5.2

Open Traffic Data for Mobility-as-a-Service Applications – Architecture and Challenges

Mathias Schneider[1], Mina Marmpena[2], Haris Zafeiris[2], Ruben Prokscha[1], Seifeddine Saadani[1], Nikolaos Evangeliou[2], George Bravos[2] and Alfred Höß[1]

[1]Ostbayerische Technische Hochschule Amberg-Weiden, Germany
[2]Information Technology for Market Leadership, Greece

Abstract

Data-driven approaches will be a pivotal tool to interpret traffic data and to optimise operations to enable more efficient, individual, public transport. Whereas nowadays data remain a proprietary resource, Finland pioneered an open ecosystem. In this work, we present an architecture to acquire heterogeneous data sources and different data refinement strategies at the edge-level, such as a map-matching approach for inaccurate vehicle GPS traces. Finally, data quality monitoring at the cloud-level is highlighted by introducing and applying an *Errors-to-Data Ratio (EDR)* metric.

Keywords: mobility-as-a-service, edge computing, cloud computing.

5.2.1 Introduction and Background

Mobility-as-a-Service (MaaS) is set to revolutionize urban transport by enabling the orchestration of multiple means of transportation [1]. Thereby, Artificial Intelligence (AI) is a key technology capable of transforming vast volumes of historical and real-time data generated by edge devices, such as vehicles, traffic sensors and cameras to valuable knowledge for MaaS [2]. The utilization of traffic data at scale is a critical factor for training predictive

AI systems. They will power a MaaS operator to successfully manage a fleet of automated driving vehicles for real-time, multi-modal and on-demand transportation [3]. Traffic data are collected from heterogeneous sources, and they come in large volumes, diverse formats, and different rates of speed. To unlock the full potential of the traffic data and make them applicable for training AI algorithms suitable for Intelligent Transportation Systems (ITS), we conceptualised and implemented a complete data management stack that entails processing pipelines applied both at the edge and the cloud. Data processing at the edge involves raw data acquisition, pre-processing for feature engineering and the utilisation of an unstructured database for storage. Data management is resumed in the cloud with pipelines that include structuring, further processing, data quality monitoring and storing in a time-series database.

5.2.2 Data Acquisition

Initiated by the strategic Open Tampere program in 2012, the City of Tampere, Finland, is publishing several data sources under the Open Data licence [4]. Traffic-related data are maintained by the ITS Factory Community [5] and InfoTripla [6]. They comprise information of public transport positioning [7], traffic cameras [8] and loop detectors, measuring traffic amount, congestions, and queue lengths [9].

Data scrapers extract, synchronize, and retain data for each of the sources, as illustrated in Figure 5.2.1. Whenever applicable, existing data formats are kept, including the Service Interface for Real Time Information (SIRI) [10] for public transport vehicle activity, as well as DATEX II [11] for traffic amount measurements. Utilising standardised data formats increase the reusability of subsequent processing components. Raw data is stored in an unstructured MongoDB database. Table 5.2.1 presents database statistics, including the amount of data and sampling rates of the different sources. Thereby, bus traces comprise around 3000 traces of about 150 bus lines. As indicated in the table, traffic cameras capture images with different frequencies.

5.2.2.1 Bus Traces

ITS Factory's public transport Application Programming Interface (API) allows to continuously monitor active vehicles with an overall sampling rate between 0.5 Hz and 1 Hz. Utilizing information of the related bus route,

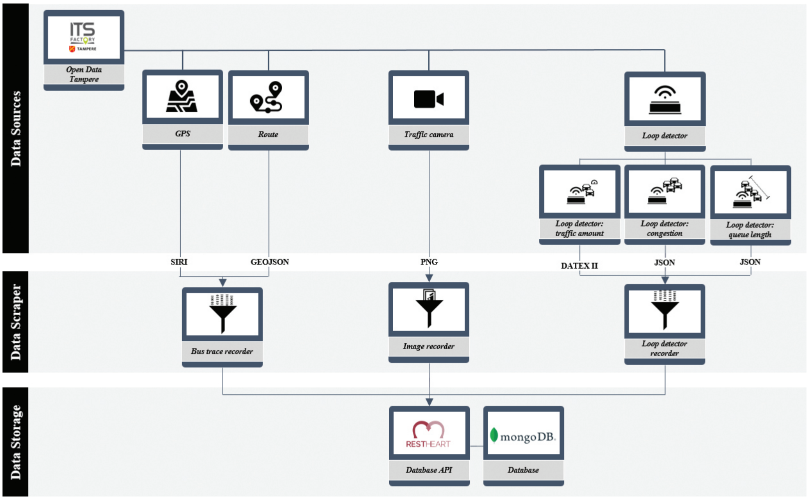


Figure 5.2.1 Open data tampere: design for data acquisition.

Table 5.2.1 Statistics for open traffic data in tampere (2021-5-31). (*) Traffic cameras images are available starting from November 2019 but are not stored in the MongoDB.

	# Samples	**Total size**	**Avg. size**	**Start date**	**Measurements**	**# Sensors**
		[GB]	**[KB]**		**per day/sensor**	
Traffic amount	104,616	65.24	653.88	2020-11-18	∼1,440	∼510
Congestion	97,584	12.18	130.88	2020-11-18	∼1,440	∼480
Queue length	59,994	7.24	126.5	2020-11-18	∼720	∼300
Bus traces	597,251	107.21	188.22	2020-11-17	∼3,000	∼150
Traffic camera	7,163,364	657.46	96.24	2021-01-15*	96/192/1,440	∼140

Global Positioning System (GPS) traces are used to generate durations spent from one bus stop to another. They provide valuable information about the traffic flow in general by deriving metrics such as average speed and stop times. Since the GPS accuracy varies especially in urban regions, the trace is subsequently processed to match the true track.

5.2.2.2 Traffic Cameras

About 140 traffic cameras are available around Tampere. Due to privacy reasons, only images are publicly accessible (maximal one per minute). While certain parts of the image are censored (buildings, etc.), the view of the camera focuses on the street and intersections. Image resolutions vary (e.g.,

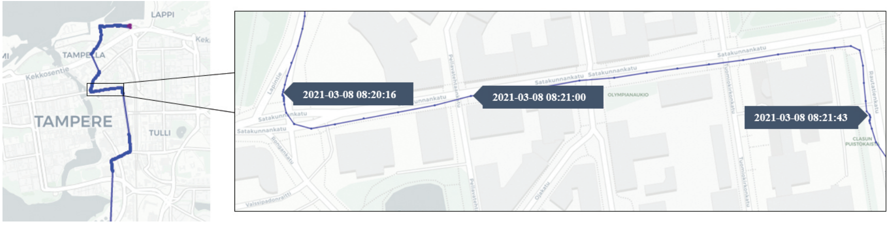


Figure 5.2.2 Bus GPS trace, Line 32 Ranta-Tampella to TAYS Arvo.

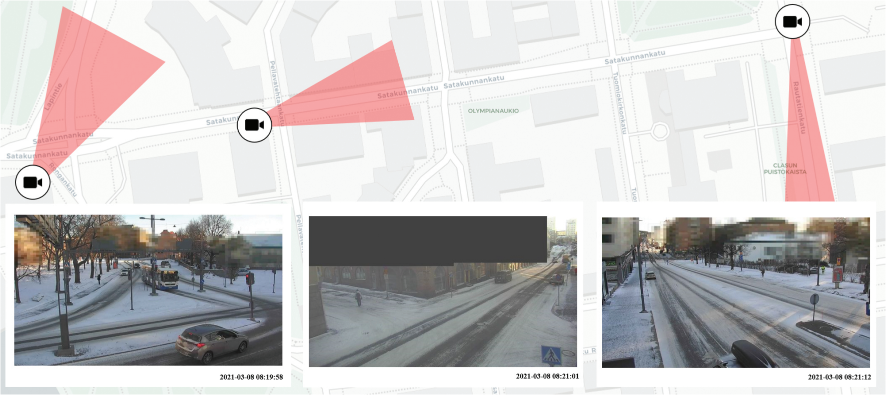
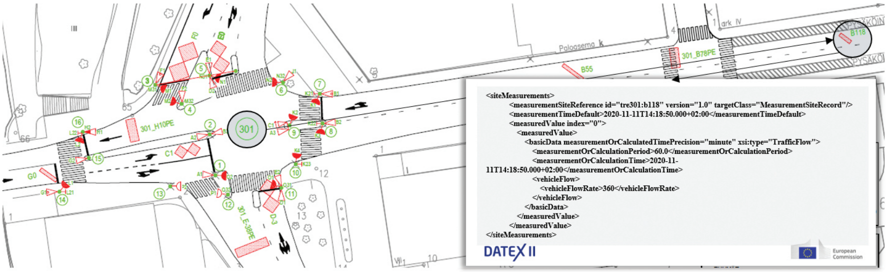


Figure 5.2.3 Traffic cameras and their field of view in Tampere.

640 x 360 px, 704 x 576 px) and objects are largely distorted due to the large perspective. Background objects tend to become very small (less than ten pixels wide) and are often partially occluded. As shown in Figure 5.2.2 and Figure 5.2.3, traces and cameras are roughly synchronized as the passing bus is visible on the images corresponding to its GPS position.

5.2.2.3 Loop Detectors

Tampere provides a vast amount of loop detector measurements, including metrics for traffic amount, congestions, and queue lengths. Data are updated each minute. The spatial information of each sensor is documented separately for each traffic intersection as shown in Figure 5.2.4. Whereas congestions and queue lengths are formatted in JavaScript Object Notation (JSON), traffic amounts are structured using DATEX II standard developed by the European Committee for Standardization (CEN/TC 278).



```
<siteMeasurements>
        <measurementSiteReference id="tre301:b118" version="1.0" targetClass="MeasurementSiteRecord"/>
        <measurementTimeDefault>2020-11-11T14:18:50.000+02:00</measurementTimeDefault>
        <measuredValue index="0">
          <measuredValue>
            <basicData measurementOrCalculatedTimePrecision="minute" xsi:type="TrafficFlow">
              <measurementOrCalculationPeriod>60.0</measurementOrCalculationPeriod>
              <measurementOrCalculationTime>2020-11-
11T14:18:50.000+02:00</measurementOrCalculationTime>
              <vehicleFlow>
                <vehicleFlowRate>360</vehicleFlowRate>
              </vehicleFlow>
            </basicData>
          </measuredValue>
        </measuredValue>
</siteMeasurements>
```

Figure 5.2.4 Loop detectors for traffic amount measurements using DATEX II.

5.2.3 Data Processing at the Edge

Depending on the data source, raw sensor data are not yet suitable for scaling AI-based MaaS applications. This subsection presents data refinement strategies as illustrated in Figure 5.2.5. The architecture comprises object detection for traffic camera images to condense valuable information related to the traffic flow as well as map-matching algorithms to normalize travel times from bus GPS traces. Whereas this kind of pre-processing is nowadays often implemented as a cloud solution, our architecture leverages heterogeneous edge platforms to orchestrate the required computations. Since the edge platforms cannot be physically deployed to the test field in Tampere, a dedicated hardware-in-the-loop (HIL) laboratory cluster is set up for this task.

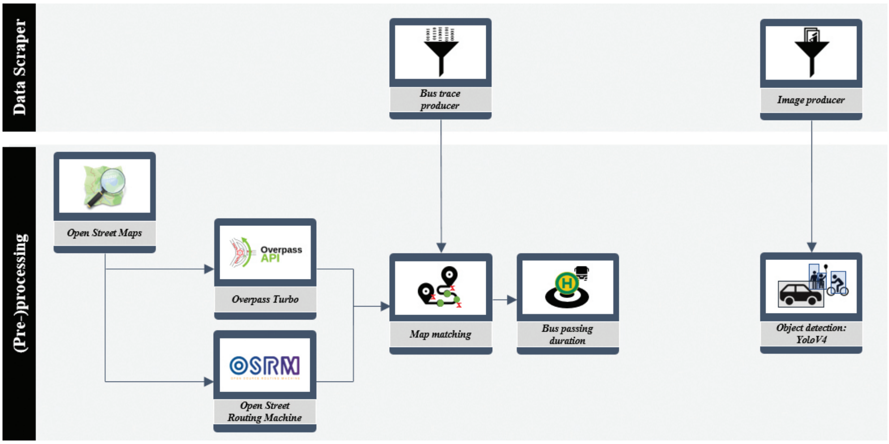


Figure 5.2.5 Architecture for data preparation at the edge.

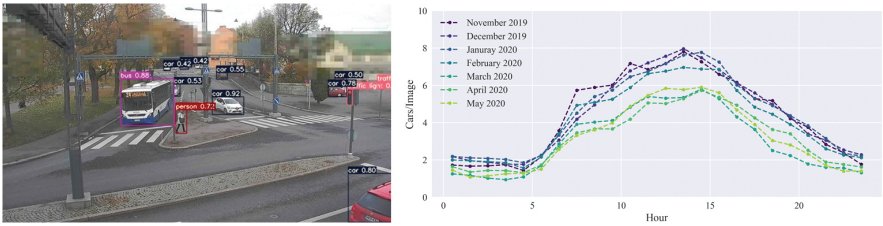


Figure 5.2.6 Traffic object detection (left) and hourly car quantity (right).

5.2.3.1 Object Detection

Object detection is applied to reduce raw, traffic camera image footage to the number of different road objects. Therefore, a YOLOv4 network [12], trained on the MS COCO dataset [13], is leveraged to detect six different types of road users (car, truck, bus, bicycle, motorbike, and person), as well as traffic lights. Although improvements can be introduced to increase the quality of the detection (e.g., excluding parking cars), a first evaluation reveals that it is capable to outline the traffic situation (Figure 5.2.6): whereas the accumulated cars-per-image metric is stable between November 2019 and February 2020, a decline can be observed starting March 2020, likely influenced by the effects of the COVID-19 pandemic.

5.2.3.2 Bus GPS Trace

Bus GPS traces contain a high amount of information about the current traffic state and are utilised to estimate travel times between bus stops and timings for the passenger transfer at a station. Since coordinates are imprecise as shown in Figure 5.2.7, multiple processing steps are conducted to increase the quality of this data source.

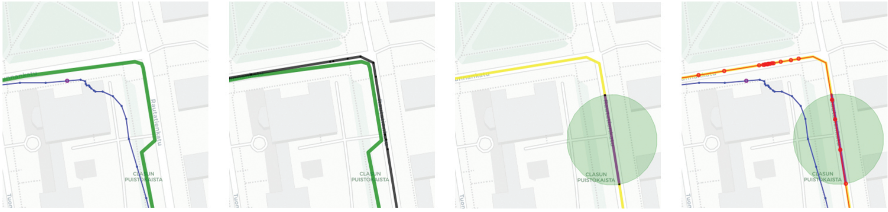


Figure 5.2.7 Refinement of GPS bus traces: (a) Raw GPS [blue] and planned bus route [green] (b) Snapped bus route to OSM road network [black] (c) Partitioned route according to bus stop vicinity [yellow/purple] (d) Map-matching GPS trace [red].

In our approach, the given route provided by the bus API is first snapped to the Open Street Map (OSM) road network. Based on the bus stop positions and a predefined radius, the aligned route is split into segments which allow differentiating a segment between two bus stops and a segment in the vicinity of a stop. GPS coordinates are mapped to this aligned route while applying additional consistency checks, e.g., filtering positions too far away from the route, or physically impossible heading deviations introduced by the inaccuracy of raw GPS. This transformation rectifies timings for each segment and further enables to augment additional OSM-based information, e.g., road segment IDs [14] or amenity characteristics [15].

5.2.4 Data Processing in the Cloud

Historical traffic data stored in MongoDB are further processed to extract structural time series features which can be used for machine learning algorithms. Data quality metrics are monitored before and after the final cleaning and imputation to improve the integrity and inherit information value of the training features. The data extraction is performed with Dask, a Python library for parallel computation. The final features are stored in an InfluxDB, a time-series database optimized for fast, high-availability storage and retrieval of time series data. For high-quality visualizations, Grafana, an open-source monitoring and observability platform, is configured to run queries on InfluxDB data.

5.2.4.1 Data Quality Monitoring

In the context of AI-based MaaS applications, data management processes can be influenced by principles that are quite different from those ruling more traditional computing environments. Cloud deployments, streaming data, data volume, volatility and heterogeneity pose new challenges for data-driven analytics. Moreover, the limited explainability of many broadly used AI models adds another layer of ambiguity, since performance issues can be attributed to various factors (e.g., model selection, implementation, data quality). Therefore, data quality assessment and improvement are the first steps in an iterative process of designing, building and evaluating AI solutions. Even after deployment, continuous monitoring of data distributions is critical for detecting data shifts and promptly enact retraining to avoid performance deterioration. To improve data quality and integrity, we defined, quantified, and monitored four classes of errors: 1) duplicate data, 2) missing

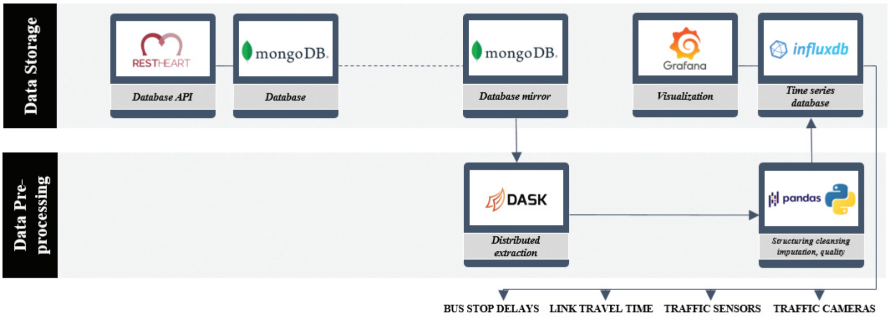


Figure 5.2.8 Data storage and processing in the cloud. The processed features can be retrieved and visualized as time-series and used for training AI prediction models.

data, 3) inconsistent values (e.g., outliers for traffic sensors and cameras, or negative values for travel-time durations, and 4) incomplete items (e.g., bus route segments with less than two GPS traces, or sensor measurements with a count period less than the one defined in the specifications). All these types of errors are considered of critical importance for obtaining a high-quality dataset to train machine learning models [16]. For each error class and each category of traffic data, we calculated the Errors-to-Data Ratio (EDR), i.e., the number of errors divided by the total number of items. To derive an overall data quality metric for each traffic data category, we used the unweighted EDR average across all error classes in the category. EDRs have been calculated before and after removing erroneous measurements. For missing data in the categories of sensors and cameras, the elimination was applied sensor-wise, only for those sensors that exceeded 50% of missing values. The remaining missing values are dealt with imputation by interpolation through time. The threshold was decided to retain a balance between losing information and injecting imputation related bias into the dataset.

5.2.4.2 Data Quality Observations

This section presents some of the preliminary observations obtained from applying the cloud-based data management pipelines on data collected for the week of February 19 to 25, 2021. Data observability is the first step to troubleshoot, understand, and explore the data. Figure 5.2.9 presents the weekly traffic data and bus traces stored as time-series in InfluxDB as they are captured in Grafana dashboards. Expected patterns of seasonality or

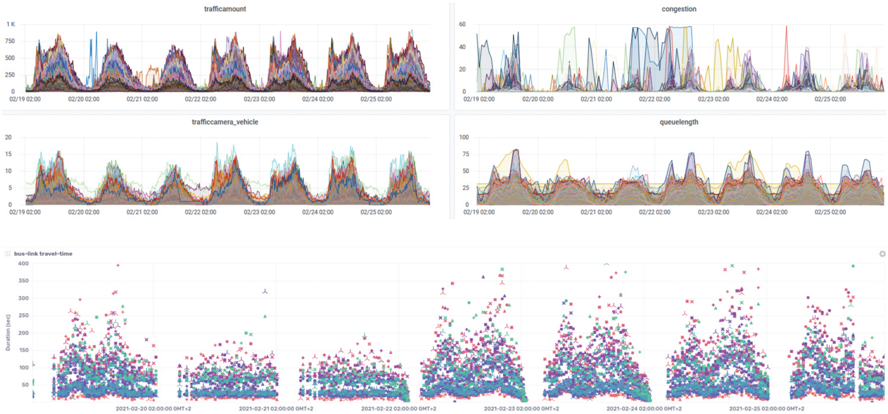


Figure 5.2.9 Weekly data from left to right, top to bottom: traffic amount, congestion, vehicle counts derived from traffic-camera images, queue length and travel-time durations for the segmented bus routes (bus-links).

unexpected outliers can be readily detected to assess the maturity of the data components and decide on further actions.

Subsequently, the EDR metrics were calculated for each category of traffic data and error type, before and after eliminating erroneous samples. Table 5.2.2 presents the ratios and the mean EDR reduction percentage in each category of traffic data. In addition, the number of total measurements is shown before and after the elimination. Our data quality monitoring strategy improves the data by reducing the errors by 26.95% and up to 100%. While the total number of measurements is only reduced by 14.91%, data quality

Table 5.2.2 *Errors-to-Data Ratio* (EDR) for five categories of traffic data collected for the week of February 19 to 25, 2021. EDR is given as a percentage before and after the first step of data cleaning, which involves eliminating erroneous observations.

	EDR (%) pre / post-processing				**% EDR**	**# Measurements**
	Duplicates	**Missing**	**Impossible**	**Incomplete**	**Reduction**	**pre/post processing**
Traffic amount	6.93 / 0	15.19 / 12.54	0.9 / 0	0 / 0	45.5	5,836,320 / 5,554,080
Congestion	0 / 0	15.4 / 12.58	2.88 / 0	0 / 0	31.18	5,473,440 / 5,090,400
Queue length	1.82 / 0	51.09 / 39.04	0.51 / 0	0 / 0	26.95	3,376,800 / 1,975,680
Bus traces	0 / 0	13.16 / 0	0.004 / 0	1.63 / 0	100	494,716/ 426,629
Traffic camera	0 / 0	56.64 / 3.97	0 / 0	0 / 0	93	223776 / 60,480
					Measurements Reduction	14.91%

analysis reveals a higher loss of information in the category of ‘queue length’, in which the lowest EDR reduction is recorded. This observation indicates that this category of data might be of low quality as a feature and needs to be further assessed to decide if it has to be excluded.

5.2.5 Conclusion

With advancing digitisation in the domain of ITS exploiting generated data becomes a key challenge to optimise operations to establish greener and more resource-efficient mobility. In this work, we presented a system architecture to acquire and process open traffic data which will allow AI-based modelling. Our architecture addresses two major challenges for such a system - data volume and quality. To compensate for a high data quantity and related communication overhead, computations are scaled and distributed to different layers in the edge-cloud continuum. Further, the presented monitoring strategies improve the quality of training data sets that are required by data-driven approaches. In future work, we will leverage the data to develop MaaS applications, such as predicting the estimated time of arrival (ETA) for public transport, optimising passenger transfer timing in a last mile use case.

Acknowledgements

This work is conducted under the framework of the ECSEL AI4DI “Artificial Intelligence for Digitising Industry” project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway.

References

[1] G. Smith, J. Sochor and M. Karlsson, “Mobility as a Service: Implications for future mainstream public transport,” in International Conference Series on Competition and Ownership in Land Passenger Transport (Thredbo), Stockholm, 2017.

[2] J. Wu, L. Zhou, C. Cai, J. Shen and S. K. Lau, “Data Fusion for MaaS: Opportunities and Challenges,” in IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), Nanjing, 2018.

[3] C. O. Cruz and J. M. Sarmento, "“Mobility as a Service” Platforms: A Critical Path towards Increasing the Sustainability of Transportation Systems," Sustainability 2020, vol. 16, no. 6368, 7 August 2020.

[4] City of Tampere, "Avoin data -lisenssi," [Online]. Available: https://www.tampere.fi/tampereen-kaupunki/tietoa-tampereesta/avoin-data/avoin-data-lisenssi.html. [Accessed 24 3 2021].

[5] ITS Factory, "ITS Factory - Innovative Tampere Site," [Online]. Available: https://itsfactory.fi/. [Accessed 24 3 2021].

[6] "infoTripla - Smart Mobility," [Online]. Available: https://infotripla.fi/. [Accessed 24 3 2021].

[7] City of Tampere, [Online]. Available: https://lissu.tampere.fi/timetable/. [Accessed 2021 March 24].

[8] City of Tampere, "Tampere traffic camera API," [Online]. Available: https://traffic-cameras.tampere.fi/.

[9] City of Tampere, "Trafficlightdata Service - API," [Online]. Available: http://trafficlights.tampere.fi/. [Accessed 24 3 2021].

[10] VDV Die Verkehsunternehmen, "CEN TS 15531 Service Interface for Real time Information (SIRI)," [Online]. Available: https://www.vdv.de/siri.aspx. [Accessed 24 3 2021].

[11] DATEX II, "DATEX II version 3 documentation portal," [Online]. Available: https://docs.datex2.eu. [Accessed 24 3 2021].

[12] A. Bochkovskiy, C.-Y. Wang and M. H.-Y. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, L. Zitnick and P. Dollár, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312, 21 Feb 2015.

[14] J. Schmid, P. Heß, A. Höß and B. Schuller, "Passive monitoring and geo-based prediction of mobile network vehicle-to-server communication," in 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, 2018.

[15] J. Schmid, M. Schneider, A. Höß and B. Schuller, "A Comparison of AI-Based Throughput Prediction for Cellular Vehicle-To-Server Communication," in 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, 2019.

[16] V. N. Gudivada, A. Apon and J. Ding, "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations," International Journal on Advances in Software, vol. 10, no. 1 & 2, 2017.