Real-Time Automatic Air Pollution Services from IOT Data Network

Claudio Badii, Stefano Bilotta, Daniele Cenni, Angelo Difino, Paolo Nesi, Irene Paoli, Michela Paolucci
Department of Information Engineering, DISIT Lab, University of Florence, Florence, Italy
<name>.<surname>@unifi.it, https://www.disit.org, https://www.snap4city.org

***Abstract*— In recent years, there is an increasing attention on Air Quality and derived services. In most cases, the main objective consists in providing services independently on the number of sensors, and on any position of the users. On the other hand, sensors are collected in limited positions, then a dense grid is computed to satisfy the needs of services such as: conditional routing, alerting on data values for personal usage, general heatmap production for Dashboards in control room of the operators, and for web and mobile applications for the city users. The paper formalizes the development process and describes how it is possible to automatically integrate Data Analytics in data flow real time processes, called in the paper IoT Applications. Two interpolation methods have been compared and validated in order to assess the accuracy. Interpolation errors trends have been used to detected devices' dysfunctions on sensors. The specific case presented in this paper refers to the data and the solution of Snap4City for Helsinki. Snap4City (https://www.snap4city.org) has been developed as a part Select4Cities PCP of the European Commission, and it is presently used in a number of cities and areas in Europe.**

Keywords*— *smart city, pollution, interpolation, IOT application, dashboards, sensors network, early warning system, devices dysfunction

I. Introduction

Most of the cities and regions are increasing their attention to the real-time monitoring of environmental and weather parameters. In the past, the usage of environmental data sensors has been a prerogative of public administrations. The main difficulties have been due to the very limited number of sensors for a whole city. So that, it does not mean that few values are valid for the whole city as well. In fact, it is reasonable to have large differences on data when the air quality data is measured on a high traffic road rather than in a garden, just on the back of a house located on the same road. Recently, there is a deeper understanding of the environmental parameters (for example, PM_{10}, $PM_{2.5}$, CO, CO_2, SO_2, O_3, H_2S, NO, NO_2, NO_x, etc.), and how much they are influenced by the context, city's structures. Thus, cities have a strong interest on air quality to properly regulate urban mobility as well as other activities, to improve the quality of life.

Today, there is also the direct interest and the economic possibility of city users to buy and install their own sensors to monitor data values on their location. And, in most cases, they are putting the resulting data at disposal of the community. Even if some of low-cost sensors are of low quality, the increased number of data, and the procedures for their calibration can compensate. The city users are interested in hosting sensors to take decisions on their activity in the city, based on the measured data. For example, opening the back windows in certain cases, rather than the front ones, or getting out in the garden with the baby, or choosing the path for jogging. Thus, the air quality influences decisions of city users, e.g., on routing, on walking, on selecting locations to spend time with family and/or sport.

Nowadays, the estimation of the concentration of air pollution in smart cities is an important issue. The literature dealing with the monitoring of air quality and health condition is extensive. In order to better monitoring air pollution it's important to estimate pollution levels in any point of the city, while the sensors are usually only a limited number. To this aim, interpolation approaches can be developed to handle big data and support real-time analysis. The use of deterministic and stochastic interpolation methods to estimate unknown values such as inverse distance weighting (IDW), kriging, spline, radial basis function, natural neighbour or trend surface are widely used in literature. Only few studies explored a system for a real-time fast air pollutant visualization [1], [2], especially at a large geographic scale.

This paper presents the Snap4City solution for the development and integration of Data Analytic processes for Smart Cities. This paper presents three main contributions. The first describes the development process and related tools, that permitted at the developers to create their own Data Analytics algorithms and see them exploited as a basis of smart solutions (e.g., for computing heatmaps, predictions, etc). The second contribution consists in the solution for creating pollution heatmaps, and at the same time providing support for the implementation of Smart Services such as conditional routing on pollutant, alerting/warning on the basis of pollutant values, providing users with general information via mobile app and dashboards, may be as a support of What-If Analysis. The third contribution regards the comparison of different algorithms for computing dense grid values of pollutants, based on the scattered sensor data coming from the IoT sensors. On this regard, the error analysis allowed us to validate the solution, and to exploit the error analysis as a tool for detecting device failures: on the basis of this it is possible to understand anomalies on devices, which can help to avoid taking into account sensors' data that can compromise the service quality. In the cases of failure detection, a warning is sent to the data ingestion process manager.

The paper is organized as follows. In **Section** II, Snap4City Framework and Development facilities are described. **Section** III presents the general workflow and interpolation algorithms at the basis of the creation of a number of smart services, based on the sensors data assessing environmental variables. The smart services could be computing green path routing (for jogging, walking with kids or with highly sensitive people), sending personal warning on specific points, representing heatmaps on mobile and web page, etc., at the support for at disposal on mobile, and web apps and dashboards. **Section** IV provides a description of the Helsinki's city use case scenario of Helsinki, adopted to identify and validate the models and the framework. It is important to note that, in Helsinki the official sensors'

978-1-7281-8086-1/20/$31.00 ©2020 IEEE

network for air quality assessment has been enriched with data coming from IOT devices installed and manage by city users. In **Section** V, the approach used for detecting anomalies that can be useful for the detection of problems related to device sensors, and thus for increasing the quality in computing smart service data. Conclusions are drawn in Section VI**.**

II. Snap4City vs Data Analytic

The Snap4City framework is 100% open source and allows to: (i) ingest and manage Big Data coming from IoT devices, applications and services; (ii) compute actions for users, providing notifications and engagement; (iii) produce interactive visual analytics and dashboards supporting decision-making processes (useful for many different kinds of users: Public Administrations, final users, developers etc.) [1], [3]. Snap4City supports the creation of data-driven applications, based on MicroServices in Node-RED, also exploiting data analytics. The Snap4City solution can also be used for developing mobile and web apps, data flows and data processing [4]. Data Analytics can access and save data by using Advanced Smart City API, as the Dashboard and Mobile Apps, plus some internal APIs.

Snap4City provides generic Data Analytics Micro-Service where Data Analytics algorithms developed in R Studio can be put in execution. A collection of more than 150 Smart City Micro-Services has been developed as Nodes for the Node-RED programming tool. For the Data Analytic development, it is possible to access the Big Data store respecting the privacy and data licensing. Thus, the access is provided by using authenticated Smart City APIs. The solution allows to access at both historical and real time data, and to save the resulting data provided by the algorithms. For example, parking, pollution and heatmaps related predictions, the assessment of data quality, and anomaly detection. To this end, in **Figure** 1, the general schema for developing a Data Analytics algorithm is reported. The structure of Data Analytics processes has to permit to: activate the computing, combine the process with other processes. This is obtained by imposing the structure of Data Analytics processed by means of parameters, and reporting errors coding, share the process with multiple processes and other developers.

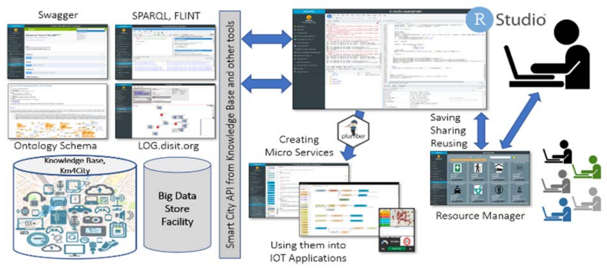


Figure 1 - Snap4City Real Time Data Analytics architecture.

In detail, in Snap4City a Data Analytic process can be developed by using R Studio or Python and the processes can be instantiated into a cluster management of containers. Each R Studio process has to present a specific API structure, which is accessible from an IoT Application as a MicroService, that is a node of the above-mentioned Node-RED visual programming tool for data flow. The Data Analytic code can be shared among developers with the so-called Resource Manager, that also allows the developers to perform queries.

III. Dense Map Estimation and Services Provided

According to the above sketched applications, the users should be capable to see satisfied the following requirements, and thus the user should be capable of:

A. *getting environmental data values close to a point: in which they live, work; where they would know the status, etc. This means that the solution must be capable to provide a value for environmental data in any GPS locations of the area. On the contrary, the sensor data are scattered and limited in the city.*

B. *getting environmental data values along a path in which they run, walk with kits, walk with sensitive people, etc., which can be regarded as an extension of Req.(A) for many points at the same time. This feature can be also extended to be a routing conditioned to the pollutant values, so as answering to questions such as: "which the less polluted path from point (i) to (ii) in terms of NOX".*

C. *receiving warnings/alerts on values of environmental data in a set of their preferred positions and values. The user should be capable to subscribe to be informed when certain pollutants are getting value over threshold. This means that the feature implemented to satisfy Req.(A) has to be automatically exploited for informing the users.*

D. *visualizing heatmaps on mobile and web page, on which it has to be possible to select/pick the point in which to get the value of the desired environmental data as in Req.(A) requirement.*

So that requirements from (**A**) to (**D**) have to be satisfied at the same time.

Thus, there is the needs of providing environmental data values in any GPS position of the city, and for a large number of different pollutants and weather data: PM_{10}, $PM_{2.5}$, CO, CO_2, SO_2, O_3, H_2S, NO, NO_2, NO_x, air temperature, air humidity, velocity of wind speed, dew point, etc. A solution could be to estimate a dense grid of values by interpolation of the sensor values. On the other hand, in a large city or geographic area, the computation of a dense grid every few meters would be computationally expensive, and storage consuming. For these reasons, we decided to find a compromise and computing an interpolated grid of values at the resolution useful for generating heatmaps on mobile Apps and Dashboards, which are the typical maps requested by the majority of users. Furthermore, we compute on demand the values in any GPS geographical coordinate when the user requests them. The workflow of this approach is described in **Figure** 2.

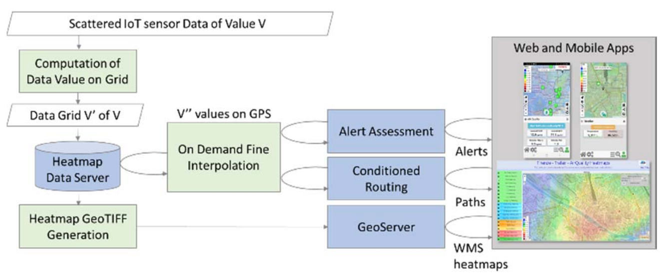


Figure 2 - Workflow for the data computation vs services. Green blocks are implemented as IoT Applications node.js. Scattered IoT Sensor Data are collected via Smart City API on the Big Data Storage.

In order to satisfy req.(D), the approach described in **Figure** 2 allows to create the GeoTIFF heatmaps that may be distributed on web and mobile Apps can be provided to several thousands of users. Thus, they can be download in tiled and zooming by many users at the same, from the GeoServer according to the WMS standard protocol. This approach also allows to produce sequences of heatmaps on the day, and to provide corresponding animations. Each GeoTIFF heatmap is generated on the basis of a color map, and thus the precise value in a certain area is discretized in a limited number of color level.

On the contrary when specific values at certain GPS point(s) have to be provided to satisfy Req.s (A), (B), (C) and (D), the access to the heatmap tiles as GeoTIFF is not efficient, neither precise. And thus, the original data exploited to create the heatmap tiles have to be exploited. So that an Heatmap Data Server is queried for extracting the values in smaller area and thus for interpolating the value in the specific GPS point requested. This means that the *On Demand Fine Interpolation* has to be capable to produce results in real time for each requested GPS point.

For Req.(C) the service has to be performed for each Variable, V, and each GPS Point, P, preferred by each User, U, getting a complexity of O(UPV).

The Req.(B) could be satisfied by a tool performing one access to the Heatmap Data Server for each possible path and road segment of each routing request, R, and thus for each point or Segment, S, of the path and value, V, obtaining a complexity of O(RSV).

The above discussion has provided the evidence of the complexity of the solution and the need of having a compromise in terms of size of the area in which the On Demand Fine Interpolation and related algorithm.

The following subsections present the description of two different solutions for the *Computation of the Data Value on Grid* and the solution adopted for the *On Demand Fine Interpolation*. It should be noted that reference data for a given area may be affected by severe errors. The lack of data from a certain device could restrict the area in which the results and services may be provided. While relevant errors/noise/anomaly in some device can produce severe errors in the resulting services. Thus, to avoid problem we removed the problematic data from the list of the scattered data to be used for the computing. During our research different techniques for the *Computation of the Data Value on Grid* have also been tested in terms of velocity and precision. At the end we decided to use deterministic methods instead of statistical ones, due to the small number of measures (i.e., sensor data locations) to interpolate. In fact, in the case of kriging, the data have to be used to choose a variogram or covariance function, which turns out to be very difficult in the case of a small number of data locations, resulting in insufficient information.

The following two subsections provide an overview of the interpolation techniques considered as the most appropriate for the solution: *Bivariate Interpolation method*, and the *Inverse Distance Weighting method (IDW)*.

E. Bivariate Interpolation Method

The creation of heatmaps for particulate matter may be based on a gridded bivariate interpolation for irregular data [5]. The bivariate interpolation method consists of five steps:

1. *triangulation* (i.e., partitioning of the area into a number of triangles) of the *x-y* plane. For a unique partitioning of the plane, the *x-y* plane is divided into triangles by the following steps. First, determine the nearest pair of data points and draw a line segment between the points. Next, find the nearest pair of data points among the remaining pairs and draw a line segment between these points if the line segment to be drawn does not cross any other line segment already drawn. Repeat the second step until all possible pairs are exhausted;
2. *selection* of several data points that are closest to each data point (sensor) and are used for estimating the partial derivatives;
3. *organization* of the resulting data with respect to triangle numbers;
4. *estimation* of partial derivatives at each data point;
5. *computation of* the interpolation at each output point.

In more details, the z value of the function at the point of coordinates *x-y* in a triangle is interpolated by a bivariate fifth-degree polynomial in *x* and *y*:

$$(x, y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k$$

The coefficients of the of the polynomial are determined by the given z values at the three vertexes of the triangle and the estimated values of partial derivatives (i.e. z, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}) at the vertexes, together with the imposed condition that the partial derivative of z by the variable measured in the direction perpendicular to each side of the triangle, must be a polynomial of third degree three; at least, in the variable measured along the side [5].

F. Inverse Distance Weighting Method

The Inverse Distance Weighting method (IDW) is a deterministic mathematical method widely used in the geoscience field [8]. The IDW is based on the premise that the predictions are a linear combination of data. Each interpolated value of a point is identified as the following equation:

$$z(x, y) = \sum_{i=1}^{n} w_i z_i$$

$$w_i = (1/d_i)^p / \sum_{k=1}^{n} (1/d_i)^p$$

where $z(x, y)$ is the interpolated value at the location (x, y); z_i is the observed value; d_i is the Euclidean distance between the point i and the interpolated point; and w_i is the weight for the point each point (x_i, y_i) and (x, y). The parameter p is the power value, i.e, the exponent that influences the weighting of w_i on z, [6]. In this research and the following experiments, we have tested several values of the parameters: then we setup the power value of p equal to 2.

G. On Demand Fine Interpolation

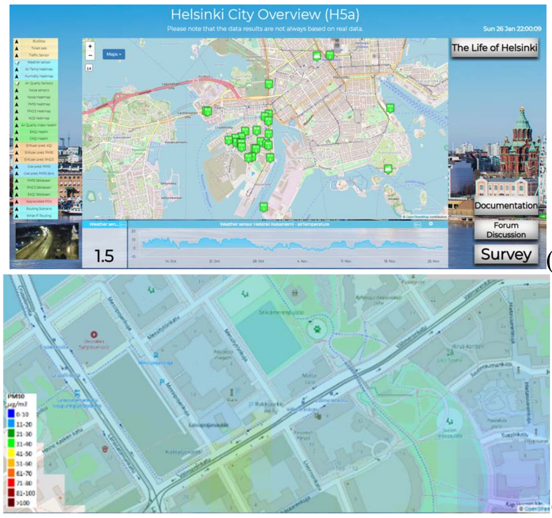
This subsection describes the solution adopted for the on-demand computation of the fine interpolation. The solution is based on performing an IDW interpolation within each square of the interpolated grid using Vincenty's distance [7] according to the steps reported in the following pseudocode.

Input: (x_k, y_k) = GPS position in the picked location k at time t
for each iteration i = 1 to 15
  compute $b_{(x_k,y_k)}$: a circular area around the picked location k of

radius equal to 2 km
get $\hat{z}(t) = (\hat{z}_1(t), \cdots, \hat{z}_j(t), \cdots, \hat{z}_n(t))$ vector of interpolated values at time t inside $b_{(x_k,y_k)}$
  if $\hat{z}(t) = \emptyset$, means no values in $b_{(x_k,y_k)}$ **then**
    $\text{radius}(b_{(x_k,y_k)}) = \text{radius}(b_{(x_k,y_k)}) * 2$
  continue
    for each GPS position $(x_{\hat{z}_j}, y_{\hat{z}_j})$ of $\hat{z}(t)$ vector
      compute $d_{(x_k,y_k),(x_{\hat{z}_j},y_{\hat{z}_j})}$: Vicenty distance between (x_k, y_k) and $(x_{\hat{z}_j}, y_{\hat{z}_j})$
        if $d_{(x_k,y_k),(x_{\hat{z}_j},y_{\hat{z}_j})} < 0.0000000001$ **then**
          $\hat{z}_k(t) = \hat{z}_j(t)$
        else compute $\hat{z}_k(t)$: interpolated value at time t in the k-th picked locations using IDW method
        end if
    end for
  end if
end for

IV. City of Helsinki's Scenario and Final Users Tools

In this section, it is shown how the solutions have been used in the case of City of Helsinki's. The environmental monitoring use case has been primarily performed in a new smart district of Jätkäsaari: a small connected island to the South of the city. In addition to 20.000+ future inhabitants and workplaces for 6000 people, including various hotels and office facilities, Jätkäsaari also encompasses the main part of Helsinki's passenger harbour. The large construction sites, the intensive and obstructed traffic, and the growing population create environmental challenges in Jätkäsaari. Thus, there was the need for a platform that can could integrate data from different sources and services, provide tools for data analytics about the current state of the air quality in different parts of Jätkäsaari. Therefore, in the context of Snap4City a number of Dashboards have been developed for the City Operators and ICT Officials, while a Mobile App has been developed and published both on the Google Play Store and the Apple Store for the Citizens and Tourists. The main Dashboard is reported in **Figure** 3a, while a detail regarding one heatmap is reported in **Figure** 3b.



(a)
(b)

Figure 3 – (a) Dashboard on Helsinki Environmental aspects, the map presents an high density of sensors in the Jätkäsaari Island, (b) Air Quality PM_{10} interpolation heatmap for a small area of Jätkäsaari Island (Akima method), transparency set at 30%. The legend describes a colour map of 9 colours, while the heatmap is produced with a color map of 150 colours.

The above mentioned dashboard can be found at https://www.snap4city.org/dashboardSmartCity/view/index.php?iddasboard=MTQwNg== Moreover, the mobile Apps allow the city users to access the heatmaps, and to subscribe to notifications that are activated on their POI when the selected pollutant is above the critical values. In **Figure** 4, the snapshot of the Mobile App "Helsinki in a Snap" is reported.

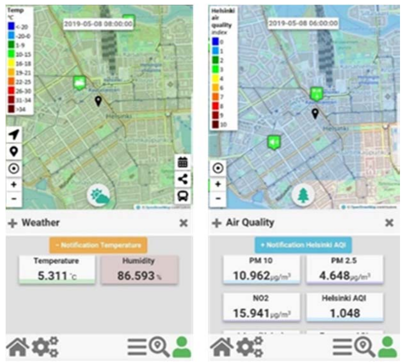


Figure 4 – Mobile App Helsinki in a Snap, visualization of Heatmap and subscription to alerts.

A. Validation in the context of Helsinki Data

In Jätkäsaari Island about 25 devices are collected and taken into account for the interpolation. Each device performs two measures of: PM_{10}, $PM_{2.5}$. For each kind of pollutant, the interpolated heatmap is computed, with a resolution of 2×2 m. In this way, final users have a dense overview of the quality in the area quality, situation within the smart zone and close to their homes. In the city, the heatmap resolution of the pollutant is 100×100 m. This justify the second step of interpolation for providing more personalized and precise values. In order to assess the precision, the interpolation accuracy of PM_{10} has been evaluated in terms of percentage error. The error evaluation of the interpolation approach is based on the alternate exclusion of selected air quality sensor in contributing to the model and using the excluded as true value for validation in that point on the basis of the estimation performed exploiting all the others. More precisely, for each time t we are going to estimate the error between the calculated interpolated value $\hat{z}_i(t)$ in at the position which locates the selected i-th sensor, with respect to the measured/real-time air quality value $z_i(t)$ from the i-th sensor. The absolute relative error of the i-th sensor e_r^i at time t is calculated as $e_r^i(t) = \frac{|\hat{z}_i(t) - z_i(t)|}{|z_i(t)|}$.

The accuracy of the whole approach is estimated by considering the same procedure for each data sensor at time t. In order to explore the performances of the interpolation approaches, two different error statistics have been calculated: the mean absolute percentage error (MAPE) per time slot and the root mean squared error (RMSE) per time slot:

$$\text{MAPE} = \frac{1}{S} \sum_{i=1}^{S} e_r^i(t);$$

$$\text{RMSE} = \sqrt{\frac{(\hat{z}_i(t) - z_i(t))^2}{S}};$$

where S is the number of sensors. About 2 months of data (from September 2019 to November 2019) have been used for the interpolation errors evaluation for the PM_{10} pollutant. The error measures have been computed for:

- weekends and working days: Ewe, Ewd. This produces a single error by considering all devices and time slots;
- weekends and working days per time slots, Ewe(t), Ewd(t). This produces an error value for each time slot, thus resulting in a trend as depicted in Figures 6 and 7.

Error Measures	Akima	IDW
MAPE	**0.69**	0.79
RMSE	**8.90**	12.20
MAPE-we	**0.60**	0.95
MAPE-wd	**0.70**	0.93
RMSE-we	**8.60**	10.70
RMSE-wd	**9.70**	17.00

Table 1 – Error measures vs interpolation methods

The resulting error measures, MAPE and RMSE for the two methods are reported in table 1. From the table the Akima method resulted to be the best method in terms of validation errors. In **Figures** 5 and 6, RMSE (a) and MAPE (b) Ew(t) are reported for working days.

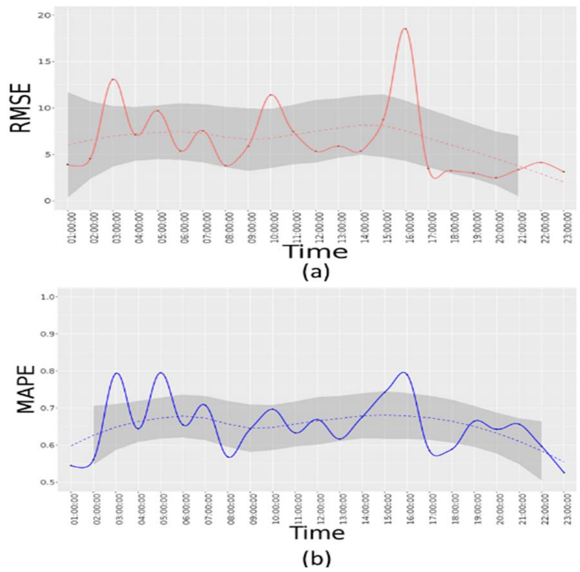


Figure 5 - PM_{10} working days RMSE (a) and MAPE (b) per time slots (Akima Method)

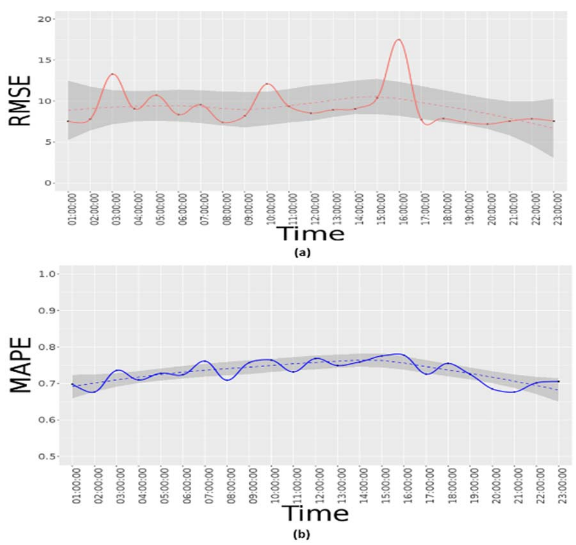


Figure 6 - PM_{10} working days RMSE (a) and MAPE (b) per time slots (IDW Method)

V. Anomaly Detection of Sensor Dysfunctions

Measurement errors can be caused by a variety of factors and some countermeasures need to be taken accordingly. When a sensor error occurs, it is important to examine the cause of measurement errors thoroughly, in order to implement anomaly detection systems. This is important for ensuring stable quality in measurement and in the creation of the interpolation map.

Errors in measures can be manifold.

- *Errors caused by the measurement system*: calibration error; measurement errors originating in the measurement system; deterioration of measurement accuracy over time (deterioration caused by wear in consumable components).
- *Errors caused by the user*: bad positioning of the devices, mishandling of the measurement system, different degrees of skill of the users; user-specific methods of reading the scale, turning off the device.
- *Errors caused by environmental conditions*: deformation of the measurement target caused by rapid changes in air quality measure; measuring in locations with varying air quality measure levels.

Errors caused by the measurement system or environmental conditions can be easily identified as a relevant change with respect to the average trend of the measure. When a device is left in the user's hands, one of the most likely error that may occur can be a dysfunction due to a bad positioning or turning off the device. The detection of these types of error is important because an alert message can be sent to the user which can solve the problem.

An idea of countermeasure is to use the validation error as a detector of the device's dysfunctions: it's possible to understand anomalies on devices comparing error trends with respect to the trend of the sensors of the same device. If the error trend is higher than the error confidence interval, it's likely to be a problem on the device. Once checked the error trend, the second step is to monitor the error on the other sensors (pollutant measure) installed on the same device. If the second measure trend error is similar to the first one, the presence of a dysfunction on the device is highly probable. This error control is quite different from a simple real-time measure trend control. A positive/negative change on the trend can be due to multiple factors and is possible to detect it by comparing the device with the nearest one while. In this case, the detected dysfunction is related to an un-correct trend over time. Such anomalies may often be useful to alert the users about a problem on the device by sending them warning messages.

To check possible dysfunctions, for each time slot *t*, all the points in the area of interest have an estimated/interpolated air quality value. The interpolation error and the confidence intervals are computed every 24 hours based on the validation method presented in Section IV. The confidence interval for the average error has been computed considering a period of 2 months (working days and weekends distinctly).

A. Basic Computational Approach

A computational approach for detecting dysfunction in real-time observations can be executed according to the following steps:

Input: S = sensor number
Input: $z_i(t)$ real-time value of the i-th sensor at time t
for each time t do
  for $i = 1$ to S **do**
  compute $\bar{z}_i(t)$ average value of the i-th sensor at time t
  compute $CI_{\bar{z}_i(t)}$ 95% confidence interval for the average value
  compute $\hat{z}_i(t)$ interpolated value in the i-th sensor location
  compute $e_r^i(t)$ error interpolation measure in the i-th sensor
  compute $\bar{e}_r^i(t)$ average error interpolation measure
  compute $CI_{\bar{e}_r^i(t)}$ 95% confidence interval for the average error
  if $|z_i(t) - \bar{z}_i(t)| > CI_{z_i(t)}$ **then**
    print high probability of error in measure in the i-th sensor
    mark i-th sensor on the map
  end if
  if $|z_i(t) - \bar{z}_i(t)| < CI_{z_i(t)}$ and $|\bar{e}_r^i(t)| > CI_{\bar{e}_r^i(t)}$ **then**
    print high probability of device dysfunction
    save the i-th sensor coordinates
    send alert message to the user
  end if
 end for
end for

Figure 7 shows PM_{10} interpolation error trends in terms of absolute percentage error. In **Figure** 7(a) the trend of the device with dysfunction (Device 6) and other five devices related' error trends are compared. In **Figure** 7(b) the trends for of the five devices without any dysfunctions are compared.

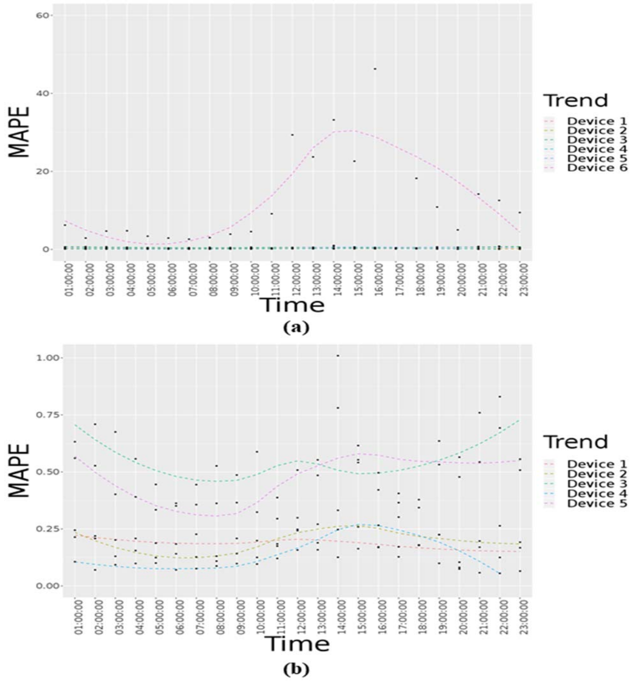


Figure 7 - Air Quality PM_{10} working days interpolation error trends per hour in terms of mean absolute percentage error for (a) six personal devices including the device with a dysfunction; (b) five personal devices

VI. CONCLUSIONS

The environmental data collected from devices hosted by city users and from data providers, have been used to provide informative view to city users regarding environmental data via some mobile App, and to provide detailed information about the Environmental data to city officials for decision making. In particular, the use of a personal device gives the possibility of city users to better monitoring a specific area. Further, the intention of putting the resulting data at disposal of the community has a double benefit: on one hand, it is possible to know the values of measured parameters on their premise; on the other hand it is possible to have a global view of data from the city with a denser sensor network, also thanks to the creation of interpolation heatmaps. However, the use of personal sensors has some disadvantages. For example, it is not possible to check if the device is correctly positioned (e.g., inside or outside the house). To solve this problem, a solution can be to monitor the trend of the interpolation mean absolute percentage errors. To check possible dysfunctions, one-week interpolation data has been used for the error's evaluation, and the detected dysfunction was related to a bad trend over time. Such anomalies may often be useful to alert the user about a problem on the device by sending them warning messages.

ACKNOWLEDGMENT

The authors would like to thank the European Union's Horizon 2020 research and innovation program for funding the Select4Cities PCP project (supported within the Snap4City framework) under the grant agreement No. 688196, and all the companies and partners involved. Snap4City and Km4City are 100% open source technologies and the platform of DISIT Lab can be accessed at https://www.snap4city.org.

REFERENCES

1. Bellini, Pierfrancesco, et al. *Smart City architecture for data ingestion and analytics: processes and solutions*. 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2018.
2. Kalo, Marc, et al. *Sensing air quality: Spatiotemporal interpolation and visualization of real-time air pollution data for the contiguous United States.* Spatiotemporal Analysis of Air Pollution and Its Application in Public Health. Elsevier, 2020. 169-196.
3. C. Badii, P. Bellini, D. Cenni, G. Martelli, P. Nesi, M. Paolucci, *Km4City Smart City API: an integrated support for mobility services*, 2nd IEEE International Conference on Smart Computing (SMARTCOMP 2016), St. Louis, Missouri, USA, 18-20 May 2016.
4. C. Badii, P. Bellini, A. Difino, P. Nesi, *Smart City IoT Platform Respecting GDPR Privacy and Security Aspects*, accepted for publication on IEEE Access, 2020. 10.1109/ACCESS.2020.2968741 https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8966344
5. Akima, Hiroshi. *A method of bivariate interpolation and smooth surface fitting for values given at irregularly distributed points.* Vol. 75. No. 70. US Department of Commerce, Office of Telecommunications, 1975.
6. Shepard D., *A two-dimensional interpolation function for irregularly spaced data* Proc. of the 23rd National Conference ACM, ACM, New York, NY (1968), pp. 517-524
7. Vincenty, Thaddeus, *Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations*. Survey review 23.176 (1975): 88-93.
8. Hwang Y, Clark M, Rajagopalan B, et al. *Spatial interpolation schemes of daily precipitation for hydrologic modelling.* Stochastic Environmental Research and Risk Assessment. 2012, 26(2): 295-320.