Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 27 June, 2024.

Digital Object Identifier 10.1109/OJVT.2024.0627000

Roads Infrastructure Digital Twin: Advancing Situational Awareness through Bandwidth-Aware 360° Video Streaming and Multi-View Clustering

Oussama El Marai*, Sotirios Messinis†, Nikolaos Doulamis†, Tarik Taleb‡, and Jukka Manner*

[1] Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland.
[2] Institute of Communications and Computer Systems, National Technical University of Athens, Greece.
[3] Ruhr University Bochum, Bochum, Germany.

CORRESPONDING AUTHOR: Oussama El Marai (e-mail: oussama.elmarai@aalto.fi).

ABSTRACT Future-facing cities increasingly integrate smart and autonomous objects for their smooth functioning and operations, which ultimately benefit city dwellers and the ecosystem at large. In such highly complex and digital environments, the increased situational awareness is very important for the safety of road participants. In this paper, we propose a new bandwidth-aware framework that maximizes the situational awareness of a given region, using mobile digital boxes and 360° cameras, mounted on connected vehicles, taking into account the constrained uplink capacity. The proposed framework leverages the multi-view spectral clustering approach and the K-Means++ algorithms to ensure efficient clustering of vehicles based on their GPS coordinates. The clustering step is crucial for larger spatial coverage and, thus, higher situational awareness. Vehicle selection and video quality attribution, under limited uplink constraints, are then performed per cluster to fairly cover the region. Extensive simulations and comparisons against state-of-the-art solutions have been conducted to evaluate the performance of the proposed framework, in terms of region coverage rate and normalized mutual information score, at both small- and large-scale deployments. The results obtained demonstrate the superiority of the proposed approach.

INDEX TERMS Situational awareness, connected vehicles, multi-view clustering, digital twin, 360° streaming, Dynamic Adaptive Streaming over HTTP, smart cities.

I. Introduction

SMART Cities is a hot topic that is gaining ample traction nowadays due to its wide impact on most aspects of human life. It uses Information and Communication Technology (ICT) to improve the lifestyle of city dwellers and protect the ecosystem while ensuring future sustainability. To a greater extent, smart cities promise to address most challenges of current cities such as crowded roads, preserve energy, foster safety, reduce gas emissions, promote intelligent transportation, and improve social life. The surging number of people at an unbridled rate, especially in large and dense cities accounting for most of the world's population, urges governments to harness emerging technologies in favor of smarter and more connected cities. To do so, we need to bridge the chasm between a gamut of essential smart services and systems, where each has its own data sources.

A key feature of smart cities is that different objects should be connected and have the ability to exchange information with both infrastructure (e.g., edge and cloud) and other objects via a variety of network technologies. A very promising technology to achieve this lofty goal is the Internet of Things (IoT) paradigm. It is a revolutionary technology that empowers *things* with intelligence and ubiquitous connectivity to the Internet, enabling physical objects to be sensed and controlled remotely [1]. The abundant amount of data generated from billions of IoT devices can be used to create the digital twin of their corresponding physical objects. By doing so, we are actually creating a virtual world in cyberspace that reflects what is happening in the real world [2]. The more diversified the deployed sensors, the higher the accuracy of the digital twin environment. The collected data can be leveraged to monitor and analyze the physical asset, ultimately improving its performance in different aspects.

One crucial aspect that is required in smart cities is the per-object and per-region Situational Awareness (SA). It heavily relies on both real-time and historical data from the road's participants. While the per-object SA aims to raise the awareness of the road's participants about their surroundings, the objective of the per-region SA is to enable the city authorities to keep an eye on the different streets for enhanced safety. This is mainly achieved through live video streams coming from the streets. In addition, recent advances in integrated sensing and communication between the so-called Internet of Vehicles (IoV) [3] pave the way for more innovative combined solutions to address the recent challenges of SA considering both video streaming and advanced vehicle-to-infrastructure (V2I) communications.

In a previous study [4], the authors proposed to install Digital Twin Boxes (DTB) to create the digital twin of the roadways. Two types of DTB were proposed, namely static and mobile, whereby static DTBs could be deployed on, for instance, street and traffic light poles. However, this kind of deployment is too costly and can only be applied in some specific areas. Alternatively, mobile DTBs can be carried on board moving vehicles, including cars, bicycles, motorcycles, buses, and even Unmanned Aerial Vehicles (UAVs) [5]–[7], with the vehicle owner bearing the purchase and maintenance costs. These mobile DTBs will dynamically cover much larger surfaces, thus increasing overall SA.

Focusing on ground vehicles, this paper investigates the problem of increasing the SA for a specific region based on video stream feeds, while optimizing its delivery under constrained uplink capacity. The ultimate goal is to cover the maximum surface of the area of interest (AoI) at optimal video resolution for better monitoring and computer processing (e.g., for object detection and recognition). In bandwidth-constrained scenarios, our solution ensures efficient network utilization by preventing redundant streams in densely covered areas, where additional streams offer diminishing returns, in favor of giving room to other vehicles in underserved or uncovered regions. This maximizes situational awareness by preventing clustered vehicles from exhausting the network bandwidth to cover an area which is already covered by other streams. In this work, we consider the use of the dominant video streaming technique, namely Dynamic Adaptive Streaming over HTTP (DASH) [8]. The motivation beneath DASH lies in its high adaptation flexibility in response to varying network conditions at chunk-level granularity. Also, in this work, we assume 360° video streams to acquire scenes omnidirectionally [9], which would allow us to disregard the vehicle's heading direction in the proposed solution. Ideally, all vehicles located in the AoI would be allowed to stream at the highest resolution. However, the generated traffic from a large number of active live-streaming vehicles would create an uplink bandwidth crunch, likely to severely deteriorate the user's Quality of Experience (QoE) [10]. This compels us to employ a control mechanism that takes into consideration both the vehicles' positions, for the SA, as well as the received video quality from each vehicle to avoid overwhelming the shared link which leads to overall QoE degradation. It is worth noting that the live streams from a subset of selected vehicles are transmitted to a remote center for an increased SA of the targeted area, not for the operational functioning and manoeuvering of the vehicles, which are basically performed in the vehicles.

To confront the aforementioned challenges, we devise an unsupervised learning approach, deployed on the server, for efficiently clustering all the vehicles in a region with respect to specific distance-based criteria. The design choice is motivated by the nature of the addressed problem, which compels us to address the trade-off between the optimal area coverage and the bandwidth allocated to that area. Our approach contributes to the preliminary phase of choosing which vehicle will stream and at which video quality. After that, computationally speaking, it would be more efficient to perform the learning with relatively small "data islands" [11] on limited computation resources and transfer the learning parameters of the chosen individual vehicles to train a more global model at the server side.

The remainder of this paper is organized as follows. Section II presents the current status of machine learning implementation in video streaming and situational awareness problems in the recent literature. Section III formulates the target problem introducing basic notions towards the suggested solution. Section IV describes the proposed framework of our solution to the joint optimal situational awareness and bandwidth allocation problem. Section V evaluates and compares the performance results of our solution against previous work in different scale scenarios and our conclusions are given in Section VI.

II. Related Work

A crucial requirement of smart cities consists of maintaining a high level of awareness of their environment for a timely reaction when necessary. This can be achieved by creating the digital twin of the roads by deploying different sensors not only at the strategic spots of the roads but also along the ways and streets. While the strategic spots can be covered with static poles (e.g., on top of traffic lights or roadside units RSUs), the other segments of the roads, which constitute the majority, can be covered using moving vehicles such as cars, bicycles, buses, and trains. A highly important sensor that has been always used for monitoring is traditional video streaming. The use of 360° video streams would offer better awareness by capturing spherical streams per camera [12]. In addition, the use of VR devices, such as head-mounted displays (HMDs), would ease the monitoring task. However, due to resource limitations (e.g., uplink limitation) and unnecessary content redundancy, it is not possible to allow all existing video sources to stream, especially in highly dense areas [13]. Hence, a video source selection process should be employed. In recent years, machine learning (ML) techniques have proved to be highly efficient in making intelligent decisions for various complex problems in different domains including vehicular communications [14], which motivates us to leverage them in this work. In this section, we review previous work related to digital twinning, 360° video streams, and machine learning applications in the context of situational awareness.

A. Digital Twins

Digital twins are a popular technology that progressively integrates recent advancements in the IoT and ML scientific fields for the optimal representation of a physical asset to its virtual prototype in the digital world. In the framework of Industry 4.0 objectives, this kind of technology was initially deployed for manufacturing and product development cases as well as for their life cycle assessment and monitoring [15]. Recently, there have been many efforts in both industry and academia on the scalability of the digital world representation of more complex systems like smart cities and supply chains. In the context of smart cities, this technology enables data collection from various sources. In sequence, it will enable extensive and sophisticated simulations of urban processes for creating social and economic benefits [16]. As a result of this direction, multidimensional aspects and deeper insights can be better exploited for decision-making across multiple interests, including real-time situational awareness of the system, multilateral insurance schemes, disaster management, and strategic planning [17].

In the agricultural field, specifically in aquaculture farming, the authors of [18] propose a new framework for Planetary Digital Twin to monitor, simulate, and control aquaculture processes in nearly real-time, ultimately to support and foster the *AquaGreen* goals. In the proposed framework, traditional IoT devices are used, while AI-based models are employed to detect and control water anomalies through a closed-loop feedback framework. Using IoT devices and cloud computing services, the authors of [19] propose an end-to-end system architecture and implementation for environmental monitoring, particularly carbon dioxide concentration. In [20], the author addresses the problem of processing real-time data collected from various systems in the Smart City. A large-scale data processing architecture and proof-of-concept implementation have been proposed for public transport systems to achieve this goal. This architecture monitors multiple activities of different actuators in the Smart City, creates their Digital Twin, processes the large-scale data to understand the current state of the city, and ultimately produces actions to be executed in the physical space. In addition, machine learning models are used to restrict the amount and type of processing.

B. DASH-based 360° Video Streaming

Nowadays, 360° streaming is a hot topic due to the vivid user experience that it offers clients, especially when it is viewed through HMDs. Initially, its use has been mostly in gaming, but it has rapidly expanded to various other domains such as education and healthcare. However, this technology has been suffering numerous hindrances at different levels, including acquisition, transmission, and display. The phenomenal technological advances at both the hardware and network levels have paved the way for 360° streaming technology to be increasingly adopted by both content producers and consumers. For example, the 5G breakthrough in link capacity and lower latency and the various network slicing frameworks [21], [22] shall enable the end-users to enjoy such immersive and bandwidth-hungry service in both shifted and live modes [23], [24]. This paper focuses on using 360° streaming for digital twinning the roads, which can be used for limitless purposes such as increasing the situational awareness of a specific environment through live and shifted roads' monitoring applications. For aerial video-based surveillance and awareness, we refer the reader to [25]. This section describes previous work leveraging 360° video streams for monitoring purposes as well as pull-based 360° streaming techniques. The latter exhibits high flexibility in controlling who can stream and at which video quality based on many factors such as the busyness of the roads and the uplink capacity [26]. This flexibility stems from the pull-based paradigm of the DASH technology which enables a content producer to intelligently control the encoding rate based on the link variation.

An AQ360 system is proposed in [27], which relies solely on 360° streams captured by unmanned aerial vehicles (UAVs) for monitoring air quality. The reason for using UAVs is that they are able to provide images with greater depth, which is not possible with ground-based images due to obstacles. The authors also proposed a location selection algorithm that reduces system energy consumption in large-range areas. The evaluation of the proposed method shows that it achieves high accuracy by using a pre-trained inference model on small data sets while it consumes less energy in larger areas.

In a similar vein to traditional streaming, 360° video content can be delivered over the network via different low latency protocols such as Real-Time Messaging Protocol (RTMP), Real-Time Streaming Protocol (RTSP) [28]–[30], WebRTC, as well as the universal HTTP, using DASH [8] in all its variants. Despite the fact that DASH exhibits higher latency compared to the aforementioned protocols, as its name indicates, it offers the advantage of being much more flexible for adapting the video quality to changes in conditions such as fluctuating network and server workloads. There is however a need to amend the MPEG-DASH specifications to consider the spatial representation proposed in [9], [31]–[33] in order to support tile-based streaming, where the tiles within the active viewport are delivered at higher quality, while the rest are delivered with a lower quality to reduce the bandwidth usage.

C. Machine Learning Techniques for the Situational Awareness

Situational awareness is defined as the current perception of the environment or system status with predictive capabilities referring to future system states. Based on our application-specific scenario and setting, the system refers to a particular area including the passing-by vehicles and its road infrastructures. In this study, machine learning will contribute to the required SA of our system with the clustering of vehicles in a specific AoI and their subsequent approximate selection based on clustering properties. Some research studies with relevant machine learning applications in SA-oriented problems were recently presented in the relevant literature. A recent study in [34] presents a solution to mitigate the effects of adverse weather conditions that worsen traffic for vehicles by introducing a dual attention- and dual frequency-guided dehazing network for immediate visibility improvement. Their model uses an attention mechanism and an innovative frequency-related information fusion strategy to extract both general and local features, ensuring the recovery of clear high-frequency patterns and fine low-frequency details. In [35], the author presents the use of knowledge graphs for learning the representations of objects while preserving the required system information. With the use of path-based embeddings and graph neural networks, a joint object representation learning from multiple entities is experimentally achieved. In [36], the authors present deep learning techniques using multi-modal information from imagery, text, video, and other sensor sources for extending situational awareness performance. Deep Multimodal Image Fusion is selected as the most promising approach for distance-based system performance modeling. In [37], a serverless computing approach for multimodal computing is presented that enables the effective use of neural networks and machine learning approaches in relevant situation awareness implementations. Recently, the authors of [38] proposed an innovative framework that combines 360-degree video streaming with graph neural networks to create a road digital twin with enhanced predictive situational awareness capabilities for intelligent transportation systems.

D. Multi-view K-Means and Spectral Clustering

The traditional single-view K-means clustering is a centroid-based clustering method, which partitions the data space into a structure. Due to its low computational cost and easily parallel process, the K-means clustering method has often been applied to solve large-scale data clustering problems [39]. A usual challenge is when the data needs to be interpreted and represented by multi-view aspects. To overcome this challenge, a group of unsupervised multi-view clustering methods have been presented in the literature [40]. Although these methods can achieve interactions between heterogeneous features, there are still some problems regarding high computational complexity or the curse of dimensionality. Spectral clustering is also one of the most commonly used clustering methods due to its clear mathematical principles and good performance compared to other clustering methods. Multi-view clustering is a special clustering subcategory in the relevant literature that mainly refers to the partitioning of data considering multiple views, where the information from the individual views can be used together for more efficient cluster creation and visualization. These views may of course come from different sources or correspond to subsets of the same underlying feature space. To exploit the diverse and complementary information contained in different views, numerous multi-view clustering methods have been proposed [41]–[43]. To this end, multi-view spectral clustering aims to group data into different categories by optimally exploring complementary information from multiple Laplacian matrices.

In this paper, we propose a new framework that maximizes situational awareness in urban areas using DASH-based 360° live streams while preserving the overall users' QoE. The live streams originate from mobile DTBs onboard various vehicle types (e.g., cars, bicycles, etc.). The motivation behind opting for 360° is the spherical Field of View (FoV) which offers better awareness of the target area. The DASH-based streaming offers great flexibility in switching between different video qualities. In addition, it makes it easy to dynamically enable/disable the different DTBs, based on their GPS locations and the vehicular density of the area, at short time scales. It is worth noting that, in this work, we consider only the case of outdoor vehicles moving in urban areas without any specific restrictions to their GPS availability on the system.

For the efficacy of our solution to the problem of optimal bandwidth allocation with situational awareness constraints, we leverage ML techniques, namely the multi-view Spectral and K-Means clustering algorithms for the approximate optimal coverage of the AoI. It is noteworthy to mention that the choice of the specific algorithms is based on their performance advantages compared to other clustering algorithms and refers to the nature of our research problem [44]. Moreover, the multi-view aspect has a double advantage in the nature of our problem, as it can be shown to lead to better algorithmic performance with less computational complexity. This is further enhanced by dividing the AoI into smaller subdomains during the actual multi-view clustering process. The novelty of this work lies in its significant contribution to the situational awareness goals and requirements for future smart cities and their corresponding digital twins with the implementation of combined machine learning algorithms for joint near-optimal coverage and bandwidth allocation. To the best of our knowledge, this is the first work to address the specific problem of maximizing situational awareness using machine learning techniques.

III. Problem Formulation

The objective of this work is to maximize the situational awareness (SA) of a specific region $\mathcal{R}$. To better evaluate the covered region $\mathcal{R}$, we virtually divide it into $\mathcal{N}$ smaller circular and adjacent subareas $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots \mathcal{R}_{\mathcal{N}}\}$ of identical radius r. The center of each subarea $\mathcal{R}_n$ is denoted by λ_n representing the global positioning system (GPS) coordinates. Each subarea $\mathcal{R}_n$, $n \in [1, \mathcal{N}]$, contains a number of vehicles that might be equipped with mobile DTBs. We denote by $\mathcal{D}^n = \{\mathcal{D}^n_1, \mathcal{D}^n_2, \cdots \mathcal{D}^n_{\mathcal{M}}\}$ the set of available DTBs in a subarea $\mathcal{R}_n$, where $\mathcal{M}$ denotes the number of DTBs in a subarea. Each DTB $\mathcal{D}^n_m$, $n \in [1, \mathcal{N}]$, $m \in [1, \mathcal{M}]$, has a specific GPS coordinate $\mathcal{G}^n_m$. We denote by $\mathcal{G}^n = \{\mathcal{G}^n_1, \mathcal{G}^n_2, \cdots \mathcal{G}^n_{\mathcal{M}}\}$ the set of latest GPS coordinates, of a given snapshot of the system, corresponding to the set of DTBs $\mathcal{D}^n$ co-located in the same subarea $\mathcal{R}_n$.

In this work, we consider 360° video cameras capable of streaming a spherical view of the environment. For computational and demonstration reasons, we assume the cameras can have different values of lens focal length (LFL) $\mathcal{L} \in (r/4, r)$. It actually represents the distance ahead it could cover. These cameras stream live using DASH technique at $\mathcal{K}$ predefined video qualities $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \cdots \mathcal{Q}_{\mathcal{K}}\}, \mathcal{Q}_k, k \in [1, \mathcal{K}]$, where $\mathcal{Q}_1$ and $\mathcal{Q}_{\mathcal{K}}$ represent the lowest and highest video qualities, respectively. We denote by $\hat{\mathcal{D}^n_m} \in \mathcal{Q}$ the actual video quality selected by the mobile DTB $\mathcal{D}^n_m$. We also denote by $\mathcal{B}$ the total uplink bandwidth available in the whole region, where $\mathcal{B}^n$ denotes the fair bandwidth budget allocated to a subarea $\mathcal{R}_n$.

We define the matrix $\mathcal{P}^n$ that refers to all the distances among the different DTBs co-located within the same subarea $\mathcal{R}_n$, where both the rows and columns represent the DTBs $\mathcal{D}^n$ available in $\mathcal{R}_n$. The matrices $\mathcal{P}^n, \forall n \in \mathcal{N}$ are pre-calculated, based on the latest GPS coordinates $\mathcal{G}^n_m$ of each DTB $\mathcal{D}^n_m$, and provided as input to the system, where the cell values $\mathcal{P}^n[i, j], \forall i, j \in \mathcal{M}$ contain the distances between the DTBs $\mathcal{D}^n_i$ and $\mathcal{D}^n_j$. Obviously, the distance between the $\mathcal{D}_i$ and itself is always 0, i.e.,

$$\forall i = j, \forall i, j \in [1, \mathcal{M}], \mathcal{P}^n[i, j] = 0 \quad (1)$$

We also define a 1-dimensional matrix $\mathcal{T}^n$, $n \in [1, \mathcal{N}]$, per each subarea $\mathcal{R}_n$, to store the distances between all the DTBs in $\mathcal{D}^n$ and the center of the subarea λ_n. We denote by $||\mathcal{D}^n_m \lambda_n||$, the Euclidean distance between the m-th DTB in the n-th subarea and the center λ_n of the subarea $\mathcal{R}_n$, i.e.,

$$\forall n \in [1, \mathcal{N}], \forall m \in [1, \mathcal{M}], \mathcal{T}^n[m] = ||\mathcal{D}^n_m \lambda_n|| \quad (2)$$

We ensure that the minimum coverability of a region $\mathcal{R}_n$ by a DTB $\mathcal{D}^n_m$ is satisfied when:

$$||\mathcal{D}^n_m \lambda_n|| \leq r$$

We also denote by $\mathcal{C}^n = \{\mathcal{C}^n_1, \mathcal{C}^n_2, \cdots \mathcal{C}^n_{\mathcal{H}}\}$ the set of the computed clusters from the set of DTBs $\mathcal{D}^n$ in the subarea $\mathcal{R}_n$, where each cluster $\mathcal{C}^n_h, h \in [1, \mathcal{H}]$ contains a non-empty and disjoint subsets of DTBs, i.e.,

$$\forall h \in [1, \mathcal{H}], \mathcal{C}^n_h \subseteq \mathcal{D}^n \quad (3)$$

$$\forall h \in [1, \mathcal{H}], \mathcal{C}^n_h \neq \emptyset \quad (4)$$

$$\forall n \in [1, \mathcal{N}], \bigcup_{h=1}^{\mathcal{H}} \mathcal{C}^n_h = \mathcal{D}^n \quad (5)$$

$$\forall h \neq v, \forall h, v \in [1, \mathcal{H}], \mathcal{C}^n_h \cap \mathcal{C}^n_v = \emptyset \quad (6)$$

Similarly, we denote by $\widehat{\mathcal{C}^n_h}$ the centroid of the cluster $\mathcal{C}^n_h$.

In this work, we aim to enhance situational awareness in a region (a.k.a. the area of interest) by covering the maximum surface of that area with the FoV of the DTBs' cameras, while taking into account the number of video sources and the user's QoE. Due to the limited uplink capacity $\mathcal{B}$ and to preserve the QoE from degradation, only a subset of DTBs $\mathcal{S}^n \subseteq \mathcal{D}^n$ (called active DTBs) should be allowed to stream, where the DTB selection process uses their actual GPS coordinates $\mathcal{G}^n$. In addition to the DTB selection process, a subsequent video quality identification process is also performed to assign a video quality $\hat{\mathcal{D}^n_m} \in \mathcal{Q}$ to each DTB in $\mathcal{S}^n$. Figure 1 helps understand the main notations and concepts used in this section. The maximization of road situational awareness will then be addressed with the framework proposed in Section IV.

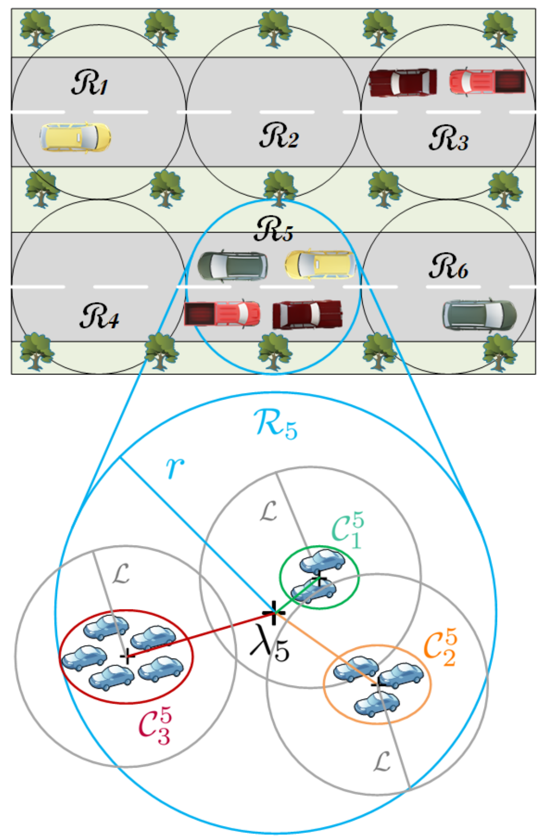

FIGURE 1. Cluster selection based on the distance to the center of the subarea.

IV. System Design and Proposed Framework

In this paper, we address the problem of maximizing the situational awareness of a given environment by creating its digital twin that mainly uses live video streams. Specifically, we target the creation of the digital twin of the road's infrastructure using 360° live streams. This can be achieved by deploying DTBs composed of single onboard computers, with different processing capabilities, that are equipped with various sensing devices such as cameras and

GPS devices [4]. Two types of DTBs are identified, namely static and mobile. In the present study, we primarily focus on the selection of mobile DTBs that are mounted onboard moving vehicles such as cars and buses. In dense areas (e.g., downtown), a potentially large number of connected vehicles will simultaneously livestream to create a digital twin of roadways, which would lead to a bandwidth crunch. A more critical situation might arise when 360° live streams are delivered, where very high frame rates and ultra-high definition are used. The dynamic vehicle selection and flexible video quality allocation are therefore necessary to cope with the coverage maximization across regions while considering the uplink capacity limitation. The objective of this work is to dynamically select relevant vehicles that allow better coverage of the targeted region while taking into account the limited uplink capacity $\mathcal{B}$ and preventing video stalls.

The key innovation of our study lies in the novel algorithm we have developed, which simultaneously achieves two key objectives: (1) maximizing situational awareness through optimal selection of geographically distributed vehicles using advanced clustering techniques, and (2) enhancing the video quality per region while ensuring fair bandwidth allocation across different regions. This is achieved through the development of road digital twins [4] based on distributed 360° video streaming sources for optimal area coverage considering video quality requirements. Optimal area coverage is necessary to maintain the integrity of digital twin services, while the allocated video quality directly impacts their performance and enables additional functionalities such as object detection tasks. Unlike previous works that focus solely on video streaming or machine learning clustering algorithms, our solution innovatively combines state-of-the-art clustering and adaptive streaming techniques (DASH), carefully selected and integrated to effectively address the challenge of enhancing situational awareness in smart city environments under contrained network bandwidth.

A. Architecture Description

Figure 2 illustrates the overall architecture of the proposed situational awareness framework at a given region in the presence of a limited shared uplink $\mathcal{B}$ for the entire region. In this work, we use 360° video streaming as it provides a spherical view that enhances flexibility while monitoring via HMD. Furthermore, it will reduce the complexity of the SA maximization problem when it comes to vehicle steering. As shown in Fig. 2, the red circles represent the spherical field of view of the 360° cameras, while the yellow vehicles represent all connected vehicles (e.g., autonomous vehicles or legacy vehicles equipped with DTBs) that are capable of sending 360° live streams, whereas the rest are conventional vehicles. The 360° live video streams are sent to a control center where surveillance operators can monitor the roads using either legacy monitors or HMDs to enable enhanced monitoring flexibility. In order to reduce the end-to-end (E2E) latency, the 360° video streams are primarily viewed at a local control center, if one is available, through multi-access edge computing (MEC) servers that are located in close proximity to the AoI [28]. In case there is no local control center available, live streams are sent to cloud servers, so a remote control center can perform surveillance through the cloud. In what comes next, we describe the main phases of the proposed framework to maximize the SA of the covered region, while shielding users' QoE from degradation. It is noteworthy that we interchangeably use the terms vehicles and DTBs since we only consider connected vehicles in this work.

B. Divide the AoI into Smaller Subareas

To efficiently measure the coverage of the entire region, we divide the AoI into smaller circular subareas, as shown in Fig. 1, where the radius equals the double of the lens focal length $\mathcal{L}$ of the camera. The clustering of the DTBs belonging to each subarea is then performed separately from the others, and the aggregated individual results provide coverage of the entire global area. The number of circular subareas can be configured and adapted to any size of the AoI according to its density and the number of obstacles. Intuitively, in urban areas where too many obstacles and buildings exist, a smaller value of $\mathcal{L}$ would be more appropriate and vice versa. It is worth noting that the clustering step works well for the entire region without any consideration for logical divisions. The latter only helps in assessing the coverage rate of the AoI and reduces the computational complexity of the clustering algorithm in large-scale cases.

C. Clustering

In order to achieve the best possible coverage of the AoI without compromising the video streaming QoE of the users at the control center, only a subset of DTBs should be allowed to live stream. Owing to the fact that adjacent and nearby vehicles would stream duplicate scenes, especially when a spherical view is delivered, it would be more efficient to select video sources whereby scenes do not overlap with each other. To do so, we cluster existing vehicles, based on their actual GPS coordinates, and group all geographically close-by vehicles together because they would stream similar scenes. It is worthwhile mentioning that clustering is performed within subareas, not for the entire region. In what follows, we describe the different unsupervised learning algorithms that are used to produce better clustering results.

As a first step, we use the Elbow [45] method to determine the optimal number of clusters (Line 8 in Algorithm 1). Next, we employ the Density-Based Spatial Clustering of Applications with Noise (DBSCAN-CORE) algorithm [46] ultimately to identify the core samples of high density (i.e., dense areas of vehicles) and the outlier vehicles, corresponding to the two subsets $\mathbb{A}$ and $\mathbb{B}$ in Algorithm 1, respectively. The next step (Line 10 in Algorithm 1) consists of clustering the vehicles in subset $\mathbb{A}$ (i.e., dense areas) using the one-step multi-view spectral clustering approach [47]. The clustering step is guided by the input from the Elbow method to produce ζ clusters. Two different views are identified:

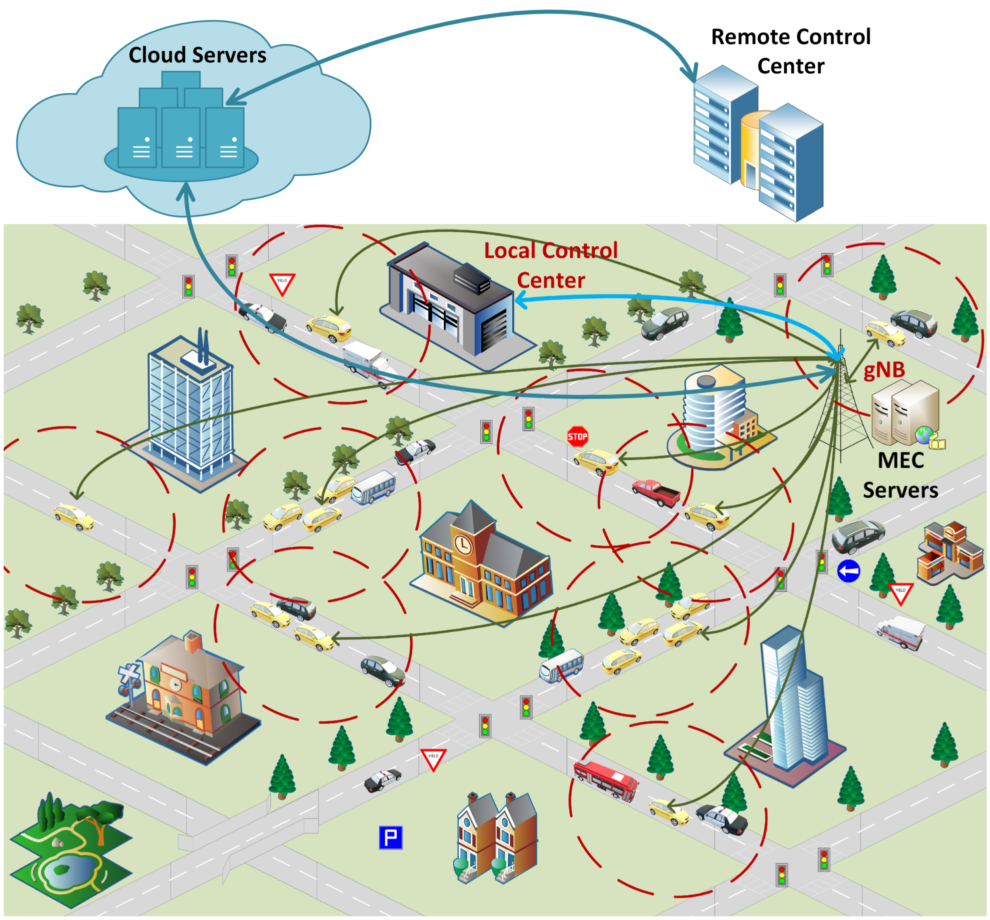

FIGURE 2. **The overall architecture of the situational awareness digital twin framework.**

- *View 1:* is the standard input for the spectral clustering algorithm. It consists of a 2-dimensional matrix, namely $\mathcal{P}^n$ (defined in Section III), that stores the computed distances among the DTBs themselves. This view considers only the distances between the vehicles and the clustering outcome is based on this distance-based matrix.
- *View 2:* a 1-dimensional matrix, namely $\mathcal{T}^n$, that contains the distances between the center λ_n of a subarea $\mathcal{R}_n$ and the DTBs $\mathcal{D}_m^n$ belonging to it. This view considers only the distances of the vehicles from a specific point of reference (e.g., the centre of the subarea) and the clustering outcome is based on this distance-based matrix.

A given DTB belongs to the subarea $\mathcal{R}_n$ if the distance between its GPS coordinates and the center λ_n of the subarea is strictly less to a certain threshold (r). The second view ensures that the clustering process considers only the DTBs belonging to the same subarea separately from the DTBs of other subareas. The inputs for the multi-view spectral clustering step are mainly computed based on the latest GPS coordinates of the different DTBs at a given snapshot of the system. The outliers data in subset $\mathbb{B}$ along with the outcome of the Multi-view spectral clustering, which is a set $\mathcal{C}^n$ of clusters per subarea $\mathcal{R}_n$, is then provided as input to the k-nearest neighbors (k-NN) algorithm (Line 11 in Algorithm 1) to update the set of clusters $\mathcal{C}^n$ by allocating the outliers vehicles to the nearest neighboring cluster $\mathcal{C}_h^n$ located in the same subarea $\mathcal{R}_n$. Each generated cluster $\mathcal{C}_h^n$ has a centroid, where the distance between the clusters' centroids satisfies a minimum threshold using the K-Means++ algorithm [48], which is implicitly applied when forming the clusters at step 10 in Algorithm 1.

D. Clusters & Vehicles Selection

Following the previous step, the different vehicles are assigned to the set of generated clusters that are sufficiently distant from each other, allowing for better coverage of the subareas, given the GPS coordinates of the vehicles. The next step consists of selecting a subset of vehicles that are allowed to stream live. To do so, we perform iterations over the different clusters within the same subarea, while prioritizing clusters whose centroids are closer to the center of the subarea, and we pick only one vehicle from each cluster per iteration to spatially maximize the coverage of the area, which would result in better SA, and fairly share the uplink bandwidth among vehicles in distinct clusters.

Rationally, the closer the selected vehicle is to the center of the subarea, the larger the covered surface of that subarea. Similarly, the closer the cluster is to the circle's perimeter, the less subarea is covered by vehicles belonging to that cluster. Figure 1 provides an illustration of the AoI division into smaller subareas as well as clustering. The big blue circle represents a subarea, while the smaller circles (shown in green, orange, and red) illustrate an example of clusters and their distances to the center λ_5 of the subarea $\mathcal{R}_5$. The grey circles represent the FoV of the 360° cameras with focal length $\mathcal{L}$. As to the vehicle selection at each iteration step, we start off by selecting the closest vehicle to the cluster's centroid in the first round. In subsequent rounds, a minimum distance threshold, equal to half the FoV (i.e., $\mathcal{L}/2$) of the 360° camera, should be verified whenever possible between previously selected vehicles and the one to be selected within the same cluster. By doing so, we avoid selecting multiple vehicles in close proximity to one another, which basically stream similar scenes, while potentially leaving other spots uncovered due to bandwidth scarcity. It is worth mentioning that the selected vehicles are the ones allowed to stream live, also called active DTBs, while the rest remain inactive (i.e., do not livestream), but can still record and store the video streams locally for later upload.

E. Bandwidth Allocation & Video Quality Selection

Once vehicles are selected, the last step consists of deciding which video qualities the selected DTBs are allowed to livestream so that the QoE is not compromised at either local or remote control centers, while taking into account the uplink capacity $\mathcal{B}$. Initially, $\mathcal{B}$ is evenly distributed among the subareas $\mathcal{R}_n \in \mathcal{R}, \forall n \in [1, \mathcal{N}]$ to ensure the bandwidth is fairly shared across the different subareas. The latter are then considered in ascending order, in terms of number of vehicles located in each subarea $\mathcal{R}_n$, to proceed with the vehicle selection process starting from the less crowded subareas. By doing so, the more crowded subareas would benefit from the leftover bandwidth budget that was initially allocated to non-crowded subareas.

As explained in Section D, the generated clusters $\mathcal{C}^n$ at each subarea $\mathcal{R}_n$ are sorted in ascending order according to their distances to the center of the subarea to prioritize the clusters which are in the middle of the subareas. The process of video quality allocation is then performed by iterating over the different clusters in the predefined order while selecting a single vehicle from each cluster at a given iteration to maximize spatial area coverage. In the first round, the default video quality assigned to the selected vehicle is the highest one. In the following rounds, and based on the number of selected vehicles, we progressively reduce the video quality in order to accommodate as many vehicles as possible for wider spatial coverage and better situational awareness. The aim is to offer at least one high-resolution stream per cluster and possibly several distant vehicles covering other angles of the cluster at lower resolutions, while the essential objective is to avoid video stalls, which are the most serious cause of QoE degradation.

Algorithm 1 Situational Awareness Maximization (SAM)

1: **procedure** MSA($\mathcal{R}$, $\mathcal{D}$, $\mathcal{G}, \mathcal{B}$)
2: $\widetilde{\mathcal{B}} \leftarrow \mathcal{B}$ ▷ $\widetilde{\mathcal{B}}$ is the left over bandwidth
3: $\mathcal{R} \leftarrow sort(\mathcal{R}, ASC)$
4: **for** $\mathcal{R}_n \in \mathcal{R}$ **do**
5: $\mathcal{S}^n \leftarrow \emptyset$; $\mathcal{B}^n \leftarrow \widetilde{\mathcal{B}}/n$; $\widetilde{\mathcal{B}} \leftarrow \widetilde{\mathcal{B}} - \mathcal{B}^n$; $iter \leftarrow 0$
6: $\mathcal{P}^n \leftarrow get\_dtbs\_distances(\mathcal{D}^n, \mathcal{G}^n)$
7: $\mathcal{T}^n \leftarrow get\_dtbs\_proximity(\lambda_n, \mathcal{D}^n, \mathcal{G}^n)$
8: $\zeta \leftarrow get\_elbow\_clusters(\mathcal{D}^n, \mathcal{G}^n)$
9: $\mathbb{A}, \mathbb{B} \leftarrow cluster\_dbscan\_core(\mathcal{D}^n, \mathcal{G}^n)$
10: $\mathcal{C}^n \leftarrow cluster\_SC(\zeta, \mathbb{A}, \mathcal{P}^n, \mathcal{T}^n, ASC)$
11: $\mathcal{C}^n \leftarrow kNN\_allocate\_outliers(\mathcal{C}^n, \mathbb{B}, \mathcal{D}^n, \mathcal{G}^n)$
12: **while** $\mathcal{B}^n \geq \mathcal{Q}_1$ AND $C_h^n \neq \varnothing$ **do** ▷ Vehicle & video quality selection
13: $iter$++;
14: **for** $h \leftarrow 1$ to $\mathcal{H}$ **do**
15: **if** $\mathcal{B}^n \leq \mathcal{Q}_1$ **then**
16: $break$
17: **end if**
18: $\eta \leftarrow select\_dtb(\mathcal{C}_h^n)$
19: $\hat{\eta} \leftarrow select\_video\_quality(\eta, \mathcal{B}^n, iter)$
20: $\mathcal{B}^n \leftarrow \mathcal{B}^n - \hat{\eta}$
21: $\mathcal{C}_h^n \leftarrow \mathcal{C}_h^n \setminus \eta$
22: $\mathcal{S}^n \leftarrow \mathcal{S}^n \cup \eta$
23: **end for**
24: **end while**
25: $\widetilde{\mathcal{B}} \leftarrow \widetilde{\mathcal{B}} + \mathcal{B}^n$
26: **end for**
27: **end procedure**

We keep iterating over the generated clusters in a given subarea until there is no longer enough bandwidth to be allocated to the vehicles (Line 12 in Algorithm 1), or all the vehicles in that subarea are satisfied in terms of bandwidth coverage. The complete pseudo-code of the situational awareness maximization framework is provided in Algorithm 1. It should be noted that Algorithm 1 is executed at discrete timesteps to account for vehicles' mobility. The higher the vehicles' velocity (e.g., on an expressway), the shorter the timestep interval for algorithm execution, and vice versa. A more sophisticated solution which takes into account, during the DTB selection, other mobility-related parameters such as the vehicle's velocity and its planned trajectory, can be found in [38].

V. Performance Evaluation

A. Evaluation Setup

To demonstrate the performance of our proposed framework, we conducted extensive simulations under several scenarios, namely low, middle, large, and very large-scales, corresponding to cases of 20, 100, 200, and 400, for evaluating the effect of subareas and the coverage performance while considering the region division into subareas of different number of vehicles. In addition, we considered also scenarios of 600, 900, and 1200 vehicles for testing the scalability of Algorithm 1. For the GPS coordinates we used a real-world dataset, namely the Road Vehicle Localization dataset [49], while our ground truth dataset is defined, using NumPy, Pandas and Shapely Geometry libraries, by choosing the coordinates set that gives the maximum area coverage. The area is defined by the largest x and y values among all the vehicle GPS coordinates. The algorithm implements single-view and multi-view clustering with K-Means++ to obtain the final clustering results. The clustering parameters are defined in Table 1.

TABLE 1. Algorithm 1 Set-up Parameters

Single-View & Multi-View Clustering	
Clusters	8
Random state	None
Info view	None
Max iterations	10
No. of initializations	10
Affinity	Nearest Neighbors
Gamma	None
Neighbors	10
DBSCAN Algorithm	
Eps	0.5
Minimum Samples	5
Metric	Euclidean
k-Nearest Neighbor algorithm	
Metric	Euclidean
Shrink Threshold	None

The evaluation process was conducted on a DELL Inspiron 3847 with a processor Intel Dual Core i3 1600MHz, 8GB RAM. As for the software configuration, we ran our experiments on a Windows 10 OS with Python 3.7.13 using the NumPy 1.23.0 library. In our simulations, we considered a large area divided into relatively small circular subareas of fixed radius r = 80 meters. The set of video qualities is defined as $\mathcal{Q} = \{10, 8, 6\}Mbps$ with $10Mbps$ representing the highest video quality and $6Mbps$ the lowest acceptable case. To account for critical situations where the uplink capacity is extremely disproportional to the number of vehicles in a certain region, we set the uplink capacity as $\mathcal{B} = 1500Mbps$ to evaluate the coverage performance when a region is divided into several subareas. For the larger scenarios of 600, 900, and 1200 vehicles, we further tested them as distinct whole regions (i.e., without divide them into subareas) with $\mathcal{B} = 3.000Mbps$ for each one, without subdivision into smaller subareas. It is noteworthy that we used a radius of r meters for the circular adjacent subareas to generate the results presented in Table 5. Alternatively, a radius of $10r$ meters was used when considering the entire region during the scalability evaluation to obtain the results shown in Table 8.

Furthermore, in terms of coverage performance, we included in Tables 6 to 9 the comparison results with the algorithms proposed by [50] and [51]. In [50], the authors presented 3 different scenarios comparing their algorithm's performance against others in the literature. Their second scenario was about a sensor deployment case for the connectivity improvement among the robots that move in a region of interest. In correspondence with our setup, the cameras are addressed in the same way as the sensors mounted on the robots to satisfy the optimal area coverage requirements. In [51] the authors proposed an improved Flower Pollination Algorithm (FPA) to guarantee the coverage and connectivity requirements in a specific region. The algorithm was properly configured in accordance with our coverage requirements and compared to our proposed Algorithm 1. More particularly, we assumed that the target points of their experiment were the centers of our subareas, while the sensor nodes were addressed as vehicles in our scenarios. The optimal locations of the nodes determined by the algorithm are then compared with the closest vehicles in our scenario to make their selection for video coverage. To this end, we select the vehicles that are closest to the node coordinates given by the algorithm.

Given that the main contribution of this work is to study the SA of roads' infrastructure and how to maximize it in a given area by leveraging 360° live streams, it is noteworthy to add that our focus in the current section is on the evaluation and comparison of the proposed framework against state-of-the-art algorithms in terms of covered surface rather than the live video streams performance. Furthermore, it is worth mentioning that transmitted live streams are mainly consumed by humans (e.g., national security agents) for monitoring purposes and the targeted use cases by this work are not subsecond-latency critical, which tolerate for few seconds of latency in DASH-based streaming, fluctuating between $1s$ to $3s$.

B. Performance Evaluation Metrics

For the evaluation process, we adopted two Key Performance Indicators (KPIs), namely the overall percentage of the covered region and the performance of the algorithms in terms of Normalized Mutual Information (NMI) [52]. To compute the coverage percentage of the subareas, and eventually the overall regional coverage, we deduct the subarea surfaces covered by the 360° cameras, after considering surface overlaps, from the accumulated subarea surfaces.

Thus, let S_m be the virtual circular surfaces of the cameras $\mathcal{D}_m^n$ and S_n the circular surfaces of the subareas $\mathcal{R}_n$. The surface coverage computation is provided by equations (7) and (8), based on the notions and formulations of [53] and [54]. The coverage percentage of each subarea is given in Table. 6. The overall percentage of the covered region, S_P, is calculated by:

$$S_p = \frac{S_A}{\sum_{n=1}^{N} S_n} \times 100, \tag{7}$$

where:

$$S_A = \sum_{m=1}^{M} S_m - \sum_{\xi=1}^{M} \sum_{\eta=\xi+1}^{M} \Theta_{\xi\eta}, \tag{8}$$

where S_m denotes the virtual circular surfaces of the active DTBs and $\Theta_{\xi\eta}$ denotes the overlapping area between S_ξ and S_η.

C. Performance Results

The evaluation process will consider the widely used K-Means and spectral clustering approaches in both single- and multi-view schemes. Our scenarios are based only on these approaches as they present better clustering performance against other clustering approaches, see Table 10, while they present much better computation times for real-time scenarios. The performance evaluation of both schemes is assessed through different metrics, namely cluster formation, DTB selection, video quality attribution, and per-subarea SA percentage. In addition, performance benchmarking between multi-view K-Means and spectral clustering is presented referring to the aforementioned vehicular scalability cases.

1) Single-view K-Means and Spectral Clustering

In this section, we evaluate the single-view implementation of both the popular K-Means, in a similar concept as in a network traffic application in [55], and spectral clustering algorithms. In this scheme, the two views are combined into one single view to be used in the clustering process by concatenating the input adjacency matrices. We assume a region of 720 vehicles is divided into four equal and adjacent circular subareas with different vehicular densities, defined at 20, 100, 200, and 400 vehicles respectively. The clustering results for Subarea 1 are given in Table. 2, where the vehicle IDs in bold denote the detected outlier vehicles before their reallocation to their actual clusters based on line 11 of Algorithm 1. The clustering outcome of the remaining three subareas is not presented here due to the lengthy results. They are, however, included in Table. 5 which shows bandwidth allocation and video quality selection. In Figure 3 and due to visual presentation clarity, we can see an indicative single-view spectral clustering outcome of an area with 100 vehicle coordinates divided into four subareas.

2) Multi-view K-Means and Spectral Clustering Schemes

Contrary to the previous scheme, the multi-view clustering consists of simultaneously considering the two separate views described in Section IV. In our computational experiments, we use a well-known open-source software package developed in Python for multi-view clustering approaches, namely the *mvlearn* [56]. Classical methods of inference and analysis are often poorly suited when considering multiple views of the same data sample. With *mvlearn*, we can implement leading multi-view machine learning methods. The library has been demonstrated through specific multi-view example cases highlighting the performance superiority

TABLE 2. Single-view Clustering for the subarea 1 (N = 20 vehicles)

Subareas	Cluster ID	Spectral Clustering	K-Means
Subarea 1	1	12,14,15,7	2,10
	2	1,17,20,9	13,14,16,19
	3	4,5	8,18
	4	16	1,5,11
	5	13,19	4,6,7,9
	6	6,18,11	3,12
	7	2,3	**17**,20
	8	8,10	15

of multi-view algorithms compared to their respective single-view ones, as presented in the library repository[1]. Similarly to the single-view, the generated multi-view oriented clusters per algorithm for Subarea 1 are presented in Table. 3. In Figure 4 we graphically depict the clustering outcome of the multi-view spectral algorithm for Subarea 1, which has 20 out of 720 total vehicles in the whole region of our tested scenario.

TABLE 3. Multi-view Clustering for Subarea 1 (N = 20 vehicles)

Subareas	Cluster ID	Spectral Clustering	K-Means
Subarea 1	1	7,9,10,11,12	12,14,15,17,20
	2	2,5,20	4,6,7,9,**10**
	3	13,14,15	2
	4	4,6,18,19	3
	5	3	1,16
	6	1	8,18
	7	8	13,19
	8	16,**17**	15,11

3) Video Quality Selection

For scalability demonstration reasons, we assumed that the region consists of four subareas with completely different vehicular densities, i.e., $20, 100, 200,$ and 400 vehicles, respectively. Considering that the above region has an uplink capacity of $1500 Mbps$, and the vehicles are included in the predefined four circular and adjacent subareas in a 2×2 layout, we proceed with our suggested framework for bandwidth allocation per subarea and per DTB. Any bandwidth left in each subarea is added to the next one following an ascending order based on the number of vehicles it includes. The bandwidth allocation and the video quality selection results for Subarea 1 are presented in Table. 4, including only the IDs of the DTBs that are selected for live streaming, along with their attributed video qualities. In Table. 5, we present the summary of the results from the remaining subareas, in terms of the total number of clusters and selected vehicles as well as the number of selected vehicles per video quality.

[1] https://mvlearn.github.io/

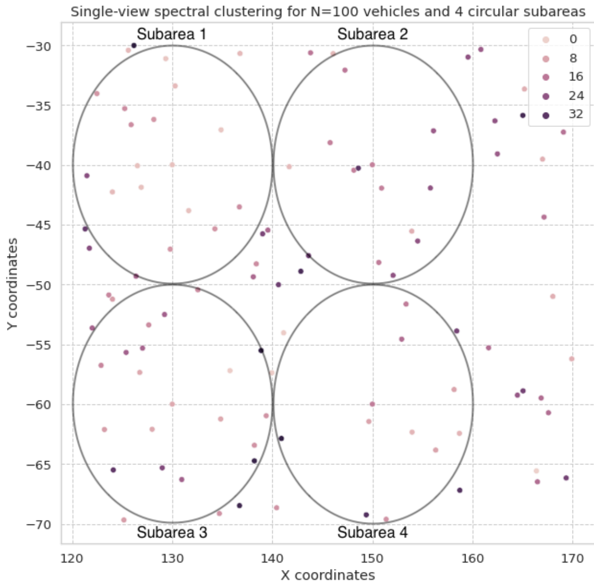

FIGURE 3. A single-view clustering for n=100 vehicles with 4 subareas.

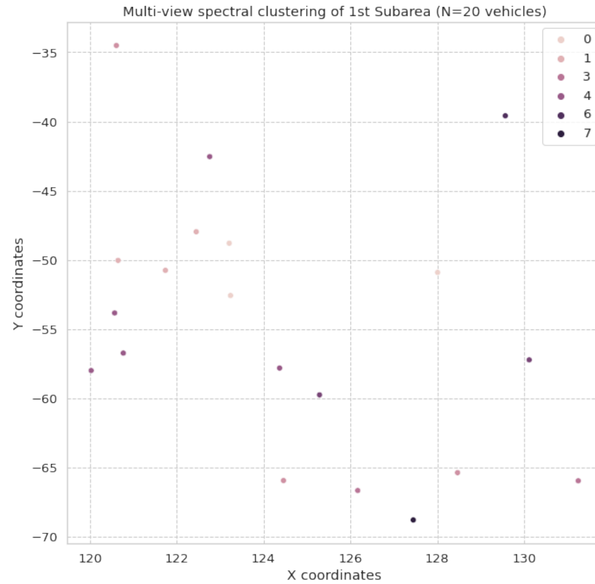

FIGURE 4. Multi-view Spectral Clustering in the 1st Subarea (n=20 vehicles).

It is important to note that the average video quality is consistently optimized on the basis of the selected vehicle set at each timestep. To achieve this, our algorithm guarantees that at least one vehicle per cluster streams in the highest possible video quality, given the available bandwidth during each round.

TABLE 4. Bandwidth Allocation for n=720 vehicles and B=1500 Mbps (Multi-view Spectral Clustering).

Subarea 1: $n = 20$ vehicles					
Residual Bandwidth Iterations (Mbps)		Cluster ID	Selected Vehicles	Assigned Video Qualities per Iteration (Mbps)	
1	2			1	2
375	297	5	16,10	10	8
365	289	2	6,4	10	8
355	289	7	14	10	-
345	289	4	8	10	-
335	281	6	18,19	10	8
325	281	1	1	10	-
315	281	3	13	10	-
305	**281**	8	15	10	-

Note: Initial allocated bandwidth and the final residual bandwidth are in **bold**.

4) Situational Awareness

Table. 6 presents the area coverage results, expressed in percentage, for the four virtual subareas, of different densities, from the whole region containing 720 vehicles. Among the spectral and K-means clustering algorithms, the results show the superiority of the multi-view clustering approaches over their single-view counterparts. However, it is observed that in small-scale scenarios, single spectral view clustering seems to have better coverage performance. It happens because small datasets often have simpler relationships that can be captured effectively by focusing on a single feature. Adding multiple views with variable FoV introduces unnecessary complexity, which might not improve clustering but instead overfit the limited data. Also, it is apparent that the multi-view spectral clustering algorithm outperforms the multi-view K-means algorithm.

In addition, we conducted similar computations for very large-scale scenarios starting from 600 vehicles up to 1200 vehicles and measured the SA percentage for the whole region, without dividing it into subareas. In these simulations, we used a radius of $r = 2.000$ meters for the global area and an uplink capacity of $B = 3.000 Mbps$. The large-scale results given in Table. 8 corroborate the observations derived from the analysis of multiple subareas at smaller scale. Furthermore, we expanded our area coverage analysis in Tables 7 and 9 by considering a realistic scenario where vehicles are equipped with cameras featuring varying FoVs, namely $r/4$, $r/2$ and r with equal distribution into the vehicle populations, thereby demonstrating the robustness and consistency of our solution across diverse conditions.

Based on the results in Tables. 6 - 9, we clearly see that our Algorithm 1 with the multiview spectral clustering approach outperforms the multiview K-Means clustering algorithm, which highlights its effectiveness in vehicle selection towards a near-optimal solution for our joint video streaming and SA problem. It also outperforms the proposed and properly con-

TABLE 5. Bandwidth allocation for n=720 vehicles and B=1500 Mbps (Rest of subareas)

Subarea IDs	Allocated Bandwidth (Mbps)	Consumed Bandwidth (Mbps)	Num. of Clusters	Total Num. of Vehicles	Selected Vehicles	Total Vehicles per Video Quality		
						High	Medium	Low
2	656	530	40	100	57	40	14	3
3	501	500	75	200	50	50	0	0
4	376	376	125	400	38	37	0	1

TABLE 6. Effect of Subareas: Approximate Area Coverage Results with N=720 vehicles and 4 subareas (S_p) - Fixed Fov

$L = r/2$	Single-view		Multi-view			
Subareas	Spectral	K-Means	Spectral	K-Means	[50]	[51]
1	50.3%	47.71%	52.33%	38.29%	49.28%	44.56%
2	67.85%	62.17%	71.23%	66.19%	63.4%	70.05%
3	61.13%	51.78%	67.24%	52.41%	64.11%	62.13%
4	58.92%	53.46%	69.7%	54.45%	66.1%	61.55%
TOTAL	59.55%	53.78%	**65.12**%	52.83%	60.72%	59.57%

TABLE 7. Effect of Subareas: Approximate Area Coverage Results with N=720 vehicles and 4 subareas (S_p) - Variable Fov

$\mathcal{L} \in (r/4, r)$	Single-view		Multi-view			
Subareas	Spectral	K-Means	Spectral	K-Means	[50]	[51]
1	44.2%	41.53%	42.23%	31.17%	38.88%	36.62%
2	56.15%	50.76%	58.13%	57.9%	53.1%	62.15%
3	48.43%	37.18%	49.41%	40.11%	52.12%	49.15%
4	39.92%	41.62%	61.19%	42.51%	51.1%	44.18%
TOTAL	47.17%	42.77%	**52.74**%	42.92%	48.8%	48.02%

TABLE 8. Effect of Scalability: Approximate Region Coverage Results with B=3.000 Mbps and radius r=2.000 meters (S_p) Fixed FoV

$L = r/2$	Single-view		Multi-view			
Vehicles	Spectral	K-Means	Spectral	K-Means	[50]	[51]
600	59.66%	52.11%	**67.88%**	62.31%	63.29%	65.32%
900	61.87%	64.12%	**68.47%**	64.73%	65.25%	67.49%
1200	59.67%	56.17%	**65.18%**	61.34%	61.56%	62.15%

figured state-of-the-art algorithms in [50] and the Improved FPA algorithm in [51].

5) Comparison of clustering algorithms performance

Table. 10 shows the NMI comparison results of the K-Means (1) and spectral clustering (2) algorithms in both single- and multi-view schemes as well as the results of two selected popular single-view algorithms from the literature, namely the Gaussian Mixture Model [57] (3) and the Fuzzy C-Means [58] (4). For multi-view spectral and K-Means,

TABLE 9. Effect of Scalability: Approximate Region Coverage Results with B=3.000 Mbps and radius r=2.000 meters (S_p) Variable FoV

$\mathcal{L} \in (r/4, r)$	Single-view		Multi-view			
Vehicles	Spectral	K-Means	Spectral	K-Means	[50]	[51]
600	50.3%	42.13%	**52.62%**	51.11%	50.9%	48.12%
900	49.71%	51.2%	**53.56%**	51.33%	51.96%	47.39%
1200	57.21%	54.1%	**63.88%**	60.47%	58.62%	60.3%

NMI scores demonstrate better performance against all the rest of the compared algorithms. Our scenarios were based only on the multi-view clustering approaches as they present better clustering performance against the other 4 cases, while the Gaussian Mixture Model and Fuzzy C-Means are computationally expensive because of the larger computation times for real-time scenarios and the need to estimate the covariance matrices [59], [60]. The comparison is performed at different scale scenarios using the NMI bounded scores.

TABLE 10. Comparison of the clustering algorithms' performance per scenario scale (nmi score).

	Single-view				Multi-view		
Vehicles	1	2	3	4	1	2	Num. of Clusters
20	**1.0**	0.87	0.961	0.923	**1.0**	0.94	8
100	0.863	0.734	0.81	0.785	**0.882**	0.767	40
200	0.691	0.727	0.703	0.722	0.703	**0.769**	75
400	0.668	0.711	0.699	0.708	0.665	**0.732**	125
600	0.657	0.696	0.668	0.679	0.613	**0.708**	200
900	0.583	0.621	0.607	0.619	0.520	**0.698**	350
1200	0.614	0.633	0.638	0.657	0.610	**0.685**	500

The NMI metric demonstrates that in low- and middle-scale scenarios, the multi-view K-Means clustering algorithm performs better, while in large-scale scenarios the multi-view spectral clustering algorithm gives slightly better results, with its performance being comparably the best as scale grows. Given that our proposed framework will be used in real-life scenarios in high-density areas, multi-view clustering is the prevalent solution. Besides, it should be pointed out that if the region is divided into many small equal subareas, the multi-view K-Means should be the designer's preferred choice, while in larger vehicle populations with fewer subareas the multi-view spectral clustering presents better results.

6) Complexity Analysis

From a complexity analysis point of view, we examined both single-view and multi-view spectral clustering in combination with the rest of the computations made in Algorithm 1. To this end, DBSCAN presents a time complexity of O($nlogn$) with n defined as the number of vehicles (data points), spectral clustering presents a time complexity of O(pn^3) with p defined as the number of views, kNN presents a time complexity of O(pn) and Euclidean distance presents a time complexity of O(n).

TABLE 11. Simulation Execution Time in Seconds (s).

Vehicles	Time	Clusters	Subareas
20	3.1s / 2.4s / 0.6s	8	0 / 4 / 8
100	5.2s / 3.4s / 1.2s	40	0 / 6 / 12
200	6.6s / 4.2s / 2.3s	75	0 / 6 / 12
400	7.3s / 5.1s / 2.8s	125	0 / 8 / 12
600	9.8s / 6.7s / 4.8s	200	0 / 8 / 16
1200	11.3s / 8.7s / 4.4s	500	0 / 10 / 16

In Table 11 we present the algorithm execution time results for various numbers of subareas and vehicles in order to underline the fact that the more the sub-areas, the better the execution time of the algorithm. Since we refer to temporal snapshots of bandwidth allocation, the temporal execution is not suitable for a large number of vehicles. This drawback can be removed by increasing the number of sub-areas for presenting better computational efficiency. Another solution should be the use of hardware acceleration techniques and GPU-related deployments, which we will include and evaluate in our future research.

VI. Conclusions

In this paper, we successfully applied a machine learning approach for maximizing the situational awareness of a given region leveraging moving DTBs and using 360° live streams. The ultimate goal consists of enhancing the overall SA while preserving the QoE at the consumer side (e.g., remote surveillance agents) under uplink capacity constraint. This is achieved by allowing only a subset of vehicles, notably in dense areas, to live stream when the bandwidth becomes critical. These vehicles are optimally selected, so that the coverage of the AoI is maximized. Towards this end, we leveraged the K-Means and spectral clustering approaches at both single- and multi-view schemes to obtain optimal clustering of DTBs based on the distances between themselves as well as the distance to the subareas centers. We also ensured that the K-Means++ module is included in the algorithms' configuration for achieving the best possible cluster centroid distribution. Then, we proposed and implemented vehicle selection and video quality attribution logic to ensure maximized situational awareness based on the DTBs' instant GPS coordinates. We have conducted extensive simulations and demonstrated the effectiveness of the proposed solution in terms of situational awareness meeting the requirements for optimal coverage and the performance of the algorithms validated with the NMI score. We also compared the results of the K-Means and spectral clustering approaches in different scenarios at low, medium, and large-scales and with two state-of-the-art coverage algorithms from the recent literature. Our selection of K-Means++, Spectral Clustering, and DBSCAN was driven by their proven efficiency in multi-view clustering scenarios. To justify our selection, we compared our approach against alternative novel clustering methods, demonstrating why our choice remains competitive for real-time smart city applications.

In terms of clustering performance, it was demonstrated that the multi-view algorithms outperform their single-view counterparts. Subsequently, the proposed SAM algorithm gives a discrete quantized bandwidth allocation in different video qualities based on the resulting centroids of the clusters from the multi-view clustering part of the algorithm. This outcome satisfies the joint objective of the approximate optimal vehicle selection and bandwidth allocation confirmed by both the results of the subarea's coverage efficiency and scalability. A crucial finding of this work is that multi-view spectral clustering performs better in environments with a large number of vehicles (i.e. dense areas), while the multi-view K-Means algorithm performs better in environments with a large number of sub-areas.

In the future, we plan to implement more sophisticated machine and deep learning approaches for addressing the problem of maximizing situational awareness and optimal bandwidth allocation among vehicles in smart cities. From a federated learning aspect and the digital twin system's perspective, we plan to examine, test, and integrate incentive mechanisms for the choice of cameras that will take part in the learning process without violating the joint optimal coverage requirements. Predicting the vehicle's position, based on its velocity and trajectory, is another crucial feature for vehicle selection that needs to be considered in our next research endeavor.

Acknowledgment

This work was partially conducted at ICTFICIAL Oy. It is partially supported by the European Union's HE research and innovation program HORIZON-JUSNS-2023 under the 6G-Path project (Grant No. 101139172) and HORIZON-JU-SNS-2022 under the 6G-SANDBOX project (Grant No. 101096328). The paper reflects only the authors' views, and the European Commission bears no responsibility for any utilization of the information contained herein.

REFERENCES

[1] A. Rayes and S. Salam, *Internet of Things From Hype to Reality - The Road to Digitization*. Springer, Cham, 11 2016. [Online].

Available: https://link.springer.com/book/10.1007/978-3-319-44860-2

[2] H. Wang, Z. Wang, D. Chen, Q. Liu, H. Ke, and K. K. Han, "Metamobility: Connecting future mobility with the metaverse," *IEEE Vehicular Technology Magazine*, 2023.

[3] Y. Gong, Y. Wei, Z. Feng, F. R. Yu, and Y. Zhang, "Resource allocation for integrated sensing and communication in digital twin enabled internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 4510–4524, 2023.

[4] O. El Marai, T. Taleb, and J. Song, "Roads infrastructure digital twin: A step toward smarter cities realization," *IEEE Network*, vol. 35, no. 2, pp. 136–143, 2021.

[5] T. Taleb, A. Ksentini, H. Hellaoui, and O. Bekkouche, "On supporting uav based services in 5g and beyond mobile systems," *IEEE Network*, vol. 35, no. 4, pp. 220–227, 2021.

[6] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.

[7] L. Zhang and J. Chakareski, "Uav-assisted edge computing and streaming for wireless virtual reality: Analysis, algorithm design, and performance guarantees," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3267–3275, 2022.

[8] T. M. P. E. Group, "Mpeg-dash: Dynamic adaptive streaming over http, iso/iec 23009," https://mpeg.chiariglione.org/standards/mpeg-dash, 2023, accessed: 2022-09-30.

[9] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, "A survey on 360° video streaming: Acquisition, transmission, and display," *ACM Comput. Surv.*, vol. 52, no. 4, aug 2019. [Online]. Available: https://doi.org/10.1145/3329119

[10] W. Liu, H. Zhang, H. Ding, and D. Yuan, "Delay and energy minimization for adaptive video streaming: A joint edge caching, computing and power allocation approach," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9602–9612, 2022.

[11] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[12] M. Hu, L. Wang, B. Tan, and S. Jin, "Two-tier 360-degree video delivery control in multiuser immersive communications systems," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 4119–4123, 2022.

[13] H. Masuda, O. E. Marai, M. Tsukada, T. Taleb, and H. Esaki, "Feature-based vehicle identification framework for optimization of collective perception messages in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2120–2129, 2023.

[14] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8810–8824, 2022.

[15] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE access*, vol. 8, pp. 108 952–108 971, 2020.

[16] J. Zheng, T. H. Luan, Y. Hui, Z. Yin, N. Cheng, L. Gao, and L. X. Cai, "Digital twin empowered heterogeneous network selection in vehicular networks with knowledge transfer," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12 154–12 168, 2022.

[17] F. Dembski, U. Wössner, M. Letzgus, M. Ruddat, and C. Yamu, "Urban digital twins for smart cities and citizens: The case study of herrenberg, germany," *Sustainability*, vol. 12, no. 6, p. 2307, 2020.

[18] R. Teixeira, J. Puccinelli, B. de Vargas Guterres, M. R. Pias, V. M. Oliveira, S. S. d. C. Botelho, L. Poersch, N. D. Filho, A. Janati, and M. Paris, "Planetary digital twin: A case study in aquaculture," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 191–197. [Online]. Available: https://doi-org.libproxy.aalto.fi/10.1145/3477314.3508384

[19] F. X. Ming, R. A. A. Habeeb, F. H. B. Md Nasaruddin, and A. B. Gani, "Real-time carbon dioxide monitoring based on iot & cloud technologies," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, ser. ICSCA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 517–521. [Online]. Available: https://doi-org.libproxy.aalto.fi/10.1145/3316615.3316622

[20] S. Van Den Berghe, "A processing architecture for real-time predictive smart city digital twins," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 2867–2874.

[21] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman, and A. Jamalipour, "An end-to-end (e2e) network slicing framework for 5g vehicular ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7103–7112, 2021.

[22] A. Awad Abdellatif, A. Abo-Eleneen, A. Mohamed, A. Erbad, N. V. Navkar, and M. Guizani, "Intelligent-slicing: An ai-assisted network slicing framework for 5g-and-beyond networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1024–1039, 2023.

[23] T. Taleb, Z. Nadir, H. Flinck, and J. Song, "Extremely interactive and low-latency services in 5g and beyond mobile systems," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 114–119, 2021.

[24] T. Taleb, A. Boudi, L. Rosa, L. Cordeiro, T. Theodoropoulos, K. Tserpes, P. Dazzi, A. Protopsaltis, and R. Li, "Towards supporting xr services: Architecture and enablers," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[25] S. Hu, W. Ni, X. Wang, and A. Jamalipour, "Disguised tailing and video surveillance with solar-powered fixed-wing unmanned aerial vehicle," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5507–5518, 2022.

[26] K. Nihei, H. Itsumi, Y. Shinohara, T. Araki, and T. Iwai, "Multiple cars remote monitoring system using ai-based video streaming and alert," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*. IEEE, 2023, pp. 1–7.

[27] J. Gao, Z. Hu, K. Bian, X. Mao, and L. Song, "Aq360: Uav-aided air quality monitoring by 360-degree aerial panoramic images in urban areas," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 428–442, 2021.

[28] O. El Marai, T. Taleb, and J. Song, "Ar-based remote command and control service: Self-driving vehicles use case," *IEEE Network*, pp. 1–8, 2022.

[29] Z. Nadir, T. Taleb, H. Flinck, O. Bouachir, and M. Bagaa, "Immersive services over 5g and beyond mobile systems," *IEEE Network*, vol. 35, no. 6, pp. 299–306, 2021.

[30] T. Taleb, N. Sehad, Z. Nadir, and J. Song, "Vr-based immersive service management in b5g mobile systems: A uav command and control use case," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[31] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "Mpeg dash srd: Spatial relationship description," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi-org.libproxy.aalto.fi/10.1145/2910017.2910606

[32] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi-org.libproxy.aalto.fi/10.1145/2910017.2910641

[33] L. D'Acunto, J. van den Berg, E. Thomas, and O. Niamut, "Using mpeg dash srd for zoomable and navigable video," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi-org.libproxy.aalto.fi/10.1145/2910017.2910634

[34] Y. Guo, R. W. Liu, Y. Lu, J. Nie, L. Lyu, Z. Xiong, J. Kang, H. Yu, and D. Niyato, "Haze visibility enhancement for promoting traffic situational awareness in vision-enabled intelligent transportation," *IEEE Transactions on Vehicular Technology*, 2023.

[35] M. Jiang, "Improving situational awareness with collective artificial intelligence over knowledge graphs," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, vol. 11413. SPIE, 2020, pp. 144–154.

[36] E. Blasch, Z. Liu, Y. Zheng, U. Majumder, A. Aved, and P. Zulch, "Multisource deep learning for situation awareness," in *Automatic Target Recognition XXIX*, vol. 10988. SPIE, 2019, pp. 131–142.

[37] A. Nesen and B. Bhargava, "Towards situational awareness with multimodal streaming data fusion: serverless computing approach," in *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments*, 2021, pp. 1–6.

[38] S. Messinis, O. E. Marai, N. E. Protonotarios, T. Taleb, and N. Doulamis, "Roads digital twin: Predictive situational awareness using 360° video streaming and graph neural networks," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2024.

[39] S. Bettoumi, C. Jlassi, and N. Arous, "Collaborative multi-view k-means clustering," *Soft Computing*, vol. 23, no. 3, pp. 937–945, 2019.

[40] H. Lu, Q. Gao, X. Zhang, and W. Xia, "A multi-view clustering framework via integrating k-means and graph-cut," *Neurocomputing*, 2022.

[41] S. Zhou, X. Liu, J. Liu, X. Guo, Y. Zhao, E. Zhu, Y. Zhai, J. Yin, and W. Gao, "Multi-view spectral clustering with optimal neighborhood laplacian matrix," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6965–6972.

[42] Y. Pan, C.-Q. Huang, and D. Wang, "Multiview spectral clustering via robust subspace segmentation," *IEEE Transactions on Cybernetics*, 2020.

[43] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 46–58, 2018.

[44] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[45] C. Shi, B. Wei, S. Wei, W. Wang, H. Liu, and J. Liu, "A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–16, 2021.

[46] S. Miyahara and S. Miyamoto, "A family of algorithms using spectral clustering and dbscan," in *2014 IEEE International Conference on Granular Computing (GrC)*. IEEE, 2014, pp. 196–200.

[47] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 2022–2034, 2019.

[48] S. Lattanzi and C. Sohler, "A better k-means++ algorithm via local search," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3662–3671.

[49] S. B. Cruz and A. Aguiar, "Road vehicle localization dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/92ya-v240

[50] P. Tirandazi, A. Rahiminasab, and M. Ebadi, "An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 1–23, 01 2022.

[51] W. Jiao, R. Tang, and Y. Xu, "A coverage optimization algorithm for the wireless sensor network with random deployment by using an improved flower pollination algorithm," *Forests*, vol. 13, no. 10, 2022. [Online]. Available: https://www.mdpi.com/1999-4907/13/10/1690

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[53] F. Librino, M. Levorato, and M. Zorzi, "An algorithmic solution for computing circle intersection areas and its applications to wireless communications," *Wireless Communications and Mobile Computing*, vol. 14, no. 18, pp. 1672–1690, 2014.

[54] M. P. Fewell, "Area of common overlap of three circles," Defence Science and Technology Organisation Edinburgh (Australia) Maritime Operations Div, Tech. Rep. ADA463920, Oct. 2006. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA463920.pdf

[55] H. Purnawansyah and H. Haviluddin, "K-means clustering implementation in network traffic activities," in *International Conference on Computational Intelligence and Cybernetics*, vol. 10, 2016.

[56] R. Perry, G. Mischler, R. Guo, T. Lee, A. Chang, A. Koul, C. Franz, H. Richard, I. Carmichael, P. Ablin *et al.*, "mvlearn: Multiview machine learning in python." *J. Mach. Learn. Res.*, vol. 22, pp. 109–1, 2021.

[57] G. J. McLachlan and S. Rathnayake, "On the number of components in a gaussian mixture model," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 341–355, 2014.

[58] R. Suganya and R. Shanthi, "Fuzzy c-means algorithm-a review," *International Journal of Scientific and Research Publications*, vol. 2, no. 11, p. 1, 2012.

[59] S. Ghosh and S. K. Dubey, "Comparative analysis of k-means and fuzzy c-means algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 4, 2013.

[60] G. J. McLachlan and S. Rathnayake, "On the number of components in a gaussian mixture model," *WIREs Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 341–355, 2014. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1135