

## A comparison of AI-based throughput prediction for cellular vehicle-to-server communication

Josef Schmid, Mathias Schneider, Alfred Hos, Björn Schuller

### Angaben zur Veröffentlichung / Publication details:

Schmid, Josef, Mathias Schneider, Alfred Hos, and Björn Schuller. 2019. "A comparison of AI-based throughput prediction for cellular vehicle-to-server communication." In *15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 24-28 June 2019, Tangier, Morocco, 471–76. Piscataway, NJ: IEEE.  
<https://doi.org/10.1109/IWCMC.2019.8766567>.

### Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:

#### Deutsches Urheberrecht

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/>



# A Comparison of AI-Based Throughput Prediction for Cellular Vehicle-To-Server Communication

Josef Schmid, Mathias Schneider and Alfred Höß  
Ostbayerische Technische Hochschule  
Amberg-Weiden  
92224 Amberg, Germany  
Email: {j.schmid, mat.schneider, a.hoess}@oth-aw.de

Björn Schuller  
ZD.B Chair of Embedded Intelligence  
for Health Care and Wellbeing,  
University of Augsburg, 86159 Augsburg, Germany  
Email: schuller@ieee.org

**Abstract**—Nowadays, on-board sensor data is primarily used to detect nascent threats during automated driving. Since the range of this data is locally restricted, centralized server architectures are taken into consideration to alleviate challenges caused by highly automated driving at higher speeds. Therefore, a server accumulates this sensor data and provides aggregated information about the traffic situation utilizing mobile network-based vehicle to server communication. To schedule communication traffic on this fluctuating channel reliably, various approaches on throughput prediction are conducted. On one hand there are models based on aggregation depending on the position, e.g. connectivity maps. On the other hand there are traditional machine learning approaches, i.a. Support Vector Regression. This work implements the latter including OSM-based feature engineering and conducts a comprehensive comparison on the performance of these models utilizing a uniform dataset.

**Index Terms**—throughput prediction, support vector regression, connectivity map, cellular network

## I. INTRODUCTION

In the last couple of years, the development of safety and convenience features in the automotive context is progressing rapidly. The focus of development heads to reach higher levels for on-road motor vehicle Automated Driving Systems (ADSs), defined by the Society of Automotive Engineers International in the J3016 Standard. Separating the progress of automation into Levels 0-5, Conditional Driving Automation (Level 3) describes systems, which provide a conditional hands-off and eyes-off automated mode. Since current systems solely rely on on-board sensor to fulfill automated driving tasks, implementing features for higher speeds are challenging, because they exceed the available sensor range. For example, the range of a automotive radar covers at most 240 m of the upcoming environment. Considering the advisory speed limit on German motorways of 130 km/s, this results in a time interval of 6.6 s for a driver to retake the driving task and initiate a suitable maneuver. Since the take-over transition takes approximately 4 s according to recent studies [1], it does not leave the driver with enough time to react properly. Therefore, new concepts are investigated to increase the driver's safety by the means of vehicle-to-everything (V2X) communication. One approach, introduced in [2], is a centralized server, sharing static and dynamic traffic events within a high resolution map. Using mobile network technologies, e.g. Long-Term

Evolution (LTE) to exchange this data, a reliable connection is required to maximize the passenger's safety. As a side-effect of the progressive development of the automation levels, the passenger user experience steps forward, creating a higher demand for mobile network depending services, such as video streaming and working on-board (rolling office). Nevertheless, since the safety aspects must always have the highest priority, it must be ensured to have a minimum of downlink bandwidth for driving related tasks. Based on the introduced prediction, various applications can be developed subsequently, e.g. to improve the reliability of the connection or resource efficiency.

Therefore, the paper is structured as follows. In Section II, the related work is discussed, followed by an introduction of the collected dataset in Section III. In Section IV a detailed explanation of the data preprocessing, required by the prediction models, is provided. Subsequently, AI-based time series regression models are described in Section V, before Section VI conducts a comprehensive evaluation of the predictors and geo-based methods. Finally, we conclude our paper and present an outlook to future work.

## II. RELATED WORK

This section outlines the related work of the state-of-the-art in geo-based throughput prediction including our previous work as well as AI based approaches. Geo-based models aggregate datapoints according to their geographical position and can be visualized as a map overlay. Therefore, they either use the historical data to calculate a model for a defined area with respect to the aggregation function and segment shape. For the latter, different approaches were examined in previous work, e.g. predefined segments from a map provider as leveraged in [3–5], manually calculated road segments with a fixed size [6], [7] or grid based segmentation. In our previous work [8] we evaluated these models and revealed their limitations. Furthermore, advanced geostatistical approaches such as Kriging are considered [9].

Apart from geo-based concepts, methods based on Machine Learning (ML) are considered for forecasting Quality of Service (QoS) parameters such as the downlink throughput as well. Whereas geo-based models depend on exact vehicle's position, this prerequisite is obsolete for ML-based time series regression. Thus, these estimators can be harnessed at areas where no

historical data is available and allows to comprise multivariate inputs in contrast to autoregressive models. A commonly used regression model in former work is the Support Vector Regression (SVR) [10], [11], which will also be evaluated in this work. Thereby, SVR takes multiple input variables to generate a prediction and utilize a kernel function, allowing to design predictors with high accuracy. Nevertheless, according to Liu et al., the linear kernel outperforms the more complex radial kernel function [12].

Another estimator is the Random Forest Regression [13], [14] which is an ensemble of decision trees averaging the results. Furthermore, there further exist more complex ML-based algorithms such as *HOAH*, implemented by Wei et al. in [15]. They leverage Support Vector Classification (SVC) to determine which time series regression model should be applied.

Besides these Supervised Learning models, additional methods from the area of Reinforcement Learning are presented, e.g. for DASH adaptation algorithms [16], and genetic-algorithm-based prediction model [17]. Nevertheless, these ML models are not in the scope of the evaluation for this paper.

### III. DATA ACQUISITION

For this work, the measurement setup and dataset introduced in [8] is leveraged, composing QoS parameters from a tool called TCPAnalyzer. This tool monitors and aggregates parameters from three different sources, the LTE modem, network interface and Global Positioning System (GPS) device, and calculates the metrics based on a passive network sniffing procedure during repeated download of a 4 MB file from a local server. Whereas QoS parameters are calculate with a frequency of 3 Hz, the GPS position updates are limited to 1 Hz. The Germany network provider Vodafone is chosen for this setup.

The dataset comprises the measurement of a campaign in the vicinity of Amberg, Germany, and covers 46 rounds overall of a predefined test round as shown in Table I. This test round is about 28 km long and is equidistantly separated to cover urban, interurban and motorway scenarios. The majority of the rounds are driven in a counterclockwise manner starting in the west of Amberg, a minor part is recorded in a reverse direction.

Table I: Test rounds of the dataset at measurement campaign in Amberg

Date	Rounds	Distance [km]	Datapoints
13.11.2017 (Mon.)	11	298	54 836
14.11.2017 (Tue.)	11	301	58 117
15.11.2017 (Wed.)	14	385	69 146
16.11.2017 (Thu.)	10	270	49 453
<b>Overall</b>	46	1254	231 552

### IV. DATA PREPROCESSING

This chapter deals with the general data preprocessing workflow for setting up a throughput download, denoted as  $TP_{DL}$ , prediction. Each block is implemented in Python as a

Table II: OSM properties retrieved by the Overpass API and corresponding priority groups for feature engineering

	Property (in OSM)	$r_q$ [m]	$p$	Values (descending priority)
categorical	highway (highway)	[10, 100, 250]	1	motorway, trunk, primary, secondary, tertiary, residential
			2	motorway_link, trunk_link, primary_link, secondary_link, tertiary_link
	landuse_natural (landuse)	250	1	farmland, farmyard, forest, grass, greenfield, meadow, orchard, village_green
			2	recreation_ground, basin, conversation, plant_nursery
			3	salt_pond, quarry
	landuse_anthropological (landuse)	250	1	retail, residential, industrial, commercial
			2	military, religious, allotments, brownfield, cementry, construction, landfill, port, vineyard
			3	depot, garages, railway
	amenity (amenity)	250	1	university, school, college, library, clinic, hospital, arts_centre, market_place
			2	place_of_worship, kindergarden, language_school, driving_school, music_school, research_institute, parking, nursing_home, social_centre, social_facility, cinema, townhall, biergarden
			3	bar, cafe, food_court, archive, pub, restaurant
numerical	building (building)	100	-	Number of buildings
	building_level (building:level)	500	-	Avg. building level

self-containing process for improving re-usability and allowing to transfer the pipeline in a distributed manner. Thereby, several steps are shared between geo-based and ML-based time series regression models.

#### A. Geo-based feature engineering

To compensate for the disadvantage of ML regression models of the missing vehicle's GPS position, this work features the concept of adding derived geographical information to the model input. Thereby, the actual position is utilized to query the Overpass API which is a web-based service interface for requesting OSM map data efficiently [18]. It provides the vast knowledge of the community-maintained OSM database containing geographical items and their properties such as position, type and shape. The objective is to retrieve further geographical features, which correlate with the  $TP_{DL}$  to compensate missing operator side features as the amount of users in a cell.

Therefore, OSM item key tags such as amenity, land use and buildings are requested for a given region of interest. Aside from these features which may affect the utilization of the mobile network, another aim is to gather information about the natural environment. This can be collected by querying key tags such as highway and building. Table II summarizes all queried OSM item properties and their values. The granularity is increased by clustering and prioritizing these properties according to the following algorithm. The Overpass API is called with increasing query radius  $r_q$  as long as no valid item is returned. If no item is available for the maximum range, the feature is undefined for the queried position. For each categorical property the OSM value is selected which has the highest group priority  $p$  and most occurrences in the response. If two values have the same  $p$  and number of occurrences, the first value is utilized according to the table. For numerical features this is procedure is simplified by returning the number of building items and their average level.

The derived result for the highway key tag for the test round in Amberg, as shown in Figure 1, proves that this approach provides reliable environmental information such as the road scenario and can be determined in an automated way.

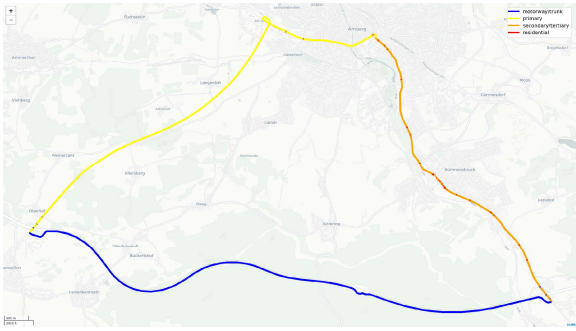


Figure 1: Derived feature from OSM highway key for test round

### B. Filter

Several filter steps have the purpose to refine the raw data before the following preprocessing stages. The implementation uses a flexible, layer based, architecture. Each filter step corresponds to a distinctive transitioning method, which receives one frame and decides whether the input vector passes.

- *Invalid cellular network technologies:* Since the  $TP_{DL}$  prediction focuses solely on LTE based connection, samples using other technologies are removed.
- *TCP slow-start:* Because the aim is to predict the maximal throughput, it is necessary to remove all samples recorded within the TCP slow-start phase of the connection. Therefore, a conservative interval up to the first 1000 kB of each download is removed. The filter utilizes the RTT\_SYN value provided by the TCPAnalyzer to detect a new download and calculates the downloaded size of the connection by the means of the DL\_SUM property. After exceeding 1000 kB, the samples pass. Additionally,

a side-effect of this implementation is that samples during download pauses are automatically eliminated.

- *Invalid GPS:* Because the GPS module takes some time for initialization, the first positions after each TCPAnalyzer start are set to 0°N, 0°E. Because this creates inconsistencies, e.g. for the geo-based feature derivation, all samples nearby the geographic origin are removed.
- *Sampling period:* Despite using an outgoing sampling frequency of 3 Hz, the TCPAnalyzer uses an internal clock to calculate the current  $TP_{DL}$ . This clock is provided by the TCPAnalyzer in the PERIOD field. Especially at the start or end of the download, these values differ from the intended 300 ms, which also results in contradictory  $TP_{DL}$  in comparison with the remaining measurements for this download. Consequently, if the PERIOD is lower than 100 ms or higher than 350 ms, the sample is dropped.

This introduced filter layer reduces the overall number of training samples to approx. 100000. Thereby, the slow-start filter is responsible for the removal of the majority of the samples (approx. 50 %).

### C. Downsampling

In contrast to the previously described processing components, the next four stages are exclusively applied for the ML models. When dealing with the problem of forecasting a time series, the best result is archived by predicting one step ahead. The requirement is to predict the  $TP_{DL}$  in the horizon  $T_p$  of 15 s. Because there is a discrepancy between  $T_p$  and the sampling time  $T_s$  of the TCPAnalyzer (300 ms), a sliding window downsampling mechanism has to aggregate up to  $n_D = T_p/T_s = 50$  samples using the median. Besides rearranging the data to match the required interval of 15 s, downsampling offers the possibility to automatically engineer additional numerical features within the downsampling interval, such as drifts, minima and maxima, as well as standard deviations.

### D. Encoding

Since regression models cannot handle the usage of categorical features such as highway or amenity natively, it is necessary to encode these values. Therefore, one-hot encoding is applied to all categorical features, which creates a binary column for each category. The resulting sparse matrix replaces the original feature within the existing table.

### E. Shifting

Additional preparation is conducted by labeling the training samples by shifting the subsequent value as the prediction value. Since the  $TP_{DL}$  trend is not a stationary time series, instead of predicting the upcoming value directly the difference between these measurements is selected as the value for forecasting. Furthermore, by applying autocorrelation function it is shown that previous  $TP_{DL}$  values are significant. To add them as additional input features, a foreshift operation within this shifting component recycles these values. This procedure results in dropping the samples at the beginning and ending of the series due to missing values as depicted in Figure 2.

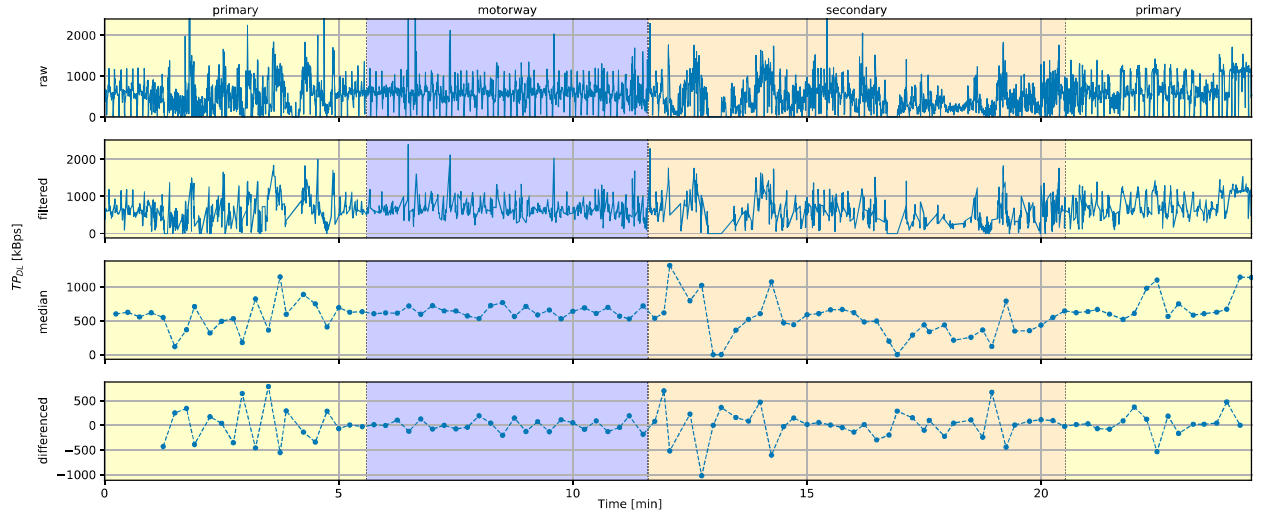


Figure 2:  $TP_{DL}$  during different preprocessing steps

#### F. Feature selection

Since the introduced preprocessing steps increase the number of features, an iterative approach for the feature selection is conducted.

- *Manual elimination*: Because the prediction should be independent of the actual GPS location, all identifying positional features are removed from the set. This affects features such as LAT and LON as well as Mobile Country Code (MCC) and Tracking Area Code (TAC). CELL\_ID field is removed accordingly, so that the model works for an unknown cell.
- *Variance threshold*: A variance threshold filters features with overall measurements having a variance lower than 1%. It primarily eliminates one-hot encoded categorical features, which never occurred in the dataset.
- *Feature correlation*: Detecting and eliminating redundant numerical features, which e.g. can be transformed by a linear function, is conducted by using the Pearson correlation coefficient between all feature series. If correlating features are determined, such as the pair ASU and RSRP, one of them is removed.
- *Principal Components Analysis (PCA)*: To reduce the number of significant dimensions, the PCA is calculated. By summarizing the features' variances, the number of relevant features can be determined. For this model the number of features is selected which cover at least 80% of the explained variance.
- *Recursive Feature Elimination (RFE)*: This greedy algorithm, implemented in the scikit-learn framework, discovers appropriate features by iteratively training new models, while removing either the best or the worst feature for the next run. This step is repeated, until a limit of variables is reached. RFE is calculated for all ML-based models that will be introduced in Section V. A voting mechanism selects the features shared by all model

eliminations. Thereby, the number of relevant feature is set by the preceding PCA calculation.

Overall, 29 features are utilized for the ML-based models. Table III depicts an overview of the selected base-features with their corresponding derived values. Interestingly, neither the Absolute Radio Frequency Channel Number (ARFCN), having a significant impact on the  $TP_{DL}$ , nor the highway type are used for the model according to this selection approach.

#### V. AI-BASED PREDICTION

For the AI-based time series regression three different models are harnessed, implemented by the scikit-learn framework:

- *Linear Regression (LR)*: LR uses the linear least squares method for regression. To further optimize the result, it leverages a regularization term. In this paper we use Ridge regression, which uses the Euclidean distance ( $L_2$  norm) for shrinking. Thereby, the regularization strength  $\alpha$  is set to 0.5.
- *Random Forest Regressor (RFR)*: As an ensemble of multiple decision trees, RFR calculates its outcome based on the averaged results of the trees. This work parameters the RFR with a maximal depth of 4 and ensembles 96 estimators.
- *Support Vector Regression (SVR)*: SVR uses the concept of optimising the flatness of hyperplane, specified by support vectors, while deviating at most by  $\epsilon$  from the training data samples. Thereby, different kernels can be harnessed. As recommended in previous work [12], a linear kernel is utilized.

All models are training using the same feature selection and dataset for predicting the upcoming  $\Delta TP_{DL}$ , excluding the test rounds depicted in Table IV. The actual  $TP_{DL}$  prediction is reconstructed by adding  $\Delta TP_{DL}$  with corresponding measurement. Furthermore, the Persistence Algorithm (PA), which sets the current value as the prediction result, is utilized as a baseline to validate the ML-based models.

Table III: Selected features with description and their derived function

	Feature name	Description	Unit	Min	Max	Drift	max. Drift	Std. dev.	Mean	Median	Lags
TCPAnalyzer	ASU	Arbitrary strength unit	dB						✓		
	INTRAF	No. of intrafrequency cells	-	✓					✓	✓	
	LTE_N_UMTS	No of. neighboring UMTS cells	-	✓				✓			
	RSRQ	Reference signal received quality	dB		✓		✓		✓		
	RSSI	Received signal strength indicator	dB	✓	✓						
	RTT_SYN	Round trip time for SYN-ACK	ms	✓						✓	
	SINR	Signal to interference noise ratio	-	✓				✓	✓		
	SPEED	Vehicle speed	m/s	✓							
OSM	TP_DL	Throughput downlink	kB/s		✓	✓	✓		✓	✓	1-4
	BUILDING	No. of buildings	-						✓		
	BUILDING_LEVELS	Average building levels	-	✓							
	LANDUSE	Land use enumeration	-	[retail]							

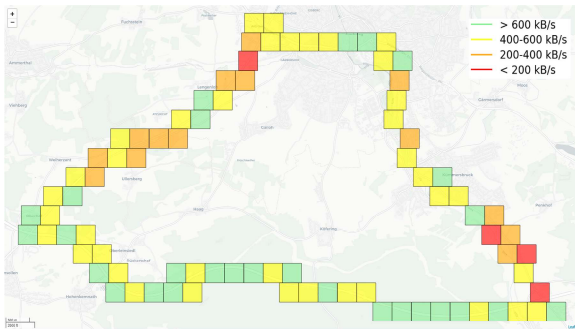
## VI. COMPARISON OF AI AND GEO BASED PREDICTION

This section conducts the comparison of the performance of the trained GEO-based and the ML-based models. Therefore, four dedicated test rounds, one for each test day, as shown in Table IV, are utilized.

Table IV: Overview of test rounds

	Datetime	Round	Option	Datapoints
I	13.11.17, 13:32 (Mon.)	7	-	4832
II	14.11.17, 9:45 (Tue.)	1	-	5643
III	15.11.17, 11:40 (Wed.)	4	-	4820
IV	16.11.17, 11:21 (Thu.)	5	reverse	6056

For comparing ML-based models, a grid-based Connectivity Map (CM) is calculated based on the same dataset. It is configured to aggregate the samples within tiles which have an edge size of  $d_{ROI} = 400$  m and uses a median for averaging the measurements. Figure 3 depicts the generated CM.

Figure 3: Grid-based Connectivity Map,  $d_{ROI} = 400$  m

For each test round upcoming  $TP_{DL}$  values are forecasted according to the respective model and its preprocessing steps. The accuracy is determined by calculating several performance metrics. Since the inserted sample series are discontinuous, caused by the data acquisition, filter mechanism and prediction shift, it is not ensured for ML-based models, that both, the actual value and the predicted value, are defined at a time. Thus, the performance metric function is applied for samples

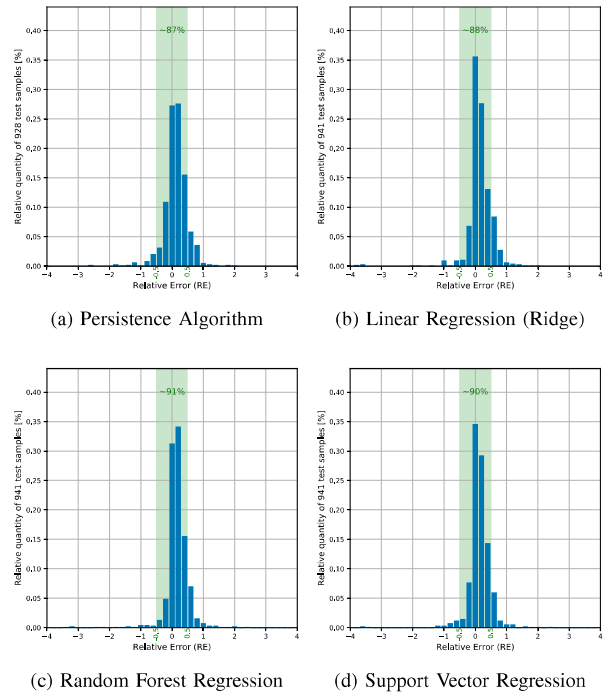


Figure 4: Comparison of relative histogram of relative errors for ML-based model in Round I

that include both values. In case of the geo-models, it is assumed, that the upcoming position is available. Consequently, the predicted value has not to be shifted by  $T_p$ . Besides of Root Mean Squared Relative Error (RMSRE) used by our previous work [8], a metric is defined, which corresponds to the ratio of samples with a Relative Error (RE) smaller than 50 %, denoted as  $\rho_{0.5}$ . Thereby, the RE is defined as in Equation 1, with the measured value  $R$  and the predicted value  $\hat{R}$ .

$$RE = \left| \frac{R - \hat{R}}{\min(R, \hat{R})} \right|, \text{ subject to } R, \hat{R} > 0 \quad (1)$$

This metric is depicted in Figure 4, showing histograms of the relative error distribution of the ML model forecasts. It



reveals that all models, including the PA, are quite accurate. A similar result is deduced for the other validation rounds as shown in Table V. The metrics show that all ML-based models have nearly the same accuracy and outperform the geo-based models and the PA baseline slightly. Thus, the AI-based models are a promising alternative for the  $TP_{DL}$  prediction, because of their independence of the availability of historical data at a certain area. This is indicated by the validation of the round recorded reversely, but must be evaluated further by validating the trained models against other locations.

Table V: Validation results including Mean Absolute Error (MAE), Mean Relative Error (MRE), RMSRE and  $\rho_{0.5}$

	MAE [kB/s]	MRE	RMSRE	$\rho_{0.5}$ [%]
CM-Grid	92.83	0.30	1.09	84.95
I	82.40	0.22	0.59	91.91
II	116.48	0.33	0.79	79.38
III	78.02	0.18	0.34	89.10
IV	94.43	0.47	2.65	79.41
PA	110.62	0.37	1.95	84.11
I	110.27	0.27	0.49	87.28
II	120.22	0.35	0.80	78.64
III	98.04	0.22	0.58	87.75
IV	113.94	0.64	5.92	82.75
RFR	106.61	0.29	0.75	84.91
I	98.21	0.25	0.46	88.95
II	127.67	0.33	0.73	78.31
III	85.33	0.18	0.53	93.36
IV	115.26	0.40	1.29	79.04
LR	102.72	0.29	0.95	86.55
I	89.98	0.23	0.47	91.07
II	129.22	0.34	0.78	80.81
III	79.12	0.17	0.59	94.10
IV	112.56	0.43	1.95	80.22
SVR	103.79	0.29	0.80	86.11
I	92.98	0.23	0.46	90.44
II	128.33	0.33	0.74	80.24
III	82.76	0.18	0.57	93.73
IV	111.11	0.40	1.44	80.05

## VII. CONCLUSION

In this paper, we continued our work on the prediction of QoS parameters such as the throughput for downloads using LTE cellular networks for vehicle-to-server communication by implementing Artificial Intelligence (AI)-based models and compared them with a geo-based model implementation. Based on the same data acquisition setup, we elaborated a comprehensive preprocessing pipeline for preparing the data for time series prediction including a novel OSM-based feature engineering step. Comparing the performance of ML-based and geo-based models, it is shown that they are a promising alternative to the CMs, based on the results in our work, especially SVRs and RFRs are worth considering. For future work, two aspects should be further investigated. First, the performance estimators using deep neural network topologies

should be examined. Based on these results, a second approach is to develop and evaluate hybrid geo- and AI-based methodologies to further improve the prediction of QoS parameters.

## VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the following H2020 - ECSEL Joint Undertaking projects and agencies for financial support including funding by the German Federal Ministry for Education and Research (BMBF): AutoDrive (Grant Agreement No. 737469, funding code 16ESE0255) and PRYSTINE (Grant agreement No. 783190, funding code 16ESE0330).

## REFERENCES

- [1] J. Radlmayr et al., "How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 58, 1., Los Angeles, CA, 2014, pp. 2063–2067.
- [2] F. Jomrich et al., "Analysing communication requirements for crowd sourced backend generation of HD Maps used in automated driving," in *2018 IEEE Vehicular Networking Conference (VNC)*, 2018.
- [3] T. Pögel and L. Wolf, "Prediction of 3G network characteristics for adaptive vehicular connectivity maps (poster)," in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 121–128.
- [4] L. Kelch et al., "Cqi maps for optimized data distribution," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, 2013, pp. 1–5.
- [5] T. Pögel and L. Wolf, "Optimization of Vehicular Applications and Communication Properties with Connectivity Maps," in *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*, Clearwater Beach, FL, 2015, pp. 870–877.
- [6] J. Yao et al., "Using bandwidth-road maps for improving vehicular internet access," in *2010 Second International Conference on Communication Systems and Networks (COMSNETS2010)*, 2010, pp. 460–461.
- [7] G. Murtaza et al., "Creating personal bandwidth maps using opportunistic throughput measurements," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 2454–2459.
- [8] J. Schmid et al., "Passive monitoring and geo-based prediction of mobile network vehicle-to-server communication," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 1483–1488.
- [9] B. Taani and R. Zimmermann, "Spatio-temporal analysis of bandwidth maps for geo-predictive video streaming in mobile environments," in *Proceedings of the 2016 ACM on Multimedia Conference*, 2016, pp. 888–897.
- [10] C. Lee et al., "Analytical modeling of network throughput prediction on the internet," in *IEICE TRANSACTIONS on Information and Systems*, Vol. E95-D, No. 12, 2012, pp. 2870–2878.
- [11] B. Wei et al., "A History-Based TCP Throughput Prediction Incorporating Communication Quality Features by Support Vector Regression for Mobile Network," in *2017 IEEE International Symposium on Multimedia (ISM)*, 2017, pp. 374–375.
- [12] Y. Liu and J. Lee, "An empirical study of throughput prediction in mobile data networks," in *Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.
- [13] A. Samba et al., "Instantaneous throughput prediction in cellular networks: Which information is needed?," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 624–627.
- [14] E. A. Walelgne et al., "Analyzing throughput and stability in cellular networks," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–9.
- [15] B. Wei et al., "HOAH: A Hybrid TCP Throughput Prediction with Autoregressive Model and Hidden Markov Model for Mobile Networks," in *IEICE Transactions on Communications (2018)*, 2017.
- [16] M. Gadaleta et al., "D-DASH: A deep Q-learning framework for DASH video streaming," in *IEEE Transactions on Cognitive Communications and Networking 3.4 (2017)*, 2017, pp. 703–718.
- [17] C. Hernandez Benet et al., "Predicting expected TCP throughput using genetic algorithm," in *Computer Networks*, Vol. 108, 2016, pp. 307–322.
- [18] OpenStreetMap contributors, "Planet dump retrieved from <https://planet.osm.org>," in <https://www.openstreetmap.org>, Online; accessed on 19th October 2018