MEKALA, M.S., EYAD, E. and SRIVASTAVA, G. 2023. Object-aware multi-criteria decision-making approach using the heuristic data-driven theory for intelligent transportation systems. In *Proceedings of the 10th IEEE (Institute of Electrical and Electronics Engineers) Data science and advanced analytics international conference 2023 (DSAA 2023), 9-13 October 2023, Thessaloniki, Greece*. Piscataway: IEEE [online], 10302554. Available from: https://doi.org/10.1109/DSAA60987.2023.10302554

Object-aware multi-criteria decision-making approach using the heuristic data-driven theory for intelligent transportation systems.

MEKALA, M.S., EYAD, E. and SRIVASTAVA, G.

2023

Object-aware Multi-criteria Decision-Making Approach using the Heuristic data-driven Theory for Intelligent Transportation Systems

M S Mekala
School of Computing, Robert Gordon University, Aberdeen, AB10 7AQ, UK, ms.mekala@rgu.ac.uk

Elyan Eyad
School of Computing, Robert Gordon University, Aberdeen, AB10 7AQ, UK, e.elyan@rgu.ac.uk

Gautam Srivastava
Dept. of Math and Computer Science Brandon University, Brandon, MB R7A 6A9, Canada, srivastavag@brandonu.ca

***Abstract*—Sharing up-to-date information about the surrounding measured by On-Board Units (OBUs) and Roadside Units (RSUs) is crucial in accomplishing traffic efficiency and pedestrians safety towards Intelligent Transportation Systems (ITS). Transferring measured data demands $\geq 10Gbit/s$ transfer rate and $\geq 1GHz$ bandwidth though the data is lost due to unusual data transfer size and impaired line of sight (LOS) propagation. Most existing models concentrated on resource optimization instead of measured data optimization. Subsequently, RSU-LiDARs have become increasingly popular in addressing object detection, mapping and resource optimization issues of Edge-based Software-Defined Vehicular Orchestration (ESDVO). In this regard, we design a two-step data-driven optimization approach called Object-aware Multi-criteria Decision-Making (OMDM) approach. First, the surroundings-measured data by RSUs and OBUs is processed by cropping object-enabled frames using YoLo and FRCNN at RSU. The cropped data likely share over the environment based on the RSU Computation-Communication method. Second, selecting the potential vehicle/device is treated as an NP-hard problem that shares information over the network for effective path trajectory and stores the cosine data at the fog server for end-user accessibility. In addition, we use a non-linear programming multi-tenancy heuristic method to improve resource utilization rates based on device preference predictions (Like detection accuracy and bounding box tracking) which elaborately concentrate in future work. The simulation results agree with the targeted effectiveness of our approach, i.e., mAP($\geq 71\%$) with processing delay ($\leq 3.5 \times 10^6$ bits/slot), and transfer delay ($\leq 38ms$). Our simulation results indicate that our approach is highly effective.**

***Index Terms*—Edge computing, RSU selection, Cyber-physical systems, object detection, path trajectory, Multi-criteria Decision-Making method.**

I. Introduction

IN INDUSTRY 4.0, vehicles currently rely on human intervention up to some extent, there is ongoing development towards a fully automated vehicle system design through the use of novel artificial intelligence-inspired edge computing systems for automobile industry. The main objective of these autonomous vehicle (AV) systems is to identify objects [1], localize them, and segment them [2], which can be achieved through the integration of artificial intelligence and computer vision mechanisms based on edge computing. Subsequently, the Internet of Things (IoT) and Cyber-Physical Systems (CPS) infrastructure have become feasible platforms for fog/edge systems, which enable the processing and computing of sensor data at the data generation source, a crucial aspect of autonomous vehicle systems. Accurate measurement of information quality from edge-based Software-Defined Vehicular Orchestration (ESDVO) is essential for ensuring vehicle and pedestrian safety in Intelligent Transportation Systems (ITS). Edge orchestration facilitates the use of various modalities such as Lidars, vision cameras, and radars. However, the processing capabilities of edge devices are limited, making it impossible for them to process the collected data and respond to environmental requirements.

A. Motivation:

Ensuring vehicle safety through timely obstacle detection is important, but identifying road conditions, pedestrians, and curves can further enhance safety, security, and fuel economy. However, there exist specific challenges to achieving this objective.

1) Due to their limited resources, OBUs are unable to identify road conditions, pedestrians, and curves, which can result in blind spots caused by occlusion issues.
2) Sharing information between vehicles and infrastructure remains a challenge due to coverage issues arising from joint communication and computation problems caused by the high mobility of vehicles and UAVs.
3) The classification of measured data affects resource usage, computation time, and detection accuracy. RSUs, being static, only require a one-time removal of background infrastructure, whereas OBUs such as vehicles and UAVs need attention for every timestamp.

To address these challenges, we have developed AI-RSU-LiDARs that strategically address object detection, mapping, and resource optimization difficulties in Edge-based Software-Defined Vehicular Orchestration (ESDVO). *Suppose Lidar and radar measure data at 20FPS, which can be visualized in 3D projection, and Stereocamera measures data at 30FPS, which can be visualized in 2D projection (as shown in Fig. 1). Sharing such a vast amount of raw data is wasteful of network resources. To address this issue, we propose to eliminate*

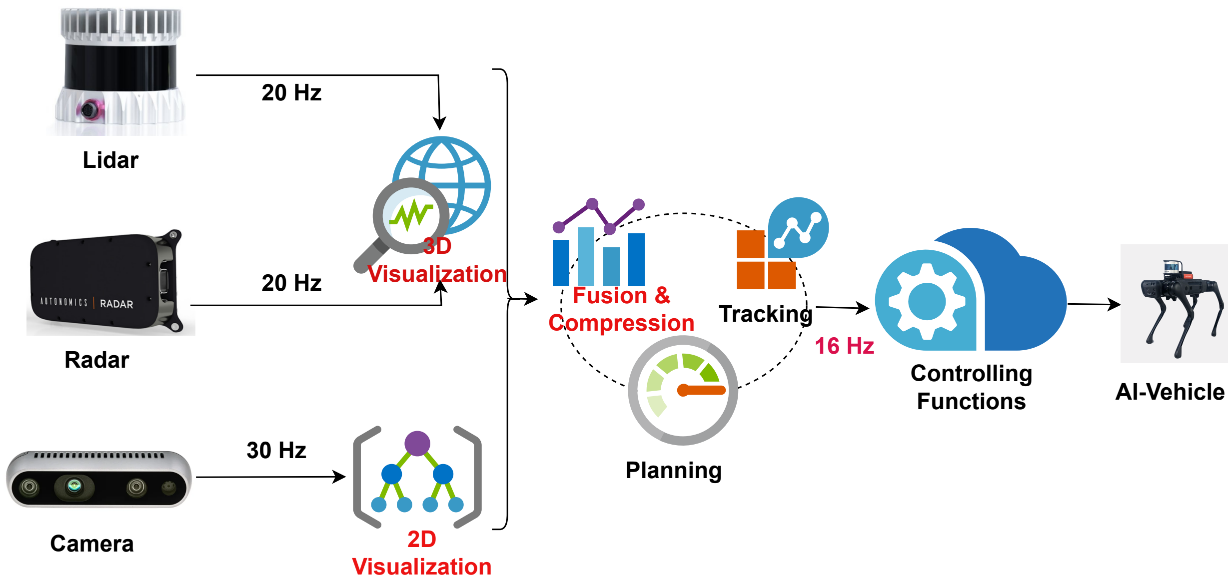


Fig. 1: Data fusion and compression challenges on cloud-assisted Vehicle Edge-Orchestration

unimportant data leads and reduce the size (approximately 16 FPS) for efficient sharing and processing at both RSU and OBU that returns minimized computation and communication overhead of the orchestration.

Automated RSUs share critical object data, including road conditions, pedestrians, vehicles, and curbs, over a network that enhances the functionality of the ITS network. For instance, OBU-shared information can improve scene perception by accurately localizing objects, as demonstrated by simulation results [3], [4]. However, spatial relationship mapping and real-time object detection can be challenging due to limited coverage range, obstacles, and increased computation overhead. Nevertheless, V2X coordination has several advantages, such as

1) Compared to single-sensor vehicle data analysis, RSU-based scene perception has a better field of view (FOV) for better scene understanding.
2) In the case of RSUs, spatial calibration filtering only needs to be performed once since they are statically deployed on the roadside, making it advantageous for industrial applications.

The goal of this paper is to address communication and scene perception challenges and establish a research baseline for future scholars. The main contributions are outlined below:

1) Design a two-step data-driven optimization approach called Object-aware Multi-criteria Decision-Making (OMDM) adaptive fusion approach for optimizing the bandwidth and network latency issue related to raw measured data upload.
2) Design a data compression strategy by deploying the YOLO algorithm for object detection of surroundings and crop the frames based on the object presence of each timestamp, which helps to optimize the use of storage resources.
3) Selecting the potential vehicle/device is treated as an NP-hard problem for sharing essential information over the network for effective path trajectory and storing the cosine data, which is crucial for cloud-assisted applications such as training deep learning models and scene construction.

The manuscript continues as Section II briefly explains related work. Section III describes the proposed system and its mathematical model and algorithm in detail. Section IV, evaluates the investigation outcomes, section V concludes the manuscript, and Section VI describes the feature work.

II. Related Work

As discussed in section I, the limitations can be overcome by adopting an automated V2X approach. In the V2X environment, object detection and tracking challenges can be addressed comprehensively through augmented sensing, and real-time information accuracy is crucial for achieving vehicle safety and fuel economy. However, there are still some challenges related to real-time deployment issues that need to be addressed, as outlined in [5].

A. RSU-based data-processing for bandwidth and network latency optimization

RSUs are crucial for meeting the requirements of ITS, and RSU-BEV technology helps to address issues related to occlusions, obstacles, and network coverage. Combining data from multiple vehicle sensors can resolve far-away object detection issues, as demonstrated in [6]. Cloud-assisted vision functions have been developed to manage traffic and monitor infrastructure continuously, and the authors of [7] have explored the use of digital twin technology for 3D visualization. Another uplooking imperative technique is the digital twin to scale up the ITS service reliability ratio. An additional benefit RSU is sharing the leveraged measures data over the network helps enhance the response time of the environment [6], [7]. An automated RSU-based framework is developed using an ON/OFF strategy to control traffic signals along with reducing energy usage [8]. Software Defined Networking (SDN) is an efficient paradigm for resolving joint computation and communication issues by incorporating multi-programming data-driven methods. To the best of the author's knowledge, SDN has been developed and deployed for vehicular frameworks, which has garnered significant attention from scholars and researchers. SDN has proven to be a valuable platform for measuring and localizing sensor data coordinates, processing vision data, and sharing data via routing methods for the V2X environment, as demonstrated in [9].

B. Object Detection Strategies

There are two main strategies for object detection: conventional techniques and deep learning techniques. Conventional techniques involve several steps such as feature extraction, segmentation, background filtering, object clustering, and classification. On the other hand, deep learning techniques use automated feature extraction and rely on massive amounts of data for the classification process, specifically for detecting vehicles.

To detect vehicles, a clustering algorithm is utilized to segment the point cloud into clusters, and features are used to classify each cluster. However, the accuracy of the classification results depends on the precision of the clustering, which can be affected by incorrect clustering. In the past, a rule-based segmentation was recommended for scene understanding [10], but for RSU LiDAR, the point cloud is processed using the density-based spatial clustering method (DBSCAN) to eliminate background points. Nevertheless, this method does not work effectively with non-uniform data [11]. Therefore, an effective clustering strategy is crucial for integrating modelling graphs on the point cloud to filter out background points [12]. The excess points were processed using a triangulation-based clustering technique after completing background filtering as stated in [13]. Vehicle and non-vehicle moving points were separated and assembled using the Euclidean cluster algorithm in [14], followed by classification using the SVM algorithm. However, parameter collection remains a challenge in clustering algorithms during rush hour traffic scenes.

C. Preliminaries: ROS and OpenC2X

For our simulation, the RSU framework is developed using the Robot Operating System (ROS) with RYU, a popular SDN environment for automated vehicles. RYU is designed to control traffic by creating an adaptive routing policy for sharing information based on environmental changes. ROS, along with its plethora of updated versions, provides a customized platform for sensing and communication facilities. OpenC2X [15] is an open-source tool for vehicular networking which we have used for testing the network reliability. Additionally, MATLAB simulation links serve as the second simulation platform for effective performance analysis. With these improvements, the extended work is currently being drafted.

III. Proposed Approach

In our proposed network, 3-OBUs (i.e, $j = 1, \cdots, m = 3$) and 4-RSUs (i.e, $i = 1, \cdots, k = 4$) are deployed in a certain 1 km/h road length. The 2D coordinates of RSUs are aware, but the OBUs are unaware since they are enabled with mobility mode. Total m number of OBU's and k number of RSU's communicate each other within a transmission range. Fig. 2 illustrates 4-RSUs with 200 meters average equal distance between two RSUs since the road length is $1KM$.

A. RSU Computation and Communication

We aim to select potentially resource-rich RSUs for sharing significant data to avoid latency and reduce the communication-computation overhead ratio. In this regard, we derived sharing policy based on the size of the bandwidth of the service. Note: we treated each scene measurement as a service and denoted as $s = 1, 2, \cdots, N$. The policy is tuned based on queue time and delay for more information please refer to our previous contributions [16]–[20]. In our simulations, the queue waiting is infinite when the RSU video processing ratio and service arrival ratio from OBU remain the same. We considered both latency and queue length parameters while formulating the policy, and those values range should be low. Choosing the j^{th} RSU based on minimised communication and computation delay (υ_i)

$$\upsilon_{\substack{i\\ \min}} = \upsilon_i^{com} + \upsilon_i^{cmp} \tag{1}$$

Where υ_i^{com} indicates communication delay for sharing cropped data over the network and it is calculated as follows

$$\upsilon_i^{com} = \sum_{i=1}^{k} \left(1 - O_s^{\tau}\right) \upsilon_i^{cmp} + \sum_{i=1}^{k} \frac{\varpi_{s,i}^{\tau} \times \alpha_i}{\varphi_i \log_2 \left(\frac{1 + \left(G_i \times \zeta_t^{i,i+1}\right)}{\varphi_i \times (\zeta_A)^2} \right)} \tag{2}$$

Here, ζ_i is energy usage for sharing the data, G_i is channel gain, ζ_A is channel amplification power, and O_s^{τ} indicates the place of service execution. The default value is equal to 0 when it performs on the current RSU; otherwise, offload the video cropping service to another RSU. The computation delay is calculated as follows

$$\upsilon_i^{cmp} = \sum_{i=1}^{k} O_s^{\tau} \times \varpi_i \times CPI \times \tau \tag{3}$$

CPI stands for CPU cycles per instruction or Million Instructions Per Second (MIPS)

B. Object-aware Data-frame Cropping

We create a matrix in our data processing module that includes data collected from all vehicles connected with the relative RSU, as well as a time stamp. This is done for effective analysis, as shown below.

$$U_{i,j} = \begin{bmatrix} \text{MAC}_1, & \text{Time-stamp}, & \text{RSU}_1, & \text{SS}_1 \\ \cdots & \cdots & \cdots & \cdots \\ \text{MAC}_j, & \text{Time-stamp}, & \text{RSU}_i, & \text{SS}_j \end{bmatrix} \tag{4}$$

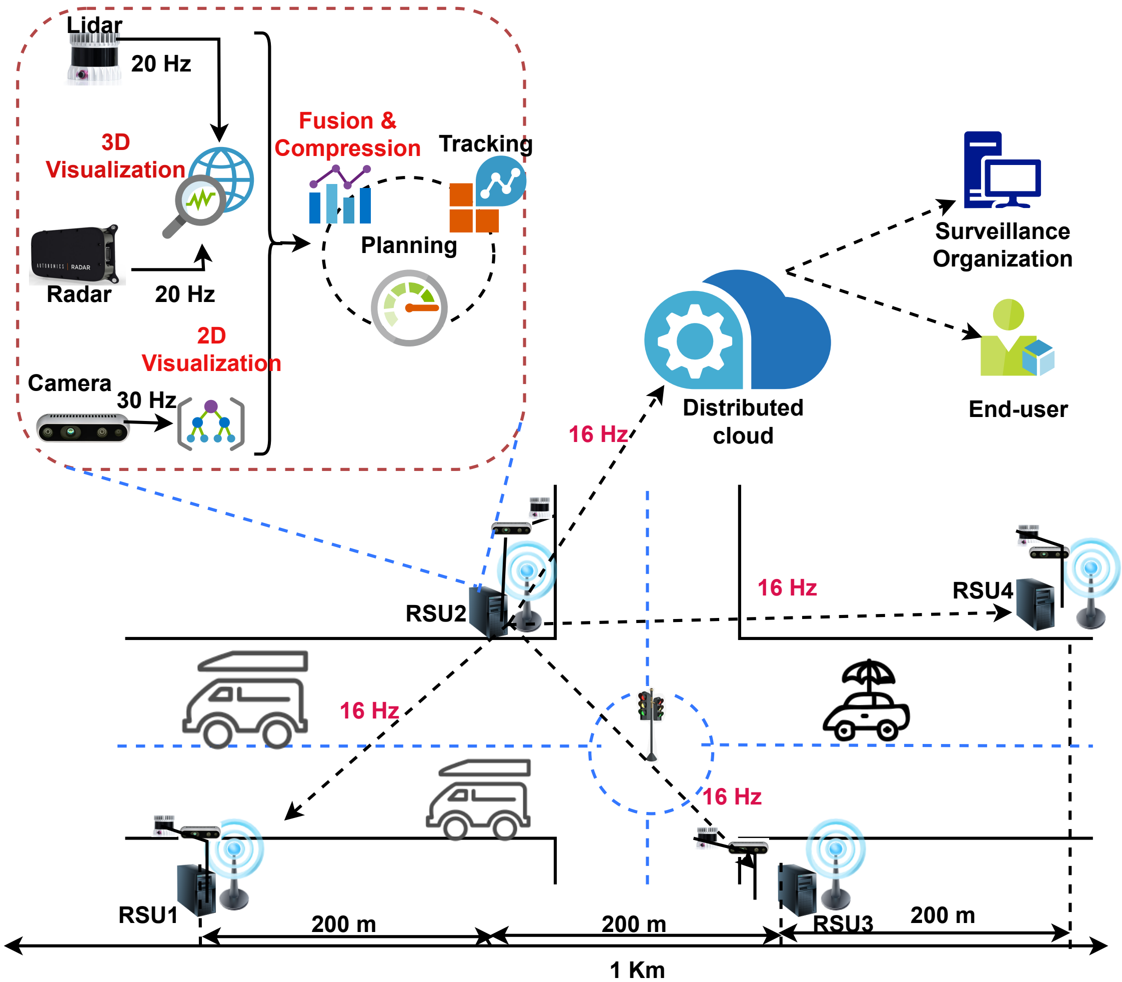


Fig. 2: Sharing the measured data that RSU has processed over a network

Where MAC refers to the OBU address, the time-stamp refers to the time of the received data to the RSU-1 with determined signal strength. Further, the data is classified as RSU and OBU at the data center for effective storage and retrieval purposes. The matrix also helps to reduce the search time and background reduction process. Let us assume that each RSU measures data at 10 Mbps, which circulates throughout the network. In this case, we can calculate the overall required bandwidth ratio as follows:

$$\varphi_i = \sum_{i=1}^{k} \sum_{j=1}^{m} \sum_{\tau=1}^{\Gamma} \varpi_i^{\tau} \times k_i^{\tau} \tag{5}$$

According to the network size, we currently have 4 RSUs where ϖ_i^{τ} represents the measured data size of a single RSU at time τ. The overall bandwidth requirement is calculated as:

$$\varphi_i = 10Mbps \times 4 = 40Mbps$$

However, sharing, processing, and storing 40 Mbps of single time-stamp data from one RSU is not cost-effective and can cause inadequate latency and wastage of network resources in lightweight environments. To address this issue, we design a soft computing method based on data-centric parameters to crop unnecessary data. The YOLO algorithm is applied to identify objects in each frame.

$$\varphi_i^{\iota} = \sum_{i=1}^{k} \sum_{j=1}^{m} \sum_{\tau=1}^{\Gamma} \varpi_i^{\tau} \cdot \phi_i \cdot \zeta_i^{\tau} \cdot \omega \cdot \left(1 + K_{i,j}^{\tau}\right) \tag{6}$$

In this problem, we refer to the object-based cropped data size as ϖ_i^{τ}, the average number of objects detected per frame as ϕ_i, and the ratio between the number of frames per second of cropped and measured data as ζ_i^{τ}. The measured data FPS may differ depending on the sensor used, with cameras typically having a higher FPS ratio compared to Lidar and radar. We also refer to the cropped image size as ω per detection per frame. To aid in understanding, we present the following problem.

Let's assume that the length of the road is 1 kilometer with [45-60 kph] speed, and approximately 4 RSUs are required if a distance of 200 meters is considered between two RSUs. The measured data size is 10 Mbps with a resolution of 1080p and a frame rate of 30fps. The probability of object detection per frame is 0.5, and the cropped data size as per the detection of objects has a frame rate of 16 fps. The initial cropped size ratio is $\omega = 1/100$. Given data is

$$k_i^\tau = 4, \omega = 1/100, \varpi_i^\tau = 10Mbps$$

$$\zeta_i^\tau 1 = 16/30, \phi_i = 0.5$$

The expected required brand-width is $\varphi_i^t = 132.5Kbps$.

Algorithm 1: Computation-Communication Strategy

input : RSU set k, service set n, OBU set m
output: Choosing right RSU
1 Let initialize $\upsilon_i^{com} \neq 0$, $\upsilon_i^{cmp} \neq 0$, $\varpi_i \neq 0$, $\varphi_i \neq 0$, $\phi_i \neq 0$, $\zeta_i^\tau \neq 0$
2 **while** $k \neq 0$ **do**
3   **for** $s = 1$ to $length(m)$ **do**
4   **# Delay Analysis #**
5   Calculate υ_i^{com} using Eq. 4;
6   Calculate υ_i^{cmp} using Eq. 5;
7   Update the probability of overall delay $\upsilon_i \leftarrow \upsilon_i^{com} + \upsilon_i^{cmp}$;
8   **if** $\upsilon_i \geq 0.5$ **then**
9   Calculate the probability of waiting for service queue length;
10   Update $\delta_i^\tau \leftarrow \sum_{s=1}^{n-1} \delta_s^\tau$;
11   **if** $\delta_i^\tau \geq 0.4$ **then**
12   Choose current i^{th} RSU from global manager pool set;
13   **end**
14   **else**
15   Select the second RSU from the list and send data to the distributed cloud for storage;
16   **end**
17   **end**
18   **else**
19   Go to step-2, measure environment response;
20   **end**
21   **end**
22 **end**

In our proposed environment, we treat each RSU as an *agent* and define it with a set of characteristics, such as the count of *MIPS*, transmission power, channel frequency, bandwidth, coverage range, service time, and average number of connecting OBUs. The aim of Algorithm 1 is to select the potential RSU for sharing cropped information. In Line-1, we initialize the required parameters, which are RSU-centric measurements. Then, in Line-2, we check each RSU with its defined characteristics, and in Line-3, we check each scene video clip that has been listed and cropped after the deployment of the YOLO algorithm [21]. Lines 4-7 assess the computation and communication delay and their probabilities. The final probability helps to decide on the selection of the RSU with the highest probability of sharing the data, as can be observed from Lines 8-12. If the whole process is not likely, then we select the second most eligible RSU. Otherwise, we repeat step 2 to measure the environment reward and response.

IV. Result Analysis

In this section, we present a concise summary of the proposed system's performance through a series of simulations. The simulations were conducted using MATLAB on a 64-bit Ubuntu 20.1 LTS operating system, with hardware consisting of an Intel Core i7-10700 CPU @ 3.80GHz × 16 and an NVIDIA GeForce RTX3090. To evaluate the system's performance, we utilized OpenC2X and ROS and made necessary adjustments to the generated results to demonstrate the achievement of the stated objectives.

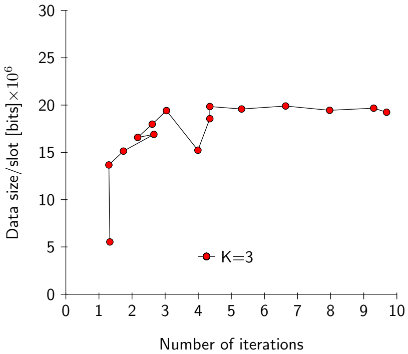
In our designed environment, we considered the presence of 4 RSUs and 6 OBUs, interconnected via a cellular network with a backhaul link providing a capacity of 10 Gb. To assess the system's performance, we conducted a total of 300 simulation rounds, each consisting of 50 epochs. These simulations aimed to measure the effectiveness of both the optimal and superior environments.

For the specific video scene (service) under examination, the size was expected to be $\varpi_i = 15Mbps$. The Central Processing Unit (CPU) executed 1 cycle per second (CPI), and the channel energy usage was $\zeta_i = 8 \times 10^{-5}W$. The lidar device collected data by performing 24 firing cycles per packet, and the size of each packet was calculated using the formula:

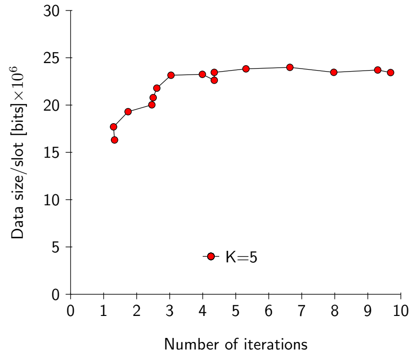
$$d_p = \frac{1248bytes}{24 \times 55.296\mu s}$$

The performance analysis of the proposed model utilized two approaches: the Optimized Service Offloading (OSO) [22] approach and the Online Scheduling (ONS) [23] approach. Fig. 3 presents the analysis of data computation concerning the number of iterations. The sub-figures illustrate the data computation time in different scenarios, where the k value scales from 3 to 8. Fig. 3(a) showcases the average data size (in bits), referred to as the cropped data, which is ready to be shared over the network. It is evident that the data size significantly increases from the first iteration, and the sixth iteration onwards, there is consistency in size. The proposed algorithm aims to reduce the size through bbox detection, selecting potential RSUs for consistent frequency, and minimizing computation-communication overhead. Fig. 3(b) and Fig. 3(c) also illustrate similar objectives as discussed earlier, but with an increased number of RSUs from 5 to 8, and an increase in the size of each data packet as the iteration count rises.

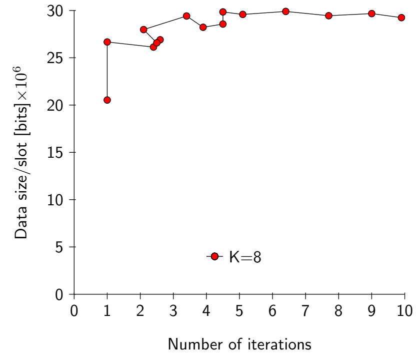
Fig. 4 presents two different analysis reports: Random Mean Square Error (RMSE) concerning sampling time and Hovering time concerning collected data size (in Mbytes). Fig. 4(a) provides a visualization of the signal strength and error ratio for each RSU, along with a detailed analysis based on state-of-the-art approaches. The sampling time is expressed in minutes, and it is observed that the ONS model exhibits an unusual



(a) Data analysis with $K = 3$

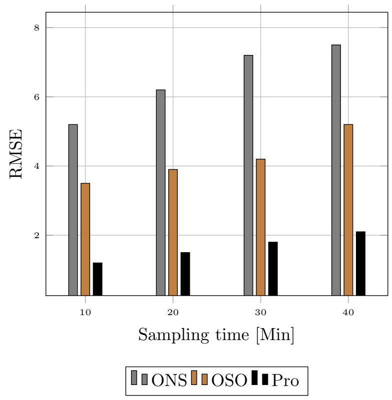


(b) Data analysis with $K = 5$

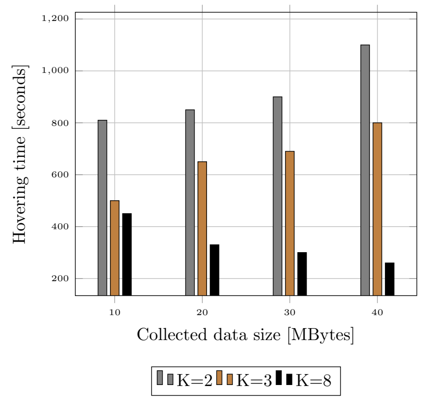


(c) Data analysis with $K = 8$

Fig. 3: Data processing analysis with various counts of RSU

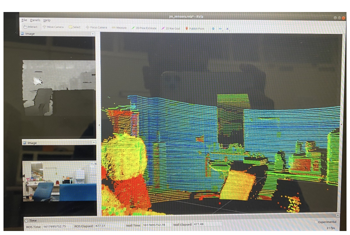


(a) Error analysis based on sampling time



(b) Hovering time analysis

Fig. 4: Data processing analysis with various counts of RSU



(a) Indoor frame and ROS interface

(b) Filtered frame-1 with objects

(c) Filtered frame-2 with objects

Fig. 5: Frame filtering analysis, still far-way objects are not detected as intended

error ratio, while the OSO model performs better than ONS. However, our proposed model achieves a lower error ratio compared to both approaches. For instance, with a sampling time of 40 minutes, the error ratios for ONS, OSO, and our model are 7.2%, 5.6%, and 2.1% respectively.

Fig. 4(b) illustrates the required hovering time for processing (cropping the frames) the data at the RSU. Our algorithm crops the data size based on object awareness by validating each frame. The simple heuristic-based cropping method effectively filters the frames, but it still relies on the detection methods. We employed YoLO, which has certain limitations such as difficulty in detecting small objects and objects that are far away. However, from a hovering time perspective, the computing environment demonstrates acceptable processing time, which varies depending on the number of RSUs.

Fig. 5 showcases the ROS interface and the detection of objects in each frame. Due to the limitations of the YoLo3 version, some objects may not be identified, as observed in Fig. 5(b). Additionally, a faraway object (car) was not detected, as seen in Fig. 5(c).

In this paper, we did not focus on addressing the detection

Model	mAP	FLOPS (B)	Layers	FPS
YoLo [21]	0.71	140.69	106	20
FRCNN [24]	0.65	62.94	32	24
Model	**CAR**	**Truck**	**Persons**	**Buses**
YoLo [21]	0.88	0.92	0.89	0.91
FRCNN [24]	0.77	0.85	0.81	0.86

TABLE I: Comparison results for customized dataset

issues, but we plan to address and contribute to them in future drafts. For our experiments, we utilized our dataset consisting of short video scenes from Deagu City, Republic of Korea. Each video clip contained approximately 416×416 colour images with detailed annotations and flags to highlight differences in the background and frames. The dataset comprised around 5,900 images of size 12080×720. We divided the dataset into 80% for training and 20% for validation purposes. A total of 9,000 labels were annotated, covering four major classes including pedestrians, cars, trucks, and buses.

During the model preprocessing stage, the video clips served as input to the model. Each frame was filtered, and the resulting information was passed to the base learner architecture to extract features and localize each object in every frame, as shown in Fig. 5(b) and Fig. 5(c). To evaluate the detection accuracy, we employed the Intersection Over Union (IOU) metric. Each bounding box was cross-evaluated to determine whether the detection was true or false, based on the calculated IOU value.

$$IoU = \begin{cases} \text{is false} \quad \text{for} \quad < 0.5 \\ \text{is true} \quad \text{for} \quad \geq 0.5 \end{cases} \tag{7}$$

The achieved results, including the tracking and detection accuracy of the model, are presented in Table I. The mapping of floating point operations (FLOPs) with FPS enhanced the detection accuracy and ratio of true detection. Additionally, we observed that the concatenation of layer counts did not have a significant impact on accuracy improvement.

V. Conclusion

This paper developed a two-step data-driven optimization approach called Object-aware Multi-criteria Decision-Making (OMDM) method for accomplishing traffic efficiency and pedestrians safety towards Intelligent Transportation Systems (ITS). We achieved the targeted objectives based on RSU-LiDARs through Edge-based Software-Defined Vehicular Orchestration (ESDVO). The first objective is accomplished by employing the YoLo algorithm to detect the ground truth objects and assign bounding boxes to effectively trim the frames that enabled objects. This process has efficiently preserved the storage space and network resources by sharing truthful cropped data that enhanced the reliability of continuous path trajectories for surveillance. The second objective is accomplished by leveraging the node-centric parameters before selecting a potential device over the network that had played a vital role in mitigating individual processing costs. This process has been effectively formulated using a non-deterministic algorithm within the polynomial time. Our simulation results indicate that our approach is highly effective. The simulation proved the effectiveness of our approach, i.e., mAP($\geq 71\%$) with processing delay ($\leq 3.5 \times 10^6$ bits/slot), and transfer delay ($\leq 38ms >$).

VI. Future Work

In addition, a non-linear programming multi-tenancy heuristic method will consider improving resource utilization rates based on device preference predictions (Like detection accuracy and bounding box tracking) which elaborately concentrate in future work. Few future work challenges

1) The future work will enhance detection accuracy, and recover the information loss, i.e., free space which is the projected Field of View (FOV) of the ego vehicle as illustrated in Fig. 6. One possible way to recover the free space will be by reconstructing the scene perception using Bird's-Eye-View (BEV) models based on RSU-measured data by mapping angular coordinates, which will be elaborately concentrated on in the future.
2) Maintain individual index as per the classification of RSU OBU/UAV measured data for mitigating the execution, communication and storage cost.
3) Detected objects may remain repeatedly detected in a few successive frames. So again filtering those redundant frames is a mathematically challenging task but there is the possibility to resolve the issue using Bayesian theorem or data compression theory.
4) Design a 3D-Perception model for effective reconstruction of the scene based on BEV and FOV at Edge-RSU.

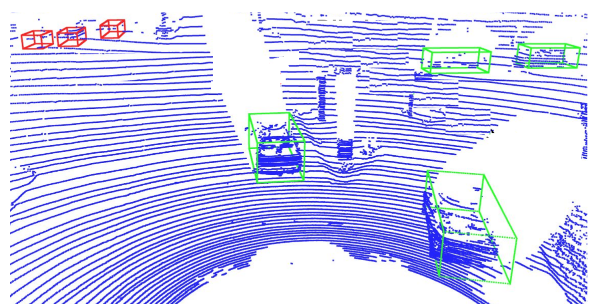


Fig. 6: Detection challenges include the red boxes failing to detect at each frame, and the free space behind the green bbox is likely to reconstruct to formulate the occlusion issues as well [25].

References

[1] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau, "Road scenes analysis in adverse weather conditions by polarization-encoded images and adapted deep learning," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 27–32.

[2] Z. Yang, H. Yu, M. Feng, W. Sun, X. Lin, M. Sun, Z.-H. Mao, and A. Mian, "Small object augmentation of urban scenes for real-time semantic segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 5175–5190, 2020.

[3] Z. Xiao, D. Yang, F. Wen, and K. Jiang, "A unified multiple-target positioning framework for intelligent connected vehicles," *Sensors*, vol. 19, no. 9, p. 1967, 2019.
[4] F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors*, vol. 17, no. 2, p. 271, 2017.
[5] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of v2x communication in support of cooperative autonomous driving," *IEEE communications magazine*, vol. 53, no. 12, pp. 64–70, 2015.
[6] X. Zhang, A. Zhang, J. Sun, X. Zhu, Y. E. Guo, F. Qian, and Z. M. Mao, "Emp: Edge-assisted multi-vehicle perception," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 545–558.
[7] V. Tihanyi, A. Rövid, V. Remeli, Z. Vincze, M. Csonthó, Z. Pethő, M. Szalai, B. Varga, A. Khalil, and Z. Szalay, "Towards cooperative perception services for its: Digital twin in the automotive edge cloud," *Energies*, vol. 14, no. 18, p. 5930, 2021.
[8] M. Laha and R. Datta, "Intelligent on/off switching of mmrsus in urban vehicular networks: A deep q-learning approach," in *2022 National Conference on Communications (NCC)*, 2022, pp. 221–226.
[9] Z. Li, T. Yu, R. Fukatsu, G. K. Tran, and K. Sakaguchi, "Towards safe automated driving: Design of software-defined dynamic mmwave v2x networks and poc implementation," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 78–93, 2021.
[10] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor," *IEEE Robotics and Automation Letters*, 2021.
[11] Z. Zhang, J. Zheng, H. Xu, X. Wang, X. Fan, and R. Chen, "Automatic background construction and object detection based on roadside lidar," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4086–4097, 2019.
[12] A. Saglam, H. B. Makineci, N. A. Baykan, and Ö. K. Baykan, "Boundary constrained voxel segmentation for 3d point clouds using local geometric differences," *Expert Systems with Applications*, vol. 157, p. 113439, 2020.
[13] A. P. Tarko, K. B. Ariyur, M. Romero, V. K. Bandaru, and C. L. Jimenez, "Guaranteed lidar-aided multi-object tracking at road intersections: Usdot region v regional university transportation center final report." 2016.
[14] J. Zhang, W. Xiao, B. Coifman, and J. Mills, "Image-based vehicle tracking from roadside lidar data," *ISPRS Geospatial Week 2019*, 2019.
[15] F. Klingler, G. S. Pannu, C. Sommer, B. Bloessl, and F. Dressler, "Field Testing Vehicular Networks using OpenC2X," in *15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2017), Poster Session*. Niagara Falls, NY: ACM, 6 2017, pp. 178–178.
[16] M. S. Mekala and P. Viswanathan, "Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for iot," *Computers & Electrical Engineering*, vol. 73, pp. 227–244, 2019.
[17] M. S. Mekala, G. Dhiman, W. Viriyasitavat, J. H. Park, and H.-Y. Jung, "Efficient lidar-trajectory affinity model for autonomous vehicle orchestration," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2023.
[18] M. S. Mekala, H. Zhang, J. H. Park, and H.-Y. Jung, "A service consolidation approach for edge-vehicle network using multi-agent decision-making method," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023, pp. 1–7.
[19] M. S. Mekala, G. Dhiman, G. Srivastava, Z. Nain, H. Zhang, W. Viriyasitavat, and G. P. S. Varma, "A drl-based service offloading approach using dag for edge computational orchestration," *IEEE Transactions on Computational Social Systems*, pp. 1–9, 2022.
[20] M. S. Mekala, A. Jolfaei, G. Srivastava, X. Zheng, A. Anvari-Moghaddam, and P. Viswanathan, "Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 245–254, 2022.
[21] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.
[22] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5506–5519, 2018.
[23] K. Liu, J. K. Ng, V. C. Lee, W. Wu, and S. H. Son, "Towards scalable, fair and robust data dissemination via cooperative vehicular communications," in *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2014, pp. 1–9.
[24] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated rcnn and multi-class deep sort for multi-object detection and tracking," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 1, pp. 171–181, 2021.
[25] Y. Wang, A. Fathi, A. Kundu, D. A. Ross, C. Pantofaru, T. Funkhouser, and J. Solomon, "Pillar-based object detection for autonomous driving," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 18–34.