IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 4, 2019, accepted October 6, 2019, date of publication October 21, 2019, date of current version November 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948757

Trustworthiness of Context-Aware Urban Pollution Data in Mobile Crowd Sensing

MARCO ZAPPATORE[1,3], (Member, IEEE), CORRADO LOGLISCI[2,4], ANTONELLA LONGO[1,3], (Member, IEEE), MARIO A. BOCHICCHIO[3,4], LUCIA VAIRA[1,4], AND DONATO MALERBA[2,4], (Member, IEEE)

[1]Hesplora srl, 73100 Lecce, Italy
[2]Department of Computer Science, University of Bari Aldo Moro, 70121 Bari, Italy
[3]Department of Engineering for Innovation, University of Salento, 73100 Lecce, Italy
[4]National Interuniversity Consortium for Informatics (CINI), 00185 Rome, Italy

Corresponding author: Marco Zappatore (marcosalvatore.zappatore@unisalento.it)

This work was supported in part by the research project "APOLLON - environmentAl POLLution aNalyzer", within the "Bando INNONETWORK 2017" funded by Regione Puglia (Italy) in the framework of the "FESR - Fondo Europeo di Sviluppo Regionale" - "POR Puglia FESR FSE 2014-2020 - Asse Prioritario 1 - Ricerca, sviluppo tecnologico, innovazione - Azione 1.6"

ABSTRACT Urban pollution is usually monitored via fixed stations that provide detailed and reliable information, thanks to equipment quality and effective measuring protocols, but these sampled data are gathered from very limited areas and through discontinuous monitoring campaigns. Currently, the spread of mobile devices has fostered the development of new approaches, like Mobile Crowd Sensing (MCS), increasing the chances of using smartphones as suitable sensors in the urban monitoring scenario, because it potentially contributes massive ubiquitous data at relatively low cost. However, MCS is useless (or even counter-productive), if contributed data are not trustworthy, due to wrong data-collection procedures by non-expert practitioners. Contextualizing monitored data with those coming from phone-embedded sensors and from time/space proximity can improve data trustworthiness. This work focuses on the development of an algorithm that exploits context awareness to improve the reliability of MCS collected data. It has been validated against some real use cases for noise pollution and promises to improve the trustworthiness of end users generated data.

INDEX TERMS Classification, data quality, data trustworthiness level prediction, machine learning, mobile crowd sensing, transductive learning algorithm, urban pollution monitoring.

I. INTRODUCTION

The pervasiveness of mobile technologies owned by the crowd discloses the opportunity to exploit them for civic and social purposes. Mobile Crowd Sensing (MCS) is a promising approach to observe real-world phenomena at a very large scale. The amounts of applications that have emerged in the recent years well show the added value offered by the MCS paradigm. From smart cities environmental assessment to scientific education, mobile devices allow non-expert practitioners as well as scientists to approach several scenarios without the hindrances of traditional data collection procedures (e.g., lack of skilled personnel, high costs, etc.). Amongst current MCS-based initiatives, the APOLLON Project is a research effort granted by Apulia Region (Italy) aimed at developing a platform for urban environmental (i.e. atmospheric, acoustic and UV) monitoring and analysis, based on the integration of heterogeneous data from several sources (e.g, citizens-owned personal devices, city-managed monitoring stations, etc). The project aims at: 1) integrating low-cost sensors scattered across the territory to create large observation areas; 2) engaging citizens in environmental monitoring campaigns; 3) involving city managers in proper management and exploitation of this data. Therefore, one of the specific requirements of the Apollon platform is to build a hybrid data layer able to integrate flows gathered from IoT sensors, mobile devices, open data, historical data, and social media feeds thanks to semantic technologies and

The associate editor coordinating the review of this manuscript and approving it for publication was Hongxiang Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

geo-localized data analysis utilities, which enable near-/real-time monitoring services for several city end-users. In this paper we will focus on a significant issue of MCS-based systems, which is the data quality trustworthiness.

The sensors embedded in smartphones contribute valuable quantitative observations about the urban environment (e.g., noise, temperature, atmospheric pressure, humidity, light, magnetism), which come along with the related spatial and temporal data. The potential of MCS seems enormous, but this is only one side of the coin. The major challenge facing effective crowdsensing is related to the quality of collected data, which depends on the accuracy of the contributing sensors and the adequacy of the sensing context. In fact a typical MCS system is hindered by several data quality issues: improper device usage, partial knowledge of the usage scenario, limited sensing capabilities of the device, time and space sparsity of contributed data, uneven distribution of sensors in a given geographical area, partial contributed data and so on. These elements usually lead to unbiased readings and outliers in the monitored scenario. Some of them can be tackled by properly training users and by calibrating in advance sensors, so that systematic errors in captured data can be coped with. Data classification approaches allow to not rely on user and sensor behaviour solely. For such reasons, in order to perform data quality assessment, a machine learning classification approach is proposed in this paper, so that mobile crowdsensed data can be categorized in terms of their data trustworthiness. In MCS-gathered datasets the proposed approach leverages on the contemporary presence of known readings (i.e., reliable sensor data for which everything or quite everything is known, *labelled data*) and of unknown (or partially unknown) readings (i.e., sensor data whose reliability is not known a priori, *unlabelled data*). Spatial and temporal auto-correlation of these two typologies of sensor readings can help to assess the reliability of unlabelled data that are spatially and temporally closed to labelled data. To that purpose, a transductive machine learning algorithm has been devised in order to train a classifier capable of inferring on the unlabelled data categories starting from the labelled ones. In order to estimate the achievable accuracy, three variants of the algorithm have been built, by exploiting three base learners to train the predictory (i.e., Decision Tree, Random Forest and Logistic Regression).

The validation has been performed in the Apollon platform just introduced above, with specific reference to noise levels collected via smartphone-embedded microphones. The paper is organized as follows: after the introductory Section I, the MCS paradigm is examined in Section II, in terms of analysis of its features, core elements of MCS systems and context awareness in MCS scenarios. Section III is devoted to present related works on MCS data quality and the motivations of our research work. The proposed algorithm is described in Section IV, where feature augmentation, prediction confidence and the classification process are illustrated. The application scenario, represented by project Apollon, is dealt with in Section V by analysing project architecture, as sensor readings typology, specificity and reference values as well as the test case. Results are discussed in Section VI, while conclusions are drawn in Section VII.

II. MCS PARADIGM APPLIED TO SMART CITIES

Mobile devices can be exploited to collect several kinds of data from multiple scenarios so that both individual users and groups of users can benefit from them. The advantages are manifold, and they are especially noticeable in urban contexts, thus contributing to the implementation of the Smart City paradigm. Indeed, mobile devices allow defining and provisioning innovative services to citizens and city managers so that the smartness of their community can improve considerably. By managing contextual information and by offering suitable and effective ways for interacting with user's social and physical situations, mobiles can be exploited in several different scenarios. Moreover, these devices represent a promising solution when people have to be engaged in collaborative or participatory activities, ranging from environmental sensing experiences to people monitoring in emergency conditions. Another common application scenario is data harvesting and information collection: citizens' mobile devices can be used to gather data from urban environments autonomously (i.e., without the direct intervention of the user) in order to manage and forward to policy makers, thus allowing city managers to be more aware of the potential issues affecting their municipalities, without additional relevant costs. Therefore if a large number of mobile devices can be used to collect sensor data, traditional monitoring campaign can be spared and addressed only where they are actually needed, so that the expensive deployment and maintenance of professional sensing equipment as well as personnel costs can be carefully controlled.

The features briefly sketched so far represent some of the core elements of the so-called Mobile Crowd Sensing (MCS) paradigm [1], according to which the capillary worldwide diffusion of smartphones and tablets can be leveraged by defining how sensor data collection should be performed directly by mobiles [2]. Mobiles equipped with proper applications capable of collecting sensor data can be scattered dynamically across large geographical areas so that they can acquire data from the surrounding environment opportunistically. Therefore, mobiles allow gathering location and time-based data so that, on the one hand, citizens can acquire greater knowledge on about their city and authorities can achieve better knowledge on people's perception of their city, thus tailoring civic policies more effectively to the real needs of the population. In addition, several MCS applications nowadays are intertwined with social networks, so that mobile users can contribute with both their social posts and the data sensed by their devices. The number of MCS-related projects is improving constantly and it can be broadly categorized according to monitoring domains.

- **Environment**: in this category, several projects have addressed so far data collection and monitoring for multiple parameters such as air pollutants [3], water

pollutants [4], noise levels [5], [6] and electromagnetic fields [7].

- **Transportation**: transportation systems, road [8], traffic and parking [9] can be controlled via MCS-based applications, where users can provide context information about road status, parking availability and so on.
- **First response**: critical situations such as first response to natural disasters [10] or emergency management [11] may benefit from the availability of mobile-provided sensor readings (provided that in such emergency scenarios wireless coverage is guaranteed).
- **Large-scale events**: sport competitions or music festivals, can exploit MCS-related solutions for monitoring crowds [12] in order to analyse crowd behaviour in real time or to promptly tackle unwanted events such as thefts, disappearance, etc.

Several elements have to be considered when designing a MCS applications [13]. First of all, contributed data must be collected and managed through a dedicated mobile app. This app has to sense, (possibly) pre-process locally and (optionally) send sensed data to remote/cloud-based servers [14]. Sensors data should be associated to location-based and geo-referenced data (i.e., situated data creation), without any significant time-related or space-related restriction (i.e., time continuity). Sensing data should be collected with high-spatial resolution and from a wide variety of scenarios, in order to make the data collection process representative of a real scenario. Then, users should be allowed to opt for a specific sensing typology by selecting between actively monitoring and contributing data (i.e., participatory sensing) and letting the app to collect data in background autonomously (i.e. opportunistic sensing). For supporting participatory sensing, a MCS-based application should exhibit training elements in order to improve user's awareness on the correct procedure to collect data and should be capable of motivating users and of involving them into active participation. Usually in the case where the MCS application attracts committed users, then the knowledge from the collected observations is not worth the spending. Participatory sensing allows enhancing the data quality but results in much less data than the opportunistic approach. Therefore we posit the need for developing intelligent systems to support opportunistic MCS. The intelligent layer that collects the sensing data on the device must act beyond merely interfacing with the embedded/connected sensors to transfer the data to the cloud. It must as far as possible enhance locally the quality of the observations, from calibration to contextualization. While calibration may be achieved through regression analysis [15], contextualization requires prediction. This paper is related how contextual data can infer the accuracy of collected data.

The next subsection will describe how context awareness has been referred to in scientific literature, while Section III will explain why, once collected, data have to be processed properly for assessing their accuracy level. Indeed, proper data manipulation and management procedures must be considered in order to increase data quality or assess data quality in a reliable way. MCS systems can benefit from contextual information in order to increase their effectiveness. It is widely-accepted to consider a *context* as any piece of information capable of characterizing the situation or the status of an entity [16]. Therefore, several sources of context can be identified in the area of MCS and the role of context awareness is pivotal to assess data quality, as deeply addressed in [17]. The authors of [17] introduces a three-level context representation, where the low-level context is the one provided by raw data collected through physical sensors, virtual sensors (i.e., software applications) or logical sensors (e.g., databases, logs), the high-level context is inferred through meta-data and the topmost-level context is the estimation of the user state that can be achieved by combining the subsumed levels. More specifically, the high-level inferred context can be subdivided into: 1) device context, related to the mobile technical parameters (e.g., connectivity, computational resources); 2) user context, related to user's profile and location; 3) physical context, provided by parameters such as temperature, noise level, etc.; 4) temporal context, related to the specific time frame during which the situation to be characterized is happening. Also in the area of MCS-based noise monitoring, that will be examined in the case study proposed in Section V, data quality can be affected by poor context awareness, as described in [18], where poor or misleading sensor calibration, location and metadata determine whether contributed readings are reliable or not.

III. RELATED WORK AND MOTIVATION

One of the critical issue in MCS is represented by data quality, sometimes referred to as data credibility or data trustworthiness, which is endangered because of multiple factors, such as the deployment scale, resource constraints, loss of connection [19] or even because the data sources are often barely controllable, unevenly skilled, and hardly accountable due to either their sensor or the managing personnel. In the literature, a large variety of approaches have been proposed and a consistent part focuses on a working life-cycle aiming at enhancing data collected and therefore improving the quality. The life-cycle encompasses operations to be performed both on the data sources side and on the data collection side as well, such as, interpolation, cleaning, de-duplication and integration [20].

However it is necessary to assess data quality, recognize low quality and distinguish unreliable data from the others [21]. The traditional solution relies on the application of ad-hoc defined or standard objective quality metrics, which however require the identification of the one more appropriate to the specific data scenario [22] and calculation of the metrics for all the data instances, often unfeasible and expensive especially in IoT architectures or Big Data environments [23].

One of the promising approaches is represented by the data-driven methods [24]. A research stream considers the use of general-purpose outlier detection methods for data produced in MCS, which has been argued that is is

ineffective [25] because of the difficulty to conciliate outlier-based notions with the quality metrics. The alternative is represented by classification models built on empirical MCS readings. In [26], the authors propose the use of the Hierarchical Temporal Memory based on neocortex learning to detect anomalous patterns in spatio-temporal data. Wu *et al.* [27] investigate classification and forecasting of the trustworthiness by exploiting collaborative filtering. Huang *et al.* [28] focuses on the device quality and report a classifier based on the Gompertz function to calculate device reputation scores as a reflection of the trustworthiness of data contributed by that device.

However, the classification of MCS data is not a trivial task for several reasons. First, low quality can raise several effects and can be manifested in different shapes, such as extreme events, erroneous recordings or anomalies [29]. They can even occur simultaneously and often follow the probability distribution of high quality data so as they can be assimilated to normal MCS readings. Second, being geo-localized and time-stamped, sensing data suffer the property of auto-correlation, which makes traditional machine learning poorly accurate due to the violation of the independence assumption (i.i.d.). To remedy these effects, auto-correlation should be explicitly accommodated in the classification models. Auto-correlation has a double facet since it involves both the spatial component and the temporal component of the sensed data. Spatial auto-correlation refers to the dependence among readings done by sensors close to each other and it is reflected in the similarity of the values [30]. Indeed, the larger the spatial closeness the higher the (positive) correlation. Temporal auto-correlation refers to the dependence between data readings done by the same sensor within a short time, so values recorded within a short time are more similar than those far away [31]. Third, classification models need user effort in recognizing reference sensing readings, which requires large collections of manually labelled data. This can be obtained through the intervention of authoritative annotators onto the sensing devices or implementation of collaborative labeling processes [19]. One alternative has been studied in the recent literature under the paradigm of *transductive learning* and focuses on the exploitation of unlabelled data in combination with few labelled data, which asks for much less human intervention [32]. In the transductive learning, the classification models are required to perform inferences as accurate as possible on the same set of unlabelled data, on which the models are learnt. So, it is not necessary they are general and applicable on any sensor, which is the same scenario we have in this paper, when recognizing the cases of low-quality data in a specific architecture of MCS.

To our best knowledge, this is one of the first studies of transductive learning on sensing data, or more generally spatio-temporal data. In the literature, we can find works which investigate the two dimensions separately. For instance, Bruzzone *et al.* [33] propose transductive Support Vector Machines (SVMs) for classifying spectral-spatial images. The algorithm is iterative and gradually searches the optimal discriminant hyper-plane in the feature space. In [34] the authors explore the use of graph-based representation for time-series classification in the transduction. They use Gaussian fields and harmonic functions in an iterative process where data labels' instances are repeatedly propagated through the neighboring data instances. The strength of label propagation is proportional to the strength of a connection considering all connections of a node or all connections of a pair of nodes.

In this paper, we propose to assess data quality of MCS sources through a classification approach able to distinguish MCS readings in categories of data trustworthiness. The approach leverages the spatial auto-correlation and temporal auto-correlation to label readings whose trustworthiness is unknown (unlabelled) by exploiting the labels of (known) readings which are spatial closed and temporally contiguous. We consider the data scenario of scarcely labeled readings and design a transductive learning algorithm to train the classifier both on labelled and unlabelled information, in order to perform accurate inferences on the categories of the unlabelled part.

IV. THE ALGORITHM

This section is devoted to the description of the algorithm we design to predict the level (label) of trustworthiness of the sensing data above described. We first provide basic notions and then explain how the algorithm works.

Let $\mathcal{D}$ be the set of sparsely labelled data which comprises the set $\mathcal{L}$ of labelled data and set of $\mathcal{U}$ of the instances with unknown trustworthiness ($\mathcal{D} = \mathcal{L} \cup \mathcal{U}$). The set $\mathcal{D}$ is spanned on a vector $\mathbf{X}$ of (numeric and discrete) attributes and a discrete attribute Y, which denotes the trustworthiness level. For the instances of $\mathcal{D}$ included into the set $\mathcal{L}$, the labels are known, while for the data of $\mathcal{U}$, the values of Y are determined by the algorithm.

Following the transductive paradigm, the algorithm inputs both the full information represented by $\mathcal{L}$ and the partially given information represented by $\mathcal{U}$, it learns a classification model and predicts the trustworthiness for the instances of the unlabelled part. This is done through an iterative convergence approach [35], [36] aiming at improving the accuracy of the classification model through a procedure that converges to a configuration of the predictions on $\mathcal{U}$ as accurate as possible. Before the iterations starts, the algorithm performs a feature augmentation step, which generates an extended set of descriptive properties for every instance and which has the final effect of making that instance "aware" about the distribution of the values (for each attribute) over the instances which are more correlated to it. These new attributes are updated during the iterative process, in order to "propagate" accurate predictions over correlated instances. To do that, it is necessary to take only the predictions that could truly improve the classification model, so we do consider only the predictions with high confidence, generated for the current iteration, and feed them the learning process of the next iteration.

The presence of the spatial autocorrelation and temporal autocorrelation implies the existence of a smoothing effect, for which nearby instances tend to share the same labels. Therefore, we should capture the local patterns of auto-correlation through the construction of ''neighborhoods'' of instances. This suggests us to define the new attributes so as they mirror the distribution of the values in the neighborhoods and determine the confidence of the predictions by considering the labels of the neighbors.

To capture the two-fold nature (spatial and temporal) of the sensing data, the notion of neighborhood should account for the presence of the spatial autocorrelation and of the temporal auto-correlation jointly, therefore the neighbours of an instance will be the data which are ''spatially'' and ''temporally'' close to that instance. Indeed, this allows us to nicely capture the typical scenario in which the data recorded by the same device in a short time tend to have the same trustworthiness, as well as, the data, spatially close to each other, which have been recorded in a short time tend to have the same trustworthiness level.

For every instance m, we build the spatio-temporal neighborhood $N(m, \delta_s, \delta_t)$ composed of the instances whose spatial distance from m does not exceed δ_s and which have been recorded within a time δ_t from m, formally:

$$\begin{aligned} N(m, \delta_s, \delta_t) = \{p | p \in \mathcal{D}, distance(m.c, p.c) \leq \delta_s, \\ m.t - p.t \leq \delta_t, if\ m.t \geq p.t, \\ p.t - m.t \leq \delta_t, otherwise\} \end{aligned} \quad (1)$$

The terms $m.c$ and $p.c$ denote the spatial coordinates of the instances m and p respectively, while the terms $m.t$ and $p.t$ denote the recording times.

A. FEATURE AUGMENTATION

The notion so defined of neighborhood is used to define new attributes which realize a feature augmentation step in the proposed algorithm. By taking recent studies into account which address the auto-correlation property through the computation of variation summarization statistics [37], we consider two classes of new attributes for the feature space $\mathbf{X} \times Y$, respectively, formulated as follows:

- **Class 1.** Given the base numeric attribute A, we build two new attributes, $AN(mean)$ and $AN(stDev)$, based on A. Both attributes are computed by aggregating A over the neighborhoods $N(m, \delta_s, \delta_t)$ constructed with maximum spatial distance δ_s and maximum temporal contiguity δ_t. Let m be an instance, $AN(u, mean)$ and $AN(u, stDev)$ are computed as the mean and standard deviation of the values of A falling in the neighborhood $N(m, \delta_s, \delta_t)$. Both the new attributes allow us to summarize average and variance, local to the neighborhoods, of the numeric attributes.
- **Class 2.** Given the base discrete attribute A that takes d distinct values, we build d new attributes. These attributes represent the frequency histogram of A, as it is computed on the neighborhoods $N(m, \delta_s, \delta_t)$. In practice, we build one attribute for every distinct value of A. Let m be an instance, val be a distinct value of A, $AN(m, val)$ is computed as the frequency of val over the neighborhood $N(m, \delta_s, \delta_t)$.

We remark that the Class 2 is used to build the new attributes associated to the attribute Y (trustworthiness levels). It should also be noted that the attributes of Class 2 associated to the attribute Y can change during the iterative procedure, because of the refinement process of the labels of the instances $\mathcal{U}$, while the attributes of Class 1 associated to the feature space $\mathbf{X}$ are valued once and remain unchanged.

B. PREDICTION CONFIDENCE

We measure the confidence of the labels predicted at each iteration, in order to select those more confident that are then fed back into the learning process for the next iteration. Intuitively, confident predictions should manifest the property of auto-correlation, so that similar labels can be plausibly propagated to the neighbours. The higher the autocorrelation of the label with neighbour labels, the more confident its prediction. To define the measure, we estimate the presence of the predicted label associated to the instance m over the neighborhood $N(m, \delta_s, \delta_t)$ by quantifying the times in which the prediction on m is identical to the labels of its neighbours included in the set $\mathcal{L}$ (labelled neighbours). The choice of comparing the predicted labels against those original of the set $\mathcal{L}$ is done to provide validity to the estimation of the confidence.

However, we should note that the temporal component could have a contribution larger than spatial component in the calculation of the confidence because onsets and effects of the urban processes often depend on the timing of human lifestyles and daily periods more than phenomena related to the spatial dislocation. For instance, onsets of the environmental pollution could be concentrated on the sunlight rather than moonlight, even when they are observed in the same geographic area. At the same way, we can register high concentration of noise pollution during daytime than nighttime. To encode this, when checking for the equality between the prediction of m and labels of the neighbours, we inject the temporal distances between the instance m and its neighbours into the confidence measure and assign weights to the distances dependently on their values. In practice, we build two sorted sets with the neighbours, one set with the instances recorded before the instance m and one set with the instances recorded after m. So, the weights are determined by the number of instances that separate m from the neighbours. For instance, given four neighbours p, q, r and s so sorted $p.t < q.t < m.t < r.t < s.t$, the weights of the comparisons between p and m, q and m, m and r, m and s will be 3, 2, 3, 2 respectively, resulting from the difference between the half cardinality of the neighborhood (that is, 5) and the number of separating instances. Whether the cardinality is odd, the half cardinality is rounded to the greater integer. Considering the weighting schema above illustrated, the confidence measure for the predicted label done on the

Algorithm 1 Transductive Classification Process $L, U, \delta_s, \delta_t) \mapsto \hat{U}$

Require: L: the labeled set spanned on $\mathbf{X} \times Y$; U: the unlabeled set spanned on $\mathbf{X}$
Require: δ_s: the maximum threshold for spatial closeness; δ_t: the maximum threshold for temporal contiguity
Ensure: $\hat{U}$: the set U labeled with the predicted labels $\hat{Y}$
1: $\mathbf{N_{st}} \leftarrow \mathbf{N_{st}} \cup$ buildNeighborhood$(L \cup U, \delta_s, \delta_t)$
2: $\mathbf{X}N \leftarrow$ performFeatureAugmentation$(L \cup U, \mathbf{X}, \mathbf{N_{st}})$
3: $\mathbf{Y}N \leftarrow$performFeatureAugmentation$(L \cup U, Y, \mathbf{N_{st}})$
4: $F \leftarrow$learnClassificationModel$(L, \mathbf{X} \times \mathbf{X}N \times Y \times \mathbf{Y}N)$
5: $\hat{U} \leftarrow$ labeling(U, F)
6: **repeat**
7: $\quad \mathbf{R_U} \leftarrow$ computeConfidence$(\hat{U}, \mathbf{N_{st}})$
8: $\quad B \leftarrow$ pickBetterRanked(sort$(\mathbf{R_U})$, $size_r$)
9: $\quad \hat{U} \leftarrow \hat{U} \cup B$
10: $\quad U \leftarrow U - B$
11: $\quad \mathbf{Y}N \leftarrow$updateAttributes$(L \cup U \cup \hat{U}, Y, \mathbf{N_{st}})$
12: $\quad F \leftarrow$learnClassificationModel$(L, \mathbf{X} \times \mathbf{X}N \times Y \times \mathbf{Y}N)$
13: $\quad \hat{U} \leftarrow$ labeling(U, F)
14: **until** $(U \neq \oslash \; OR \; \#it < MAX\_IT)$

instance m is so formulated:

$$\mathcal{R}(m) = \frac{\sum_{p \in \{N(m,\delta_s,\delta_t) \cap \mathcal{L}\}} (\sigma(m.t, p.t) \times (equal(\bar{y}, p.y)))}{\sum_{p \in \{N(m,\delta_s,\delta_t) \cap \mathcal{L}\}} (\sigma(m.t, p.t))}, \tag{2}$$

where $\sigma(m.t, p.t)$ determines the weight associated to every comparison (that is, temporal distance between m and p), $equal(\bar{y}, p.y))$ is 1 when the prediction $\bar{y}$ equals $p.y$, 0 otherwise. It has values in the range [0,1], where 1 denotes the highest number of occurrences of the label $\bar{y}$ in the neighborhood and therefore corresponds to the largest confidence, while 0 indicates the prediction is poorly confident.

C. TRANSDUCTIVE CLASSIFICATION PROCESS

A top-level description of the transductive classification process is reported in Algorithm 1), which performs learning and prediction along two stages, that is, initialization and iteration.

In the initialization stage (Algorithm 1, lines 1-5), it performs three main operations:

1) For every instance of $D = L \cup U$, it constructs the respective neighborhood with the instances which have spatial distance less than δ_s and temporal distance less than δ_t. This is done by considering the spatial coordinates and recording time of m, as illustrated in the formula 1 (Algorithm 1, line 1). The values of δ_s and δ_t are set by the user. Then, for each attribute X of the attribute vector $\mathbf{X}$, it generates new attributes of *Class 1* and *Class 2*, dependently on whether X is numeric or discrete. The computation considers both the labelled instances and unlabelled instances of each neighborhood $N_m \in \mathbf{N_{st}}$ previously determined (Algorithm 1, line 2). Finally, it generates new attributes of *Class 2* for the label-attribute Y with the procedure used for the attributes $\mathbf{X}$. In this case, the computation considers only the labelled instances of each neighborhood $N_m \in \mathbf{N_{st}}$ because the unlabelled instances have no prediction for the attribute Y at the initialization stage (Algorithm 1, line 3). Clearly, all the instances will have the same set of new attributes, while the values are specific per instance m and depend on the data distribution over the respective neighbors N_m.
2) The algorithm learns a classification model F from the training set L, which is now represented with an augmented feature space $\mathbf{X} \times \mathbf{X}N \times Y \times \mathbf{Y}N$ (Algorithm 1, line 4). This allows us inject the auto-correlation into the learning process since the beginning, without making the subsequent computation burden because the new attributes are built once only.
3) The model F is finally used to initialize the unknown labels of the instances U (Algorithm 1, line 5), which are stored as $\hat{U}$. This way, the predictor F is able to estimate the data trustworthiness by considering additionally the contextual information provided by the nearby sensors (spatial auto-correlation) and by the readings done by the same sensors in the past (temporal auto-correlation), besides of information the sensors record in themselves.

In the iteration stage (Algorithm 1, lines 6-14), we aim at improving the predictive accuracy of F and, to this end, we exploit the auto-correlation property from the most confident predictions inferred along the iterations. Basically, the algorithm carries out the following operations:

1) For every instance m previously labelled and stored in the set $\hat{U}$, it computes the confidence values by comparing the prediction of m against the originally known labels of the instances of L included in the neighborhood N_m, as illustrated in the formula 2 (Algorithm 1, line 7).
2) The confidence values $\mathbf{R_U}$ are sorted and then we pick only the first $size_r$ instances with higher rank, being considered as mostly reputable. The assigned labels (stored in B) will be maintained as such because they will contribute to the subsequent operations, since the instances are now "stabilized". However, these instances will have a role different from the originally labelled instances L, in accordance with the philosophy of the transductive learning and, in fact, they are removed from the target set (unlabelled instances) U and moved in the set $\hat{U}$, which is different from L (Algorithm 1, lines 8-10).
3) The new configuration of labels, caused by the reduction of U and extension of $\hat{U}$, is propagated over all the instances (L,U,$\hat{U}$) through the update of the new attributes. It should be noted that only the attributes $\mathbf{Y}N$ are influenced by the update, since those of the

set $\mathbf{X}N$ remain unchanged, being derived by the attributes $\mathbf{X}$ (Algorithm 1, line 11).

4) In accordance with the transductive learning, the classification model F is (re-)trained on the originally labelled instances L, which are now "aware" about the new labeling scenario (Algorithm 1, line 12). So, the predictor F can leverage the *i)* confidence of the predictions ($\hat{U}$) and *ii)* reinforced configuration of the descriptive attributes, in order to improve the accuracy of the instances left in U (Algorithm 1, line 13).

This iterative procedure stops when one of the two stopping criteria is satisfied, specifically, either the set U is empty or the number of iterations completed reaches an user-defined threshold MAX_IT. However, the depletion of U is guaranteed to happen why every iteration removes a portion of instances equal to $size_r$ (which is defined by the user).

V. APPLICATION TO THE APOLLON PROJECT

This section is devoted to the presentation of the Apollon project as regards to the application scenario, featuring aspects, platform architecture and addressed test cases.

A. PROJECT DESCRIPTION AND SPECIFICITY

The Apollon project, as already anticipated, aims at developing an efficient monitoring system exploiting heterogeneous environmental sensors. Collected data are processed, aggregated and validated in near-time before making them available to final users via proper visual dashboards hosted by the platform frontend.

One of the featuring core aspects of the Apollon project is the coexistence of several types of sensors, scattered across the geographical area under observation. These devices may range from smartphones (which collect measurements thanks to either their embedded sensors or external pluggable sensors) to low-cost mobile sensing stations (e.g., Arduino-based sensing boards hosting several sensors). According to a more general perspective, involved devices can be categorized into:

- **Fixed stations**: this group encompasses low-cost metering equipment deployed by city administrators when/where needed and under their maintenance and control. These stations can be deployed in a fixed location for long periods or, alternatively, can be deployed on vehicles provided by city authorities and city responders (e.g., metropolitan police cars, traffic police cars, etc.) so that their sensing devices can be moved around the city without additional costs.
- **Mobile Crowd Sensing**: this category refers to data sources whose behaviour complies with MCS requirements. Therefore, any citizen owning a mobile device can participate to monitoring campaigns and contributing her/his measurement data to the platform. Such a category exploits self-scalability and dynamic infrastructures of edge/cloud computing.

It is plausible to estimate limited number of available devices belonging to the first category if compared to the mobile devices used in the framework of MCS activities. That is the reason why the MCS paradigm is commonly suggested as a way to improve traditional sensor network functionalities and deployment strategies in terms of dynamicity, automatic scalability and low-costs.

B. PLATFORM ARCHITECTURE AND MCS-BASED MOBILE APP

The Apollon project platform encompasses several components. The platform architecture in the large is represented in Figure 1. Starting from the bottom, the following layers have been defined:

- **Edge and IoT Layer**: it collects heterogeneous devices adopted as data sources. The project manages stationary sensors (e.g., fixed monitoring stations provided by authorities and/or environmental protection agencies for pollution control) as well as mobile sensors (e.g., monitoring stations placed on top of vehicles routing across urban areas for mobile pollution control). A further type of mobile sensors is represented by personal electronic devices such as smartphones and tablets, operating according to the MCS principles and collecting data via their embedded sensors or via external sensors (either plugged or wirelessly connected to the smartphones). Additional data sources such as open data and institutional data repositories as well as social media streams belong to this layer, too. In the framework of the Apollon project, several physical parameters are monitored: noise levels, particulate matter (i.e., PM10, PM2.5, PM1), volatile organic compounds (VOCs), UV-A/B rays.
- **Hybrid Data Collection and Processing Layer**: this layer is devoted to data storage, management, filtering and integration operations performed on the data received from the sources just enlisted above. This layer also performs geo-referencing and time-stamping processes in order to reference measurement data in time and in space properly.
- **Business and services Layer**: this layer is in charge of performing complex operations on data coming from the data layer in order to feed services hosted in the Service sublayer. Business-to-Business (B2B) and Business-to-Consumer (B2C) services are exposed. Several specific modules are present in this layer, such as advanced geo-referencing, sentiment analysis, user management, semantic analysis and integration, open data creation.
- **Presentation Layer**: it represents the multi-faceted interface used by project stakeholders to access the platform. The layer offer visual dashboards thanks to a dedicated Web portal, a mobile app and a Telegram BOT. The contents accessible via the presentation layer vary depending on the specific stakeholder type that request them.

As depicted on top of Figure 1, several end-users types have been envisioned for this project. They range from

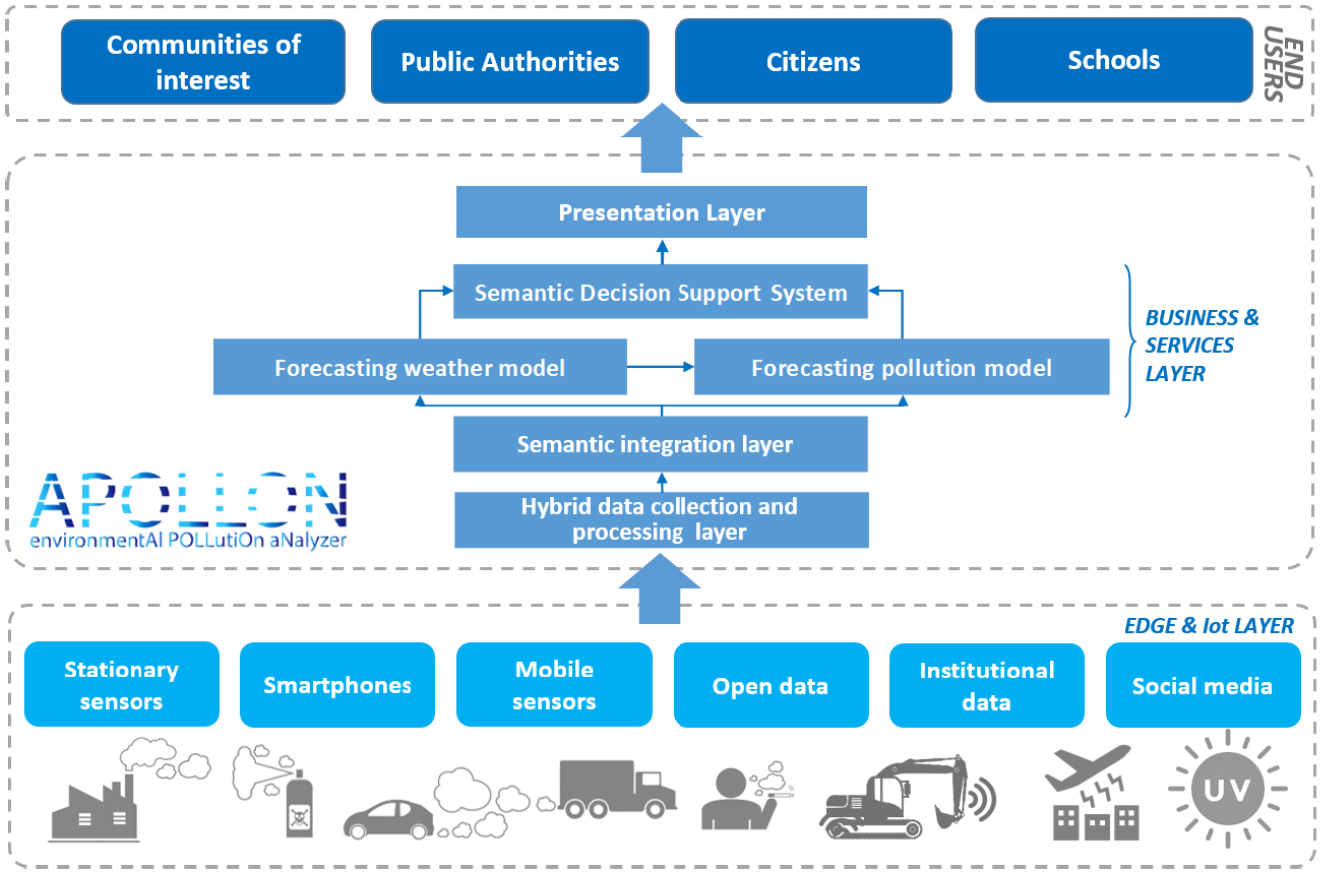

FIGURE 1. Platform architecture in the large for the Apollon project.

communities of interest (e.g., citizens groups and associations interested in performing pollution measurement, factories and industries interested about their level of environmental pollution) to public authorities, from citizens (both single individuals and associations) to other end user categories such as schools or healthcare providers.

By starting from such premises, it is clear that the entire project architecture as well as its purposes are complex and variegated. Therefore, in order to focus on the specific aspects dealt with this paper, it is now worth to point out the components that will be considered in this research. We will concentrate our analysis on MCS data sources (i.e., smartphones) by applying the processing and classification algorithm described in Section IV in order to determine the level of reliability of such data.

Therefore, the first aspect to be considered is the way users can contribute to monitoring campaigns, according to the MCS principles and to the citizen science aims described in Section II. To address those requirements, a mobile app has been designed and developed. It allows users to participate to measurement campaigns by providing sensor data related to noise levels and particulate matter levels. Noise levels can be sensed via internal or external, pluggable microphones. Particulate matter levels can be sensed by using proper external devices for indoor-outdoor air quality control. Only sensor readings coming from smartphone microphones will be considered for the data quality assessment in this research work. The trustworthiness analysis of air monitoring data will be addressed in a forthcoming research study.

Two screenshots from the mobile app (Android version) are depicted in Figure 2. The first one (on the left) is the opening app screen: from this panel, the user can start to monitor noise levels or particulate matter levels. Since we are focusing on noise only, the second option will be discarded in this description. Once the noise monitoring functionality has been selected, the user can decide whether starting an automatic measurement session (i.e., the mobile app works in background and sends measurement values each 60 seconds) or a manual session (i.e., the user selects in advance the duration of the measurement session). These two working modes are representative of the two broad categories in MCS: the opportunistic sensing (i.e., users are not directly involved in collecting data but they offer sensing devices and platforms only) and the participatory sensing (i.e., users are engaged in data collection activities more actively, thanks to informative materials, feedback from the devices/apps etc.) respectively. The second screenshot shows how ongoing noise measurements are shown to the user. The app monitors the sound

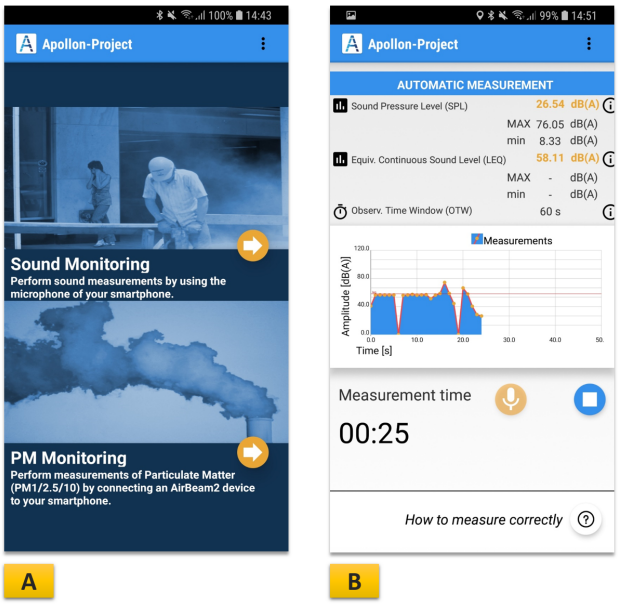

FIGURE 2. **Screenshots from the project Apollon mobile app: opening screen (A) and noise measurement dashboard (B).**

pressure level (SPL, an instantaneous quantity) and the equivalent continuous sound level (L_{EQ}, which is the time-averaged SPL on a given time window). Both these quantities are shown in the numerical dashboard on the upper part of the screen and also charted in the area graph below. Three interaction buttons (i.e., *start measurement*, *stop measurement* and *show measurement suggestions*, depicted with a microphone icon, a square icon and a question mark icon, respectively) complete the interface.

It is important to point out that, while the user, especially when involved in a participatory sensing activity, can receive suggestions from the app itself on how to perform measurement correctly, thus directly intervening on the measurement quality, the same does not apply to the opportunistic sensing which, conversely, can produce large amount of data from untrusted/unrealiable source. Indeed, as explained in Section IV, once sensor readings are collected, additional systems of assessing their trustworthiness are needed. The next sub-section is devoted to describe managed data.

C. SENSOR READINGS FROM MCS SOURCES

Only sensor data related to noise level control have been considered in this study. These data originate from the Apollon mobile app and they are addressed to the data processing components hosted in the Data Layer. The sensing device adopted to perform the noise metering sessions is the smartphone built-in microphone. In addition, in order to improve measurement quality, an external pluggable microphone can be used, provided that it is properly connected to the smartphone where the mobile app of the project is installed and running.

Independently from the specific microphone type in use, the mobile app collects several *contextual parameter* in order to improve the elements supporting measurement quality analysis. These additional parameters are readings that come from other internal sensors of the same smartphone used for measuring noise levels and they contribute to create the so-called *context awareness of the device* referred to the scenario where noise levels are being monitored. Each time a noise measurement is sent from the app to the data layer of the project platform, it is enriched by the contextual parameters. It is also clear that the more additional sensors are available, the richer the achievable contextual dataset is and, consequently, the more reliable the trustworthiness analysis of the mobile-generated data can be. As already asserted in Section II, context awareness has become a core element for determining sensor readings quality [17], [18]. Before examining the proposed use case and how noise levels and contextual parameters are fed into and processed by the classification algorithm, let us examine in more details the data typologies managed by the Apollon project mobile app. The app collects and sends the quantities specified in the following list.

1) NOISE LEVELS

Noise levels are measured in dB(A), they oscillate within the measuring range of the smartphone microphone. Microphone precision, resolution and accuracy affect noise levels value. A common smartphone-embedded microphone exhibits a directional response (i.e., the sensitiviy is higher when incoming sounds are picked up from the front of the microphone) and can detect incoming sounds in the range [+20;+110] dB(A). When an external, pluggable microphone is used, it has an omnidirectional response (i.e., equal sensitivity in all directions) and a sensing range between [+20;+110] dB(A). Moreover, smartphone microphones do not have any shielding against wind and, therefore, their readings are impacted differently depending on whether they are used outdoor or indoor. Table 1 reports typical noise values for different scenarios. The features of smartphone microphones just enlisted above demonstrate how noise level measurements introduce a bias in comparison to sensor readings provided by professional (or semi-professional) sound level meters. Therefore, the very first issue to be addressed, even before considering data trustworthiness assessment via pre-processing algorithms once sensor readings have been collected, is to determine the adopted sensors' accuracy. We have already examined this aspect in a previous work [38], by comparing sensor readings from different smartphone microphones against a professional, class-I sound level meter. Sensor readings were provided by a previous version of the mobile app described in this research work while the sound source was a digitally-generated, steady, mid-level and broadband signal at a fixed frequency and with a fixed waveform. Repeated comparisons were performed at different locations, representative of different urban scenarios. The overall average accuracy we achieved was ±5 dB(A), thus confirming similar outcomes from other research work [39]. Moreover

TABLE 1. Noise levels for typical scenarios.

Noise level [$dB(A)$]	Outdoor scenario	Indoor scenario
$0 < x < 10$	Outside microphone sensing range	Outside microphone sensing range
$10 \leq x < 20$	Absence of noticeable sounds	Absence of noticeable sounds
$20 \leq x < 30$	Slightly noticeable sounds	Slightly noticeable sounds (e.g., bedroom, library, etc.)
$30 \leq x < 40$	Rural setting with typical night sounds	Room with low voices
$40 \leq x < 50$	Urban setting, not noisy	Room with normal voices
$50 \leq x < 60$	Commercial area	Office room with normal voices; machineries running in adjacent rooms
$60 \leq x < 70$	Urban setting, noisy	Machineries running within 5m to 10m; loud voices
$70 \leq x < 80$	Urban setting, very noisy	Machineries running at no more than 5m
$80 \leq x < 90$	Vehicle engines running at close distance	Noisy machineries running at no more than 5m; yellings
$90 \leq x < 100$	Industrial machineries running at close distance	Very noisy machineries running at no more than 5m; yellings
$100 \leq x < 110$	Noisy industrial machineries running at close distance	Very noisy events (e.g., rock concert)
$110 \leq x < 120$	Airplane landing or takeoff, explosions, gunshots	Explosions, gunshots
$x \geq 120$	Outside microphone sensing range	Outside microphone sensing range

we have already cited that regression algorithms [15] can even improve this aspects

2) CONTEXT: LUMINOSITY

Luminosity values are measured in lux, this parameter is provided by the luminosity sensor placed in the top-front side of the smartphone. It is used to acquire additional knowledge on the light conditions in the environment where the smartphone is placed. A common smartphone-embedded luminosity sensor has a range of [+0.01;+5000] lux. Such a piece of information can be used to infer whether the smartphone is in daylight or in different levels of darkness. However, readings from this sensor cannot be used to assess univocally where the device is. For instance, very low luminosity levels can be collected in different scenarios: the device is within a closed bag; the device is outdoor, during the night, without any external source of light, the device is indoor, in a closed room without any light. The same applies with high luminosity values, which can be determined by sunlight, artificial lights and so on. Luminosity values change depending on whether the user is indoor or outdoor. Table 2 reports typical luminosity values for different scenarios.

3) CONTEXT: PROXIMITY

The proximity parameter is provided by the proximity sensor of the smartphone, which is placed in the top-front side of the device. Its values represent the distance between the sensor and any obstacle placed in front of it. Readings from such sensor are usually provided in cm ranging from 0 to greater than 10. Such a contextual parameter is useful in order to determine whether any obstacle is close to the device. Since the microphone and the proximity sensor are both placed in the front side of the smartphone, it is plausible to infer that when the proximity sensor values are low, an object is close to the device and therefore its presence can also affect how the microphone collects sounds incoming from the surroundings. Moreover, proximity values can be correlated properly to the luminosity values: for instance, a low proximity value and a low luminosity value plausibly indicate that an obstacle is close to the device and few considerations can be done on the surroundings, while a high proximity value and a low luminosity value plausibly indicate that the device is unobstructed and placed in a dark environment. Table 3 reports typical proximity values for different scenarios.

TABLE 2. Luminosity values for typical scenarios.

Luminosity value [lux]	Outdoor scenario	Indoor scenario
$0 < x < 0.1$	Complete darkness	Complete darkness
$0.1 \leq x < 10$	Attenuated light, partial darkness	Attenuated light, partial darkness
$10 \leq x < 50$	Sunset	Very scarcely illuminated room
$50 \leq x < 100$	Day, very scarce illumination	Scarcely illuminated room
$100 \leq x < 200$	Day, scarce illumination (e.g., cloudy)	Room with soft illumination
$200 \leq x < 500$	Day	Room with adequate illumination
$500 \leq x < 1000$	Full day light	Room with good illumination
$1000 \leq x < 10000$	Direct day light	Room with professional illumination
$x \geq 10000$	Maximum illumination	Maximum illumination

TABLE 3. Proximity values for typical scenarios.

Proximity value [cm]	Outdoor scenario	Indoor scenario
$0 \leq x < 5$	Obstructed sensor	Obstructed sensor
$x \geq 5$	Non obstructed sensor	Non obstructed sensor

4) CONTEXT: DETECTED ACTIVITY

This contextual parameter is provided directly from the Android operative system. By leveraging on the readings from the device 3-axis accelerometer and gyroscope, the Android operative system can detect the user's activity patterns with an associated confidence level. Detectable activities are of the following types: *device in a vehicle*, *device on a user who is walking*, *device in still position*, etc. The activity recognition feature is offered amongst the publicly available Android APIs (since release 21) in order to develop apps that can detect when a user starts or stops a specific activity (e.g., an app that must determine whether the user is driving in order to disable push notifications or not).

TABLE 4. Detected activity codes according to Android OS APIs.

Detected activity code	Description
IN_VEHICLE	The device is in a vehicle, such as a car.
ON_BICYCLE	The device is on a bicycle.
ON_FOOT	The device is on a user who is walking or running.
RUNNING	The device is on a user who is running.
STILL	The device is still (not moving).
TILTING	The device angle relative to gravity changed significantly.
UNKNOWN	Unable to detect the current activity.
WALKING	The device is on a user who is walking.

In our case, activity recognition can be performed so that activities not compatible with proper noise measurement procedures are identified (e.g., noise measurements performed in a still position are preferable than measurements performed by a user who is running). The available activity codes are enlisted in Table 4.

5) CONTEXT: LOCATION AND TIMESTAMP

On the one hand, the device location can be used to enrich context awareness if correlated to time stamping. On the other hand, measurement timestamps can be used to enrich context awareness if correlated to device locations. Timestamps are provided by the mobile app as the date and time (plus timezone) reported by the device at the moment of noise level recordings. Locations are provided by the mobile app in terms of latitude and longitude coordinates.

6) CONTEXT: CORRELATION OF LOCATION AND TIMESTAMP

If location s and timestamp t have their own importance as context parameters, their proper correlation is even more important. Indeed, as anticipated in Section IV, sensor readings exhibit an intrinsic two-fold nature, as they are closely related in space and time. Therefore, it is highly likely that noise levels acquired in a short time lapse by the same device show the same reliability (or unreliability). Similarly, it is plausible to attribute the same trustworthiness to noise levels originated in a short time lapse by different devices sharing a close spatial proximity.

D. REFERENCE CASES

The availability of several contextual parameters can improve the trustworthiness assessment effectiveness as for data and sensor readings coming from MCS-based applications. However, in order to achieve such an outcome, proper considerations on how to combine contextual parameters in a meaningful way are needed. To this aim, we have examined several reliability and unreliability scenarios where the parameters described in the previous section have been considered. These sets of reference values help to train the model properly.

Let us consider a noise level measurement $m_{i,s,t}$ originated by a mobile device located in a location s at the timestamp t. Now let us consider a set of reference cases where the reliability of the provided measurement is defined in terms of the contextual parameters (i.e., luminosity l, proximity p, detected activity da, timestamp t, location s). The reliability assessment spans across the following values: *reliable*, *poorly reliable*, *not reliable*. Reference test cases are enlisted in Table 5 and are now discussed in details.

The simplest reference case entails measurements exceeding sensing boundaries of the adopted sensing device: these values can be considered not reliable as the sensing device cannot provide such readings. Therefore, since our sensing device is a smartphone-embedded (or external) microphone, we mark as unreliable measurements below 20 dB(A) or beyond 120 dB(A) independently from the values of contextual parameters.

If a measurement is within the range]+20;+50] dB(A), it is reliable if luminosity goes below 200 lux and proximity is beyond 5 cm, independently from the other parameters. This scenario corresponds to a smartphone that is measuring low noise levels in a scarcely lit environment without any obstacle in its proximity (e.g., the device could be located in a poorly lit room or outdoor at nightime). Measurements from the same noise levels range are poorly reliable if the proximity sensor shows a value under the 5-cm threshold, independently from the other parameters, as this reading indicates an object close to the device and, consequently, a potential obstruction to the incoming sounds. We also marked it as poorly reliable those measurements, in the same noise level range, for which the detected activity is different from *STILL* and the proximity is beyond 5 cm as they are representative of a user moving around with its smartphone: since smartphone microphones do not have wind shielding, it is unlikely that user's movements (either by vehicles or on foot) do not raise noise levels over the 50 dB(A) threshold. Instead, a measurement in this noise level range is unreliable when it shows luminosity below 100 lux (i.e., significant darkness), proximity over 5 cm (i.e., no obstacles in front of the device), a timestamp belonging to the daytime range (i.e., from 6 a.m. to 10 p.m.) and a location not compatible with such a timestamp. This scenario represents a dark setting with no obstruction or environmental condition apparently determining it and, therefore, should not be considered as reliable. In order to clarify further this assumption, let us refer to the *timestamp vs location compatibility*, which can be explained as the combination of latitude and longitude values for which at a given time and date it is plausible to have specific illumination conditions (e.g., at the same time, illumination conditions vary depending on the latitude where the observer is located).

Let us now consider measurements coming from the interval]+50;+90] dB(A). They are reliable when the associated luminosity is above 10 lux and proximity is above 5 cm (i.e., no close obstacles), without any other restriction on detected activity, timestamp and location. Such measurements are poorly reliable when proximity is below 5 cm as an obstacle may obstruct the smartphone microphone. Similarly, poor reliability is present in dark settings (i.e., $l < 50$ lux), during daytime (i.e., between 6 a.m. and 10 p.m.) and at locations not compatibile with such timestamp for the same reasons

TABLE 5. Reference cases for evaluating reliability of noise level measurements depending on contextual parameters luminosity *l*, proximity *p*, detected activity *da*, timestamp *t* and location *s*. Each noise level measurement is sent by a mobile device with corresponding contextual parameters attached.

Measurement [$dB(A)$]	Parameters					Reliability
	Luminosity l [lux]	Proximity p [cm]	Detected activity da	Timestamp t	Location s	
$m_{i,s,t} < 20$	$\forall l$	$\forall p$	$\forall da$	$\forall t$	$\forall s$	Not reliable
$m_{i,s,t} \geq 120$	$\forall l$	$\forall p$	$\forall da$	$\forall t$	$\forall s$	Not reliable
$20 < m_{i,s,t} \leq 50$	$l < 200$	$p > 5$	$\forall da$	$\forall t$	$\forall s$	Reliable
	$\forall l$	$p < 5$	$\forall da$	$\forall t$	$\forall s$	Poorly reliable
	$\forall l$	$p > 5$	$\forall da \neq STILL$	$\forall t$	$\forall s$	Poorly reliable
	$l < 100$	$p > 5$	$\forall da$	$6am \leq t \leq 10pm$	$s \not\Rightarrow\!\!\!\Leftarrow t$	Not reliable
$50 < m_{i,s,t} \leq 90$	$l > 10$	$p > 5$	$\forall da$	$\forall t$	$\forall s$	Reliable
	$l < 50$	$p > 5$	$\forall da$	$6am \leq t \leq 10pm$	$s \not\Rightarrow\!\!\!\Leftarrow t$	Poorly reliable
	$\forall l$	$p < 5$	$\forall da$	$\forall t$	$\forall s$	Poorly reliable
$90 < m_{i,s,t} \leq 120$	$l > 10$	$p > 5$	$\forall da$	$\forall t$	$\forall s$	Reliable
	$l < 10$	$p > 5$	$\forall da$	$\forall t$	$\forall s$	Poorly reliable
	$\forall l$	$p < 5$	$\forall da$	$\forall t$	$\forall s$	Not reliable

The notation $s \not\Rightarrow\!\!\!\Leftarrow t$ indicates an incompatibility between a location s and a timestamp t (e.g., a timestamp t that belongs to a daytime range and a location l that belongs to a region where it is night actually).

explained when measurements in the range]+20;+50] dB(A) were considered.

Finally, if noise levels are within the]+90;+120] dB(A) range (i.e, very high noise levels), they are reliable if no obstacles are in front of the microphone (i.e., $p > 5$ cm) and the setting is not completely dark (i.e., $l > 10$ lux). If darkness increases (i.e., $l < 10$ lux), these measurements are less reliable as it is unlikely to have very high noise levels in urban settings in pitch black. Further context parameters should be needed in order to clarify the reason of so low luminosity values but, since the majority of commercial smartphone cannot provide such additional elements, we mark this scenario as poorly reliable. When the microphone is obstructed (i.e., $p < 5$ cm), these measurements are considered as not reliable, as it is unlikely that a covered microphone can provide readings close to the upper bound of its sensing range.

E. PROBLEM SIZE AND TEST CASES

In order to provide empirical evidence to the proposed solution for trustworthiness assessment, we performed experiments on a dataset of MCS data collected by the architecture illustrated in Figure 1. The dataset includes noise pollution readings recorded by a set of five smartphone devices, moving aroud the geographic area of Lecce (Apulia Region, Southern Italy), during the period 2019/05/27 – 2019/07/01. The considered geographic area has a surface extension of nearly 12 squared kilometers and measurements have been performed by volunteers scattered across this area. As for the spatial distribution of gathered readings, data captors were requested to move all around the city and to station, if possible, in the proximity of *noise hotspots*, such as congested road junctions, construction sites, and so on.

Specifically, we have 4335 readings uniformly distributed over the categories (991 instances for *not reliable*, 1782 instances for *poorly reliable* and 1562 for *reliable*), so we have no imbalanced concern for the classification task. The reliable-to-overall instance rate confirms that MCS scenarios require significant amounts of collected data in order to achieve meaningful insights. Moreover, it is worth to remark that, at this stage, we considered data collected via mobile crowd-sensing campaigns only. The APOLLON project, however, is aimed at addressing wider monitoring scenarios, where fixed monitoring stations are available as well. This specific kind of sensor source will be considered in an upcoming version of the proposed algorithm.

The dataset of 4335 readings has been used to arrange experiments aiming at training and testing the classification models. In particular, we performed a quantitative evaluation on the predictive capabilities of the transductive algorithm in performing accurate inferences on the trustworthiness level of the unknown noise pollution readings. The accuracy was measured in terms of the F-score and averaged over 5 trials executed according to the inverse 5-fold cross validation. More precisely, for each trial, the algorithm is trained on one fold (which represents L) and tested on the set U composed of the remaining four folds. We guaranteed that the training set L was balanced. By following the transductive setting, the set L contains a smaller part of the whole dataset, and, more precisely, it has a balanced percentage of 10%. The accuracy was estimated on three variants of the algorithm, which were built by using three base learners to train the predictor F. Specifically, we integrated the classification algorithms of *Decision Tree* (DT), *Random Forest* (RF) and *Logistic Regression* (RF) available in the framework Apache Spark -MLlib [40]. They were used in the default setup suggested by the framework.

VI. RESULTS AND DISCUSSION

The three variants of the algorithm were tested along two main experimental setups, in order to study their influence on the predictive accuracy, these are *i)* size of the set of confident predictions ($size_r$), *ii)* size of the neighborhoods in terms of the values of the thresholds δ_s and δ_t. To this end, we considered three different values of $size_r$, that is, 5%, 10%, 15% and three different neighborhood configurations, that is, $\delta_s = 250$ meters and $\delta_t = 10$ minutes (thereafter, n_10_250), $\delta_s = 500$ meters and $\delta_t = 5$ minutes (thereafter, n_5_500), and $\delta_s = 500$ meters and $\delta_t = 10$ minutes (thereafter, n_10_500), which let us build neighborhoods

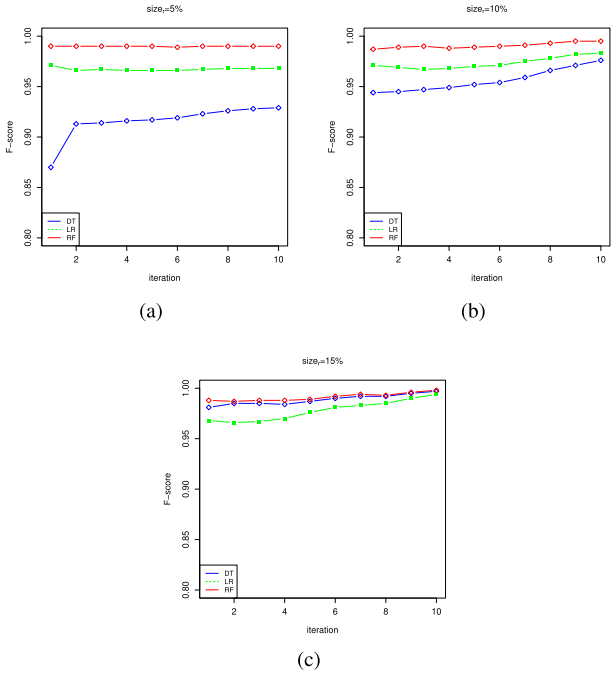

FIGURE 3. The F-score values computed on the unlabelled st along the iterations when $\delta_S = 250$ meters, $\delta_S = 10$ minutes. The results include the three variants designed with three different base learners respectively, Decision Tree, Logistic Regression, Random Forest.

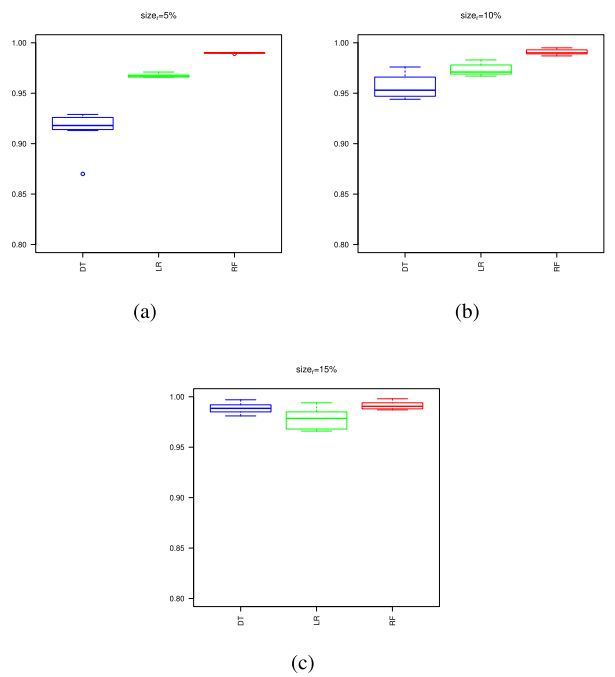

FIGURE 4. The average values and standard deviation values computed on the F-score values obtained with $\delta_S = 250$ meters, $\delta_S = 10$ minutes.

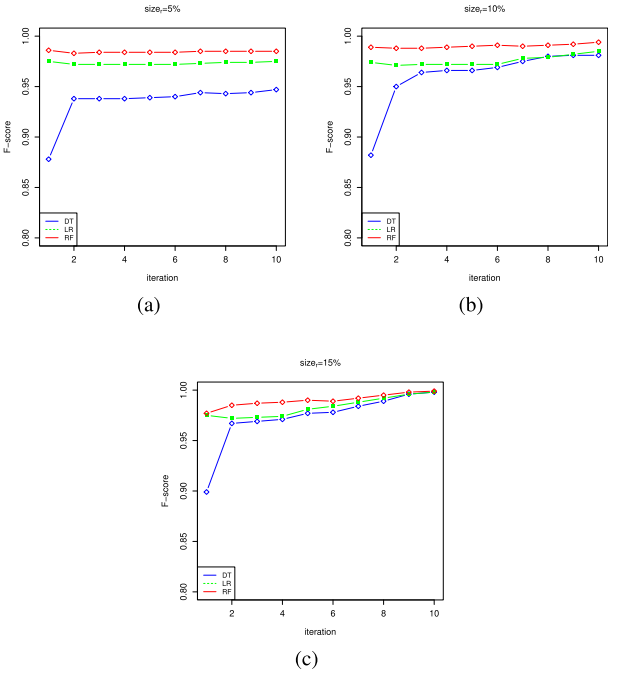

FIGURE 5. The F-score values computed on the unlabelled st along the iterations when $\delta_S = 500$ meters, $\delta_S = 5$ minutes. The results include the three variants designed with three different base learners respectively, Decision Tree, Logistic Regression, Random Forest.

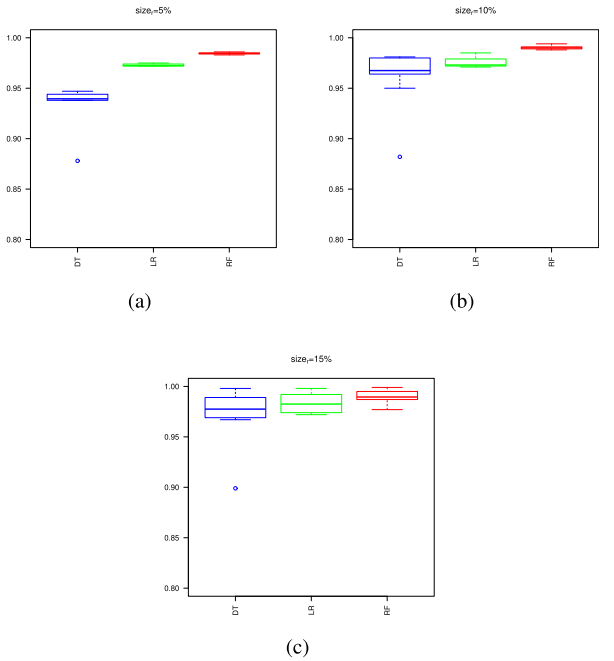

FIGURE 6. The average values and standard deviation values computed on the F-score values obtained with $\delta_S = 500$ meters, $\delta_S = 5$ minutes.

with different sizes (number of instances contained), that is, 4, 5 and 7, on average, respectively. For a fair comparison, we set *MAX_IT* to 10, which allows the experimental trials terminate under different conditions.

The F-score values computed along the iterations with n_10_250 are reported in the Figures 3, while those computed with n_5_500 are reported in the Figures 5, finally, the F-score values computed with n_10_500 are reported in Figures 7. In the Figures 4, 6, 8, there are the average

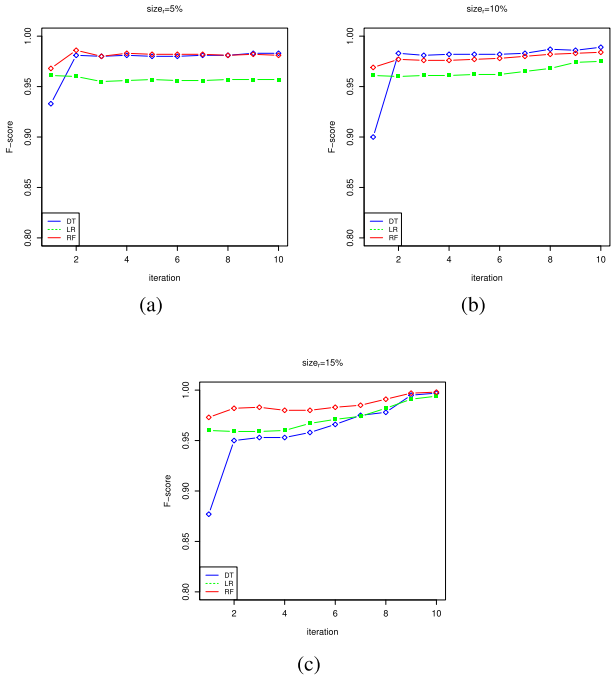

FIGURE 7. The F-score values computed on the unlabelled st along the iterations when $\delta_S = 250$ meters, $\delta_S = 10$ minutes. The results include the three variants designed with three different base learners respectively, Decision Tree, Logistic Regression, Random Forest.

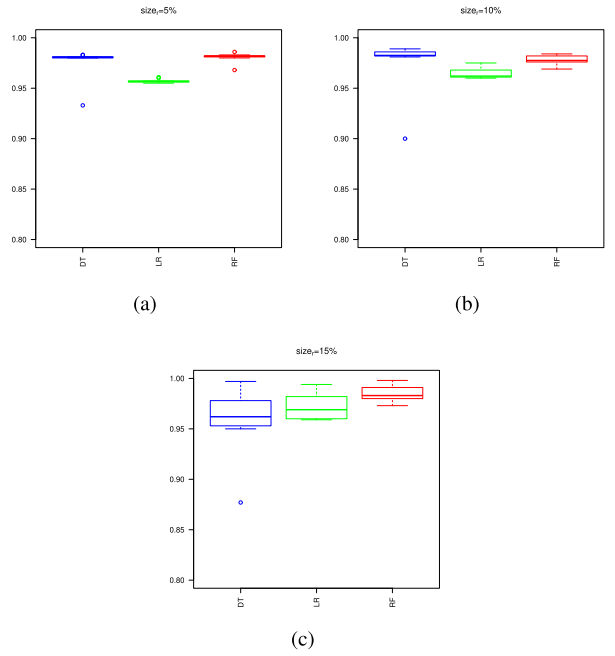

FIGURE 8. The average values and standard deviation values computed on the F-score values obtained with $\delta_S = 500$ meters, $\delta_S = 10$ minutes.

values and standard deviation values computed on the F-score values.

The first consideration we can do is on the number of the iterations. Regardless of the neighborhood configuration, the predictive accuracy increases as new iterations are performed. The highest gain accuracy is obtained at the initial iterations, which indicates the algorithm benefits from the best confident predictions since at the early. This confirms the effectiveness of the iterative learning approach. We should also note that acceptable F-score values can be reached even before the execution of 10 iterations. Clearly, this leads benefits from the viewpoint of the running times.

Another consideration deserves the behaviour of the accuracy with respect to the number of confident predictions selected during the iterative process. We see that the lowest value of $size_r$ (5%) guarantees the more stable (less variable) F-score response over the three base learners (Figures 4a, 6a, 8a), meaning that the refinement process of the predictor F allows effectively us to improve the predictions of the instances, which are selected later, instead of removing them from U at the early. This is confirmed by the higher variance of the F-score when $size_r$ is 15%.

As to the neighborhoods, the indication we can draw is the higher accuracy is obtained with the larger number of neighbours. In fact, we see F-score values greater than 0.95 only for the configuration n_10_500 (Figures 8), on the contrary, for n_5_500 and n_10_250, the accuracy is under the threshold of 0.95 (Figures 4 and 6). This is why the use of greater neighborhoods generally leads to increase the ''awareness'' of the predictor F about the surrounding instances of a target instance and, consequently, improve the prediction of the trustworthiness levels. In any case, this confirms the advantages of the use of feature augmentation to account for ''contextual'' noise readings.

Finally, as to the base learners, we observe that the Random forest implementation offers the better F-score values. Compared to the Decision Tree implementation, it gains less accuracy, especially in the last iterations.

VII. CONCLUSION

Enriching Mobile Crowd Sensing (MCS) data with contextual details is essential to maximize the effectiveness of contributed data without explicitly requesting additional information to the end-user. This paper proposes to leverage on machine learning to contextualize the gathered observations, in a way that is both resource efficient and accounts for the specific of the crowdsensors, spanning the device characteristics, the end-user's behavior and the environment. We have designed a computational solution working on the most usual and recurrent scenarios for monitoring urban noise pollution, assuming that we know the context and situation the devices operate only for a part of them. Therefore, we have developed a transductive learning algorithm able to learn on the data coming from fully known devices and infer context and working situations for those devices with a partial description. Additionally, transductive learning is able to refine the accuracy of the inferential model with computational costs we can keep under control. The proposed algorithm works on a scenario of the balanced classification, but we plan to investigate the case of imbalanced data.

The application scenario has been provided by the MCS-enabled platform *Apollon* for urban pollution

monitoring (currently in deployment in Apulia Region - Italy). Amongst the several architectural components and monitored environmental pollutants only the management of MCS-collected noise levels has been considered. In addition to noise level peculiarities, an extensive description of the contextual metadata achievable from MCS sources as well as their proper combination with noise levels has been addressed. A validation session has been conducted to quantitatively measure the influence of the working conditions on the accuracy. We firmly consider that crowdsensing will be increasingly a significant source of data to be exploited for civic purposes. However, this also means attracting a large-enough crowd over time. This is known to be a hard problem and solutions lie in the ability for the crowdsensing applications to self-adapt to the end user scenarios. We are currently investigating such solutions where urban computing leverages on intelligent systems to maximize the exploitation of data collected from the crowd.

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[2] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *Proc. IEEE 12th Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2011, pp. 1–6.

[3] C. Leonardi, A. Cappellotto, M. Caraviello, B. Lepri, and F. Antonelli, "SecondNose: An air quality mobile crowdsensing system," in *Proc. 8th Nordic Conf. Hum.-Comput. Interact., Fun, Fast, Foundational (NordiCHI)*, Oct. 2014, pp. 1051–1054.

[4] E. Minkman, "Citizen science in water quality monitoring," M.S. thesis, Dept. Sci. Educ. Commun., TU Delft, Delft, The Netherlands, 2015.

[5] A. Bartonova, "CITI-SENSE—Citizens' observatories and what they can do for you," CITI-SENSE, Oslo, Norway, Tech. Rep., 2016.

[6] M. Zappatore, A. Longo, and M. A. Bochicchio, "Crowd-sensing our smart cities: A platform for noise monitoring and acoustic urban planning," *J. Commun. Softw. Syst.*, vol. 13, no. 2, pp. 53–67, Jun. 2017.

[7] A. Longo, M. Zappatore, and M. A. Bochicchio, "Collaborative learning from mobile crowd sensing: A case study in electromagnetic monitoring," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Mar. 2015, pp. 742–750.

[8] "Smart parking: Towards building smarter cities," Beecham Res., Cambridge, U.K., Tech. Rep., 2014.

[9] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Trans. Sensor Netw.*, vol. 11, no. 4, Dec. 2015, Art. no. 55.

[10] A. Salfinger, "Crowd^{SA}—Towards adaptive and situation-driven crowdsensing for disaster situation awareness," in *Proc. IEEE Int. Multi-Disciplinary Conf. Cogn. Methods Situation Awareness Decis. Support (CogSIMA)*, Mar. 2015, pp. 14–20.

[11] J. Radianti, J. Dugdale, J. Gonzalez, and O.-C. Granmo, "Smartphone sensing platform for emergency management," in *Proc. 11th Int. Conf. Inf. Syst. Crisis Response Manage.*, S. Hiltz, M. Pfaff, L. Plotnick, and A. Robinson, Eds., May 2014, pp. 379–383.

[12] A. Stopczynski, J. E. Larsen, S. Lehmann, L. Dynowski, and M. Fuentes, "Participatory Bluetooth sensing: A method for acquiring spatio-temporal data about participant mobility and interactions at large scale events," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2013, pp. 242–247.

[13] M. Louta, K. Mpanti, G. Karetsos, and T. Lagkas, "Mobile crowd sensing architectural frameworks: A comprehensive survey," in *Proc. 7th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2016, pp. 1–7.

[14] A. Longo, M. Zappatore, and A. De Matteis, "An osmotic computing infrastructure for urban pollution monitoring," *Softw., Pract. Exper.*, pp. 1–25, Jun. 2019. doi: 10.1002/spe.2721.

[15] F. Sailhan, V. Issarny, and O. Tavares-Nascimiento, "Opportunistic multi-party calibration for robust participatory sensing," in *Proc. IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2017, pp. 435–443.

[16] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proc. Int. Symp. Handheld Ubiquitous Comput.* Cham, Switzerland: Springer, 1999, pp. 304–307.

[17] O. Yurur, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung, "Context-awareness for mobile sensing: A survey and future directions," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 68–93, 1st Quart., 2016.

[18] B. Lefevre and V. Issarny, "Matching technological & societal innovations: The social design of a mobile collaborative app for urban noise monitoring," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 33–40.

[19] T. Luo, J. Huang, S. S. Kanhere, J. Zhang, and S. K. Das, "Improving IoT data quality in mobile crowd sensing: A cross validation approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5651–5664, Jun. 2019. doi: 10.1109/JIOT.2019.2904704.

[20] Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric Internet of Things," 2014, *arXiv:1407.2704*. [Online]. Available: https://arxiv.org/abs/1407.2704

[21] C. Cichy and S. Rass, "An overview of data quality frameworks," *IEEE Access*, vol. 7, pp. 24634–24648, 2019. doi: 10.1109/ACCESS.2019.2899751.

[22] R. J. Price, D. Neiger, and G. G. Shanks, "Developing a measurement instrument for subjective aspects of information quality," *Commun. Assoc. Inf. Syst.*, vol. 22, Jan. 2008, Art. no. 3.

[23] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Sci. J.*, vol. 14, p. 2, May 2015.

[24] A. Karkouch, H. Mousannif, H. A. Moatassime, and T. Noël, "Data quality in Internet of Things: A state-of-the-art survey," *J. Netw. Comput. Appl.*, vol. 73, pp. 57–81, Sep. 2016. doi: 10.1016/j.jnca.2016.08.002.

[25] N. Javed and T. Wolf, "Automated sensor verification using outlier detection in the Internet of things," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Macau, China, Jun. 2012, pp. 291–296. doi: 10.1109/ICDCSW.2012.78.

[26] K. J. Hole, *Anomaly Detection With HTM*. Cham, Switzerland: Springer, 2016, pp. 125–132.

[27] C. Wu, T. Luo, F. Wu, and G. Chen, "Endortrust: An endorsement-based reputation system for trustworthy and heterogeneous crowdsourcing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–6. doi: 10.1109/GLOCOM.2015.7417352.

[28] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: The case for a reputation system in participatory sensing," in *Proc. 13th ACM Int. Conf. Modeling, Anal., Simulation Wireless Mobile Syst.*, Bodrum, Turkey, Oct. 2010, pp. 14–22. doi: 10.1145/1868521.1868526.

[29] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 159–170, 2nd Quart., 2010. doi: 10.1109/SURV.2010.021510.00088.

[30] P. Legendre, "Spatial autocorrelation: Trouble or new paradigm?" *Ecology*, vol. 74, no. 6, pp. 1659–1673, Sep. 1993.

[31] C. Loglisci and D. Malerba, "Leveraging temporal autocorrelation of historical data for improving accuracy in network regression," *Stat. Anal. Data Mining*, vol. 10, no. 1, pp. 40–53, Feb. 2017. doi: 10.1002/sam.11336.

[32] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 245–284, Feb. 2015. doi: 10.1007/s10115-013-0706-y.

[33] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[34] C. A. R. de Sousa, V. M. A. Souza, and G. E. A. P. A. Batista, "An experimental analysis on time series transductive classification on graphs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[35] J. Neville and D. Jensen, "Iterative classification in relational data," in *Proc. 17th Intl. Joint Conf. Artif. Intell.*, Jul. 2000, pp. 13–20.

[36] L. McDowell, K. M. Gupta, and D. W. Aha, "Case-based collective classification," in *Proc. 20th Int. Florida Artif. Intell. Res. Soc. Conf.*, D. Wilson and G. Sutcliffe, Eds., May 2007, pp. 399–404.

[37] O. Ohashi and L. Torgo, "Spatial interpolation using multiple regression," in *Proc. IEEE 12th Int. Conf. Data Mining*, Brussels, Belgium, Dec. 2012, pp. 1044–1049.
[38] A. Longo, M. Zappatore, M. Bochicchio, and S. B. Navathe, "Crowd-sourced data collection for urban monitoring via mobile sensors," *ACM Trans. Internet Technol.*, vol. 18, no. 21, Dec. 2017, Art. no. 5. doi: 10.1145/3093895.
[39] C. A. Kardous and P. B. Shaw, "Evaluation of smartphone sound measurement applications," *J. Acoust. Soc. Amer.*, vol. 135, no. 4, p. EL186, Jan. 2014.
[40] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "Mllib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, Jan. 2016.

MARCO ZAPPATORE received the master's degree in telecommunication engineering and the Ph.D. degree in information engineering from the University of Salento, Italy, in 2008 and 2012, respectively. He is currently a Consultant for Hesplora s.r.l. He collaborates with the Database and Information Systems research group, Engineering for Innovation Department, University of Salento. He is the coauthor of more than 70 scientific publications in national and international conferences and journals. His research interests include data and knowledge management, mobile crowd sensing, sensor data management, and wireless propagation estimation. He is also a Reviewer for several international scientific journals in sensors, data management, and wireless communications research areas.

CORRADO LOGLISCI received the Ph.D. degree in computer science, in 2008. He has been assigned as a Foreign Teacher for Master courses at Universitè de Sciences, Montpellier, France, and University of New York, Tirana, Albany. He is currently an Assistant Professor with the Department of Computer Science, University of Bari Aldo Moro, Italy. He is a Co-Supervisor of Ph.D. students enrolled at the Ph.D. Programme in computer science and mathematics of the University of Bari. He has published several articles in refereed journals, international conferences, and workshops. He has participated in European and national research projects concerning data mining and knowledge discovery. His current research interests include data mining, text mining, and machine learning.

He is serving/has served as a Program Committee Member and a Reviewer for international and national conferences and journals. He has participated to the organization of the international workshops, winter schools. He is also a Guest Editor of several of special issue numbers for *Journal of Intelligent Information Systems*.

ANTONELLA LONGO received the Ph.D. degree in information engineering, in 2004. She teaches data management and big data management for supporting decision making at management engineering and business school master courses. She is currently an Assistant Professor with the Department of Engineering for Innovation, University of Salento. She carries out her research activity at the Software Engineering and Telemedia Laboratory (SET- Lab), University of Salento, where she coordinates the research activities about service modeling and computing, and the applications in smart cities. Her research interests deal with information systems and databases, service-oriented architectures design for cloud infrastructure, technology-enhanced learning, and citizen science. Her current research interests include big data management and exploration of cloud architecture integration with edge computing in smart cities. On these topics, she published more than 80 articles in peer-reviewed journals and international conference proceedings.

MARIO A. BOCHICCHIO received the degree in electronic engineering and the Ph.D. degree from Bari Polytechnic, in 1991 and 1995, respectively, and the Habilitation degree from the University of Lecce, in 1997. He has more than 16 years' experience in technology-enhanced learning, in applied informatics and databases, and more than 20 years' experience in teaching informatics, databases, information systems, conceptual modeling, remote engineering, and online labs. He was involved in more than 15 EU- and national projects in eLearning as a Project Leader or as a collaborator, including FP6, Leonardo-Youth joint project, FIRB, FISR, PON, PO-FESR, and so on. He is currently an Associate Professor of database with the School of Information Engineering, University of Salento. He is the coauthor of more than 90 publications. He was an elected member of the Executive Committee of the International Association of Online Engineering. He is the National Coordinator of the Digital Health Working Group at the National Interuniversity Consortium for Informatics (CINI).

LUCIA VAIRA received the master's and Ph.D. degrees in information engineering from the University of Salento, in 2012 and 2016, respectively. She is currently a Consultant for Hesplora s.r.l. and collaborates with the Database and Information Systems research group, Engineering for Innovation Department, University of Salento. She is the coauthor of more than 30 publications. She has published and presented articles at various journals, national and international workshops, and conferences in computer science and healthcare informatics, such as WSE, IVAPP, SEBD, BIBE, BIBM, ITBAM, expAT, and BHI. Her research interests include database modeling, data warehouses, conceptual design of data warehouses, multidimensional databases, data warehouse security and quality, OLAP, healthcare databases, and incompleteness, inconsistency, and uncertainty issues when dealing with medical data. Eng. She has served as a Program Committee Member of several workshops and conferences, such as BIBM, IWDM, IWBBIO, IISSCC, DEXA, PSCare, DBKDA, and icts4ehealth, and has also spent some time as a reviewer of journals, such as TLDKS and JSCI.

DONATO MALERBA received the M.Sc. degree in computer science from the University of Bari, Italy, in 1987. He is currently a Full Professor with the Department of Computer Science, University of Bari. He has been responsible for the local research unit of several European and national projects. He is the Director of the Computer Science Department, University of Bari, and the CINI Lab on Big Data. He is on the Board of Directors of the Big Data Value Association and the Partnership Board of the PPP Big Data Value. He has published more than 200 articles in international journals and conference proceedings. His research interests include machine learning, data mining, and big data. He is on the editorial board of several international journals. He received an IBM Faculty Award, in 2004. He was a Program (Co-)Chair of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA-AIE) 2005, International Symposium on Methodologies for Intelligent Systems (ISMIS) 2006, SEBD 2007, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD) 2011, and was the General Chair of ALT/DS 2016.