IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 February 2024, accepted 6 April 2024, date of publication 10 April 2024, date of current version 22 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387295

RESEARCH ARTICLE

Reinforcement Learning for Intelligent Sensor Virtualization and Provisioning in Internet of Vehicles (IoV)

SLIM ABBES, (Member, IEEE), AND SLIM REKHIS

Higher School of Communication of Tunis (SUP'COM), LR11TIC04, Communication Networks and Security (CNAS) Research Laboratory, University of Carthage, Tunis 1054, Tunisia

LR11TIC02, Green and Smart Communication Systems (GRES'COM) Research Laboratory, University of Carthage, Tunis 1054, Tunisia

Corresponding author: Slim Abbes (slim.abbes@supcom.tn)

This work was supported by the Ministry of Higher Education and Scientific Research, through the Scientific Program of the Communication Network and Security (CNAS) Research Laboratory.

ABSTRACT The Internet of Vehicles (IoV) is a powerful application of the Internet of Things (IoT) in the Intelligent Transportation System (ITS). It enables device connectivity, and interaction with the environment, and improves efficiency in utilizing sensor data. Leveraging the capabilities of vehicle sensors to provide virtual sensor service presents an opportunity to make the most of underutilized sensor resources and offer on-demand sensor services. However, challenges remain in ensuring spatio-temporal sensor availability due to the high mobility of vehicles and frequent changes in the IoV network topology. Furthermore, providing sensor services in the dynamic IoV-Cloud market, while balancing cost-effectiveness for consumers and profitability for service providers, poses a significant challenge. To address these challenges, we develop an IoV-Cloud architecture tailored for vehicle virtual sensor provisioning, integrating interactive functional layers and components for a comprehensive solution. This architecture supports vehicle sensor virtualization and management, facilitating the intelligent provisioning of on-demand, elastic, and scalable vehicle virtual sensors from existing physical sensors within mobile vehicles. To address vehicle mobility and enhance sensor availability, our design permits virtual sensors to maintain data provision even when switched to different physical sensors. Further optimizing sensor utilization, our system allows multiple virtual vehicle sensors to share a single physical sensor by employing a configuration mapping mechanism. At the heart of our strategy is a reinforcement learning model that dynamically selects the most suitable vehicle physical sensor for each time slot throughout the service period, considering both physical availability and cost-effectiveness for the Sensor Cloud Service Provider (SCSP). We simulate the proposed intelligent sensor selection model using both Q-learning and SARSA algorithms, demonstrating its effectiveness in intelligent and dynamic sensor provisioning.

INDEX TERMS Internet of Vehicles, reinforcement learning, sensor virtualization, intelligent sensor provisioning.

I. INTRODUCTION

In recent years, researchers and automobile industry groups have gained much interest in the development of Intelligent Transportation Systems. IoV as an integral component of ITS, is likewise undergoing steep growth. In the IoV, vehicles serve as sensor hubs to gather environmental data through heterogeneous, diversified, and versatile on-board embedded sensors, which are then made available to provide novel applications and added-value services as on-demand elastic sensor services. Because of their wealth of on-board resources, wide-spread spatio-temporal accessibility, and mobility, nowadays vehicles may be viewed as the best candidates for providing ubiquitous information services. In addition, the diversity of in-vehicle sensors and the capabilities of vehicles in terms of storage, processing, and communication allow vehicles to offer a wide range of opportunities under the public sensing paradigm, outperforming

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

other potential mobile resources like smartphones. However, the vehicle's capabilities remain limited, and it is difficult to solely meet the computing requirements and the complex sensor provision tasks [1]. Moreover, most of the existent applications co-exist with inefficiently used infrastructures, either because resources are duplicated (dedicated devices and networks) or are under-used (idle time, one-time data-processing) along with possible sensor replication in the same geographical zone, especially in urban zones with high density. The use of IoV has resulted in a wide range of applications that require the utilization of heterogeneous components. These components include high encoding and decoding capabilities for data, as well as various hardware and software platforms. Additionally, these applications are built on top of different types of devices, each with its own unique data format and sensor configuration [2]. As a result, there is a growing need for a significant effort to ensure the manageability and adjustability of sensors over time. Furthermore, the expectations for a substantial improvement in sensor provision mechanisms have also risen to meet the diverse and multiple service requests that arise from the use of IoV applications. Therefore, it is essential to efficiently exploit and share the vehicle sensors among several applications.

The virtualization is envisioned as a key technology to make the sensor sharing and exploitation possible to meet the efficient use of sensor resources [3]. Moreover, It represents a promising opportunity that enables the migration from the traditional WSN data acquisition towards the intelligent Sensor-as-a-Service provision. Motivated by the endless opportunities of virtualization for creating novel services and developing added-value applications, we utilize the virtualization technology to virtualize the vehicle sensors on the fog and the cloud servers to offload the processing tasks from vehicle's On-Board Units (OBU) and to provide continuous access to the instantiated virtual sensors for various and multiple sensors service consumers. In particular, we propose sensor virtualization and provisioning a solution that consists in selecting at each time slot the suitable vehicle physical sensor to compose the vehicle virtual sensor based on the vehicle mobility and sensor allocation cost.

Nevertheless, there are still several challenges that are mainly related to the availability of vehicles embedding the sensors due to the high vehicles mobility, frequent topology change, and driving behavior volatility [4]. Furthermore, vehicles in the IoV networks are characterized by their intermittent and short-lived availability in the target geographic zone of the requested service, which leads to an unsustainable service provision [5]. In fact, attaining perfect knowledge of the vehicular network for making definitive decisions on vehicle availability is practically unachievable. This limitation arises from various factors, such as the difficulties in gathering precise mobility data, transmission errors, and the scarcity of detailed model information. Furthermore, the vehicular network frequently faces unpredictable events, including climate variations and unforeseen incidents like traffic jams, accidents, and traffic detours. Additionally, existing approaches are vulnerable to the influence of inaccurate measurements, which can lead to a reduction in the overall effectiveness of localization efforts [6], [7]. All these factors overlap to make the decision on the selection of the vehicle physical sensor to compose the vehicle virtual sensor is more and more complicated.

To avoid such challenging situations, this study proposes a reinforcement learning (RL) method for dynamically and adaptively selecting at each time slot the appropriate vehicle physical sensors to compose the vehicle virtual sensor. The selection is based on the availability of vehicles in the network, the sensor allocation cost, and the optimization of sensor utilization. To the best of our knowledge, this work is the first study that combines intelligent sensor selection and vehicle virtual sensor provision based on vehicle availability and cost-effectiveness in the field of IoV in a single study.

This study is motivated by the lack of research that combines both RL model and availability-based sensor provisioning in a single study. Besides, the existing studies do not consider the IoV mobility and the sensor provision architecture and components, i.e., edge, fog, and cloud, and thus we propose that the virtualization is executed at the cloud after performing the mobility data pre-processing and the sensor configuration adjustability in the fog layer.

The proposal of this work is to design an IoV-Cloud architecture for vehicle virtual sensor provisioning in the vehicular context. Therefore, reinforcement learning is used for intelligent vehicle physical sensor selection based on the spatio-temporal availability of the vehicles that represent the sensor suppliers. The proposed selection mechanism takes into consideration the optimization of both resource utilization and allocation cost of the vehicle physical sensors. The cost depends on the number of physical sensors allocated to compose the vehicle virtual sensor and on the vehicle virtual sensor service production period.

In this paper, we design an IoV-Cloud architecture for vehicle virtual sensor provisioning that encompasses interactive functional layers and components. A sensor virtualization and management solution is proposed to enable the intelligent provisioning of on-demand, elastic, and scalable vehicle virtual sensors, starting from a set of physical sensors embedded in mobile vehicles. To cope with the mobility of vehicles, and to increase availability, we allow the virtual sensor to continue providing data when being attached to a different physical sensor from one-time slot to another.

To reduce the number of required physical sensors and increase their availability, we also allow the sharing of the same physical sensor with multiple virtual vehicle sensors, and we provide a configuration mapping mechanism to enable this sharing. The configuration mapping is a function of transformation of the data. it gets as input the data generated by a vehicle physical sensor configured with a configuration C (e.g., sensitivity, resolution, precision) and outputs produced data as if they were generated by the same vehicle physical sensor configured with different configuration C'. Such a scheme allows for generating multi-tenant virtual vehicle sensors based on physical sensors sharing.

Finally, a reinforcement learning model is used for intelligent and dynamic selection of the vehicle physical sensor (at each time slot during the whole required service period) which takes into consideration its physical availability and also the cost-effectiveness for the Sensor Cloud Service Provider (SCSP). The proposed reinforcement learning model for sensor selection is tested using Q-learning and SARSA algorithms.

This paper is an extension of the work we presented in [8]. In particular, the four following aspects are developed in this work:

1) We developed the mechanisms of sensor configuration mapping which allows software mapping of sensor configuration and transformation of collected data, so that the same physical sensors (e.g., camera) can be used with multiple configurations (e.g., zoom, angle of view, resolution) simultaneously, as if they were multiple sensors allocated physically. This feature impacts the virtual sensor model becoming multi-tenancy, and the used rewards functions in reinforcement learning.
2) We introduced a global cost calculation function for vehicle virtual sensors, while considering a set of physical sensors with different unit costs based on the quality of supplied sensors and the configurations allowed by each physical sensor. We highlighted the impact of new defined metrics such as "multi-tenant sensor sharing", "sensor re-selection" and "true sensor selection" on the global provisioning cost of the vehicle virtual sensor.
3) In addition to the Q-learning model, we also developed a SARSA-based model for the dynamic and intelligent selection of physical sensors composing the virtual vehicle sensors. We compared the obtained results for the two reinforcement learning models Q-learning and SARSA, and we also compared their efficiency with respect to random sensors selection algorithm.
4) We extended the simulation results to evaluate the virtual sensor provisioning costs and the rate of economics with respect to the parameters of networks such as the density of vehicles and the vehicles' speed variation.

The remaining part of this paper is structured as follows: Previous works related to sensor virtualization and provisioning are reviewed in Section II. In Section III, the IoV-Cloud architecture for vehicle virtual sensors is presented. The concept of vehicle sensor virtualization in the IoV context is explained in Section IV. The Vehicle virtual sensor modeling and provisioning cost computation are presented in Section V. In Section VI, we propose a vehicle sensor selection approach based on reinforcement learning. Then, the performances of the proposed intelligent vehicle sensor selection are tested and evaluated using reinforcement learning models in Section VII. Finally, we conclude this paper in Section VIII.

II. RELATED WORK

In this section, works related to sensor virtualization and provision are discussed and analyzed to justify the necessity of the proposed solution. Recent research addressed the sensor virtualization in various research areas and application domains including intelligent buildings [9], Healthcare monitoring [10], and Smart City [11]. Because of the large number of vehicle sensors used to collect environmental, traffic, and mobility data and the necessity for managing the sensor resources for providing novel services, sensor virtualization in the IoV field has been one of the most demanding areas of research. The authors in [12] present a sensor virtualization platform (SenseWear) for seamless integration of wearable sensors into smartphone applications to enhance their functionality. Sensor virtualization frameworks has been presented that rely on virtual sensor formation and provision with a trade-off between QoS requirements satisfaction [13] and energy consumption minimization [14] and [15]. The authors in [16] design a sensor virtualization solution in the WSN using multi-threading middleware at the sensor supplier nodes in order to manage the service provision priorities and therefore minimize the processing delays owed to the aggregation of the sensed healthcare data for Ambient Assisted Living (AAL) applications.

However, the limited capabilities of IoT devices in terms of energy preservation, due to the huge amount of data to store and process, were overcome with the rise of fog and cloud computing technologies. The advantages of IoT sensor virtualization techniques using cloud computing are used for optimizing the sensor resources sharing, and therefore handling the massive growing demands of IoT application [17], [18]. In the same context, a solution of Sensor as a Service (Se-aaS) is designed to offer access to IoT framework services via authorization access polices [19]. A sensor-cloud platform was implemented to perform the management and also virtualization of heterogeneous wireless sensors [20]. In addition, that solution takes into consideration the sensor failure detection based on network connection status. To gather and process sensing data in IoV, a two-tier system for data routing and processing is suggested to achieve mobile Crowd-sensing tasks. The solution consists in using a vehicle fog architecture to improve communication efficiency and reduce the processing load made on vehicles [21]. To address the challenges of energy constraints, rapidly changing topology, and frequent network disconnections in VANETs, the cluster head (CH) selection process incorporates a weighted metrics combination including mobility factor (incorporating vehicle speed, distance, velocity, and acceleration changes), community neighborhood, eccentricity, and trust. The befit factor includes dynamic vehicle location predictions through the Kalman filter for accurate CH stability assessment [22]. The authors in [23] introduce a selection process in Cognitive

Radio (CR) VANETs that employs a fuzzy logic-based scheme to enhance network stability, security, and reliability, addressing challenges inherent in vehicular networks. This approach uses CR technology for efficient spectrum sensing, prioritizing parameters like the vehicle's average velocity, distance, network connectivity level, lane weight, and trustworthiness.

However, the virtualization in a vehicular environment should not be limited to the resource management and sharing but also to the provision of the sensor while considering the mobility, heterogeneity and availability of sensor suppliers. Furthermore, the dissemination of mobile sensors in the IoV networks at large topology and scale often increases uncertainty about their availability, making sensors virtualization is a possible solution.

To address these issues, a novel virtualization concept for IoV sensors is developed in [24]. It relies on configuration mapping and sensors reconfiguration to guarantee an optimized IoV sensor utilization and dynamic resource sharing while coping with high mobility and frequent topology change issues. Sensor virtualization in the IoV system obviously requires a prior knowledge of vehicle sensor location and thus a consistent methods to select the most available sensors to compose the virtual sensors. However, using conventional techniques that exploit, for instance, the Euclidean distance to determine the target sensor position [25] or the sensor's residual energy to compose the virtual sensor. In [26], the authors argue the advantages of virtual sensors to address the limitation of using a high number of sensors for service provision mostly when considering the high mobility limited knowledge of system status.

Addressing availability and mobility challenges in vehicular networks requires robust strategies that account for the dynamic nature of vehicle movements and ensure reliable connectivity and service availability [27]. In [28], the authors propose a novel system that combines positioning and vehicle direction to provide a predictive evaluation. Their approach utilizes a combination of Machine Learning (ML) and Multiple-Input Multiple-Output (MIMO) techniques to estimate the location and driving direction of nearby vehicles. The authors in [29] address the issue of GPS positioning errors by employing an extended Kalman filter method. They propose a vehicle positioning scheme that leverages the existing GPS trajectory as a reference and trains a model-free neural network to enhance the accuracy of positioning. In [30], the authors employ a vehicle-mounted camera to capture vehicle images and utilize a deep neural network to process these images and estimate the vehicle's location.

Since traditional techniques cannot handle the dynamical requirement of IoV [31], in recent years, artificial intelligence technologies have highly attracted the interest of researchers dealing with sensor resource management [32] using the machine learning with supervised [33] and unsupervised [34] models. These ML models require a large dataset and show many limitations regarding the diversity of scenarios and incidents that are generated in the vehicular networks due to the high mobility and drastic topology change. In addition, employing a massive data set to train a machine learning algorithm is exhausting and requires a frequent update of the data set.

Therefore, the use of Reinforcement learning is being an attractive approach for resources management and service provision. A proposed policy iteration-based RL solution in [35] relies on dynamically selecting the most suitable configuration of VM resources to meet the service consumer requirement in QoS, while minimizing the overhead of resource allocation during a short time window in the vehicular cloud environment. The authors in [36], use a clustering-based ML along with Q-learning to select the cluster head (CH) and the vehicle service supplier (in the form of virtual machines (VMs)) based on their mobility, bandwidth, and computing capabilities in the vehicular cloud. A reinforcement learning solution combined with the Semi-Markov Decision Process (SMDP) is proposed in [37] for the provision of virtual units (VUs) to request vehicles based on the arrival and departure of vehicular requests in the vehicular cloud computing. The authors in [38] introduce a Deep Reinforcement Learning (DRL)-based mechanism for managing network resources in vehicular communication systems, emphasizing its utility in dynamic and complex environments. Their approach innovatively optimizes vehicle transmission power, IRS-reflection configurations, and BS detection coefficients, aiming to maximize network energy efficiency. This contribution showcases the potential of combining DRL with real-time network status analysis to enhance VANET performance.

Using the RL to manage cloud servers or even processing and storage of vehicle resources is different from those of vehicle sensors since the latter is characterized by its high heterogeneity and therefore more complexity in ensuring the virtualization. Moreover, almost the majority of works do not take into consideration the cost-effectiveness and the economics earned from providing vehicular services and resources. In addition, it should be noted that in applications of the existent works, the array of sensors is typically a fixed set and is generally presumed to be consistently available. However, in the context of mobile vehicular environments, this assumption does not hold due to the inherent dynamism and mobility, which can lead to variable sensor availability and changing configurations. This contrast underscores the need for more adaptive and flexible approaches to managing sensor resources in vehicular settings.

III. FOUR-LAYER IoV-CLOUD ARCHITECTURE FOR VEHICLE SENSOR PROVISIONING

This section first provides an overview of a four-layer architecture for sensor virtualization and provision. Then, the sensor virtualization model in the cloud and fog layers are specified, respectively. Subsequently, the optimization problem focusing on the sensor selection, and the overall cost minimization is formulated.

Vehicle sensor virtualization involves isolating the physical infrastructure of the vehicle from the application level by using a number of interface components. Therefore, we present a set of layers that involves interactive actors including the sensor suppliers (*SS*), the Sensors Cloud Service Provider (*SCSP*), and the Service Consumer (*SC*) and functional components including the fog nodes and the edge nodes. Moreover, we split the entire network into primary zones and designate the *SCSP* to control and manage one or many primary zones. The *SCSP* deploys a fog node at the level of each primary zone. Each primary zone is divided into equal cell zones. At each cell, an *RSU* is installed to collect the data generated by the vehicles and to track their mobility.

Application Layer: consists of the set of the services and applications requester such as the meteorological agencies (request for weather information including temperature, humidity, and rain level), telecommunication operators (demand drive testing samples to adjust and optimize the deployment of their equipment), and road monitoring authorities (request for the information related to the road traffic congestion and polluting gas emissions). These services and applications are created and served on-demand by the *SCSP* with respect to an SLA that is completed and signed by the *SC* and the *SCSP*. Therefore, the *SCSP* allocates sensors from mobile vehicles and provides vehicle virtual sensor services for the service consumer. Thereby, leveraging the geographical wide-spread and coverage of vehicles and the evolution of vehicle's on-board sensors. Moreover, the provision of the virtual sensor services can be a great alternative to overcome the high expensive cost incurred by the installation, the deployment, and maintenance of dedicated sensor infrastructure.

Edge Layer: The edge layer comprises the vehicular network consisting of mobile vehicles acting as sensor suppliers, the sensor network consisting of diverse and adaptable vehicle physical sensors embedded in vehicles' *OBU*s, and the Road Side Units (*RSU*s) that communicate with vehicles within their communication range under a cell zone. RSUs and vehicles communicate via the IEEE 802.11p standard, supporting dedicated short-range communication (DSRC) like vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communication. The vehicle physical sensor (ϕs) represents the key component of the proposed model, converting physical phenomena into processable electric signals, capturing driving behavior (e.g., speed, acceleration, and localization), and environmental data (e.g., pollution, temperature, and humidity). To optimize resource utilization, the vehicle physical sensors can remain idle until triggered or be programmed for on-demand sensing periods, allowing underused sensor resources to be utilized by cloud providers. Each vehicle physical sensor is identified by its ID, location, type, and configuration. A single sensor on a vehicle can allow multiple setups and handle numerous requests simultaneously. On the other hand, the *RSU* collects mobility information from vehicle physical sensors within its communication range and transfers it to the upper fog layer of the relevant primary zone for processing.

Cloud Layer: leveraging the almost limitless storage and processing capabilities it can provide, the Sensors Cloud Service provider (*SCSP*) uses the cloud computing resources to orchestrate and manage the fog nodes as well as the physical sensors that are leased from mobile vehicles. Therefore, it can offer on-demand and elastic virtual sensor services and applications. The cloud layer is used at the top of the fog layers. The vehicle virtual sensor represents a software program or pure logical object that receives a collection of inputs about the service type, service period, and geographic zone. A continuous service delivery is achieved thanks to the use of virtual sensors (*VS*) during the service duration. To deliver continually, on-demand and scalable services, the *VS* offers an abstraction layer to separate the diverse and adaptable physical sensors from their original physical ID and functionality. Due to its rapid mobility and predilection for following less congested paths, a single-vehicle physical sensor might not be able to offer an exhaustive data collection throughout the course of the whole service term. Therefore, the *VS* is applied to overcome the aforementioned constraint by dynamically choosing a collection of the most available vehicle physical sensors (i.e., one selected sensor per time slot).

Fog Layer: We leverage virtualization technologies to virtualize vehicle-mounted sensors on the fog layer, offloading computational tasks from resource-constrained terminals. This enables the development of efficient virtual sensor services while optimizing sensor resource utilization. Fog computing supports mobility and location awareness, which are lacking in commercial cloud computing models [39]. Our proposed model deploys fog nodes in each primary zone of the controlled area, allowing end users to connect to multiple fog nodes using different Application Level User (ALU) interfaces. The goal is to handle as many tasks as possible in the fog server, reducing vehicle energy consumption. The fog node consists of computing and storage servers, along with an intelligent component that collects and manages information from vehicle sensors and RSUs at the edge layer. It performs initial virtualization steps, including mobility data extraction. The fog node manages and collects data from vehicles in a geographical area, performing operations and configuration mapping based on the sensor's capabilities and consumer needs. These operations are executed by software execution on the fog node, managed by a distributed orchestration system.

IV. SENSOR VIRTUALIZATION IN THE CONTEXT OF IoV

The proposed virtualization consists in providing an abstraction layer by executing processing functions over vehicle physical sensors and achieving an intelligent vehicle physical sensor selection. In other words, it is about adding an abstraction layer between the vehicle physical sensor (including its type, configuration and also the information related to the vehicle embedding the sensor such as its location, speed, and

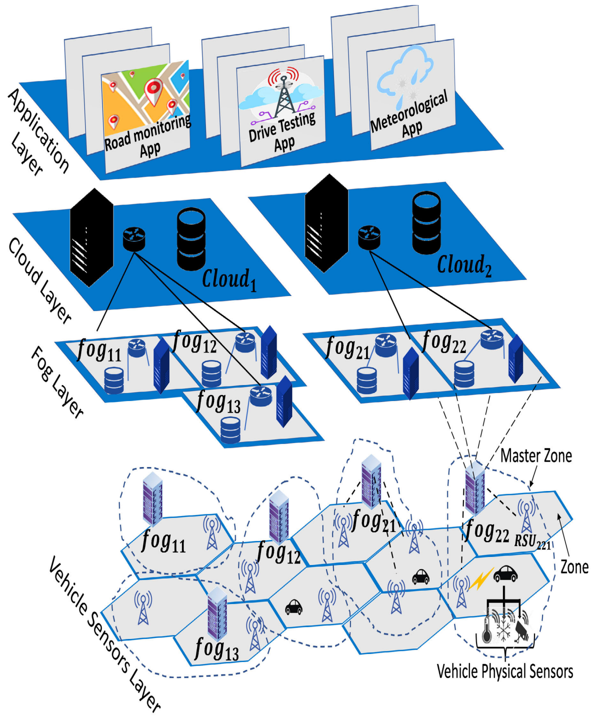


FIGURE 1. Cloud IoV architecture.

availability in the network) and the deliverable service to the service consumer. In this way, calling for virtualization allows hiding the sensor provision complexity from the service level consumers by defining a clear abstract layer. In the following, the advantages of vehicle sensor virtualization are presented. Then, the two virtualization steps are detailed.

A. ADVANTAGES OF VEHICLE PHYSICAL SENSORS VIRTUALIZATION

The virtualization technology has brought a lot of opportunities due to the advantages offered by it in many research fields and industrial disciplines. In this work, we describe the advantages and benefits of sensor virtualization in the IoV context.

1) UNDER-USED SENSOR RESOURCE EXPLOITATION AND SENSOR INFRASTRUCTURE SHARING

Exploiting under-used sensor resources and sharing sensor infrastructure are key benefits of virtualization. By sharing the physical infrastructure of sensors among multiple service consumers, virtualization optimizes sensor resources, reduces costs, and ensures maximum satisfaction for a larger user base. Typically, the physical sensors installed in vehicle onboard units operate in event-driven, periodic, or on-demand modes, resulting in significant under-utilization. Virtualization addresses this issue by scheduling tasks for under-used sensors through an abstraction layer, providing appropriate functions and operations. Furthermore, sensor reuse through virtualization minimizes network overhead and improves device efficiency, enabling the serving of multiple applications and services.

2) VIRTUALIZATION FOR SENSOR SERVICE MANAGEABILITY

Cooperative vehicles approach for distributed crowd-sourcing and task accomplishment may not be the appropriate solution due to potential conflicts of interest among vehicles when selecting service or task providers. However, performing virtualization in a separate abstraction layer by a neutral side (e.g., Broker, Sensor Cloud Service Provider) allows a better manageability of sensor service. Thereby, decoupling services from sensor infrastructure (its software and hardware) without need to know the technical details of the sensor involved in the virtual sensor composition and for the provision of new applications and services. Furthermore, the virtualization enables the concealment of sensor provision complexity from service consumers, requiring only minimal intervention from the sensor supplier.

3) VIRTUALIZATION FOR SENSOR SERVICE ELASTICITY AND SCALABILITY

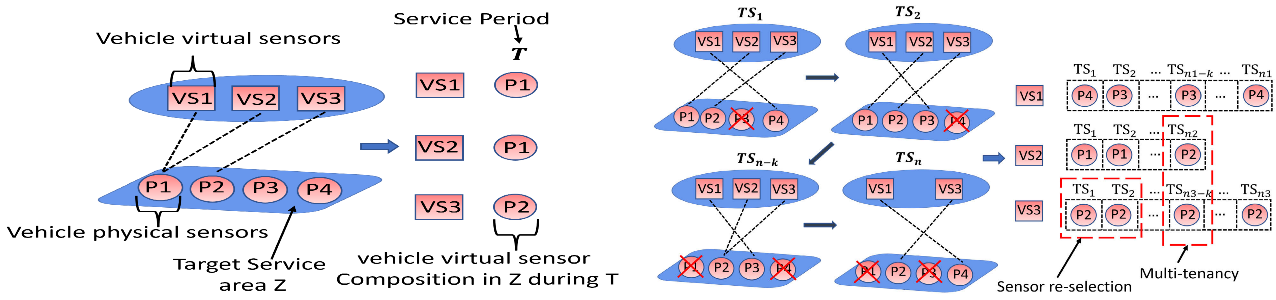
By applying sensor virtualization it becomes possible to ensure the sensor service scalability by maintaining the performance of the virtual sensor when expanding the target service zone, the service period, the number of requests, and also the number of service suppliers to be involved in the virtual sensor composition (i.e., the number of sensors per virtual sensor). Furthermore, it allows the sensor service elasticity by making it possible a dynamic, adaptive, and on-demand adjustability of the virtual sensor's parameters throughout its execution without affecting the system's performance.

4) NEW BUSINESS MODEL CREATION FOR INCREASING PROFITABILITY

The introduction of virtualization in vehicle sensor applications opens up new avenues for revenue generation and the creation of a business model. Third-party brokers can leverage these applications and services to generate additional income. This new business model involves various roles, including Sensors Cloud Service Providers (SCSP), Service Consumers,(SC) and Sensors Suppliers (SS), each benefiting from different opportunities. Sensors Cloud Service Providers focus on increasing profitability and reducing sensor allocation prices, while Service Consumers aim for high service level satisfaction and meeting quality of service requirements, such as service continuity. The intervention of Sensor Suppliers is limited to providing their mobility information and supplying sensors to the concerned SCSP. Moreover, increased profitability may result from simultaneously sharing the same vehicle physical sensor among many vehicle virtual sensor applications by executing the suitable operations for the selection of the most available sensor suppliers during the service provision.

5) VIRTUAL SENSOR SERVICE AVAILABILITY

By executing sensor selection and data pre-processing operations separately from the vehicle's physical sensor



(a) Sensor virtualization (b) Proposed vehicle virtual sensor virtualization

FIGURE 2. **Concept of the proposed vehicle virtual sensor.**

infrastructure at a higher layer, the availability of virtual sensor services can be improved. This approach provides a clearer overview of the vehicles' mobility information, leading to enhanced predictability regarding their availability in the network.

B. SENSOR CONFIGURATION MAPPING AND SENSOR RE-CONFIGURATION

The capacity to select sensor nodes for application activities is a key necessity. It is crucial to select suitable sensor nodes considering that a single-deployed vehicle physical sensor can be used by numerous applications simultaneously, since the majority of these applications may have specific spatial and temporal requirements. Thereby, continuously collecting sensor data from heterogeneous and mobile vehicle sensors. The heterogeneity of sensors is observed when considering the diversity of sensor types such as the embedded hardware (concerns the observable, physical parts of the sensor system) and the installed software (refers to the set of instructions which enable the hardware to execute the assigned tasks and processes). The sensor hardware is responsible for performing the physical capability of the sensor in terms of sensitivity, accuracy, and precision. The sensor software is responsible for defining the sensor configuration and Protocol implementation (e.g., protocol of communication with the on-board vehicle units and inter-vehicle components and interfaces, data reporting modes including event-driven, periodic, and on-demand, and data coding/decoding). On the other side, the mobility of vehicles embedding the physical sensors represents a major issue in the virtual sensors provision due to the high variation of the vehicles' speed and driving behavior, and also a cause of data collection errors and mobility information losses during transmission. Therefore, we propose a software-based sensor configuration to be executed through a secure interface beyond the physical sensor on top of the fog node. The software-based sensor configuration allows the creation of multiple end-to-end virtual sensors on top of the same vehicle physical sensor, so that each configuration can be optimized to the requirements (e.g., precision, sensitivity, resolution, and availability) of specific services and application domains (e.g., public road monitoring, industrial). Each virtual sensor can then be allocated to a different tenant (e.g., a communication provider, a mobile virtual network operator (MVNO)), that can use it to provide services to its own service consumers.

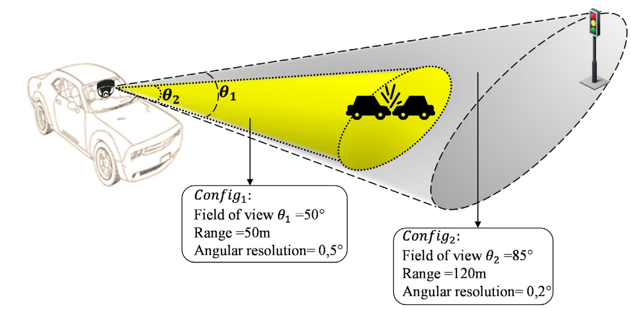


FIGURE 3. **Sensor's configuration mapping.**

Let $C_1, C_2, \ldots, C_p$ be the set of sensor configurations available for use. A configuration can be for instance the camera resolution when streaming or imaging, or the location precision for some road monitoring tasks. We assume that configurations have discrete values and can be compared to each others forming an order relation. We have $C_x \succeq C_y$ if and only if the data generated by a sensor with configuration C_x can be transformed, leading to the same data generated when the configuration C_y is applied. In practice C_x and C_y can represent two resolutions of a sensor camera. The configuration C_x allows the camera to generate photos with a higher resolution than C_y. In that case, we say that a mapping from C_x to C_y can be done using software tools, allowing the same sensor camera to simultaneously serve two users with two requested configurations C_x and C_y. We admit that a Service Consumer (SC), when he requests a configuration C_y can be served either by a sensor with configuration (C_y) while calling for software mapping, or a sensor with a configuration C_x (i.e., $C_x \succeq C_y$).

For example, given a Sensor Supplier (SS) which has a first sensor configured with C_x and a second sensor configured with C_z, and given the set $\{C_x;\ C_y;\ C_z\}$ of all possible configurations in the market. Assuming that ($C_x \succeq C_y \succeq C_z$), one can note that configuration C_z can be served either directly by the second sensor or by the first sensor through configuration mapping. Similarly, while there is no device configured with C_y, the first device configured with C_x can serve C_y and also C_z through configuration mapping. Thus, the *SCSP* is able to allocate the same device for different applications that request compatible configurations simultaneously and raises its profit. Also, he can reconfigure the device if it is idle to directly satisfy the request.

As illustrated in Figure 3, it is depicted that a single dash-cam installed on a vehicle offers the flexibility to adapt its lens for two different configurations catering to distinct applications. Configuration $Config_1$ is specifically designed for collision detection within a smaller coverage range. On the other hand, Configuration $Config_2$ is optimized for detecting traffic light situations, providing a superior angular resolution (i.e., capable of capturing more detailed information as the angular resolution decreases), an extended range, and a broader field of view.Thus, we say $Config_1 \succeq Config_2$.

In most existing works, the virtualization consists in selecting the allocated physical sensor during the whole service period T. In that case, the virtualization is limited in decoupling the required data type from the allocated physical sensor by considering only idle (the case of $P3$) or active (the case of $P1$ and $P2$) states of the sensor, as shown in Figure 2. While virtualization can effectively utilize underused sensor resources and enable the development of new services and applications, it faces limitations in providing continuous sensor service delivery. This is primarily due to the inherent mobility of vehicles, which introduces uncertainty regarding the availability of sensors in the network.

Hence, our proposition entails a dynamic virtualization approach that leverages the selection of an appropriate physical sensor from a set of vehicles in the target service area Z for each time slot. Illustrated in Figure 2b, $VS1$ is initially formed by the inclusion of the vehicle physical sensor $P4$ at TS_1. However, at TS_2, $P4$ becomes unavailable in zone Z, prompting the selection of $P3$ to be part of $VS1$ until TS_{n1}, at which $P4$ becomes available in zone Z once again. We assume, in this example, that both $VS2$ and $VS3$ request the same sensor configuration C_a (C_a is allowed by $P1$ and $P2$) and $VS3$ request the sensor configuration $C_b(C_b$ is allowed by $P3$ and $P4$), as shown in Figure 2b.

V. MODELING VEHICLE VIRTUAL SENSOR

We propose a sensor virtualization that consists in continuously involving a different vehicle physical sensor depending on its location and cost to meet the users' service requirements in terms of service continuity.

A. VEHICLE SENSOR VIRTUALIZATION AND COST COMPUTATION

The vehicle sensor virtualization consists in appending an abstraction layer between the ϕs, such as its ID and location, and the type of data. In other words, the *SC* customizes the service without being aware of the ID or the location of the service supplier (*SS*). Therefore, the *SCSP* dynamically selects a new ϕs at a same time slot i, when the older one (selected at the time slot $i-1$) becomes unavailable in the target service zone.

Assuming that the quality of information is guaranteed by using solely one sensor per time unit, it becomes imperative to predict the availability of ϕs and therefore the continuity of the service delivery. We define a virtual sensor (*VS*) by 6-tuple attributes, $VS =< T, Z, Data_{in}, PREP, Data_{out}, COMP >$, where $T =< TS_1, TS_2, \ldots, TS_n >$ denotes the required service period (divided into equal time-slots TS), Z is the target service zone (it is assumed that a service is maintained on the coverage of the target service zone Z which is composed of a set of cells controlled by the *SCSP* [40]), $Data_{in}$ is the input data to be collected from a physical sensor ϕs before being pre-processed, $PREP$ is the set of functions and operations performed on $Data_{in}$, $Data_{out}$ is the output data after being processed, and $COMP(VS, Z, T)$ is the function that realizes the spatio-temporal mapping to select one physical sensor ϕs per time slot TS such that:

$$\begin{aligned} COMP(VS, Z, T) = &<Map(VS, Z, TS_1), \\ &Map(VS, Z, TS_2), \ldots, Map(VS, TS_n)> \\ &= <\phi s_1, \phi s_2, \ldots, \phi s_n> \\ &\forall \phi s_i \in \mathcal{J} \text{and } i \in [1, 2, \ldots, n] \end{aligned} \tag{1}$$

where $Map(VS, Z, TS_i)$ is the function that maps the physical sensor ϕs (from a set of available vehicle physical sensors $\mathcal{J}$ registered with *SCSP* and located in Z) to the *VS* at the time slot i so that $location(\phi s_i, TS_i) = Z$. Note that the same ϕs_i can be selected for many consecutive time slots ($\phi s_i = \phi s_{i-1}$), especially as the physical sensor may remain in Z during TS_i and TS_{i-1} and the *SCSP* needs to minimize the cost of sensor provision. Similarly, we define a physical sensor ϕs by 4-tuple attributes $\phi s = <ID, SCSP, Data_{out}, cost>$, where *ID* stands for the identity of ϕs in the network, *SCSP* is the provider registered with it (a vehicle physical sensor is registered with only one broker), $Data_{out}$ is data type generated by the physical sensor that will feed the virtual sensor, and *cost* is its allocation cost which is dependent on the sensor type and the provided data quality (e.g., accuracy, integrity, and consistency).

Optimizing both the profitability of the *SCSP* and the cost-effectiveness for the *SC* relies on the consistency of the decision made to select the most available physical sensor with the lowest allocation cost. In this work, the *SCSP* aims to dynamically select the most available vehicle physical sensor at each time slot (*TS*) based on its stay time in the target service zone Z. We define η_i as a binary selection indicator to indicate whether the selected vehicle physical sensor ϕs_i is available at TS_i. If it is available, the *SCSP* pays only the unit allocation cost, $UCost(\phi s_i)$, for one-time slot of the vehicle physical sensor ϕs_i that is part of the virtual sensor (*VS*). Otherwise, a penalty (*Pe*) is charged for the false selection, and the $SCSP$ has to pay $(\eta_i + Pe) \cdot UCost(\phi s_i)$ for the SC.

$$\eta_i = \begin{cases} 1, & \text{if the selected } \phi s_i \text{ is available in } Z \text{ at } TS_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Moreover, the $SCSP$ seeks to reuse the same vehicle physical sensor, when it is allocated, to simultaneously compose multiple virtual sensors (VSs). We define ω_i as a binary indicator to indicate whether the selected vehicle physical sensor ϕs_i is reused.

$$\omega_i = \begin{cases} 1, & \text{if } Map(VS, Z, TS_i) \in \{COMP(VS_u, Z, TS_i), \\ & \quad COMP(VS_v, Z, TS_i)\}, \text{ for some } u \neq v \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Additionally, the $SCSP$ is motivated to re-select the same vehicle physical sensor during consecutive time slots to compose the VS to avoid the charges of allocating new vehicle physical sensors. We define ψ_i as a binary indicator to indicate whether the same vehicle physical sensor ϕs_i is selected during TS_i and TS_{i-1}. In particular, ψ_i is given by:

$$\psi_i = \begin{cases} 1, & \text{if } i = 1 \text{ or } Map(VS_u, Z, TS_i) \\ & \quad = Map(VS_u, Z, TS_{i-1}) \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

To minimize the global allocation cost of a virtual sensor (i.e., the cost related to the composition $COMP(VS, Z, T)$) during the entire service period T, we define the global cost $GCost(VS, T)$ of allocating a virtual sensor as follows:

$$GCost(VS, T) = \sum_{i=1}^{n} \omega_i \cdot ((\eta_i + Pe)^{1-\eta_i} \cdot UCost(\phi s_i) + (1 - \psi_i) \cdot f_{reconf}) \tag{5}$$

where, $UCost(\phi s_i)$ represents the unit allocation cost (for one time slot) of the vehicle physical sensor ϕs_i that is part of VS, and f_{reconf} is a fixed fee charged by the $SCSP$ every time the vehicle physical sensor is reconfigured to be reused for composing another vehicle virtual sensor.

To ensure efficient and secure accessibility to the fog resources, we apply the multi-tenancy architecture, allowing the same ϕs resources to be shared among multiple virtual sensors and multiple service consumers (SCs) that request the same configuration at the same TS. Each virtual sensor has access to the configuration table responsible for the adjustment of the sensor provided via secure APIs which are made available by the $SCSP$.

B. SERVICE REQUEST PROCESSING STEPS

The service request processing goes through different steps as explained in Figure 4:

The Road Side Units (RSUs), managed by a Sensor Cloud Service Provider (SCSP), are pivotal in instantly collecting mobility data from registered vehicles within their designated zone Z. These vehicles, acting as sensor suppliers, feed their mobility data to the RSUs, which is then transferred to a fog server for the initial processing. This stage includes essential operations like interpolating missing data and averaging values, in preparation for further analysis.

A Service Consumer (SC) initiates the process by submitting a detailed request to the SCSP, (r = <Type, T, Config, Z>), outlining the service type, period, desired parameter configurations, and the target service coverage zone. Upon receiving an acknowledgment of the request, the SCSP proceeds to instantiate a virtual vehicle sensor. Throughout the service period, divided into equal time slots, the SCSP dynamically requests and receives updated data from the fog server.

The SCSP evaluates all mobility data by applying Equations 6 and 7 to decide on vehicle candidates for sensor allocation. The decision-making the process follows a structured path based on availability, previous selections of vehicles, and simultaneous sensor sharing among multiple virtual sensors employing Equations 2, 3, and 4 to guide these decisions. If a selected vehicle (V_u) is available at the current timestep (TS), marked by ($\eta_u = 1$, as determined by Equation 2), and was neither selected in the preceding TS ($\psi_u = 0$, according to Equation 4) nor engaged in another virtual sensor composition at the current TS ($\omega_u = 0$, as per Equation 3), the SCSP requests V_u to allocate its sensor. Upon confirmation from V_u, the SCSP updates the virtual sensor composition by integrating the new physical sensor and configures it to meet the SC's requirements.

Alternatively, if V_u is already part of another virtual sensor composition at the same TS ($\omega_u = 1$), indicating that the same physical sensor is shared across different virtual sensors but with varied data configurations, the SCSP updates the virtual sensor accordingly. This flexibility allows for efficient use of resources.

In cases where V_u remains available ($\eta_u = 1$) at the current TS and had been selected in the previous TS ($\psi_u = 1$), indicating continuous allocation, the SCSP solely extends the allocation period, maintaining the sensor's composition and keeping the same previous TS's configuration.

However, if V_u becomes unavailable ($\eta_u = 0$), the SCSP releases the previously selected sensor and seeks a new available vehicle (V_w, where $w \neq u$) for sensor allocation. Following confirmation from V_w, the SCSP updates the virtual sensor by selecting the newly allocated physical sensor and applying the configurations on the data to meet the required service exigencies.

VI. REINFORCEMENT LEARNING FOR VEHICLE SENSOR SELECTION

In this section, we formulate the vehicle physical sensor selection under a Markov decision problem-solving. We designate the Reinforcement Learning as a technique that falls under machine learning technology to solve the vehicle physical sensor selection. The RL allows a software agent that has limited information about the environment to learn which

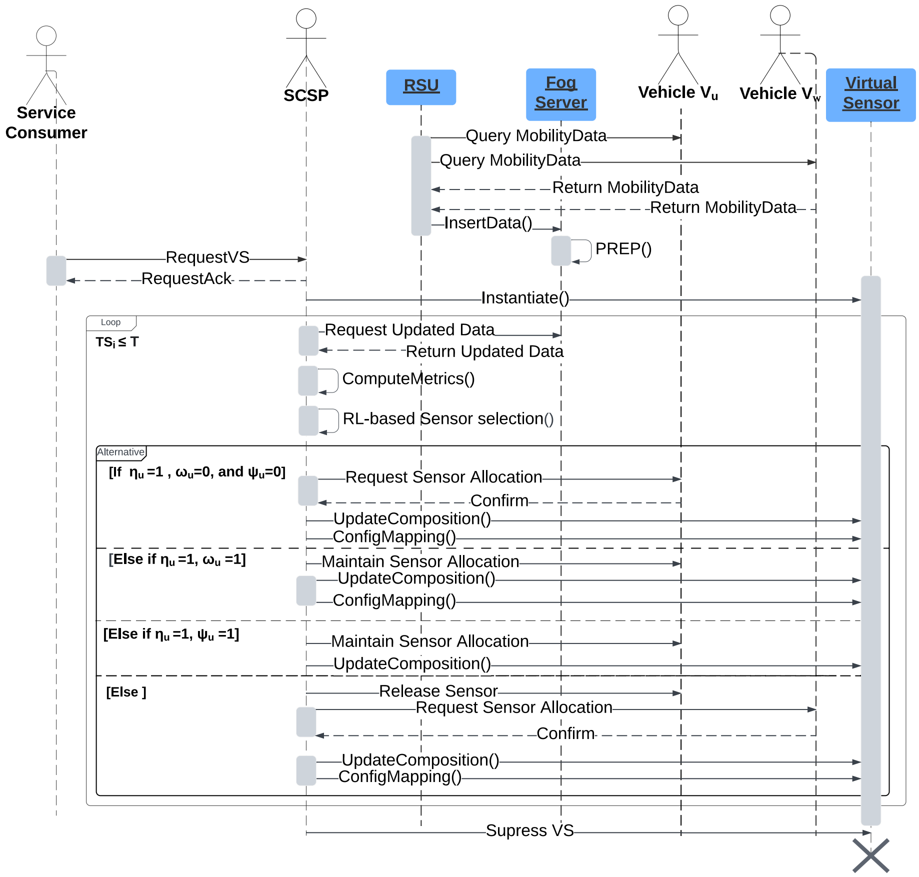


FIGURE 4. Sequence diagram of the vehicle virtual sensor management and provisioning.

action to take in order to react with the corresponding state. Therefore, we represent the set of states, action,s and rewards to enable the RL agent to interact with the mobile vehicular environment.

A. VIRTUAL SENSOR COMPOSITION AS A MARKOV DECISION PROCESS (MDP)

The proposed model consists in selecting the most suitable vehicle physical sensor ϕs at each time slot TS during the service period T. The selection is based on the mobility of the vehicles (as the sensor supplier (SS) including their positions in the network, speed, and time stayed in the target service area. The selection also considers the sharing of the same vehicle physical sensor ϕs by many vehicle virtual sensors simultaneously. We approximate the selection process of vehicle physical sensors in a target service area Z during T as a Markov Decision Process (MDP) to maximize both the decision certainty made on the availability of the vehicle physical sensor (ϕs) and the cost-effectiveness of the vehicle virtual sensor (VS) in a dynamic IoV environment. The MDP is defined by a tuple (S, A, T, R), where S, $(s \in S)$, is the state space, A, $(a \in A)$ is the action space, $T(s, a, s')$ is the transition probability between a state s and a state s' when executing an action a, and $R(s, a, s')$ is the immediate reward when an action a is taken under state s leads to a transition to state s'. Solving a problem of an MDP consists in finding the optimal policy allowing the mapping from states to actions while maximizing the cumulative reward. This purpose can be achieved by learning the transition probability T from state s to s', which is called model-based RL. However, the IoV environment is characterized by its high dynamics and incompleteness in the collection of mobility information which leads to a high complexity in calculating these probabilities. All these properties make the complexity of defining a consistent mobility pattern within a time period very enormous. Therefore, solving the MDP problem is NP-hard undoubtedly [41].

B. REINFORCEMENT LEARNING FOR SOLVING VIRTUAL SENSOR COMPOSITION PROBLEM

Reinforcement Learning has gained significant popularity as an MDP-based approach for developing agents that acquire knowledge through interactions with their environment. Unlike supervised learning, where explicit feedback is provided, RL methods rely on the agent's exploration of the environment and the immediate rewards it receives. This process enables the agent to learn optimal behavior without prior knowledge of the best actions for a given state. The

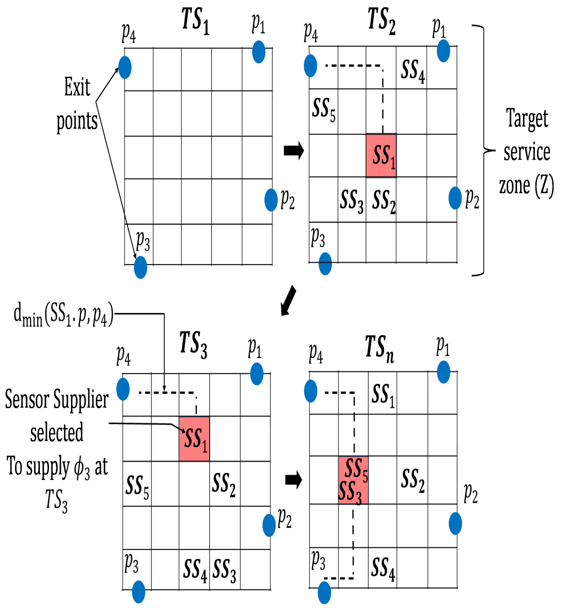


FIGURE 5. **vehicle physical sensor selection steps.**

necessity for known sensor positions is dictated by the system's architecture and specific application needs [42]

State: represents the dependencies related to the availability of the vehicle physical sensor ϕs_i to be involved in the VS's composition at the current time slot TS_i, in the target service zone Z. Moreover, we define the state as a combination of attributes that include the identity (id) of the cell in the target service zone Z (it is assumed that a zone Z is divided into a set of square cells) where the vehicle physical sensor is located, the quantified mean speed $\bar{v}$ of the vehicle SS_i (embedding the vehicle physical sensor ϕs), the remaining time Tr before the vehicle SS_i reaches the nearest exit point p_x in the zone Z, and the number of times q the same ϕs_i is selected to compose simultaneously many vehicle virtual sensors. We denote the set of states by $\mathbb{S}= \{s_1, s_2, \ldots, s_{|\mathbb{S}|}\}$, where $|\mathbb{S}|$ is the cardinality of $\mathbb{S}$ which counts the number of elements in the set $\mathbb{S}$. Thus, $|\mathbb{S}|$ is given by: $|\mathbb{S}| = H \times M \times TR \times Q$, where H, M, TR, and Q denote the set of cells' identities, the set of quantified mean speeds, the set of discretized remaining times, and the set of potential virtual sensors that can be simultaneously composed by the same vehicle physical sensor, respectively.

Action: represents the decision made over a given state. We define the action as the ID of the vehicle physical sensor ϕs to be selected to compose the vehicle virtual sensor VS at a time slot TS. Suppose there are n vehicles in the network, then the action set is denoted by, $\mathbb{A}= \{\phi s_1, \phi s_2, \ldots, \phi s_n\}$. In the selection process, we can meet a situation where more than one vehicle is situated in the same cell and therefore, the same distance is separating these vehicles and the closest exit point. Because only one vehicle is selected every time slot, the best action consists in selecting the vehicle whose speed belongs to the lowest speed interval.

Reward: is assigned to the action under the current state. Accordingly, we put either positive reward or negative discount, denoted by r, to the action under a given state.

We define three rewards r_1, r_2, and r_3. r_1 is defined with respect to Tr which is the lowest traveling period Tr of SS_i to move from its position p at the current time slot TS_i to the nearest exit point p_x. We have $Tr = \frac{d_{min}(SS_i.p,p_x)}{\bar{v}}$, where d_{min} is the shortest distance between $SS_i.p$ and p_x, and $\bar{v}$ is the weighted average speed of the vehicle SS. Moreover, the availability of the vehicle in the IoV context depends on its current location and its speed which is impacted by the density of the visited zone, the traffic signalization status and the incurred congestion in the cell c_h as shown in Figure 5. The $SCSP$ aims to select the vehicle that stays the longest period in the target service zone:

$$SI = \frac{Tr}{T_{thr}} = \frac{d_{min}(SS_i.p, p_x)}{\bar{v} \cdot T_{thr}} \tag{6}$$

where T_{thr} is a threshold value. We calculate the weighted average speed $\bar{v}$ of a sensor supplier (vehicle) SS_i at a time slot i starting from the average speed measured over a window L up to the time slot $i-1$ and its current speed v at time slot i.

$$\bar{v}_i = \sigma \bar{v}_{[i-L,\, i-1]} + (1-\sigma) v_i \tag{7}$$

Thus, the related reward r_1 is obtained as:

$$r_1 = \begin{cases} +\delta, & if\ SI \geq 1 \\ -\delta, & if\ SI < 1 \end{cases} \tag{8}$$

where SI stands for the satisfactory indicator that represents the normalized measure of the remaining time period (before the vehicle SS_i reaches the nearest exit point with respect to its current position p at TS_i) and the availability period threshold.

The second reward r_2 is defined to measure the overuse of the same ϕs, simultaneously by different VSs as follows:

$$r_2 = \begin{cases} +e^q, & if\ q > 0 \\ -\theta \cdot e^q, & if\ q = 0 \end{cases} \quad , \forall\, q \in \{0, 1, \ldots\} \tag{9}$$

where q is the number of already available vehicle virtual sensors that simultaneously share the same ϕs_i and θ is a penalty weight. Moreover, we define r_2 to motivate the sharing of the same ϕs_i simultaneously by many VSs and therefore to maximize the utilization of the vehicle physical sensors.

The third reward r_3 is defined to motivate the minimization of the number of vehicle physical sensor modifications from a time slot to another during T when feeding the same vehicle virtual sensor VS with data. The reward r_3 consists in compensating the re-selection of the same vehicle physical sensor ϕs_i during two consecutive time slots.

$$r_3 = \begin{cases} +\beta, & \phi s_i = \phi s_{i-1} \\ -\beta, & otherwise \end{cases} \tag{10}$$

We define the current reward expression $R = r_1 + r_2 + r_3$ as the resulting reward. The agent seeks to find the best combinations of state-reward pairs, after sufficient training.

The agent seeks to find the best combinations of state-reward pairs. After sufficient training, the one-by-one selection leads to the maximal accumulative rewards $R_{acc} = \sum_{i=1}^{n} R_i$ over the service period T composed of n time slots.

Value-Based Reinforcement Learning:

In Value-Based Reinforcement Learning, the state value function V plays a crucial role in evaluating the quality of a state for the agent. It estimates the value or expected return, of being in a particular state s under a given policy π as:

$$V^{\pi}(s) = \mathbb{E}[R\,|s, \pi] \tag{11}$$

The optimal state value function $V^*(s)$ represents the maximum state value function achievable under a specific policy for all states. It captures the highest expected return that can be attained by following the optimal policy.

$$V^*(s) = \max_{\pi} V^{\pi}(s), \ \forall s \in S \tag{12}$$

To consider the effect of action value, a state-action value function (*Q-function*) is used to estimate the expected return when taking a specific action a under a given state s. It considers the potential rewards and future states resulting from that action as:

$$Q^{\pi}(s, a) = \mathbb{E}[R\,|s, a, \pi] \tag{13}$$

The optimal *Q-function*, is computed in a similar manner to the optimal state value function. It involves maximizing the expected return over all states. The relation between the optimal state value function and the optimal action-value function is expressed as follows:

$$V^*(s) = \max_{a} Q^*(s, a), \ \forall s \in S \tag{14}$$

The optimal $Q^*(s, a)$, results the optimal policy π^* by selecting the action a that maximizes the *Q-value* for a given state s. In fact, $Q^*(s, a)$ guides the decision-making process by identifying the action that yields the highest *Q-value*, thereby leading to the optimal policy selection.

$$\pi^* = \underset{a}{argmax}\, Q^*(s, a), \ \forall s \in S \tag{15}$$

The *Q-function* is exploited to update the reward score for each action $a \in \mathbb{A}$ under a state $s \in \mathbb{S}$ to fill the *Q-table* via the recursive nature of Bellman equations using the Markov property as follows:

$$Q^{\pi}(s, a) \leftarrow Q^{\pi}(s, a) + \alpha\,(target_{RL} - Q^{\pi}(s, a)) \tag{16}$$

where $\alpha \in [0, 1]$ is the learning rate that specifies to what extent the newly acquired information overrides the old information.

Exploration vs Exploitation:

In line with the existing literature, it is necessary to consider that if we keep selecting the action with the highest reward score in the *Q-table*, the algorithm may obtain a local optimum. To remedy this problem, we need to explore during learning, (i.e., sometimes try actions rather than the best one). We therefore use the ϵ-*greedy* algorithm to perform action selection. Specifically, under a certain state, we select the best action according to the *Q-table* with probability $1 - \epsilon$ and randomly select one of the other actions with probability ϵ. At the beginning of the learning process, we set a relatively large value of ϵ so that we can try more; then, as the learning process progresses, we gradually reduce the value until the *Q-table* is converged.

1) Q-LEARNING-BASED ALGORITHM FOR THE SELECTION OF VEHICLE PHYSICAL SENSOR

Q-learning, as a model-free RL, is one of the fundamental algorithms to deal with the MDP problems. It relies on using trial and error method to cope with the dynamic vehicle network environment changing and the incompleteness of information related to vehicle mobility (e.g., speed, trajectory, and state). Motivated by the advantages offered by it, we adopt Q-learning to select vehicle physical sensors (embedded in vehicles) at every time slot to guarantee continuous service delivery while minimizing the allocation cost.

Thereby, leading the agent to learn a state value function V. Q-learning is an off-policy learning algorithm that updates the agent's actions by maximizing Q-values across the available action space. Q-learning is considered as an off-policy learning algorithm that updates the agent's actions by maximizing *Q-values* across the available action space based on the target, $target_{Q-learning} = R + \gamma \max_{a} Q^{\pi}(s', a)$, where $\gamma \in [0, 1]$ is the discount factor indicating the myopic view of the Q-learning regarding the future reward.

We propose a Q-learning-based algorithm for the selection of the sequence of vehicle sensors to be involved in the vehicle virtual sensor composition, as shown in Algorithm 1. The algorithm gets as input the request r_x received from the Service Consumer (SC) including the type of service ($Type$), its configuration $Config$, the service period T, the target service zone Z, and also the set of exit points $\bigcup\{p_x\}$. It outputs the vehicle virtual sensor composition $COMP(VS, Z, T)$ as a sequence of vehicle sensors ϕs_i at each TS_i during the service period T with respect to the requirements of r_x. The agent starts by updating the set $\mathcal{J}$ of sensor candidates to exclude the sensors that exit the target service zone Z from being considered in the training. The Q-learning agent starts each episode by initializing the state s. For each step of the episode, the agent selects an action a based on the current state s and uses a policy that is derived from the Q-values. Then, it executes the chosen action a in the environment, receives the reward R associated with taking that action and transitions to the next state s'. The new state s' represents the environment's response to the action taken. Subsequently, it selects an action a from the new state s' using a greedy policy (with the highest Q-value) and updates the Q-value for the current state-action pair $Q(s, a)$ by applying the Bellman equation (*Q-function*). Next, the agent updates the current state s to the new state s' for the next iteration. At the end of each step, the obtained reward R is being used later for the accumulative reward R_{acc}. Simultaneously, R_{acc} is used to decide on the optimal VS's sequence and also employed with itr_{max} as a terminal condition.

Algorithm 1 Q-Learning Algorithm

Input: $r_x =< Type, T, Config, Z >, \bigcup\{p_x\}$
Output: $COMP(VS, Z, T) =< \phi s_1, \phi s_2, \ldots, \phi s_n >, \forall \phi s \in \mathcal{J}$
Initialization: itr_{max} and $Q(s, a), \forall s \in \mathbb{S}, a \in \mathbb{A}$
Update the set $\mathcal{J}$ of ϕs candidates
Repeat (for each episode)
Initialize $s_0 = [id, \bar{v}, Tr, q]$
for each step of episode
Choose a from s using policy derived from Q (using ϵ-*greedy*)
Take action a, receive R, and observe s'
Choose a from s' using greedy policy (*i.e.*, $\epsilon = 0$), subject to $\max_a Q^{\pi}(s', a)$
Update *Q-value* by applying *Q-function*,
$Q^{\pi}(s, a) \leftarrow Q^{\pi}(s, a) + \alpha(target_{Q-learning} - Q^{\pi}(s, a))$
$s \leftarrow s'$
end for
Compute R_{acc}
until $(R_{acc,iter} - R_{acc,iter-1} < \varepsilon)$ or (itr_{max} is reached)

2) SARSA-BASED ALGORITHM FOR THE SELECTION OF VEHICLE PHYSICAL SENSOR

In value-based RL, it exists SARSA algorithm which is classified as an on-policy RL method, thereby updating the agent's actions based on the policy π, which is derived from the *Q-values* based on the target, $target_{SARSA} = R + \gamma Q^{\pi}(s', a')$ and via the Bellman equation as: $Q^{\pi}(s, a) \leftarrow Q^{\pi}(s, a) + \alpha(target_{SARSA} - Q^{\pi}(s, a))$. The SARSA algorithm (presented in Algorithm 2) begins each episode by initializing the state s. At each step of the episode, the agent chooses an action a based on the current state s using a policy derived from the Q-values using ϵ-*greedy*. The selected action a is then executed in the environment, resulting in a reward R and a transition to the next state s'. This updated state s' represents the environment's response to the action taken. Subsequently, the agent chooses the next action a' from the new state s' using a policy based on the Q-values, often employing an ϵ-*greedy* strategy. The Q-value for the current state-action pair $Q(s, a)$ is updated via the Bellman equation. The current state s is then replaced with the new state s', and the current action a is replaced with the next action a' for the subsequent iteration. At the end of each step, the obtained reward R contributes to the accumulated reward R_{acc}. The convergence of the SARSA algorithm and the termination condition is determined by evaluating the difference between consecutive accumulated rewards R_{acc} against a predefined threshold or by reaching the maximum number of iterations itr_{max}.

Algorithm 2 SARSA Algorithm

Input: $r_x =< Type, T, Config, Z >, \bigcup\{p_x\}$
Output: $COMP(VS, Z, T) =< \phi s_1, \phi s_2, \ldots, \phi s_n >, \forall \phi s \in \mathcal{J}$
Initialization: itr_{max} and $Q(s, a), \forall s \in \mathbb{S}, a \in \mathbb{A}$
Update the set $\mathcal{J}$ of ϕs candidates
Repeat (for each episode)
Initialize $s_0 = [id, \bar{v}, Tr, q]$
Choose a from s using policy derived from Q (using ϵ-*greedy*)
for each step of episode
Take action a, receive R, and observe s'
Choose a' from s' using policy derived from Q (using ϵ-*greedy*)
Update *Q-value* by applying *Q-function*:
$Q^{\pi}(s, a) \leftarrow Q^{\pi}(s, a) + \alpha\ (target_{SARSA} - Q^{\pi}(s, a))$
$s \leftarrow s', a \leftarrow a'$
end for
Compute R_{acc}
until $(R_{acc,iter} - R_{acc,iter-1} < \varepsilon)$ or (itr_{max} is reached)

The main difference between Q-learning and SARSA lies in their action selection and *Q-value* update strategies. In Q-learning, action selection is performed at each step within an episode using a policy derived from the *Q-values*. Q-learning aims to learn the optimal action-value function by selecting actions based on the maximum *Q-value* of the next state. On the other hand, SARSA performs action selection once per episode using a policy derived from the *Q-values*. SARSA updates *Q-values* based on the *Q-value* of the next state-action pair resulting from the action selected under the policy. Another distinction is that SARSA updates both the state and action variables while Q-learning only updates the state variable.

VII. RESULTS AND DISCUSSION

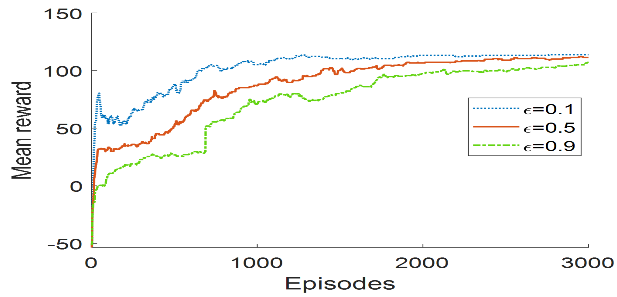
We conducted simulations to evaluate the performance of our model, employing Simulation of Urban Mobility (SUMO) to generate mobility trace datasets and MATLAB for executing the proposed reinforcement learning algorithms. Our study focuses on a vehicular grid network within an urban topology consisting of two-way roads, 25 intersections, 80 edges (represents a one-way connection between two intersections and defines a potential route for vehicles in the simulation). The simulation involves 100 vehicles traveling with an average speed of 13.8 m/s, operating within a service zone covering an area of 0.64 km^2.

A. CONVERGENCE SPEED COMPARISON BETWEEN Q-LEARNING AND SARSA ALGORITHMS WITH RESPECT TO THE EXPLORATION PROBABILITY ϵ

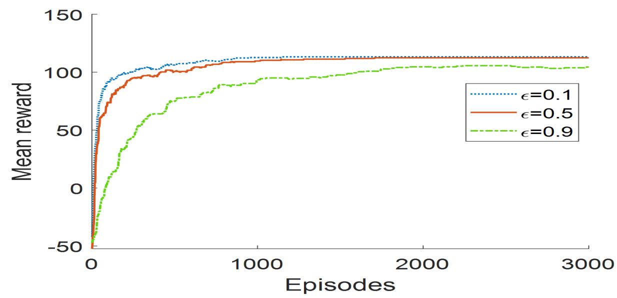
In Reinforcement Learning (RL), model convergence refers to the process where the learning algorithm reaches a point where further learning does not significantly change the policy or the value function. This implies that the algorithm has learned an optimal or near-optimal policy that maximizes the expected cumulative reward over time, given the environment dynamics and the reward structure. Thus, we evaluated the convergence of the mean reward with respect to the number of episodes, as depicted in Figure 6. Therefore, we set the discount factor γ to 0.7 and the learning rate α to 0.2. To assess the impact of exploration versus exploitation for both Q-learning and SARSA, we varied the exploration probability ϵ from 0.1 to 0.9 as illustrated in Figures 6a and 6b, respectively.The analysis of the Q-learning algorithm's performance, with varying ϵ values offers deep insights into the impact of exploration levels on learning outcomes. Starting uniformly across all strategies, the mean rewards soon diverge significantly. A conservative exploration strategy ($\epsilon = 0.1$) leads to a stable and efficient learning curve, peaking at a mean reward of 113.8, highlighting the benefits of a gradual approach that favors known good actions. Meanwhile, a balanced strategy ($\epsilon = 0.5$) achieves a harmonious exploration-exploitation the trade-off, resulting in consistent progress and a final mean reward of 111.4. On the other hand, an aggressive exploration approach ($\epsilon = 0.9$) initially delays convergence, as evidenced by a modest mean reward early on, but eventually secures a significant learning outcome with a final mean reward of 107.2. This analysis underscores the nuanced trade-offs between exploration and exploitation in reinforcement learning, emphasizing the importance of strategic ϵ selection for optimizing Q-learning performance.

The SARSA algorithm's exploration strategies exhibit distinct patterns in learning dynamics across different epsilon values. Initially, the mean rewards start at -42.0 for $\epsilon = 0.1$, -52.4 for $\epsilon = 0.5$, and -47.4 for $\epsilon = 0.9$, reflecting the algorithm's early exploration phase. As the episodes progress, by the 50th episode, $\epsilon = 0.1$ shows a promising increase in mean rewards to 74.6, indicating efficient and stable learning. $\epsilon = 0.5$ progresses to a mean reward of 60.6, demonstrating a balanced approach to learning. In contrast, $\epsilon = 0.9$, with its aggressive exploration, shows a decrease in mean reward to -14.6, suggesting that higher exploration initially hampers convergence. However, towards the later stages, stability is observed as the mean rewards for $\epsilon = 0.1$ reach 113.2, $\epsilon = 0.5$ at 112.4, and $\epsilon = 0.9$ improves significantly to 104.4, showcasing the algorithm's capacity to learn and adapt over time. This analysis highlights the nuanced impact of epsilon values on the SARSA algorithm's learning speed and stability, emphasizing the importance of strategic epsilon selection for optimizing performance.

Both Q-learning and SARSA algorithms demonstrate sensitivity to ϵ -driven exploration strategies, impacting convergence and reward optimization. Q-learning, with its off-policy nature, slightly outperforms SARSA, an on-policy algorithm, under aggressive exploration ($\epsilon = 0.9$), due to its ability to evaluate policies independently of the agent's actions. This distinction highlights the critical role of ϵ in balancing exploration-exploitation trade-offs and underscores the importance of algorithm selection based on the specific dynamics of the learning environment.



(a) Impact of ϵ variation on the Q-learning model convergence

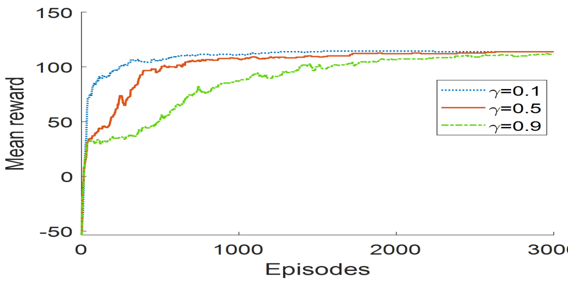


(b) Impact of ϵ variation on the SARSA model convergence

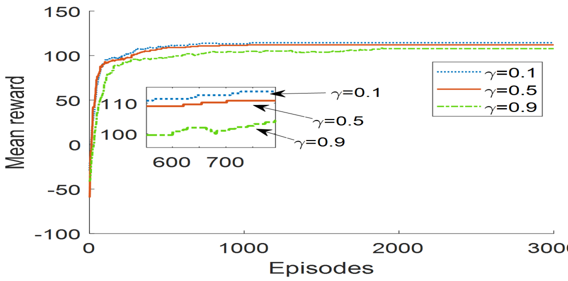
FIGURE 6. Convergence speed comparison between Q-learning and SARSA w.r.t the ϵ parameter.

B. CONVERGENCE SPEED COMPARISON BETWEEN Q-LEARNING AND SARSA WITH RESPECT TO THE DISCOUNT FACTOR PARAMETER γ

We conducted a simulation to evaluate the influence of the discount factor γ on the convergence of both Q-learning and SARSA in terms of mean reward as shown in Figure 7. The analysis of Q-learning convergence across γ values of 0.1, 0.5, and 0.9 (as shown in Figures 7a), supported by mean rewards and convergence metrics, revealing distinct learning dynamics. $\gamma = 0.1$ converges quickly and stably, indicated by a convergence start at episode 1355 and a mean reward of approximately 113.80, highlighting a preference for immediate rewards but potentially sacrificing the pursuit of higher long-term rewards. $\gamma = 0.5$, starting its convergence around episode 1660 with a similar mean reward, suggests a balanced exploration-exploitation strategy, leading to an effective yet slightly delayed convergence compared to $\gamma = 0.1$. Meanwhile, $\gamma = 0.9$, with its convergence beginning later at episode 2111 and a slightly lower mean reward of 110.74, demonstrates a strategy that values future rewards more, leading to slower convergence and a less stable learning process. These findings underscore the trade-offs between convergence speed, stability, and reward optimization in Q-learning, influenced significantly by the choice of γ value. Analyzing the SARSA algorithm's convergence across γ values of 0.1, 0.5, and 0.9 (as shown in Figure 7b), considering the provided mean reward curves, reveals unique insights into its learning behavior. Like Q-learning, lower γ

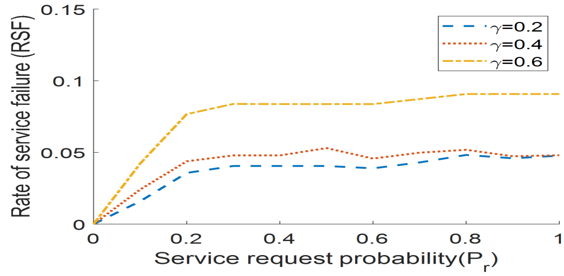


(a) Impact of γ variation on the Q-learning convergence

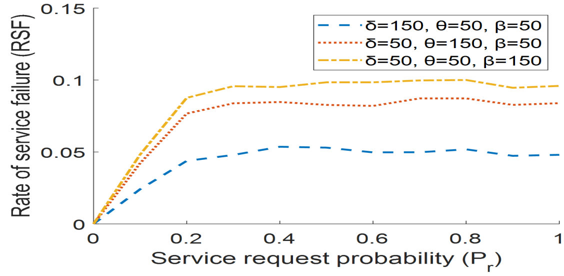


(b) Impact of γ variation on the SARSA model convergence

FIGURE 7. Convergence speed comparison between Q-learning and SARSA w.r.t the γ parameter.



(a) Rate of Service Failure (RSF) in function of P_r and γ



(b) Rate of Service Failure (RSF) in function of P_r and reward weights

FIGURE 8. Assessment of the rate of service failure *RSF*.

values in SARSA lead to quicker, more stable convergence, likely due to a focus on immediate rewards over future rewards, potentially at the cost of achieving higher long-term rewards. Mid-range γ values offer a balanced approach, achieving effective convergence with a mix of short and long-term rewards, likely leading to more optimal policies. Higher γ values emphasize future rewards, resulting in slower convergence and less stability, indicating ongoing exploration and adjustment in policy. These dynamics underscore the trade-offs between convergence speed, stability, and reward optimization in SARSA, significantly influenced by the γ value choice.

C. IMPACT OF THE PROBABILITY OF SERVICE REQUEST P_R ON THE RATE OF SERVICE FAILURE (RSF)

In the subsequent step, we evaluate the Rate of Service Failure (RSF) in relation to the service request probability, denoted as P_r, as depicted in Figure 8. RSF quantifies the number of requests that fail to be served when a specific vehicle physical sensor (ϕs) is selected for the vehicle virtual service (VS), but the corresponding sensor is unavailable during the expected time slot. For this evaluation, we set the maximum number of received requests to 30, and the service duration to 30 time slots. Initially, we analyze the RSF with respect to the discount factor γ, as shown in Figure 8a. It is evident that a higher value of γ leads to a higher RSF, as the algorithm prioritizes the next action over the current one. This preference for the next action significantly impacts the decision-making process for sensor selection, ultimately leading to a higher rate of service failure.

Furthermore, we assess the RSF while varying the reward weights in relation to P_r. As shown in Figure 8b, the RSF remains below 0.1 throughout the assessment. The curves exhibit an increasing trend until they reach their respective maximum values, after which they stabilize regardless of P_r. Notably, the curve with the lowest values corresponds to the configuration with the highest δ. In this case, the algorithm is trained to prioritize selecting the sensor with the highest availability within the target service zone. Conversely, the curve with the highest values corresponds to the configuration with the highest β. Here, the algorithm is trained to maintain the same sensor selection for as long as possible, even if it is not the most available one.

D. IMPACT OF THE NUMBER OF VEHICLES VARIATION ON THE VIRTUALIZATION METRICS USING Q-LEARNING AND SARSA ALGORITHMS

We conducted simulations to evaluate the performance of the proposed intelligent sensor selection using Q-learning and SARSA while varying the number of vehicles in the network, as shown in Figure 9.

We started by varying the number of vehicles and testing the rate of true sensor selection based on Equation 2. This metric consists in computing the total number of sensors that are selected and available (i.e with respect to Equation 6) to compose the VS during the required service period T over the total number of selected sensors as follows: $\frac{\sum_{i=1}^{n} \eta_i \cdot \phi s_i}{\sum_{i=1}^{n} \phi s_i}$. As depicted in Figure 9, the rate of true sensor selection increases decreasingly as the number of vehicles increases because there will be more vehicles as sensor supplier candidates in the target service zone Z to be selected to compose the VS. In addition, when the network is more dense, there will be more congestion and more preference of the vehicles to travel different path in the network and therefore spend longer stay time in the target service zone Z. We can notice that the Q-learning algorithm has higher performance in selecting true available sensors compared to SARSA.

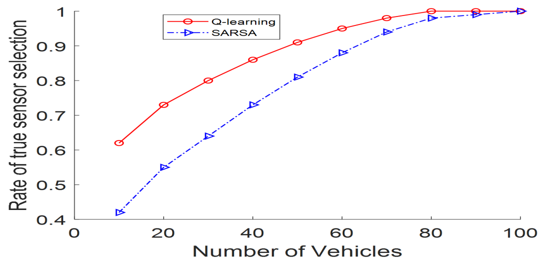
Then, we evaluated the rate of sensor re-selection based on Equation 4 using Q-learning and SARSA. This metric consists in computing the total number of time slots where the vehicle physical sensor is selected consecutively to compose the same VS over the total number of time slots (during the required service period T): $\frac{\sum_{i=1}^{n}\psi_i \cdot TS_i}{\sum_{i=1}^{n} TS_i}$. As shown in Figure 9b, the rate of sensor re-selection decreases as the number of vehicles increases using Q-learning and SARSA because when the number of vehicle candidates is low, the preference of the RL agents is to prioritize the re-selection of the same sensor and therefore to get more rewards.

We also evaluated the rate of multi-tenant sensor sharing based on Equation 3 using Q-learning and SARSA. This metric measures the number of times the same vehicle physical sensor (selected to compose a VS at TS_i) is re-utilized to simultaneously compose other virtual sensors at the same TS_i. The latter metric is defined as follows: $\frac{\sum_{i=1}^{n}\omega_i \cdot \phi s_i}{\sum_{i=1}^{n}\phi s_i}$. As illustrated in Figure 9c, the rate of multi-tenant sensor sharing increases as the number of vehicles in the network increases. The SCSP prefers to reuse the same vehicle physical sensor to create multiple virtual sensors simultaneously, especially when it is available in the target service zone Z. This preference is based on the higher number of vehicles traveling the network, enabling greater utilization of the sensor resources.

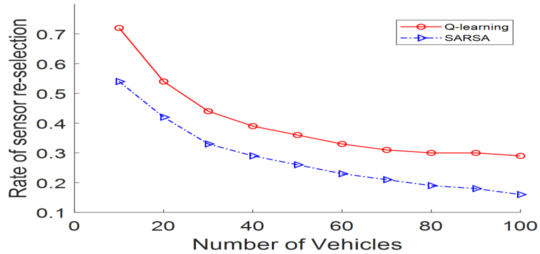
E. ASSESSMENT OF THE GLOBAL COST (GCOST) AND THE RATE OF ECONOMICS WHILE VARYING THE NUMBER OF VEHICLES

We also conducted a simulation to test the impact of the number of vehicles on the global provisioning cost (GCost) which is calculated using the Equation 5 and therefore to compute the rate of economics. The later represents the economics earned by the $SCSP$ from the provisioning of the vehicle virtual sensor (VS) using the intelligent sensor selection (considering the metrics of true sensor selection, sensor re-selection, and sensor re-utilization). Both of GCost and economics are computed for the three phases: Q-learning, SARSA, and random sensor selection. The random sensor selection consists in randomly allocating at each time slot a sensor from a vehicle that is seen available in Z at the current time slot without being aware of its stay time in the network. The GCost using random selection is computed based on the true sensor selection without considering the multi-tenant sharing nor the re-selection of sensors. Thus, GCost is obtained by: $GCost_{Random}(VS, T) = \sum_{i=1}^{n}(\eta_i + Pe)^{1-\eta_i} \cdot UCost(\phi s_i)$. The rate of economics is calculated as follows: $Rate\ of\ economics = \frac{\sum_{i=1}^{n} UCost(\phi s_i) - GCost}{\sum_{i=1}^{n} UCost(\phi s_i)}$.

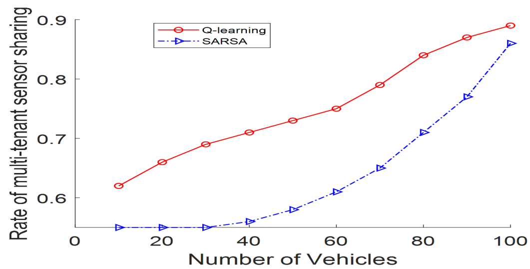
The results show that GCost decreases as the number of vehicles increases. In addition, GCost using Q-learning reaches earlier an approximately constant value than SARSA. However, even with random selection, GCost decreases linearly but remains considerably higher compared to both Q-learning and SARSA. This is primarily because randomly selecting a sensor from an available vehicle based solely on its



(a) Assessment of the rate of true sensor selection with respect to the number of vehicles



(b) Assessment of the rate of sensor re-selection with respect to the number of vehicles



(c) Assessment of the rate of multi-tenant sensor sharing with respect to the number of vehicles

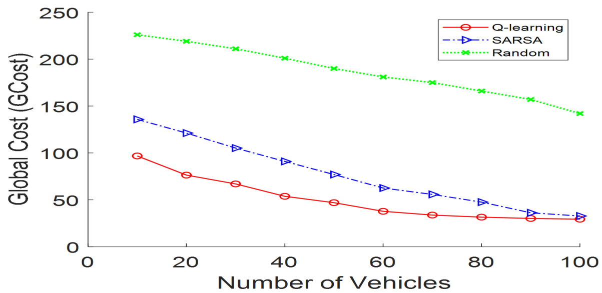
FIGURE 9. Comparison of Q-learning and SARSA while varying the number of vehicles in the network.

current time slot does not guarantee its continuous availability throughout the time horizon. Consequently, penalties are incurred due to virtual sensor service interruptions.

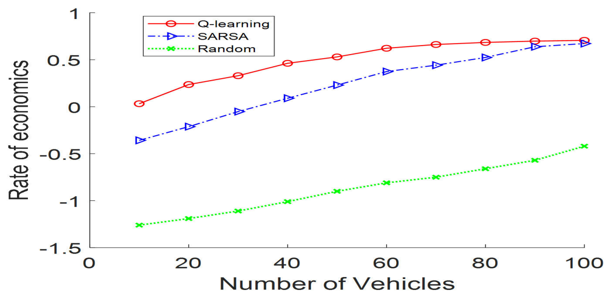
Accordingly, since the rate of SCSP economics is calculated based on GCost, it then increases as the number of vehicles increases where both of the RL algorithms select actually available vehicles and find more candidates to compose the vehicle virtual sensors. We can also observe that the rate of economics using Q-learning is always positive contrarily to that using SARSA which starts holding positive economics from a number of vehicles close to 40. On the other side, the random selection shows negative economics regardless of the number of vehicles due to the exhaustive penalty incurred by the lack of awareness on vehicle availability and also the non-exploitation of sensor resources by many virtual sensors simultaneously.

F. IMPACT OF THE VEHICLES' SPEED VARIATION ON THE VIRTUALIZATION METRICS USING Q-LEARNING AND SARSA ALGORITHMS

We conducted simulations to assess the proposed model for intelligent sensor selection using Q-learning and SARSA algorithms, while varying vehicle speeds. As shown in

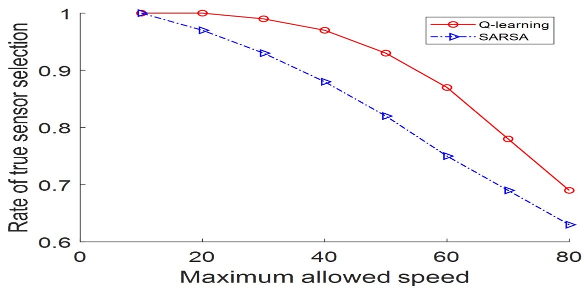


(a) Impact of the number of vehicles on GCost

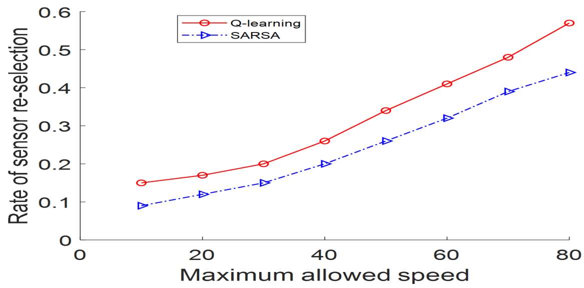


(b) Impact of the number of vehicles on the SCSP economics

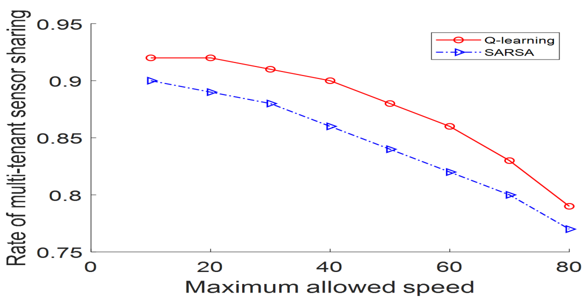
FIGURE 10. Assessment of the SCSP profitability using intelligent and random sensor selection approaches while varying the number of vehicles.



(a) Assessment of the rate of true sensor selection with respect to the maximum allowed speed



(b) Assessment of the rate of sensor re-selection with respect to the maximum allowed speed



(c) Assessment of the rate of multi-tenant sensor sharing with respect to the maximum allowed speed

FIGURE 11. Comparison of Q-learning and SARSA with respect to the maximum allowed speed.

Figure 11a, the rate of true sensor selection initially peaks at a value of one when the vehicle's speed is equal to 10 km/h. This is because when vehicles travel at a lower speed, they spend more time within the target service zone Z. This increased duration enhances their availability, resulting in a larger pool of candidates for selection by the $SCSP$. However, as the speed increases to 80 km/h, the vehicles quickly exit the service zone Z, leading to a gradual decrease in the rate of true sensor selection, falling below 0.7.

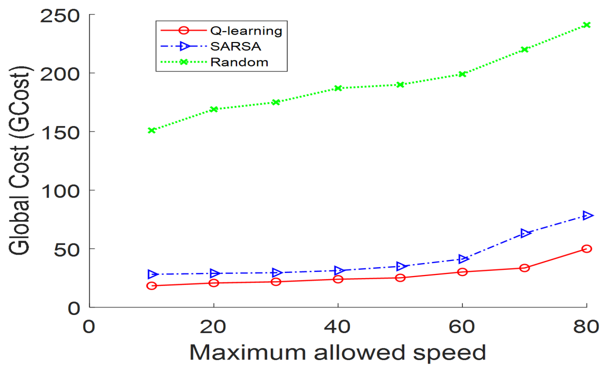
Regarding the rate of sensor re-selection, both the Q-learning and SARSA curves initially start from low values as depicted in Figure 11b. This can be attributed to the fact that the RL agents prioritize learning and selecting the sensors of new vehicles with higher availability, rather than re-selecting the sensors of the same vehicle to increase their rewards.

In terms of the rate of multi-tenant sensor sharing, the RL agents tend to re-allocate the same vehicle's sensor for multiple virtual services when the certainty of its true availability is higher, particularly when the vehicle speed is low. This behavior is illustrated in Figure 11c.

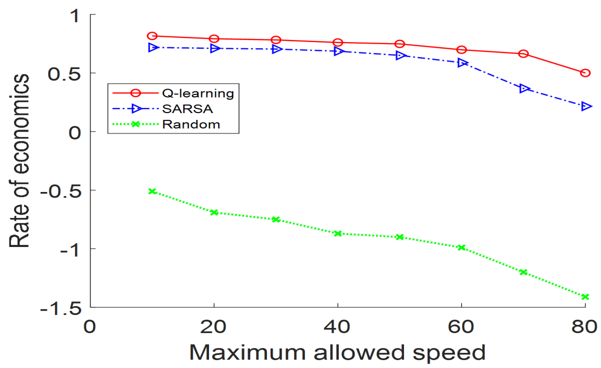
G. ASSESSMENT OF THE GLOBAL COST (GCOST) AND THE RATE OF ECONOMICS WHILE VARYING THE MAXIMUM ALLOWED SPEED

We conducted simulations to investigate the impact of vehicle speed on the global provisioning cost (GCost) and the rate of economics. A comparison study is performed between Q-learning and SARSA (both as RL models) and a random selection approach, as illustrated in Figure 11. The GCost when utilizing RL models demonstrate a slight increase with vehicle speed until a certain threshold is reached. Beyond this threshold, the GCost experiences a substantial escalation. However, when employing a random sensor selection approach, the GCost continues to rise significantly as the vehicle speed increases, resulting in a substantial gap compared to the GCost obtained from Q-learning and SARSA algorithms as shown in Figure 12.

Regarding the rate of economics, both Q-learning and SARSA algorithms consistently yield positive and significant rates of economics, particularly for lower vehicle speeds. The SCSP, when employing Q-learning and SARSA, tends to prioritize sensor selection from vehicles with lower speeds, leading to improved economic performance. In contrast, when utilizing random sensor selection, the rate of economics remains consistently negative. This is especially due to the penalties incurred and the sub-optimal utilization of sensors caused by the lack of awareness of vehicle availability and the absence of an optimized selection mechanism.



(a) Impact of the vehicle's speed on GCost



(b) Impact of the vehicle's speed on the SCSP economics

FIGURE 12. **Assessment of the SCSP profitability using intelligent and random sensor selection approaches with respect to the vehicle's speed.**

VIII. CONCLUSION

In this paper, we introduced a reinforcement learning-based solution for managing and provisioning virtual sensors within the Internet of Vehicles (IoV). Our method begins with the establishment of an interactive IoV-Cloud architecture, which facilitates the on-demand provisioning of vehicle virtual sensors. We designed a concept of sensor virtualization that involves performing configuration mapping on vehicle physical sensors to optimize their utilization and enable them to serve multiple application requests. Additionally, we have developed a reinforcement learning model that dynamically selects a vehicle physical sensor at each time step to compose the vehicle virtual sensor. This intelligent sensor selection process is based on sensor localization, availability, and the cost-effectiveness of provisioning vehicle virtual sensors. Simulation results demonstrate the effectiveness of our proposed reinforcement learning solution in achieving stable sensor composition and significant provisioning cost savings. As future work, ensuring security in sensor virtualization is crucial for preventing unauthorized access within interconnected IoV systems. The need for security is heightened for applications accessed through the cloud, due to the inherent vulnerabilities of cloud computing, such as shared infrastructure risks. Furthermore, the use of data from heterogeneous vehicle sensors, which form the virtual sensor, introduces the potential for data falsification by malicious vehicles. These vehicles might tamper with mobility or sensor data to influence the future sensor selection to compose the virtual sensors.

Thus, we plan to introduce a blockchain-based framework to preserve the data history and reputation of vehicles selected to contribute to virtual sensors, while ensuring the anonymity of participants using pseudonyms. This approach also includes using Smart Contracts to manage data updates, vehicle selection, and sensor access impartially, thereby eliminating bias.

Regarding data transmission security, it is essential that the data remains private and anonymized from vehicles not involved in the virtual sensor network. We propose enhancing this aspect by generating and distributing a new session key between the vehicle virtual sensor and the selected physical sensor at each timeslot. This method ensures that access to sensor data is tightly controlled and restricted to authorized periods. The session key exchange can be established using the ephemeral Elliptic Curve Diffie-Helman (ECDH) protocol after completing authentication of the selected physical sensor and the vehicle virtual sensor using Pre-Shared Key (PSK) form, with an emphasis on Perfect Forward Secrecy (PFS). PFS prevents the decryption of past or future transmissions with compromised keys, thus enhancing security against breaches.

The proposed security measures should conform to the standards set by the European Telecommunications Standards Institute (ETSI) and the Internet Engineering Task Force (IETF). Such compliance is essential for establishing a Public Key Infrastructure (PKI) within the vehicular domain, enabling the issuance of necessary authentication and authorization certificates for secure communication among vehicles.

REFERENCES

[1] L. Du, R. Huo, C. Sun, S. Wang, J. Guo, and T. Huang, "Adaptive resource allocation of vehicles under dynamic environment," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, May 2022.

[2] N. Pathak, S. Misra, A. Mukherjee, A. Roy, and A. Y. Zomaya, "UAV virtualization for enabling heterogeneous and persistent UAV-as-a-service," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6731–6738, Jun. 2020.

[3] D. Martin, N. Kühl, and G. Satzger, "Virtual sensors," *Bus. Inf. Syst. Eng.*, vol. 63, no. 3, pp. 315–323, Jun. 2021.

[4] Z. Wang, P. Jochem, and W. Fichtner, "A scenario-based stochastic optimization model for charging scheduling of electric vehicles under uncertainties of vehicle availability and charging demand," *J. Cleaner Prod.*, vol. 254, May 2020, Art. no. 119886.

[5] A. Singh, L. Gaba, and A. Sharma, "Internet of Vehicles: Proposed architecture, network models, open issues and challenges," in *Proc. Amity Int. Conf. Artif. Intell. (AICAI)*, Feb. 2019, pp. 632–636.

[6] I. Javed, X. Tang, M. A. Saleem, A. Javed, M. A. Zia, and I. A. Shoukat, "Localization for V2X communication with noisy distance measurement," *Int. J. Intell. Netw.*, vol. 4, pp. 355–360, Aug. 2023.

[7] M. A. Saleem, Z. Shijie, and A. Sharif, "Data transmission using IoT in vehicular ad-hoc networks in smart city congestion," *Mobile Netw. Appl.*, vol. 24, no. 1, pp. 248–258, Feb. 2019.

[8] S. Abbes and S. Rekhis, "Reinforcement learning-based virtual sensors provision in Internet of Vehicles (IoV)," in *Proc. IEEE 21st Int. Symp. Netw. Comput. Appl. (NCA)*, vol. 21, Boston, MA, USA, Dec. 2022, pp. 217–224.

[9] Y. Choi and S. Yoon, "In-situ observation virtual sensor in building systems toward virtual sensing-enabled digital twins," *Energy Buildings*, vol. 281, Feb. 2023, Art. no. 112766.

[10] A. Gupta, S. Saha, and S. Adhikary, "Delay-sensitive remote health framework for secured WBAN using sensor virtualization," *Appl. Innov. Mobile Comput. (AIMOC)*, 2016, pp. 99–105.

[11] O. Chenaru, C. E. Hanganu, D. Popescu, and L. Ichim, "Virtual sensor for behavior pattern identification in a smart home application," in *Proc. 8th Int. Conf. Syst. Control (ICSC)*, Oct. 2019, pp. 388–392.

[12] J. Xu, A. Bhattacharya, A. Balasubramanian, and D. E. Porter, "Sensor virtualization for efficient sharing of mobile and wearable sensors," in *Proc. 19th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2021, pp. 460–466.

[13] D. Rajavel, A. Chakraborty, and S. Misra, "QoS-aware sensor virtualization for provisioning green sensors-as-a-service," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1128–1137, Sep. 2021.

[14] M. Lemos, R. Filho, R. Rabêlo, C. de Carvalho, D. Mendes, and V. Costa, "An energy-efficient approach to enhance virtual sensors provisioning in sensor clouds environments," *Sensors*, vol. 18, no. 3, p. 689, Feb. 2018.

[15] Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient IoT virtualization framework with peer to peer networking and processing," *IEEE Access*, vol. 7, pp. 50697–50709, 2019.

[16] Z. Khalid, U. Khalid, M. A. Sarijari, H. Safdar, R. Ullah, M. Qureshi, and S. U. Rehman, "Sensor virtualization middleware design for ambient assisted living based on the priority packet processing," *Proc. Comput. Sci.*, vol. 151, pp. 345–352, Jan. 2019.

[17] R. Liu and M. Srivastava, "VirtSense: Virtualize sensing through ARM TrustZone on Internet-of-Things," in *Proc. 3rd Workshop Syst. Softw. Trusted Execution*, Jan. 2018, pp. 2–7.

[18] N. Almurisi and S. Tadisetty, "Cloud-based virtualization environment for IoT-based WSN: Solutions, approaches and challenges," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 10, pp. 4681–4703, Oct. 2022.

[19] S. Alam, M. M. R. Chowdhury, and J. Noll, "SenaaS: An event-driven sensor virtualization approach for Internet of Things cloud," in *Proc. IEEE Int. Conf. Networked Embedded Syst. Enterprise Appl.*, Nov. 2010, pp. 1–6.

[20] L. Hang, W. Jin, H. Yoon, Y. Hong, and D. Kim, "Design and implementation of a sensor-cloud platform for physical sensor management on CoT environments," *Electronics*, vol. 7, no. 8, p. 140, Aug. 2018.

[21] G. Sun, L. Song, H. Yu, X. Du, and M. Guizani, "A two-tier collection and processing scheme for fog-based mobile crowdsensing in the Internet of Vehicles," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1971–1984, Feb. 2021.

[22] M. A. Saleem, Z. Shijie, M. U. Sarwar, T. Ahmad, A. Maqbool, C. S. Shivachi, and M. Tariq, "Deep learning-based dynamic stable cluster head selection in VANET," *J. Adv. Transp.*, vol. 2021, pp. 1–21, Jul. 2021.

[23] M. A. Saleem, S. Zhou, A. Sharif, T. Saba, M. A. Zia, A. Javed, S. Roy, and M. Mittal, "Expansion of cluster head stability using fuzzy in cognitive radio CR-VANET," *IEEE Access*, vol. 7, pp. 173185–173195, 2019.

[24] S. Abbes and S. Rekhis, "Sensor virtualization and provision in Internet of Vehicles," in *Advanced Information Networking and Applications*, L. Barolli, F. Hussain, and T. Enokido, Eds. Cham, Switzerland: Springer, 2022, pp. 386–397.

[25] A. Muhammad, M. Saqib, and W.-C. Song, "Sensor virtualization and data orchestration in Internet of Vehicles (IoV)," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2021, pp. 998–1003.

[26] B. Marques and M. Ricardo, "Energy-efficient node selection in application-driven WSN," *Wireless Netw.*, vol. 23, no. 3, pp. 889–918, Apr. 2017.

[27] I. Javed, X. Tang, M. A. Saleem, M. U. Sarwar, M. Tariq, and C. S. Shivachi, "3D localization for mobile node in wireless sensor network," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–12, Mar. 2022.

[28] S. Baek, C. Liu, P. Watta, and Y. L. Murphey, "Accurate vehicle position estimation using a Kalman filter and neural network-based approach," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.

[29] J. Liu and G. Guo, "Vehicle localization during GPS outages with extended Kalman filter and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[30] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "DeepLanes: End-to-end lane position estimation using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 38–45.

[31] H. Liang, X. Zhang, X. Hong, Z. Zhang, M. Li, G. Hu, and F. Hou, "Reinforcement learning enabled dynamic resource allocation in the Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4957–4967, Jul. 2021.

[32] A. Verma, M. Hanawal, C. Szepesvari, and V. Saligrama, "Online algorithm for unsupervised sensor selection," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 3168–3176.

[33] M.-Z. Zhang, L.-M. Wang, and S.-M. Xiong, "Using machine learning methods to provision virtual sensors in sensor-cloud," *Sensors*, vol. 20, no. 7, p. 1836, Mar. 2020.

[34] H. Mahboubi, S. Blouin, and A. G. Aghdam, "A machine learning assisted method for coverage optimization in a network of mobile sensors," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7301–7311, Sep. 2022.

[35] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 128–135, Aug. 2016.

[36] H. R. Arkian, R. E. Atani, A. Diyanat, and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," *J. Supercomput.*, vol. 71, no. 4, pp. 1401–1426, Apr. 2015.

[37] H. Liang, X. Zhang, J. Zhang, Q. Li, S. Zhou, and L. Zhao, "A novel adaptive resource allocation model based on SMDP and reinforcement learning algorithm in vehicular cloud system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10018–10029, Oct. 2019.

[38] Q. Pan, J. Wu, J. Nebhen, A. K. Bashir, Y. Su, and J. Li, "Artificial intelligence-based energy efficient communication system for intelligent reflecting surface-driven VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19714–19726, Oct. 2022.

[39] T. S. J. Darwish and K. A. Bakar, "Fog based intelligent transportation big data analytics in the Internet of Vehicles environment: Motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15679–15701, 2018.

[40] S. Abbes and S. Rekhis, "A blockchain-based solution for reputation management in IoV," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*. Harbin City, China: IEEE, Jun. 2021, pp. 1129–1134.

[41] M. Grześ and D. Kudenko, "Online learning of shaping rewards in reinforcement learning," *Neural Netw.*, vol. 23, no. 4, pp. 541–550, May 2010.

[42] I. Javed, X. Tang, K. Shaukat, M. U. Sarwar, T. M. Alam, I. A. Hameed, and M. A. Saleem, "V2X-based mobile localization in 3D wireless sensor network," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, Feb. 2021.

SLIM ABBES (Member, IEEE) received the master's degree in telecommunication and network systems from the National School of Electronics and Telecommunications of Sfax (ENET'COM), University of Sfax, Tunisia. He is currently pursuing the Ph.D. degree in information and communication technologies with the Higher School of Communication of Tunis (SUP'COM), University of Carthage, Tunisia. He is a member of the Communication Networks and Security (CNAS) Research Laboratory. His research interests include the Internet of Vehicles, sensor management and provisioning, reinforcement learning, blockchain, and cloud computing.

SLIM REKHIS received the Ph.D. and University Habilitation degrees in information and communication technologies from the University of Carthage, Tunisia, in 2007 and 2013, respectively. Since October 2016, he has been the Director of the Communication Networks and Security (CNAS) Research Laboratory, University of Carthage. He is currently an Associate Professor of telecommunications with the SUP'COM School of Communication Engineering, University of Carthage. His research interests include digital investigation of cyber security incidents, formal modeling, security of implantable medical devices, cyber security and defense, cloud computing, intrusion prevention and detection, wireless sensor networks, networking and quality of service, and wireless security.