Research Summary: Enhancing Localization, Selection, and Processing of Data in Vehicular Cyber-Physical Systems

Citation for the original published paper (version of record):
Havers, B., Papatriantafilou, M., Gulisano, V. (2024). Research Summary: Enhancing Localization, Selection, and Processing of Data in Vehicular
Cyber-Physical Systems. Proceedings of the 2024 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems: 91-95. http://dx.doi.org/10.1145/3663338.3663680

(article starts on next page)

Research Summary: Enhancing Localization, Selection, and Processing of Data in Vehicular Cyber-Physical Systems

Bastian Havers
Chalmers University & Volvo Cars
Gothenburg, Sweden
havers@chalmers.se

Marina Papatriantafilou
Chalmers University
Gothenburg, Sweden
ptrianta@chalmers.se

Vincenzo Gulisano
Chalmers University
Gothenburg, Sweden
vincenzo.gulisano@chalmers.se

ABSTRACT

Increasing amounts of data are sensed at the edge of the Edge-to-Cloud (E2C) continuum, enabling the rapid development of data-driven applications based on, e.g., Machine Learning. This is especially true for Vehicular Cyber-Physical Systems (VCPSs), networks of connected vehicles equipped with high-bandwidth sensors, where Big Data originating on the vehicles is crucial for the advancement of autonomous drive, developing new cars, and more. Limited bandwidth and storage mean that moving this vehicular Big Data from the edge to central processing increasingly poses challenges. In this work, we present our research on how to alleviate these through efficiently *localizing* data on the edge, *selecting* relevant data in a data stream, and *distributing* the processing of data in a VCPS.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **Information systems** → **Online analytical processing engines**.

KEYWORDS

Edge-To-Cloud, Stream Processing, Distributed Analysis

ACM Reference Format:
Bastian Havers, Marina Papatriantafilou, and Vincenzo Gulisano. 2024. Research Summary: Enhancing Localization, Selection, and Processing of Data in Vehicular Cyber-Physical Systems. In *Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems (ApPLIED'24), June 17, 2024, Nantes, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3663338.3663680

1 INTRODUCTION

The Edge-to-Cloud (E2C) continuum, encompassing a hierarchy of networked devices from tiny sensors to powerful cloud servers, is experiencing explosive data growth [26]. The ever-increasing output of sensors and diverse devices causes the overall data sphere to double roughly every three years, from one zettabyte (1 ZB = 10^9 terabytes) in 2010 to 200 ZB by 2025 [36]. The transformative power of this *Big Data* is evident across diverse domains: financial forecasting through the analysis of market fluctuations [4], data monitoring and validation in smart grids [31], quality assurance in additive manufacturing [17], predictive maintenance exemplified by anticipating brake pad wear in automobiles [21], or the advancement of autonomous mobility [12, 40]. The sheer size of modern datasets serves as a critical driver for the emergence of powerful Machine Learning algorithms, as those employed in image recognition [23] and the advances seen in Large Language Models exemplified by GPT-3 [5]. These advancements collectively contribute to the ongoing saga of digital automation.

Challenges. Enabling these developments with Big Data requires extensive processing that has sparked the development of a corresponding processing ecosystem. This ecosystem has to constantly develop, as exemplified by the following aspects:

(1) Mobile data speeds increase on average only linearly [35], as wide-spread adoption of next-gen communication technology is costly and slow [22]. As data is increasingly sensed on the resource-constrained edge of the E2C continuum, moving Big Data processing workloads to *the cloud* is proposed as a solution to the increasing computational requirements [26]. However, this increasingly clashes with (mobile) network limitations. Processing data "close to the edge" emerges as a promising solution.

(2) Data production in 2020 exceeded global storage capacity by a factor of ten [36]. Also, as data output grew 30× between 2010 and 2020, storage costs only fell to a third [25]. As storing all raw data becomes untenable (particularly for edge devices), data should be aggregated, filtered, and prioritized *before* it is potentially stored.

(3) The interplay of exponential data growth, limited mobile speeds, and escalating storage costs, particularly at the edge, necessitates a further paradigm shift: not all data merits equal treatment. While definitive pre-analysis identification of valuable data points remains challenging, implementing intelligent early-stage filtering and selection strategies holds immense potential [30]. This relieves strain on communication networks, storage, and processing resources by prioritizing critical information for transmission, storage, and in-depth analysis. Less relevant or redundant data can be filtered out early, reducing costs and resource burden.

While these discussions are relevant for many systems characterized under the E2C continuum, they are especially significant for *Vehicular Cyber-Physical Systems (VCPSs).* These are connected fleets of cars, equipped with numerous sensors such as cameras, LiDARS, or GPS receivers, on-board computers with limited capability, and mobile broadband. At the center of a VCPS sits a powerful central or cloud server that coordinates data analysis in this system.

Contributions. Relating to the previous paragraphs **(1-3)**, we present in this research summary approaches and novel techniques that address **(1)** *Where* and **(2)** *How* to process **(3)** *Which* data in

ApPLIED'24, June 17, 2024, Nantes, France

ACM ISBN 979-8-4007-0670-7/24/06.
https://doi.org/10.1145/3663338.3663680

systems such as VCPSs. Specifically, our work aims at distributing the computation in the E2C continuum along a spectrum of edge involvement, from at-edge summaries [19] and data discovery [11] to full analysis involvement [20]. Furthermore, we leverage *Stream Processing* to enable full streaming edge-to-cloud data analysis [19] and the efficient selection of relevant data [30].

2 STREAM PROCESSING PIPELINES UTILIZING ONLINE LOSSY COMPRESSION

Traditional workflows in VCPSs involve vehicles gathering data, wireless transmission to a central server, and subsequent data analysis. As discussed in § 1, network limitations mean that transmitting raw data incurs several drawbacks. With the *Driven* framework [19] (overview sketch in Figure 1), we propose to distribute work and reduce data volumes through compact data summaries generated on-vehicle via lossy compression, and employ Stream Processing throughout an analysis pipeline for minimal latencies.

Online Lossy Compression Through Piecewise-Linear Approximations (PLA). PLA reduces time series commonly produced by edge devices to a set of line segments that encode a segment's data points using only segment length, slope, and y-intercept. This trades off accuracy (from approximating sets of points with straight lines) with data size. Recent PLA research emphasizes its feasibility in streaming contexts, prioritizing low resource consumption and minimal latency to enable on- or near-sensor deployment [9, 15, 39]. Our implementation of streaming-based PLA in [18, 19] is adapted for use in vehicular data analysis scenarios through the following: (i) it has a per-point *bound* for the approximation error (e.g., for a bound of 1m, no data point reconstructed after compression is off from its raw data counterpart by more than 1m), unlike other works in the field [15] that bound the maximum deviation only per segment. Such fine-grained control allows analysts to set bounds in accordance with their analysis needs. (ii) Our implementation processes the time dimension of time series data independent from other data dimensions, allowing independent compression and decompression of all data dimensions. This improves on previous techniques that can introduce compounding errors from decompressing a data dimension using a decompressed time signal [9].

Continuous Data Clustering. For a fully streaming pipeline, also the eventual data analysis must be performed in an online manner. In [19], we take the problem of clustering as example analysis and present a fully online version, extending previous work from [27]. Our approach leverages the fact that some sensors deliver data partially ordered among at least one dimension of interest. For a rotating LiDAR mounted on a vehicle, for example, measurements generated close in time have a bounded distance. This bounds

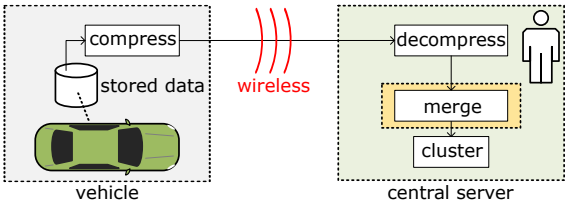

Figure 1: Overview of *Driven* for clustering LiDAR data.

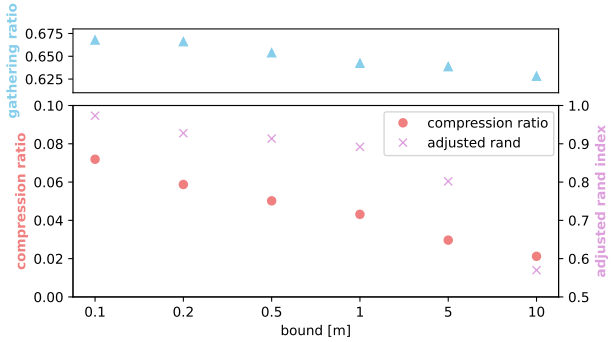

Figure 2: Performance of *Driven* for the LiDAR use case (baseline: streaming pipeline without compression).

the search space for neighboring points, allowing processing in $O(n)$, where traditional clustering techniques require $O(n \log n)$ computations [38] (n = # of points to cluster).

Evaluation Results and Conclusion. Using an Odroid XU-3 [19] as stand-in for a vehicle's available processing power, we evaluate various use cases from traffic density analysis to LiDAR point cloud clustering. In all these, the Driven framework leads to significant data reductions and speedups. We present the LiDAR results in Figure 2. Here, the baseline uses no compression and streams raw data directly from vehicle to central server. We evaluate the gathering time ratio (time of data transmission incl. potential compression and decompression), compression ratio, and adjusted rand index (ARI, a metric for clustering similarity [37]), comparing the corresponding values for various PLA bounds (which regulate the compression strength) with the baseline. For a bound of 1m, the ARI remains above 90%, while gathering time is reduced by 35% (indicating low impact of the [de-]compression work), and the compression ratio reaches below 0.1. This yields a balance of fast transmission of starkly reduced data volumes at limited analysis accuracy loss.

In summary, our work shows that tunable lossy compression can lead to significant speed-ups of the gathering phase and considerate data savings, while bounding the loss in analysis accuracy, enabling analysts to choose the maximum compression that still yields acceptable eventual accuracy.

3 DATA LOCALIZATION AT THE EDGE

While on-vehicle compression or summarization as in the previous section reduces the amount of transmittable data, it does not address the issue of *global aggregation*: depending on the analysis, a central server may gather unfit or excessive amounts of data when querying the fleet too broadly. This results in unnecessary strain on the network, the central server and the participating vehicles. Employing the edge processing paradigm, we suggest in [10, 11] to leverage *data localization*: to (i) push the task of filtering for relevant data to the vehicles, and (ii) employ orchestrated data aggregation that avoids excessive transmission of data and minimizes the overall workload on the fleet during the data localization stage.

In our system model, we assume a central server that spreads *requests* for checking compliance of local data with an analysis-dependent condition to the vehicles (e.g., "Have you driven close

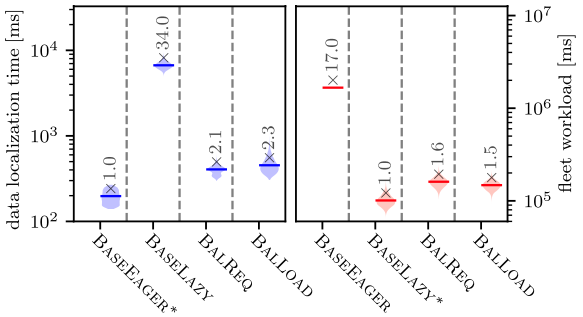

Figure 3: Evaluation of data localization algorithms. Factors "×" given with respect to starred algorithm.

to a parking spot today?"). These reply with a binary answer, and the procedure proceeds in rounds until a minimum number of affirmative answers is collected. While each vehicle holds an unknown amount of data and thus answering time and required local workload are highly variable between vehicles, we attempt to optimize for both the duration of the data localization phase as well as for the workload induced on the complete fleet.

Data Localization Algorithms. The work's core lies in the following data localization algorithms that describe the orchestration of request spreading: `BaseEager` sends a request to the entire vehicle fleet, resulting in the fastest possible conclusion of the localization phase, but inducing maximal workload. `BaseLazy` asks vehicles in a 1-by-1 fashion, in best case reducing the induced workload to a minimum, while requiring in worst-case maximum time. Crucially, we introduce `BalanceRequests` that navigates between the aforementioned algorithms by including on-the-fly heuristics about the share of affirmative answers from vehicles to optimize when to send more requests to how many vehicles, proceeding in overlapping rounds that avoid excessive wait times for vehicles to answer. Allowing for multiple concurrent requests for different types of data, individual vehicles may be involved in processing several requests simultaneously. Addressing this, `BalanceLoad` extends the `BalanceRequest` with the notion of fairness to balance the induced load over the fleet such that no vehicles are excessively tasked.

Evaluation Results and Conclusion. We extensively evaluate our novel data localization algorithms for different fleet scenarios, static and dynamic (in which vehicles can join and leave the fleet even during the processing of requests). Furthermore, we model realistic 5G communication latencies for the transmission of requests and answers. Here, we show the results for the proprietary *Volvo* dataset, where we evaluate 15 simultaneous requests that cover combinations of GPS and drive mode data, with a *static* fleet size of 3462 cars. Figure 3 shows several of the core conclusions of this evaluation. We see that for the data localization time (left plot), the novel balanced algorithms achieve times very close to the respective baseline `BaseEager`. Regarding the total workload (right plot), similarly, the balanced results are close to the baseline `BaseLazy`; indicating that the balanced algorithms balance the trade-off space to identify sweet spots that approach the optimal baseline performance.

In summary, these novel data localization algorithms can help to efficiently and swiftly localize data on the edge, overall reducing the necessity of excessive global data gathering.

4 STATEFUL IN-STREAM DATA SELECTION

While in many contexts it is useful to localize data across several data sources (in the previous section, cars), oftentimes localizing or selecting data *within* a data source is required. In VCPSs, such data sources commonly exist as real-time sources. We model these using Stream Processing, where a canonical, *stateless Filter* function exists (e.g., in frameworks such as [3, 6]). This *Filter* regards its current input, forwarding the latter only if it passes an internal condition. *Stateful* selection, spanning the time dimension of data, is a more powerful alternative that can allow for the definition of complex conditions for selection of data, e.g., *"find all video frames of bicyclists stopping in front of a vehicle for 30s"*. In [30], we show how such stateful selection can be achieved via *Forward Provenance*.

Forward Provenance (FP). FP in Stream Processing builds on the well-established concept of *Backward Provenance* (BP) [14, 29]. BP connects the outputs (the result) of a query to the inputs causing the result. This causal relationship is used to both debug an application, or to understand how a result came to be [13]. An example, for two parallel Stream Processing queries processing a stream of position reports from two vehicles, is shown in Figure 4 b): a graph, connecting the results a_1^i, a_2^1 from each query Q_1, Q_2 to the input data t_l^m that caused it. As the graph shows, input data can occur multiple times, and it is not possible to tell when some input data will not contribute to any future results anymore and can be seen as finalized (from the processing standpoint).

We amend these issues in [30] as the first to introduce FP and our corresponding framework *Ananke*, which produces a single graph (cf. Figure 4 c)) of dependencies: every node, corresponding to an input or output, is unique, and special labels (here represented as green checkmarks) indicate once data is finalized. To achieve this, Ananke ingests a stream of BP annotations and time flow information (*watermarks*, cf. [1, 16]) from every considered query.

Using Ananke, analysts can write complex queries in such a way that relevant data contributes to query results, and leverage FP to obtain this relevant data without duplicates and with additional temporal information useful for further processing.

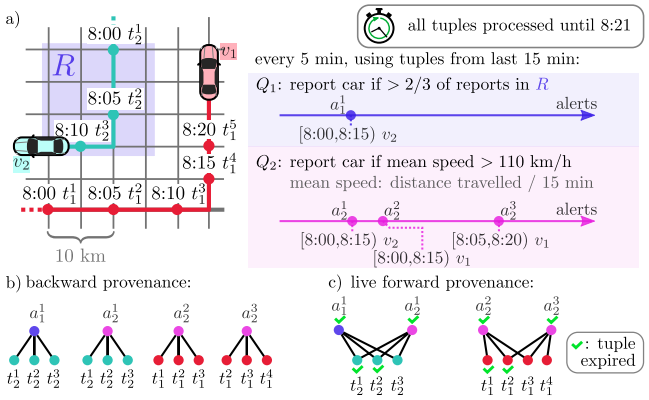

Figure 4: a) Two Stream Processing queries, Q_1 and Q_2, producing alerts as output; b) Backward Provenance graph of outputs and inputs; c) Forward Provenance graph.

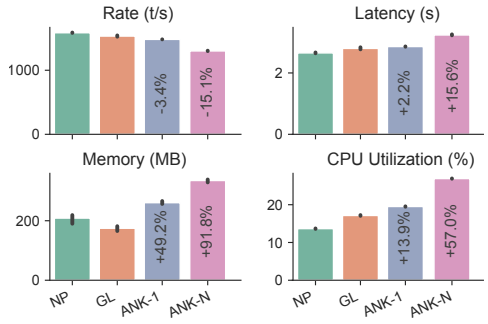

Figure 5: Overheads of Forward Provenance (Ananke - ANK-1/N) vs. BP (GL [29]) for a vehicular tracking query running on an Odroid XU-3. NP = no provenance (original query).

Evaluation Results and Conclusions. We present an optimized single-operator (ANK-1) and a parallelizable implementation (ANK-N) of Ananke, and evaluate these over a range of use cases and queries. The results show that ANK-1 incurs small overheads to queries, close to those of sheer BP, while ANK-N can sustain high data rates on manycore systems. An excerpt of our evaluation is shown in Figure 5 for a query processing vehicle GPS traces, running on a low-powered Odroid XU-3 emulating available computational headroom on modern vehicles. As shown, the overheads of especially ANK-1 are small, with < 4% impact on processing rate and latency compared to BP (GL = GeneaLog [29], the SoA).

In conclusion, via FP at low overheads, Ananke enables the efficient selection of data from Stream Processing sources.

5 VALIDATION OF DISTRIBUTED DATA PROCESSING ALGORITHMS FOR VCPS

The works presented so far in this paper leverage the capabilities of distributed computing and Stream Processing to enhance the gathering and discovery of data. The next step in data processing in VCPS is the analysis or learning from the data, which for modern applications often involves Machine Learning (ML) methods [2, 28, 34]. This, as the preceding steps, may also be distributed over a fleet of vehicles. Recent approaches for distributed ML in VCPSs have included Federated Learning (FL) [8] (centrally-coordinated) and Gossip Learning (GL) [33] (peer-to-peer and centerless), that can lead to remarkable reductions in data transmission by exchanging summaries and leveraging direct vehicle-to-vehicle communication (*V2X*, cf. *IEEE 802.11p*). In [20], we are the first to deeply explore the challenges of creating a learning simulator for evaluating such distributed learning algorithms end-to-end in VCPSs, by identifying among others the requirements of

(1) *Realistic ML support.* Handling modern ML frameworks (e.g., Keras [7] or PyTorch [32]) and realistic data distributions
(2) *Realistic communication model.* Simulate cellular and direct vehicle-to-vehicle connections
(3) *Flexible distributed algorithm support.* Allowing flexible implementations of distributed learning algorithms to allow for easy experimentation and iteration

From (1)-(3), we build *Roadrunner*, a prototype simulator, and demonstrate its capabilities using a novel distributed ML algorithm.

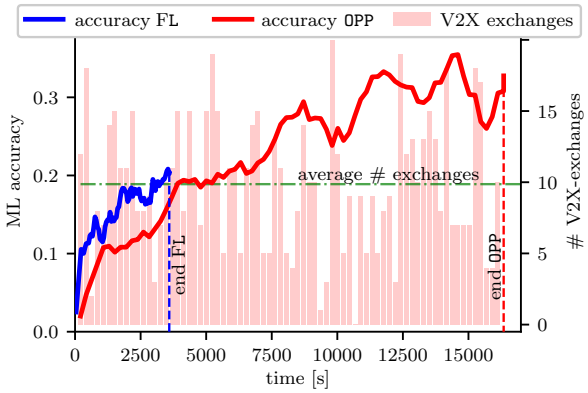

Figure 6: Accuracy of OPP using our prototype Roadrunner learning simulator.

Evaluation Results and Conclusion. With OPP, we present a novel algorithm that extends FL with elements from GL, that is difficult to evaluate appropriately with previously existing VCPS simulation tools (such as Sumo [24]). For a fixed communication budget, OPP allows many more ML iterations than FL, as shown in Figure 6, reaching a higher final accuracy (as measure for the learning goal).

As this example shows, tools like Roadrunner that are based on our specifications can help in the discovery and tuning of novel distributed learning algorithms in VCPSs.

6 CONCLUSIONS

We presented novel approaches for processing data in VCPSs more efficiently by focussing on reducing data volumes and involving the vehicles (the system's edge) more actively than traditional workflows. Specifically, we presented Driven, a framework that incorporates a novel lossy compression algorithm for full streaming edge-to-cloud pipelines that demonstrably reduce transmitted data volumes and speed up the data gathering phase; a family of data localization algorithms that enable more targeted involvement of edge nodes in data transmission while balancing localization speed and workload; Ananke, the first forward provenance framework, enabling low-overhead selection of relevant data in Stream Processing; and the specifications for and an implementation of the first comprehensive simulation tool for distributed learning in VCPSs.

ACKNOWLEDGMENTS

Work supported by the Swedish Government Agency for Innovation Systems VINNOVA, prjs. "Onboard/Offboard Distributed Data Analytics (OODIDA)" (DNR 2016-04260) and "AutoSPADA" (DNR 2019-05884) in the program FFI: Strategic Vehicle Research and Innovation, the Marie Skłodowska-Curie Doctoral Network prj. RELAX-DN, funded by the European Union under Horizon Europe 2021-2027 Framework Programme Grant Agreement nr. 101072456, the Chalmers AoA frameworks Energy and Production, WPs INDEED, and "Scalability, Big Data and AI", respectively, the Swedish Energy Agency (SESBC) prj. TANDEM, the VR grant "EPITOME" (VR 2021-05424) and the Wallenberg AI, Autonomous Systems and Software Program and Wallenberg Initiative Materials for Sustainability prj. STRATIFIER.

REFERENCES

[1] Tyler Akidau, Robert Bradshaw, Craig Chambers, Slava Chernyak, Rafael J Fer Andez-Moctezuma, Reuven Lax, Sam Mcveety, Daniel Mills, Frances Perry, Eric Schmidt, and Sam Whittle Google. 2015. The Dataflow Model: A Practical Approach to Balancing Correctness, Latency, and Cost in Massive-Scale, Unbounded, Out-of-Order Data Processing. *VLDB* 8, 12 (2015), 1792–1803. https://doi.org/10.14778/2824032.2824076

[2] Furqan Alam, Rashid Mehmood, and Iyad Katib. 2017. D2TFRS: An object recognition method for autonomous vehicles based on RGB and spatial values of pixels. In *International Conference on Smart Cities, Infrastructure, Technologies and Applications*. Springer, 155–168.

[3] Apache. 2020. Storm. Retrieved November 12, 2020 from http://storm.apache.org/

[4] Girija V Attigeri, Manohara Pai MM, Radhika M Pai, and Aparna Nayak. 2015. Stock market prediction: A big data approach. In *TENCON 2015-2015 IEEE Region 10 Conference*. IEEE, 1–5.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. 2015. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 36, 4 (2015).

[7] François Chollet et al. 2015. Keras. https://keras.io.

[8] Sonal Doomra, Naman Kohli, and Shounak Athavale. 2020. Turn Signal Prediction: A Federated Learning Case Study. *arXiv preprint arXiv:2012.12401* (2020).

[9] Romaric Duvignau, Vincenzo Gulisano, Marina Papatriantafilou, and Vladimir Savic. 2019. Streaming Piecewise Linear Approximation for Efficient Data Management in Edge Computing. In *34th ACM/SIGAPP Symposium On Applied Computing SAC'19*. 593–596.

[10] Romaric Duvignau, Bastian Havers, Vincenzo Gulisano, and Marina Papatriantafilou. 2019. Querying large vehicular networks: How to balance on-board workload and queries response time?. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2604–2611.

[11] Romaric Duvignau, Bastian Havers, Vincenzo Gulisano, and Marina Papatriantafilou. 2021. Time-and Computation-Efficient Data Localization at Vehicular Networks' Edge. *IEEE Access* 9 (2021), 137714–137732.

[12] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 22, 3 (2020), 1341–1360.

[13] Boris Glavic et al. 2021. Data provenance. *Foundations and Trends® in Databases* 9, 3-4 (2021), 209–441.

[14] Boris Glavic, Kyumars Sheykh Esmaili, Peter Michael Fischer, and Nesime Tatbul. 2013. Ariadne: Managing Fine-Grained Provenance on Data Streams. In *Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems (DEBS '13)*. Association for Computing Machinery, New York, NY, USA, 39–50. https://doi.org/10.1145/2488222.2488256

[15] Florian Grützmacher, Benjamin Beichler, Albert Hein, Thomas Kirste, and Christian Haubelt. 2018. Time and Memory Efficient Online Piecewise Linear Approximation of Sensor Signals. *Sensors* 18, 6 (2018), 1672.

[16] Vincenzo Gulisano, Dimitris Palyvos-Giannas, Bastian Havers, and Marina Papatriantafilou. 2020. The role of event-time order in data streaming analysis. In *Proceedings of the 14th ACM International Conference on Distributed and Event-based Systems*. 214–217.

[17] Vincenzo Gulisano, Marina Papatriantafilou, Zhuoer Chen, Eduard Hryha, and Lars Nyborg. 2022. Towards data-driven additive manufacturing processes. In *Proceedings of the 23rd International Middleware Conference Industrial Track*. 43–49.

[18] Bastian Havers, Romaric Duvignau, Hannaneh Najdataei, Vincenzo Gulisano, Ashok Chaitanya Koppisetty, and Marina Papatriantafilou. 2019. Driven: a framework for efficient data retrieval and clustering in vehicular networks. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1850–1861.

[19] Bastian Havers, Romaric Duvignau, Hannaneh Najdataei, Vincenzo Gulisano, Marina Papatriantafilou, and Ashok Chaitanya Koppisetty. 2020. DRIVEN: A framework for efficient Data Retrieval and clustering in Vehicular Networks. *Future Generation Computer Systems* 107 (2020), 1–17.

[20] Bastian Havers, Marina Papatriantafilou, Ashok Koppisetty, and Vincenzo Gulisano. 2022. Proposing a framework for evaluating learning strategies in vehicular CPSs. In *Proceedings of the 23rd International Middleware Conference Industrial Track*. 22–28.

[21] Kenneth M Jensen, Ilmar F Santos, and Harry JP Corstens. 2023. Estimation of brake pad wear and remaining useful life from fused sensor system, statistical data processing, and passenger car longitudinal dynamics. *Wear* (2023), 205220.

[22] Wei Jiang, Bin Han, Mohammad Asif Habibi, and Hans Dieter Schotten. 2021. The road towards 6G: A comprehensive survey. *IEEE Open Journal of the Communications Society* 2 (2021), 334–366.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[24] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. 2018. Microscopic Traffic Simulation using SUMO, In The 21st IEEE International Conference on Intelligent Transportation Systems. *IEEE Intelligent Transportation Systems Conference (ITSC)*. https://elib.dlr.de/124092/

[25] Jojn C. McCallum. 2022. Our World in Data - Historical cost of computer memory and storage. https://ourworldindata.org/grapher/historical-cost-of-computer-memory-and-storage

[26] D. Milojicic. 2020. The Edge-to-Cloud Continuum. *Computer* 53, 11 (2020), 16–25. https://doi.org/10.1109/MC.2020.3007297

[27] Hannaneh Najdataei, Yiannis Nikolakopoulos, Vincenzo Gulisano, and Marina Papatriantafilou. 2018. Continuous and Parallel LiDAR Point-Cloud Clustering. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 671–684.

[28] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli. 2016. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles. *IEEE Transactions on Intelligent Vehicles* 1, 1 (2016), 33–55. https://doi.org/10.1109/TIV.2016.2578706

[29] Dimitris Palyvos-Giannas, Vincenzo Gulisano, and Marina Papatriantafilou. 2018. GeneaLog: Fine-Grained Data Streaming Provenance at the Edge. In *Proceedings of the 19th International Middleware Conference*. ACM, 227–238.

[30] Dimitris Palyvos-Giannas, Bastian Havers, Marina Papatriantafilou, and Vincenzo Gulisano. 2020. Ananke: a streaming framework for live forward provenance. *Proceedings of the VLDB Endowment* 14, 3 (2020), 391–403.

[31] Dimitris Palyvos-Giannas, Katerina Tzompanaki, Marina Papatriantafilou, and Vincenzo Gulisano. 2023. Erebus: Explaining the outputs of data streaming queries. In *Very Large Data Base*, Vol. 16. 230–242.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[33] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. 2021. Federated learning in vehicular networks: opportunities and solutions. *IEEE Network* 35, 2 (2021), 152–159.

[34] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. 2018. Understanding Pedestrian Behavior in Complex Traffic Scenes. *IEEE Transactions on Intelligent Vehicles* 3, 1 (2018), 61–70. https://doi.org/10.1109/TIV.2017.2788193

[35] Petroc Taylor. 2023. Statista - Cellular Network Average Speed in US 2016-2023. https://www.statista.com/statistics/995096/average-cellular-network-speed-in-the-us/

[36] Petroc Taylor. 2023. Statista - Data Growth Worldwide. https://www.statista.com/statistics/871513/worldwide-data-created/

[37] Silke Wagner and Dorothea Wagner. 2007. *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe.

[38] Ingo Wald and Vlastimil Havran. 2006. On building fast kd-trees for ray tracing, and on doing that in O (N log N). In *2006 IEEE Symposium on Interactive Ray Tracing*. IEEE, 61–69.

[39] Qing Xie, Chaoyi Pang, Xiaofang Zhou, Xiangliang Zhang, and Ke Deng. 2014. Maximum error-bounded Piecewise Linear Representation for online stream approximation. *The VLDB Journal* 23, 6 (2014), 915–937.

[40] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. 2020. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access* 8 (2020), 58443–58469.