Quantitative Analysis of Storage Requirement for Autonomous Vehicles

Yuxin Wang
University of Delaware
yuxw@udel.edu

Yuankai He
University of Delaware
willhe@udel.edu

Ruijun Wang
Ford Motor Company
rwang53@ford.com

Weisong Shi
University of Delaware
weisong@udel.edu

Abstract

This study addresses the critical aspect of data storage requirements for Autonomous Vehicles (AVs). With AVs generating substantial amounts of data daily, understanding these requirements is vital for AV storage systems design, enhancing vehicle safety, efficiency, and operational integrity. Through a comprehensive analysis of onboard sensor and CAN bus data, alongside a novel mathematical model, this research offers insights into the storage needs, assesses system durability, and proposes a tailored storage solution and system architecture. The findings aim to guide the development of future AV storage systems, emphasizing the importance of data-driven decision-making in AV technology advancements.

CCS Concepts: • **Information systems → Storage architectures**; • **Computer systems organization → Embedded and cyber-physical systems**.

Keywords: Storage, Autonomous Vehicles, Sensor Data

ACM Reference Format:
Yuxin Wang, Yuankai He, Ruijun Wang, and Weisong Shi. 2024. Quantitative Analysis of Storage Requirement for Autonomous Vehicles. In *16th ACM Workshop on Hot Topics in Storage and File Systems (HOTSTORAGE '24), July 8–9, 2024, Santa Clara, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3655038.3665948

1 Introduction

The increasing prominence of Autonomous Vehicles (AVs) has spotlighted the essential role of data in ensuring their reliability and safety. The utilization of on-board Random Access Memory (RAM) facilitates real-time sensor data processing, while Solid-State Drives (SSD) and Hard Disk Drives (HDD) archive historical data, allowing for ongoing improvements to AV safety and efficiency. [2, 4, 5, 12].

Despite the critical nature of data storage in autonomous vehicle (AV) systems, discussions and analyses in existing literature remain sparse. This gap highlights a lack of detailed investigation into the immense data generated by AVs and their storage implications. This paper endeavors to bridge this gap by comprehensively analyzing AV data storage requirements, focusing on onboard sensors and Controller Area Network (CAN) bus data.

Specifically, we introduce a mathematical model to estimate memory and storage needs based on various driving scenarios and validated with real-world data collected from our autonomous vehicle equipped with multiple sensors including 3D LiDAR, camera, and Global Navigation Satellite System (GNSS). These are the most common sensors used on AVs to monitor the environment [10, 15, 17]. The proposed model and subsequent analysis illuminate the path toward optimized storage system design for future AVs, marking a significant step forward in addressing the complexities of AV data management. In general, this work provides a guideline for understanding the AV onboard storage capacity based on specific requirements.

The structure of the paper is as follows: Section 2 summarizes related works and presents general data types generated by AVs. Section 3 provides a detailed examination of onboard sensors and CAN bus data, alongside the mathematical model for sensor data rate calculation. Section 4 evaluates and validates our mathematical models in real-world tests on autonomous vehicles. Section 5 discusses preliminary findings on memory and storage capacities and design a storage system architecture for AVs. Finally, Section 6 discusses the limitations and future work and proposes a few open-ended questions.

HOTSTORAGE '24, July 8–9, 2024, Santa Clara, CA, USA

ACM ISBN 979-8-4007-0630-1/24/07…$15.00
https://doi.org/10.1145/3655038.3665948

2 Background and Related Work

This section introduces the different sensor data generated by an autonomous vehicle and discusses the limitations of recent works on data storage.

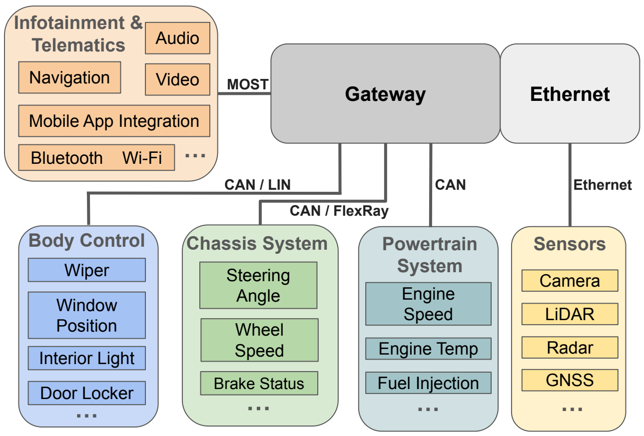

Figure 1. The data generated by a typical autonomous vehicle.

2.1 Sensor Data from an Autonomous Vehicle

As shown in Fig 1, a typical AV generates three types of data: 1) safety-related data that monitor vehicle status and environment, 2) car's interior data like doors, seats, windows, etc, and 3) media, communication, and entertainment data. Prior research demonstrates that the safety-related Electrical Control Unit (ECU) is usually connected with the Controller Area Network (CAN) bus and FlexRay to achieve high-speed and stable communication. The car's interior control only needs serial communication, so it is usually connected with the Local Interconnect Network (LIN) bus. Infotainment units use Media Oriented Systems Transport (MOST) bus to achieve communication [1, 18]. The differences between these communication protocols are summarized in Table.1.

2.2 Related Works

Data storage for autonomous vehicles has only been discussed in the past few years. In 2020, Wang et al. introduced the first conceptual storage framework for AVs, which, while insightful, lacked a detailed storage solution tailored to address the substantial volumes of data generated in real-time [13]. Subsequently, in 2022, Kim embarked on research focused on the data storage needs of AVs, estimating the volume of data produced hourly by utilizing an open dataset. However, this study did not offer a granular analysis of data rates for individual sensors, nor did it validate the proposed data through practical assessments [9]. Kim's further exploration into AV sensors' data provided a general estimation of data volume contingent on the level of vehicle autonomy but fell short of delivering a precise analysis or recommendations [8].

It's important to differentiate existing works on autonomous vehicle storage and retrieval systems, which predominantly address indoor warehouse or parking lot management, from our focus. Our work is dedicated to analyzing the on-board data storage requirements specifically tailored for autonomous vehicles.

3 Data Analysis and Modeling

This section is dedicated to delivering an in-depth examination of the sensor array commonly found in autonomous vehicles, coupled with the presentation of a mathematical model designed to estimate the volume of data these sensors generate.

3.1 Sensors Data Approximation

RGB Camera: The data generated by the RGB camera is in the format of images or frames in video. Each frame or image consists of pixels, and the value of each pixel is determined by the value of the RGB channel.

The math model for the data rate of RGB Camera is presented below:

$$DataRate_{RGB}(bytes/s) = \frac{1byte}{8bit} \times H \times V \times BPP \times FPS \quad (1)$$

H represents the horizontal number of pixels and V represents the vertical number of pixels in an image. BPP represents the pixel bit depth - the number of bits to represent each pixel. The larger the BPP, the more colorful the image. FPS represents the frame rate per second.

3D LiDAR: The LAS (LIDAR Aerial Survey) file format is a widely used binary file format designed to store 3D point cloud data collected by LiDAR. LAS files contain a collection of individual LiDAR points, each with a set of attributes such as X, Y, and Z coordinates, intensity values, return numbers, and classification codes. There are 5 point-data-record formats according to the LAS SPECIFICATION VERSION 1.3 [1], the data depth of each point could be further calculated through these data formats.

The math model for the data rate of 3D LiDAR is presented below:

$$DataRate_{LiDAR}(bytes/s) = N \times B(bytes) \quad (2)$$

N represents the number of returned points. B denotes the bit depth per point, a parameter determined by the data format.

Radar: The data generated by radar is represented using the PointCloud2 Message in ROS2. and the math model for the data rate of Radar is presented as:

$$DataRate_{Radar}(bytes/s) = f \times N_p \times B(bytes) \quad (3)$$

[1] LAS SPECIFICATION VERSION 1.3 – R11. 2010. [online] Available: https://www.asprs.org/wp-content/uploads/2010/12/LAS_1_3_r11.pdf

Table 1. Communication protocols comparison in AVs.

Protocols	Data Rate	Topoogy	Determinism
LIN	Up to 19.2Kbps	Single-wire serial bus	Non-deterministic
CAN	Up to 1Mbps (Classical CAN)	Multi-master serial bus	Non-deterministic (Classical CAN)
FlexRay	Up to 10Mbps	Dual-channel bus	Highly deterministic
MOST	Up to 150Mbps (MOST150)	Ring or star	Deterministic
Ethernet	Up to 10Gbps (10 Gigabit Ethernet)	Bus or star	Non-deterministic

The parameter of f and N_p could be calculate through the following equation:

$$f = \frac{1}{CycleTime}$$

$$N_p = \frac{FOV\ Azimuth}{Azimuth\ Resolution} \times \frac{FOV\ Elevation}{Elevation\ Resolution} \quad (4)$$

N_p represents the number of points that can be resolved within the radar's field of view (FOV) based on its azimuth and elevation resolution. f is the scanning rate which represents the number of scans per second. B is the bit depth per point. The number of points reflected from each pulse varies with the complexity of the environment. So $DataRate_{Radar}$ can only approximate the upper limit of the data generated by the Radar sensor per second.

GNSS: The messaging protocol used to send GNSS data is NMEA-0183 [11]. NMEA[2] and has different sentence formats that could be used in different applications. For example, GPGLL stands for Geographic Position, Latitude / Longitude, and Time, and it will provide this specific information. No matter which sentence is used, the maximum length of an NMEA message is 82 characters or 82 Bytes if the data bits are 8. The math model for the data rate of GNSS is conducted below:

$$DataRate_{GNSS}(bytes/s) = r_{update} \times D(bytes) \quad (5)$$

r_{update} represents the number of messages it generates per second. D is a constant data size for each message. If the messaging protocol follows NMEA and only sends one NMEA sentence, then the maximum value of D is 82 Bytes.

3.2 CAN Bus Data

The Controller Area Network (CAN) bus operates as a sophisticated network system allowing multiple masters to broadcast messages at a signaling rate of up to 1 million bits per second [6]. This system's broadcast capability ensures that all nodes within the network can simultaneously receive messages, giving them the autonomy to either process or discard these messages as needed. Such a mechanism facilitates the swift exchange of data among the Electronic Control Units (ECUs) without requiring direct addressing, significantly boosting the network's communication efficiency. Moreover, the CAN bus employs a deterministic approach to message transmission, prioritizing messages through a unique identifier system where messages with lower identifiers are given precedence. This prioritization is crucial for the timely delivery of vital information, especially for the vehicle's safety-critical systems [3, 7]. Sensors that track fundamental vehicle metrics such as wheel speed, brake status, and steering angle typically utilize the CAN bus for their data communication needs, characterized by compact message sizes and the necessity for rapid transmission.

4 Profile and Evaluation

This section showcases our research framework and validates the mathematical model we proposed with data collected from real-world scenarios. It features a comparative analysis, including the theoretical data rates predicted by our model with the empirical data gathered during our experiments. It is important to note that the math model approximation is purely based on the datasheet provided by the manufacturers, and we also include an adjusted approximation to compensate for this discrepancy between the datasheet and the physical sensor.

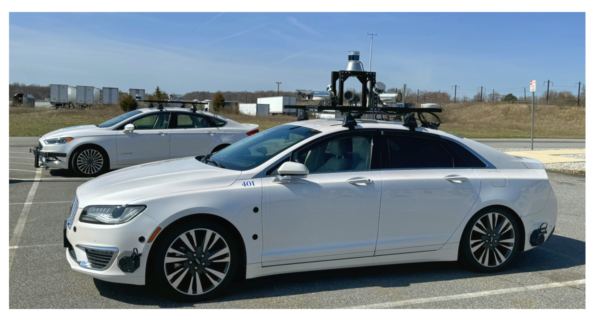

Figure 2. Hardware setup for AV data collection.

The autonomous vehicle, a 2018 Lincoln MKZ, is used as our research platform, shown in Fig 2. It is retrofitted with a range of sensors: one Hesai Pandar64 LiDAR positioned on the top, two VLP-16 LiDARs on the side, seven Basler ace cameras on the top facing various directions, and two Novatel OEM7 GNSS units. CAN bus data are collected through the drive-by-wire system, including metrics like wheel speed, position, brake status, throttle information, turn signals, and

[2]NMEA 0183 is a combined electrical and data specification for communication between marine electronics like GPS receivers, and many other types of instruments. It has been defined and is controlled by the National Marine Electronics Association (NMEA)

control messages. Due to the absence of radar sensors on the Lincoln, the radar math model will not be validated in this experiment.

The evaluation encompasses four distinct scenarios: 1) traversing rural terrain at speeds of 15 miles per hour (MPH), 2) 25 MPH in rural terrain, 3) urban environments, and 4) highway driving with varying speeds. These scenarios serve as a means to investigate the impact of velocity and environmental factors on sensor data acquisition rates.

The data rate for each sensor is shown in Fig 4, and the math model approximation error is shown in Fig 5. The figure for each sensor includes 13 trial points along with 1 model point or adjusted point. 'math model' is the results calculated using a built equation with the data sheet giving message rate. 'adjusted' shows the results using the built equation with actual message rate. Each point gives us information about the data size generated per second and the message rate which is decided by the sensor configuration. For example, the frequency of GPS data published is set to 50Hz by the Novatel OEM7 GNSS kit, which means it will send 50 GPS messages per second.

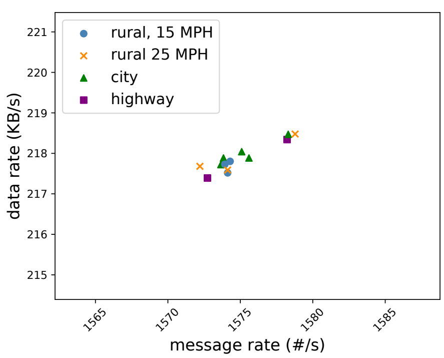

Figure 3. CAN Bus data rate.

CAN Bus Data: As shown in Fig 3, the CAN bus data rate varies little when the vehicle is in constant motion. This is expected because the CAN bus publishes information about the vehicle's status. It is also interesting to note that the driving mode does not influence the CAN bus publishing data rate, whether in normal, sports, or echo mode, the data rate still stayed around 217.833KB/s.

RGB Camera: The RGB Camera on Lincoln is Basler ace, which has a resolution of $1920px \times 1200px$. The default pixel bit depth is 8 bits, and the default frame rate is 5 fps. The average data rate for RGB cameras in the city, rural, and highway are **11.623MB/s**, **11.531MB/s**, and **11.645MB/s** respectively. Our math model approximates the data rate of one RGB camera to be **11.52 MB/s**. However, because the camera's measured message rate is 5.004, we adjust the math model approximation, shown in Fig 4 (a) and Fig 5 (a), resulting in a maximum approximation error of 2.5%.

3D LiDAR: There are three 3D LiDARs, one Hesai Pandar64 operating at 10Hz and publishing 2,304,000 points

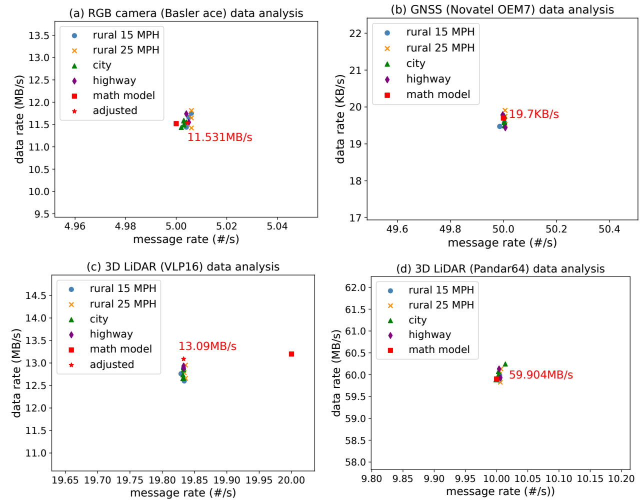

Figure 4. Data generated per second.

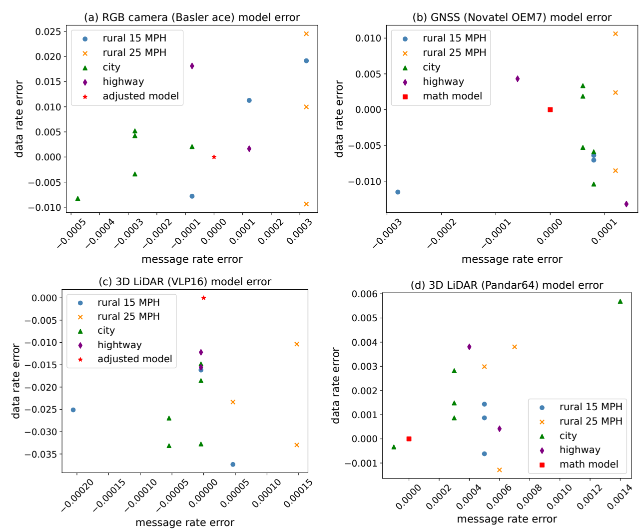

Figure 5. Math model approximation error.

per second and two VLP16 LiDARs each operating at 20Hz and publishing 600,000 points per second. The ROS2 data messages are sized at 26 bytes and 22 bytes, respectively, encompassing details such as x, y, z coordinates, intensity, ring, and timestamp.

In the 13 trials, the Pandar64 message frequency ranged from 9.999Hz to 10.014Hz, and the VLP16 message frequency ranged from 19.829 to 19.836Hz. The average data rate is **60.009MB/s** for the Pandar64 and **12.788MB/s** for the VLP16. The approximate data rate for the Pandar64 is **59.904MB/s** with datasheet message rate and **13.2MB/s** for the VLP16 with the actual message rate showing as an 'adjusted model'. As shown in Fig 5 (d) and (c), the approximation error for the Pandar64 is 0.6% and the approximation error is 3.5% after adjustment. *It is important to note that, contrary to common belief, the LiDAR data size* ***is not*** *influenced by speed or the complexity of the environment.*

GNSS: The test platform is equipped with 2 Novatel OEM7 GNSS sensors. For each GNSS sensor, the math model approximates the data rate to be **19.7 KB/s**, and the measured average data rate is **19.624KB/s**, as shown in Fig 4 (b).

5 Storage Solution

This section introduces the storage requirements for autonomous vehicles based on previous experiments and proposes a storage system architecture embedded into AV.

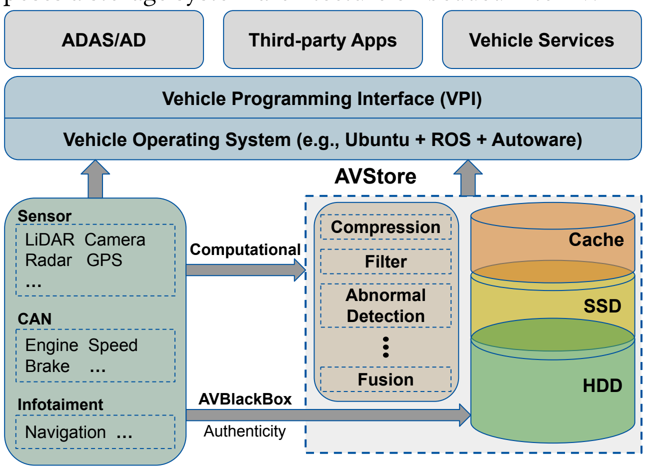

Figure 6. Storage system architecture in AV.[3]

Fig 6 illustrates the storage system architecture for autonomous vehicles, and the details of the AVStore will be designed in the future. In autonomous vehicles, data from sensors, CAN bus, or infotainment sources is continuously published and processed in real-time before being stored in the storage system. Alternatively, algorithms can access raw data directly when needed. For AVStore, data writing involves two key strategies: firstly, the storage system can retain raw data for a limited time without any processing, akin to a BlackBox functionality, and the authenticity will be applied to the data writing protocols to keep the data integrity. Secondly, data can undergo various processing steps such as compression, filtering, fusion, or abnormality detection and only store the computational results. The storing way depends on specific application requirements. The choice of storage medium, whether Cache, SSD, or HDD, is determined by data usage frequency. For instance, data critical for life support, small in size, and frequently accessed may be stored directly in Cache.

On the data read side, we have developed a Vehicle Programming Interface (VPI) [16] abstraction to bridge the lower-level operating system and hardware with upper-level user applications. A preliminary implementation of VPI is built on Ubuntu, Robotic Operating System (ROS), and Autoware [4], leveraging these lower-layer modules to retrieve real-time or stored data and deliver computing results to upper-layer applications. To elucidate the data flow, consider the LiDAR Simultaneous Localization and Mapping (SLAM) algorithm in autonomous driving (AD) as an example. This algorithm utilizes VPI to access computed results from the Autoware localization module. The raw 3D data points generated by the LiDAR sensor undergo filtering by a voxel grid filter before being stored in the storage system. Subsequently, they are retrieved and processed by the Autoware module. For this preliminary storage system design, the data storing location and content are decided by the data type and using requirements.

As shown in the previous sections, the amount of data an autonomous vehicle generates over a one-day period can be summarized as follows. $T_{storage} = 86400 seconds/day \times (N_c \times DataRate_{RGB} + N_l \times DataRate_{LiDAR} + N_r \times DataRate_{Radar} + N_g \times DataRate_{GNSS} + DataRate_{CAN})$.

The specific results for each sensor are further calculated. And the memory and storage requirements are summarized in Table 2 accordingly.

Table 2. Storage requirements for our platform.

Type	Memory Access Requirements	Storage Size (collected)	Storage Size (math model)
RGB Camera (Basler ace)	11.530MB/s	1.001TB/day	0.996TB/day
3D LiDAR (VLP16)	12.788MB/s	1.105TB/day	1.131TB/day
3D LiDAR (Pandar64)	60.009MB/s	5.185TB/day	5.176TB/day
GNSS (Novatel OEM7)	19.624KB/s	1.696GB/day	1.702GB/day
CAN	217.833KB/s	18.821GB/day	-
Total	166.915MB/s	14.424TB/day	14.432TB/day

Interestingly, even with such a complex sensor stack, commercially available DDR4 RAM, with a read/write speed of up to 26 GB/s, is sufficient to store, read, and write all sensor data. On the data storage side, if we measure the storage size in 24 hours per day, the autonomous vehicle generates up to 14.424 TB of daily data. According to the latest report from the American Automobile Association(AAA), the average driving time in the US is 60.2 mins/day. So, the storage size will fluctuate from one TB to hundreds of TB as driving time, sensor numbers, and sensor data quality increase. This prompts us to rethink what data should be stored and how data is stored.

6 Discussion

In this paper, we presented some preliminary findings on a comprehensive analysis of storage requirements for autonomous vehicles. At the heart of our work, we built a mathematical model for on-board sensors and evaluated

[3]Short-terms used in the figure: Advanced driver-assistance system (ADAS); Autonomous Driving (AD); Robotic Operating System (ROS); Global Positioning System (GPS)
[4]Autoware: An open-source software project for autonomous driving.

them with an actual autonomous vehicle. This combination of theoretical modeling and empirical data analysis will further assist in the development of storage system designs for autonomous vehicles. However, a notable concern is the limited experimental data available. With a dataset encompassing merely 13 test groups for each sensor, expanding the breadth of our experiments could facilitate a more accurate determination of standard deviations and variances, offering a deeper insight into data size fluctuations. To this end, we also plan to incorporate additional radar sensors into our research platform to assess the accuracy of the radar mathematical model.

Furthermore, our study has mainly focused on analyzing the output characteristics of the data collected. Understanding the dynamics of data retrieval frequency and identifying the algorithms or applications that predominantly access this data is crucial for tailoring storage solutions to meet user demands. Consequently, future endeavors will aim to decode these data usage patterns, thereby refining the architecture of data storage schedulers for autonomous vehicles.

Another challenge highlighted by our study involves determining the optimal duration for data retention and establishing criteria for data retrieval within AV storage systems. Our present model operates under the assumption of a one-day data retention period. As we advance, our research intends to leverage an AV benchmark [14] to explore storage scheduling methodologies. This exploration will include identifying efficient mechanisms for the loading and unloading of data, which could encompass both wired and cloud-based approaches. Regarding the choice of storage mediums, it's also crucial to broaden our consideration beyond Solid State Drives (SSDs). Hard Disk Drives (HDDs), for example, may offer a more suitable option for storing long-term data, which is vital for running failure detection models. Addressing these strategic considerations is imperative for advancing toward more effective storage solutions designed for autonomous vehicles.

Acknowledgments

This work is funded by the US National Science Foundation (NSF) under Grant CNS 2231523. Special thanks are extended to Western Digital for their support. The authors also wish to express their gratitude to all individuals and organizations that contributed directly or indirectly to the success of this work.

References

[1] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. 2021. Cyberattacks and countermeasures for in-vehicle networks. *ACM computing surveys (CSUR)* 54, 1 (2021), 1–37. https://doi.org/10.1145/3431233

[2] Mrinal R Bachute and Javed M Subhedar. 2021. Autonomous driving architectures: insights of machine learning and deep learning algorithms. *Machine Learning with Applications* 6 (December 2021), 100164. https://doi.org/10.1016/j.mlwa.2021.100164

[3] Hannah M Boland, Morgan I Burgett, Aaron J Etienne, and Robert M Stwalley III. 2021. An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment. *Technology in Agriculture* (July 2021). https://doi.org/10.5772/intechopen.98444

[4] Western Digital. 2020. *electronicsforu.com*. Technical Report. https://electronicsforu.com/wp-contents/uploads/2020/04/Automotive-Whitepaper.pdf

[5] Asif Faisal, Md Kamruzzaman, Tan Yigitcanlar, and Graham Currie. 2019. Understanding autonomous vehicles. *Journal of transport and land use* 12, 1 (2019), 45–72. https://www.jstor.org/stable/26911258

[6] Steve Corrigan HPL. 2002. *Introduction to the controller area network (CAN)*. Technical Report. 1–17 pages. http://masters.donntu.ru/2005/fvti/trofunenko/library/sloa101.pdf

[7] Karl Henrik Johansson, Martin Törngren, and Lars Nielsen. 2005. Vehicle applications of controller area network. *Handbook of networked and embedded control systems* (2005), 741–765. https://doi.org/10.1007/0-8176-4404-0_32

[8] Insup Kim, Ganggyu Lee, Seyoung Lee, and Wonsuk Choi. 2022. Cybersecurity and Capacity Requirement for Data Storage of Autonomous Driving System. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, London, UK, 1–7. https://doi.org/10.1109/VTC2022-Fall57202.2022.10012699

[9] Insup Kim, Ganggyu Lee, Seyoung Lee, and Wonsuk Choi. 2022. Data Storage System Requirement for Autonomous Vehicle. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*. IEEE, Jeju, Korea, 45–49. https://doi.org/10.23919/ICCAS55662.2022.10003785

[10] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. 2018. Sensors and Sensor Fusion in Autonomous Vehicles. In *2018 26th Telecommunications Forum (TELFOR)*. 2018 26th Telecommunications Forum (TELFOR), 420–425. https://doi.org/10.1109/TELFOR.2018.8612054

[11] Richard B Langley. 1995. NMEA 0183: A GPS receiver interface standard. *GPS world* 6, 7 (1995), 54–57. https://gge.ext.unb.ca/Resources/gpsworld.july95.pdf

[12] Piergiuseppe Mallozzi, Patrizio Pelliccione, Alessia Knauss, Christian Berger, and Nassar Mohammadiha. 2019. Autonomous vehicles: state of the art, future trends, and challenges. *Automotive systems and software engineering: State of the art and future trends* (July 2019), 347–367. https://doi.org/10.1007/978-3-030-12157-0_16

[13] Ruijun Wang, Liangkai Liu, and Weisong Shi. 2020. HydraSpace: Computational data storage for autonomous vehicles. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, Atlanta, GA, USA, 70–77. https://doi.org/10.1109/CIC50333.2020.00033

[14] Yifan Wang, Shaoshan Liu, Xiaopei Wu, and Weisong Shi. 2018. CAVBench: A benchmark suite for connected and autonomous vehicles. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, Seattle, WA, USA, 30–42. https://doi.org/10.1109/SEC.2018.00010

[15] Zhangjing Wang, Yu Wu, and Qingqing Niu. 2020. Multi-Sensor Fusion in Automated Driving: A Survey. *IEEE Access* 8 (2020), 2847–2868. https://doi.org/10.1109/ACCESS.2019.2962554

[16] Bao-Fu Wu, Ren Zhong, Yuxin Wang, Jian Wan, Ji-Lin Zhang, and Weisong Shi. 2024. Vpi: Vehicle programming interface for vehicle computing. *Journal of Computer Science and Technology* 39, 1 (2024), 22–44. https://doi.org/10.1007/s11390-024-4035-2

[17] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. 2021. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors* 21, 6 (2021), 2140. https://doi.org/10.3390/s21062140

[18] Weiying Zeng, Mohammed AS Khalid, and Sazzadur Chowdhury. 2016. In-vehicle networks outlook: Achievements and challenges. *IEEE Communications Surveys & Tutorials* 18, 3 (January 2016), 1552–1571. https://doi.org/10.1109/COMST.2016.2521642

Appendix[5]

Appendix Table 1: CAN BUS data

scenarios	Duration(s)	Size(MB)	# Message	Data Rate(KB/s)	Message Rate(#/s)
Rural 15 MPH	282	61.4	443849	217.730	1573.933
	264	57.5	415614	217.803	1574.295
	274	59.6	431310	217.518	1574.124
Rural 25 MPH	199	43.3	313250	217.588	1574.121
	181	39.4	284570	217.680	1572.210
	184	40.2	290496	218.478	1578.783
City	314	68.6	495588	218.471	1578.306
	395	86.0	621594	217.722	1573.656
	521	113.6	820624	218.042	1575.094
	369	80.4	581395	217.886	1575.596
	246	53.6	387164	217.886	1573.837
Highway	229	50.0	361414	218.341	1578.227
	230	50.0	361728	217.391	1572.730

Appendix Table 2: RGB camera (Balser ace) data

scenarios	Duration(s)	Size(GiB)	# Message	Data Rate(MB/s)	Message Rate(#/s)
Rural 15 MPH	253.389	2.7	1268	11.441	5.004
	220.997	2.4	1106	11.661	5.005
	246.697	2.7	1235	11.752	5.006
Rural 25 MPH	163.593	1.8	819	11.814	5.006
	159.794	1.7	800	11.423	5.006
	184.396	2.0	923	11.646	5.006
City	333.797	3.6	1670	11.580	5.003
	379.797	4.1	1900	11.591	5.003
	516.400	5.5	2583	11.436	5.002
	364.395	3.9	1823	11.492	5.003
	241.597	2.6	1209	11.555	5.004
Highway	237.794	2.6	1190	11.740	5.004
	278.896	3.0	1396	11.550	5.005

[5]In all appendix tables, 'Duration(s)', 'Size(xx)', and '#Message' represent the original numbers from rosbag information. The calculated results for 'Data Rate(xx)' and 'Message Rate(#/s)' are displayed with precision up to three decimal places.

Appendix Table 3: 3D LiDAR (VLP16) data

scenarios	Duration(s)	Size(GiB)	# Message	Data Rate(MB/s)	Message Rate(#/s)
Rural 15 MPH	258.462	3.1	5126	12.878	19.832
	218.769	2.6	4338	12.761	19.829
	247.106	2.9	4901	12.601	19.834
Rural 25 MPH	161.174	1.9	3197	12.658	19.836
	157.493	1.9	3124	12.954	19.836
	184.787	2.2	3665	12.784	19.834
City	334.296	4.0	6630	12.847	19.833
	387.76	4.6	7690	12.737	19.832
	525.977	6.2	10431	12.656	19.832
	373.163	4.4	7401	12.661	19.833
	249.781	3.0	4954	12.896	19.833
Highway	224.974	2.7	4462	12.886	19.833
	249.125	3.0	4941	12.930	19.833

Appendix Table 4: 3D LiDAR (Hesai Pandar64) data

scenarios	Duration(s)	Size(GiB)	# Message	Data Rate(MB/s)	Message Rate(#/s)
Rural 15 MPH	225.987	12.6	2261	59.867	10.005
	238.188	13.3	2383	59.956	10.005
	230.892	12.9	2310	59.990	10.005
Rural 25 MPH	207.305	11.6	2074	60.083	10.005
	149.995	8.4	1501	60.132	10.007
	209.983	11.7	2101	59.827	10.006
City	334.895	18.7	3350	59.956	10.003
	384.288	21.5	3844	60.073	10.003
	521.770	29.1	5217	59.884	9.999
	368.695	20.6	3688	59.993	10.003
	245.958	13.8	2463	60.245	10.014
Highway	273.202	15.3	2733	60.132	10.004
	245.462	13.7	2456	59.929	10.006

Appendix Table 5: GNSS (Novatel OEM7) data

scenarios	Duration(s)	Size(MiB)	# Message	Data Rate(KB/s)	Message Rate(#/s)
Rural 15 MPH	262.498	4.9	13126	19.574	50.004
	220.780	4.1	11036	19.473	49.986
	246.579	4.6	12330	19.561	50.004
Rural 25 MPH	166.421	3.1	8322	19.532	50.006
	159.301	3.0	7966	19.747	50.006
	184.339	3.5	9218	19.909	50.006
City	334.198	6.3	16711	19.766	50.003
	390.617	7.3	19532	19.596	50.003
	535.417	10.0	26773	19.584	50.004
	377.198	7.1	18861	19.737	50.003
	252.798	4.7	12641	19.495	50.004
Highway	227.895	4.3	11394	19.785	49.997
	140.241	2.6	7013	19.440	50.007