IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 September 2023, accepted 14 November 2023, date of publication 15 December 2023,
date of current version 20 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343620

RESEARCH ARTICLE

Cooperative Multi-Agent Traffic Monitoring Can Reduce Camera Surveillance

DAVIDE ANDREA GUASTELLA[1] **AND EVANGELOS POURNARAS**[2]
[1]Machine Learning Group, Université Libre de Bruxelles, 1050 Brussels, Belgium
[2]School of Computing, University of Leeds, LS2 9JT Leeds, U.K.

Corresponding author: Davide Andrea Guastella (davide.andrea.guastella@ulb.be)

This work was supported by the Traffic prOcessing foR uRban EnvironmentS (TORRES), a Joint Research and Development Project, funded by ''Région de Bruxelles-Capitale-Innoviris'' under Grant 2022-RDIR-59b. The work of Evangelos Pournaras was supported in part by the UK Research and Innovation (UKRI) Future Leaders Fellowship: '*Digitally Assisted Collective Governance of Smart City Commons–ARTIO*' under Grant MR/W009560/1, and in part by the Alan Turing Fellowship.

ABSTRACT Smart mobility initiatives encompass innovative methods to support traffic management experts in decisions for how to improve urban infrastructures and reduce carbon footprint. Accurate and continuous information about traffic is necessary to implement effectively such decisions. This is not always possible because of the cost of the information: it is not possible to install sensor devices at large scale because of financial costs and privacy; employing a plethora of sensors requires significant computational capabilities to process the generated data. A centralized data analysis can hinder real-time applications, and limit their practical deployment in traffic management systems. This paper introduces a novel privacy-aware method for estimating traffic density using edge computing and without over-deploying privacy-intrusive surveillance technologies such as cameras. The objective is to reduce the cost of collecting data while providing accurate information to support traffic operators in decision making. We evaluate the proposed solution using a realistic traffic data of Bologna in Italy. Results shows that it yields a 45% lower average estimation error compared to standard prediction methods. Virtual traffic monitoring devices are associated with software agents that collect data from simulated traffic and estimate traffic density measurements when this information is not available. In our experiments, when we replace 50% of camera devices with cooperative low-cost edge devices, we obtain an average percentage error of just 22%. This result indicates that the cooperation between virtual traffic monitoring devices offers a means to avoid massive deployment of camera surveillance devices using low-cost information provided by connected vehicles. We also compared the results to those obtained by standard regression techniques.

INDEX TERMS Smart city, traffic monitoring, multi-agent systems, missing information estimation, Internet of Things, urban sensing.

I. INTRODUCTION

Smart Cities have emerged as a technological means to address problems of resource optimization, creating better living conditions, and safeguarding the environment [1]. Implementing the smart city requires monitoring the urban environment to extract and analyze information that can support the decision-making process to improve infrastructures and services for the citizens. One such application in smart cities is urban traffic monitoring. With accurate information about traffic status, urban traffic control operators can monitor, control and redirect traffic, and apply policies to reduce traffic congestion.

In this paper, the goal of traffic density estimation is to calculate traffic density measurements in points where traffic monitoring devices such as CCTVs are not available. Herein, we define the traffic density as the number of vehicles situated in a local region of the environment.

Traffic monitoring is often performed using solutions based on CCTV cameras. Camera-based methods require an extensive network of cameras to provide accurate information about traffic. However, massive instrumentation of the urban

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

environment with CCTV raises privacy issues [2], and also suffers from environmental interferences such as harsh weather and air pollution [3]. Processing several images acquired from sensors on a large scale requires significant computational power and a pertinent infrastructure for storing, processing, and transferring information. The edge computing paradigm allows reducing the load on the core network, especially when considering bandwidth-intensive applications such as traffic monitoring through camera devices [4].

This article introduces a novel edge-based method for estimating urban traffic density in local parts of a physical environment by aggregating data acquired from IoT-connected vehicles. The novelty of this paper is a traffic density estimation method based on **virtual sensors**, software components associated with low-cost edge devices. Virtual sensors estimate traffic density by aggregating historical information, and by cooperating with the available physical sensors. Virtual sensors cooperate with physical sensors by exchanging information that can help calculate accurate traffic density estimates in local parts of the environment.

To evaluate the proposed method, we simulate the presence of physical (CCTV) and virtual sensors in a traffic model of the city of Bologna. We use the open-source traffic simulator of SUMO to emulate the dynamics of the real traffic of Bologna. Using this traffic simulator software, we can evaluate the performance of the proposed method (that is, the accuracy of the traffic density estimates) in a realistic traffic context.

The contributions of this paper are summarized as follows, together with a brief description of the outcomes:

- A novel technique to pursue privacy-aware, geofenced estimation of traffic density in points of the environment where CCTV devices are not available. Thanks to a cooperative protocol between virtual sensors, we obtain accurate traffic density estimates that outperform state-of-the-art regression techniques;
- A method for identifying causal relationships between different traffic variables. We explore the associations between traffic density and other types of data to improve the accuracy of traffic density estimates;
- An experimental procedure to evaluate the accuracy of the estimation method in the context of an urban traffic simulation, and a cost-benefit analysis to assess both economic and privacy advantages of virtual sensors compared to CCTV cameras.

The remainder of this paper is organized as follows: Section II discusses state-of-the-art propositions to address traffic density estimation in urban environments. Section III introduces the challenges in urban traffic estimation that we address with our proposal. Section IV presents the key concepts used to introduce the proposed method. Section V introduces the proposed technique for estimating traffic density using virtual sensors. Section VI illustrates the experimental setup, the results obtained from the agent-based traffic density estimation method, and a cost-benefit analysis to motivate the use of our proposal to avoid the massive deployment of camera-based traffic monitoring devices. In Section VII we highlight some insight from the results and the proposed technique. In Section VIII, we conclude our work and point to future perspectives.

II. RELATED WORK

This section introduces the main state-of-the-art solutions for traffic monitoring and estimation. The work presented make use of camera-based technologies, information collected from cellular devices, sensors embedded in vehicles. We also report some work that implements traffic monitoring applications on an edge computing infrastructure.

Traffic monitoring is one of the most essential elements for the planning, management and control of transportation systems [5]. Existing methods for traffic monitoring rely on data collected from devices such as loop detectors and video cameras, processed by centralized computational infrastructures. If we consider integrating data from personal IoT-equipped vehicles, the volume of the data to process can increase exponentially [6]. In this case, computational methods that rely on centralized infrastructure are not pertinent because of the elevated latency introduced for data processing [7].

Balamuralidhar et al. [8] propose a real-time vehicle detection, tracking, and speed estimation system that can run on embedded computers mounted on an Unmanned Aerial Vehicle (UAV). The method employs a neural network for vehicle detection, an object tracker to maintain object identity across frame sequences, and a vehicle speed estimation algorithm. The authors use the YOLO object detection algorithm to detect vehicles in images collected from UAVs. Vehicle tracking is achieved using the Minimum Output Sum of Squared Error (MOSSE) algorithm. The authors consider two datasets of images collected at different altitudes to validate the proposed method. The first is the Aeroscapes dataset, including 3269 720p non-sequential aerial images collected from drones at altitudes ranging from 5 to 50 meters [9]. The second dataset includes 52 images captured at 30, 60, and 120-meter altitudes with a resolution of 5472×3648. The proposed system achieves vehicle detection and tracking at 29.41 Frames Per Second (FPS) for an input resolution of 512×320 pixels, with an accuracy of 88%. At a resolution of 3072×1728 pixels, the method processes frames at a rate of 3.74 FPS. This method has the following drawbacks: (*i*) it is not possible to continuously monitor the traffic status, due to battery limitation of UAVs; (*ii*) coordination of autonomous drones is required for monitoring a large environment without significant costs for hardware and specialized human operators [10], [11]; (*iii*) the use of images can again raise privacy issues and suffer from environmental interferences such as harsh weather or air pollution [12], [13].

Li et al. [14] propose a method to detect traffic level by aggregating data collected from cellular devices in a

large-scale urban environment. The data include the number of time a vehicle is detected in a mesh cellular network, the number of handoff events (a handoff event happens when a device moves from one cellular network area to another one), and the time of day. A support vector machine trained with the collected cellular data is used to detect the traffic status according to three classes: low, medium, and high traffic. For this, the authors calculate a relative threshold to class the top $m\%$ traffic volume as high-traffic volume, the last $m\%$ as low-traffic volume, the rest is considered as medium-traffic.

The authors evaluate the proposal in a simulated scenario, modeling the traffic during 24 hours, in the city of Taicang, China, using data acquired from loop detectors as a ground truth. The simulations were carried out using the VISSIM simulator. The proposed technique allows obtaining a level of accuracy in traffic status detection of over 83% compared to the ground truth data. Although the technique allows combining information acquired from personal devices to detect traffic status, the authors focus on a generic classification of traffic status and do not provide a quantitative measure of the traffic density.

Other studies such as [15] and [16] focus on the use of sensors embedded in vehicles to estimate traffic density measures.

Lee et al. [15] estimate traffic density using inter-vehicle distance measured by probe vehicles equipped with sensors. First, the authors calculate inter-vehicle distance values using two vehicles equipped with cameras and radar sensors. The inter-vehicle distance is defined as the distance from the front bumper of sensor-equipped vehicle to the rear bumper of target vehicle. The authors perform the data collection in an experimental road section that is not public. Then, the authors perform simulations (using the CORSIM traffic simulation model), to generate a synthetic data set of inter-vehicle distance measures. With the ground truth values and the synthetic inter-vehicle distance values, the authors calculate traffic density values according to various traffic conditions such as traffic congestion levels and ratios of probe vehicle equipped with camera and radar sensor. The main drawback of this method is related to privacy and environmental factors: the former is due to the use of images acquired from cameras mounted on personal vehicles; the latter is due to factors such as lightness, or fog, that can influence the estimation of inter-vehicle distance.

Nam et al. [16] propose a technique for estimating traffic density using data acquired from Connected and Autonomous Probes (CAPs) equipped with radar sensors to observe surrounding traffic conditions. The probe vehicle radar system can detect the surrounding vehicles at real-time, and trace their trajectories, used to calculate Vehicle Miles Traveled (VMT) and Vehicle Hours Traveled (VHT). VMT is defined as the product of traffic volume on a link and the length of the link. Similarly, VHT is defined as the total time traveled by all the vehicles on a road link in a specific time horizon.

To estimate traffic density, authors use Long Short-Term Memory (LSTM), a particular type of artificial neural network which provides short-term and long-term memory components. This type of neural network is typically used to handle sequential data, where the output of the network is supposed to be influenced by previous historical data patterns. LSTMs are composed of a stack of layers of neural networks (an input gate, an output gate, and a forget gate), and the weights of each layer are updated by backpropagation. The LSTM uses multiple features as input variables from CAPs to estimate traffic density at a given time instant. Such features include also VMT, VHT, vehicles speed, and the number of lanes. The authors evaluated the proposed method by calculating the Root Mean Squared Error (RMSE) of the traffic density on different sections of a road network. The LSTM is not a suitable solution for retaining long-term information: the forget gates tend to remove some patterns that are not used recurrently. This means that the latest information dynamics are considered anomalous, leading to inaccurate estimates of traffic density.

Chen et al. [17] propose a vehicle tracking system based on deep learning and deployed on edge computing nodes. The authors combine vehicle detection (YOLOv3) and vehicle tracking (Deep-SORT) and use a virtual detection line method to count the number of vehicles passing through the detection line. To optimize the computation and reach an almost instantaneous calculation of traffic measures, the authors deploy the technique in edge devices equipped with Nvidia Jetson TX2. For validation, the authors use the UA-DETRAC dataset, featuring 8250 labeled vehicles and 1.21 million target object frames. The dataset includes frames captured in Beijing and Tianjin, China, shot at a frequency of 25 FPS with a Canon EOS550D camera. The authors obtained an average accuracy of vehicle detection of about 94%. However, the accuracy is affected by the quality of the images: on the edge devices used by the authors, the average FPS is 7.5, therefore the technique cannot achieve real-time detection. The main drawback of this technique is related to the privacy. The authors acknowledge that offloading the computation to the edge can mitigate privacy issues since the information is not processed centrally. In this way, the risk of information leakage is reduced. Nevertheless, no solution has been identified to protect sensitive information. This might require additional computation, making the proposed technique unsuitable for real-time application.

Gia et al. [18] propose an edge-fog-IoT architecture for traffic management and monitoring applications. The proposed architecture consists of sensor devices, smart edge devices (either battery-powered or connected to a power source) whose goal is to collect, encrypt and remove noise from data before this is sent to fog gateways via LoRa, smart LoRa-based gateways with fog computing, cloud-based services, and end-user terminal applications. To evaluate the proposed architecture, the authors propose an edge-based traffic density estimation technique that uses

TABLE 1. Comparison of state-of-the-art approaches for heterogeneous data integration and estimation.

	Privacy-awareness	Operational costs	Hardware requirements	Edge-based method
Lee et al. (2022) [15]	−	−	−	−
Balamuralidhar et al. (2021) [8]	−	−	+	+
Chen et al. (2021) [17]	−	−	+	+
Gia et al. (2020) [18]	−	−	−	+
Li et al. (2020) [14]	−	+	+	−
Nam et al. (2020) [16]	−	−	−	−

real-time image processing of traffic camera feeds. The traffic estimation method is based on an object detection technique using a Haar feature-based cascade classifiers available in the public domain library OpenCV [19]. The authors compare the latency measured by using the proposed estimation technique on their edge architecture, and the latency obtained by using the algorithm available in the YOLO framework, on the same edge-computing architecture. Both frame loading and analysis latency measures are low using the proposed method (respectively 5ms and 144ms), showing that the proposed edge architecture is pertinent to avoid the computational bottlenecks of centralized method in traffic analysis applications. However, a large-scale use of this method can raise privacy issues due to the extensive use of image data.

Table 1 lists the illustrated methods along with their strengths and weaknesses according to the following:

- **privacy-awareness**: the technique ensures that data is being anonymized before their use (that is, it is not possible in any way to trace the identity of citizens);
- **operational costs**: the technique requires limited computational capabilities and network hardware to operate;
- **hardware requirements**: the technique requires low-cost hardware to collect and process data from the urban environment and can integrate existing monitoring infrastructures;
- **edge-based method**: the technique can be deployed on an edge-computing infrastructure.

We use two indicators to depict the strengths and weaknesses of each described method: (+) the authors discuss and address the challenge; (−) the authors do not mention (nor address) the challenge.

Compared to the methods listed in Table 1, our goal is to address the four challenges described above.

III. CHALLENGES

This section discusses the main challenges that motivate our proposal.

A. LOCAL, PRIVACY-AWARE, DISTRIBUTED, GEOFENCED ESTIMATION OF TRAFFIC DENSITY USING HETEROGENEOUS INFORMATION FROM IoT-CONNECTED VEHICLES

Challenge: The volume of data generated by personal IoT continues to grow, causing the cloud to become significantly overburdened. The traffic estimation may encounter latency issues due to the amount of data to process for providing real-time traffic information. The massive amount of information collected from an urban environment could cause network bandwidth to be overburdened, as well as place a strain on the data center [20].

We propose an edge-based method to decentralize the traffic density estimation task and reduce the computational burden of centralized data analysis techniques. The advantage of our proposal is that citizens become the primary actor of a traffic monitoring system, as they decide when to share the information required for the estimation task. In this way, we tackle the privacy issue generally related with the use of camera-based devices.

B. FINDING CAUSALITIES IN TRAFFIC AND ENVIRONMENTAL VARIABLES

Challenge: the goal is to find causal relationships between different types of information from the environment, and that provided by IoT-connected vehicles.

Finding causal relationships between human activity and information collected from the environment is crucial for traffic management experts to improve urban infrastructures and ensure good quality of life for citizens. For instance, verifying a causal relationship between traffic levels and noise pollution can help determine if it is necessary to install noise barriers to provide sound insulation.

Our proposal enables finding relationships between different types of information collected from the environment.

C. INTEGRATING LOW-COST IOT DEVICES INTO EXISTING TRAFFIC MONITORING INFRASTRUCTURE TO REPLACE CAMERAS

Challenge: the goal is to integrate low-cost devices into an existing traffic monitoring infrastructure.

We assume that using low-cost IoT devices has a twofold advantage: first, to lower the management costs of a traffic monitoring infrastructure. Second, to avoid using "*heavy*" and privacy-sensitive information such as images, requiring significant computational power to perform traffic analysis at a large scale.

IV. KEY CONCEPTS

Table 2 summarizes the symbols, and provides a brief description, of the main concepts used to introduce the proposed method.

Figure 1 shows the relations between traffic density, traffic state, data window, and knowledge base, introduced afterwards.

In the following, we summarize the key concepts that we use to introduce our proposal in the next section.

A. VEHICLE AGENT

This type of agent is related to any physical instrumentation, such as IoT-connected vehicles, that can provide information such as GPS coordinates, CO_2 emissions, or vehicle speed.

TABLE 2. Table of symbols.

Symbol	Description
A_i	i-th sensing agent
$\lambda_{\phi}^{i,t,\ell}$	Value of the traffic variable ϕ calculated at time t by the i-th sensing agent and present in the ℓ-th data window
S_t	Traffic state assembled at time t
C_t^i	Data window created by A_i at time t
$d^{\phi}\left(C_t^i, C_k^i\right)$	Distance between data windows C_t^i and C_k^i of sensing agent A_i considering the traffic variable ϕ
$r_\ell \in R$	ℓ-th Voronoi region. R is the set of all Voronoi polygons
$\mathcal{K}^i$	Knowledge base of sensing agent A_i
$\mathcal{X}_{t-1}^{\tau,i}$	subset of the knowledge base $\mathcal{K}^i$ used to estimate a missing traffic density value
$w_t^{\tau,i}$	Estimation weight
v	Actual traffic density value
$\hat{v}$	Estimated traffic density value
$\overline{v}$	Expected traffic density value. It is the number of information received from IoT-connected vehicles during a specific time interval.

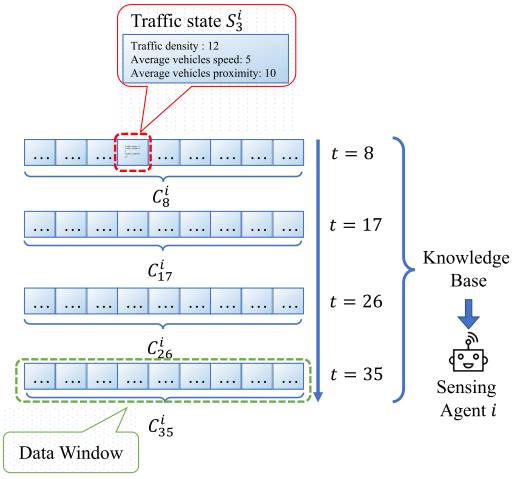

FIGURE 1. Relationships between traffic state, data window, knowledge base and sensing agent. The values t depict the time instants at which each data windows is added into the knowledge base of the sensing agent.

B. SENSING AGENT

A sensing agent A_i is an autonomous computation unit that can run on edge devices for pursuing low-latency transmission and computation of data to achieve real-time traffic density measurements. Edge devices can range from low-power Raspberry Pi[1] modules to custom Arduino devices,[2] to more computationally powerful nodes (such as the NVIDIA Jetson Nano[3]), capable of collecting, processing data from IoT-connected vehicles and exchange information to other nodes within the network. A sensing agent can collect and aggregate data from physical sensors at an aggregation frequency f_i to provide traffic density estimates. The aggregation frequency f_i, measured in seconds, defines the **sensing configuration** of a sensing agent. A sensing agent also aggregates data to provide a numeric value for each traffic variable $\lambda \in \Lambda_i$, where Λ_i is a set of traffic variables.

[1] www.raspberrypi.com
[2] www.arduino.cc
[3] https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/

C. TRAFFIC VARIABLE

A traffic variable λ is a data type whose value can be calculated by aggregating data collected by IoT-connected vehicles. In this study, we consider the following traffic variables: traffic density (λ_τ), average estimation delay (λ_θ), average vehicles proximity (λ_ϕ), average vehicles' speed (λ_σ), average CO_2 emissions (λ_χ). We provide details about these variables and how to calculate them in the following section.

D. TRAFFIC STATE

A traffic state (or simply **state**) S_t is a container of traffic variables, created at time t, that aggregates information collected during the time horizon $[t - f_i, t]$, where f_i is the aggregation frequency of agent A_i. A traffic state represents a snapshot of the environment state at time t.

E. VORONOI REGION

Given a set of point P, a Voronoi region [21] r is a convex polygon that includes all points closer to $p_i \in P$ than to any other point $p_k \in P, k \neq i$. Each point p_i depicts the position of sensing agent A_i. In typical cellular networks it is necessary to choose precisely the location of the antennas to obtain a mesh of regular hexagonal-shaped areas of the same size. However, it is not always possible to install an antenna at any point because of physical or environmental constraints. Conversely, the use of Voronoi tessellation allows relaxing this constraint: virtual sensors can be installed at arbitrary points in the environment where antenna cannot be placed. Figure 6 (Section V) shows how a Voronoi tessellation looks like.

F. DATA WINDOW

A data window C_t^i is a vector of traffic states representing the traffic dynamics in a discrete time interval $[t - n \cdot f_i, t]$, where n is the number of traffic states in C_t^i, and f_i is the aggregation frequency of sensing agent A_i. The data window C_t^i is associated with an index t, that corresponds to the time instance when the window has been assembled by sensing agent A_i.

Data windows can be seen as two-dimensional arrays of size $n \times k$, where n is the number of traffic states within a data window and k is the number of several types of information, including the density of the traffic (the other types of information are described later). In other words, each row (or slice) of the matrix contains values that refer to a particular type of information.

G. DISTANCE BETWEEN DATA WINDOWS

The distance between two data windows C_t and C_k is defined as the absolute difference of the values of the two windows, divided by the number of states $m = |C_t| = |C_k|$. The smaller the difference, the more similar the two windows are.

The distance between two windows C_t and C_k, considering an arbitrary traffic variable λ, is defined by the following formula:

$$d^{\lambda}\left(C_t^i, C_k^i\right) = \frac{\sum_{\ell \in [1,m]} |\lambda^{i,t,\ell} - \lambda^{i,k,\ell}|}{m} \quad (1)$$

where i is the index of the agent A_i, $\lambda \in \Lambda_i$ is the traffic variable used to compare the values in the two windows C_t^i and C_k^i. The values $\lambda^{i,t,\ell}$ and $\lambda^{i,k,\ell}$ are the values of the traffic variable (λ) in the data windows C_t and C_k respectively, $m = |C_t^i| = |C_k^i|$ is the number of states, ℓ is an index in the range $[1, m]$.

When a sensing agent compares a data window to another one considering a specific traffic variable, it selects the row of the matrix, in the data windows, referring to the same traffic variable. Then, the agent compares the values of the two slices using the defined distance metric d.

H. KNOWLEDGE BASE

A knowledge base $\mathcal{K}^i$ is a set of data windows assembled by the sensing agent A_i. A knowledge base is created by the sensing agent with which it is associated.

V. PROPOSED METHOD

We propose a method to estimate traffic density measures by aggregating historical data and information from IoT-connected vehicles. We use the multi-agent approach of HybridIoT [22] to estimate traffic density values. This approach allows estimating missing information through cooperation between sensing agents. Among the advantages, HybridIoT enables (*i*) introducing new sensing agents at run-time without reconfiguring the technique and (*ii*) estimating heterogeneous information without configurations dependent on the data type.

The novelty of our work is as follows:

- a method that can be deployed in an edge infrastructure to provide traffic analysis capabilities in local parts of the environment;
- a method to analyze causal relationships between traffic data to extract insights on traffic and environmental dynamics.

The proposed technique achieves the following properties:

- **generic**: our proposal can be implemented in different urban contexts without any prior knowledge about either the traffic dynamics or the route network topology;
- **unsupervised**: the method does not require human intervention to operate;
- **distributed**: the analysis and estimation of traffic measures are carried out autonomously by the sensing agents in their local part of the environment (their Voronoi regions).

Figure 2 sketches the functioning of the proposed method. The arrows indicate the direction of data flow. Data windows are created as a result of local traffic density estimation. Following their creation, the data windows are added to the knowledge bases of the sensing agents and used for the subsequent estimation of the traffic density.

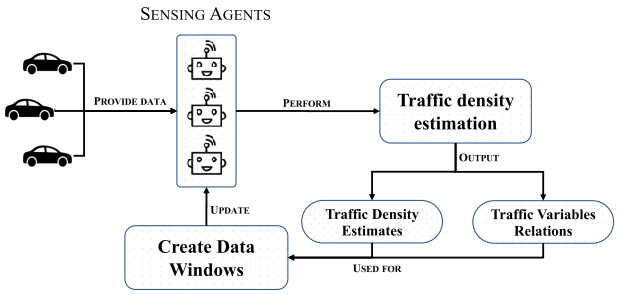

FIGURE 2. Sketch of the functioning of the proposed method. Traffic density is estimated by aggregating data from IoT-connected vehicles.

The estimation procedure is executed autonomously from each agentified device (that is, a device associated with a sensing agent); this makes it possible to deploy our proposal on the edge to address low latency and support real-time traffic monitoring.

Before introducing the estimation method, we distinguish between the expected ($\overline{v}$) and actual traffic density value (v): the **expected traffic density** is the number of information provided by IoT-connected vehicles; their aggregation outputs an estimate of the local traffic density. If a vehicle sends more than one data record when situated in each Voronoi region, the amount of data from that vehicle is counted as one within one aggregation time period. We refer to this value as the *expected* traffic density because the vehicles that provide data may no longer be in the same Voronoi region when a sensing agent calculates the estimate ($\hat{v}$), or because the sensing agent does not receive information from the vehicles due to problems in network communication. Contrarily, the **actual traffic density value** at time t is the number of vehicles (IoT-connected and not) that are situated in the Voronoi region of a sensing agent at time t. The actual traffic density values are only used to assess the accuracy of the proposed estimation method. For the sake of description, we recall that **estimated traffic density** is a numerical quantity that approximates the expected number of vehicles that are located in the region of a sensing agent at a given time instant. The difference between the expected and estimated traffic density is that the latter is calculated by HybridIoT, using historical information and those obtained by cooperation between sensing agents.

A. TRAFFIC DENSITY ESTIMATION

Figure 3 sketches the main steps of the estimation technique. The computation is done by the sensing agents in their Voronoi region.

Let A_i be the sensing agent that must estimate a traffic density measure at time t. First, it chooses the top n most similar data windows to pursue the estimation process. For this example, we set *a priori* the value $n = 3$, so A_i uses these three data windows to estimate traffic density values. The estimation process can be divided into two phases: a

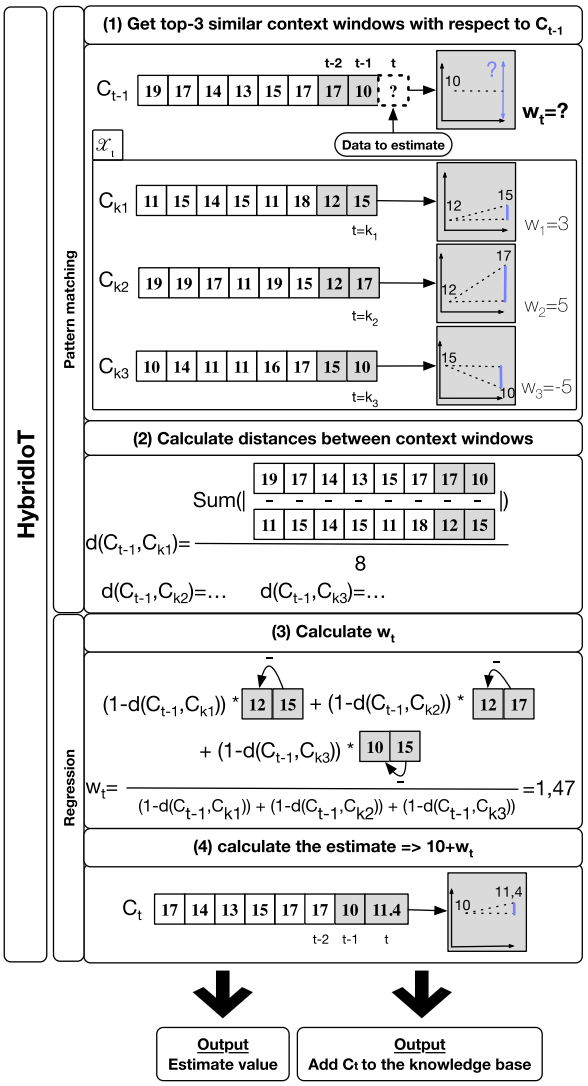

FIGURE 3. Example of data estimation using HybridIoT. For the sake of simplicity, the data windows (observed in distinct time instances) are shown as sequences of traffic density values. Also, we omit the index of the sensing agent, the type of variable (here we use only traffic density data), and w_t is the estimation weight.

pattern matching phase (steps (1)-(2)) and a **regression phase** (steps(3)-(4)).

Let τ be the traffic density variable. In the former phase (step (1)-(2)), A_i evaluates a set $\mathcal{X}_{t-1}^{\tau,i}$ of data windows, defined as follows:

$$\mathcal{X}_{t-1}^{\tau,i} = \{C_k^i \in \mathcal{K}^i : d^\tau(C_{t-1}^i, C_k^i) < th\} \tag{2}$$

where $\mathcal{K}^i$ is the knowledge base of sensing agent A_i. The data windows in $\mathcal{X}_{t-1}^{\tau,i}$ are sorted by their distance $d^\tau(C_{t-1}^i, C_k^i)$ (Equation 1), considering the values of traffic density (τ) in the data windows, th is the distance between C_{t-1} and the k-th data window in $\mathcal{K}^i$. The value th depends on the size of $\mathcal{X}_{t-1}^{\tau,i}$; for the sake of example, if we configure sensing agents so that the estimation is pursued using the top-3 similar data windows, then th is the value that allows selecting the top-3 most similar data windows. Herein, the number and the size of data windows, required to estimate traffic density values, are determined *a priori*. The size of the data windows must be chosen according to the resolution of the data perceived by the edge devices. In future works, we will consider using hyperparameter tuning techniques to evaluate the most pertinent value for this variable. On the one hand, if a device collects information at a low frequency and the size of data windows is small, then the estimation process will not be effective. In this case, the data windows do not present distinctive patterns that would enable the sensing agents to discriminate windows, and thus environmental dynamics, to provide accurate traffic estimates. A solution to this issue is to increase the size of the data windows. This is generally related to **underfitting**, as the agents cannot assess the dominant trend within the data, resulting in poor performance of the estimation method. On the other hand, if the size of the data windows is too high, then the data windows contain too specific information. The sensing agents are unable to provide accurate estimates based on the information they have. This is generally related to **overfitting**, as the data windows contain too many features, making agents unable to generalize well to new data.

The number of data windows to use for estimating missing information is a parameter that depends on the traffic dynamics. Because of the unpredictability of the traffic, it is not possible to know in advance how many windows are needed to provide, at a specific time instance, an accurate traffic density estimate: too many windows may introduce noise into the traffic estimate, few windows may not suffice to provide an accurate estimate.

The regression phase follows (steps (3)-(4)). In step (3), the sensing agent A_i calculates an estimation weight $w_t^{\tau,i}$. The idea behind the calculation of $w_t^{\tau,i}$ is the following: the data windows in the set $\mathcal{X}_{t-1}^{\tau,i}$ are similar to C_{t-1}^i (the similarity is calculated using the Equation (1)). Consequently, as the dynamics of the windows are similar, we can argue that the slope between the last two pieces of information in each data window in $\mathcal{X}_{t-1}^{\tau,i}$ should be similar to the slope between the data at $t-1$ and t, that is, the slope between the last perceived value at $t-1$ and the value to estimate at t. The value $w_t^{\tau,i}$ represents the slope between the data at $t-1$ and t, and is calculated as follows:

$$w_t^{\tau,i} = \frac{\sum_{C_k^i \in \mathcal{X}_{t-1}^{\tau,i}} \left(\left(\lambda_\tau^{i,k,\ell} - \lambda_\tau^{i,k,\ell-1}\right) \cdot \left(1 - d^\tau(C_{t-1}^i, C_k^i)\right)\right)}{\sum_{C_k^i \in \mathcal{X}_{t-1}^{\tau,i}} \left(1 - d^\tau\left(C_{t-1}^i, C_k^i\right)\right)} \tag{3}$$

where t is the current time instant, τ indicates the traffic density variable, $\left(\lambda_\tau^{i,k,\ell} - \lambda_\tau^{i,k,\ell-1}\right)$ is the difference between the last two values of the data window C_k^i, referring to traffic density variable τ, $\ell = |C_k|$ is the index of the last traffic state in C_k. Before calculating $w_t^{\tau,i}$, the distance values calculated for all data windows are normalized in [0, 1].

Finally, the traffic density estimate $\hat{v}_t^\tau$ is calculated as follows:

$$\hat{v}_t^{\tau,i} = \lambda_\tau^{i,t-1,\ell} + w_t^{\tau,i} \tag{4}$$

where $\lambda_{\tau}^{i,t-1,\ell}$, with $\ell = |C_{t-1}|$, is the last traffic density value available in C_{t-1} and sensed by A_i, $w_t^{\tau,i}$ is the estimated slope between traffic density measure at $t-1$ and t. The symbol τ indicates that the data windows used to calculate the estimate (in $\mathcal{X}_{t-1}^{\tau,i}$) have been obtained by comparing the traffic density values in the data windows of the knowledge base $\mathcal{K}^i$.

The output of the estimation procedure is an **endogenous estimate**. We calculate traffic density estimates by using historical traffic density values in the knowledge base of the sensing agents. In other words, endogenous estimation outputs estimate measures using historical data of the same type and unit.

The sensing agent A_i adds the estimate $\hat{v}_t^{\tau}$ into a new traffic state S_t. Then, A_i uses the collected information from IoT-connected vehicles to provide a value for each supported traffic variable. These values are also added to the same traffic state S_t. Finally, the agent inserts the traffic state S_t into a temporary data window. If this has size m (the size being decided *a priori*), this is added into the knowledge base $\mathcal{K}^i$ of the agent. When so, the temporary data window is cleared for assembling the next data window.

1) COOPERATIVE ENDOGENOUS ESTIMATION

In endogenous estimation, a sensing agent chooses a set of data windows from its knowledge base and calculates one traffic density estimate at each time instance. Due to the possible presence of noise or errors in data windows, we cannot ensure that the estimate will be accurate. To achieve accuracy, sensing agents could estimate traffic density using different, disjoint sets of data windows. However, the number of data windows can grow significantly, making this solution unsuitable for a great quantity of traffic data. Another possibility is to use error correction mechanisms to improve the quality of the knowledge base. However, this can be a computationally intensive task because the number of data windows can grow significantly over time.

To tackle this issue, we have defined a simple cooperation model in which the burden of evaluating several combinations of data windows to estimate traffic density is distributed among the available sensing agents, resulting in an augmented quantity of data that can help to sense agents in providing accurate estimates. Let A_i be the agent that must estimate traffic density at t. The cooperative estimation works as follows: A_i provides the last observed data window C_{t-1}^i to the other available sensing agents. A cooperative agent A_k then estimates a traffic density value using C_{t-1}^i: first, it compares this to the data windows in its knowledge base, then calculates an estimate and provides this to A_i. Finally, A_i chooses the traffic density estimate (among those provided by the other sensing agents) that minimizes the distance from the expected traffic density value. Herein, agents use the expected value to discriminate the values received cooperatively from other agents. However, we assume that other methods can be used to choose or weigh the values received from agents (e.g., the weighted median of values considering the distance between sensors). In this final choice, A_i also considers its estimate, that is, the one calculated by the non-cooperative method described in the previous section.

In this work, agents do not use any specific criteria to determine the subset of sensing agents with which to cooperate: a sensing agent simply cooperates with all the other available agents. These criteria can be, for instance, the sensors' proximity, the network topology, or the similarity between observed traffic data.

2) COOPERATIVE EXOGENOUS ESTIMATION

In endogenous estimation, sensing agents use only information of the same type (traffic density). The cooperative endogenous approach exploit traffic density values from a multitude of agents to output accurate traffic estimates. Conversely, in exogenous estimation, sensing agents use different traffic variables as predictors for estimating traffic density. The benefit of the exogenous method is twofold: first, sensing agents can estimate traffic density using simultaneously information of different type (unit, scale); second, it is possible to analyze the correlation between heterogeneous information about traffic. For instance, analyze the traffic density as a function of the pollution or the noise level. This can be useful for traffic management experts to improve infrastructures and ensure good quality of life for citizens.

We consider the following traffic variables (in addition to traffic density):

- **Vehicles proximity**: the average of all pairwise distances between vehicles situated in a local region of a sensing agent. To calculate the value of this variable, we make use of the standard Euclidean distance. Herein, the vehicles' position is in XY coordinates on a Cartesian plane. Therefore, this information is measured in number of pixels.
- **Vehicles speed**: the average speed of vehicles in a local region of the environment. This information is measured in meters per second (*m/s*).
- **CO_2 emissions**: the average CO_2 emissions of vehicles. This information is measured in milligrams per second (*mg/s*).
- **Estimation delay**: the average of differences between the time instances when the sensing agent receives the information and the aggregation time. First, we calculate the difference $t - t_i$, where t is the time instant when the sensing agent estimates the traffic density, and t_i is the time instant when a vehicle provides a piece of information to the agent ($t_i < t$). If a vehicle sends multiple data when passing through the same region, the sensing agent considers only the last one received. The result is calculated as the average of all temporal differences. This information is measured in seconds (*s*).

Algorithm 1 shows the steps performed by a sensing agent A_i to estimate a traffic density measure at time t using the cooperative exogenous method. For the sake of clarity, we specify at each line the agent that pursues the operation.

Algorithm 1 Exogenous Cooperative Estimation (A_i)

Require: $\overline{v}_t^{\tau,i}$ ← expected traffic density at time t by the agent A_i

1: $[A_i]$ Let $S \leftarrow \emptyset$ be the set of estimates received cooperatively from each agent A_ℓ, with $\ell \neq i$
2: $[A_i]$ Provide C_{t-1}^i to the other sensing agents {—*cooperative estimation*—}
3: **for all** sensing agent A_ℓ, with $\ell \neq i$ **do**
4: **for all** traffic variable $\lambda \in \Lambda_\ell$ **do**
5: $[A_\ell]$ $\mathcal{X}_{t-1}^{\lambda,\ell} \subset \mathcal{K}^\ell \leftarrow$ set of data windows of A_ℓ that minimize the distance to C_{t-1}^i considering λ
6: $[A_\ell]$ $\hat{v}_t^{\lambda,\ell} \leftarrow$ estimate the traffic density using the data windows in $\mathcal{K}^\ell$
7: $[A_\ell]$ provide $\hat{v}_t^{\lambda,\ell}$ to A_i
8: $[A_i]$ $S \leftarrow S \cup \hat{v}_t^{\lambda,\ell}$
9: **end for**
10: **end for** {—*end of cooperative estimation*—}
11: $[A_i]$ choose $\hat{v}_t^{\lambda,k} \in S$ such that $\min_k \left(\left| \hat{v}_t^{\lambda,k} - \overline{v}_t^{\tau,i} \right| \right)$
12: **return** $\hat{v}_t^{\lambda,k}$

First, the sensing agent A_i creates an empty set S that is used to keep all traffic density estimates provided cooperatively by the other agents (line 1). Then, A_i provides the last observed data window C_{t-1}^i to the other available sensing agents (line 2). This step follows the cooperative exogenous estimation (lines 3- 10), where the cooperative agents provide A_i with a traffic density estimate calculated using exogenous traffic variables.

The cooperative sensing agent A_ℓ evaluates a set $\mathcal{X}_{t-1}^{\lambda,\ell}$ of data windows for each supported traffic variable λ (line 5). To obtain such a set of data windows, A_ℓ selects the slices of the data windows (in its knowledge base) that are related to the traffic variable λ, then compares the windows C_{t-1}^i by using the Equation 1. Then, the cooperating agent A_ℓ selects the data windows in its knowledge base that minimize the distance to C_{t-1}^i considering the variable λ. The cooperating agent A_ℓ calculates an estimate of the traffic density measure using the data windows in $\mathcal{X}_{t-1}^{\lambda,\ell}$ (line 6). At line 7, A_ℓ provides the traffic density estimate to the sensing agent A_i. The steps in lines 3- 10 are repeated for each sensing agent and each traffic variable.

Algorithm 1 results in a set S of traffic density values, each one provided by cooperative sensing agents. Each estimate is calculated using a unique set of data windows (one per sensing agent and traffic variable). Finally, A_i selects the estimate from the set S that minimizes the distance to the expected traffic density value $\overline{v}_t^{\tau,i}$ (line 11).

VI. EXPERIMENTAL RESULTS

This section presents the results obtained by using HybridIoT for estimating traffic density values by aggregating information from IoT-connected vehicles. This section is composed as follows: first, we present the traffic scenario used to model realistic traffic dynamics via the SUMO simulator, then we present the results obtained by the HybridIoT technique. Finally, we present a cost-benefit analysis to assess the advantages of using sensing agents to avoid the deployment of a high number of camera-based devices.

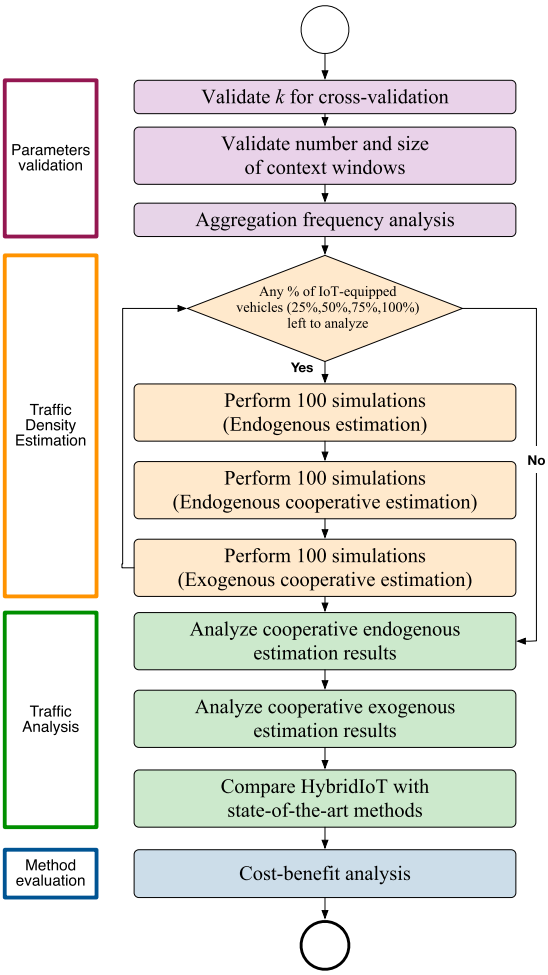

FIGURE 4. Experimental procedure used to evaluate our proposal.

Figure 4 shows the steps of the experimental procedure used to evaluate our proposal. The validation of k for cross-validation, the number and size of context windows, and the results of endogenous estimation are in Appendix A.

A. TRAFFIC MODEL

For this research, we simulated the traffic of the city of Bologna (Italy) using SUMO, an open-source traffic simulator [23]. SUMO allows simulating the dynamics of traffic, such as stop-and-go effects and traffic lights. The scenario is publicly available[4] and models the area around the "Andrea Costa" road in Bologna (Figure 5).

The available data is based on traffic measurements collected from induction loops. The city administration of Bologna provided the vehicle counts for three days

[4]https://github.com/DLR-TS/sumo-scenarios/tree/main/bologna/; Last visited: August 18, 2023

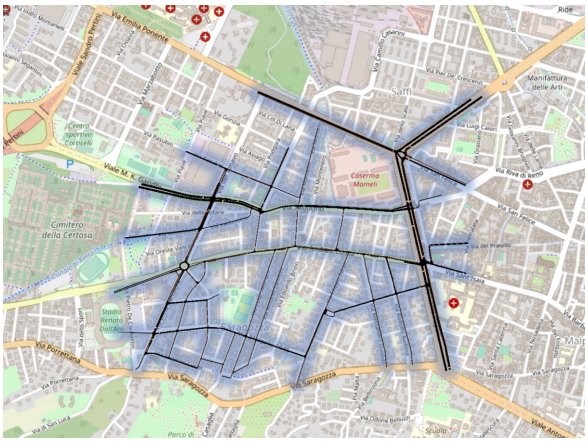

FIGURE 5. Street network of the bologna scenario (highlighted in blue) located at approximately 44.494554, 11.314393 (latitude/longitude) obtained through OpenStreetMap [24].

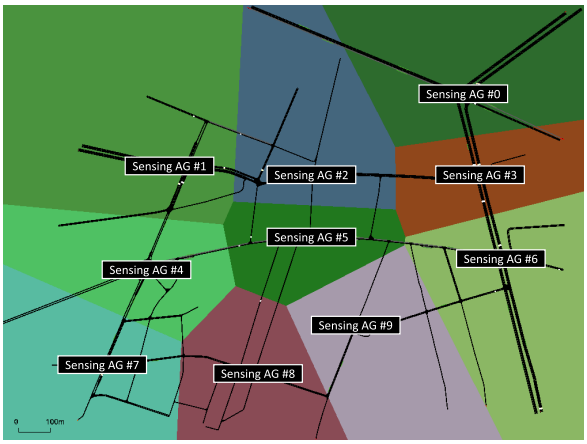

FIGURE 6. Placement of the sensing agents and the related Voronoi regions. Best seen in color.

(11 to 13 November 2008). In addition to vehicular traffic, the scenario includes public transport, bus stops, and special lanes. The authors use raw data from detectors, sensed hourly, to create a simulation that replicates the typical traffic patterns during the 8 AM–9 AM rush hour [25].

The scenario includes models of vehicle types such as hatchbacks, sedans, wagons, vans, and buses. The vehicles belong to CO_2 emission classes Euro 1 to Euro 4, according to the classification in the Handbook Emission Factors for Road Transport (HBEFA) [26]. SUMO simulates vehicles' CO_2 emissions according to their class.

We run different simulations with SUMO on the scenario of Bologna. In each simulation, the traffic dynamics in SUMO are randomized according to the following parameters:

- Random departure offset: each vehicle is added in the simulation with a delay between 0 and 5 seconds with respect to the insertion time (defined by the scenario).
- Vehicle type distribution: when a new vehicle is inserted in the simulation, SUMO decides the type of the vehicle according to a probability p associated with each class;
- Speed deviation: the desired driving speed among the vehicle of a fleet;
- Reaction time: the time required by a driver to react to their surroundings in every simulation step;
- tau: measures the traffic safety level for the vehicle, in case a vehicle in front suddenly decelerates or brakes.

The values for the previous parameters are chosen randomly at the beginning of the simulation. Adding randomness in the traffic model enables evaluating the accuracy of the estimates in different and unpredictable traffic conditions.

We chose the location of the sensing agents in correspondence with the main intersections and roundabouts (Figure 6).

During a simulation, the IoT-connected vehicles send information to the nearest sensing agent. Then, the sensing agent aggregates the information collected to estimate the local traffic density.

To simulate the lack of data from IoT-connected vehicles (because of missing sensors, or citizens who do not share information) we specify, for each simulation in SUMO, the percentage of IoT-connected vehicles providing data to the sensing agents. For instance, if we use a percentage of 25%, then at each time instance only 25% of the vehicles in the simulation send information to the nearest sensing agent. We consider the following percentages of IoT-connected vehicles: 25%, 50%, 75%, and 100%. For each percentage, we run 100 simulations using SUMO. To assess that 100 simulations are sufficient to obtain statistically significant results, we calculate the cumulative mean of the traffic density estimation error among all simulations and for each sensing agent. The plots of the cumulative average traffic density estimation errors are available in Figure 20, 21 and 22 of the Appendix D. The cumulative mean estimation error converges to a stable mean after about 50 simulations, confirming that the chosen number of simulations is appropriate.

We use a time series cross-validation technique to evaluate the accuracy of the estimation results. Let $y = \{y_1, \ldots, y_n\}$ be a time series. In time series cross-validation, samples from y_1 to y_{t-1}, with $1 \leq t - 1 < n$ are used as a training set for estimating y_t. Contrarily to standard cross-validation, we do not consider the time series future at t. Since it is not possible to obtain a reliable forecast based on a small training set, the earliest observations are not considered as test sets [27].

To evaluate the accuracy of the results obtained by our proposal, we use the following error measures:

- **Absolute traffic density estimation error**: the Mean Absolute Error (MAE) is calculated as the difference between the estimate and the actual traffic density. This value represents the number of vehicles that a sensing agent misses to estimate. The lower the value, the better the result.
- **Traffic density estimation error percentage**: the Mean Absolute Percentage Error (MAPE), calculated as the difference between the estimate and actual traffic density, divided by the actual traffic density. This value

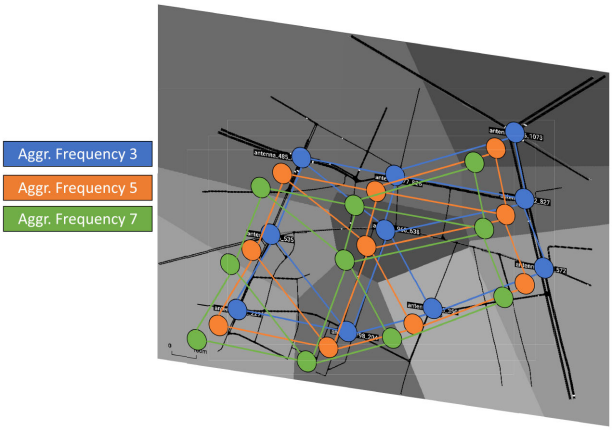

FIGURE 7. Sensing agents, represented as a vertex, cooperate with those that support the same frequency. For the sake of simplicity, not all the edges are shown. Best seen in color.

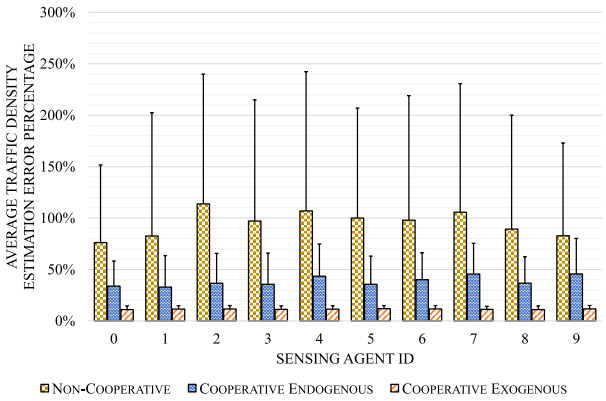

FIGURE 8. Comparison of average absolute traffic density estimation percentage error (MAPE) obtained from HybridIoT.

represents the percentage of vehicles that a sensing agent misses to estimate. The lower the percentage, the better the result.

B. AGGREGATION FREQUENCY ANALYSIS

In HybridIoT, it is possible to specify the aggregation frequency of each sensing agent. This parameter affects the estimation of traffic density, as the information collected (and estimated) by sensing agents has a different resolution depending on the aggregation frequency.

In cooperative estimation (both endogenous and exogenous), sensing agents cooperate only with others having the same aggregation frequency. By doing so, agents cooperate by exchanging data windows having information at the same resolution.

Figure 7 shows the cooperation relationships between agents. We represent each agent as a set of three vertices. Each vertex indicates a specific aggregation frequency of an agent. When an agent aggregates information at a frequency f, it can only cooperate with other agents having the same aggregation frequency f. In Figure 7, this cooperation mechanism is represented by the edges between vertices having the same color (same aggregation frequency). This results in different complete graphs, one for each aggregation frequency. We say that the graphs are complete because we assume that each sensing agent can cooperate with all the other available agents with the same frequency.

In Appendix E, we report the best aggregation frequencies for all the percentages of missing vehicles and estimation methods (non-cooperative, cooperative endogenous, and cooperative exogenous).

The best aggregation frequency is the one that yields the lowest estimation error (MAPE). The optimal value for the aggregation frequency depends on the location of the sensing agents and therefore the typical traffic flow in the observed regions.

A high aggregation frequency generally requires a significant amount of data windows that accurately describe the traffic dynamics. Contrarily, using low frequencies provides data windows that describe information less accurately because they lack fine-grained details about variations in traffic dynamics. Using low frequencies is reasonable when sensing agents do not have much information to calculate traffic estimates.

In the following sections, we present the results obtained by HybridIoT using the best aggregation frequency for each sensing agent. The best aggregation frequency for a sensing agent is the one that allows minimizing the traffic density estimation error.

C. TRAFFIC DENSITY ESTIMATION RESULTS

This section presents the results collected by using the HybridIoT for estimating traffic density measures. We compare the results to those obtained through state-of-the-art regression techniques.

The goal is to show that the cooperative method outputs accurate traffic density estimates, and the cooperative exogenous estimation allows for further improvement in the results over the cooperative endogenous method. For this, we report the results obtained from the cooperative estimation method, whereas the results obtained from the non-cooperative estimation are in Appendix C.

The key result of this work is shown in Figure 8, that compares the MAPE obtained from non-cooperative, cooperative endogenous, and cooperative exogenous estimation techniques.

The results in Figure 8 show that cooperation between sensing agents outputs accurate traffic density estimates. The cooperation provides a way for agents to tackle the lack of data in their knowledge base.

1) COOPERATIVE ENDOGENOUS TRAFFIC DENSITY ESTIMATION

Figure 9 shows the average estimation error (MAE) obtained from HybridIoT, using the cooperative endogenous estimation.

The average MAE is 3.95 vehicles, and the standard deviation 1.72.

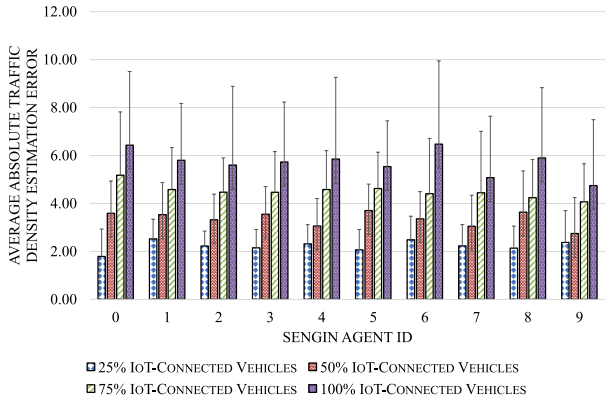

FIGURE 9. Cooperative endogenous estimation method assessment. Average absolute traffic density estimation error (MAE).

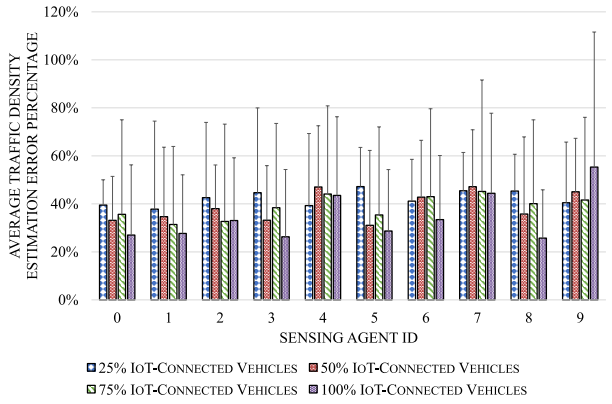

FIGURE 10. Cooperative endogenous estimation method assessment. Average absolute traffic density estimation percentage error (MAPE).

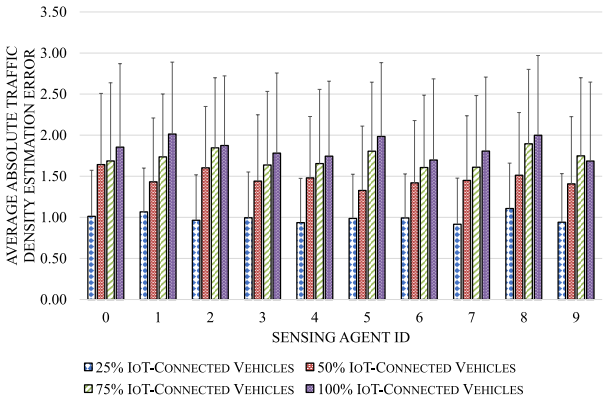

FIGURE 11. Cooperative exogenous estimation method assessment. Average absolute traffic density estimation error (MAE).

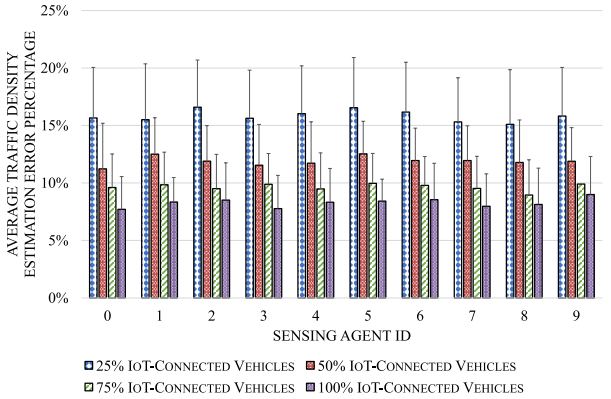

FIGURE 12. Cooperative exogenous estimation method assessment. Average traffic density estimation error percentage (MAPE).

The trend of the error varies almost uniformly because of the size of the sample studied. Using larger sample sizes, which means a higher percentage of connected vehicles, results in a more accurate representation of the population. However, this also results in a higher magnitude of the error.

Figure 10 shows the average estimation percentage error (MAPE) obtained by using the cooperative endogenous estimation.

We note that the MAPE follows a different trend compared to MAE. More specifically, we remark that the error does not increase linearly with the percentage of IoT-connected vehicles. When the number of connected vehicles increases, the information provided by sensing agents reflects more accurately the traffic dynamics, as the number of detected vehicles is closer to the actual one. Using this information in a cooperative estimation context enables agents to achieve a more precise assessment of the traffic state. This results in a lower percentage error, as shown in Figure 10.

2) COOPERATIVE EXOGENOUS TRAFFIC DENSITY ESTIMATION

In exogenous estimation, cooperative agents calculate traffic density estimates using different variables as predictors. In this study, we consider the average vehicle proximity, average estimation delay, average vehicle speed, and average CO_2 emissions. The estimated values calculated using these variables are provided to the agent that must estimate the traffic density. This agent outputs the traffic density value (among those received cooperatively) that minimizes the distance to the expected traffic density measure.

Figure 11 shows the estimation error (MAE) obtained by using the cooperative exogenous technique, Figure 12 shows the estimation accuracy (MAPE) obtained by using the exogenous estimation technique.

The cooperative exogenous estimation allows obtaining a lower estimation error compared to cooperative endogenous estimation (considering the MAPE, 39% using the cooperative endogenous method, 11% using the cooperative exogenous method). The magnitude of error in the MAE increases linearly with the number of connected vehicles. Conversely, the MAPE decreases as the number of connected vehicles increases. In this case, we argue that this is because the system is capable of providing accurate estimates even if there are a large number of IoT-connected vehicles, thanks to the use of heterogeneous data.

When using the exogenous estimation, a sensing agent assembles several sets of data windows, one per traffic variable. The agent selects the traffic variable associated with the set of data windows that allows for obtaining the most

TABLE 3. Cooperative exogenous estimation method assessment. Each cell contains the number of times a traffic variable is chosen to estimate traffic density. The values are normalized in the range [0,1].

	Normalized average number of traffic variable selection			
Percentage of IoT-connected vehicles	Average Vehicles Proximity	Average Vehicles Estimation Delay	Average Vehicles Speed	Average Vehicles CO2 emissions
100%	0.88	0.83	1.00	0.77
75%	0.87	0.76	1.00	0.77
50%	0.80	0.65	1.00	0.79
25%	0.76	0.88	1.00	0.79

accurate estimate. This is done each time a sensing agent must estimate a missing traffic density value.

Table 3 lists the number of times a traffic variable is chosen by the sensing agents. We average the values of all agents among the 100 simulations and the percentages of IoT-connected vehicles (25% to 100%). The results are normalized in [0, 1].

The results show that the most used variable as a predictor for estimating traffic density is the average vehicle speed, whereas the least used is the CO_2 emissions.

This information is relevant because it proves that the dynamics of the two pieces of information are related. Therefore, it is possible to explain the traffic density estimation error as a function of the average vehicle speed.

Table 4 shows the number of cooperative involvements between sensing agents. Each value in the table is calculated as follows. Let A_i be the agent to calculate the traffic density estimate. When A_i chooses the estimation value from an agent A_i, with $k \in [0, |A|]$, i excluded, then the value (i, k) in the table is incremented by 1.

The diagonal is zero because, in exogenous cooperative estimation, we assume that the estimate of the traffic density value is obtained only from the values of cooperative agents. By doing so, we avoid the selfish behavior of sensing agents that could lead these to choose only their data windows. Also, we assume that by using the values from cooperative agents it is easier to group agents that exhibit more similar dynamics.

3) COMPARISON TO STATE-OF-THE-ART

We compare the results obtained from HybridIoT to those obtained by state-of-the-art regression techniques.

We use different regressors available in the library Skforecast.[5] We use the default parameters for all the regressors. These parameters include the number of forward steps to estimate and the size of the training set.

Figure 13 compares the estimation error (MAE) obtained from the standard regression techniques in Skforecast and HybridIoT.

HybridIoT outperforms standard regression techniques for traffic density estimation. The cooperative endogenous method outperforms 2/3 of the standard techniques, while the exogenous estimation outperforms all methods. The exogenous estimation technique yields estimation errors that are on average 45% lower than the standard methods.

[5]https://joaquinamatrodrigo.github.io/skforecast/0.11.0/index.html; Last visited: December 12, 2023

TABLE 4. Cooperative exogenous estimation method assessment. Number of times the estimate of agent j is selected by agent i as the closest to the expected traffic density value observed by A_i.

	Number of cooperative involvements (100% IoT-connected vehicles)									
Sensing AG ID	0	1	2	3	4	5	6	7	8	9
0	0	47	59	59	34	78	42	52	**84**	45
1	**72**	0	42	58	53	59	41	57	69	49
2	57	62	0	50	47	68	41	47	**81**	47
3	71	59	46	0	35	56	47	54	**75**	57
4	48	**76**	61	55	0	51	44	52	75	38
5	60	59	46	67	60	0	30	57	**72**	49
6	49	58	52	53	62	59	0	43	**77**	47
7	43	52	49	61	56	**95**	49	0	58	37
8	53	63	51	61	31	70	55	**73**	0	43
9	40	57	52	35	49	80	42	62	**83**	0

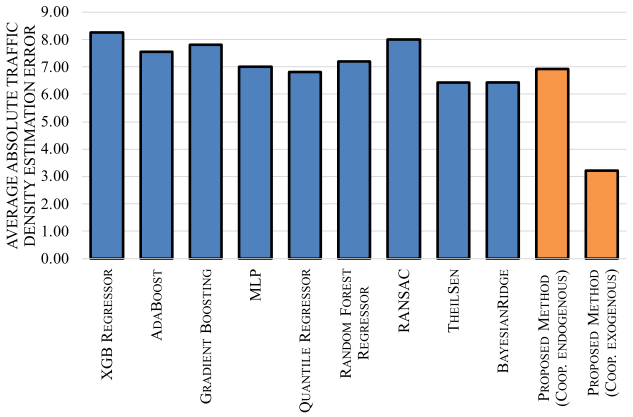

FIGURE 13. Average absolute traffic density estimation error (MAE) obtained from state-of-art regression techniques in Skforecast (in blue), and HybridIoT (in orange). The percentages indicate the percentage of connected vehicles sending information to sensing agents at each time instant.

Cooperation brings a benefit in estimation, as each sensing agent can access information that would miss otherwise. Moreover, the result obtained from cooperative exogenous estimation indicates that using different traffic variables can provide insight into traffic dynamics. This proves to be beneficial for providing accurate traffic density estimates.

D. COST-BENEFIT ANALYSIS

We refer to data acquired from traffic monitoring devices such as cameras as "expensive" mainly due to the computational cost (to extract traffic information from images) and privacy cost. A way to reduce the cost of camera-based devices is to use sensing devices that are privacy-preserving by design.

For this experiment, we configure some sensing agents to provide the actual traffic density, not the estimate. In this way, the sensing agents simulate the behavior of real traffic monitoring devices that provide accurate measurements.

We run 100 simulations in SUMO using a specific configuration (Table 5), and a cooperative estimation method (endogenous and exogenous). We refer to "system configuration" as the placement of both sensing agents and CCTVs in the environment. The output of this experiment is a quantitative indication of the stability of the estimation results obtained by the sensing agents in different configurations and environmental contexts.

TABLE 5. The parameters used for cost-benefit analysis.

% CCTV	90	80	70	60	50	40	30	20	10
% Sensing agents	10	20	30	40	50	60	70	80	90
% IoT-connected vehicles	25/50/75/100								

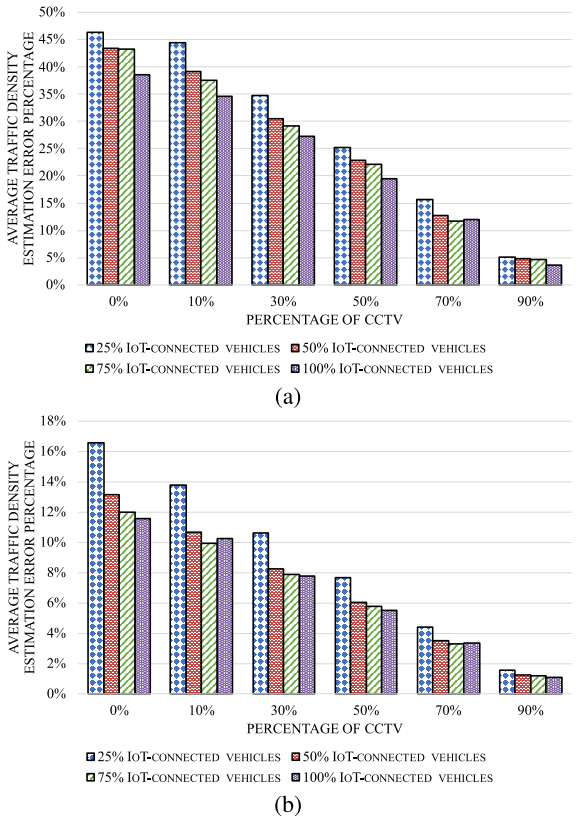

FIGURE 14. Cost-benefit analysis assessment. Traffic density estimation error percentage obtained from cooperative endogenous (14a) and exogenous (14b) estimation techniques, using different percentages of CCTVs (simulated) and sensing agents.

If a given percentage of sensing agents provides accurate estimates independently of the number of agents (either sensing agents or CCTVs) with which it cooperates, then we can argue that using a CCTV at that point of the environment is not necessary. This information can be used by traffic management experts to decide if it is suitable to sacrifice the accuracy of the traffic density measurement while ensuring both privacy and economic gain.

Each row of Table 5 identifies a group of four experiments, one per each percentage of IoT-connected vehicles (25-50-75-100). For each configuration, we perform 100 simulations using random traffic dynamics, as in previous experiments. We use both cooperative estimation methods (endogenous and exogenous).

Figure 14 shows the plot of the accuracy of estimates obtained from cooperative endogenous (Figure 14a) and exogenous estimation (Figure 14b), and considering different percentages of CCTVs.

The accuracy of the estimation method increases with the number of camera-based devices. The average percentage of error obtained through the endogenous estimation technique, when using 50% of sensing agents (and 50% of CCTV), is just 22%. This information proves the cost-effectiveness of the proposed method: the accuracy of traffic density estimation depends linearly on the amount of sensing agents in the system.

We consider the work by Matczak et al. [2] as a case study to assess the cost savings induced by using HybridIoT for traffic monitoring. The authors pursue a cost-effectiveness analysis of CCTV surveillance systems in eight Polish cities: Poznan, Gdansk, Katowice, Kielce, Lublin, Lodz, Poznan, Warsaw, and Wroclaw. They consider two categories of costs: the cost of installing cameras during the 2005–2014 period, the maintenance costs, and the personnel costs. The costs are aggregated. These are provided by the administrative bodies of the cities.

Table 6 lists the installation cost, the personnel cost, and the technical maintenance per camera. We consider, as a reference, the aggregated costs for the CCTV system in the city of Poznan over the period 2010-2012 [2]. We also report the hypothetical costs related to the use of different percentages of sensing agents. We say that these results are hypothetical because they are meant to show how much sensing agents, when integrated with an existing CCTV system, enable saving money on traffic monitoring infrastructure.

From these results, we can assess that using sensing agents can bring a twofold advantage to a traffic monitoring system. The first is economic because the use of low-cost devices associated with sensing agents allows municipalities to save on installation, maintenance, and personnel costs. Second, the use of sensing agents makes it possible to estimate traffic density information in a privacy-aware manner, ensuring accurate estimates even when a limited quantity of information is available from citizens' devices or IoT-connected vehicles. To determine the best percentage of sensing agents, it is necessary to determine a solution that minimizes the costs while maximizing the accuracy (Pareto optimal solution). This is not possible in this cost-benefit example, since we only have the costs and not the estimation accuracy.

The cost of sensing agents must be considered when implementing an *hybrid* traffic monitoring infrastructure (that is, including physical devices and sensing agents). Generally, commercial solutions like Arduino (about 30$ in December 2023, board only)[6] or Tulip (from 120$ to 600$ in December 2023)[7] can be employed to implement such an infrastructure.

VII. DISCUSSION

A key factor for the decisions of traffic operators is the availability of data from the urban environment. The use of CCTV-based monitoring infrastructures can be an obstacle in this regard, as cameras cannot be placed everywhere in an urban environment, consequently limiting the amount

[6]https://www.arduino.cc; Last visited: December 12, 2023
[7]https://tulip.co/products/edge-devices/; Last visited: December 12, 2023

TABLE 6. Yearly installation, personnel, and technical maintenance costs (in USD) per camera. We report the estimated economic savings due to the use of different percentages of sensing agents (from 10 to 90%). The baseline costs (0% sensing agents) are the baseline, taken from [2].

Year	% of Sensing Agents	Installation cost per camera	Personnel cost per camera	Technical maintenance cost per camera	Year	% of Sensing Agents	Installation cost per camera	Personnel cost per camera	Technical maintenance cost per camera
2010	0%	11.639	1898	516	2011	0%	11.344	2201	356
	10%	10.4751	1708.2	464.4		10%	10.2096	1980.9	320.4
	20%	9.3112	1518.4	412.8		20%	9.0752	1760.8	284.8
	30%	8.1473	1328.6	361.2		30%	7.9408	1540.7	249.2
	40%	6.9834	1138.8	309.6		40%	6.8064	1320.6	213.6
	50%	5.8195	949	258		50%	5.672	1100.5	178
	60%	4.6556	759.2	206.4		60%	4.5376	880.4	142.4
	70%	3.4917	569.4	154.8		70%	3.4032	660.3	106.8
	80%	2.3278	379.6	103.2		80%	2.2688	440.2	71.2
	90%	1.1639	189.8	51.6		90%	1.1344	220.1	35.6
2012	0%	10.876	1125	278	2013	0%	10.488	963	268
	10%	9.7884	1012.5	250.2		10%	9.4392	866.7	241.2
	20%	8.7008	900	222.4		20%	8.3904	770.4	214.4
	30%	7.6132	787.5	194.6		30%	7.3416	674.1	187.6
	40%	6.5256	675	166.8		40%	6.2928	577.8	160.8
	50%	5.438	562.5	139		50%	5.244	481.5	134
	60%	4.3504	450	111.2		60%	4.1952	385.2	107.2
	70%	3.2628	337.5	83.4		70%	3.1464	288.9	80.4
	80%	2.1752	225	55.6		80%	2.0976	192.6	53.6
	90%	1.0876	112.5	27.8		90%	1.0488	96.3	26.8

of data that can be collected from the environment. The use of data collected from IoT-connected vehicles may represent a trade-off between the high operating costs of camera-based infrastructure and privacy. However, citizens must become primary actors in the decision-making process and are involved in data collection activities: would they be willing to provide information such as location, commuting, and usual routes to avoid the installation of CCTV cameras in the city? We argue that answering this question requires an extensive analysis of acceptance by the citizens, as this can influence where to install sensing agents.

VIII. CONCLUSION AND FUTURE WORKS

This paper proposes a novel method for estimating urban traffic density. We use the HybridIoT estimation technique to pursue traffic estimation at a large scale. Our proposal enables estimating traffic density measures by aggregating data from IoT-connected vehicles and addressing privacy awareness, operational costs, and hardware requirements challenges. Our proposal can integrate existing sensing infrastructures to limit the deployment of additional sensor devices. The sensing agents provide accurate estimates using the information collected from the environment and cooperation with other available sensing devices.

We carried out experimentations using the traffic simulator SUMO with a realistic scenario from the city of Bologna, Italy. We compared the results of traffic density estimation to those obtained by state-of-the-art regression technique and compared endogenous and exogenous results, and cooperative and non-cooperative results from sensing agents. The results obtained by the cooperative estimation of HybridIoT outperform the state-of-the-art methods.

The key benefit of the proposed technique stems from leveraging cost-effective information. Our proposal aggregates FCD and other data collected from environmental sensors to calculate traffic estimates. The use of such information offers economic advantages, as there is no need for costly surveillance cameras or high-performance computing modules to achieve real-time traffic monitoring at the edge. Also, our proposal ensures privacy by avoiding the use of confidential data such as images: this type of data may not be accessible at every point, requires eliminating any personal information (such as the license plate of each vehicle detected by the camera), and significant computational resources to detect traffic.

In the context of the TORRES project[8] we will apply the same estimation technique on a traffic simulation model of the city of Brussels, Belgium. The project TORRES aims at enriching knowledge about the dynamics of urban mobility in the Brussels region through the integration of real data, collected from existing monitoring infrastructures and opportunely anonymized, and synthetic data created through data augmentation methods. In the context of this project, we will develop new AI-based methods for interpolating mobility data considering the uncertainties and unpredictable dynamics of the physical environment.

In future work, we will also investigate methods that enable the sensing agent to determine autonomously the most pertinent aggregation frequency. Also, we will focus on the use of traffic variables to infer the status of traffic. This includes, for instance, predicting anomalies such as accidents on roads by aggregating heterogeneous traffic variables, but also extracting city semantics from data collected from

[8]https://mlg.ulb.ac.be/wordpress/portfolio_page/torres/; Last visited: August 17, 2023

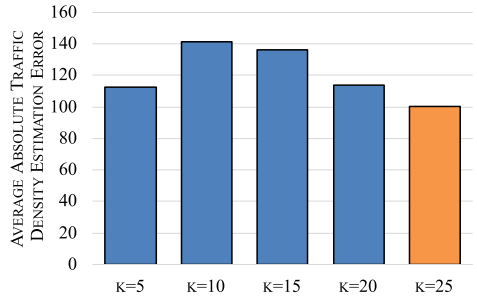

FIGURE 15. Average absolute traffic density estimation error obtained from using different values of k in time-series cross-validation.

vehicles. Further future work concerns the use of HybridIoT for traffic planning [28] using prototyping toolkits such as TRAPP [29] and implementing HybridIoT in edge computing testbeds such as SMOTEC [30].

APPENDIX A
ESTIMATION MODEL CALIBRATION

A. K-FOLD TIME SERIES CROSS-VALIDATION

Let us suppose ℓ is the temporal horizon of the simulation. Consequently, each agent observes or estimates ℓ values at the end of the simulation. When a sensing agent estimates or perceives information at time t, it verifies that the amount of data observed is equal to the size of the partitions (ℓ/k). To do so, we use a simple counter that measures the quantity of data that each sensing agent observes. This counter is reset to zero every time the amount of data is equal to the size of the partitions. If the counter is reset to zero at time t, then all the data windows assembled before t are used as a training set to estimate the next traffic information.

We now verify the most pertinent value for k to be used for time series cross-validation. For *best*, we mean the value that enables minimizing the average traffic density estimation error. We evaluated the following k values (chosen arbitrarily): 5, 10, 15, 20, 25. For this validation, we do not use any source of randomness in SUMO; the reason is that we want to find the best value of k under the same traffic condition.

Figure 15 resumes the results obtained from estimating traffic density measurements using different values of k. In all experiments, we use time-series cross-validation.

Under the same traffic condition (no source of randomness in traffic), the most pertinent value of k is 25. This value will be used in all subsequent experiments.

B. NUMBER OF SENSING AGENTS

The number and location of each sensing agent depend on the urban context in which the MAS operates; traffic management operators may decide the configuration of sensing agents in points of the urban environment where traffic density measurements are required, and camera devices cannot be installed. Herein, the configuration includes the location of the sensing agents or the frequency at which they provide aggregated counts of the vehicles. The information collected from the sensing agents is used to estimate traffic density measurements.

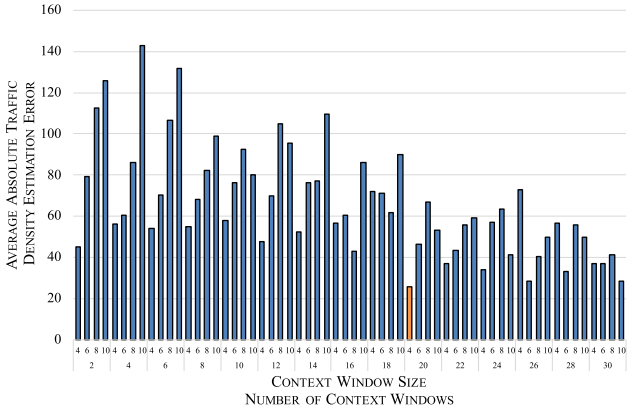

FIGURE 16. Average absolute traffic density estimation error obtained from using different parameters related to context windows. The orange bar indicates the error related to the chosen combination of parameters used to perform simulations.

C. CONTEXT WINDOWS CONFIGURATION

The goal of this validation is to find the most pertinent values for the following parameters, related to context windows:

- **Number of context windows**: the number of context windows that each sensing agent uses when estimating a traffic density measure. We evaluated all values in range [2, 30] at step of 2 for this parameter;
- **Size of context windows**: the exact size of each context window in sensing agents' knowledge base. We evaluated the following values: 4, 6, 8, 10.

We evaluated all combinations of previous values. We consider the most pertinent combination of parameters as the one that allows for minimizing the traffic density estimation error. In this validation, we do not consider any source of randomness in traffic simulations. Figure 16 shows the average absolute traffic density estimation error obtained using different combinations of context windows size and the number of context windows. These parameters are used by all sensing agents when estimating traffic density measures.

From the plot in Figure 16, we observe that the estimation error decreases as we consider a higher number of context windows in the estimation process. Many context windows enable the sensing agents to calibrate the traffic estimation response. The motivation is intrinsic to the estimation technique: where the most similar context windows receive higher weights than the others. If few context windows are used, having noisy or incorrect values, these windows receive a high score in the calculation of the weight w, resulting in inaccurate traffic density estimates. In other words, a small number of context windows makes sensing agents unable to well discriminate between different traffic dynamics, leading to high estimation error.

There is no specific rule for finding the minimum number of context windows to be used for obtaining accurate estimates. This value depends on the environmental context

in which sensing agents operate. In some regions, the traffic dynamics can vary unpredictably so that at a specific time instant, a variable quantity of context windows would be needed to estimate traffic density measures. For example, if few context windows contain information whose dynamics are similar to those observed by the sensing agent, then adding further context windows in the estimation process could add noise to the traffic estimate, resulting in a high estimation error. This is a consequence of the way the sensing agents chose context windows according to the distance d: the more context windows are chosen, the more likely the sensing agent chooses context windows that have a higher distance from the reference context. However, this also depends on the quantity of data in knowledge bases and the noise in context windows.

APPENDIX B
COMPUTATIONAL TIME

All experiments have been carried out on a server machine equipped with Intel Xeon Gold 6242 processors, and Linux operating system (kernel version 4.15). For the sake of comparison, here we also report the performance of the proposed method on a desktop machine equipped with an Intel i5-8279U processor and Windows 11 operating system.

For both configurations, we profiled the execution of the proposed method using the "cProfile" tool for Python. We report, for both configurations, the following metrics:

- **Per call time**: the amount of time spent on a single function call. This is measured in seconds.
- **Cumulative time**: is the cumulative time spent on this and all sub-functions (from invocation until exit). This figure is accurate even for recursive functions. This is measured in seconds.

The following tables report the time (in seconds) required to perform the main steps of the proposed method, for both machine configurations. Specifically, we report the time for the following steps:

- **Agent cycle**: the amount of time required for a sensing agent to complete its nominal behavior. This includes aggregating data, estimating traffic density, and operations on context windows;
- **Traffic density estimation**: the amount of time required to estimate a traffic density measure;
- **Context windows retrieval**: the amount of time required to obtain the set of most similar context windows needed to estimate the traffic density measures;
- **Traffic variables evaluation**: the amount of time required to evaluate the values of the traffic variables.

Table 7 reports the cumulative time for the main steps of the proposed method, obtained from profiling the execution of a single simulation on the server machine (with Xeon CPU).

Table 8 reports the Per Call time for the main steps of the proposed method, obtained from profiling the execution of a single simulation on the server machine (with Xeon CPU).

TABLE 7. Cumulative time (in seconds) obtained by profiling one simulation with SUMO, and using different traffic density estimation techniques. These results relate to the machine equipped with a Xeon CPU. Colors indicate the time required (from blue, less time, to red, more time).

	Cumulative time		
Operation	Non cooperative	Cooperative endogenous	Cooperative exogenous
Agent cycle	2953.31	2921.72	4656.78
Traffic density Estimation	86.02	137.36	1723.69
Context Windows Retrieval	44.88	86.26	1618.07
Traffic variable evaluation	2445.18	2374.03	2525.00

TABLE 8. Per call time (in seconds) obtained by profiling one simulation with SUMO and using different traffic density estimation techniques. These results relate to the machine equipped with a Xeon CPU. Colors indicate the time required (from blue, less time, to red, more time).

	Per call		
Operation	Non cooperative	Cooperative endogenous	Cooperative exogenous
Agent cycle	0.017	0.017	0.027
Traffic density Estimation	0.002	0.004	0.056
Context Windows Retrieval	0.002	0.002	0.002
Traffic variable evaluation	0.080	0.077	0.082

TABLE 9. Cumulative time (in seconds) obtained by profiling one simulation with SUMO, and using different traffic density estimation techniques. These results relate to the machine equipped with an i5 CPU. Colors indicate the time required (from blue, less time, to red, more time).

	Cumulative time		
Operation	Non cooperative	Cooperative endogenous	Cooperative exogenous
Agent cycle	3564.049	3681.972	5292.547
Traffic density Estimation	103.473	148.435	1878.678
Context Windows Retrieval	50.062	98.138	50.598
Traffic variable evaluation	2952.08	2997.585	2910.594

TABLE 10. Per call time (in seconds) obtained by profiling one simulation with SUMO and using different traffic density estimation techniques. These results relate to the machine equipped with an i5 CPU. Colors indicate the time required (from blue, less time, to red, more time).

	Per call		
Operation	Non cooperative	Cooperative endogenous	Cooperative exogenous
Agent cycle	0.02	0.021	0.03
Traffic density Estimation	0.003	0.005	0.061
Context Windows Retrieval	0.002	0.002	0.002
Traffic variable evaluation	0.096	0.098	0.095

Table 9 reports the cumulative time for the main steps of the proposed method, obtained from profiling the execution of a single simulation on the server machine (with i5 CPU).

Table 10 reports the Per Call time for the main steps of the proposed method, obtained from profiling the execution of a single simulation on the server machine (with i5 CPU).

Figure 17 shows the average cycle time among all sensing agents, and the average number of context windows. We calculated the results using a moving average over the data with a window of size 50.

From the figure, we can see that the time required to complete an agent cycle time initially increases significantly with the number of context windows, and then slowly converges in the case of non-cooperative and cooperative endogenous estimation. In the case of the exogenous estimation, the time required to complete an agent cycle is on average more than twice as long as the endogenous estimation (either cooperative or non-cooperative). In exogenous estimation,

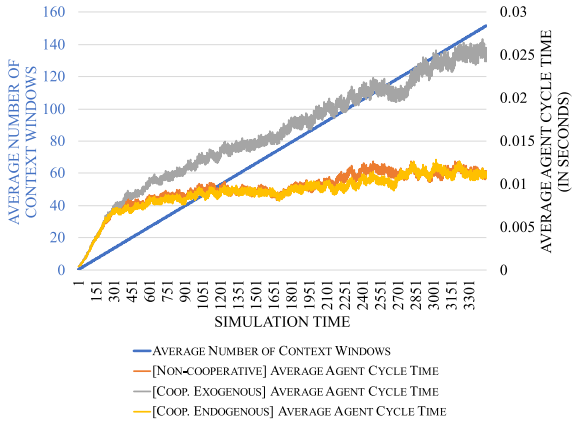

FIGURE 17. Average number of context windows (left axe) and average agent cycle time (right axe) obtained by HybridIoT.

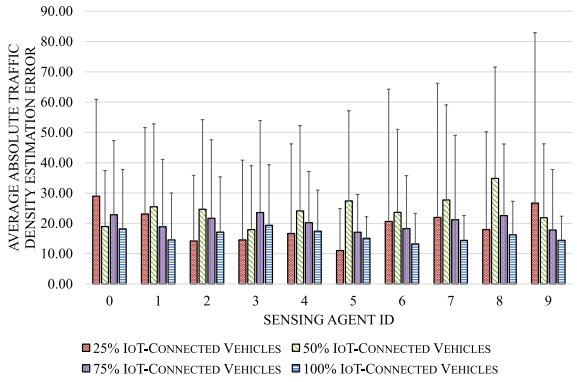

FIGURE 18. Non-cooperative estimation method assessment. Average absolute traffic density estimation error (MAE) obtained from HybridIoT.

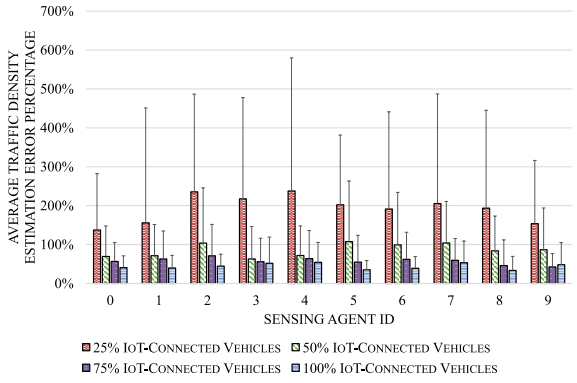

FIGURE 19. Non-cooperative estimation method assessment. Average traffic density estimation error percentage (MAPE) obtained from HybridIoT.

the sensing agents estimate one traffic density measure for each set of context windows, one per traffic variable. However, the time required to complete an agent cycle is limited, considering the average number of agent context windows.

APPENDIX C
NON-COOPERATIVE ENDOGENOUS TRAFFIC DENSITY ESTIMATION

In this appendix, we present the results obtained from the estimation technique HybridIoT, using non-cooperative

TABLE 11. Best aggregation frequency (in seconds) obtained by the non-cooperative estimation method.

	Non-Cooperative			
	25%	50%	75%	100%
SensingAG #0	11	11	11	9
SensingAG #1	11	11	11	9
SensingAG #2	9	11	11	9
SensingAG #3	3	11	11	11
SensingAG #4	11	11	11	9
SensingAG #5	3	9	11	9
SensingAG #6	11	11	9	9
SensingAG #7	9	11	11	11
SensingAG #8	9	11	11	9
SensingAG #9	9	11	11	9

estimation. Herein, the sensing agents use only their historical data to provide an estimate of the traffic density.

We run 100 simulations with SUMO, using the scenario from Bologna, for each percentage of IoT-connected vehicles (from 25% to 100%). Figure 18 shows the MAE obtained for each sensing agent and each percentage (25% to 100%) of IoT-connected vehicles.

The average traffic density estimation error (MAE) is 38.43 vehicles, the standard deviation is 22.95.

When considering the MAE, the lower the error, the better the model performs in the task of estimating traffic density. However, it is not possible to assess the quality of the results based only on this error measure. For this, we also measure the MAPE for the estimation error generated by the non-cooperative method in HybridIoT (Figure 19).

Considering the MAPE, the non-cooperative method output a significantly high error. The high error is mainly due to the cold start problem: agents have few data windows when they start estimating data, thus leading to high errors. When the agents start their functioning, the amount of data windows can be insufficient to ensure accurate estimates. Consequently, the estimation error produced by the agents propagates in their data windows and the subsequent estimates, leading to high errors.

APPENDIX D
STABILITY OF THE ESTIMATION ERROR

In this experiment, we verify that the chosen number of simulations (100 for each percentage of IoT-connected vehicles) is sufficient to obtain a statistically significant sample. Because traffic behavior is variable among simulations, it is important to validate the proposed technique on a sufficient number of simulations to assess the accuracy of the estimation results compared to state-of-the-art techniques. Herein we calculate the cumulative mean of the traffic density estimation error among sensing agents, for all simulations. Despite the random behavior of the traffic, after a certain number of simulations, the cumulative mean estimation error should converge to a stable result. In other words, after a certain number of simulations, two subsequent cumulative mean results vary by an irrelevant quantity. If so, we can argue that the number of simulations performed is sufficient to obtain a statistically significant sample.

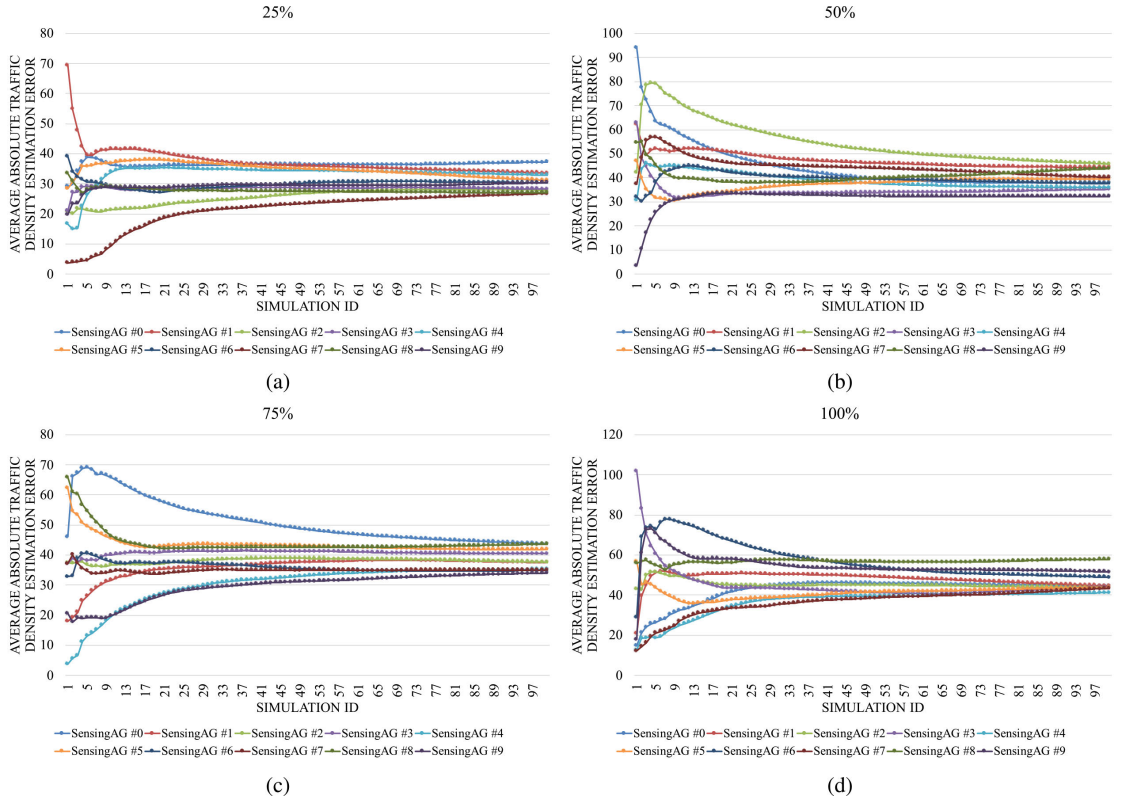

FIGURE 20. Cumulative mean stability calculated from the traffic density estimation error, using non-cooperative estimation technique. Each plot refers to a specific percentage of IoT-connected vehicles (25%, 50%, 75% and 100%).

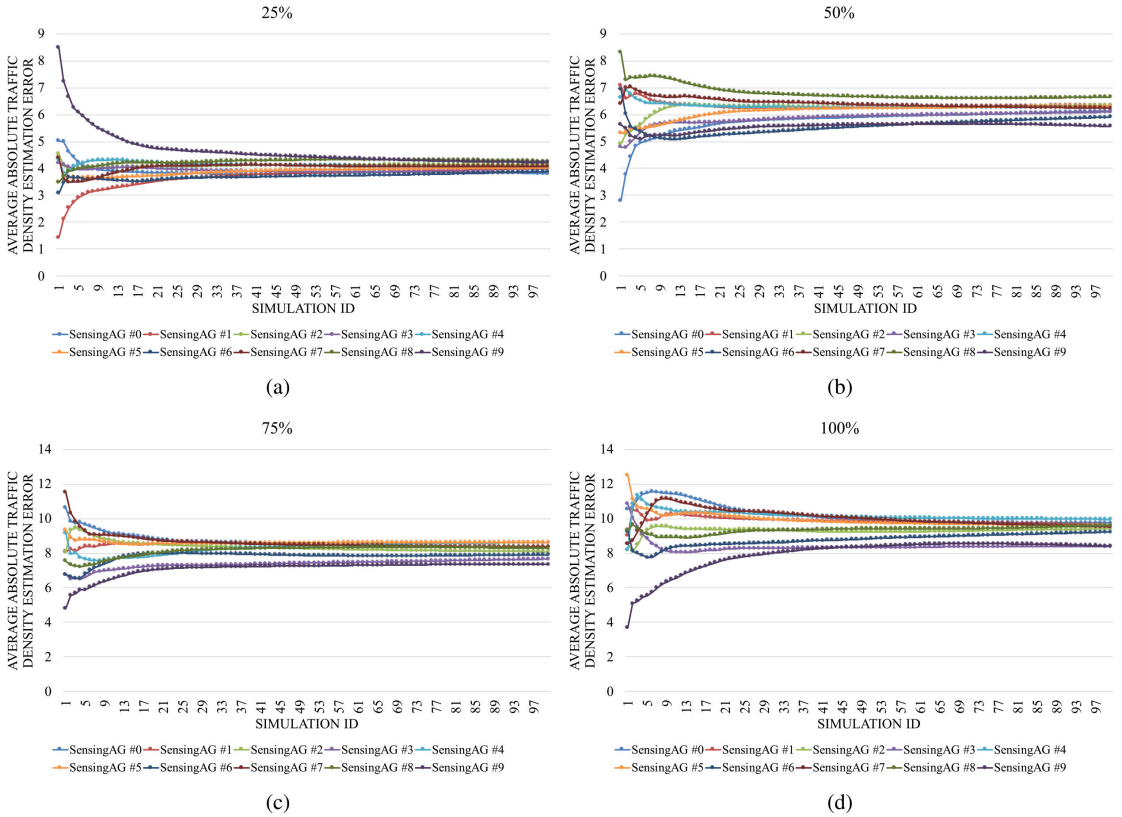

FIGURE 21. Cumulative mean stability calculated from the traffic density estimation error, using cooperative endogenous estimation technique. Each plot refers to a specific percentage of IoT-connected vehicles (25%, 50%, 75% and 100%).

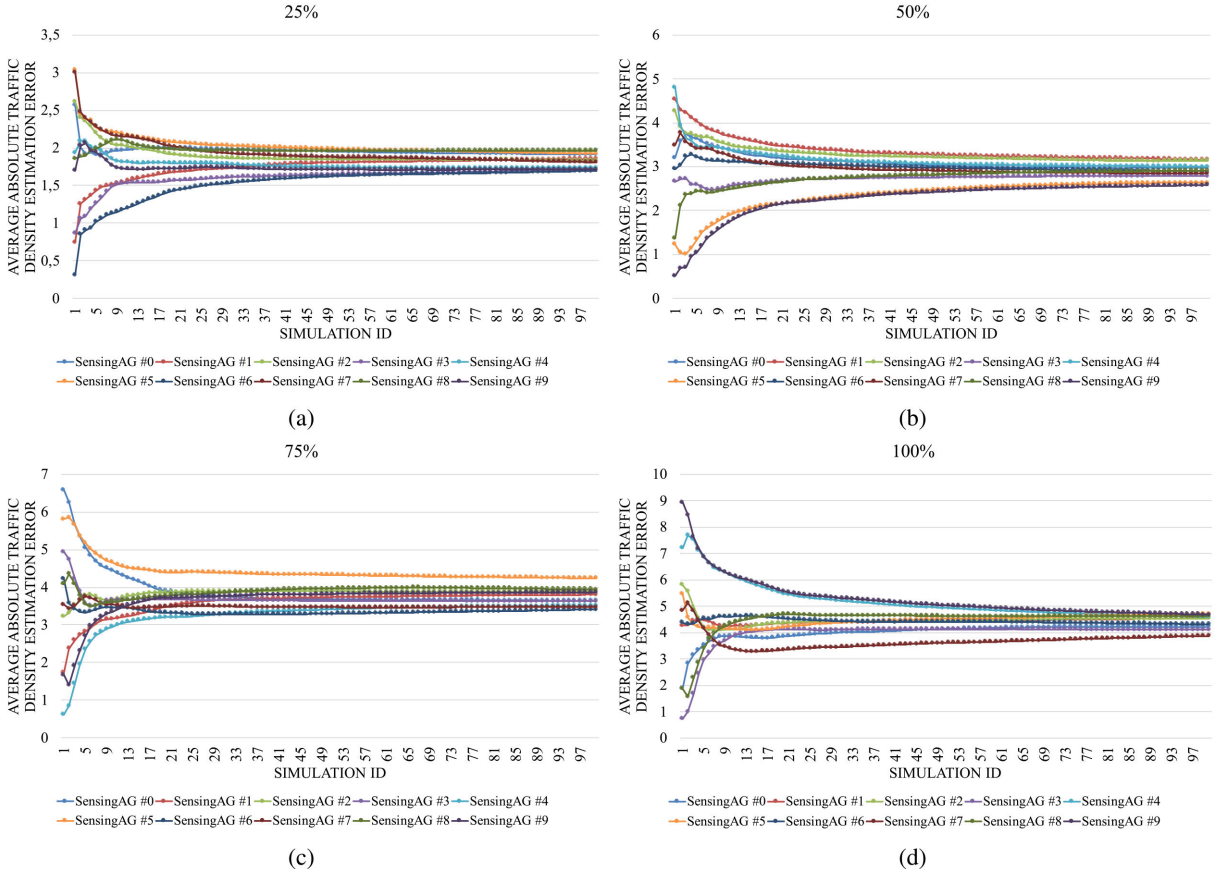

FIGURE 22. Cumulative mean stability calculated from the traffic density estimation error, using cooperative exogenous estimation technique. Each plot refers to a specific percentage of IoT-connected vehicles (25%, 50%, 75% and 100%).

TABLE 12. Best aggregation frequency (in seconds) obtained by the cooperative endogenous estimation method.

	Cooperative Endogenous			
	25%	50%	75%	100%
SensingAG #0	3	3	7	5
SensingAG #1	9	9	9	7
SensingAG #2	9	3	7	3
SensingAG #3	9	3	3	5
SensingAG #4	7	3	3	3
SensingAG #5	3	7	7	3
SensingAG #6	3	3	5	7
SensingAG #7	3	3	7	3
SensingAG #8	3	9	3	5
SensingAG #9	7	3	3	5

TABLE 13. Best aggregation frequency (in seconds) obtained by the cooperative exogenous estimation method.

	Cooperative Endogenous			
	25%	50%	75%	100%
SensingAG #0	7	7	5	5
SensingAG #1	9	5	5	3
SensingAG #2	7	5	5	5
SensingAG #3	7	7	5	5
SensingAG #4	7	7	5	5
SensingAG #5	7	5	5	3
SensingAG #6	7	5	5	5
SensingAG #7	7	5	5	5
SensingAG #8	9	7	5	5
SensingAG #9	7	5	5	5

Figure 20 shows the plots of the cumulative average estimation error obtained from the non-cooperative estimation technique. The plots show that the cumulative mean converges towards a stable result even when a limited percentage of vehicles provide data to sensing agents.

Figure 21 shows the plots of the cumulative average estimation error obtained from the non-cooperative estimation technique. The plots show that the cumulative mean converges towards a stable result even when a limited percentage of vehicles provide data to sensing agents.

Figure 22 shows the plots of the cumulative average estimation error obtained from the non-cooperative estimation technique. The plots show that the cumulative mean converges towards a stable result even when a limited percentage of vehicles provide data to sensing agents.

APPENDIX E
BEST AGGREGATION FREQUENCY FOR SENSING AGENTS

Tables 11, 12, 13 list the best aggregation frequencies obtained from non-cooperative, cooperative endogenous and cooperative exogenous estimation methods in HybridIoT.

REFERENCES

[1] D. Helbing, F. Fanitabasi, F. Giannotti, R. Hänggli, C. I. Hausladen, J. van den Hoven, S. Mahajan, D. Pedreschi, and E. Pournaras, "Ethics of smart cities: Towards value-sensitive design and co-evolving city life," *Sustainability*, vol. 13, no. 20, p. 11162, Oct. 2021.

[2] P. Matczak, A. Wójtowicz, A. Dąbrowski, and K. Mączka, "Cost-effectiveness of CCTV surveillance systems: Evidence from a Polish city," *Eur. J. Criminal Policy Res.*, vol. 29, no. 4, pp. 555–577, Dec. 2023.

[3] Z. Wang, J. Zhan, Y. Li, Z. Zhong, and Z. Cao, "A new scheme of vehicle detection for severe weather based on multi-sensor fusion," *Measurement*, vol. 191, Mar. 2022, Art. no. 110737.

[4] C. Caiazza, S. Giordano, V. Luconi, and A. Vecchio, "Edge computing vs centralized cloud: Impact of communication latency on the energy consumption of LTE terminal nodes," *Comput. Commun.*, vol. 194, pp. 213–225, Oct. 2022.

[5] U. Mittal, P. Chawla, and R. Tiwari, "EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models," *Neural Comput. Appl.*, vol. 35, no. 6, pp. 4755–4774, Feb. 2023.

[6] M. Younan, E. H. Houssein, M. Elhoseny, and A. A. Ali, "Challenges and recommended technologies for the industrial Internet of Things: A comprehensive review," *Measurement*, vol. 151, Feb. 2020, Art. no. 107198.

[7] H. Guo, J. Liu, and H. Qin, "Collaborative mobile edge computation offloading for IoT over fiber-wireless networks," *IEEE Netw.*, vol. 32, no. 1, pp. 66–71, Jan. 2018.

[8] N. Balamuralidhar, S. Tilon, and F. Nex, "MultEYE: Monitoring system for real-time vehicle detection, tracking and speed estimation from UAV imagery on edge-computing platforms," *Remote Sens.*, vol. 13, no. 4, p. 573, Feb. 2021.

[9] I. Nigam, C. Huang, and D. Ramanan, "Ensemble knowledge transfer for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1499–1508.

[10] C. Qin and E. Pournaras, "Coordination of drones at scale: Decentralized energy-aware swarm intelligence for spatio-temporal sensing," *Transp. Res. C, Emerg. Technol.*, vol. 157, Dec. 2023, Art. no. 104387.

[11] C. Qin and E. Pournaras, "Short vs. long-term coordination of drones: When distributed optimization meets deep reinforcement learning," 2023, *arXiv:2311.09852*.

[12] M. Humayun, F. Ashfaq, N. Z. Jhanjhi, and M. K. Alsadun, "Traffic management: Multi-scale vehicle detection in varying weather conditions using YOLOv4 and spatial pyramid pooling network," *Electronics*, vol. 11, no. 17, p. 2748, Sep. 2022.

[13] Y. Mekdad, A. Aris, L. Babun, A. E. Fergougui, M. Conti, R. Lazzeretti, and A. S. Uluagac, "A survey on security and privacy issues of UAVs," *Comput. Netw.*, vol. 224, Apr. 2023, Art. no. 109626.

[14] S. Li, G. Li, Y. Cheng, and B. Ran, "Urban arterial traffic status detection using cellular data without cellphone GPS information," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 446–462, May 2020.

[15] H. Lee, J. Lee, and Y. Chung, "Traffic density estimation using vehicle sensor data," *J. Intell. Transp. Syst.*, vol. 26, no. 6, pp. 675–689, Nov. 2022.

[16] D. Nam, R. Lavanya, R. Jayakrishnan, I. Yang, and W. H. Jeon, "A deep learning approach for estimating traffic density using data obtained from connected and autonomous probes," *Sensors*, vol. 20, no. 17, p. 4824, Aug. 2020.

[17] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021.

[18] T. N. Gia, J. P. Queralta, and T. Westerlund, "Exploiting LoRa, edge, and fog computing for traffic monitoring in smart cities," in *LPWAN Technologies for IoT and M2M Applications*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 347–371.

[19] A. A. Levin, D. D. Klimov, A. A. Nechunaev, L. S. Prokhorenko, D. S. Mishchenkov, A. G. Nosova, D. A. Astakhov, Y. V. Poduraev, and D. N. Panchenkov, "Assessment of experimental OpenCV tracking algorithms for ultrasound videos," *Sci. Rep.*, vol. 13, no. 1, p. 6765, Apr. 2023.

[20] S. K. Rout, B. Sahu, P. K. Mohapatra, S. N. Mohanty, and A. K. Sharma, "IoT and an intelligent cloud-based framework to build a smart city traffic management system," in *Enabling Technologies for Effective Planning and Management in Sustainable Smart Cities*, M. A. Ahad, G. Casalino, and B. Bhushan, Eds. Berlin, Germany: Springer, 2023, pp. 283–302.

[21] D. Lo Castro, D. Tegolo, and C. Valenti, "A visual framework to create photorealistic retinal vessels for diagnosis purposes," *J. Biomed. Informat.*, vol. 108, Aug. 2020, Art. no. 103490.

[22] D. A. Guastella, V. Campss, and M.-P. Gleizes, "A cooperative multi-agent system for crowd sensing based estimation in smart cities," *IEEE Access*, vol. 8, pp. 183051–183070, 2020.

[23] M. Behrisch and P. Hartwig, "A comparison of SUMO's count based and countless demand generation tools," in *Proc. SUMO Conf.*, vol. 2, 2022, pp. 125–131.

[24] B. Herfort, S. Lautenbach, J. Porto de Albuquerque, J. Anderson, and A. Zipf, "A spatio-temporal analysis investigating completeness and inequalities of global urban building data in OpenStreetMap," *Nature Commun.*, vol. 14, no. 1, p. 3985, Jul. 2023.

[25] L. Bieker, D. Krajzewicz, A. P. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: A real world traffic scenario from the city of Bologna," in *Modeling Mobility With Open Data*, M. Behrisch and M. Weber, Eds. Berlin, Germany: Springer, 2015, pp. 47–60.

[26] R. J. Gräbe and J. W. Joubert, "Are we getting vehicle emissions estimation right?" *Transp. Res. D, Transp. Environ.*, vol. 112, Nov. 2022, Art. no. 103477.

[27] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, VIC, Australia: OTexts, 2018.

[28] B. Davis, G. Jennings, T. Pothast, I. Gerostathopoulos, E. Pournaras, and R. E. Stern, "Decentralized optimization of vehicle route planning—A cross-city comparative study," *IEEE Internet Comput.*, vol. 25, no. 3, pp. 34–42, May 2021.

[29] I. Gerostathopoulos and E. Pournaras, "TRAPPed in traffic? A self-adaptive framework for decentralized traffic optimization," in *Proc. IEEE/ACM 14th Int. Symp. Softw. Eng. Adapt. Self-Managing Syst. (SEAMS)*, Montreal, QC, Canada, May 2019, pp. 32–38.

[30] Z. Nezami, E. Pournaras, A. Borzouie, and J. Xu, "SMOTEC: An edge computing testbed for adaptive smart mobility experimentation," 2023, *arXiv:2307.11181*.