Efficiently Integrating Mobility and Environment Data for Climate Change Analytics

Isam Mashhour Al Jawarneh, Paolo Bellavista , Antonio Corradi, Luca Foschini , Rebecca Montanari

Dipartimento di Informatica – Scienza e Ingegneria, University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
{isam.aljawarneh3, paolo.bellavista, antonio.corradi, luca.foschini, rebecca.montanari}@unibo.it

***Abstract*—Recent research focuses on building Cloud-based solutions for big geospatial data analytics. Avalanches of georeferenced mobility data are being collected and processed daily. However, mobility data alone is not enough to unleash the opportunities for insightful analytics that may assist in mitigating the adverse effects of climate change. For example, answering complex queries such as follows: "what are the Top-3 neighborhoods in Buenos Aires in terms of vehicle mobility where the index of PM10 pollutant is greater than 40". Similar queries are necessary for emergent health-aware smart city policies. For example, they can provide insights to municipality administrators so that they foster the design of future city infrastructure plans that feature citizen health as a priority. For example, building mobile maps for daily dwellers so that to inform them which routes to avoid passing-through during specific hours of a day to avoid being subjected to high-levels PM10. However, answering such a query would require joining real-time mobility and environment data. Stock versions of the current Cloud-based geospatial management systems do not include intrinsic solutions for such scenarios. In this paper, we report the design and implementation of a novel system MeteoMobil for the combined analytics of information representing mobility and environment. We have implemented our system atop Apache Spark for efficient operation over the Cloud. Our results show that MeteoMobil can be efficiently exploited for advanced climate change analytics.**

Keywords— Meteorology, climate change, spatial, Apache Spark, smart city

I. INTRODUCTION

The introduction of cheap IoT devices have caused an unprecedented accumulation of mobility (human, vehicles etc.,) and other kinds of georeferenced data (such as tweets from the micro-blogging Twitter, and images that may be tagged with location where they have been taken) [1]. But mobility and other georeferenced data do not occur in isolation. Often, to be able to get useful insights from such data, the context that is surrounding its temporal existence is required.

Context is loosely defined as any associated information that is useful for characterizing the situation of an object. Objects include people, locations, and any information that is relevant for modelling the correlation between dwellers the surrounding environment [2]. Metrological and climatological information are considered context.

Innumerable scenarios in smart cities and urban informatics require joining metrological information with mobility data to get useful insights that can inform better strategic decisions for city planning. A canonical scenario is the case when a municipality administration in a metropolitan city wants to figure out the relationships that may exist between climate change and the mobility of vehicles and dwellers. In doing so, they aim to plan a city for a better health of citizens. For example, by restricting access of vehicles to specific zones of the city during peak hours of a busy day. This could be a decision that results from noticing that vehicle-caused high levels of Particulate Matters (PM10 and PM2.5) are generated in those zones, which are basically a consequence of high traffic congestion.

Contextual climatology data is also georeferenced and is often collected by moving or in-situ stations. This implies the fact that joining them with mobility data requires applying spatial join operators, which are typically expensive [3]. In this paper, we present a QoS-aware Cloud-based system for joining georeferenced environment (e.g., meteorological) with mobility traces. Our system can efficiently tag mobility data with environment (e.g., meteorological) data at scale. Also, it features an SQL-alike API for simplifying queries such as aggregation, grouping and statistics on environmentally-tagged mobility data.

The remainder of the paper is organized as follows. We first briefly discuss the related background and literature. We thereafter show the design and characteristics of our system. In what follows, we discuss the results that we have obtained comparing our new methods with a representative baseline. We conclude the paper by relevant remarks and recommending future research frontiers.

II. THEORITICAL FOUNDATIONS AND RELATED LITERATURE

A. Spatial Join Processing

Georeferenced data streams are normally served to processing systems as parametrized objects, typically represented in table-like formats [4]. For example, a point in a trajectory of a moving object is represented by two coordinates, longitude, and latitude. Moving data this way relieves the pressure on the network as floating points representations are lighter than spatial objects. However, data

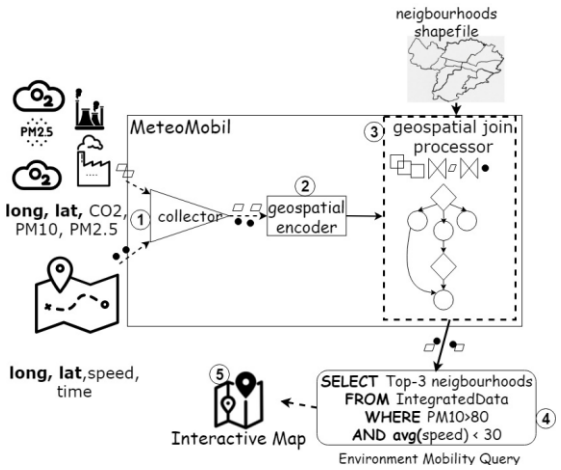

Fig. 1. Overview of MeteoMobil architecture

loses its original shape, requiring the receiving system to reconstruct it into its original shape (i.e., multidimensional), which implies the application of a costly operation known as Point-in-Polygon (PIP). PIP seeks to specify to which zone in real geometries a parametrized pair of longitude and latitude belongs [3]. Efficient join algorithms are based on a multidimensionality reduction approach known as filter-and-refine. It simply works by applying a cheap filtering stage at first, pairing the reduced representation of a point with a list of values representing the covering area (e.g., neighborhoods in a city). This works as a quick-and-dirty sieve that leaves few points with uncertainty known as false positives, where there reduced representation version matches some of the representations of the covering area, while they do not belong to those areas in real geometries [5]. A well-performing multidimensionality reduction approach is the geohash encoding, which is based on z-order curves. It is as simple string representation of pairs of parametrized coordinates. Cloud-based efficient algorithms typically work by generating a geohash encoding for every parametrized pair of a trajectory point. Also, a list of geohashes representing each zone in the study area (e.g., neighborhoods in a city) is generated. The filtering approach then pairs each geohash-represented point with the geohash covering list. The refinement stage thereafter proceeds to applying the costly PIP geometric operation to refine the false positives and specify which of them belongs to the real geometry that has been presumed during the filter phase [6]. Since both environment (e.g., meteorological) and mobility data are georeferenced, spatial join is an indispensable operation for achieving the data integration. However, since spatial join is expensive, novel algorithms are required to adapt so that they can integrate those data sources. This is so because given two parametrized datasets, the join is applied to each dataset individually.

B. Applications of mobility-environment data integration

There are several interesting scenarios where the integration of mobility and environment data is beneficial. For example, the work by [7] has designed a 'Green Paths' routing software for the assessment of environmental exposure to traffic-cause pollutants. The software features an interactive map for recommending healthy routes to dwellers in the Helsinki Metropolitan Area in Finland. However, the system is not

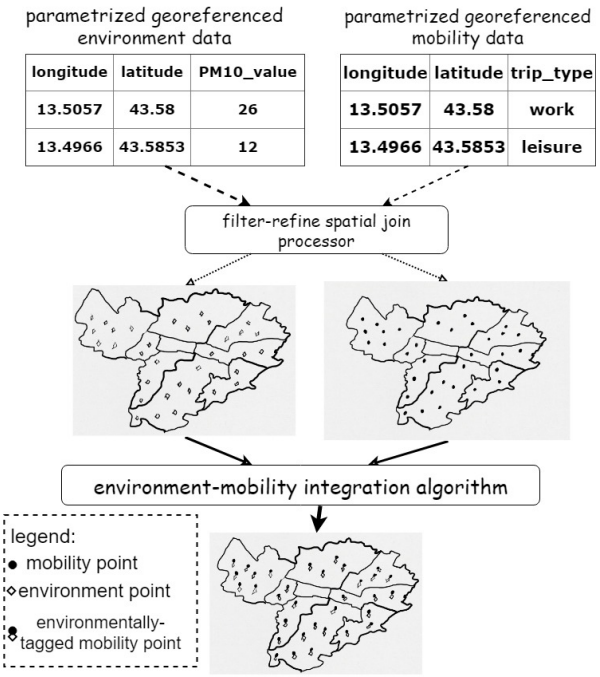

Fig. 2. Heuristic view of MeteoMobil

optimized for joining mobility and environment data at scale as it is not designed to be operated in the Cloud.

As the costly spatial join is a fundamental operation in dynamic smart city scenarios. Various works from the relevant literature have focused on optimizing the join processing by applying multidimensionality reduction and custom spatial indexing techniques. For example, in a previous work [8], we have designed a Cloud-based efficient methods for processing proximity-alike queries on geospatial data. The system is engineered atop Apache Spark for supporting QoS-aware spatial proximity queries and aggregations. The system has potential in scenarios that require joining spatial mobility and environment data and it can be further customized in that direction.

In the same vein, in a previous work [9], we also have designed a Cloud-based storage version for optimizing the storage and management of big geospatial data within NoSQL storage databases. Specifically, we have presented spatial optimization layers for MongoDB. This work complements our work in [8] to allow most common spatial analytics that require storing in addition to processing of geospatial mobility data streams. This work also has a potential to be extended to a mobility-environmental data integration scenario.

An important work appears in [10], where dwellers represent active collectors of air pollution's measurements in few locations of a city. This information can then be integrated with mobility data to form mobile health communities for better health of citizens. The integrated version of data then can be fed to a recommender system that recommends the less-pollutant areas to dwellers based on their health condition.

In addition, a recent work from the relevant literature by [11] has focused on tackling the problem of mobility-environment data integration from different perspective. Particularly, instead of joining mobility and environment data, they have designed a piecemeal emission model that quantifies four air-

borne pollutants including the PM10. Their framework proceeds as follows. It first applies a prefiltering stage to select points with specific speed and acceleration values that do not exceed a prespecified threshold. Thereafter, they project each point with a network graph representing road segments, which is analogous to solving PIP spatial join for each point independently. They further utilize the ball-tree nearest neighbourhoods' algorithm for a quicker Haversine computation. Haversine is a well-known equation for calculating the distance between two points given their longitude/latitude coordinates. Thereafter, they apply the model that appears in [12] for calculating the microscopic emission resulted from a moving vehicle at each point in time. However, the system is not designed to join mobility data with other more accurate sources of environment data at scale. Also, it is not designed to be operated on Cloud deployments.

III. METEOMOBIL: AN EFFICIENT SYSTEM FOR ENVIRONMENT AND MOBILITY DATA INTEGRATION

In this section, we introduce the architecture and features of our novel system that we term as MeteoMobil (short for meteorological mobility). MeteoMobil is a novel system for tagging, at-scale, massive amounts of mobility with environment (e.g., meteorological) and climatological contextual information. It basically constitutes three main components: data collector, join and query processors.

The architecture of MeteoMobil is depicted in Fig. 1. Georeferenced data that is coming from in-situ (or moving) environment (e.g., meteorological) and weather stations is collected by the data collector. Also, mobility data is collected by the same component. A shapefile representing the city is also ingested by the collector. After collecting all needed sources of data, it is fed to a geospatial encoder, which is responsible for encoding the mobility and environment (e.g., meteorological) data. It does so by reducing the dimensionality through applying a geohash function. Encoded data is then served to a spatial join processor, which is responsible to integrate environment (e.g., meteorological) data with mobility data at various levels. The result of this integration is a unified view of environmentally-tagged mobility data traces. This view is then materialized and fed to the query processor, which receives an environment (e.g., meteorological) mobility query from the user and computes the result interactively. Environment (e.g., meteorological) data is ingested as GRIB files from the source and transformed into a CSV file format by the data collector component of our system.

In the following subsection, we describe a novel simple, yet effective, algorithm for joining georeferenced environment (e.g., meteorological) and mobility data at scale.

A. Environment (e.g., meteorological) and mobility data integration at scale

MeteoMobil features a join processor that implements a novel simple spatial join processing algorithm that we have designed. The method starts by geocoding the mobility and environment (e.g., meteorological) data using geohash. As geohash is a dimensionality reduction approach that acts as a quick-and-dirty sieve [13]. We utilize an approach similar to filter and refine approach. First, we geocode the mobility traces and produce the corresponding geohash of each longitude/latitude pair. We do the same to the environment data, generating a geohash code for each record. Thereafter, we apply an efficient spatial join method that is based on filter-and-refinement to find out to which polygonal area (e.g., neighbourhood) each point from the two datasets belongs. The result is two datasets containing geohash representation and polygon (e.g., neighbourhood) for each point. Afterwards, we perform a simple cheap equijoin on the two datasets, joining the geohash-encoded and neighbourhood fields from both datasets. This strategy reduces the cost of the join significantly as it will be discussed in section IV.C (results and discussion). Our method is equivalent to the heuristic overview that is shown in Fig. 2. It resorts to overlaying corresponding maps of both datasets with a cheap equijoin operation.

B. Supported Queries

We currently support two basic groups of queries; from which other complex queries such as spatial clustering are easily constructible.

Group1 (G1). Single queries (a.k.a. **linear**). We support single statistics queries such as the following: "what is the average PM10 for all neighbourhoods in the City of Bologna in Italy where mobility data records are greater than 3K". Such kind of queries helps in characterizing and modelling the relationship between mobility traffic intensity and the concentration levels of air-borne pollutants such as PM10. This helps in identifying the autocorrelation and maybe accepting or rejecting some other factors that contribute to high levels of PM10 such as the availability of industrial factories in specific areas. For example, to specify whether those factories are complying with the set rules of acceptable limits of air pollutant's emissions.

Group2 (G2). aggregation queries (Top-N). The other type of queries we are supporting is the Top-N, such as the following query: "which are the top-3 neighbourhoods in Bologna in Italy in terms of mobility traffic where concentration of PM10 is greater than 12". This query reveals the Top-N regions in a city where a significant increase in PM10 is associated with the intensity of the traffic, thus helping municipalities in deciding future urban planning infrastructures. For example, increasing the greenery areas in those zones and limiting the access of vehicles during specific hours of the day or specific days of a week.

IV. IMPLEMENTATION INSIGHTS

To show the capabilities of MeteoMobil and its optimizers, we have engineered a standard-compliant prototype above Apache Spark [14]. The stock version of Spark does not support spatial data management, but it is an efficient jumping-off point for building optimized spatial layers for geospatial data processing [15].

A. Deployment Settings and Benchmarking

This section elaborates the deployment settings that we have chosen to validate the effectiveness of MeteoMobil and its optimizers.

Dataset. For benchmarking, we use two datasets. The first dataset comes from the Urban SIS [16]. It offers environment (e.g., meteorological) data at a granular scale of $1km^2$ for few European countries including the city of Bologna in Italy. Most importantly, the dataset contains the daily concentration values of PM10 pollutant covering the city of Bologna in addition to other European cities, measured in $\mu g/m^{-3}$ units

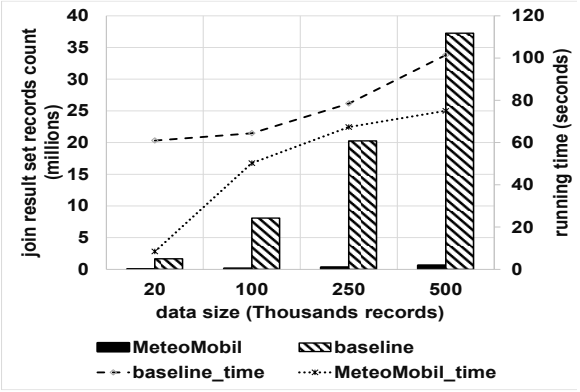

Fig. 3. Running times and number of records in the result set comparing MeteoMobile against the baseline using the ParticipAct and Urban SIS datasets. Parameters: geohash 30

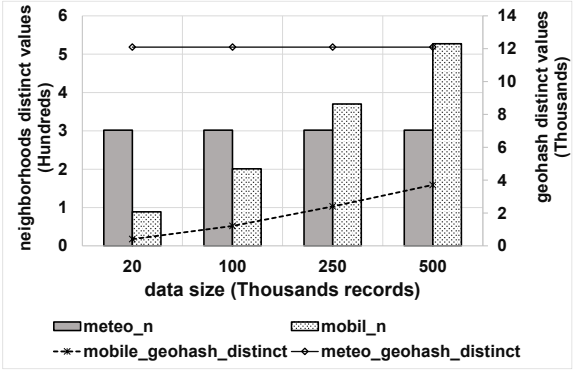

Fig. 4. Number of distinct geohashes and neighbourhoods in both datasets. In the legend: 'meteo_n' is the distinct meteorological neighborhoods values, while 'mobil_n' is the distinct mobility neighborhoods values.

with a maximum value that reaches 50 $\mu g/m^{-3}$. This data comes in a NetCDF file format, and our collector component features a module that transforms it into a more manageable Comma-Separated Value (CSV) format.

The other dataset is a cohort of 500k mobility points collected within the ParticipAct project [17], which is a project that has been conducted at University of Bologna in Italy and aims at achieving the People as a Service (PaaS) vision, where people act as active collectors of data that can be exploited and applied to interesting smart city scenarios. Every spatial point has a user locational data (in planar GPS coordinates longitude/latitude) in addition to timestamps indicating times of data collection.

Deployment and experimental settings. We have deployed MeteoMobil on a Microsoft Azure HDInsight Cluster hosting Apache Spark version 2.2.1. It consists of 6 NODES (2 Head + 4 Worker) with 24 cores. Head (2 x D12 v2) nodes , and Worker (4 x D13 v2) nodes. Each head node operates on 4 cores with 28 GB RAM and 200 GB Local SSD memory, and quantities are double those figures for worker nodes.

B. Testing Procedure and Performance Metrics

We have selected a plain representative baseline to compare its performance with the performance of our novel system MeteoMobil. The baseline system performs the spatial join on a piecemeal resolution, with a pairwise comparison pairing every mobility point with all possible environment points. This is a Cartesian product join which results in a massive result set. The baseline system first joins every set separately with the neighborhoods. It then pairs points from

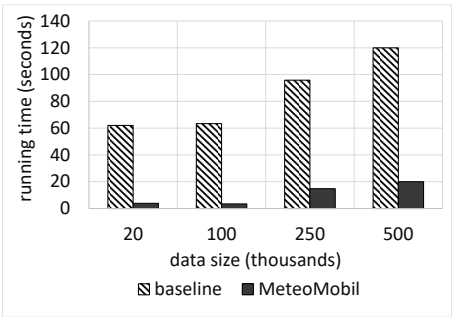

Fig. 5. Running times of Top-N queries, MeteoMobil against the plain baseline.

both datasets on their neighborhoods and point-to-point values (i.e., longitude and latitude coordinates). We simply compute the running time of both systems. The only configurable parameter in our system is the geohash precision, where we vary the geohash from 25 to 30. Geohash precision dictates the coverage area and the number of points that fall within it. A higher precision indicates a smaller area, and the opposite applies to a lower precision. We also vary the mobility data size between 20k, 100k, 250k and 500k to measure the pattern at which both systems perform under various intensities of workloads. Each experiment has been run ten times, thereafter we calculate the average of the 95th percentile.

C. Results and Discussion

We have tested our system MeteoMobil by comparing the performance of the two possible designs by which mobility-environment data integration can be achieved.

For both designs, we vary the data size and geohash precision. Then we calculate the running times and the number of records in the result set that results from the join processing. Fig. 3 shows that we obtain roughly 98% gain by for MeteoMobil against the baseline in terms of number of records in the result set, with an associated reduction in running time that is roughly equals to an average of 37%. This is easily explainable by Fig. 4, which shows the distinct number of geohash values and neighbourhoods for mobility and environment (e.g., meteorological) data. The figure shows that neighbourhood's range is far smaller compared to that of the geohash values. This means that joining at the filtering stage on neighbourhood values would result in pairing each neighbourhood with a short range of corresponding neighbourhoods in the second dataset, resulting thus in a cost-inefficient Cartesian product. On the other hand, since the geohash values in each dataset has a larger range, pairing the two datasets on geohashes first would result in a smaller result set because each geohash value in one side of the join matches only few from the other dataset.

We have also measured the running time of both systems for running an aggregate Top-N query. We specifically test on the following query: "which are the top-3 neighbourhoods in Bologna in Italy in terms of mobility traffic where concentration of PM10 is than 12". Fig. 5 shows the running

times, and it suggests that our optimized methods in MeteoMobil are able to achieve a significant performance in gain as compared to the baseline. It is natural that both systems show a monotonic increase in the running time as the data size increases. That is a significant average reduction in running time which roughly equals to 89%.

V. Conclusions and Future Works

In this paper, we have shown the design and realization of a novel efficient Cloud-based system, that we term as MeteoMobil, for environment (e.g., meteorological) and mobility data integration at-scale. MeteoMobil features a novel join algorithm that simplifies the integration. In addition, it supports SQL-alike queries which simplifies analytics on environmentally-tagged spatial data. This paves the way for the application of new queries and workloads that were hard to achieve before. For example, being able to perform aggregations on mobility data, considering the context (weather, climate) that is surrounding the existence of the points. MeteoMobil currently support single queries such as statistics (mean, count, sum) and aggregations.
Our system is applicable to smart city scenarios that feature human health as a priority. For example, it can help municipalities to decide upon locations where there is a need to construct greenery areas in an aim to promote restorative environments in urban areas. This could be useful in reducing adverse effects of traffic pollutions on human health [18].

Future research perspective would include joining other data sources that may assist in getting insightful information that reflect the socio-spatial variations, which governs the selection of alternative health-aware trip routes. This requires joining sociodemographic data of dwellers.
Future research efforts should consider developing Cloud-based open-source geospatial solutions that foster a streamlined integration with other data sources. For example, since the data that need to be collected is massive and sometimes may exceed the processing and storage capacities, there would be a need for efficient spatial- and climatologically-aware approximate techniques (similar to those that appear in [19]) for compressing and summarizing the data, probably before even reaching the Cloud-based deployment (by utilizing Fog and Edge computing).

Acknowledgment

This research was supported by the project "[H2020] SimDOME – Digital Ontology-based Modelling Environment for Simulation of Materials", Grant agreement ID: 814492.
We also would like to thank Microsoft for providing us with the free Microsoft Azure resources (through the AI for Earth project) for our project titled "Supporting Highly-Efficient Machine Learning Applications for Reducing the Impact of Climate Change on Human Health in Metropolitan Cities". All the experiments for obtaining the results presented in this paper have been conducted on a Microsoft Azure deployment (as part of the foresaid project) that comprises of an HDInsight cluster (hosting Apache Spark).

References

[1] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "QoS-Aware Approximate Query Processing for Smart Cities Spatial Data Streams," *Sensors,* vol. 21, no. 12, 2021, doi: 10.3390/s21124160.

[2] I. M. Al Jawarneh *et al.*, "A pre-filtering approach for incorporating contextual information into deep learning based recommender systems," *IEEE Access,* vol. 8, pp. 40485-40498, 2020.

[3] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "Big Spatial Data Management for the Internet of Things: A Survey," *Journal of Network and Systems Management,* vol. 28, no. 4, pp. 990-1035, 2020.

[4] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "Locality-Preserving Spatial Partitioning for Geo Big Data Analytics in Main Memory Frameworks," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 2020: IEEE, pp. 1-6.

[5] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "Efficient QoS-Aware Spatial Join Processing for Scalable NoSQL Storage Frameworks," *IEEE Transactions on Network and Service Management,* 2020.

[6] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "Spatially Representative Online Big Data Sampling for Smart Cities," in *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020: IEEE, pp. 1-6.

[7] A. Poom, J. Helle, and T. Toivonen, "Journey planners can promote active, healthy and sustainable urban travel," 2020.

[8] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, R. Montanari, and A. Zanotti, "In-memory spatial-aware framework for processing proximity-alike queries in big spatial data," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018: IEEE, pp. 1-6.

[9] I. M. Al Jawarneh, P. Bellavista, F. Casimiro, A. Corradi, and L. Foschini, "Cost-effective strategies for provisioning NoSQL storage services in support for industry 4.0," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018: IEEE, pp. 01227-01232.

[10] I. M. Aljawarneh, P. Bellavista, C. R. De Rolt, and L. Foschini, "Dynamic Identification of Participatory Mobile Health Communities," in *Cloud Infrastructures, Services, and IoT Systems for Smart Cities*: Springer, 2017, pp. 208-217.z

[11] M. Bohm, M. Nanni, and L. Pappalardo, "Quantifying the presence of air pollutants over a road network in high spatio-temporal resolution," in *Climate Change AI, NeurIPS Workshop*, 2021.

[12] M. Nyhan *et al.*, "Predicting vehicular emissions in high spatial resolution using pervasively measured transportation data and microscopic emissions model," *Atmospheric environment,* vol. 140, pp. 352-363, 2016.

[13] I. M. Al Jawarneh, P. Bellavista, L. Foschini, and R. Montanari, "Spatial-Aware Approximate Big Data Stream Processing," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019: IEEE, pp. 1-6.

[14] M. Zaharia *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM,* vol. 59, no. 11, pp. 56-65, 2016.

[15] I. M. Aljawarneh, P. Bellavista, A. Corradi, R. Montanari, L. Foschini, and A. Zanotti, "Efficient spark-based framework for big geospatial data query processing and analysis," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017: IEEE, pp. 851-856.

[16] L. Gidhagen *et al.*, "Towards climate services for European cities: Lessons learnt from the Copernicus project Urban SIS," *Urban Climate,* vol. 31, p. 100549, 2020.

[17] G. Cardone, A. Corradi, L. Foschini, and R. Ianniello, "Participact: A large-scale crowdsensing platform," *IEEE Transactions on Emerging Topics in Computing,* vol. 4, no. 1, pp. 21-32, 2015.

[18] A. Ojala, K. Korpela, L. Tyrväinen, P. Tiittanen, and T. Lanki, "Restorative effects of urban green environments and the role of urban-nature orientedness and noise sensitivity: A field experiment," *Health & place,* vol. 55, pp. 59-70, 2019.