IEEE Open Journal of the Communications Society

Received 2 August 2022; revised 2 September 2022; accepted 23 September 2022. Date of publication 28 September 2022; date of current version 10 October 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3210289

An Open Mobile Communications Drive Test Data Set and Its Use for Machine Learning

STEFAN FARTHOFER, MATTHIAS HERLICH, CHRISTIAN MAIER, SABRINA POCHABA, JULIA LACKNER, AND PETER DORFINGER

Intelligent Connectivity, Salzburg Research, 5020 Salzburg, Austria

CORRESPONDING AUTHOR: M. HERLICH (e-mail: matthias.herlich@salzburgresearch.at)

This work was supported in part by the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology and in part by Austrian State Salzburg.

ABSTRACT The capability to provide guarantees for network metrics, such as latency, data rate, and reliability will be an important factor for widespread adoption of next generation mobile networks and hence, such metrics play a central role in standards for new wireless communication technologies. However, due to the inherently stochastic nature of mobile communications, any guarantees can only be of statistical nature and are highly dependent on the actual physical environment. To analyze the stochastic behavior, this paper presents a tool chain for measurement, collection, evaluation, and prediction of controlled mobile communications drive test data. We also publish the underlying data set of measurements covering two years' worth of highway traffic on a 25 km long section comprising 267 198 data points. We statistically evaluate the data set and validate it with a corresponding data set from another source. Applying machine learning to the data set illustrates possible use cases: Feed-forward neural networks to predict the data rate in five application scenarios, LIME to explain the behavior of the model, and an autoencoder to describe the interaction of five signal strength parameters. The data set and the tool chain show how machine learning can be applied to wireless networks and provide fellow researchers with the means to make further experiments.

INDEX TERMS Drive-tests, mobile communications data, machine learning, prediction.

I. INTRODUCTION

HISTORICALLY, there have been various disjoint approaches to measuring and collecting quality metrics for mobile communications. These span from crowdsourced data collection (private [1], [2], [3] and regulatory [4], [5], [6]), over dedicated (drive test) measurement campaigns [7], [8] to monitoring within a mobile network itself.

The network centered approach, however, requires full access to the communication network and is typically only feasible for the network operator. Network operators constantly monitor the state of their network to provide the targeted quality of service to the users, or to recognize network anomalies or failures. Their supervision technology is capable of monitoring the network usage and all of its parameters in full detail and provides essential insights for network planning. The availability of such data can be an enabler for certain aspects of research, such as comprehensive user behavior analyses [9] or traffic type analyses [10]. In principle, machine learning approaches can even be used for adaptive control loops (e.g., power control) on lower layers of the communication stack. However, our data set targets higher layer approaches that operate on more coarse grained data (see application scenarios in Section V-B). Unfortunately, due to the delicate nature of such extensive data, network operators typically do not disclose this data or disclose it only to selected researchers under strict confidentiality agreements.

This level of secrecy impedes the reproducibility of scientific results and makes it impossible for other researchers to do further work with this data. For this reason, scientific and commercial approaches to the measuring and collecting of mobile communications quality metrics have evolved, that do not rely on access to a network operator's data. These approaches can be categorized as either crowdsourced measurements or controlled measurements. Commercial services typically favor crowdsourced approaches, since no expensive

and personnel-intensive measurement campaigns are needed. Due to scaling effects, it is possible to relatively quickly generate many measurements, albeit with the drawback of a lack of control of measurement accuracy.

Controlled mobile communications measurements require knowledge about the measuring equipment and need careful design of the measuring methodology. Ideally, measurements cover the full geographical and temporal region of interest without gaps. This requires enormous effort and is thus rarely possible for larger regions. Therefore, depending on the focus of a survey these measurements are usually geographically static or moving (e.g., drive tests) and specified as either random or systematic.

The contribution of this paper is two-fold: first, we present a tool chain for measurement, collection, evaluation, and prediction of controlled mobile communications drive test data. Second, we statistically evaluate an exemplary data set and demonstrate the tool chain, which includes machine learning methods for prediction. The data set contains 267 198 data points coming from two years of driving a car on a 25 km long highway section. We actively measure the maximum achievable data rate in the LTE network of a major Austrian cellular network operator and monitor all important general parameters, such as GPS position, time, and signal parameters.

To demonstrate the potential of such a data set, we use methods of classical statistics and modern data science. The methods described include a basic analysis of the data set, such as distribution of the data and fitting parameters of distributions to approximate those. Additionally, an autocorrelation analysis shows how the data rate changes with changes in time and space. A validation with a third party data set increases the confidence in the quality of the data. We use classical means and machine learning to predict the maximum achievable data rates for five scenarios. The methods of Explainable AI are used to understand how the machine learning model depends on the input parameters. These methods calculate which input parameters have the biggest influence on the prediction, to check whether the machine learning algorithm learns as intended. Lastly, we use an autoencoder for the parameters describing the signal strength to compress the information contained into fewer values and detect outliers.

The remainder of this paper is organized as follows. Section II summarizes the related work on gathering and analysis of wireless-communication data sets and explains the main differences to our paper. Section III describes the data set in detail and introduces the measurement setup, the data collection, and the data anonymization methods used. Then, the data set is evaluated and validated in Section IV. In Section V, we apply machine learning to predict the data rate and illustrate possible use cases.

II. RELATED WORK

Gathering wireless-communication data sets has always played a crucial role in mobile communications due to its inherent stochastic nature. Wireless-communication data sets range from low level physical layer measurements to social network analyses. For example, channel sounding has a long tradition to validate theoretical channel models in real environments and to determine the model parameters. Increased application of signal processing techniques and advances in machine learning extend the focus of attention also towards the analysis of higher layers.

In earlier work [11], we gave an overview of available wireless communication data sets, clustered by whether they are concluded, one-time experiments or ongoing data-collection efforts. The first type is typically conducted to build physical layer models or used as training data for machine learning models. Nowadays, ongoing data-collection efforts are widely conducted both by the public and private commercial sector.

One example [12] measured the parameters of a wireless network with dedicated measurement equipment (DME) and commercial off-the-shelf (COTS) mobile phones. They cooperated with the mobile network provider and ran a dedicated measurement campaign to collect the data. They conducted their measurements within one week in a highly controlled 5G test field, including control over, e.g., artificial load or different base station configurations. In contrast, our data set spans over two full years with measurements conducted in a commercial mobile network, where the behavior of the network is largely determined by its active mobile users. Additionally, our data set is publicly available.

Regulatory agencies in many countries conduct comprehensive measurement campaigns to assess legal compliance of operators and to gather insight about the current state of the cellular networks. Some provide their measurements as open data, e.g., the telecommunications regulatory agency of Austria [5] and Germany [6]. OECD provides a detailed report [4] that examines the approaches being taken in its member countries to measure broadband performance.

Nowadays, commercial crowdsourced data approaches, such as Ookla's Speedtest® [1] are becoming more important, because they can generate large amounts of data. The data generated by such approaches can be used for similar applications as dedicated measurement campaigns. Especially during the COVID-19 pandemic in 2020 and 2021, mobile phone data from network operators was often used for advanced data analytics and data intelligence, and was used to estimate people's mobility behavior [13], [14], [15].

Unfortunately, many rather straight-forward "quantity over quality" based crowdsourced approaches are poorly suited for benchmarking the quality of mobile networks due to irregular measurements and systematic biases. However, the ubiquity of crowdsourcing has stimulated the interest of the scientific community. Some, for example [16], try to overcome inherent limitations that are due to the enormous system complexity and large number of possible predictors, by carefully aggregating data in a network-centric approach. Others, for example [17], propose a model

based approach to estimate the throughput by signal metrics, using Opensignal's crowdsourced data [2] spanning the entire US.

Especially, data rate measurement campaigns are expensive and potentially cause high network load. Therefore, crowdsourced approaches are often used for these type of measurements. However, relying on crowdsourced data frequently causes complications due to imbalanced data components while simultaneously having a high degree of freedom in the involved unknown variables. A recent white paper provides definitions, use cases and challenges for crowdsourced measurements [18].

In [19], the authors present an empirical analysis of client-based end-to-end data rate prediction for 5G NSA vehicle-to-cloud communications, and demonstrate that machine learning-based data rate prediction, that utilizes context measurements as input features, can be a competitive approach. As current mobile communication standards already include channel state information (CSI) feedback to inform the user equipment (UE) about channel quality, this data can also be used for predictions. In [20], the authors try to predict the channel quality indicator (CQI) by the SNR using machine learning techniques on simulated data. Furthermore, the 5G network itself provides the network data analytics function (NWDAF) that could be directly used for QoS prediction, and delivering this information to a vehicle-to-everything (V2X) application, as outlined in [21].

In [22], the authors train long short-term memory (LSTM) neural networks with several diverse relatively short term traces with durations ranging from minutes to some hours. Similarly, in [23] the authors propose a cellular link bandwidth prediction model based on detailed lower-layer information obtained by the specialized Qualcomm eXtensible Diagnostic Monitor tool. They then compare this prediction performance to the prediction performance using conventionally obtained information via the standard phone API. In [24], the authors compare random forests, support vector machines, and LSTM neural networks in their throughput prediction performance. References [22], [23], [24] mainly concentrate on active measurements, whereas in [8] the authors collect data using a specific measurement design of systematically changing between idle and connected mode. They use these measurements to obtain the relationship between idle and connected mode and propose a non-intrusive throughput prediction method.

In this work, we try to find a trade-off between the complexity of crowdsourced data and limitations of controlled drive tests. We use defined commercial-off-the shelf LTE devices (see Section III-A) in a specified scenario (continuous maximum throughput in a moving vehicle) in a well-bounded geographical area to reduce the number of variables and to ensure an evenly and densely populated data set. However, due to practical reasons (typical office hours of the drivers) there is still a temporal bias towards rush-hour. We then use this densely populated data set for machine learning.

Machine learning based prediction methods use a combination of signal parameters to infer the data rate implicitly, by having learned the communication network topology and its dynamics. Signal parameters give information about the current communication conditions, such as signal-to-noise ratio (SNR) or received signal powers. Repeated measurements in the same geographical area allow the model to also learn parts of the influence of the network topology and the dynamics on the predictions.

III. DATA SET

With the objective of applying machine learning methods on the data set (see Section V) one has to find a suitable balance between the generalizability and representativeness, and the feasibility of measurement campaigns to obtain this data set in the first place. We limit the complexity on the user side, by using a strictly defined measurement setup described in Section III-A while examining a typical real world scenario of repeated drives on a specific highway section that is presented in Section III-B. Due to privacy reasons, the raw data set is anonymized to some degree, as described in Section III-C.

The final data set published in combination with this paper is openly available on github[1] and on CRAWDAD [25].

A. MEASUREMENT

We use a custom-built Raspberry Pi 2 based measurement equipment to conduct our controlled drive test measurement campaign. Each Raspberry Pi is equipped with a Huawei E3372 LTE network interface that provides the Huawei HiLink API for signal parameter monitoring. The measurement equipment constantly exchanges data at full load. Five TCP flows with 100 MiB HTTP traffic each simultaneously transmit and are continuously restarted once a download is complete. Then, the data rate is monitored for each flow and averaged over 1 second. The overall data rate is the sum of the individual flow data rates. In addition, a GPS module simultaneously records the geographical position and the time. Because the measurements start at the same time as the time sync procedure starts, the precision of the time goes through a transitory period. That is, the time (and measurements which depend on time) depend on the time sync to reach a stable state. The Huawei HiLink API is used to monitor the signal parameters that are provided by the LTE network interface in 1 second intervals. The measurements are managed using the MINER software platform [26]. Table 1 describes the measured and monitored data in more detail.

B. COLLECTION

Our drive test measurements are derived from organized measurement drives with the explicit permission from the drivers (for data anonymization see Section III-C). We chose a 25 km long Austrian highway section during an observation period

1. https://github.com/mherlich/wireless-data-set

TABLE 1. Selected columns of the data set and parameter description.

Parameter	Description	Example value	Minimum	Maximum	Mean	Standard deviation
time	ISO 8601 timestamp	2019-12-24 20:57:32	2018-01-15	2019-12-20		
lat	latitude coordinate in degrees	47.822768	47.842 587	47.856 92	47.851 03	0.004 167
long	longitude coordinate in degrees	13.040866	13.080 002	13.333	13.192 63	0.073 307
ele	elevation in meters	450.7	398.4	712.7	582.5	38.5
signal	signal strength in "bars"	5.0	0	5	3.74	1.40
rssi	received signal strength indicator (RSSI) in dB	-57.0	-111	-53	-64.37	8.28
sinr	signal-to-interference-plus-noise ratio (SINR) in dB	7.0	-20	30	10.57	9.96
rsrp	reference signal received power (RSRP) in dBm	-75.0	-141	-44	-83.03	14.11
rsrq	reference signal receive quality (RSRQ) value in dB	-7.0	-20	-3	-11.59	2.25
cell_id	Cell ID	5123075				
pci	LTE Physical Cell Identity	41.0				
netmode	reported network mode by Huawei HiLink API	19.0				
datarate	instantaneous data rate in bit s^{-1}	2498240.0	0	153 960 430	40 721 056	31 297 178

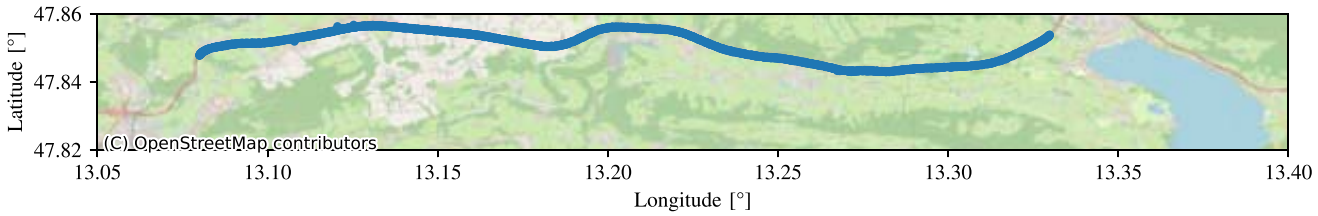

FIGURE 1. Map of the region of interest with a plot of the location of our data points.

of two years from January 2018 to December 2019. This section consists of a mix of urban highway near the city of Salzburg and rural highway in the Salzkammergut region. Fig. 1 shows the region of interest on a map. January 2018 to December 2019 is the most recent long term period in which mobile phone usage behavior was not strongly influenced by the COVID pandemic. It has been found that during the pandemic, mobile communications data shows very atypical network traffic patterns due to significant changes in user movement patterns [13], [14], [15].

We have preprocessed the raw data collected from the measurement scripts to (1) only include data from the relevant geographic area, (2) interpolate missing positions, (3) exclude data not collected on the highway, (4) removed data where our measurement hardware malfunctioned, and (5) removed measurements which were from non-LTE networks.

Often the drive tests are conducted during rush hour when high cell loads could be expected. But overall, conceptually there is no strict measurement schedule and there are drive tests at all times of day and during most of the year. Fig. 2 shows the histograms of the number of collected data points grouped by the months of the two years and time of day. Due to the absence of strict measurement scheduling, the data points are not uniformly distributed over the year; especially during the summer months August and September there are fewer data points. Nonetheless, for each month there is a minimum of 2000 data points.

As described in Section III-A, identical measurement hardware is used for the entire data collection period. However, during such long periods the network infrastructure is not static, for example, new cells may be installed or others removed by the network operator. Due to this fact–and possible behavioral changes of the network users over the two year period–some network metrics change significantly as well.

Our data set is based on travel patterns of a typical commuter and may therefore neither be representative for the cellular network in question, nor be generalizable to other networks or other car usage patterns, which differ among genders [27].

C. ANONYMIZATION

When publishing data sets anonymization is always a crucial aspect and often privacy concerns prevent the publication of data sets in the first place. Our drive test measurements are sourced from a mix of dedicated measurement drives and private commuting drives. In the case of the private

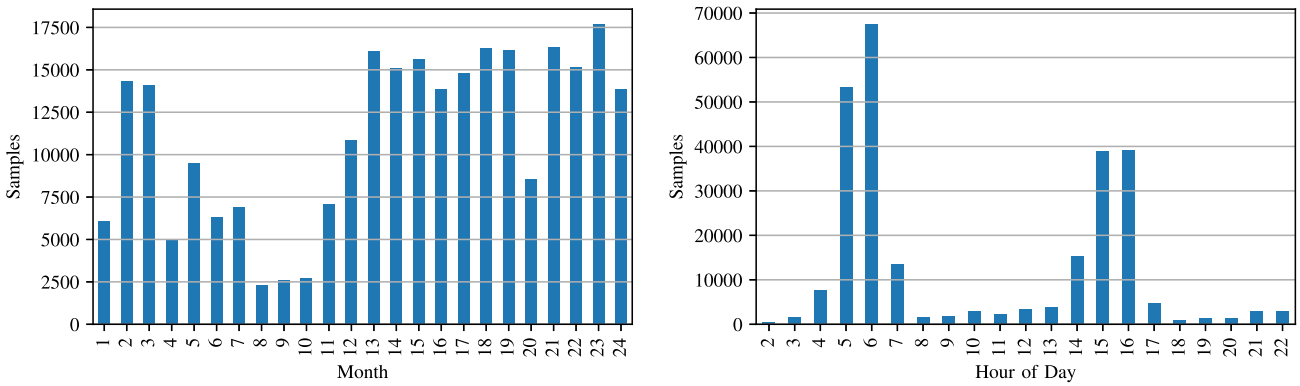

FIGURE 2. Histograms of the number of collected data points per month starting with January 2018 (left) and hour of day in UTC (right).

commuting drives, the vehicle is also previously equipped with our measurement system.

Our first action of anonymization is to restrict the data set to a fixed geographical region. This eliminates the possibility of deriving general movement profiles. Additionally, we remove all identifying data, like measurement IDs. Another anonymization step would be to remove the exact time data. Such data might allow recognizing patterns in the anonymized data and one might still be able to map drive tests to distinct individuals, in some cases. However, for our following evaluation we obtain the permissions of the individuals, to use the exact time data, and this allows for the data set to be used for a wider range of applications.

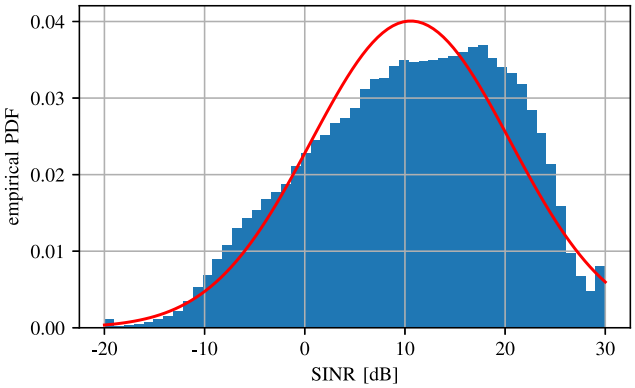

FIGURE 3. SINR distribution of the data set and Gaussian fit $\mathcal{N}(10.57367, 9.95682^2)$.

IV. EVALUATION AND VALIDATION

Next, we evaluate the data set from Section III and present its statistics to gain an in-depth insight about its properties and limitations. We perform an autocorrelation analysis in Section IV-A with respect to time, space, as well as time and space. In Section IV-B we propose an approach for validation using a third party data set covering the same time interval and geographical area.

Due to the non-systematic data collection, the distances between base stations and measurement equipment highly depend on the actual base station locations and the movement trajectories and hence the SINRs. Our measurements contain 89 LTE cells in total. Note, that the present cells change over the time span of the data set. However, we find that for our data set the actual SINR distribution resembles the log-normal distribution, hence the Gaussian distribution

$$\text{SINR} \sim \mathcal{N}\left(10.57367, 9.95682^2\right), \tag{1}$$

shown in Fig. 3 in logarithm domain, where the Gaussian distribution with mean μ and variance σ^2 is denoted by $\mathcal{N}(\mu, \sigma^2)$. The download data rate is influenced by a variety of factors, such as physical parameters (e.g., received power, noise power), cell load (since the resources are shared between multiple users), and configuration (e.g., tariff limits, network planning). In our data set, the distribution of the download data rate r shows no clear simple distribution, but is similar to an exponential distribution. In theoretical models of wireless communication, data rates are often exponentially distributed, but practically truncated by the concrete implementation. The exponential fit,

$$f_{\text{r}}(r) \approx \begin{cases} \frac{1}{40721056} \exp\left(-\frac{1}{40721056}\, r\right), & r \geq 0; \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

is shown in Fig. 4. The maximum observed download data rate of 150 Mbit s^{-1} is due to a tariff limit and does not change over the measurement time interval of 24 months.

However, the mean data rate increases abruptly and significantly in October 2018 and then gradually increases further over time (see Fig. 5). This can be explained by changing mean cell loads and by changes in the network topology due to network expansion. Fig. 6 shows that the number of distinct cells seen per month more than doubles over the observed measurement interval of 24 months. At the beginning of 2018, there are around 25 distinct cells. This number continuously increases to over 50 by the end of 2019.

A. AUTOCORRELATION

Due to the nature of the data collection procedure (moving car on freeway) we cannot separate the signal characteristics by purely temporal correlation and spatial correlation

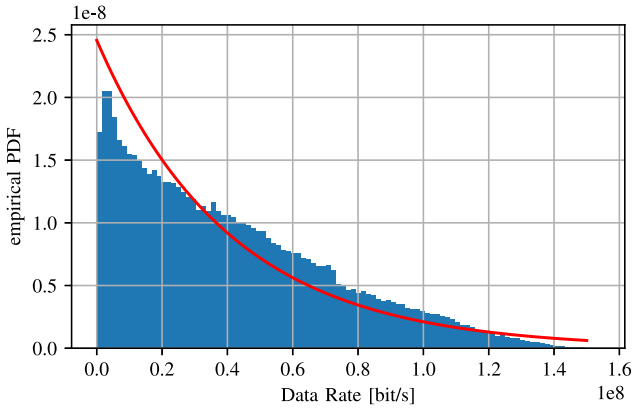

FIGURE 4. Download data rate distribution of the data set and exponential fit $\lambda = 1/40721056$.

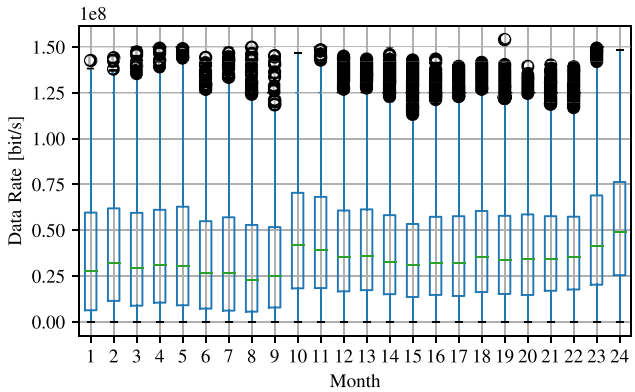

FIGURE 5. Box plot of download data rate development over the time span of 24 months starting with January 2018.

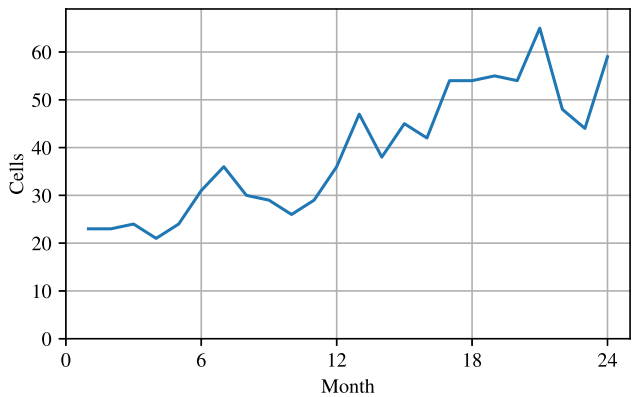

FIGURE 6. Development of the number of unique cells seen per month over the time span of 24 months starting with January 2018.

components. On the contrary, the autocorrelation function highlights this correlation of the spatial and temporal data points.

1) AUTOCORRELATION OF DATA RATE DEPENDING ON TIME

To visualize the autocorrelation of the data rate depending on time, we determine pairs of data points that are a given time difference apart from each other (0 s to 300 s), and calculate the correlation coefficient of the data rate (Pearson's r) for these pairs. The resulting correlation coefficient for each time difference from 0 s to 300 s (resp. 0 s to 60 s) is shown in Fig. 7. As expected, the autocorrelation of data rate is high for short time lags and lower when longer time lags up to 60 s are considered. However, the peak at $\approx$ 110 s hints at a weak cyclic pattern.

2) AUTOCORRELATION OF DATA RATE DEPENDING ON SPACE

Our strict 1 s sampling interval renders the data set geo-temporally irregularly sampled. This irregularity requires some additional processing in the step of marginalizing the temporal information and calculating the spatial autocorrelation. To achieve this, four major approaches exist [28]: (a) direct transform methods, (b) slotting techniques, (c) model-based estimators, and (d) time series reconstruction methods. In the latter two cases, detailed model information is needed to successfully apply these methods. However, our real world data does not correspond to any easily derivable model. Direct transform methods are often spectral estimation methods, but our data lacks a meaningful spectral representation. By contrast, slotting techniques do not require any model or spectral representations. The straight-forward slotting approach bins the original samples in suitable sizes and then calculates the correlation of the regularly sampled data due to the previous binning. Others suggest another way of slotting, where the sample correlation values are directly calculated from the original irregularly sampled data [29]. These sample correlation values are then discretized into bins, from which the (discrete) correlation function is calculated. However, both ways of slotting sometimes significantly distort the correlation function. In these cases it might be better to use so-called weighted slotting where there are no discrete slots (this is equivalent to a rectangular kernel) but softer suitable kernel functions, such as sinc or Gaussians.

For what follows we use the direct slotting technique where the original spatial dimension is slotted to bins of a size that is equivalent to the average movement of 1 s. This allows us to more directly observe the relation between time and space.

In our data set, the primary direction of movement is in longitudinal direction (see Fig. 1). Hence, only the longitude is used for the autocorrelation of the data rate subject to space. This allows to calculate the autocorrelation based on spatial distance analogous to the autocorrelation based on time, and to visualize the mixed spatial and temporal correlation in two dimensions.

Similar to the autocorrelation subject to time, we now search for time-space pairs with a certain difference in space. In contrast to time, where the time difference between two data points is already a multiple of a second, the spatial distance between two measurements is not a whole number but is continuous. The spatial distance that is driven on average in one second is $\approx 0.000\,37°$ longitude ≈ 28 m. We

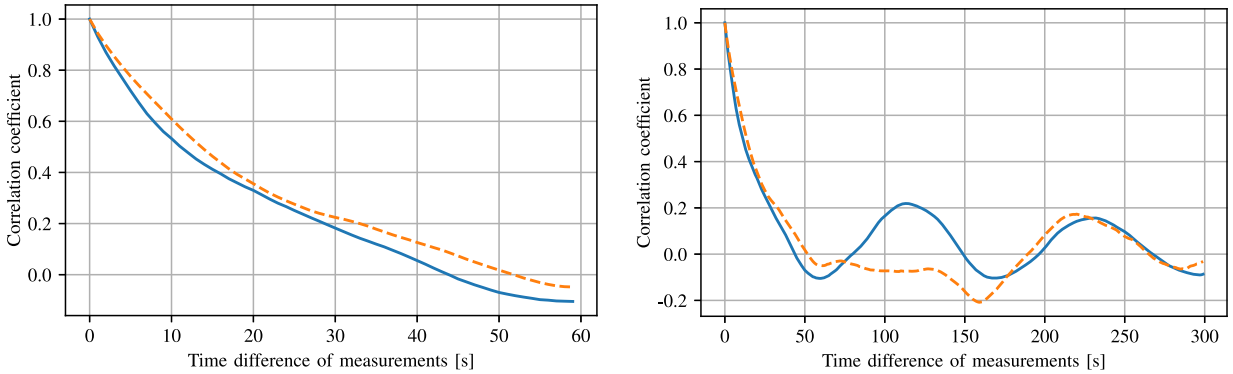

FIGURE 7. The correlation of the data rate (blue solid line) and RSSI (orange dashed line) of two measurements depends on the *time* between the measurements. The left short-term figure shows the expected reduction of correlation over time; the right long-term figure shows a cyclic pattern.

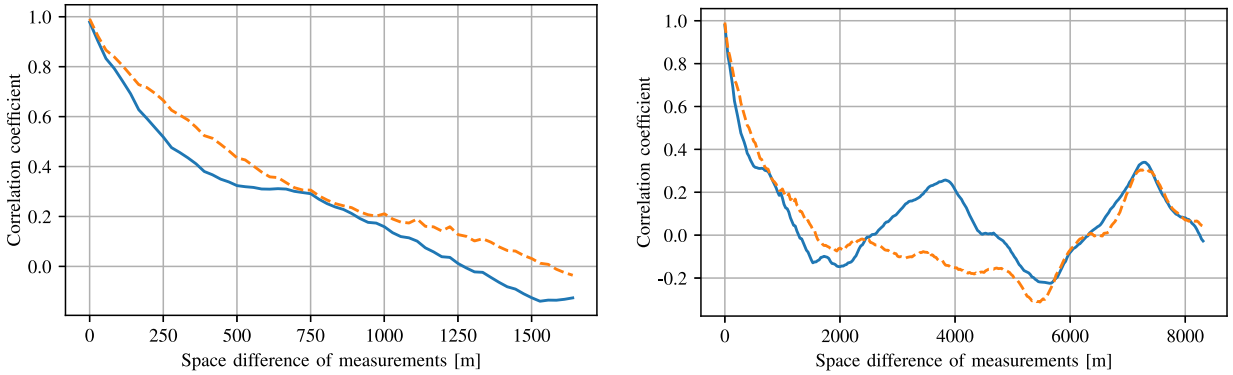

FIGURE 8. The correlation of the data rate (blue solid line) and RSSI (orange dashed line) of two measurements depends on the *space* between the measurements. The left short-distance figure shows the expected reduction of correlation over space; the right long-distance figure shows a cyclic pattern.

slot the distance differences to bins of multiples of this value. Furthermore, only pairs from the same trip are considered. Similar to the autocorrelation subject to time, Fig. 8 shows the correlation coefficient computed for given distances.

The general behavior of the correlation coefficient subject to space is similar to the general behavior of the correlation coefficient subject to time with high correlation for data rates with short spatial difference and lower correlation for data rates with larger spatial difference as well as hints at a cyclic pattern for even longer differences. We do not investigate the cyclic pattern further. Note that in our data set these two autocorrelations depend on each other, since the change of space and the change of time are highly correlated.

3) AUTOCORRELATION OF DATA RATE DEPENDING ON TIME AND SPACE

To determine if the correlation is based on changes in time or changes in space, Fig. 9 shows the autocorrelation of the data rate depending on time and space. Each correlation coefficient value is calculated from data point pairs with the corresponding difference in time and space. The heatmap shows the strength of the correlation coefficient of the data rate for each combination of time and spatial interval. It shows that as long the change in location is less than approximately 200 m the correlation of the data rate is high even for relatively longer times.

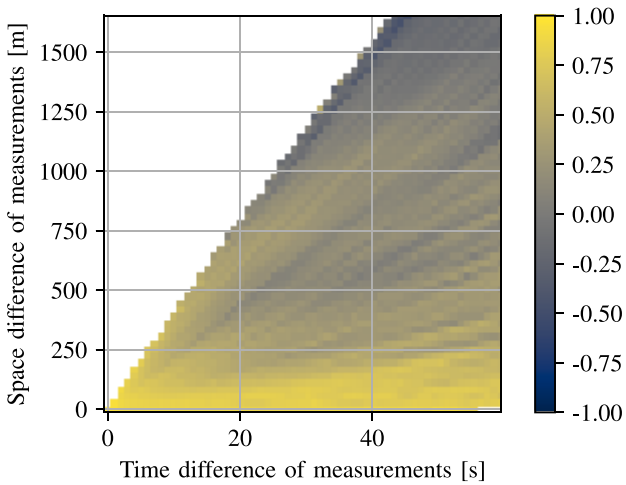

FIGURE 9. The correlation of the data rate of two measurements depends on the *time* and *space* between the measurements.

B. VALIDATION

We consider the best way to validate a data set is to use an approach that is as different as possible, from the process that generated the first data set. Because the controlled data sets and crowdsourced data sets differ in their most basic structure, we decide to use a crowdsourced data set to

TABLE 2. The correlation coefficient of the signal parameters between the two data sets depends on the parameter and the method to select the value that represents a location (median or mean). Positive values correspond to a higher value in the Ookla data set.

Parameter	Mean(diff(mean))	Mean(diff(std))	Median(diff(median))	Correlation(median)	Correlation(mean)
sinr	1.33	2.99	1.00	0.95	0.92
signal	-0.73	-0.13	-1.00	0.77	0.91
rsrq	2.51	1.24	3.00	0.85	0.91
rsrp	-3.56	3.92	-4.00	0.98	0.95

validate our controlled data set. In our case this, for example, means that the validation data set uses different devices, whereas our data set only uses a single device and while our measurements are predominantly from commuting the validation data set covers a wider range of day times.

To determine how representative our data set is for a wider range of devices and usage/movement patterns, we compare our data set with a Speedtest Intelligence data set provided to us by Ookla. Ookla uses its Speedtest app to collect performance data for Internet access worldwide. We filter the data from Ookla to the same provider, locations (see Fig. 1) and time window (2018-01-01 to 2019-12-31) as our measurements. We then compare general statistical properties of both data sets. However, because the spatial distribution differs between both data sets, a direct comparison of overall properties is not meaningful. Instead, we compare the properties for groups of data points with similar location. When grouping the measurements into bins of 0.001° width of longitude (≈ 75 m for our location), the median and mean signal values for each bin of both data sets are similar. For example, the correlation coefficient of the median SINR is 0.95. The mean SINR is correlated with 0.90. Table 2 shows a wider selection of comparisons between these data sets (see Table 1 for reference values of our data set). In addition to the correlation coefficient, it shows the mean/median of the difference of the mean/median/standard deviation in a bin. Unfortunately, our comparison data set does not include RSSI values, so we cannot use these to determine the similarity of the data sets.

We also compare the measured data rates in our data set with the Ookla data set. However, because of too few data points and different (partly unknown) data rate limits in underlying contracts, we do not consider this comparison meaningful and therefore, do not report details.

V. MACHINE LEARNING

We use our data set to predict the data rates, because these are costly to measure, both in terms of load on the network and financially in case of limited tariffs. The prediction can be used, for example, to estimate possible data rates to select a suitable video quality for a video stream or to estimate possible data rates without the need to transfer data.

At first, we use classical predictors to get a baseline to compare against using classical statistical methods. Then we use machine learning to train neural networks to predict data rates. We evaluate the results using the mean absolute error (MAE) and the coefficient of determination (R^2).

Here, the MAE is defined as the arithmetic mean of the absolute differences of the predicted values from the measured data rates:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|,$$

where $\hat{y}_i$ is the predicted value for the i-th data point, y_i is the correct value and n is the number of data points. R^2 measures the goodness of fit of a model to predict the measured data rates, where $R^2 = 1$ means perfect fit:

$$R^2 = \frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2},$$

where $\bar{y}$ is the mean of all y values.

A. CLASSICAL

To provide reference points for machine learning, we use classical (non-machine-learning) methods, such as constant values and lookup tables.

As the simplest possible predictors, we use constant value estimates (`const(median)` and `const(mean)`) that provide trivial predictions for the data rate. They provide a mean absolute error of approximately 25 Mbit s^{-1}. A simple prediction using a signal parameter is a lookup table from the selected signal parameter to the median value seen in this group (stratification). The MAE of such a predictor depends on the specific parameter used for predicting the data rate and is around 18 Mbit s^{-1} for rssi, rsrq, and rsrp, and 15 Mbit s^{-1} for sinr.

To provide a prediction based on the location, we group the measurements based on their rounded longitude, because our track does not have much variability along latitude. We round to 2, 3, 4 or 5 decimals, which correspond to groups of approximately 750 m, 75 m, 7.5 m, 0.75 m diameter. This class of predictors achieves a MAE of approximately 14 Mbit s^{-1} for the smallest groups. The last class of classical predictors use the previously measured data rate to predict the current data rate. Using the data rate measured in the previous second to predict the data rate for the current second gives a MAE of approximately 8 Mbit s^{-1}. Predicting the mean of the next and the previous second for the current value (which can probably only be used to interpolate missing values) provides an MAE of approximately 6 Mbit s^{-1}.

Table 3 provides the overview of classical predictions. We do not split the data into training and test data as for machine

TABLE 3. The quality of the classical predictions vary strongly based on the method.

Strategy	MAE [Mbit s^{-1}]	R^2
`const(mean)`	25.71	0.00
`const(median)`	25.25	-0.04
`sinr-Lookup`	15.44	0.55
`signal-Lookup`	17.26	0.43
`rsrq-Lookup`	18.62	0.33
`rsrp-Lookup`	17.44	0.43
`rssi-Lookup`	18.94	0.34
`long-Lookup2`	19.48	0.31
`long-Lookup3`	15.72	0.51
`long-Lookup4`	15.38	0.52
`long-Lookup5`	14.26	0.54
`prevDR`	7.93	0.85
`meanDR`	5.79	0.92

learning, because the degrees of freedom of the classical predictors is so low that it does not make any difference.

B. NEURAL NETWORK

To use more sources of information for a prediction, we use neural networks. For this we define several scenarios, which provide different amounts of information to the neural network:

- `optimal`: Contains all inputs for an optimal prediction, that is, the data rates of the previous three seconds, the data rates for the following three seconds (but not the data rate for the current second, which the neural network has to predict), the position of the measurement device, and the set of parameters related to signal strength. This might be useful to interpolate missing values. Additionally, we use it as a benchmark for comparison.
- `next`: Contains all inputs available, while transferring data to predict future data rates, this is, positions, signal strengths and the three previous data rates. It can be used to predict the optimal quality selection for the next segment of a streamed video.
- `nodata`: Contains all inputs to predict data rates, while no data is being transferred; that is, no data rates, but only positions and signal strengths. This can be used to predict data rates while no data is being transferred.
- `local`: Contains inputs which are not specific to our area and thus could be applied at another location; in our case, these are the signal strength parameters (rsrq, rsrp, rssi, sinr, signal).
- `map`: Contains inputs that can be provided to the neural network without physically being at a location such as longitude and latitude, that is, an enhanced coverage map.

Additionally, we add time of day and days since start of the data set to all scenarios. Velocity and a one-hot encoding of the currently connected cell ID to all is included in all scenarios except `map` and `local`. Occasionally, some parameters might not be available in the measurement. For every input to the neural network we add an input ending with "_na", that is 1 exactly when the parameter on which the input is based is not available in our data set. In this case, the original (missing) input is replaced by the mean value.

TABLE 4. Search space for hyperparameters of the trained feed-forward neural networks.

Hyperparameter	Search space
Hidden layers	[1, 2, 3, 4]
Neurons per layer	[16, 64, 512, 2048]
Activation function	ReLU
Batch size	[8, 32, 128, 512]
Training epochs	10
Optimizer	Adam
Learning rate	[1e-4, 1e-3, 1e-2, 1e-1]
Loss function	Mean absolute error

We train neural networks to predict the data rate based on the given inputs of each scenario. We run the training and evaluation on a PC with an Intel Xeon Silver 4114 Processor with 64 GiB of RAM and an Nvidia GeForce RTX 2080 Ti GPU with 11 GiB GDDR6 RAM and Ubuntu 20.04 LTS as operating system using anaconda package management. For this, we create a feed-forward neural network using TensorFlow [30] in version 2.2.0. We normalize the inputs (to mean 0, and standard deviation 1), except inputs that only take values 0 and 1. To find good hyperparameters for the neural network, we execute a random search on the parameters described in Table 4. Further tests with other features (dropout layers, other activation functions, other ranges) do not improve the result. The full code and the hyperparameter configuration is available online.[2]

After 100 training runs for each scenario on 80 % of the data, we evaluate the neural networks on the other 20 %. Table 5 shows the best performing results for each scenario. The results show that the neural networks can beat the classical strategies described in Table 3. Depending on the exact comparison, the improvement is between 5 % (`optimal` compared to `meanDR`) and 28 % (`local` compared to `sinr-Lookup`).[3] This shows that the neural network is able to make better predictions than the classical predictions based on a single parameter.

We consider it likely that the neural network can improve over the classical predictions, because it has access to additional features (past data rates, positions and signal strength respectively). We will further analyze the influence of individual features in the next section.

2. https://github.com/mherlich/wireless-data-set

3. These compare the strategies, which use similar inputs for their prediction and are thus closest to each other. Other comparable strategies are `next` to `prevDR` (17 %) and `map` compared to `long-lookup5` (16 %). The `nodata` strategy has no direct correspondent classical strategy, because it combines the information from position and signal.

TABLE 5. The results of the best performing neural networks for each scenario.

Scenario	Inputs				MAE	R^2
	Data rates		Positions	Signal strengths	[Mbit s^{-1}]	
	Future	Past				
optimal	✓	✓	✓	✓	5.48	0.93
next		✓	✓	✓	6.57	0.90
nodata			✓	✓	9.58	0.80
local				✓	11.04	0.74
map			✓		11.97	0.71

C. EXPLAINING RESULTS

Since machine learning models are often black boxes, it is helpful to gain some insight into the model [31]. There are many methods for trying to explain which input parameters affect the output and how large this impact is.[4]

One of those methods is LIME (Local Interpretable Model-agnostic Explanation), which tries to explain the importance of the input parameters using a local linear surrogate model. Therefore, the LIME algorithm needs one input vector, for which the impact of the individual inputs should be explained. On the basis of this input vector, LIME builds the explainable surrogate model in a neighborhood of the input vector using training data and the trained model [32].

To explain the predicted data rate of our model, we employ a LIME method built for regression to increase the knowledge of the input leverage. Because LIME only estimates the effect of one input vector, we compute the mean of the absolute values of the LIME output over 1000 random samples of the data set. Thus, we generate an overview of the effect that the inputs have on the predicted data rate on average. The most influential inputs for the different scenarios are illustrated in Fig. 10, where for every scenario the neural network with the smallest MAE is used as model in the explanation. We include discrete inputs in the analysis, even though the results of their LIME influences are hard to interpret.

As represented in the bar plot for the scenario, `optimal` (Fig. 10(e)) the input parameter prevDR1 exerts most influence followed by the input parameters cell_id_0 and succDR1. Here, prevDR1 corresponds to the data rate measured one second before the data rate that is being predicted. And succDR1 represents the data rate measured one second after the data rate that is being predicted. Because the cell_id parameters only take values 0 and 1 showing whether the measurement takes place in this cell or not, we are unsure how well the LIME explanation reflects their true influence. The great importance of the parameters prevDR1 and succDR1 aligns well with the observation that the autocorrelation of data rates is high for low time differences (Fig. 7) and the classical strategy `meanDR` (Table 5) results in similar precision.

4. See https://explainml-tutorial.github.io/ for an introduction into explainable AI methods.

The bar plot for the scenario `next` (Fig. 10d) shows that the most influential feature is again prevDR1. This demonstrates that the data rate measured directly before the predicted data rate remains an important input even if the succDR1 is not an input for the neural network. These observations fit the observation that the classical strategy `prevDR` (Table 5) is very similar.

Although the signal strength parameter sinr is only part of the most influential parameters in the scenarios `optimal` and `next`, this parameter heads the feature influence in the scenarios `nodata` (Fig. 10(c)) and `local` (Fig. 10(b)). Since the sinr value gains more importance in the scenario `next` than in the scenario `optimal`, it seems this input parameter accomplishes its significance through the loss of the inputs of the measured data rate as it is given in the scenarios `nodata` and `local`.

Due to the fact that there are no signal strength and data rate inputs in the scenario `map` (Fig. 10(a)), the predicted data rate depends on the parameters lat, long, time_of_day, ele and days_since_start, mostly influenced by the coordinates. Though the longitude coordinate has a wider range, the LIME explanation reveals that a small change in the latitude coordinate changes the prediction more than the longitude coordinate (both measured in standard deviations). Additionally, the LIME explanation reveals that the position as well as the daytime of measurement are more important for predicting the data rate than the input parameter ele and days_since_start.

To summarize, the explained influence of the input parameters is consistent with the results of the scenarios (Table 5), since the mean absolute error increases if the most important input parameters are removed, for example prevDR1 from scenario `next` to scenario `nodata`. Additionally, the LIME explanations support the hypothesis that the neural networks work similar to the classical strategies, but use additional features to nuance the predictions.

D. AUTOENCODER FOR SIGNAL STRENGTH PARAMETERS

Our data set contains five parameters, which describe the received signal strength: rssi, rsrp, rsrq, sinr, signal (see Table 1). Fig. 11 shows the empirical probability density functions of these parameters after removing all data points that do not have values for all five signal parameters. As one can observe, the distributions and in particular the domains are different. However, one would suspect that there are relations between these values. In this section, we explain how to use an autoencoder neural network [33] to determine these relations.

To achieve this, we omit those samples where at least one of the signal strength parameters is not provided (which is indicated by N/A in our data set). This reduces the number of samples in the data set from 267 198 to 144 226. Again, we split the data set into training and evaluation data, with 80 % of the samples used for training and 20 % for evaluation. With the former, we train a neural network with 3 hidden

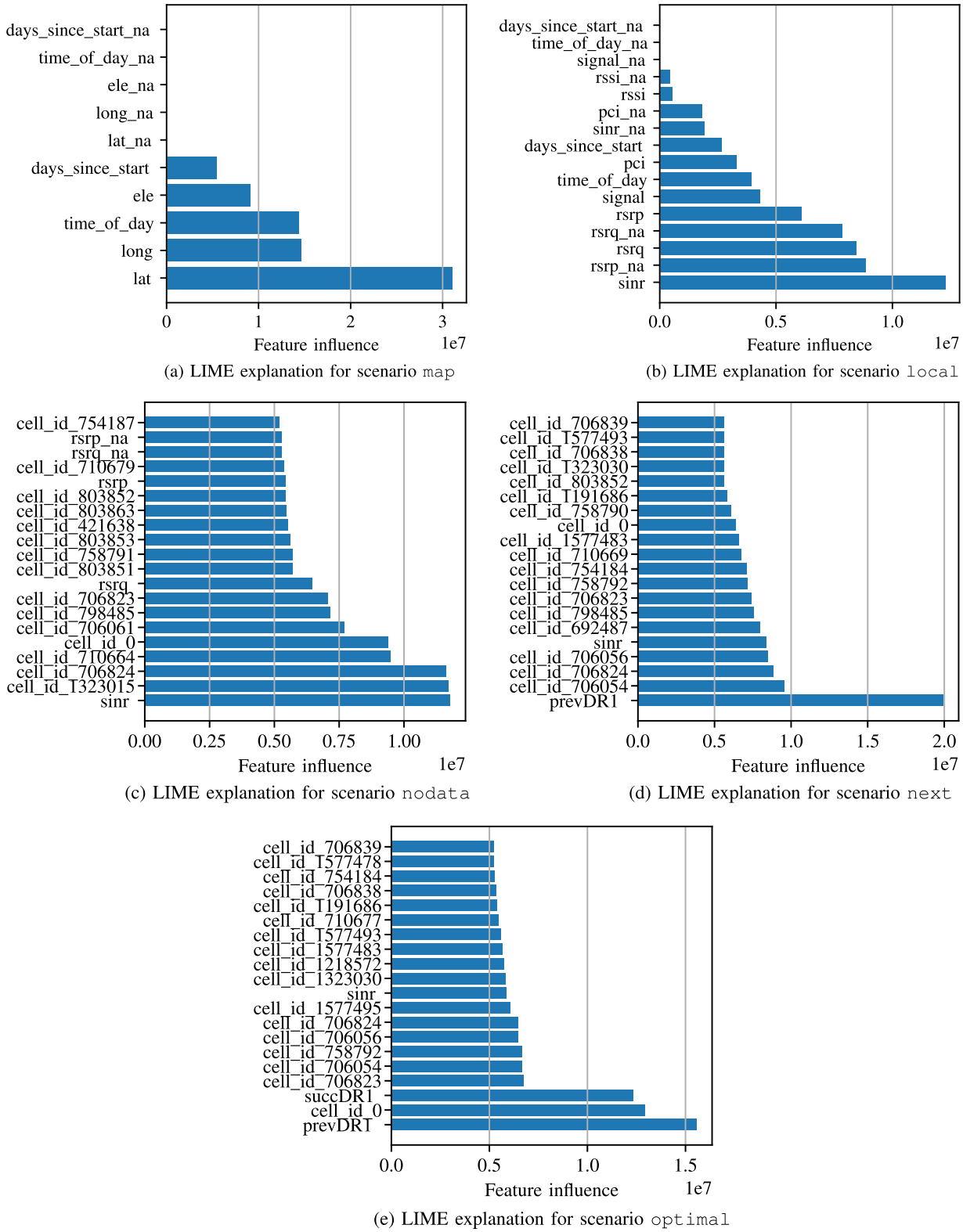

FIGURE 10. **The absolute values of the LIME explanation of all input parameters for the scenarios `map` (Fig.a) and `local` (Fig.b) as well as the LIME explanation of the 20 most influential input parameters for the scenarios `nodata` (Fig.c), `next` (Fig.d) and `optimal` (Fig.e) averaged over 1000 samples of training data.**

layers. This neural network consists of 20 neurons in the first and third hidden layer, and between 1 and 5 neurons in the second hidden layer. Thus, the second hidden layer serves as a bottleneck layer. We use linear activation functions for the bottleneck and output layer, and rectified linear unit (ReLU) activation functions for the other two hidden layers. The

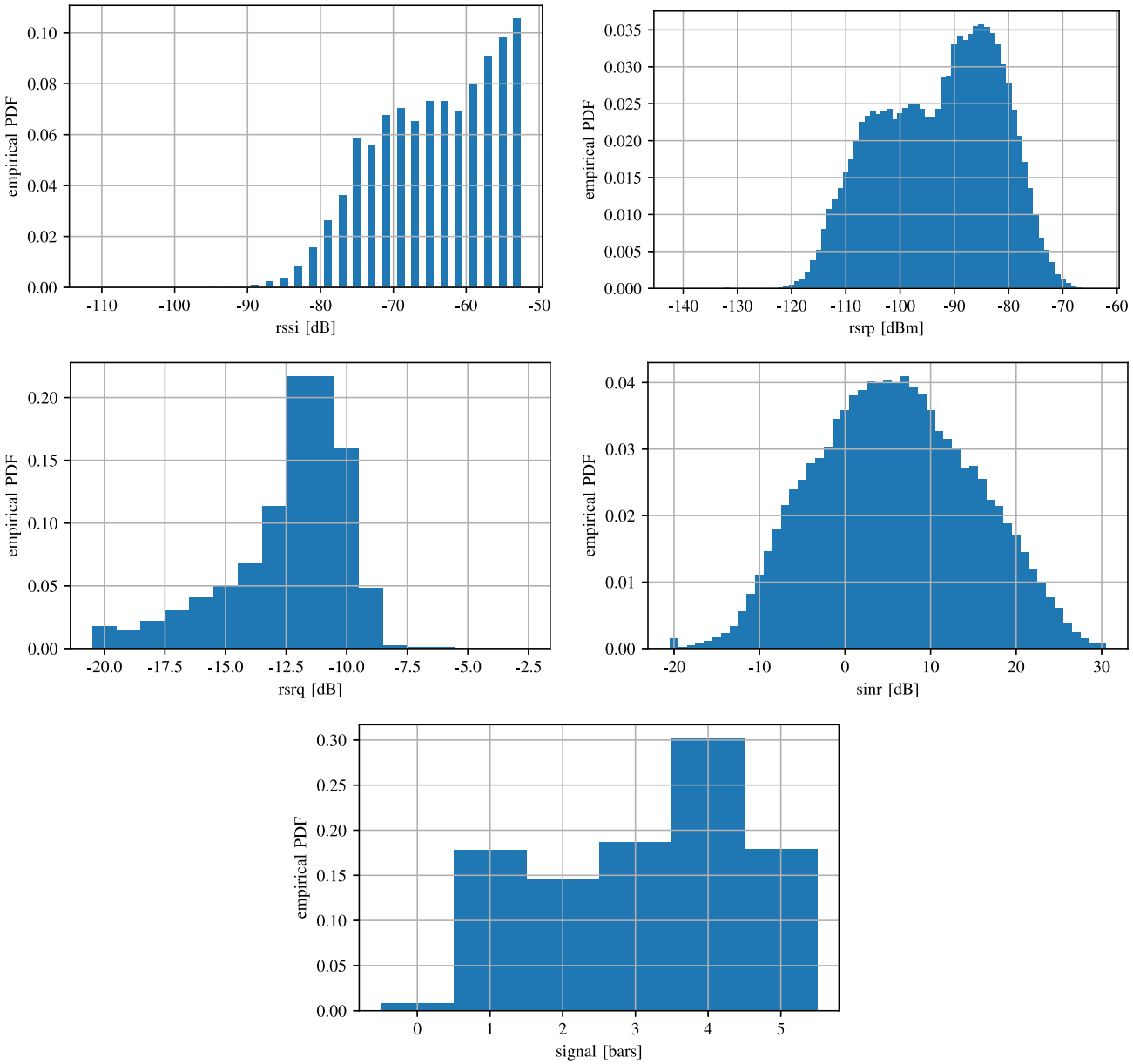

FIGURE 11. Empirical probability density functions of received signal strength parameters. For these plots, only those samples in our data set are used which have valid values for all signal strength parameters.

data is passed to the neural network as it is, that is, not normalized. We train the neural network for 10 epochs with a batch size of 128. We use the Adam optimizer [34] with a learning rate of 0.01, and MSE as loss function. The results are in Table 6. In addition, we try other hyperparameter configurations (e.g., more hidden layers), but the results are essentially the same.

TABLE 6. Evaluation of autoencoder neural network for signal strength parameters.

Bottleneck neurons	MSE	MAE	R^2
5	0.022	0.127	0.997
4	0.155	0.219	0.928
3	0.443	0.383	0.881
2	2.106	1.088	0.868
1	8.057	1.977	0.754

VI. CONCLUSION

This paper presents our open controlled drive test data set of mobile communications and a tool chain for measurement, collection, evaluation, and prediction. The open data set can be used by others in the future to test and improve machine learning for wireless communication. To facilitate this, we provide a statistical evaluation of the data set and validate it with another data set. We then outline how this data set can be used for the development of machine learning based prediction methods by providing examples for prediction

of the data rate, using LIME to explain the results, and describe an autoencoder for the signal strength parameters. We show an approach to obtaining communications data sets that can be shared openly. We believe this can be one building block to overcome the historic lack of suitable long-term mobile communications open data sets and accelerate further research in this field.

Possible future applications include prediction of data rates based on recurrent neural networks. However, to do this, it is first necessary to handle gaps in the data. To gain better understanding of the interaction of space and time on the data rate, a method to generate multiple measurements at the same time at different locations might be useful. Alternatively, a mediation analysis could be carried out.

REFERENCES

[1] "Speedtest by Ookla—The global broadband speed test." Ookla. 2021. [Online]. Available: https://www.speedtest.net/

[2] "Opensignal: Mobile analytics & insights." Opensignal. 2022. [Online]. Available: https://www.opensignal.com/

[3] "Cellular coverage and tower map." CellMapper. 2022. [Online]. Available: https://www.cellmapper.net/

[4] "Access network speed tests: OECD digital economy papers." OECD. 2014. [Online]. Available: https://www.oecd-ilibrary.org/content/paper/5jz2m5mr66f5-en

[5] "RTR-Netztest." Rundfunk und Telekom Regulierungs-GmbH. 2021. [Online]. Available: https://www.netztest.at/de/Test

[6] "Breitbandmessung." Bundesnetzagentur. 2021. [Online]. Available: https://breitbandmessung.de/

[7] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen, "From LTE to 5G for connected mobility," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 156–162, Mar. 2017.

[8] D. Minovski, N. Ogren, C. Ahlund, and K. Mitra, "Throughput prediction using machine learning in LTE and 5G networks," *IEEE Trans. Mobile Comput.*, early access, Jul. 26, 2021, doi: 10.1109/TMC.2021.3099397.

[9] M. Laner, P. Svoboda, S. Schwarz, and M. Rupp, "Users in cells: A data traffic analysis," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2012, pp. 3063–3068.

[10] F. Li, X. Jiang, J. W. Chung, and M. Claypool, "Who is the king of the hill? Traffic analysis over a 4G network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[11] M. Herlich and S. Farthofer, "wireless communication data sets for machine learning," in *Proc. 2nd KuVS Fachgespräch Mach. Learn. Netw.*, 2020, pp. 1–2. [Online]. Available: https://www.salzburgresearch.at/wp-content/uploads/2020/10/Wireless-Communication-Data-Sets-for-Machine-Learning.pdf

[12] A. Palaios *et al.*, "Network under control: Multi-vehicle E2E measurements for AI-based QoS prediction," in *Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2021, pp. 1432–1438.

[13] G. Heiler *et al.*, "Country-wide mobility changes observed using mobile phone data during COVID-19 pandemic," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2020, pp. 3123–3132.

[14] F. Schlosser, B. F. Maier, O. Jack, D. Hinrichs, A. Zachariae, and D. Brockmann, "COVID-19 lockdown induces disease-mitigating structural changes in mobility networks," *Proc. Nat. Acad. Sci.*, vol. 117, no. 52, pp. 32883–32890, Dec. 2020. [Online]. Available: https://europepmc.org/articles/PMC7776901

[15] B. Jeffrey *et al.*, "Anonymised and aggregated crowd level mobility data from mobile phones suggests that initial compliance with COVID-19 social distancing interventions was high and geographically consistent across the U.K.," *Wellcome Open Res.*, vol. 5, p. 170, Jul. 2020.

[16] V. Raida, P. Svoboda, M. Lerch, and M. Rupp, "Crowdsensed performance Benchmarking of mobile networks," *IEEE Access*, vol. 7, pp. 154899–154911, 2019. [Online]. Available: https://publik.tuwien.ac.at/files/publik_282522.pdf

[17] J. Cainey, B. Gill, S. Johnston, J. Robinson, and S. Westwood, "Modelling download throughput of LTE networks," in *Proc. 39th Conf. Local Comput. Netw. Workshops*, 2014, pp. 623–628.

[18] T. Hoßfeld *et al.*, "White paper on crowdsourced network and QoE measurements—Definitions, use cases and challenges," Universitätsbibliothek Würzburg, Würzburg, Germany, Working Paper, 2020. doi: 10.25972/OPUS-20232

[19] B. Sliwa, H. Schippers, and C. Wietfeld, "Machine learning-enabled data rate prediction for 5G NSA vehicle-to-cloud communications," in *Proc. IEEE 4th 5G World Forum (5GWF)*, 2021, pp. 299–304.

[20] K. Saija, S. Nethi, S. Chaudhuri, and R. Karthik, "A machine learning approach for SNR prediction in 5G systems," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2019, pp. 1–6.

[21] A. Kousaridas *et al.*, "QoS prediction for 5G connected and automated driving," *IEEE Commun. Mag.*, vol. 59, no. 9, pp. 58–64, Spr. 2021.

[22] L. Mei *et al.*, "Realtime mobile bandwidth prediction using LSTM neural network," in *Proc. Int. Conf. Passive Act. Netw. Meas.*, 2019, pp. 34–47.

[23] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, "LinkForecast: Cellular link bandwidth prediction in LTE networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1582–1594, Jul. 2018.

[24] D. Raca *et al.*, "On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 11–17, Mar. 2020.

[25] S. Farthofer, M. Herlich, C. Maier, S. Pochaba, J. Lackner, and P. Dorfinger. "CRAWDAD dataset SRFG/LTE-4G-highway-drive-tests-salzburg (v. 2022-01-18)." Jan. 2022. [Online]. Available: https://crawdad.org/srfg/lte-4g-highway-drive-tests-salzburg/2022-01-18

[26] C. Brandauer and T. Fichtel, "MINER—A measurement infrastructure for network research," in *Proc. 5th Int. Conf. Testbeds Res. Infrastruct. Develop. Netw. Commun. Workshops*, 2009, pp. 1–9.

[27] R. Crane, "Is there a quiet revolution in women's travel? Revisiting the gender gap in commuting," *J. Amer. Plan. Assoc.*, vol. 73, no. 3, pp. 298–316, 2007.

[28] K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths, "Comparison of correlation analysis techniques for irregularly sampled time series," *Nonlinear Processes Geophys.*, vol. 18, no. 3, pp. 389–404, 2011.

[29] R. A. Edelson and J. H. Krolik, "The discrete correlation function—A new method for analyzing unevenly sampled variability data," *Astrophys. J.*, vol. 333, pp. 646–659, Oct. 1988.

[30] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.

[31] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, 2018, pp. 80–89.

[32] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," 2016, *arXiv:1602.04938*.

[33] C. C. Tan, *Autoencoder Neural Networks: A Performance Study Based on Image Reconstruction, Recognition and Compression*. Köln, Germany: Lambert Acad. Publ., 2009. [Online]. Available: https://dl.acm.org/doi/book/10.5555/1795842

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

STEFAN FARTHOFER received the Dipl.-Ing. degree in telecommunications and the Dr.techn. degree in electrical engineering from Technische Universität Wien, Vienna, Austria, in 2014 and 2019, respectively. He is currently with the Intelligent Connectivity Department with Salzburg Research, where he is working on performance measurements in 5G mobile networks and prediction and analysis of wireless communication quality.

MATTHIAS HERLICH received the doctorate degree in computer science from the Universität Paderborn, Germany, in 2014. After an internship with Palo Alto Research Center and a Postdoctoral Research Fellow with the National Institute of Informatics, Tokyo, he went to Salzburg Research. His experience ranges from analytic approaches and simulations to evaluation of measurements. His current focus is the measurement and prediction of wireless communication quality.

CHRISTIAN MAIER received the B.Sc. degree in mathematics from the Technical University of Munich, Germany, in 2014, and the M.Sc. degree in mathematics from the Ludwig Maximilian University of Munich, Germany, in 2021. He is currently employed as a Data Scientist with Salzburg Research, Austria. His work focuses on the application of machine learning methods (mostly based on artificial neural networks) to networking.

SABRINA POCHABA passed a state examination in the subjects mathematics and biology from University Ulm, Germany, in 2018. She received the master's degree in mathematics with Ruprecht-Karls-University, Heidelberg, Germany, in 2021. She is currently working as a Data Scientist with Salzburg Research. There she is engaged in different machine learning methods focusing on networks and communication.

JULIA LACKNER is currently pursuing the master's degree in data science with the University of Salzburg, Austria and did an internship with Salzburg Research.

PETER DORFINGER received the diploma degree from the Department of Telecommunications Engineering, Salzburg University of Applied Sciences, Austria, in 2002, and the master's degree from the Department of Information Technologies and Systems Management, Salzburg University of Applied Sciences in 2010. He is currently with Salzburg Research. His research interest is focused on wired and wireless critical infrastructure networks.