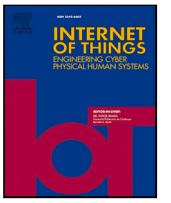
Internet of Things 25 (2024) 101009

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

In-cabin and outdoor environmental monitoring in vehicular scenarios with distributed computing

Emilio Ramos-Sorroche [a], Jesus Rubio-Aparicio [a], Jose Santa [a,*], Carlos Guardiola [b], Esteban Egea-Lopez [a]

[a] School of Telecommunication Engineering, Technical University of Cartagena, Calle del Hospital, 1, Cartagena, 30202, Murcia, Spain
[b] Departamento de Máquinas y Motores Térmicos, Universitat Politècnica de València, Camino de Vera, s/n, Valencia, 46022, Valencia, Spain

ARTICLE INFO

Keywords:
Environmental monitoring
IoT
Smart cities
Intelligent transportation systems
Crowdsensing
Edge computing

ABSTRACT

Accurate environmental monitoring is becoming the basis for assuring sustainable development in administrations at different levels, including cities and industry as key actors. However, current techniques rely on static stations that may not be representative of larger areas, for the case of outdoor scenarios, or even not considering indoor spaces where people can remain for long periods. This is the case of vehicles. The COVID-19 pandemic has remarked the importance of measuring air quality indoors, for instance. With the aim of solving this two-fold issue, this work proposes an in-cabin and outdoor air pollution monitoring system to assure healthy conditions when travelling, driving and operating vehicles, and to analyse the evolution of environmental parameters in cities. This effort is carried out exploiting distributed computing with micro-services, betting for an on-board hardware solution provided with sensors for measuring particulate matter, CO, CO_2, NO_2, O_3, temperature and humidity. While basic data pre-processing is carried out in this acquisition unit, edge processing is performed on a single board computer aboard and intermediary communication nodes in the network path from the vehicle to the cloud. Vehicle connectivity is provided by 4G cellular and Low-Power Wide-Area (LPWAN) networks. Global environmental perception is acquired by cloud-based software powered by machine learning and time series analysis. The whole solution has been validated and tested in the city of Cartagena (Spain), with good performance in terms of data collection, communication links and service offered.

1. Introduction

Sustainable evolution of urban societies is becoming of prime interest for the research community. Among all contamination sources, air pollution is continuously in the spotlight, given that it is one of the main causes of global warming and its effect on human health. Cities are in the focus of air pollution world-wide [1], and road traffic and industries are the main sources [2]. While industries can present localised polluting points, vehicular emissions, for both personal and professional use, account for 98% of carbon monoxide (CO), 61% of nitrogen oxides (NO_x) and 56% of particulate matter (PM) [3]. Hence, administrations are increasingly aware of the need for effective monitoring systems that report air quality in crowded areas.

Common air quality monitoring solutions for outdoors are based on static measuring stations that are installed at key locations in cities. The main problem here is the lack of continuity in measurements, given that only an extensive (and expensive) deployment can assure accurate monitoring across the city. When attending to indoor air monitoring solutions, the situation is even more precarious,

* Corresponding author.
E-mail address: jose.santa@upct.es (J. Santa).

https://doi.org/10.1016/j.iot.2023.101009
Received 23 June 2023; Received in revised form 23 October 2023; Accepted 21 November 2023
Available online 25 November 2023

and only the COVID-19 pandemic has encouraged institutions to measure air quality. However, current indoor deployments are mainly focused on measuring carbon dioxide (CO_2).

Crowdsensing is a technique that can help to democratise environmental monitoring. Taking advantage of people circulation in cities, fine grain measurements can be taken in urban areas if they are provided with proper equipment. Given the size and power requirements of required sensing units, a straightforward solution in this line has been equipping vehicles to take such measurements while moving. Nevertheless, as it is discussed later, current solutions in this area are mainly based on commercial units that only log data, or designs that use cellular communications to report measurements to a cloud server. When considering scenarios of thousands of units uploading data to be processed, more scalable computing and communication approaches should be applied.

In this paper, we propose a crowdsensing solution based on multi-tier edge computing for data processing, exploiting the vehicular domain. Two vehicular acquisition units have been developed for simultaneous outdoor and in-cabin air quality monitoring. Each unit is able to measure PM at diverse diameters, as well as CO, CO_2, NO_2, SO_2, O_3 gas concentrations, which are used to monitor air quality. Collected data is reported in real time using 4G, and Internet of Things (IoT) communications through Long Range (LoRa) connectivity. Data collected is pre-processed by these units and a far edge device also in the vehicle, for rapid on-board notifications to users, while other micro-services are deployed at the edge of the wired network to process data in a hierarchical way, further exploiting advantages of edge computing. Global analysis is performed in the cloud and a graphical view is given by a Web application with multi-platform support, to monitor environmental parameters. Machine Learning (ML) and time series analysis are used for pollution indexes forecasting. The solution allows regular vehicle drivers, industrial operators aboard especial vehicles and travellers aboard public transport services to be notified of harmful air conditions, to take necessary actions and avoid fatal accidents. Moreover, data collected is available to fleet and industry managers. Outdoor quality parameters measured and estimated are graphically represented, in a per-zone basis, and they are the input for additional services, such as creating healthy urban routes useful for pedestrians and riders.

The work presented in this paper offers the following scientific contributions:

1. On-board units for outdoors and in-cabin air pollution and climate monitoring, provided with IoT connectivity.
2. Multi-tier processing of environmental data using edge computing, powered by micro-services for dynamic instantiation through the data path, from vehicle to cloud.
3. Intelligent data processing for outdoor and in-cabin environmental monitoring and prediction of pollution parameters.
4. Experimental evaluation of the solution through real prototyping and tests for overall validation and performance analysis for communications and pollution forecasting.

The paper is organised as follows. Section 2 places the work in the literature. Section 3 describes the overall architecture of the solution. Section 4 details the design and implementation of the on-board equipment. Section 5 explains the processing chain of data collected, from the vehicle to the cloud. Section 6 describes the evaluation tests carried out and discusses the main results. Finally, Section 7 concludes the work, summarising the main findings and future lines.

2. State of the art

2.1. Crowdsensing for environmental monitoring

Crowdsensing using mobile sensors to gather environmental data is gaining momentum, given its potential to better reflect the status of bigger areas than the ones covered by static stations. Although there are particular proposals for using specific vehicles to measure pollution, such as recent works with drones [4], crowdsensing using personal or vehicle devices can offer more flexible solutions.

The advantages of using sensors aboard vehicles to gather air pollution information have been studied in detail in the literature. In [5], authors propose a model to estimate the air pollution due to road traffic in terms of CO_2, CO, NO, NO_2, $PM_{2.5/10}$ and hydrocarbon. However, authors only use simulation for generating traffic traces and a pollution dispersion model.

In [6], the authors describe a solution to monitor air quality in buses using commercial equipment. In [7], the results obtained from a mobile solution using Google cars is validated against certified fixed stations. The work in [8] even incorporate to the system data flows from weather stations and deal with pollutant limits. However, the works in [6–8] use off-the-shelf devices mounted on vehicles, which reduce the integration level, do not exploit novel IoT advances, cannot pre-process data, and limit flexibility of final solutions.

Recent works have developed dedicated hardware for pollution monitoring in vehicles. In [9] a prototype of a car monitoring system for CO gases is presented, although the focus is on the telematics platform and data is not further processed to offer other services. The authors of [10] developed a crowdsensing platform to monitor vehicle emissions. It is limited to CO_2 and is not focused on environmental air quality, and only a preliminary analysis of data is used to create heat maps of vehicle pollution in cities. PM and weather measurements are considered in the system presented in [11], however, it suffers from low accuracy, since a sensor in the vehicle cabin is used. Another prototype is presented and validated in a bicycle in [12], studying the number of equipped vehicles needed to cover a reference area.

Dealing with specific platforms that also add data processing advances, in [13], both a prototype and intelligent analysis of data gathered is carried out, demonstrating the potential of mobile sensors. The authors of [14] go step further, by using ML to predict and interpolate pollution levels.

The previous solutions rely on regular cellular networks (4G) to gather data, instead of IoT communications, with the associated extra costs, power consumption and lacking a real network edge for task offloading. Moreover, they are isolated proposals with limitations in scaling and data processing. On the contrary, our solution uses Low-Power Wide-Area Network (LPWAN) communications to gather pollution data in urban mobility settings, while guaranteeing data collection with an extra radio-access technology with cellular networks. It integrates within a virtualised ecosystem where data is processed in a multi-tier fashion, opting for edge computing to alleviate cloud bottlenecks and constrained on-board resources, in realistic scenarios.

2.2. In-cabin air quality assessment

Indoor air quality is receiving increasing attention these days, and this embraces the vehicular domain. Transport professionals, public transport passengers, especial machinery and regular drivers, can be exposed to polluted air for long periods. Recent literature deals with this issue. In [15], the authors analyse air pollution in terms of Volatile Organic Compounds (VOC) in two reference cars, with diesel and petrol engines, and under different driving conditions, reporting serious pollution levels in the vehicle cabin. In [16], a similar study with particulate emission is carried out with diesel and biodiesel busses, reporting potential high pollution levels for passengers, above all, when driving at streets with low ventilation due to building geometry. An equivalent study for the case of taxis is done in [17], attending to diesel and petrol engines and considering cabin isolation and air filtering, reporting noticeable pollution levels in non-modern diesel vehicles. However, these studies do not present a technological solution for continuous monitoring, since commercial meters are used in particular evaluations. In the work in [18], a prototype unit is used to continuously log CO_2, $PM_{2,5}$ and PM_{10} and predict cabin air quality in the near term. As in the case of outdoor monitoring solutions, it is based on cellular networks to report data records.

In summary, in-cabin air quality continuous monitoring still faces several challenges: requiring a higher hardware integration level, usage of IoT communication technologies, extending studies to other gasses such as CO, NO_2, SO_2 and O_3 as well, and dealing with the issue of processing huge amounts of data collected from different transport means. The solution proposed in this paper addresses these issues by a new sensing unit powered by IoT technologies, which provides both indoor and outdoor air quality monitoring capabilities to feed a data predictive analysis framework to detect harmful situations for drivers and occupants. Data processing is supported by a development based on micro-services, which makes feasible dynamic offloading of tasks among in-vehicle equipment, computing units in the network path and cloud servers.

3. Architecture of the solution

The core of the proposal is an in-cabin and outdoor air pollution monitoring system for assuring healthy conditions when travelling, driving and operating vehicles, and able to analyse the evolution of environmental parameters in cities. For this, an on-board hardware solution has been developed and equipped with sensors for PM_x, CO, CO_2, NO_2, SO_2, O_3 and climatic conditions. This integral air-quality measurement equipment is connected with a far edge device also aboard the vehicle, for local data processing. For communications, an LPWAN has been deployed to enable IoT long-range communication technology for city-wide air quality monitoring. This allow us to implement real task offloading to network edge. After data pre-processing in the vehicle, first in the two sensor units (indoor and outdoor), and then in the far edge device, extensive processing is offloaded to the edge of the network for additional intelligent analysis. While in-cabin air quality estimations will be ready at this stage, the final global assessment of outdoor air quality per geographical areas is carried out in the cloud.

Going deeper into the details of the architecture, three different layers are then identified: the on-board equipment (sensor units and far edge device), the network edge, and the cloud plane. This is illustrated in Fig. 1. The cloud/edge integration was done using the ASSIST-IoT architecture [19]. The ASSIST-IoT project[1] proposes a decentralised, layered and modular design that divides the edge-cloud continuum into independent functions and micro-services (referred as enablers according to ASSIST-IoT terminology), which are containerised, remotely deployed and orchestrated. In the context of the present work, several enablers are used to manage the life cycle of micro-services, the transmission of data to the cloud, and long-term data storage.

The vehicle side includes all far edge functionality, in the form of software and hardware entities. The in-cabin and outdoor sensor units share the same basic architecture. It comprises a specific design for air quality monitoring, in which the SOC with CPU, memory, required I/O and communication transceivers are interconnected with the different environmental sensors.

All software entities in the data path from sensor units to the cloud are developed as micro-services, with the aim of automating deployment and management of modules, and make the platform adaptable to dynamic conditions. Data collected are initially pre-processed by the same on-board (sensor) units, removing outliers and applying the required filters to output short-term averaged values. WiFi communications, these units are connected through WiFi with the on-board far edge device, which is provided with 4G communications to reach a cloud server. Pre-processed records are locally analysed in the on-board far edge device to detect harmful air quality levels for in-cabin driver and potential passengers, depending on the type of vehicle.

Both in-cabin and outdoor units are also provided with LoRaWAN communications, reporting data records continuously through a LoRa air interface. Both 4G and LoRaWAN channels are used simultaneously, but only the LoRaWAN data path exploits network edge computing. For this, a reference LoRa gateway has been set-up to gather data from vehicles driving in a reference urban area, but the deployment is open to network extensions. LoRaWAN messages are relayed by the gateway to a nearby LoRaWAN

[1] https://assist-iot.eu

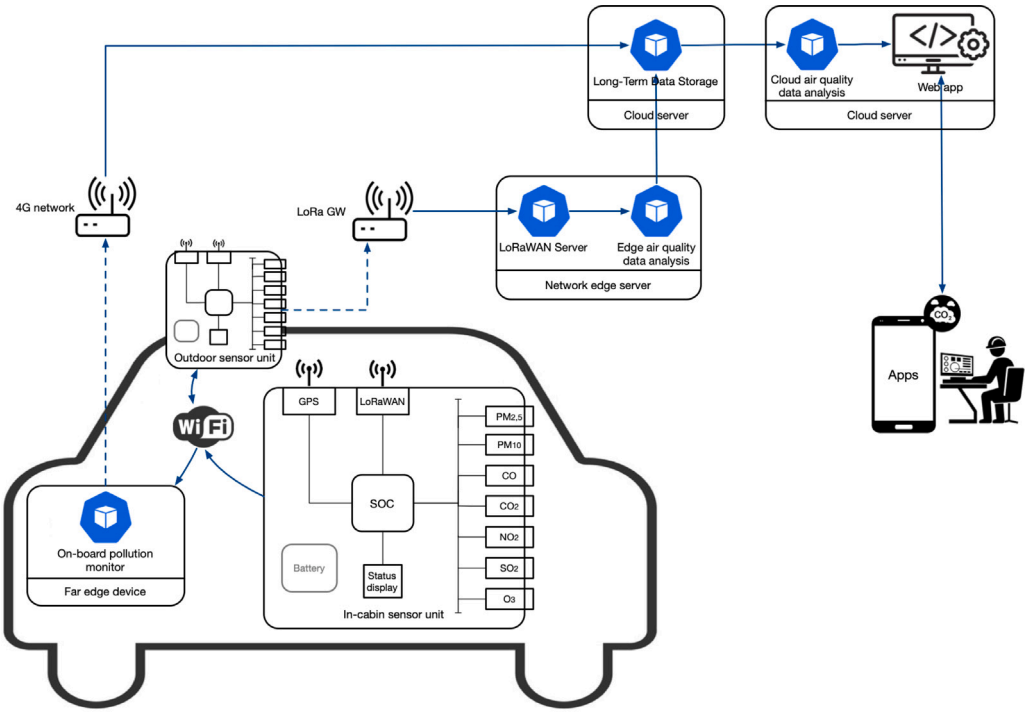


Fig. 1. Architecture of the proposed solution.

Server, which is linked with the software module in charge of edge air quality analysis. The LoRaWAN Server acts as the link-level connection with the sensor units, extracting the payload of packets and forwarding them to the edge air quality analysis module. Hence, the LoRa data path offers additional processing offloading to the network edge, especially for local and indoor air quality tasks, allowing extra data analysis computations such as using ML algorithms to not only detect harmful conditions, but also to be able to predict dangerous situations in the short and medium term.

Pre-processed data in the far edge device (in the vehicle) and the network edge server are then sent to an IoT long-term storage database, instantiated on a cloud server, which feeds data processing and visualisation modules for global analysis. The air quality data analysis carried out in the cloud is particularly addressed to compute predictions per geographical areas using outdoor measurements. Intelligent algorithms using ML and time series are used to extract patterns and predict pollution parameters within cells in a geographical grid. A Web-based application has been developed for friendly access to in-vehicle air conditions and additional services such as receiving alerts, visualising outdoor air quality per zones, and creating healthy routes in a city.

4. On-board equipment

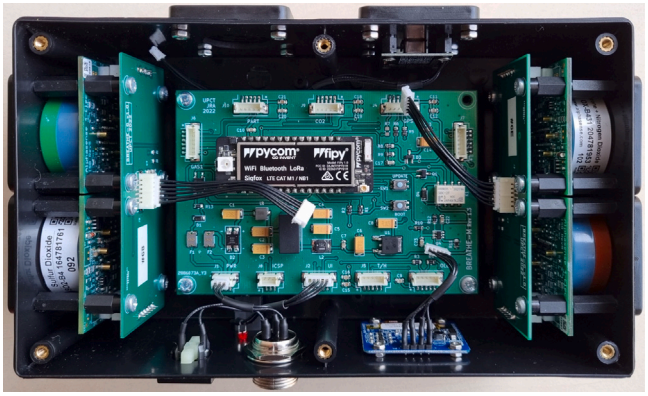
4.1. Embedded measurement units

The base sensor unit design envisioned to provide crowdsensing capabilities to vehicles is described next. The unit, powered by the vehicle electrical system, must be able to sample the most relevant air pollutants for human health, geolocate data and send them in real time. The same design has been chosen for both the inner and outer units, and the main differences between them are found in the embedded software implemented, and the provisioning of a Global Navigation Satellite System (GNSS) receiver and a magnetic surface for the outer one.

The inner view of the unit is showed in Fig. 2. On the main printed circuit board (PCB) is placed in the middle of the gas sensor boards. With this arrangement, and thanks to the provided ground plane on the lower layer of the boards, the gas sensors are partially shielded and isolated from the EMI generated in the main board due to the digital activity and the DC/DC power converter. The GNSS circuit board attached on the top case lid, remains over the main board also avoiding interference. Besides the main board, the PM sensor and the Li-Po battery are fixed on the case floor. Other auxiliary items such as the screen, push-button and connectors, are located around the main board and wired to it.

The main PCB is of a double-layer type and uses surface mount technology (SMD) for most of the components. Connectors to external signals are distributed in the periphery of the board and near to the location of the corresponding external device to improve cable organisation inside the unit. In addition, to support the microcontroller operation, a power subsystem is added to generate stable and filtered 3 V and 5 V power rails from the car variable power supply (11–14 V). Power input is polarity independent

(a) Inner view of the unit without the main board.



(b) Inner view of the unit with the main board installed.

Fig. 2. Sensor units for environmental monitoring.

to avoid damage by connection mistake. Auxiliary power input from the car ignition is optocoupler protected and can be used to switch on/off the unit as the car engine starts/stops. Both of these power lines coming from the car, are protected with resettable fuses. As the unit uses a 2200 mA h low profile LiPo battery to power the gas sensors and to keep their stability in the absence of the main power supply, a Lithium battery charger is also present in the main board, which is capable to provide up to 800 mA h and charging an empty battery in four hours. It can be disabled by the microcontroller to sample the battery output by the ADC to estimate the state of charge. A small signal relay isolates the battery from the rest of the circuit when it is used to power the gas sensors. Additional circuitry provides backup power to the GNSS module to keep navigation data updated and reduce time to first fix on start. Finally, extra filtering is carried out in power lines to improve noise figure that could affect the accuracy of measurement in sensors.

The microcontroller is a System on a Chip (SoC) module, which has been chosen to reduce the number of hardware components and, therefore, the size, power consumption and complexity of the unit. The FiPy SoC from PyCom has been selected. It integrates a dual core Espressif ESP32 CPU running at 160 MHz, 8 MB of flash memory, 4 MB of RAM, as well as diverse digital and analogue peripherals. This design uses the ADC, the I2C master controller and several I/O digital lines. Regarding communications, it includes a 2.4 GHz 802.11b/g/n WiFi and Bluetooth LE radio, LTE CAT-M1/NB-IoT capabilities, as well as a low power sub-gigahertz RF transceiver able to work in LoRaWAN and SigFox networks. In this application, the WiFi and LoRa radios are used, but a firmware update would allow other possibilities. Two external antennas are used for 868 MHz LoRaWAN and 2.4 GHz WiFi communications. They are of compact stubby type with $\frac{1}{4}$ wave radiating element and 1 dBi gain. Both of them are located externally on the top lid of the case to improve performance. Distinctive SMA and SMA-R connectors have been chosen to avoid interchange mistakes. SMA to U.FL pigtails are used internally to connect the antennas to the main circuit board. Typical transmitted power is +14 dBm to satisfy ISM band requirements. Tests have been carried out to ensure that there is no affectation between both radios and the GNSS operation.

A GNSS receiver module from Quectel is used in the external unit to get geographical coordinates. It includes an integrated patch antenna and is GPS, GLONASS, Galileo and QZSS constellations ready. Instead of placing the module on the main PCB, an additional double-side PCB has been designed to support it and provide the right ground plane for the integrated patch antenna. The GNSS PCB is attached to the case lid and the antenna is positioned up with clear path to sky. Hence, the module can be located

Table 1
Key specifications of the sensors.

Particulate Matter Sensor: Sensirion SPS-30		
Range	0 to 1000 $\mu g/m^3$	
Resolution	1 $\mu g/m^3$	
Accuracy	10 $\mu g/m^3$	
Lifetime	>8 years	
Temperature/Humidity Sensor: Sensirion SHT-30		
	Temperature	Humidity
Range	−40 to 125 °C	0 to 100%RH
Resolution	0.05 °C	0.05%RH
Tolerance	±0.3 °C	±3 %RH
Repeatability	0.06 °C	0.13%RH
Drift	<0.03 °C/year	<0.25%RH/year
CO_2 Sensor: Sensirion SCD-41		
Range	0 to 40000 ppm	
Accuracy	±(40 ppm + 5% of reading)	
Repeatability	±10 ppm	
Drift	±(5 ppm + 0.5% of reading)/year	
CO Sensor: Alphasense CO-B4		
Range	0 to 1000 ppm	
Sensitivity	420 to 650 nA/ppm at 2 ppm CO	
Linearity	20 to 35 ppb	
Sensitivity drift	<10%/year	
Operating life	>36 months (50% original signal)	
NO_2 Sensor: Alphasense NO2-B43F		
Range	0 to 20 ppm	
Sensitivity	−200 to −650 nA/ppm at 2 ppm NO_2	
Linearity	< ±0.5 ppb	
Sensitivity drift	−20 to −40%/year	
Operating life	>24 months (50% original signal)	
SO_2 Sensor: Alphasense SO2-B4		
Range	0 to 100 ppm	
Sensitivity	275 to 520 nA/ppm at 2 ppm SO_2	
Linearity	0 to −2 ppb	
Sensitivity drift	< ±15%/year	
Operating life	>36 months (50% original signal)	
O_3 Sensor: Alphasense OX-B431		
Range	0 to 20 ppm	
Sensitivity	−225 to −750 nA/ppm at 1 ppm O_3	
Linearity	< ±0.5 ppb	
Sensitivity drift	−20 to −40%/year	
Operating life	>24 months (50% original signal)	

in the topmost position of the box, avoiding RF shielding from other items, as the side sensor boards, and improving sensitivity and satellite signal acquisition. Given the limited ground plane dimensions of the patch antenna, it can be installed on the top layer of this board without affecting the main board size or the location of other components. The receiver is connected to the SoC using an UART interface, reporting location coordinates at a 1 Hz rate. A clean and stable power supply with low ripple is provided from the main board using an LDO low dropout linear regulator, decoupling capacitors and ferrite beads. According to our tests, this solution is able to achieve a fix in 90 s after powering on, and to track more than 16 satellites while moving in an urban environment.

Specific sensors have been considered for each of the pollutants to get the best sensitivity and accuracy. Particulate matter below 1, 2.5, 4 and 10 μm are measured with the SPS30 I2C sensor from Sensirion using laser scattering technology. It is fixed under the main board, towards an air entrance. CO_2 concentration is measured by the SCD41 I2C ultrasonic sensor from Sensirion, at the front side of the unit. In the case of air temperature and relative humidity, a combined sensor at the top lid, the SHT31 from Sensirion, is used to reduce the number of components. The polluting gases CO, NO_2, SO_2 and O_3 are measured by electrochemical sensors coming from the B4 series of Alphasense and located at the two case sides. They have noise figures as low as 4 ppb, which are suitable for outdoor environments and low ppb concentrations. Key specifications of all sensors can be found in Table 1.

Every sensor unit has been calibrated and their accuracy has been verified. In the case of the PM optical sensor, it includes an auto cleaning and calibration procedure triggered periodically following a predefined interval. The temperature and humidity sensor has been linearised and temperature compensated with the provided output. It includes an internal calibration memory and declares very low long-term drift. According to the manufacturer, it does not require any further intervention. This is also the case for the CO_2 photoacustic sensor, including a self-calibration algorithm. Electrochemical gas sensors, used to acquire air pollutants concentrations, are provided by the manufacturer with specific sensitivity and zero values after initial factory calibration. This data

Fig. 3. Final view of the in-cabin and outdoor sensor units.

is stored in the unit flash memory and used by the software to achieve proper measurements. Sensitivity is temperature dependent and corrected using the unit temperature sensor. For verification and fine calibration, the unit was located outside, near a reference air quality station. After the recommended stabilisation period of two hours, it was verified that gas concentrations followed the reference values for hours. Correction factors included in the embedded software allow fixing possible offset and gain errors. During operational life, sensitivity and zero drifts depend on environmental conditions, sensor type and concentration levels. Due to low gas concentrations measured (in the order of ppb), a calibration check period of six months should be established. Replacement of these sensors is recommended every 24 or 36 months, depending on the target gas, due to reduction of original output signal.

Given that gas sensors provide an analogue output in the range of mV, to reduce as much as possible the path of the output signal and the possibility of being affected by external noise, a child acquisition board has been designed. Each of these boards can support two gas sensors attached directly without any cable, as can be seen in Fig. 2. To sample and digitise the signal from the sensors, a four-channel and 16 bits ADC from Microchip is used. The ADC is managed via the I2C bus through an isolator to avoid noise from the digital circuitry. It is critical for these sensors to be fed by a noise-free power supply to output accurate measurements, so the board also includes decoupling capacitors and ferrite inductors to filter the power supplied by the LiPo battery.

The external view of the finished inner and outer units can be seen in Fig. 3. The whole set of sensors, electronic boards, antennas, screen, battery and connectors are embedded in a general-purpose black ABS plastic enclosure. External dimensions are $193 \times 113 \times 58$ mm and it is properly machined for the purpose. To optimise space, all sides of the box are used to place items. As air quality sensors need to be exposed to the surrounding environment, they have been fixed to the front and side panels and protected by individual filters. Openings drilled in the upper cover of the box allow fixing the stubby antennas, the temperature/humidity sensor and the GNSS board. Furthermore, a basic user interface has been included using a 128×64 pixels resolution I2C OLED screen and a push button. It allows to monitor the measured values by all sensors, but also to check the unit operational status. It shows the status of the WiFi and LoRaWAN network connections, the IP link between OBUs, and the GNSS fix. They are allocated next to the battery charge LED and external connectors, on the front panel. One of connectors is of standard three-pin industrial type, and it is used to power the unit from the 12 VDC car's electrical system. The remaining connector, of 3.5 mm jack type, allows firmware updates and debugging tasks. In the case of the outdoor unit, it includes on the bottom side a magnetic base with rubber finish to be firmly fixed to the car's roof. Plastic screws are considered to avoid any accidental scratch on the car. This position is chosen to minimise excessive influence of fumes from nearby vehicles in measurements.

According to the state of the art in Section 2, the unit design and development present by its own a significant contribution, given that no equivalent units for environmental monitoring using IoT capabilities and in-cabin inspection have been found in the literature.

4.2. Far edge device and in-vehicle network

The on board device at the far edge receives data from inner and outer sensor units to be pre-processed and it is in charge of data streaming via 4G network. The far edge device is implemented in the present work with a Raspberry Pi 4 Model B Rev 1.4 8 GB SoC, with a Raspbian 10 (Buster) OS (see Fig. 4). It runs a lightweight Kubernetes cluster for resource-constrained IoT devices, specifically the k3s distribution.[2] A TELTONIKA RUT955 router was used for providing 4G LTE (Cat 4)/3G/2G connectivity, and doubled as access point for the vehicular WiFi network. This internal WiFi is used for communicating the sensor units with the far edge device.

[2] https://k3s.io

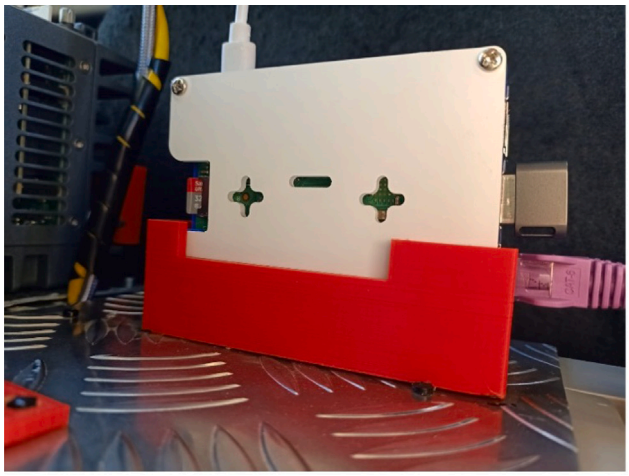


Fig. 4. Far edge device implementation.

The software philosophy based on micro-services facilitates deploying specific software components into the far edge device. Such software modules are provided in the form of Kubernetes pods that encapsulate independent functionalities and network services. For the server/decoder module that receives data from sensor units, a custom Docker image has been created, using the official node 16.18.1 image as a base, with a light Linux distribution Alpine 3.16. A StatefulSet deployment of the pod has been done, considering configuration parameters such as environment variables and NodePort type services to provide network connectivity. A MariaDB database is also encapsulated in another pod, using another custom image, optimised for use in Raspberry Pi devices, with its corresponding NodePort service. This MariaDB database locally stores the data provided by sensor units and other subsystems out of the scope of this work, focused on additional diagnostics functions connected to the vehicle control unit [20]. Finally, the MariaDB pod is connected to a main far edge pod in charge of communications that publishes the new entries into the cloud.

5. Data processing chain

One of the main contributions of the work falls on the way data is processed, given that a flexible data processing chain is proposed, in which microservices are instantiated for such a task in the network path from the sensor node to the cloud. This section details how environmental data is processed from its gathering to final presentation to users.

5.1. Data acquisition from measurement units

The embedded software of the sensor units has been coded in MicroPython, following a scalable approach that allows future adaptation to other sensors, communication transceivers or application needs. It uses threads for simultaneous management of system peripherals, battery, network communications, continuous sensors sampling and GNSS acquisition. A fault-tolerant implementation has been carried out for uninterrupted operation upon any sensor failure. The referred embedded software manages IP network communications over WiFi between sensor units and the far edge device, and the IoT link with the LoRaWAN infrastructure using the integrated ISM radio.

When a configurable timer expires or a distance threshold is exceeded, data is sent from the inner unit to the outer one using TCP/IP over WiFi, which adds its own data and perform the transmission through the LoRaWAN link and, via IP, to the on-board far edge device. Data coming from the inner unit comprises a 32-byte frame with sensor records. The outer unit is in charge of adding geolocation information before forwarding the data through LoRaWAN. The packet structure for data records has been defined to optimise available LoRa bandwidth, although it is used for IP communications with the far edge device too. Hence, it has been defined to reduce transmission errors and time on air. The packet contains metadata about the quality of measurements, including position accuracy and the stabilisation status of all sensors. All transmissions in LoRaWAN require acknowledgement, with a maximum of five attempts before dropping. This improves network reliability under mobility and signal blockage in urban settings.

5.2. In-vehicle data processing

As said, data transmission following the path through the far edge device implies that the inner unit sends the collected data to the outer one; then it adds its own data and sends them all together to the far edge device, by using a TCP/IP socket configured for such a task. Data is sent using the same raw 32-byte frame used for communicating inner and outer units.

As indicated above, the far edge device has been provided with a data collection and pre-processing Kubernetes pod. It has been programmed in node-JS language, and it populates a TCP socket-based server to which the external unit periodically sends the frames with the data agglomerating inner and outer measurements. This pod is responsible for validating data and metadata to verify that sensor readings are correct and coherent, ruling out incorrect readings due to sensor failure, as well as decoding each field to generate a data frame in JSON format. The decoded information is then sent to the Kubernetes pod that hosts the local MariaDB database. This pod is connected with the main far edge pod, implemented in Python, which sends data updates whenever they are available using the 4G link to the long-term storage in a far cloud. Apart from data reporting to the cloud, this last pod implements a logic to report pollution events in broadcast within the in-vehicle network, which are received by all nodes connected by WiFi, in such a way that air quality data in the cabin can be consulted in real time by on-board devices for the sake of preventing health risks.

5.3. Long-term data storage

The long-term data storage module, called Long-Term Storage Enabler in our prior work in [19], serves as secure and resilient storage of data collected from end devices, and it is envisioned to be deployed on the cloud rather than the edge. It has SQL and NoSQL endpoints protected by identification and authorisation capabilities. In the context of the current work, the long-term data storage module saves the data streamed from the far edge device, and serves them via a specific API for global analysis of pollution.

5.4. Processing at the wired network edge

Data sent by sensor units through the LoRaWAN link are received by the LoRa gateway, and then forwarded to the LoRaWAN Server for further packet processing. A MultiTech Conduit IP67 LoRa gateway has been used. It has been installed on the top of a university building on a hill, to improve coverage in the city centre. The LoRaWAN server has been implemented with the ChirpStack distribution,[3] instantiated as a Kubernetes pod in a laboratory server. For this purpose, a Kubernetes K8S distribution updated to its latest version at the time of testing has been used. The ChirpStack instantiation has been provided with a JavaScript plug-in to parse data collected from sensor units and populate an InfluxDB database.

Data collected from sensor units are taken regularly by an edge air quality processing pod, that encapsulates a Node.js software in charge of evaluating short-term tendencies and detecting pollution alerts to be reported in a graphical panel. The visualisation is carried out here using a Grafana[4] software that includes a set of panels for easy inspection. Data about alerts is also available through an open REST API offered by the pod.

5.5. Global analysis of pollution

The main design blocks of the software envisaged for global analysis of pollution is shown in Fig. 5. Crowdsensed environmental parameters measured from vehicles are obtained from the long-term data storage module, thanks to the provided secured REST API. The purpose is to process intelligently these data for service provision, proper visualisation and prediction. All software entities described here are installed in a laboratory server, although they are intended to be deployed in a far cloud server.

A Web platform has been developed in Angular,[5] implementing a Single Page Application (SPA) that offers a system structured in modules and independent components that facilitates scaling and adding new functions, in addition to using Typescript programming language with strict typing that improve robustness. This application has been provided with extended graphical capabilities, by integrating map visualisation tools coming from mapbox[6] to show an interactive map of the city on which the measurements are plotted in an intuitive way, depending on the selected filters. However, the views provided have been enhanced by Deck.gl,[7] which adds an extra visualisation layer over regular maps with an attractive presentation in the form of informative 3D layers of pollution levels. Variation of measurements with different metrics, both temporal and spatial, has been managed to assure simple and intuitive visual representation of air quality. Internally, the map of the city of Cartagena has been divided using a grid with cells that delimit 100 m^2 areas. The use of Deck.gl as a graphic layer engine, in conjunction with Angular, has been a challenge, since the all-in-one philosophy of the framework is quite restrictive when integrating external libraries. However, the final solution still maintains robustness and scalability of Angular, along with graphical versatility of Deck.gl.

When data prediction is required for a time window in the future for a particular zone, the Angular engine calls a series of Python-based functions in the back-end encapsulated as a Kubernetes pod. It provides a REST API developed in Flask,[8] and uses ML and time series analysis for data processing. Particularly, the scikit-learn[9] framework is used for base ML algorithms, while SKTIME[10] is used for time series analysis. Data collected in the past about environmental parameters are considered for fitting

[3] https://www.chirpstack.io
[4] https://grafana.com
[5] https://angular.io
[6] https://www.mapbox.com
[7] https://deck.gl
[8] https://flask.palletsprojects.com/)
[9] https://scikit-learn.org/
[10] https://www.sktime.org/en/stable/index.html

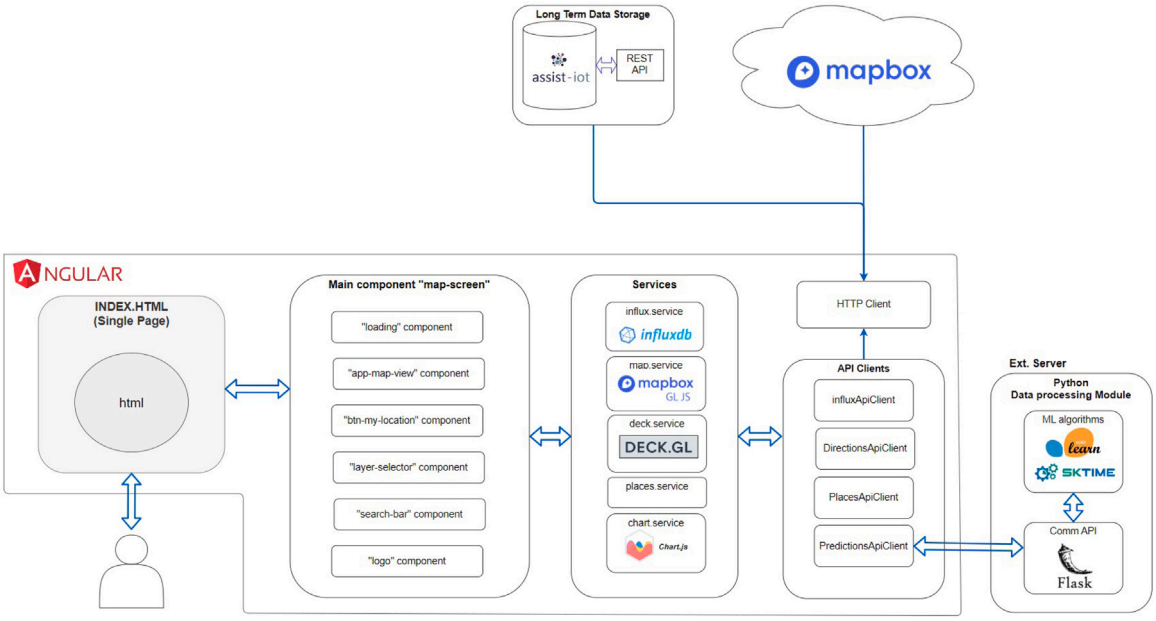


Fig. 5. Data processing software for global pollution analysis.

models that predict their future trend. This is a tricky part of the software, since new models must be created in real-time once a new visualisation area is determined within the map. The current operation of the system is satisfactory with past time windows of three years, which is the largest one, but the application is prepared to constraint the operative fitting datasets to a maximum size, in order to be responsive as the number of users increases while consistent prediction results are obtained.

Although the vehicles are in constant movement, the prediction is made on the data obtained for a certain geographical area of the grid created, so there may be temporary gaps without measurements. This is solved by performing predictions on the basis of time series taken at hourly intervals. Hence, there is a pre-processing of data prior to the prediction, in which values from the area queried are hourly aggregated. When temporal gaps with no data appear (e.g., due to low mobility or at night), imputation methods are used for missing data, taking into account the last valid measurements in the area. After data pre-processing, a prediction method based on SKTIME reduction algorithms has been chosen, consisting of transforming a regression problem with time series into a tabular regression problem. This is done by dividing the data set into different segments using sliding windows and stacking the resulting data sets to obtain a larger data set, which can be processed later with traditional scikit-learn regression algorithms. In this case we use a k-neighbours regressor with $k = 3$.

All actions from users are processed by our Angular-based engine, involving the selection of data to be shown, filters, time windows, and visualisation skin to be used, as can be seen in Fig. 6. In the application interface data are plotted on a map and, depending on the number of measurements and users, available time windows may be dynamically adapted. Users can also specify if they want to obtain and represent outdoor or indoor measurements for a vehicle. Measurements retrieved are marked with the source vehicle identifier, geographical coordinates, and an identifier indicating whether the record corresponds to an in-cabin or exterior measurement. Available data include CO, CO_2, NO_2, O_3, SO_2, particles of different diameters, temperature and humidity.

Finally, the Web application can create routes in the city by using the underlying map capabilities and offers a navigation interface to guide the user when a mobile device with GPS is used. This can be done for walking, bike and vehicle routes. Fastest and shortest paths are available, and it is also possible to plan healthy routes to avoid areas with high pollution levels.

6. Evaluation of the solution

A first set of validation and performance tests in lab and outdoors were followed by a complete evaluation in a target scenario, collecting environmental measurements and performance metrics in the city of Cartagena (Spain). The experimental evaluation campaign presents an extra contribution of the work, given the challenge of testing in real conditions the whole solution, finally demonstrating correct integration, and providing both performance and validation results of essential parts of the system, as described next.

6.1. Evaluation set-up

The reference vehicular pilot used for the evaluation was provided by the ASSIST-IoT project. This comprises a Ford Kuga vehicle, in which the far edge device and the sensor units developed were installed for evaluation.

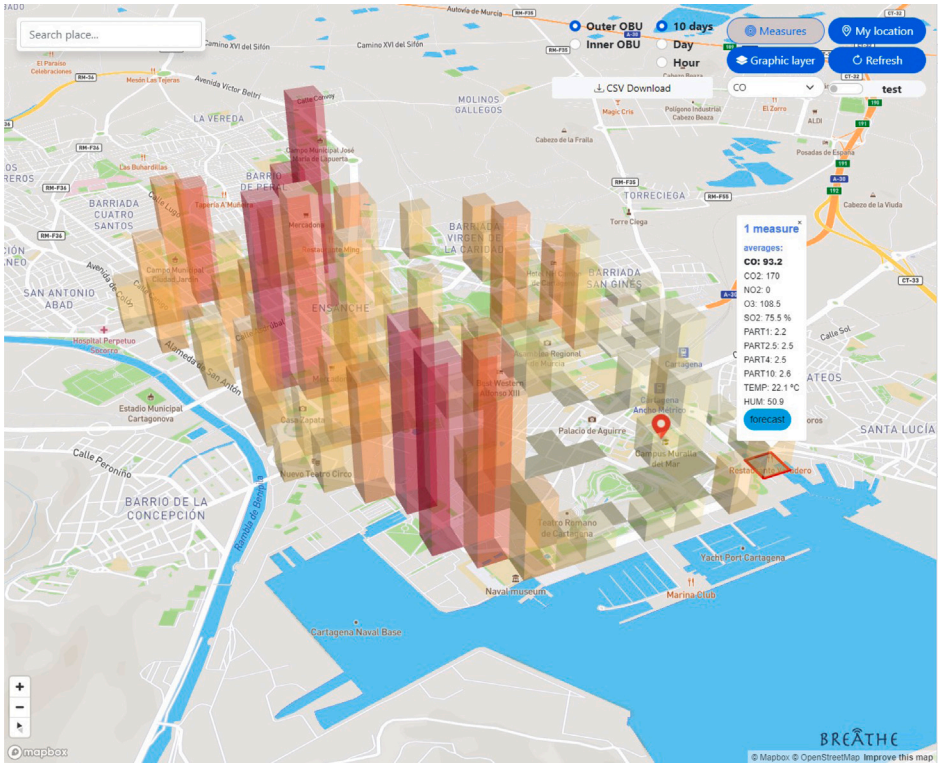


Fig. 6. Web application interface when selecting the 3D view.

Fig. 7 illustrates the set-up for the tests. The outer sensor unit was affixed on the vehicle roof using the magnetic base. This location was considered the best one for guaranteeing correct air intake. No multi-path issues with GPS nor LoRa were detected. The inner sensor unit was placed on the rear seats, in order to take measurements from the in-cabin environment. The power line of both units were connected to the vehicle battery, and they were attached to the in-vehicle WiFi network. Fig. 7(c) shows the vehicle cabin during the tests. While one of us was driving the car around the city centre of Cartagena, the co-pilot was monitoring the measurements taken by the units in real-time. On the rear-left seat, other person was seeing the same view as the co-pilot in a tablet, by replicating the laptop screen. The fourth person in the car was in charge of navigation, assisting the driver to follow the planed route, which included areas of different traffic density, without repeating streets already covered.

6.2. Performance of communications

Attending to packet losses, the usage of 4G reported a delivery success rate of 100%, while a value of 84.88% was obtained for LoRaWAN. 4G coverage and data rate were excellent along the driven route, since full coverage is offered by the mobile network operator in the city centre of Cartagena and no communication issues were detected during the tests. For the case of LoRaWAN, only one base station was used at the university campus, as explained above. A maximum distance of 2.12 km between the car and the gateway was reached. Under this situation, packets were received with RSSI levels as low as −119 dB m and SNR as low as −16.5 dB. Nevertheless, they were successfully decoded since LoRa devices are able to demodulate signals −20 dB below the noise floor. The average noise floor of the 868 MHz band in this urban area was −109 dB m, which was previously measured with a spectrum analyser connected to the gateway antenna. Main statistics of network performance are included in Table 2, taking SNR and RSSI values reported by packets received at the GW. Mean values for SNR and RSSI uplink during the route were 0.8 dB and −105 dB m, respectively. Most of the 15.12% packet losses were reported when the car was beyond 1 km in narrow streets. Good overall results were obtained, considering the mobility scenario, non-line of sight conditions, limited output RF power (14 dBm) and the low gain antenna used by the OBU. In a larger smart city deployment, LoRaWAN packet losses could be further reduced by installing extra gateways.

6.3. Evaluation of the network-edge processing solution

Fig. 8 shows a screenshot of the Grafana-based data visualisation platform populated by the edge air quality microservice. It provides a first stage to verify environmental parameters and carry out basic visual inspection. After selecting the vehicle and the

(a) Placement of the outer sensor unit.

(b) Placement of the inner sensor unit.

(c) In-cabin view during the tests.

Fig. 7. Arrangement of equipment during the validation tests.

Table 2
Communications performance in the tests.

Indicator	Result		
	Received	Delivery rate	Loss rate
Frames sent by 4G	225/225	100%	0%
Frames sent by LoRaWAN	191/225	84.88%	15.12%
	Min	Max	Avg
LoRaWAN uplink SNR	−16.5 dB	13 dB	0.8 dB
LoRaWAN uplink RSSI	−119 dB m	−66 dB m	−105 dB m

related external or internal sensor unit, the interface displays the last data acquired from all available sensors and their temporal evolution in graphs. Folding tabs are used to group related records for diverse PM diameters, gaseous pollutant concentrations and temperature/humidity environmental variables. Although it is not shown in the included figure for the sake of clarity, the radio link quality, in terms of RSSI and SNR, are also represented on maps and plots. For each sampled pollutant, there is a descriptive label and a dial indicator which changes its colour according to its impact on human health. Value ranges, colours and qualitative labels are compliant with the ones used by the European Environment Agency to calculate the Common Air Quality Index (CAQI) [21]. Furthermore, averaged historic data are shown in a plot, and the last values are listed just below.

From the data collected and plotted in the provided screenshot in Fig. 8, it can be seen that PM_1, $PM_{2.5}$, PM_4 and PM_{10} values are always below 10, 11, 12 and 12 $\mu g/m^3$, respectively, in all the sampled locations, which means that they are in the range associated to good air quality. Regarding polluting gases, the concentration of NO_2, CO_2 and O_3 keep in the most favourable value range at all locations, while CO increases at some points, due to traffic increase. For each of the concentration values at each location, partial indices are generated, with the global index referencing to the worst of the partial indexes.

6.4. Operation of the cloud-based web application

The Web application when activating the visualisation of the 3D layer is shown in Fig. 6. When selecting this visualisation type, a drop-down menu comes into play to select the air quality parameter on which the representation will be based. All measurements obtained during the tests were used to generate the view in Fig. 6. Values that fall geographically within each grid cell are averaged and added as a single representative measurement, using 3D bars that vary in height and colour, depending on the resulting value. This helps to remark areas with unhealthy conditions, alerting the user. In the image, outdoor CO conditions are monitored and the plot reveals high concentrations at locations of high traffic density in the city centre, since data were collected between 12 and 2 P.M., which includes the lunch break of many people and the end of school day, and traffic jams are common. By clicking on any of the columns, a popup appears with averaged values for all environmental parameters collected. In general, more healthy measurements were obtained at sea side.

Fig. 8. Screenshot of the data visualisation platform hosted at the network edge.

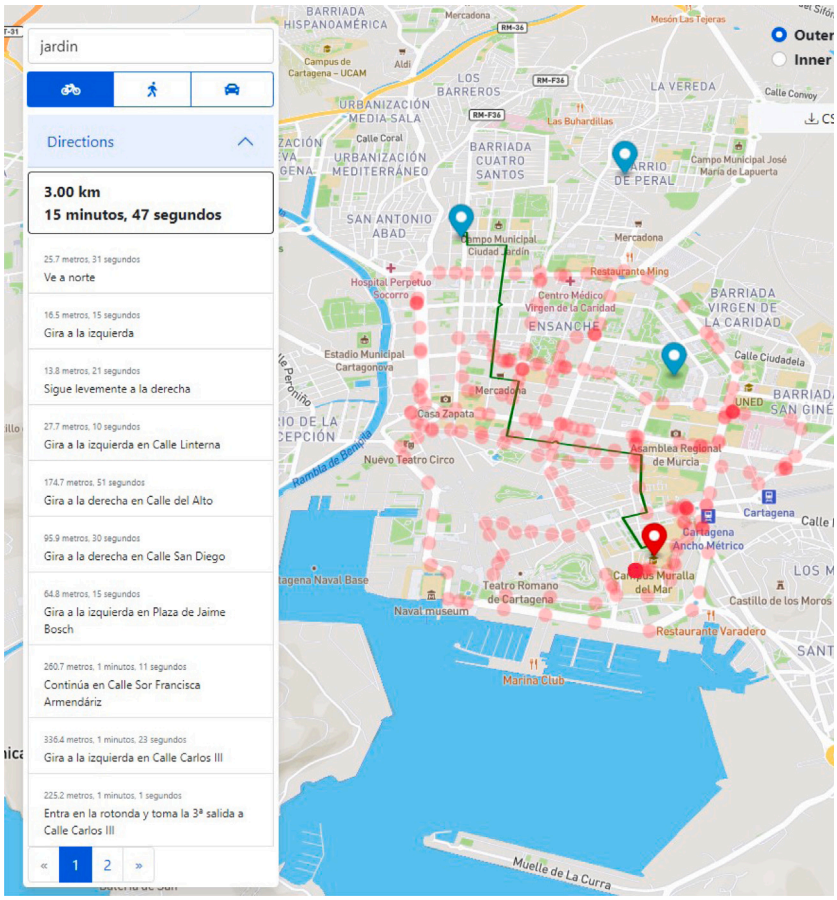


Fig. 9. Navigation view of the Web application.

A button in the Web application interface allows to view all individual geolocated measurements in 2D, increasing colour intensity depending on spatial density of measurements. Moreover, it is possible to download all data for a selected time window in a CSV file, for independent analysis if desired.

The appearance of the 2D view of the Web application can be seen in Fig. 9. Here, the navigation capabilities are also enabled for a bike route from the university campus to another part in the city. All data collected during tests are also included here, with increased colour intensity at points where more data records were collected.

Analysing the worst air quality values in this drive session and the location where they were sampled, they belong to the main avenues of the city that support high traffic density and when the car was stopped at red traffic lights. Also, narrow streets report higher pollution due to low air flow.

6.5. Evaluation of the ML-based prediction subsystem

After numerous tests, it has been verified that the pre-processing and prediction capabilities allow forecasts to be made with acceptable precision at a reasonable computational cost. It has been checked that the global data processing software is able to update these predictions when new data are available and under conditions of multiple users requiring predictions from different areas. Fig. 10 includes an example of prediction carried out for CO at a particular cell in the grid. Previous data records are plotted in red, while forecasted values are added in blue.

Although it is not the main aim of this work, we have analysed the operation of the prediction subsystem in more detail, through a set of 10 weeks of data. They have been taken from long-term air-quality monitoring observations from a nearby weather station of the city administration. This station is located on the main avenue of the city, which connects the entrance/exit highway with the city centre. The 10 weeks are selected in different seasonal circumstances. The algorithm used is trained for six days, and then a one-day prediction is carried out.

A first visual inspection of the prediction algorithm is included for one of the weeks in Fig. 11, including a reference comparison between real data and the prediction carried out for a whole day. The figure shows data reported in a whole week for the CO pollutant, measured in parts per million (ppm). Since CO is closely related to combustion vehicle traffic, it can be seen that the

Fig. 10. Web application when performing a data prediction for CO.

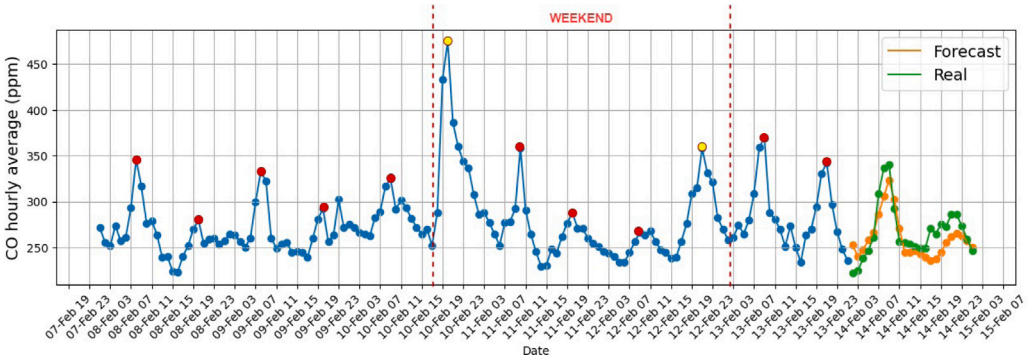


Fig. 11. Comparison between real and predicted CO values.

general daily trend is high CO concentration at the start of the working day (~8:00 a.m.) and a slightly lower peak at the end of the working day (~8:00 p.m.), as this time is somewhat more dispersed. This trend is slightly different during the weekend, due to lower traffic density. A large peak is observed at a period of weekend departure and another during the return. Both are marked in yellow in Fig. 11. During Monday the 13th, the typical trend explained was more evident due to a nearby carnival parade for which several streets were closed and that forced part of the traffic to be diverted from other areas to this avenue. Despite these divergences from the general trend, it can be seen that the algorithm used makes a fairly accurate prediction, considering such a short period. In addition, it has a low computational load, with an execution time of only 0.2 s on a commodity computer for this one-week time window.

The quality of the prediction can be calculated analytically with the SMAPE (Symmetric Mean Absolute Percentage Error), as indicated in (1), being F the forecast value and A the actual value. For the dataset of 10 weeks (ten different executions), an average SMAPE of 8.9% has been obtained, with a minimum of 2.95% and a maximum of 20.24%. It must be born in mind that these results were obtained for an algorithm addressed to provide a prompt reply in a Web application, that uses a limited time range for learning. This error could be considerably reduced as we increase the data size of the training time series, and/or selecting a more complex

algorithm, such as those used in deep learning. This is a task to be covered in future work.

$$\frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(F_t + A_t)} \tag{1}$$

7. Conclusions

The paper presents a vehicular-based environmental monitoring system, powered by distributed and edge computing capabilities. Both in-cabin and outdoor air condition samples are collected and processed in a multi-tier fashion, timely providing dangerous conditions and performing localised and global data analytics through microservices. Especial on-board sensing units have been developed for indoor and outdoor data collection. The data and processing path involves an in-vehicle WiFi network interconnected by common 4G communications and LoRaWAN, hence, extending smart city deployments to also gather sensor data from mobile nodes. Main environmental figures of merit are accessible through a data visualisation tool for technicians and a Web-based application with 3D interface that embeds extra functionalities such as healthy route guidance. Forecasting capabilities are added to the system by exploiting ML and time series analysis for selected geographical areas.

The whole solution has been evaluated in a real target scenario, validating all functionalities and extracting most important performance indicators, including the behaviour of the wireless communication links. By using a single LoRaWAN gateway, only 15% of packet were lost, while no losses were reported in the 4G link. The units operated correctly and all data collected in the testing campaign have been used to show the monitoring capabilities of the platform, being able to inspect and process main pollution factors in both the network edge and the cloud. The operation of the forecasting subsystem has been particularly evaluated in the last case, taking a reference set of data to obtain mean deviations between real and predicted values of 9%.

While there is a plethora of areas subject to future work in such a comprehensive solution, we plan the development of a simplified version of the OBUs for massive data collection that support our next steps on intelligent data analytics for environmental monitoring. Taking advantage of a growing number of sensor units, we will be able to evaluate the scaling capabilities of our system when extra communication and processing resources are needed. Also, we are working on more advanced ML solutions for both pollution and mobility prediction. Our target is to integrate new prediction algorithms that allow short and medium-term planning for decision support in terms of health assurance, detection of new infrastructures, and anticipate urban mobility issues.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jose Santa reports financial support was provided by Spain Ministry of Science and Innovation. Jose Santa reports financial support was provided by European Commission.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the grants PID2020-112675RBC41 (ONOFRE- 3), funded by MCIN/AEI/10.13039/501100011033; RYC-2017-23823, funded by MCIN/AEI/10.13039/501100011033 and by “ESF Investing in your future”; CNS2022-136150 (WILLIOT), funded by MCIN/AEI/10.13039/501100011033 and by “European Union NextGenerationEU/PRTR”; and H2020 957258 (ASSIST-IoT), funded by the European Commission.

References

[1] M.R. Jury, M.S. Buthelezi, Air pollution dispersion over Durban, South Africa, Atmosphere 13 (5) (2022) http://dx.doi.org/10.3390/atmos13050811, URL https://www.mdpi.com/2073-4433/13/5/811.

[2] L. Ehrnsperger, O. Klemm, Air pollution in an urban street canyon: Novel insights from highly resolved traffic information and meteorology, Atmos. Environ. X 13 (2022) 100151, http://dx.doi.org/10.1016/j.aeaoa.2022.100151, URL https://www.sciencedirect.com/science/article/pii/S2590162122000053.

[3] L. Khazini, M.J. Kalajahi, Y. Rashidi, S.M.M.M. Ghomi, Real-world and bottom-up methodology for emission inventory development and scenario design in medium-sized cities, J. Environ. Sci. 127 (2023) 114–132, http://dx.doi.org/10.1016/j.jes.2022.02.035, URL https://www.sciencedirect.com/science/article/pii/S1001074222001024.

[4] P. Vijayakumar, A. Khokhar, A. Pal, M. Dhawan, Air quality index monitoring and mapping using UAV, in: 2020 International Conference on Communication and Signal Processing, ICCSP, 2020, pp. 1176–1179, http://dx.doi.org/10.1109/ICCSP48568.2020.9182374.

[5] Y.-C. Wang, G.-W. Chen, Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks, IEEE Trans. Veh. Technol. 66 (8) (2017) 7234–7248, http://dx.doi.org/10.1109/TVT.2017.2655084.

[6] S. Kaivonen, E.C.-H. Ngai, Real-time air pollution monitoring with sensors on city bus, Digit. Commun. Netw. 6 (1) (2020) 23–30, http://dx.doi.org/10.1016/j.dcan.2019.03.003, URL https://www.sciencedirect.com/science/article/pii/S2352864818302475.

[7] P.A. Solomon, D. Vallano, M. Lunden, B. LaFranchi, C.L. Blanchard, S.L. Shaw, Mobile-platform measurement of air pollutant concentrations in California: performance assessment, statistical methods for evaluating spatial variations, and spatial representativeness, Atmos. Meas. Tech. 13 (6) (2020) 3277–3301, http://dx.doi.org/10.5194/amt-13-3277-2020, URL https://amt.copernicus.org/articles/13/3277/2020/.

[8] P.M. Santos, J.G.P. Rodrigues, S.B. Cruz, T. Lourenço, P.M. d'Orey, Y. Luis, C.l. Rocha, S. Sousa, S. Crisóstomo, C. Queirós, S. Sargento, A. Aguiar, J. Barros, PortoLivingLab: An IoT-based sensing platform for smart cities, IEEE Internet Things J. 5 (2) (2018) 523–532, http://dx.doi.org/10.1109/JIOT.2018.2791522.

[9] O. Briante, C. Campolo, A. Iera, A. Molinaro, S.Y. Paratore, G. Ruggeri, Supporting augmented floating car data through smartphone-based crowdsensing, Veh. Commun. 1 (4) (2014) 181–196, http://dx.doi.org/10.1016/j.vehcom.2014.08.002, URL https://www.sciencedirect.com/science/article/pii/S2214209614000412.

[10] M. Silva, G. Signoretti, J. Oliveira, I. Silva, D.G. Costa, A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective, Fut. Internet 11 (1) (2019) http://dx.doi.org/10.3390/fi11010013, URL https://www.mdpi.com/1999-5903/11/1/13.

[11] J. Huang, N. Duan, P. Ji, C. Ma, f. hu, Y. Ding, Y. Yu, Q. Zhou, W. Sun, A crowdsource-based sensing system for monitoring fine-grained air quality in urban environments, IEEE Internet Things J. 6 (2) (2019) 3240–3247, http://dx.doi.org/10.1109/JIOT.2018.2881240.

[12] M. Barros, A. Almeida, P. Santana, J. Monge, Environmental pollution monitoring based on sensor network and open-software-open-hardware, in: 2020 International Conference and Exposition on Electrical and Power Engineering, EPE, 2020, pp. 564–569, http://dx.doi.org/10.1109/EPE50722.2020.9305639.

[13] K. Yeom, Development of urban air monitoring with high spatial resolution using mobile vehicle sensors, Environ. Monit. Assess. 193 (6) (2021/06/01) 375, http://dx.doi.org/10.1007/s10661-021-09139-2.

[14] D. Zhang, S.S. Woo, Real time localized air quality monitoring and prediction through mobile and fixed IoT sensing network, IEEE Access 8 (2020) 89584–89594, http://dx.doi.org/10.1109/ACCESS.2020.2993547.

[15] E.I. Tolis, T. Karanotas, G. Svolakis, G. Panaras, J.G. Bartzis, Air quality in cabin environment of different passenger cars: effect of car usage, fuel type and ventilation/infiltration conditions, Environ. Sci. Pollut. Res. 28 (37) (2021) 51232–51241, http://dx.doi.org/10.1007/s11356-021-14349-9.

[16] A.C. Targino, P. Krecl, Y.A. Cipoli, G.Y. Oukawa, D.A. Monroy, Bus commuter exposure and the impact of switching from diesel to biodiesel for routes of complex urban geometry, Environ. Pollut. 263 (2020) 114601, http://dx.doi.org/10.1016/j.envpol.2020.114601, URL https://www.sciencedirect.com/science/article/pii/S0269749120301627.

[17] B. Bos, S. Lim, M. Hedges, N. Molden, S. Boyle, D.I. Mudway, D.B. Barratt, Taxi drivers' exposure to black carbon and nitrogen dioxide in electric and diesel vehicles: A case study in London, Environ. Res. 195 (2021) 110736, http://dx.doi.org/10.1016/j.envres.2021.110736, URL https://www.sciencedirect.com/science/article/pii/S001393512100030X.

[18] C.C. Goh, L.M. Kamarudin, A. Zakaria, H. Nishizaki, N. Ramli, X. Mao, S.M.M. Syed Zakaria, E. Kanagaraj, A.S. Abdull Sukor, M.F. Elham, Real-time in-vehicle air quality monitoring system using machine learning prediction algorithm, Sensors 21 (15) (2021) http://dx.doi.org/10.3390/s21154956, URL https://www.mdpi.com/1424-8220/21/15/4956.

[19] P. Szmeja, A. Fornés-Leal, I. Lacalle, C.E. Palau, M. Ganzha, W. Pawłowski, M. Paprzycki, J. Schabbink, ASSIST-IoT: A modular implementation of a reference architecture for the next generation internet of things, Electronics 12 (4) (2023) http://dx.doi.org/10.3390/electronics12040854, URL https://www.mdpi.com/2079-9292/12/4/854.

[20] C. Guardiola, C. Vigild, F. De Smet, K. Schusteritz, From OBD to connected diagnostics: a game changer at fleet, vehicle and component level, IFAC-PapersOnLine 54 (10) (2021) 558–563, http://dx.doi.org/10.1016/j.ifacol.2021.10.221, 6th IFAC Conference on Engine Powertrain Control, Simulation and Modeling E-COSM 2021. URL https://www.sciencedirect.com/science/article/pii/S2405896321016232.

[21] CITEAIR, Common air quality index definition, 2022, URL http://www.airqualitynow.eu/about_indices_definition.php. (Accessed 20 April 2022).