* \leq C
\end{align*}
\]

where \( a_i, a_i^* \) are Lagrange multipliers. The parameter vector
\( w \) can be described by

\[
w = \sum_{i=1} (a_i - a_i^*) \mathbf{x}_i
\]

To obtain optimal solutions, the Karush-Kuhn-Tucker
(KKT) conditions should be satisfied. There are efficient
techniques for solving the linear SVR problem such as
Sequential Minimal Optimisation (SMO) [47]. The support
vector algorithm only depends on dot products between data
items so it is convenient to use kernel functions. The SVR
optimisation problem can be formulated as:

\[
f(\mathbf{x}) = \sum_{i=1} (a_i - a_i^*) k(\mathbf{x}_i, \mathbf{x}) + b.
\]
With respect to nonlinear situations, kernels can be applied; and in our experiments, the Gaussian kernel \( (9) \) and polynomial kernel \( (10) \) were used for comparison.

\[
k(x_i, x_j) = \exp(-\|x_i - x_j\|^2)
\]  
(9)

\[
k(x_i, x_j) = (1 + \langle x_i', x_j \rangle)^p, \ p \in \{2, 3, \cdots\}
\]  
(10)

In our method, the \( w \) vector consists of weights for both temporal and spatial features,

\[
w = \{w_1, \cdots, w_m; w_{1s}, \cdots w_{ms}\}
\]  
(11)

where \( w_i \) represents the weights for temporal features \( (m \) is the dimension of the temporal features), and \( w_s \) for weights of spatial features \( (n \) is the dimension of spatial features).

**IV. EXPERIMENTS**

In order to assess the effectiveness of the ST model, we conducted extensive experiments based on a real-world, mobile sensing dataset collected from the SmartSantander\(^1\) smart city testbed, using the O model and ST model, respectively. We re-implemented the locally weighted regression proposed in [15] and other baseline techniques (e.g., kNN, compressive sensing and RANSAC) for comparison and evaluation. We also tested the performance of different methods under various missing rates, from low, moderate to extremely high.

**A. SMARTSANTANDER TESTBED**

The SmartSantander project provides a city-scale testbed for experimental research in smart city applications. It offers various types of sensing data from both fixed and mobile sensors that support environmental monitoring of the Santander city, Spain. Our experiments only made use of the data collected from mobile sensing devices installed on public buses. The data was extracted from JSON files from the SmartSantander Map during two weeks from 12th March to 25th March 2016. It consists of five different types of observations, i.e., CO, Humidity, Ozone+NO2, Particles (PM10), and Temperature. Each observation was associated with SensorID, spatial and temporal information (i.e., where and when the observation was reported). Missing rates of the data in mobile sensing applications can be substantially high. In the collected SmartSantander dataset, the missing rate of Ozone+NO2 data is around 62%.

The dataset was cleaned by removing records that have missing locations and observation values. It also contains records with unusually large absolute values, which were also removed by setting thresholds for different sensor types. After cleaning, a dataset with 5,481 records was created. The statistics of the collected data is summarised in Table III, which include minimum, maximum, mean, standard deviation, original missing rate and unit of measurement.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>STD</th>
<th>Missing Rate</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>0.00</td>
<td>0.56</td>
<td>3.45</td>
<td>17%</td>
<td>mg/m(^3)</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>0.17</td>
<td>0.89</td>
<td>0.59</td>
<td>13%</td>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Ozone+NO2</td>
<td>0.00</td>
<td>115</td>
<td>96.15</td>
<td>62%</td>
<td>ug/m(^3)</td>
<td></td>
</tr>
<tr>
<td>Particles</td>
<td>0.00</td>
<td>0.99</td>
<td>0.68</td>
<td>30%</td>
<td>mg/m(^3)</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>-0.60</td>
<td>32.20</td>
<td>13.69</td>
<td>27%</td>
<td>°C</td>
<td></td>
</tr>
</tbody>
</table>

Min = minimum value, Max = maximum value, STD = standard deviation.

\(^*\)The values of humidity are presented as percentage, which are transformed to decimal for computation.

**B. DATASET PREPARATION**

Different datasets were generated according to the O Model and ST Model. The missing data points were deliberately set for the same observations in the datasets for the two models so that results can be compared. The estimated values were compared to the true values from the original datasets to compute the errors.

The dataset for the O Model contains 5,481 rows and 5 columns. Each row consists of five observation values measured by a sensor node (a sensor node can measure different qualities) installed on a bus at a specific time point; while each column represents an observation type.

The datasets for the ST model were prepared for each observation according to Algorithm 2. Each record in the datasets contains values for Sensor ID, Time, Latitude, Longitude, Altitude, Day, Hour, Minute, and Observation. If an observation value was set as missing, then its corresponding spatial information, e.g., Latitude, Longitude and Altitude, was also set as missing. Sensor ID and temporal information were used for location inference. The inferred location as well as the temporal information was used for training dataset preparation.

**C. EXPERIMENTS**

In total, 59,810 queries were prepared for the ST model. The attribute values were normalised by taking the ratio of the difference between the attribute value and its mean to the variance of that attribute. Then for each query, we inferred its location and then created a training dataset to train a regression model.

Training of the SVR with the ST model and estimation of the missing values for the queries are presented in Algorithm 3, TrainAndEstimate. The training set was created by

---

\(^1\) http://www.smartsantander.eu/
selecting those records that were measured within a spatial range (1 kilometre) and a temporal range (±6 hours). In lines 7 and 8, if the training set does not contain sufficient records for training, the training set is recreated empirically by relaxing the search criteria, i.e., expanding the range of location to 2 kilometres and time range to ±12 hours. In lines 9-11, if the trainingSet is still empty, then the query cannot be answered and the count for the unanswered queries is increased by one. Also, we considered the results as unreliable and excluded them in the evaluation if the number of training records is lower than ten (the approach is proposed in [15]; we adopted it for the sake of comparison), and skipped the training. In lines 12-13, all the missing values were then estimated based on the trained models and compared to the true values in the original dataset. Finally, in line 16, the estimation errors were computed.

Different kernels for SVR were used and we found that the Linear and Gaussian kernels significantly outperformed the Polynomial one. We compared the performance of SVR and the ST model with other methods including LWR, RANSAC+SVR, compressive sensing, and kNN (average of the k-nearest neighbours, and Mean (average of all values for that observation in the dataset). The LWR method selects the most similar observations in the dataset). The LWR method selects the most similar k (k=10 in the experiments) records with regard to the query, and tries to solve a LASSO problem (optimisation of L2 loss and L1 regularisation). It tries to find a linear function by solving the following problem:

$$\text{minimise} \left( \frac{1}{2l} \sum_{i=1}^{l} (y_i - b - \langle w, x_i \rangle)^2 + \lambda \| w \|_1 \right) \quad (12)$$

where l is the size of training set; y represents observations; b is a constant bias; $\langle \cdot, \cdot \rangle$ denotes the dot product; w is a vector of parameters; x indicates predictor vectors, and $\lambda$ is a nonnegative parameter influencing the number of nonzero components of w. RANSAC is an iterative algorithm to fit models from data that contains outliers and has been primarily used as an outlier detection method. In our experiments, we did not use it directly for missing value estimation; instead, we used it to first select the inliers and then applied SVR with Gaussian kernel to estimate the missing data. We wished to see if this could further improve the performance in terms of RMSE. We also re-implemented compressive sensing via matrix completion with norm minimisation (i.e., nuclear norm or spectral norm) for comparison. This method does not need any training and can recover all the missing information in the matrix in a single run. The experiments showed that the nuclear norm performed much better than the spectral norm, so we only report results of matrix completion with the nuclear norm.

For the O Model, the training set was created by removing all the records with missing values. In total we prepared 65,760 queries. The training was skipped if the training set was empty or the number of records in the training set was lower than ten (it is required that each record needs to have values at least two co-located sensors). In this case, the query could not be answered and the number of unanswered queries was incremented.

The experiments were conducted with different missing rates (i.e., 10%, 20%, 30%, 40%, 60% and 80%), to cover the situations similar to those in real world mobile sensing applications as well as extreme situations. The settings allowed us to evaluate the consistency of the estimation performance under different cases. We also tried different combinations of the models and regression techniques for training and estimation in order to gain a comprehensive view of the results.

We plotted curves of the original values and estimated values for all missing data in the test dataset. Individual missing data points were sorted according to their true values. For observations of CO, Humidity and Temperature, the estimated values using the ST model and SVR successfully captured the trends of the curves with the true values. For Ozone+NO2 and Particles, the estimations were less accurate, especially for low values. However, the method did attempt to produce many low values towards the true ones. Large deviations to the true values were identified when values of CO were extremely large and values for Particles and Ozone+NO2 were extremely small. This could be attributed to the outliers and unbalanced distributions of the values. In general, the ST model produced smoother curves and fewer local peaks than the O model.
be considerably high. This often leads to insufficient amount of training data for regression analysis. In situations with extremely high missing rates (e.g., 80%), being able to answer a query (with reasonable accuracy) has higher priority over estimation accuracy. This measure can provide us a clear picture on the estimation capabilities of the ST and O models with increasing missing rates.

The second evaluation method measures the estimation errors. The standard Root Mean Squared Error (RMSE), which measures the deviation between the estimated values and the true values, was used. It is calculated by using the Equation (13),

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}
\]

where \( n \) is the number of missing data points that has been estimated; \( \hat{y}_i \) is the estimated value for the \( i \)th missing data point, and \( y_i \) is the corresponding true value.

However, RMSE does not consider the actual magnitude of the observed values. It is also difficult to compare the performance of the same algorithm over different observations as the ranges of their actual values can be considerably different. Therefore, the Mean Absolute Percentage Error (MAPE) was also adopted, which expresses errors as a percentage and eliminates the influence of the ranges of true values. MAPE is calculated by using the Equation (14). In the evaluation, RMSE and MAPE are computed for each of the observations separately.

\[
MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|
\]

The third evaluation method is the execution time, which measures the time taken for training the regression models and estimating the missing values. It allows us to assess the efficiency of the methods. This is important as in the context of smart city applications, the computation may need to be performed on capability-constrained devices, such as gateways of wireless sensor networks, or at a data centre where large amount of real-time streaming data needs to be processed efficiently.

### B. NUMBER OF UNANSWERED QUERIES

Table IV provides the statistics on the capabilities of the ST and O models in answering queries under different missing rates. The results were generated by using the SVR with Gaussian kernel.

With increasing missing rates, there were more and more queries that could not be answered using the O model. When the missing rate was 10%, this was around 5% of the queries (149 out of 2,740 queries). When the missing rate was increased to 40%, more than half of the 10,960 queries couldn’t be answered (52%). When the missing rate was 80%, almost all of the 21,920 queries could not be answered. It is obvious that estimation using the O model with high missing rates becomes extremely difficult, if not impossible.

On the contrary, estimation performance of the ST model was consistent with different missing rates. For missing rates less than 60%, only less than 3% of the queries could not be answered. When the missing rate was at 80%, only 825 out of 19,945 missing data points could not be estimated. The results could be further improved if the condition in Algorithm 2 DatasetPreparation were relaxed, e.g., increasing the temporal range. That would reduce the percentage of unanswered queries to less than 2%.

The O model needs to take the observation values of co-located sensors as inputs for the regression analysis. When the missing rate increases, it becomes more and more difficult to find enough training data from the co-located sensors. In contrast, the ST model applies temporal and spatial metadata as the inputs to the regression analysis. With the location inference algorithm presented earlier, it was straightforward to derive the approximate spatial information for a missing observation value. The results showed that the ST model is tolerant to the datasets with very high missing rates, and estimation methods built on the ST model are robust. All these features make the ST model more applicable in the real-world scenarios than the O model.

### C. ROOT MEAN SQUARED ERROR AND MEAN ABSOLUTE PERCENTAGE VALUE

Table V provides an overview on the RMSEs generated by different techniques with the ST and O models, averaged over all missing data points (abbreviations of the different
techniques are explained below the table). Only results with 10%-30% missing rates were illustrated, as there were a large number of unanswered queries with the O model when missing rate was larger than 30%. Each row shows the RMSEs of one observation with a particular missing rate; and each column indicates the method used. The missing data points that could not be estimated (i.e., unanswered queries) were not included in calculating RMSEs and MAPEs.

With missing rates below 30%, the overall performance of the ST model in general was better than the O model as can be identified from Table V, although the difference between the best results produced by the two models is hardly distinguishable. Performance of SVR-G was very consistent and stable; with different missing rates and observation types, it always generated results close to the best ones. The method of RANSAC+SVR-G first uses RANSAC to filter out outliers and then applies SVR-G to train regression models. In general, it outperformed other techniques, showing that RANSAC did help in outlier detection and learning a better estimation model. It produced the lowest RMSEs for humidity and particles under different missing rates; performance for CO and Temperature was comparable to the best results generated by others; nevertheless, the errors for Ozone+NO2 were notably high. Compressive sensing also performed reasonably well with the ST model and produced results close to the best ones. However, its performance was not stable with the O model and produced much higher RMSEs than other techniques for Ozone+NO2 and Temperature. As expected, performance of the linear SVR was worse than SVR-G in all situations. LWR produced much higher errors than all other techniques, especially for Ozone+NO2 and temperature, using both the O model and ST model.

The results clearly show that the temporal and spatial features can be used to build reliable estimation models and produce results comparable to or even better than those generated by conventional models (O model). The advantages of the ST model can be easily observed by jointly considering the figures in Table IV and Table V. On one hand, it is tolerant to situations with extremely high missing rates; on the other, it also helps generate lower estimation errors.

Another notable finding was that with LWR, the RMSEs for some of the observations, e.g., Ozone+NO2 and temperature, were extremely high, which seemed inconsistent with the results reported in [15]. We suspected that it could be attributed to the existence of outliers. We then further removed the potential outliers by fitting a simple linear regression model for every observation and comparing the predicted value with the one in the dataset. If the difference between two values is more than one and half of the predicted value with the one in the dataset. If the

<table>
<thead>
<tr>
<th>Missing Rate</th>
<th>LWR</th>
<th>SVR</th>
<th>SVR-G</th>
<th>RAN+SVR-G</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>3.612</td>
<td>3.410</td>
<td>3.390</td>
<td>3.401</td>
<td>2.964</td>
</tr>
<tr>
<td>0.2</td>
<td>3.906</td>
<td>3.819</td>
<td>3.814</td>
<td>3.815</td>
<td>2.943</td>
</tr>
<tr>
<td>0.3</td>
<td>3.459</td>
<td>3.205</td>
<td>3.195</td>
<td>3.199</td>
<td>2.922</td>
</tr>
<tr>
<td><strong>Humidity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.617</td>
<td>0.111</td>
<td>0.104</td>
<td>0.110</td>
<td>0.214</td>
</tr>
<tr>
<td>0.2</td>
<td>1.032</td>
<td>0.116</td>
<td>0.111</td>
<td>0.117</td>
<td>0.245</td>
</tr>
<tr>
<td>0.3</td>
<td>1.466</td>
<td>0.120</td>
<td>0.117</td>
<td>0.122</td>
<td>0.266</td>
</tr>
<tr>
<td><strong>Ozone+NO2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>54.110</td>
<td>39.875</td>
<td>35.581</td>
<td>42.131</td>
<td>38.658</td>
</tr>
<tr>
<td>0.2</td>
<td>54.914</td>
<td>40.208</td>
<td>34.451</td>
<td>41.247</td>
<td>38.704</td>
</tr>
<tr>
<td>0.3</td>
<td>152.958</td>
<td>37.682</td>
<td>32.644</td>
<td>38.037</td>
<td>30.941</td>
</tr>
<tr>
<td><strong>Particles</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.710</td>
<td>0.285</td>
<td>0.288</td>
<td>0.284</td>
<td>0.310</td>
</tr>
<tr>
<td>0.2</td>
<td>1.274</td>
<td>0.298</td>
<td>0.303</td>
<td>0.299</td>
<td>0.335</td>
</tr>
<tr>
<td>0.3</td>
<td>1.420</td>
<td>0.300</td>
<td>0.297</td>
<td>0.296</td>
<td>0.353</td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>20.382</td>
<td>3.339</td>
<td>3.172</td>
<td>3.373</td>
<td>7.015</td>
</tr>
<tr>
<td>0.2</td>
<td>49.804</td>
<td>3.364</td>
<td>3.119</td>
<td>3.235</td>
<td>7.906</td>
</tr>
<tr>
<td>0.3</td>
<td>56.769</td>
<td>3.457</td>
<td>3.351</td>
<td>3.464</td>
<td>9.088</td>
</tr>
</tbody>
</table>

LWR = Locally Weighted Regression, SVR = Support Vector Regression with linear kernel, SVR-G = SVR with Gaussian kernel, CS = Compressive Sensing, RAN+SVR-G = RANSAC+SVR-G.

The minimum RMSE in a row is in bold.

VOLUME XX, 2017
RMSes produced by Mean, kNN and SVR-G were stable with different missing rates, while those generated by LWR and SVR fluctuated under several settings. It was difficult to draw a clear conclusion on which method performed the best in terms of RMSes. SVR-G generated the lowest RMSes for CO and Humidity, while kNN generated the lowest RMSes for Ozone+NO2 and Particles. RANSAC+SVR-G provided the lowest RMSes for Temperature, but much larger errors for CO and Ozone+NO2. Performance of the LWR was the worst among all the methods. One potential reason is that when data becomes more and more sparse with increasing missing rate, the selected near neighbours in fact could be quite far from the query point. RMSes tended to be much higher if more weights were incorrectly assigned to those selected neighbours. RMSes generated by compressive sensing were also larger than SVR-G, kNN, and RANSAC+SVR-G. This indicated that compressive sensing via matrix completion does not offer any advantages over other methods when missing rates are extremely high.

**TABLE VI**

*ROOT MEAN SQUARED ERRORS PRODUCED BY DIFFERENT TECHNIQUES USING THE ST MODEL, WITH 10%-80% MISSING RATES.*

<table>
<thead>
<tr>
<th>MR</th>
<th>Mean</th>
<th>LWR</th>
<th>SVR</th>
<th>SVR-G</th>
<th>kNN</th>
<th>RAN+SVR-G</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.306</td>
<td>0.220</td>
<td>0.205</td>
<td><strong>0.200</strong></td>
<td>0.207</td>
<td>0.296</td>
<td>0.240</td>
</tr>
<tr>
<td>0.2</td>
<td>0.288</td>
<td>0.604</td>
<td>0.708</td>
<td>0.231</td>
<td><strong>0.230</strong></td>
<td>0.272</td>
<td>0.271</td>
</tr>
<tr>
<td>0.3</td>
<td>0.331</td>
<td>1.357</td>
<td>0.617</td>
<td><strong>0.198</strong></td>
<td>0.221</td>
<td>0.319</td>
<td>0.255</td>
</tr>
<tr>
<td>0.4</td>
<td>0.314</td>
<td>0.236</td>
<td>0.959</td>
<td>0.211</td>
<td>0.215</td>
<td>0.301</td>
<td>0.253</td>
</tr>
<tr>
<td>0.6</td>
<td>0.331</td>
<td>0.443</td>
<td>0.211</td>
<td><strong>0.203</strong></td>
<td>0.215</td>
<td>0.322</td>
<td>0.263</td>
</tr>
<tr>
<td>0.8</td>
<td>0.337</td>
<td>0.354</td>
<td>0.209</td>
<td><strong>0.192</strong></td>
<td>0.194</td>
<td>0.335</td>
<td>0.271</td>
</tr>
<tr>
<td>Humidity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.103</td>
<td>0.414</td>
<td>0.209</td>
<td><strong>0.084</strong></td>
<td>0.090</td>
<td>0.097</td>
<td>0.102</td>
</tr>
<tr>
<td>0.2</td>
<td>0.111</td>
<td>0.161</td>
<td>0.088</td>
<td><strong>0.083</strong></td>
<td>0.087</td>
<td>0.085</td>
<td>0.100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.106</td>
<td>0.258</td>
<td>0.386</td>
<td><strong>0.084</strong></td>
<td>0.089</td>
<td>0.089</td>
<td>0.102</td>
</tr>
<tr>
<td>0.4</td>
<td>0.107</td>
<td>0.197</td>
<td>0.092</td>
<td><strong>0.085</strong></td>
<td>0.091</td>
<td>0.096</td>
<td>0.102</td>
</tr>
<tr>
<td>0.6</td>
<td>0.106</td>
<td>0.270</td>
<td>0.283</td>
<td><strong>0.085</strong></td>
<td>0.089</td>
<td>0.094</td>
<td>0.101</td>
</tr>
<tr>
<td>0.8</td>
<td>0.112</td>
<td>0.357</td>
<td>0.109</td>
<td><strong>0.086</strong></td>
<td>0.089</td>
<td>0.099</td>
<td>0.102</td>
</tr>
<tr>
<td>Ozone+NO2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>20.592</td>
<td>21.041</td>
<td>17.677</td>
<td>17.391</td>
<td><strong>16.311</strong></td>
<td>21.952</td>
<td>17.714</td>
</tr>
<tr>
<td>0.2</td>
<td>20.094</td>
<td>18.316</td>
<td>15.861</td>
<td>14.618</td>
<td><strong>13.870</strong></td>
<td>21.381</td>
<td>17.269</td>
</tr>
<tr>
<td>0.3</td>
<td>19.517</td>
<td>19.264</td>
<td>15.655</td>
<td>15.610</td>
<td><strong>14.827</strong></td>
<td>20.664</td>
<td>16.872</td>
</tr>
<tr>
<td>0.4</td>
<td>20.295</td>
<td>33.032</td>
<td>15.338</td>
<td>15.069</td>
<td><strong>14.465</strong></td>
<td>21.503</td>
<td>18.356</td>
</tr>
<tr>
<td>0.6</td>
<td>19.014</td>
<td>34.193</td>
<td>16.281</td>
<td>14.656</td>
<td><strong>14.105</strong></td>
<td>20.063</td>
<td>17.737</td>
</tr>
<tr>
<td>Particles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.259</td>
<td>0.369</td>
<td>0.272</td>
<td>0.249</td>
<td><strong>0.239</strong></td>
<td>0.253</td>
<td>0.242</td>
</tr>
<tr>
<td>0.2</td>
<td>0.254</td>
<td>0.707</td>
<td>0.731</td>
<td>0.243</td>
<td><strong>0.238</strong></td>
<td>0.248</td>
<td>0.239</td>
</tr>
<tr>
<td>0.3</td>
<td>0.257</td>
<td>0.545</td>
<td>0.341</td>
<td>0.241</td>
<td><strong>0.232</strong></td>
<td>0.243</td>
<td>0.237</td>
</tr>
<tr>
<td>0.4</td>
<td>0.253</td>
<td>0.733</td>
<td>0.367</td>
<td>0.244</td>
<td><strong>0.236</strong></td>
<td>0.251</td>
<td>0.239</td>
</tr>
<tr>
<td>0.6</td>
<td>0.257</td>
<td>0.639</td>
<td>0.255</td>
<td>0.244</td>
<td><strong>0.235</strong></td>
<td>0.272</td>
<td>0.237</td>
</tr>
<tr>
<td>0.8</td>
<td>0.254</td>
<td>0.633</td>
<td>0.304</td>
<td>0.242</td>
<td><strong>0.235</strong></td>
<td>0.250</td>
<td>0.238</td>
</tr>
<tr>
<td>Temperature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>4.736</td>
<td>4.117</td>
<td>2.670</td>
<td><strong>2.363</strong></td>
<td>2.791</td>
<td>2.488</td>
<td>2.876</td>
</tr>
<tr>
<td>0.2</td>
<td>4.869</td>
<td>4.163</td>
<td>2.650</td>
<td>2.326</td>
<td><strong>2.646</strong></td>
<td>2.433</td>
<td>2.655</td>
</tr>
<tr>
<td>0.3</td>
<td>4.994</td>
<td>4.644</td>
<td>6.866</td>
<td>2.356</td>
<td>2.702</td>
<td><strong>2.347</strong></td>
<td>2.723</td>
</tr>
<tr>
<td>0.4</td>
<td>4.833</td>
<td>7.793</td>
<td>9.805</td>
<td>2.385</td>
<td>2.721</td>
<td><strong>2.304</strong></td>
<td>2.798</td>
</tr>
<tr>
<td>0.6</td>
<td>4.885</td>
<td>8.526</td>
<td>2.904</td>
<td>2.479</td>
<td>2.757</td>
<td><strong>2.442</strong></td>
<td>2.811</td>
</tr>
<tr>
<td>0.8</td>
<td>4.643</td>
<td>8.879</td>
<td>2.840</td>
<td>2.562</td>
<td>2.751</td>
<td><strong>2.532</strong></td>
<td>2.805</td>
</tr>
</tbody>
</table>

MR = Missing Rate, LWR = Locally Weighted Regression, SVR = Support Vector Regression with linear kernel, SVR-G = SVR with Gaussian kernel, kNN = k Nearest Neighbours, CS = Compressive Sensing, RAN+SVR-G = RANSAC+SVR-G. The minimum RMSE in a row is in bold.

With RMSE, it was difficult to tell which method (i.e., SVR, SVR-G, RAN+SVR-G, or kNN) is the best for mobile sensing data estimation as the measure does not take the magnitude of the observed values into consideration. We further evaluated the performance of different models using MAPE. The comparison results were plotted in Figure 1. The patterns were quite different from those with RMSes: SVR-G and RAN+SVR-G had clear advantages over other methods. SVR-G produced the lowest MAPEs for Humidity, Ozone+NO2 and Particles. The figures were very stable for all observation types with different missing rates. Performance of the RAN+SVR-G was comparable and it generated the lowest MAPEs for CO and Temperature; however, its performance for Ozone+NO2 and Particles was not satisfactory. In particular, it produced significantly high MAPEs for Particles with 30% and 80% missing rates. Performance of kNN was slightly better than linear SVR, in most of the cases, they generated higher MAPEs than SVR-G and RAN+SVR-G. Performance of the other methods, i.e., CS, Mean and LWR, was much worse. CS was very sensitive
to high missing rates and produced significantly high MAPEs when missing rate was high (e.g., 60% and 80%).

D. EXECUTION TIME

Execution time provides an indication on the efficiency of the methods and was measured in milliseconds for both the training and estimation. It is important given the fact that in smart city applications, extremely large amount of sensor streaming data need to be processed and analysed in real-time, at distributed, less powerful platforms. This evaluation allowed us to find out if the ST model with different techniques could be efficiently implemented for practical uses. The measured training and estimation time for all experiments (i.e., average time of one query) was shown in Table VII (only for those methods that need training) and Table VIII, respectively. For each observation, the execution time was averaged for different missing rates. The least training and estimation time among all methods are in bold.

### Table VII

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>O Model</th>
<th>ST Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation</td>
<td>LWR</td>
<td>SVR</td>
</tr>
<tr>
<td>CO</td>
<td>181</td>
<td>1153</td>
</tr>
<tr>
<td>Humidity</td>
<td>546</td>
<td>207</td>
</tr>
<tr>
<td>Ozone+NO2</td>
<td>382</td>
<td>1908</td>
</tr>
<tr>
<td>Particles</td>
<td>576</td>
<td>227</td>
</tr>
<tr>
<td>Temperature</td>
<td>470</td>
<td>211</td>
</tr>
<tr>
<td>Average</td>
<td>431</td>
<td>741</td>
</tr>
</tbody>
</table>

LWR = Locally Weighted Regression, SVR = Support Vector Regression with linear kernel, SVR-G = SVR with Gaussian kernel, RAN+SVR-G = RANSAC+SVR-G.

The minimum values in a row are in bold.

As expected the training time dominated the total execution time and the estimation time was several magnitudes less than training time. Regression training using the O model took much more time than using the ST model (see Table VII), while the difference between estimation time using the two models was not much (see Table VIII). The training time for different observations also varied significantly with the O model. This was largely due to the unbalanced distributions of the original observation data as well as the existence of the outliers. However, as stated earlier, if these outliers were removed, then more than half of the queries could not be answered with 30% missing rate.

Execution time for all techniques with the ST model also had much less variation compared to those with the O model. Among all the methods that need training, SVR-G and linear SVR required much less time than other methods. With RAN+SVR-G, finding outliers using RANSAC took most of the training time. Training LWR using both the O and ST models was not as efficient as other methods. As kNN and CS do not need training, they are always more efficient than regression-based methods as can be seen from Table VIII.

E. DISCUSSION

Missing data rates in real world mobile sensing applications can be considerably higher than those in other applications. A desired estimation method should be able to efficiently perform the computation with high accuracy on the fly, even when majority of the observation data is missing.

Through the experiments and evaluations, we can easily identify the advantages of using the ST model for estimating missing data. First, estimation techniques trained with the ST model could answer around 96% of the queries even when 80% of the observation data is missing; while with the O model 99% of the 21,920 queries could not be answered. Second, the estimation errors, measured by RMSE and MAPE, were lower with the ST model. When trained with SVR with Gaussian kernel, the errors of the ST model remained stable with increasing missing rates. This indicates that the ST model, built on the spatial and temporal metadata of the observations, is in fact effective in estimating missing data. Third, the ST model allows the elimination of potential outliers without having a negative impact on the estimation accuracy and capability. In fact, the estimation performance in terms of accuracy and execution time were further improved when outliers were removed. Fourth, the execution time for training and estimation using the ST model was substantially less than the time required for the O model.

We found that several methods offered competitive performance using the evaluation metrics. This raised a question regarding the choice of regression or smoothing
techniques to be used with the ST model. In terms of RMSE and MAPE, it was observed that SVR-G and RAN+SVR-G outperformed others. RANSAC can help in outlier detection and allow SVR-G to be trained with the selected inliers. However, it only improved the performance in a number of limited cases while produced larger errors in other cases. It did not show any clear advantages over SVR-G. Results produced by kNN were fair in terms of RMSE and MAPE. Surprisingly, LWR, which was reported to have superior performance in the existing work, performed poorly on the SmartSantander mobile sensing dataset. In terms of execution time, kNN and compressive sensing-based methods are always more efficient than regression-based ones as no training is needed. CS based methods can recover missing data in a matrix with a single run, however, the generated RMSEs and MAPEs were notably higher than SVR-G and RAN+SVR-G. The figures in Table VII and Table VIII show that it is computationally feasible to run those methods even on commodity computers. Estimating missing values in several or tens of milliseconds is sufficient for the requirements of smart city applications. SVR-G based on the ST model represents a reasonable trade-off between estimation error and time, compared to other methods. Furthermore, estimation using regression can be further optimised to make the process more efficient, i.e., to reuse trained models in estimating more missing observations that are spatially and temporally related.

VI. CONCLUSION AND FUTURE WORK

Although the problem of estimating missing data from incomplete datasets has been well investigated in the literature, the existing models and techniques have difficulties in dealing with datasets having very high missing rates, such as those collected from mobile sensing scenarios. A spatio-temporal model was proposed to solve the problem with the Support Vector Regression. The results were compared to those generated by the state-of-the-art and benchmark methods. It showed that the overall performance of the proposed method was better than others in terms of estimation accuracy. More importantly, it was tolerant to datasets with extremely high missing rates and its performance was stable with increasing missing rates. Evaluation on execution time confirmed that it can be implemented on distributed, capability constrained computing devices, although it is less efficient than methods that do not need training.

In the experiments, when the distribution of the observation data is highly unbalanced and the dataset contains outliers, the estimation accuracy tended to be very low. We found that with the ST model, elimination of the outliers in fact improved the estimation performance. However, this might have removed data related to some important real-world events. This has not been taken in account in the current work. One future work is to design methods to discover patterns and extract real-world events from the mobile sensing datasets. In the evaluation the proposed model does not show advantage over other estimation methods that do not need training in terms of execution time. Another future work is to further refine and optimise the regression models to avoid unnecessary training for better execution time. In particularly, we would investigate and evaluate some of the more recent advanced randomised learning models, e.g., stochastic configuration networks, to further improve the computation efficiency.

REFERENCES


YUCHAO ZHOU received his Ph.D. degree in electronic engineering at the University of Surrey in 2018. He did his B.S. degree in telecommunications engineering with management from a joint programme between Beijing University of Posts and Telecommunications, China and Queen Mary University of London, UK, in 2011 and M.Sc. degree in communications networks & software from the University of Surrey, Guildford, UK, in 2012. He is currently a Research Fellow in the Institute for Communication Systems, at the University of Surrey. His research interests include semantic Web, search techniques for the Web of Things, and IoT applications in smart cities.

SUPARNA DE received her Ph.D. and MSc degrees in Electronic Engineering from the University of Surrey in 2009 and 2005, respectively. She is currently a Senior Research Fellow in the Institute for Communication Systems, at the University of Surrey. She has been leading technical work areas related to various aspects of service provisioning and data analysis in the Internet of Things domain in several EU projects such as TagItSmart, iKaaS, IoT.est, iCore and IoT-A. Her research has been supported by grants from the EC H2020 and FP7 programs and through DTI, UK-funded programs. Her current research interests include discovery and retrieval methods, Web of Things, semantic association analysis and knowledge engineering methods. She is a member of IEEE and ACM.

WEI WANG received his PhD in Computer Science from the University of Nottingham, UK in 2009. He was an assistant professor at the University of Nottingham Malaysia Campus from 2009 to 2011, and a postdoctoral research fellow at the Centre for Communication and Systems Research, the University of Surrey, UK from 2011 to 2013. He is currently an associate professor at the Department of Computer Science and Software Engineering, Xi’an Jiaotong Liverpool University, China. His research interests include Internet of Things, knowledge discovery, semantic search, service computing, and data mining.

RUILI WANG received his Ph.D. degree in Computer Science from Dublin City University, Dublin, Ireland. He is currently Professor of Information Sciences in the Institute of Natural and Mathematical Sciences, Massey University, Auckland, New Zealand, and is also the Director of the Centre of Language and Speech Processing. His research interests include speed processing, language processing, image processing, data mining, and intelligent systems. Dr Wang serves as an Associate Editor of and a member of the editorial boards for international journals such as Knowledge and Information Systems, Applied Soft Computing and others. He has received one of the most prestigious research grants in New Zealand, i.e., the Marsden Fund.

KLAUS MOESSNER is a Professor in the Institute for Communication Systems, at the University of Surrey. He was the founding chair of the IEEE DYSBAN Working Group (WG6) on Sensing Interfaces for future and cognitive communication systems. He was involved in the definition and evaluation of cooperation management between autonomous entities, in the UniverSelf project, and was technical manager of the iCore project and has the same role in the H2020 project CPaaS.io; he was project leader of IoT.est, SocIoTal, and currently leads the iKaaS project as well as working area 6, on System Architecture in the 5G Innovation Centre at the University of Surrey. His research interests include cognitive networks, knowledge generation, as well as reconfiguration and resource management and he is a senior member of IEEE.