IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2025, accepted 20 July 2025, date of publication 24 July 2025, date of current version 1 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3592310

RESEARCH ARTICLE

Big Data and I2X Communication Infrastructure for Traffic Optimization and Accident Prevention on Automated Roads

JORGE GARCÍA-GONZÁLEZ[1], JAVIER FERNÁNDEZ-ANDRÉS[2], NOURDINE ALIANE[2], AND JAVIER SÁNCHEZ-SORIANO[3]

[1]Department of Science, Computing and Technology, Universidad Europea de Madrid, 28670 Villaviciosa de Odón, Spain
[2]Department of Industrial and Aerospace Engineering, Universidad Europea de Madrid, 28670 Villaviciosa de Odón, Spain
[3]Advanced Artificial Intelligence Group (A²IG), Escuela Politécnica Superior, Universidad Francisco de Vitoria, 28223 Pozuelo de Alarcón, Spain

Corresponding author: Javier Sánchez-Soriano (javier.sachez@ufv.es)

This work was supported in part by the I+D+i Projects funded by Ministerio de Ciencia e Innovación, Agencia Estatal de Investigación under Grant PDC2022-133684-C33 and Grant PID2022-140554OB-C33; and in part by the I+D+i Project funded by the Regional Government of Madrid under Grant TEC-2024/ECO-277/SEGVAUTO-5G-CM.

ABSTRACT This paper presents a scalable big data infrastructure designed to support traffic optimization and accident prevention in automated and connected road environments. The proposed system integrates real-time data acquisition from heterogeneous sources, including multichannel roadside camera gantries and IoT-enabled vehicle telemetry. The architecture is built upon technologies such as Apache Kafka, Apache Beam, and MongoDB, enabling high-throughput data ingestion, processing, and storage. To validate its performance, two experimental use-cases were developed: one for large-scale image ingestion and another for vehicle telemetry data streaming. The system successfully handled over 48 GB of image data and more than 3.4 million telemetry messages under real-time constraints. Results show that applying data compression techniques—such as resolution reduction and transmission throttling—reduced image upload durations by up to 77%, improving ingestion efficiency without compromising system robustness. These findings demonstrate the feasibility of deploying the proposed infrastructure as a foundational layer for future intelligent traffic management systems.

INDEX TERMS Big data, real-time, intelligent transportation systems (ITS), infrastructure, streaming, V2X communication.

I. INTRODUCTION

The vehicle automation technologies have been increasing recently, driven by the significant development of Intelligent Transportation Systems (ITS) [1], [2]. The deployment of connected vehicles forms the cornerstone of the evolving transportation paradigm. Establishing connections between vehicles and the infrastructures they navigate is essential [2], [3]. This connectivity enables sharing, querying, as well as validation and recommendation of driving-related information. Consequently, vehicles can assist human drivers in decision-making and even support other vehicles, including autonomous ones [4].

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

Connected vehicles encompass not only autonomous cars but also manually driven vehicles equipped with advanced driver-assistance systems (ADAS) and vehicle-to-vehicle (V2V) communication [1], [4], [5]. On the one hand, they can send and transmit status updates, informing other vehicles on the road. On the other hand, they can utilize information generated by their own sensors and obtained through external communications to improve their decision-making process during driving [4], [6]. Figure 1 illustrates a connected autonomous vehicle with numerous sensors, grouped into several categories [4]: long-range radars, short- and medium-range radars, cameras, or LiDAR. On the other hand, existing connections occur not only between vehicles but also with road infrastructures. The types of connections include Vehicle-to-Vehicle (V2V)

connections, Vehicle-to-Infrastructure (V2I) connections, and Infrastructure-to-Infrastructure (I2I) connections [1], [7]. These connections are illustrated in Figure 2.

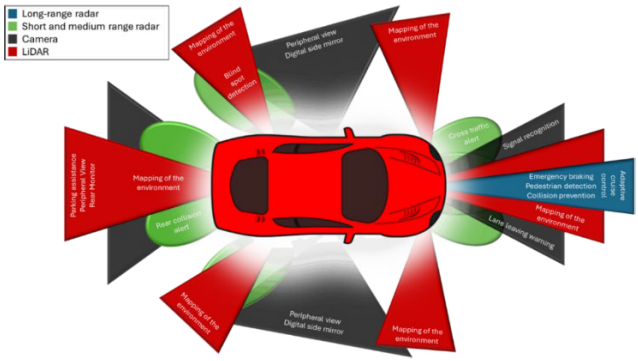

FIGURE 1. Sensor set in connected autonomous vehicles.

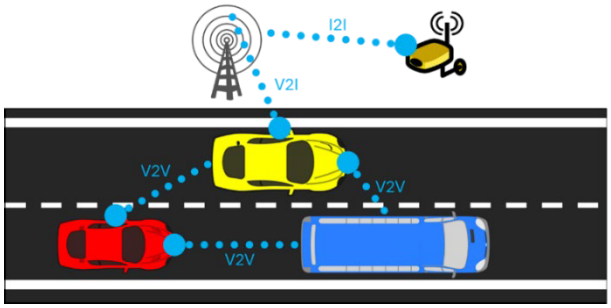

FIGURE 2. Connections between elements in automated roadways.

The efficient management of the data generated by these systems, which includes both visual information and vehicle telemetry, has become a challenge for autonomous driving systems. Despite this, recent research has proposed alternative solutions, such as the use of distributed multi-agent reinforcement learning for cooperative resource management in vehicular environments [8]. However, many of these proposals focus on simulated environments or specific subsystems, so there is still a gap in the practical implementation of big data infrastructures that are scalable, modular, and capable of operating in real time with heterogeneous data under real conditions. This proposal differs from other alternatives mainly in its experimental validation with real data, its ability to integrate both fixed and mobile sensors, and its direct focus on direct applicability in automated traffic environments.

The proposed model consists of a system that captures and stores information obtained from different sources. In this regard, the system consists of a server with two different databases. The first one is an SQL database that stores structured data for offline data analytics, and the second one is a NoSQL database for raw data collected from vehicles or road infrastructures. Example of possibles uses include processing the stored information in real time aimed at providing real-time recommendations to drivers, providing structured information ready to be used for different off-line studies and analysis, etc. This model differs from other alternatives mainly in its experimental validation with real data, its ability to integrate both fixed and mobile sensors, and its direct focus on direct applicability in automated traffic environments.

The objective of this research is to design, implement, and evaluate a scalable big data infrastructure capable of ingesting and processing heterogeneous real-time data from roadside and onboard IoT devices. The proposed system aims to support automated decision-making in connected vehicles and traffic control centers, thereby enhancing traffic flow efficiency and accident prevention capabilities.

The main contributions of this paper are as follows:

- A novel architecture for vehicular big data ingestion integrating fixed and mobile sensors.
- A practical deployment and evaluation in two real-world use-cases involving high-frequency image and telemetry data.
- An empirical performance analysis demonstrating the system's adaptability through resolution adjustment and transmission throttling mechanisms.
- A discussion on the feasibility of this infrastructure to serve as a foundation for future vehicle-to-everything (V2X) applications.

The remainder of the paper is organized as follows: Section II reviews related work in intelligent transportation systems and V2X communication. Section III presents the methodology, detailing the design principles and requirements, the proposed infrastructure architecture with the selection of key technologies, the prototype deployment environment (hardware, software and network), and describes the two experimental use cases: gantry camera image capture and vehicular telemetry data collection. Section IV presents the results and analysis of the experimental validation of these use cases. Section V discusses the practical implications, challenges and opportunities for the actual implementation of the system, as well as its effects on ITS. Finally, Section VI draws conclusions, highlights the potential of the system and suggests areas for future lines of research, including the integration of emerging technologies and attention to security and privacy.

II. RELATED WORK

The foundation of connected and automated driving lies in reliable V2X communication and scalable infrastructure to handle the massive data volumes these systems generate. Various studies have addressed key elements of this ecosystem.

Advanced driver-assistance systems (ADAS) are pivotal in the evolution of Intelligent Transportation Systems (ITS), enhancing vehicle safety and efficiency through real-time information and automated support. These technologies range from basic features like lane departure warnings and adaptive cruise control to advanced systems such as automatic emergency braking, representing various levels of vehicle automation and aiding the transition towards fully autonomous vehicles [9], [10], [11]. Integral to this

are Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) components, which enable real-time data exchange between vehicles and infrastructure, crucial for the connectivity of autonomous vehicles [10]. The transmission of millimeter-wave 5G/6G with adaptive analog beamforming is set to enhance future mobile networks' capacity and efficiency, essential for autonomous vehicle connectivity [12], [13]. V2X communications, encompassing interactions between vehicles and their environment, improve decision-making by providing informed data and proactively avoiding potential accidents [10]. The integration of advanced AI technologies and frameworks like V2X has propelled the development of autonomous driving systems, facilitating real-time decision-making and adaptation to various driving environments through features like Adaptive Cruise Control (ACC), Automatic Lane Centering (ALC), Forward Collision Warning (FCW), and Lane Keeping Assist (LKA).

Smart roads integrate technologies such as IoT to manage traffic in real-time, using infrastructure capable of predicting and detecting non-connected vehicles. These systems optimize routes, parking, and lighting, enhancing safety and road maintenance [14]. Advancements in connected intelligent transport systems (C-ITS) have paved the way for smarter and safer road infrastructure, allowing real-time data exchange between vehicles and traffic infrastructure to optimize decision-making processes. This technology has been shown to enhance traffic flow and reduce accident rates, particularly in complex road scenarios like roundabouts [15]. Kussl and Wald [16] propose a conceptual model for road transformation, positioning smart mobility as a crucial factor in infrastructure evolution. This model includes adaptive planning approaches, advanced technologies, and real-time management and monitoring systems, as well as services like electric vehicle charging stations and vehicle-to-infrastructure communication. It also emphasizes the need for organizational adaptations to effectively manage these innovations. It is essential to equip smart roads with systems that can collect real-time event data. Ranyal et al [17] highlights the importance of using smart sensors and AI to maintain the integrity and safety of road infrastructure. Beyond traffic control, these systems allow the detection of infrastructure failures, enabling preventive maintenance. To develop these systems, it is crucial to select technologies that support real-time data collection and analysis. Relevant technologies include messaging platforms, SQL and NoSQL databases, and real-time processing systems including Apache, Mosquitto, and Cassandra among others. Therefore, it is essential to start with a careful selection of candidate tools, thoroughly evaluating their advantages and limitations.

Recent approaches have explored the integration of AI-driven decision-making mechanisms into vehicular environments. For example, the use of Distributed Multi-Agent Reinforcement Learning (MARL) for cooperative edge caching has demonstrated potential in reducing latency and improving content availability in vehicular networks [18]. Similarly, incentive-based offloading strategies in Vehicular Edge Computing (VEC) environments have proven effective for balancing resource allocation and network efficiency [19].

However, many of these works are either simulation-based or focus on specific subsystems. Our work differs in that it integrates a complete and deployable infrastructure that has been validated under real-time conditions with both video and telemetry data sources. Moreover, our modular design allows for extension and integration with decision-making subsystems, forming a bridge between infrastructure-level data acquisition and higher-level AI-based traffic management.

III. METHODOLOGY

This section presents the methodological approach followed to design and validate the proposed big data infrastructure for intelligent traffic systems. The research was structured around five main stages: (A) identifying key system requirements for real-time traffic data processing, (B) selecting and integrating appropriate technologies, (C) deploying a prototype on a controlled hardware-software environment, (D) collecting data from realistic sources and simulations, and (E) evaluating system performance through defined metrics (See Figure 3).

Design Principles and Requirements:

The design of the infrastructure was guided by the need for:

- High throughput for both image and telemetry data.
- Low latency in ingestion and processing pipelines.
- Compatibility with resource-constrained IoT devices.
- Modularity to support integration with future AI-based decision systems.

These criteria are fundamental in the context of connected and automated roads, where real-time data acquisition and decision support must be both reliable and scalable. The following architecture has been designed to meet these criteria.

A. INFRAESTRUCTURE ARCHITECTURE

The proposed solution is based on a big data architecture composed of multiple interconnected services, which ensures the efficient management of data flow for processing and storage [2], [13]. The architecture is divided into three main functional units: (1) Data ingestion and processing services, (2) Data storage services and (3) REST API for data query by external sources. In the following, the functions of these functional units are detailed in more detail. Figure 3 illustrates the main components of the proposed software architecture.

The selection of appropriate technologies depends on the specific services and requirements of the infrastructure. In this regard, the system is designed to offer three main services: data ingestion and processing, data warehousing, and provide an adequate API for external data queries. A summary of these tools is listed below:

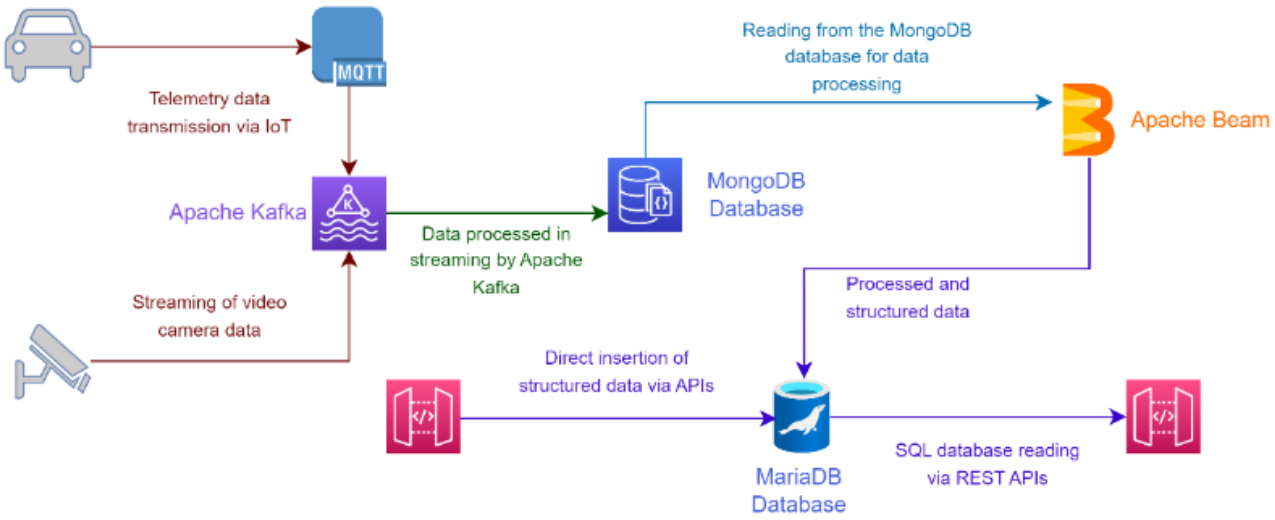

FIGURE 3. System architecture.

- **Apache Kafka** [20]. Distributed messaging platform used to handle large volumes of real-time data. Its main advantages are high scalability and data durability. However, it requires advanced configuration and management.
- **Apache Beam** [21]. A unified model for defining data processing pipelines across multiple execution environments. It is highly flexible and supports portability. Still, it may require specialized development knowledge, especially for complex transformations.
- **Eclipse Mosquitto** [22]. Lightweight MQTT broker ideal for resource-constrained architectures and intermittent connectivity. It is platform-independent and allows local caching. However, it is limited in high-performance environments and less suited for massive data throughput.
- **MariaDB** [23]. A SQL database compatible with MySQL that improves replication and query handling. Being open source, it facilitates maintenance and migration. Nevertheless, it may struggle with very large data volumes.
- **MongoDB** [24]. Flexible NoSQL document database supporting sharding and replication. It enables high horizontal scalability and performance. However, in certain cases, it may be less efficient compared to traditional relational databases.
- **Cassandra** [25]. A NoSQL database optimized for large-scale deployments with high availability. It offers consistent performance across distributed systems, but it is complex to configure and maintain and was therefore discarded in favor of MongoDB.
- **Firebase Firestore** [26]. Managed NoSQL database offering real-time synchronization and easy integration. It automatically scales and is suited for mobile/web apps. Its main drawback is the dependency on Google Cloud, which introduces vendor lock-in.
- **SQLite** [27]. Embedded SQL database with very low resource consumption, ideal for local applications. However, it is not suitable for large data volumes or production-level traffic data ingestion.
- **Amazon Kinesis** [28]. A managed real-time data streaming service from AWS. It scales easily without requiring infrastructure management. The dependency on Amazon and its associated costs led to its exclusion.
- **RabbitMQ** [29]. Flexible, low-latency messaging system for inter-process communication. While fast for smaller data volumes, it does not scale as well as Kafka when dealing with large-scale streaming data.
- **Apache Spark** [30]. High-performance engine for large-scale batch and stream data processing. It provides excellent speed and flexibility but requires complex resource management and infrastructure setup.
- **Google Cloud Dataflow** [31]. Managed service based on Apache Beam for processing data in Google Cloud. It offers automatic scaling and integration but is limited to the Google Cloud ecosystem.
- **Apache Flink** [32]. Stream processing engine focuses on event-driven applications. It provides excellent performance for real-time tasks but offers less integration flexibility compared to Apache Beam.
- **CouchDB** [33]. A document-oriented NoSQL database that supports distributed replication and flexible data formats. However, its performance is lower compared to alternatives like MongoDB.

The following points discuss the selection of suitable technologies for each of these services.

1) DATA INGESTION AND PROCESSING SERVICES

This service is responsible for collecting and processing data from various information sources. The ingest service must efficiently handle large volumes of information while

maintaining data integrity It must also support the collection of lighter data with minimal resource usage and process unstructured data, restructuring it for further analysis. To meet these requirements, the following technologies have been selected for this system:

1. **Apache Kafka** [34]**:** This is the service responsible for streaming data collection, ensuring that the received data can be properly labeled for further analysis and processing. Apache Kafka was chosen for its ability to handle large volumes of data with high reliability. It also allows seamless scalability without downtime and ensures data durability through disk storage. Amazon Kinesis and RabbitMQ were also considered but ultimately excluded. Kinesis was ruled out due to its dependency on Amazon's cloud and the associated high costs, while RabbitMQ was excluded because it lacks the capacity to handle large data volumes as effectively as Kafka.
2. **Apache Beam** [21]**:** This service is responsible for managing the data streams collected through the data collection service. It processes the information for analytics, recommendations, or structure for proper storage, depending on the origin or utility of the data received. It was selected as a unified data processing model, allowing for the definition of various data pipelines across different execution environments. Its flexibility makes it adaptable to diverse infrastructures and data processing requirements, while simplifying development by abstracting technical complexity. Considered alternatives, but finally discarded, include Apache Spark, which involves a difficult implementation due to its detailed resource management requirements; Google Cloud Dataflow, which creates a high dependency on Google Cloud infrastructure; Apache Flink, which was found to be less flexible in terms of integration with other systems.
3. **Eclipse Mosquitto** [35]**:** For telemetry data collection, the MQTT protocol is used to receive real-time data, which then is passed to the Kafka broker [36]. It was chosen for its lightweight and efficiency, making it a suitable choice for resource-limited architecture. As a cross-platform solution, it operates on nearly any operating system and allows local data storage to be transmitted once connectivity is reestablished. EMQX and RabbitMQ were initially considered but finally excluded: the first due to its higher configuration and maintenance complexity, and the later for being less efficient given the project's resource constraints.

2) DATA WAREHOUSING SERVICES

This service enables the storage of huge amounts of collected data, which can later be accessed by other hosts via REST API. It can be used to generate models for recommending, managing, or validating data, as well as exploring other infrastructure options. This system includes two distinct data storage units:

1. **NoSQL server for unstructured data:** This server is responsible for collecting that information without concrete structure collected various data collection and ingestion systems. The prosed server utilizes MongoDB system, which, as a documentary database, allows for the storage of massive data for ingestion and consultation. It can store not only data but also images, videos and various types of documents. One of the main reasons for choosing MongoDB is its schema-free nature, which allows data from multiple heterogeneous sources to be stored without the need for a predefined structure. This provides great flexibility when working with diverse and constantly evolving data formats. In addition, MongoDB supports efficient indexing and querying, making it suitable for real-time and offline analytics workflows. It also supports sharding, which facilitates horizontal scalability, and offers high performance with support for indexing and data replication. Discarded tools include Cassandra, due to its maintenance complexity; CouchDB, which was found to be less efficient in terms of performance; and, due to its dependency on Google Cloud infrastructure.
2. **SQL Server for structured data:** This server stores information that users can access externally. It stores data that has already been processed and previously stored on the NoSQL server. MariaDB was selected due to its compatibility with MySQL, which facilitates migration and maintenance of existing applications. It offers improvements in replication and query handling compared to MySQL and is open source. Several technologies were initially considered but ultimately discarded. These include MySQL, due to its limitations in replication and query handling; PostgreSQL, for its lesser suitability for data migration from MySQL; and SQLite, due to its inability to manage large data volumes.

3) DATA ACCESS REST API FOR EXTERNAL DATA QUERY

As mentioned previously, the system is designed to facilitate access to data through the REST API from various hosts. This API allows users to download stored data or access a web interface to visualize data related to specific points on the roads. The primary aim is to provide users with an API that allows to querying road data.

The API is designed to be flexible and easy to use, enabling developers to integrate their applications with the road data system. To this end, several key aspects have been implemented. These include the configuration of endpoints, creating multiple endpoints to support different types of queries by default, and creating endpoints dedicated to real-time traffic data, historical road data, and incident or accident data. Additionally, endpoints can be modified or new ones created to accommodate new types of queries. The default data can be downloaded in JSON format, making it easy to integrate and process, and can be viewed directly from a WEB API, which provide

predefined data visualization options for queries can be found.

B. IMPLEMENTATION ENVIRONMENT

Designing a robust Big Data server requires careful attention to hardware, software, and network components to ensure efficient processing and secure data management. To this end, the prototype is built on a server machine with multi-core processors to handle parallel processing, coupled with a large amount of RAM to facilitate fast data processing, and with high capacity for storage (Table 1).

TABLE 1. Hardware for the proof of concept.

Hardware characteristics for the proof of concept	
SO	Ubuntu 22.04
CPU	AMD Ryzen 7 7700X
Cores	8 cores. 16x 5.6 GHz
Mainboard	AORUS ELITE X670
RAM	32 GB
Disk	1 TB

The hardware configuration was selected to ensure sufficient computational capacity for real-time ingestion and processing of high-frequency data streams. The AMD Ryzen 7 7700X processor, with 8 physical cores and 16 threads running up to 5.6 GHz, provides high parallelism and low-latency task execution, which is essential for handling concurrent Kafka consumers and Beam pipeline stages. The 32 GB of DDR5 RAM allows for in-memory buffering of large data batches and supports the execution of multiple services without performance degradation. The 1 TB NVMe SSD ensures high IOPS and low access latency, which is critical for fast read/write operations during ingestion and transformation of image and telemetry data. This setup was sufficient to support the proof-of-concept experiments, which involved ingesting over 48 GB of image data and more than 3.4 million telemetry messages. The system architecture is modular and designed for horizontal scalability, allowing future deployments to distribute workloads across multiple nodes or migrate to cloud-based infrastructure as needed.

The server is also equipped with essential software that includes tools for data ingestion, real-time data streaming, message brokering, and data warehousing solutions. On the network side, the system is also equipped with a Gigabit Ethernet connection to ensure fast data transfer within the data center and with external sources, and access is locked so that it can only be accessed via a private SSH tunnel to provide secure access and protection against unauthorized data leaks. (See Table 2).

The architecture is designed to manage two main data sources: streaming video from gantry cameras and vehicle telemetry data capture through IoT devices. To this end, Apache Kafka serves as the central broker, capturing and centralizing all the data in the system [36]. The collected data are subsequently stored in a MongoDB database, which organizes the information into collections. These collections are processed by Apache Beam, which structures the data and inserts them into the MariaDB database. This setup allows for both the insertion of new structured data and the querying of existing data through a REST API.

TABLE 2. Software and network configuration for the proof of concept.

Software and Network configuration	
SQL database server	MariaDB 15.1
NoSQL database server	MongoDB 7.0.4
Streaming data ingestion	Apache Kafka 2.12-2.5
Streaming ingestion for IoT protocols	Mosquitto 2.0.11
Data analytics and data structuring	Apache Beam 2.55.0
SSH Port	22024
Apache Kafka Port	9092
MQTT (open/encrypted) Ports	1883/8883
SQL database Port	3306
API REST Port	3080

In terms of the configuration of these services, Apache Kafka has been implemented with topics corresponding to images and telemetry, partitioning them to allow parallel processing with multiple consumers (although partitioning was not used in the proof of concept due to the lack of multiple consumers), organized so that they can be consumed depending on the needs of the system. This structure allows for horizontal scalability of the system, ensuring its fault tolerance. On the other hand, Apache Beam was used to define processing pipelines that mainly include data validation stages for migration from MongoDB to MariaDB. For example, in the case of images, metadata was extracted from the images and collated with the records obtained from the data source. If there was any type of error, the data was not processed for storage in the structured database. For telemetry data, transformations were applied to structure MQTT messages into normalized records with relevant information about the vehicle, instead of using all the information received by telemetry. This information is then stored in MariaDB for data analytics. This configuration allowed for low latency and high real-time processing rates, even under high load conditions.

C. DATA SOURCES AND SIMULATION

This section addresses the requirements for performing traffic data capture and collection through two main approaches: The use of image collections simulating traffic gantry cameras, and data collected from vehicle telemetry. The first use-case addresses the capture of vehicle image data using cameras installed on gantries and roundabouts, and it describes the data acquisition system that simulates the recording and sending of images to the server. The second use-case focuses on collecting telemetry data from connected vehicles, implementing a module that sends state car information from a vehicle the server. These two use-cases are designed for prototype testing and validation,

allowing the identification of potential errors before real-world implementation.

1) TRAFFIC DATA CAPTURE VIA CAMERAS ON GRANTIES AND ROUDDABOUTS

The objective of this study is to evaluate the effectiveness and efficiency of the real-time traffic data capture system using cameras installed on gantries and roundabouts. The aim is to test the system's ability to handle large volumes of image data, ensure accurate data ingestion and evaluate the system's performance under various conditions.

In this case study, the system is tested for real-time traffic data capture using multiple camera sources located at various points, such as gantries and roundabouts. For the simulation, camera datasets from two different public datasets created by the Intelligent Control Systems Research Group (SIC) of the European University of Madrid (UEM) are used: "Traffic Images Captured from UAVs for Use in Training Machine Vision Algorithms for Traffic Management" [37] and "Roundabout Aerial Images for Vehicle Detection" [38]. The first set consists of 15,070 images, while the second set includes 15,474 images, giving a total of 30,544 images in the combined set. The images in the datasets range in resolution from 0.92 Megapixels to 2.07 Megapixels, with the weighted average resolution being 1.82 Megapixels. Together, they represent approximately 48GB of information. The number of images sent by each simulated camera, as well as their resolution, are summarized in table 3.

TABLE 3. Number of images per camera used in the simulation.

Camera	Number of images	Resolution
Camera 1	4500	2.07 MP
Camera 2	300	2.07 MP
Camera 3	1500	2.07 MP
Camera 4	2300	0.92 MP
Camera 5	1300	0.92 MP
Camera 6	300	0.92 MP
Camera 7	1300	2.07 MP
Camera 8	2200	2.07 MP
Camera 9	800	0.92 MP
Camera 10	440	0.92 MP
Camera 11	700	0.92 MP
Camera 12	500	0.92 MP
Camera 13	2400	2.07 MP
Camera 14	2400	2.07 MP
Camera 15	2400	2.07 MP
Camera 16	2400	2.07 MP
Camera 17	2400	2.07 MP
Camera 18	2400	2.07 MP

To emulate the operation of a distributed system composed of 18 traffic cameras, a module simulating the recording of traffic cameras and the subsequent sending of images to the server has been implemented (see Figure 4). This recording-sending module is developed as a standalone Docker container, with each instance (one per camera) downloading images from one of the two datasets available on the Kaggle data science platform. Each Docker container manages the subset of images simulating single cameras.

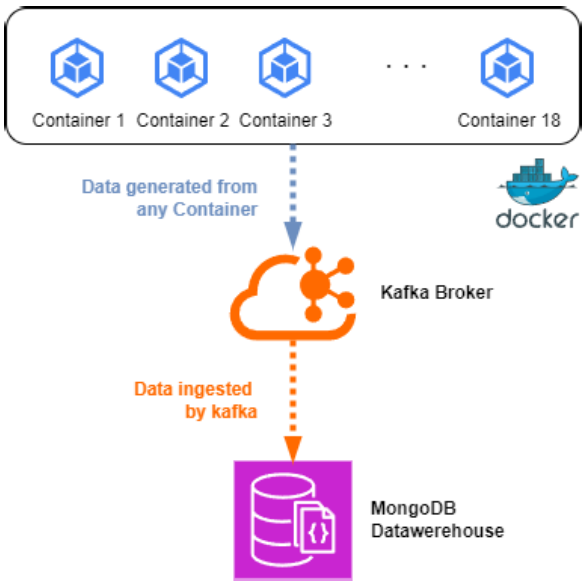

FIGURE 4. Camera data acquisition simulation.

Once all images are downloaded, iterative processing begins, where images are sent directly to the server. A Kafka broker receives the data on a specific topic, and the system automatically ingests the data into the document database, storing images from different cameras while filtering and identifying the source camera. This methodology includes several iterations to observe the server's data ingestion capabilities, with different tests proposed in each iteration to determine the best way to process the data, aiming for near real-time performance.

Two special considerations taken in the proposed experiments implementation are:

- The effect of time-lapse in images uploading is evaluated with two distinct time constraints: uploading images at rates of 1 second and 0.5 second.
- The effect of image resolution reduction is evaluated with two distinct experiments: one with a 25% reduction and the second with 50% reduction in the image's resolution.

2) VEHICLE TELEMETRY DATA COLLECTION

The goal of this use-case is to evaluate the system's capability to collect real-time telemetry data from connected vehicles. This involves testing the system's ability to handle telemetry data from both simulated and real vehicles, ensuring accurate data capture, and assessing the system's performance under various conditions. The proposed experiment includes three main sources of data telemetry: a real connected vehicle,

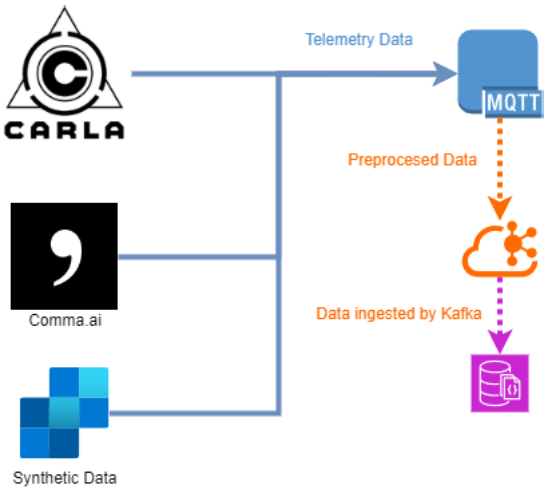

FIGURE 5. Vehicle data acquisition simulation.

a simulated vehicle, and data provided by synthetic data simulating a larger number of connected vehicles.

The real connected vehicle is equipped with a comma unit that provides connectivity and a direct connection to the vehicle's CAN-bus [39], [40]. It can deliver vehicle information such as speed, GPS location, and other road data. The comma unit runs the Open Pilot, an open-source software, which can be fed data from virtual cars simulated in the CARLA environment [41], [42] a platform for generating virtual traffic and driving scenarios. Additionally, synthetic data generator, a Python program, is also used to generate data like those produced by CARLA. In the proposed experiment, the server is responsible for receiving data from two (2) real connected vehicles equipped with their respective comma units, three (3) simulated vehicles in the CARLA environment, and 45 streams of information provided by the synthetic data generator, as illustrated in figure 5.

Data collection from real and simulated data telemetry uses the paho-mqtt service from the open-source MQTT library which can collect information from the different sources and vehicles.

D. PERFORMANCE METRICS

The main metric used to evaluate system performance was the total ingestion duration for different configurations (raw, throttled, resized). Additional qualitative observations were made regarding system responsiveness and robustness under concurrent data streams.

IV. RESULTS AND ANALYSIS

To assess the real-world feasibility and performance of the proposed infrastructure, we conducted two experimental validations using deployed prototypes. The first use-case involved a multichannel camera gantry for image acquisition and ingestion, while the second focused on onboard vehicle telemetry via IoT devices. Both scenarios were designed to simulate conditions representative of connected and automated roads. We evaluated ingestion performance based on total data upload duration under various conditions, including resolution reduction and transmission throttling. The following sections describe each test and analyze the results obtained.

A. TRAFFIC DATA CAPTURE VIA CAMERAS ON GANTRIES OR ROUNDABOUTS

The traffic data capture experiment has allowed us to analyze the system's capacity to handle large volumes of data. The tests assessed the efficiency of the system in terms of data ingestion and processing, as well as its ability to maintain data integrity and accuracy. The results of the four tests carried out to analyze the experiment were as follows:

- The first test deals with sending all images as they were processed, with no time limitation between uploads. Approximately 48 GB of video frame stream images, simulating cameras, were uploaded to the data server. The total processing and upload time was 3,401 seconds.
- For the second test, two scenarios were tested: uploading data with a time interval of one second and half a second. In the first case, all images were sent without preprocessing, but the processing and upload time increased significantly to 4,203 seconds. Reducing the interval to half a second improved performance, with an upload time of 3,738 seconds, although it was still slower than uploading raw data.
- In the third test, was tried to reduce the size of the images using two scales: 50% and 25% of the original image size. The 50% reduction resulted in a load time of 1,270 seconds, showing much better performance than loading without pre-processing. The 25% reduction improved the loading time to 776 seconds.
- The last test combined different data loading proposals to find the best resolution/time ratio. Two tests were performed: one with a 50% reduction and a half-second delay, resulting in a total processing time of 1,855 seconds, and one with a 50% reduction and a one-second delay, resulting in 2,098 seconds. The best approach was to send the 50% down sampled images with a half-second delay between uploads, achieving near real-time data ingestion.

To validate the results, the experiments were evaluated based on the execution and data processing time from start to finish. These results are summarized in Table 4.

The results obtained from the image ingestion experiments reveal that the raw upload scenario (3401 s) establishes a baseline for performance. Introducing a 1-second delay (4203 s) was counterproductive, adding overhead and increasing total duration by nearly 24%. Conversely, using a 0.5-second delay (3738 s) provided marginal improvement but remained above the baseline. Image resolution reduction strategies significantly improved performance. Reducing images to 50% of their original resolution decreased the upload time to 1270 seconds—a 62% reduction compared

TABLE 4. Data ingestion test results.

Test	Variation	Duration (seconds)
Raw Data Upload	-	3401 s
Timed Data Upload Between Image Transmission	1-second wait	4203 s
	0.5-second wait	3738 s
Data Upload with Image Resolution Reduction	Resolution reduced to 50%	1270 s
	Resolution reduced to 25%	776 s
Cross Tests	Resolution reduced to 50% with 1-second wait	2098 s
	Resolution reduced to 50% with 0.5-second wait	1855 s

to raw data. The 25% resolution test achieved the best standalone result (776 s), with a 77% decrease. Combining resolution reduction and throttling produced balanced performance. The configuration with 50% resolution and a 0.5-second delay achieved 1855 s, which, while not as fast as the pure 25% case, maintained higher image quality while still reducing upload time by 45%. These results suggest that resolution reduction is the most effective single strategy for improving ingestion time. However, combining both methods may be preferred when balancing bandwidth constraints and the need to preserve visual quality for later computer vision processing (See Figure 6).

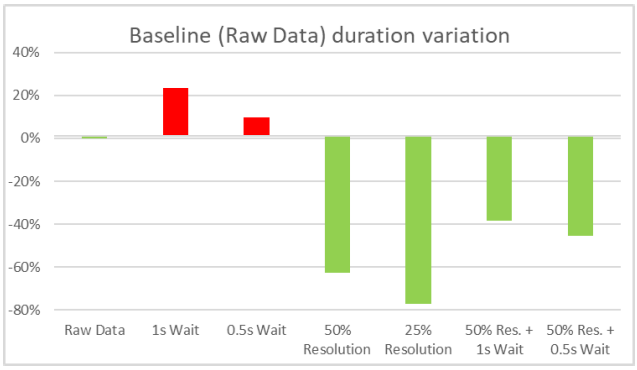

FIGURE 6. Comparison of shipping time using the raw data as a reference.

Figure 7 visually compares the data ingestion durations under different configurations and highlights the impact of various pre-transmission strategies.

The baseline scenario, with no optimization, required 3401 seconds to upload the raw data. Introducing a 1-second delay between image transmissions increased the duration to 4203 seconds, while a 0.5-second delay reduced it slightly to 3738 seconds. A significant improvement was observed when reducing the resolution of the transmitted images, particularly with a 25% resolution setting, which lowered ingestion time to 776 seconds. The most balanced result was achieved in the cross test combining 50% resolution and a 0.5-second delay, resulting in 1855 seconds. To better illustrate the contribution of each factor to the total upload duration, we decomposed the results into three estimated components: Base, Delay,

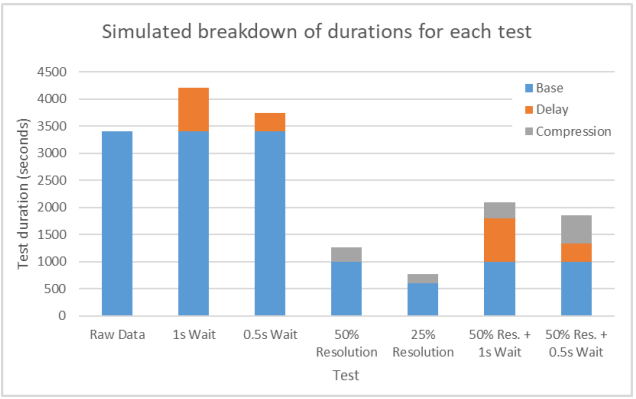

FIGURE 7. Simulated breakdown of data ingestion durations under different test conditions.

and Compression. The Base corresponds to the time required to upload raw data without modifications, as observed in the baseline scenario. The Delay represents the additional time caused by intentionally spacing out transmissions, estimated by comparing the timed uploads with the baseline. The Compression component accounts for both the savings in transmission time due to reduced image size and any additional processing overhead. In the combined tests, the total duration was broken down into a reduced base (based on image resolution), the known delay, and a residual time attributed to compression. While these values are approximate and not directly measured, they provide an intuitive breakdown of how each strategy contributes to optimizing the overall data ingestion process.

With these results it can be assumed that the system can maintain a high sampling rate in data ingest even under various operating conditions. The reduction of the image resolution and the introduction of time intervals between transmissions have made it possible to analyze different strategies to optimize the performance of the system.

B. VEHICLE TELEMETRY DATA CAPTURE

As far as the vehicles' telemetry is concerned, the test lasted 602 seconds, the time required for the 2 vehicles equipped with the comma.ai autopilot to complete the assigned route. Simultaneously, the 45 simulated vehicles in CARLA sent their telemetry data alongside them. The transmission time-period was set to 0.5 second, resulting in 1,204 frames of 58 individual information for each of the 48 vehicles, totaling 3,491,600 data points lasting the experiment a total of 602 seconds. During the experiment, all telemetry data from both simulated and real vehicles was successfully captured. The IoT server received real-time signals from the simulated vehicle via the modified Open Pilot software, with no messages lost during transmission. These parameters included essential data such as speed, position, and direction, as well as more detailed information like the road curvature.

The successful transmission and capture of data demonstrated the system's robustness and reliability in handling

high-frequency telemetry data. The ability to accurately record and transmit large volumes of data in real-time is crucial for applications in autonomous driving and vehicle monitoring. The experiment also highlighted the system's capability to filter and organize data from multiple sources, ensuring that each data point could be traced back to its respective vehicle. Furthermore, the analysis of the data collected provided insights into vehicle behavior and performance under different conditions. This included observing how vehicles responded to various driving scenarios and environmental factors. The detailed telemetry data allowed for a comprehensive evaluation of the vehicles' operational parameters, contributing to the development and refinement of autonomous driving algorithms.

No data loss was observed during the transmission from the vehicles to the server via the MQTT protocol and the system successfully processed information from different formats and sources, validating its modularity and interoperability. These findings are particularly relevant for connected vehicle ecosystems, where robust telemetry handling is essential for navigation and safety systems. Overall, the experiment confirmed that the system could effectively manage and process real-time telemetry data from many vehicles, providing a reliable foundation for further advancements in autonomous vehicle technology.

Although both the image ingestion and telemetry experiments were conducted under stable network conditions, the system demonstrated the ability to handle high-frequency data streams (0.5-second intervals) from 50 concurrent sources without data loss or processing delays. Latency benchmarks were not explicitly measured under varying network conditions such as packet loss, jitter, or bandwidth fluctuations. However, the architecture is designed to tolerate such scenarios. The use of MQTT for telemetry enables buffering during intermittent connectivity, and Kafka ensures message durability and replay capabilities. Future work will include controlled experiments to evaluate system performance under degraded or unstable network environments, which are common in real-world vehicular and roadside deployments.

V. DISCUSSION

The implementation of a real-time data collection system for vehicles on automated roads presents several challenges and opportunities that require thorough analysis. A primary challenge is the efficient management of large volumes of real-time data. While experiments have demonstrated the feasibility of using technologies such as Apache Kafka and Apache Beam to manage data streams, scalability and latency remain critical concerns. The ability of these systems to maintain optimal performance in dense traffic and high demand scenarios must be continually evaluated. In addition, while the integration of SQL and NoSQL databases is effective, it requires careful monitoring to avoid data consistency and redundancy issues. Furthermore, reducing image resolution and introducing delays between image transmissions have proven to be effective strategies for improving system efficiency. However, it is essential to consider the impact of these optimizations on data. Excessive resolution reduction may hinder computer vision algorithms' ability to detect and analyze vehicles and other objects on the road accurately. Therefore, a balance between system efficiency and data quality is crucial. Previous studies have shown that reducing image resolution can negatively impact the accuracy of object detection algorithms, particularly for small or partially occluded objects [43], [44]. In our case, resolution reduction was applied solely during the ingestion phase to optimize bandwidth and processing time, not during the execution of detection models. However, to preserve a balance between efficiency and data quality, we selected a 50% resolution reduction as a compromise that maintained sufficient visual detail for future analysis. Future work will include a quantitative evaluation of object detection accuracy under different resolution settings to better understand this trade-off. On the other hand, the use of IoT devices for data collection introduces additional complexities, such as ensuring reliable connectivity and managing the various data formats generated by multiple sensors. For this reason, the MQTT protocol, as a lightweight messaging protocol, has shown promise in efficiently managing telemetry data from resource-constrained devices such as Open Pilot. For vehicle telemetry collection, optimizing the data transmission frequency and managing the volume of telemetry data is critical. Experiments showed that the system could effectively handle high-frequency telemetry data, providing valuable information on vehicle behavior and performance.

To replicate this system in a real environment, several factors must be considered. The current server implementation does not support horizontal scalability, so it is necessary to limit the equipment to a subset of cameras managing information only for these points. Expanding the network will require additional servers to manage more traffic points across an analyzed area. Another option is to employ a cluster of distributed machines to manage the area, using containers to acquire more computational resources as needed.

Alternatively, the proposed systems can be improved using cloud services, but this would involve relying on an external provider to manage and operate the system. Compared to traditional V2X architectures, which often rely on simulation-based evaluations or focus on isolated subsystems, the proposed infrastructure offers a practical and modular solution validated with real-world data. While many existing systems emphasize theoretical scalability or algorithmic performance, our approach demonstrates the feasibility of integrating heterogeneous data sources, such as high-frequency telemetry and large-scale image streams, into a unified, real-time processing pipeline. Recent research, such as the V2X-UniPool framework [45], has highlighted the benefits of multimodal data fusion for autonomous driving using real-world datasets. Our system complements these efforts by focusing on infrastructure-level ingestion and processing, offering a foundation for future extensions

involving AI-based decision-making and large-scale deployments. From an operational point of view, the ability to collect and analyze real-time data can significantly improve traffic management and road safety. V2X communication systems and advanced sensors provide detailed information on traffic flow and road conditions, enabling proactive data-driven management. However, the effectiveness of these technologies depends on the quality and accuracy of the data collected, as well as the ability of traffic operators to interpret and act on this information in a timely manner.

- Real-time data collection and analysis enables more efficient and proactive traffic management, optimizing vehicle flow, reducing travel times and minimizing congestion.
- This system enables autonomous and connected vehicles to receive real-time alerts about hazardous conditions, improving emergency responses and reducing the risk of human error.
- The proposed infrastructure supports more efficient resource management by automating key processes and reducing the need for manual intervention, lowering operational costs.

Finally, it is crucial to recognize that the implementation of a big data infrastructure for traffic recording and analysis is only the first step towards a truly intelligent and autonomous transport system. Future research should focus on continuous system optimization, integrating emerging technologies and fostering collaboration between researchers, car manufacturers and traffic authorities.

VI. CONCLUSION

In this paper, a big data infrastructure designed for traffic optimization on automated roads has been presented. The implementation of the proof-of-concept demonstrates the system's ability to handle large volumes of data in real time through two use-cases: capturing traffic images using cameras at gantries and roundabouts and collecting telemetry data from vehicles.

The proposed infrastructure has shown remarkable efficiency in managing the ingest of traffic image and telemetry data, maintaining data integrity and accuracy under various conditions. Strategies such as reducing image resolution and introducing time intervals between transmissions have significantly improved system efficiency without compromising data quality. The results highlight the system's potential for practical applications in traffic management, providing valuable information for real-time decision making and improving road safety.

Technologies such as Apache Kafka, Apache Beam, IoT and MQTT have shown to be effective, although scalability and latency remain critical areas needing attention. Continuous system optimization, integration of emerging technologies such as 5G and advanced artificial intelligence, and collaboration between researchers, car manufacturers and traffic authorities are essential for the development of a truly intelligent and autonomous transport system.

In summary, this work lays a solid foundation for future implementations and improvements in intelligent transport infrastructures, highlighting the importance of data quality and operational efficiency in traffic management and road safety. The ability to collect and analyze real-time data not only optimizes traffic flow and reduces congestion, but also improves emergency response and lowers operational costs, laying the foundation for a safer and more efficient transport system.

This work has made it possible to consider new lines considering the development of an arbitration system that allows connected autonomous vehicles to make decisions based on the data they possess, and the data captured by this system from the road they are on. This arbitration system would integrate real-time data from multiple sources, allowing vehicles to assess traffic conditions, identify potential risks and optimize their routes and drive autonomously. In the same way, future research should focus on integrating emerging technologies such as advanced artificial intelligence and 5G to improve processing power and system latency. In addition, it is crucial to foster collaboration between researchers, car manufacturers and traffic authorities to ensure that the solutions developed are practical, scalable and aligned with the needs and expectations of all stakeholders.

Finally, the implementation of real-time data capture systems raises important security and privacy concerns. The collection and transmission of sensitive data, such as vehicle location and speed, must be protected against unauthorized access and cyber threats. In addition, it is essential to ensure compliance with privacy and data protection regulations, such as GDPR in Europe. Implementing robust security measures and adopting data anonymization practices are vital to mitigate these risks.

ACKNOWLEDGMENT

The authors would like to thank Universidad Francisco de Vitoria and Universidad Europea de Madrid for their support.

REFERENCES

[1] T. Mihalj, H. Li, D. Babić, C. Lex, M. Jeudy, G. Zovak, D. Babć, and A. Eichberger, "Road infrastructure challenges faced by automated driving: A review," *Appl. Sci.*, vol. 12, no. 7, p. 3477, Mar. 2022, doi: 10.3390/app12073477.

[2] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, G. Fortino, and M. Villari, "A multi-agent autonomous intersection management (MA-AIM) system for smart cities leveraging edge-of-things and blockchain," *Inf. Sci.*, vol. 522, pp. 148–163, Jun. 2020, doi: 10.1016/j.ins.2020.02.059.

[3] P. Sun, D. Nam, R. Jayakrishnan, and W. Jin, "An eco-driving algorithm based on vehicle to infrastructure (V2I) communications for signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 144, Nov. 2022, Art. no. 103876, doi: 10.1016/j.trc.2022.103876.

[4] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816, doi: 10.1016/j.eswa.2020.113816.

[5] S. Tang, Z. Zhang, Y. Zhang, J. Zhou, Y. Guo, S. Liu, S. Guo, Y. Li, L. Ma, Y. Xue, and Y. Liu, "A survey on automated driving system testing: Landscapes and trends," *ACM Trans. Softw. Eng. Methodology*, vol. 32, no. 5, pp. 1–62, Jul. 2023, doi: 10.1145/3579642.

[6] Y. Liu, M. Tight, Q. Sun, and R. Kang, "A systematic review: Road infrastructure requirement for connected and autonomous vehicles (CAVs)," *J. Phys., Conf. Ser.*, vol. 1187, no. 4, Apr. 2019, Art. no. 042073, doi: 10.1088/1742-6596/1187/4/042073.
[7] *NAE Website–The Role of Infrastructure in an Automated Vehicle Future*. Accessed: Aug. 1, 2024. [Online]. Available: https://www.nae.edu/183200/The-Role-of-Infrastructure-in-an-Automated-Vehicle-Future
[8] H. Zhou, K. Jiang, S. He, G. Min, and J. Wu, "Distributed deep multi-agent reinforcement learning for cooperative edge caching in Internet-of-Vehicles," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9595–9609, Dec. 2023, doi: 10.1109/TWC.2023.3272348.
[9] M. N. Tahir, P. Leviäkangas, and M. Katz, "Connected vehicles: V2V and V2I road weather and traffic communication using cellular technologies," *Sensors*, vol. 22, no. 3, p. 1142, Feb. 2022, doi: 10.3390/s22031142.
[10] B. L. Nguyen, D. T. Ngo, N. H. Tran, M. N. Dao, and H. L. Vu, "Dynamic V2I/V2V cooperative scheme for connectivity and throughput enhancement," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1236–1246, Feb. 2022, doi: 10.1109/TITS.2020.3023708.
[11] M. M. Rana and K. Hossain, "Connected and autonomous vehicles and infrastructures: A literature review," *Int. J. Pavement Res. Technol.*, vol. 16, no. 2, pp. 264–284, Mar. 2023, doi: 10.1007/s42947-021-00130-1.
[12] C. Qiu, H. Tang, Y. Yang, X. Wan, X. Xu, S. Lin, Z. Lin, M. Meng, and C. Zha, "Machine vision-based autonomous road hazard avoidance system for self-driving vehicles," *Sci. Rep.*, vol. 14, no. 1, pp. 1–15, May 2024, doi: 10.1038/s41598-024-62629-4.
[13] J. Santa, A. F. Gómez-Skarmeta, and M. Sánchez-Artigas, "Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks," *Comput. Commun.*, vol. 31, no. 12, pp. 2850–2861, Jul. 2008, doi: 10.1016/j.comcom.2007.12.008.
[14] D. Oladimeji, K. Gupta, N. A. Kose, K. Gundogan, L. Ge, and F. Liang, "Smart transportation: An overview of technologies and applications," *Sensors*, vol. 23, no. 8, p. 3880, Apr. 2023, doi: 10.3390/s23083880.
[15] J. Sánchez-Soriano, G. De-Las-Heras, E. Puertas, and J. Fernández-Andreés, "Sistema avanzado de ayuda a la Conducción (ADAS) en rotondas/glorietas usando imágenes aéreas y técnicas de inteligencia artificial para la mejora de la seguridad vial," *Logos Guardia Civil, Revista Científica del Centro Universitario de la Guardia Civil*, vol. 1, pp. 241–270, Jun. 2023. [Online]. Available: https://revistacugc.es/article/view/5708
[16] S. Kussl and A. Wald, "Smart mobility and its implications for road infrastructure provision: A systematic literature review," *Sustainability*, vol. 15, no. 1, p. 210, Dec. 2022, doi: 10.3390/su15010210.
[17] E. Ranyal, A. Sadhu, and K. Jain, "Road condition monitoring using smart sensing and artificial intelligence: A review," *Sensors*, vol. 22, no. 8, p. 3044, Apr. 2022, doi: 10.3390/s22083044.
[18] K. Jiang, H. Zhou, D. Zeng, and J. Wu, "Multi-agent reinforcement learning for cooperative edge caching in Internet of Vehicles," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Dec. 2020, pp. 455–463, doi: 10.1109/MASS50613.2020.00062.
[19] D. Meng, J. Guo, H. Zhou, Y. Zhang, L. Zhao, Y. Shu, and X. Fan, "Incentive-driven partial offloading and resource allocation in vehicular edge computing networks," *IEEE Internet Things J.*, vol. 12, no. 8, pp. 11023–11035, Apr. 2025, doi: 10.1109/JIOT.2024.3515075.
[20] J. Kreps. (2011). *Kafka: A Distributed Messaging System for Log Processing*. [Online]. Available: https://api.semanticscholar.org/CorpusID:18534081
[21] T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R. J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, and S. Whittle, "The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing," *Proc. VLDB Endowment*, vol. 8, no. 12, pp. 1792–1803, Aug. 2015, doi: 10.14778/2824032.2824076.
[22] R. A. Light, "Mosquitto: Server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, p. 265, May 2017, doi: 10.21105/joss.00265.
[23] *Documentation—MariaDB.org*. Accessed: Nov. 07, 2024. [Online]. Available: https://mariadb-org.translate.goog/documentation/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sc
[24] K. Chodorow, *MongoDB: The Definitive Guide*. O'Reilly Media, 2013. [Online]. Available: https://www.oreilly.com/library/view/mongodb-the-definitive/9781449344795/
[25] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *ACM SIGOPS Operating Syst. Rev.*, vol. 44, no. 2, pp. 35–40, Apr. 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:916681
[26] *Firestore | Firebase*. Accessed: Nov. 7, 2024. [Online]. Available: https://firebase.google.com/docs/firestore?hl=es-419
[27] *SQLite Documentation*. Accessed: Nov. 7, 2024. [Online]. Available: https://www.sqlite.org/docs.html
[28] *Amazon Kinesis Video Streams-Incorporación De Vídeo Segura Para Análisis Y Almacenamiento-Amazon Web Services*. Accessed: Nov. 7, 2024. [Online]. Available: https://aws.amazon.com/es/kinesis/video-streams/?nc=sn&loc=2&dn=1&amazon-kinesis-video-streams-resources-blog.sort-by=item.additionalFields.createdDate&amazon-kinesis-video-streams-resources-blog.sort-order=desc
[29] *RabbitMQ Documentation | RabbitMQ*. Accessed: Nov. 7, 2024. [Online]. Available: https://www.rabbitmq.com/docs
[30] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, Oct. 2016, doi: 10.1145/2934664.
[31] *Documentación De Dataflow | Google Cloud*. Accessed: Nov. 7, 2024. [Online]. Available: https://cloud.google.com/dataflow/docs?hl=es-419
[32] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink: Stream and batch processing in a single engine," *IEEE Data Eng. Bull.*, vol. 38, no. 4, pp. 28–38, Dec. 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:263802939
[33] J. C. Anderson, J. Lehnardt, and N. Slater, *CouchDB: The Definitive Guide: Time To Relax*. O'Reilly Media, 2010. [Online]. Available: https://dl.acm.org/doi/10.5555/2544030
[34] *Apache Kafka*. Accessed: Aug. 8, 2024. [Online]. Available: https://kafka.apache.org/
[35] *Eclipse Mosquitto*. Accessed: Aug. 8, 2024. [Online]. Available: https://mosquitto.org/
[36] Å. Hugo, B. Morin, and K. Svantorp, "Bridging MQTT and kafka to support C-ITS: A feasibility study," in *Proc. 21st IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2020, pp. 371–376, doi: 10.1109/MDM48529.2020.00080.
[37] S. Bemposta Rosende, S. Ghisler, J. Fernández-Andrés, and J. Sánchez-Soriano, "Dataset: Traffic images captured from UAVs for use in training machine vision algorithms for traffic management," *Data*, vol. 7, no. 5, p. 53, Apr. 2022, doi: 10.3390/data7050053.
[38] E. Puertas, G. De-Las-Heras, J. Fernández-Andrés, and J. Sánchez-Soriano, "Dataset: Roundabout aerial images for vehicle detection," *Data*, vol. 7, no. 4, p. 47, Apr. 2022, doi: 10.3390/data7040047.
[39] L. Chen, T. Tang, Z. Cai, Y. Li, P. Wu, H. Li, J. Shi, J. Yan, and Y. Qiao, "Level 2 autonomous driving on a single device: Diving into the devils of openpilot," 2022, *arXiv:2206.08176*.
[40] L. Baresi and D. A. Tamburri, "Architecting artificial intelligence for autonomous cars: The OpenPilot framework," in *Proc. Eur. Conf. Softw. Archit.*, 2023, pp. 189–204, doi: 10.1007/978-3-031-42592-9_13.
[41] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," *arXiv:1711.03938*.
[42] *CARLA Simulator*. Accessed: Aug. 8, 2024. [Online]. Available: https://carla.org//
[43] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.
[44] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-CNN for small object detection," in *Proc. ACCV*, Mar. 2017, pp. 214–230, doi: 10.1007/978-3-319-54193-8_14.
[45] X. Luo, F. Yang, F. Ding, X. Gao, S. Xing, Y. Zhou, Z. Tu, and C. Liu, "V2X-UniPool: Unifying multimodal perception and knowledge reasoning for autonomous driving," 2025, *arXiv:2506.02580*.

JORGE GARCÍA-GONZÁLEZ received the bachelor's degree in computer engineering and the master's degree in big data analytics from European University of Madrid, Spain, in 2021 and 2023 respectively. In 2021, he joined the Department of Computing and Technology, as an Associate Professor. His research interests include machine learning, autonomous driving, computer vision, and intelligent transportation systems.

JAVIER FERNÁNDEZ-ANDRÉS received the degree in industrial engineering and the Ph.D. degree in robotics and computer vision from the Polytechnic University of Madrid, Spain, in 1992 and 1998, respectively. In 1998, he joined the Computer Systems Department, European University of Madrid, as an Associate Professor, where he has been a Professor, until 2004. From 2004 to 2012, he was the Department Chair of Computer Systems and Automation. Since 2012, he is a Full Professor with the Department of Engineering, European University of Madrid. His research interests include computer vision, intelligent transportation systems, pattern recognition, and machine learning.

NOURDINE ALIANE received the degree in electrical engineering from École Nationale Polytechnique d'Alger, in 1990, and the Ph.D. degree in physics from Universidad Complutense de Madrid, in 1995. He is currently a Professor with the Engineering Department, Universidad Europea de Madrid, Spain. His research interests include intelligent transportation systems and control systems.

JAVIER SÁNCHEZ-SORIANO received the degree in computer engineering, the master's degree in information technologies, and the Ph.D. degree in artificial intelligence from the Polytechnic University of Madrid, Spain, in 2009, 2010, and 2016, respectively. In 2012, he joined the Computer Systems Department, European University of Madrid, as an Associate Professor, where he has been a Professor, until 2021. Since 2021, he has been an Associate Professor with the Polytechnic School of the Universidad Francisco de Vitoria. His research interests include machine learning, autonomous driving, computer vision, and intelligent transportation systems.