Visualizing Mobile Coverage from Repetitive Measurements on Defined Trajectories

Chad Jarvis and Cise Midoglu
Simula Research Laboratory
Oslo, Norway
Email: [chad,cise]@simula.no

Andra Lutu
Telefonica Research
Barcelona, Spain
Email: andra.lutu@telefonica.com

Ozgu Alay
Simula Metropolitan Center for Digital Engineering
Oslo, Norway
Email: ozgu@simula.no

***Abstract*—Ensuring pervasive coverage of mobile networks and good quality of service are common goals for both regulators and operators. Currently, however, the evaluation of coverage is mostly limited to maps provided by Mobile Network Operators (MNOs). In this paper, we use the Measuring Mobile Broadband Networks in Europe (MONROE) platform to characterize mobile coverage along transport routes, reliably and in an objective manner. We leverage access to MONROE nodes onboard public transport vehicles: our unique geo-referenced dataset comes from nodes active on board 15 Norwegian inter-city trains that travel 13 different routes. The data from hundreds of train trips between 2017 and 2018 on each of the routes shows the mobile coverage status as travellers experience it. We propose an algorithm to segment the measurement routes to enable efficient grouping of data samples for analysis and visualization. We present our analysis and visualization of coverage along the railway routes. The proposed approach is generic so that other type of performance maps, including latency or throughput maps, can also be generated.**

I. Introduction

Mobile Broadband (MBB) networks have become the key infrastructure for people to stay online for entertainment, communication and work related tasks. One challenging use case for MBB networks is the mobility scenarios; especially, Internet access in public transport infrastructures such as inter-city trains. Mobility is becoming more and more relevant, since up to hundreds of passengers might try to access the Internet simultaneously while their train is moving at high speeds.

Assessing the mobile network coverage and performance experienced by passengers on critical public transport routes is of great importance to many stakeholders, including consumers, regulators, governments, MNOs and businesses that provide Internet services on trains. Today, regulators and end-users are left with coverage and quality maps provided by MNOs. These maps might not reflect passengers' experience correctly, since they often rely on theoretical models and not on empirically-driven approaches. Consequently, verifying these maps is often hard, since it requires performing repeated expensive drive tests. One alternative is to leverage crowdsourcing for verification, but unfortunately crowdsourced datasets can be spatially sparse and generally lack repeatability, which makes it hard to draw firm conclusions.

In this paper, we leverage the Norwegian State Railways (NSB) deployment of the MONROE platform [1] in Norway to analyze a *geo-referenced dataset* that mimics the measurements MNOs collect through repetitive drive tests. The MONROE NSB platform enables us to easily acquire a vast amount of data for two commercial MBB networks in Norway (Telia and Telenor), including the best Radio Access Technology (RAT) available at a given measurement point. Each measurement point we collect is characterized by variable spatio-temporal coordinates. The spatial dimension of the data designates the geo-location where the measurement device captures the connection information (e.g., best RAT available) at a moment in time. In our case, the train routes dictate the spatial coordinates of the data points we register in the dataset. Global Positioning System (GPS) readings from the train system are collected every 10 seconds, resulting in a large distance between two measurement point especially when the train is traveling at high speeds. Due to this temporal sparsity, it is not always possible to evaluate the RAT at the same constant location for every measurement drive run (GPS measurements are geographically irregular). Therefore, the set of geo-tagged data points collected at different drive runs varies, bringing additional complexity to our analysis.

The interaction of these two dimensions dictates the challenges of moving from acquiring the data to drawing knowledge through data analytics approaches. Previous approaches proposed to group the data points by overlaying a grid with fixed tile size over the area of interest and identifying the grid tiles that contained measurements samples [2]. This results in an irregular segmentation of the route of interest and one can extrapolate the characteristics of the data group to the entire tile area. However, this resulted in differences between well represented areas that contained a significant portion of route and others that have the route only tangential to the grid tile. In this paper, we propose an algorithm for cleaning and morphing the dataset such that we can easily group the final dataset based on spatial locality. In particular, we identify the train routes, we divide them into equal length segments and then group the geo-referenced data points in the initial dataset around these route segments.

The contributions we make in this paper are threefold:

- We present the details of the MONROE NSB deployment, which includes 15 MONROE nodes operating abroad 15 different passenger trains in Norway. Each node measures two MNOs in the same time using customer-grade subscriptions. The platform is open to the community for running measurements under mobility conditions. [1]

- We open the dataset we collected from operating the NSB testbed for a period of over one year, from January 2017 until January 2018 [3].
- We propose an algorithm to address the challenges of drawing knowledge from the vast dataset of repetitive drive runs over 13 routes of NSB passenger trains. Our approach allows us to segment the measurement routes to enable efficient grouping of data samples for analysis and visualization. We present our analysis and visualization of coverage along the railway routes on the one-year dataset we collected. We mention that we can extend this very approach to generate other type of performance maps, including latency or throughput maps, which we leave for future work. Our `R` implementation of the algorithm is further provided as open source software [3].

II. Background and Related Work

Building accurate and reliable coverage maps has attracted the attention of the research community and a magnitude of work exists in this area [4]. Coverage maps need to closely reflect actual end-user experience and use of measurements plays a vital role towards this end [5]. However, obtaining measurements across space and time has a high cost. Drive tests are widely used by MNOs for coverage assessment and performance monitoring. Piggy-backing MBB measurements onto public transport infrastructure is an efficient, cost-effective and automated alternative to traditional drive testing [6], [7], [8], [2]. Aside from the high cost of drive tests, the data collected from them usually has a series of shortcomings, including variable spatio-temporal sampling and limitation of test repeatability. The drawbacks of drive tests act as incentive for the design of new methodologies that address these issues [9], [10]. In this sense, our experimental setup brings the benefit of repeatability at a low additional cost. Other approaches, such as leveraging crowdsourcing platforms, may help verify coverage maps[11] or increase their accuracy by merging with controlled datasets [12]. However, they bring additional limitations including the lack of control on the measurement device and lack of repeatability.

Specifying a *spatial sampling strategy* for collecting the measurements necessary to generate reliable coverage maps help reduce some of the costs of collecting data [13], [12]. In this paper, however, we use the total set of measurements throughout a period to obtain high density of data points along the trajectory. Grid-based approaches to segment the route of interest and pre-process the raw data presents with several limitations, such as unequal distribution of points per resulting segment and uneven segments [2]. We instead propose cutting the route in equal-size segments and reorganize the data around those. This approach allows us to account for noise and sparseness of the data and enable us to analyze MBB performance along the routes. Although map-matching techniques [14], [15] aim to address similar limitations of geo-referenced data, map-matching is beyond the scope of our work as here we focus on manipulating the data for enabling offline analytics for building coverage maps.

III. Measurement Setup and Dataset

MONROE [1], [16] is a European transnational open platform, and the first open access hardware-based platform for independent, multi-homed, and large-scale MBB measurements on commercial networks. The platform comprises a set of 150 nodes, both stationary (e.g., volunteers hosting nodes in their homes) and mobile (e.g., operating in delivery trucks and on board public transport vehicles such as trains or buses). MONROE is currently operational in Italy, Norway, Spain, Sweden, Portugal, Greece and the UK.

Before describing the measurement setup, we summarize the terminology used throughout this paper in Table I. Next, we describe the MONROE node hardware and software along with the deployment. We further detail the measurement campaign.

TABLE I: Terminology.

Route (R)	Train path between two distinct points
Segment (s_R)	Equidistant section of a given route
Operator (O)	The access network operated by a particular operator
Coverage (C)	The highest device mode observed for a given segment

A. Node Hardware and Software

Each MONROE node integrates 2 small programmable computers (PC Engines APU2 board) interfacing with 3 3G/4G MC7455 miniPCI express modems using LTE CAT6 (connected to 3 different MNOs) and one WiFi modem. All software components used in the platform are open source and available online [17].

The software on the nodes is based on Debian GNU/Linux "stretch" distribution. All experiments run inside a virtualized environment (Docker container) to ensure separation and containment of processes. MONROE further provides continuous monitoring measurements including active measurements such as connectivity measurements (e.g., ping) and speedtest measurements [18] as well as *Tstat* [19] passive probe that provides insights on the traffic patterns at both the network and the transport levels. Furthermore, to provide rich metadata to the experiment containers, the *metadata broadcasting service* runs continuously in the background and relays metadata through ZeroMQ[2] in JavaScript Object Notation (JSON) format to experiment containers.

Metadata collection. Since MONROE does not involve real users (which usually entail privacy protection restrictions), rich metadata collection, including geo-temporal tagging, is possible. MONROE nodes generate metadata passively and continuously: each node is instrumented to gather information relating to its MNOs. These include network parameters (RSSI, cell identifiers, link technology, etc.), node location

[1] https://www.monroe-project.eu/access-monroe-platform/

[2] ZeroMQ (ZMQ) distributed messaging: http://zeromq.org

TABLE II: MONROE metadata topics

Class	Type	Examples
Node	Sensor	CPU temperature
Node	Probe	Load, memory usage
Node	Event	Power up, reboot
Device	GPS	GPS coordinates
Device	Modem	RSSI, link technology, cell ID, IP addr.

and speed (GPS), node working parameters (CPU temperature, processing load, etc.) and node events (watchdogs).

Metadata entries are generated in a single-line JSON format, where every entry is labeled with a "topic" field. Table II illustrates the metadata "topics", which are streamed to subscriber entities within the node.[3] The metadata subscriber module subscribes to all the topics, writing JSON entries to files in a special file system location. A synchronization process transfers these files to the MONROE server when no other active, periodic, or user-defined experiment is running. In this way, metadata from all MONROE nodes is collected and stored centrally.

B. Deployment on Trains

MONROE deployment in public transportation vehicles enables the evaluation of MBB networks on wide urban mobility environments. The MONROE platform currently includes 15 nodes onboard 15 inter-city trains in Norway. These trains travel a wide range of routes indicated by the official map in Figure 3a. In Figure 1, we present photos from deployments on NSB trains, where nodes are mounted directly under the desk in the conductor room. This is a semi-closed area of roughly $1.5mx1.5m$ size, located in the mid-section of the train by the passenger seats. The deployment is carried out in such a way that the nodes mimic actual end users traveling these routes, for which reason the mobile MONROE nodes are sometimes called "passenger in a box".

(a) Conductor cab (b) Node under desk

Fig. 1: Node deployment on trains in Norway.

C. Measurement Campaign

In this study, we make use of GPS measurements and modem metadata from MONROE nodes onboard NSB trains

[3]For a complete list of MONROE metadata fields, see https://github.com/MONROE-PROJECT/data-exporter.

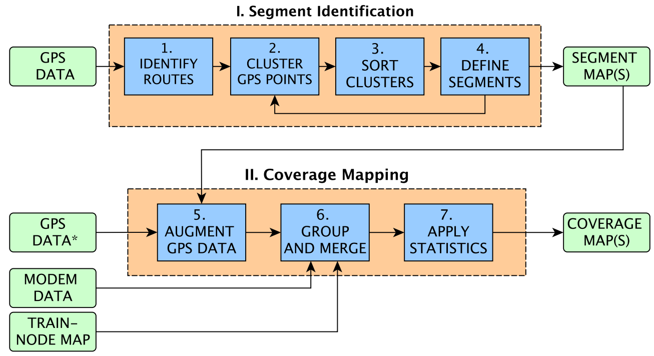


Fig. 2: Algorithm description.

in Norway. The GPS and modem measurements are collected independently, producing two separate datasets.

GPS measurements. These measurements are recorded every 10 s and gathered from the train's fleet management system. We use a GPS dataset with the following fields: `time`, `longitude`, `latitude`, and `anonymized train ID`.

Modem measurements. These measurements are event-based, meaning that *changes* in values, such as link technology, are recorded. In case of no change, new entries are made every 30 s. We use a subset of the modem metadata, including the following fields: `time`, `node ID`, `device mode`, `imsimccmnc`, and `nwmccmnc`.

We focus on 2 Norwegian operators (Telenor and Telia) and consider measurements coming from mobile MONROE nodes with their subscriptions. Their corresponding Mobile Country Code (MCC) is 242, for Norway, and Mobile Network Code (MNC) is 01 and 02 respectively. We provide the details of our measurement campaign below. For the complete dataset including GPS and modem information, readers are referred to [3].

TABLE III: Measurement campaign parameters.

Parameter	Value
Start date	01.01.2017
End date	14.01.2018
Number of nodes	15
Number of routes	13
Mobile technologies	2G, 3G, 4G
Frequency (GPS data)	every 10 s
Frequency (Modem data)	event-based (max 30 s)
Operators (MCC-MNC)	242-01, 242-02
Available datasets	GPS, modem, train-node map

IV. Algorithm

Our algorithm consists of two parts: the first part is *segment identification* with 4 steps, and the second part is *coverage mapping* with 3 steps. Figure 2 describes these two parts and their corresponding steps as a flow diagram.

The purpose of the *segment identification* component is to associate the points in the GPS point cloud which we collect from repeated measurements along the same routes (see Figure 4a), to a particular segment of the corresponding route. We aim to achieve this in 4 steps: (1) we use the raw

TABLE IV: Steps 1 and 4.

ID	Route Description	#Segments (k=100)
1	Oslo - Gothenburg	308
2	Oslo - Roa - Honefoss	78
3	Drammen - Larvik - Nordagutu	165
4	Oslo - Røros - Støren	452
5	Oslo - Eidsvoll	21
6	Dombås - Åndalsnes	106
7	Notodden	8
8	Hamar - Elverum	30
9	Trondheim - Bodø	639
10	Drammen - Stavanger	456
11	Ski - Mysen	53
12	Oslo - Trondheim	484
13	Oslo - Bergen	453

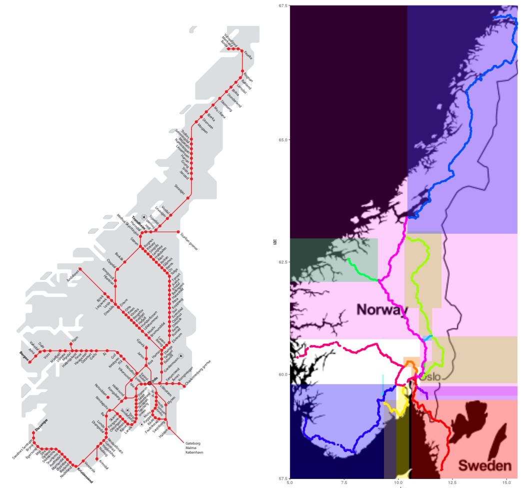
GPS data to identify each route R in the map, (2) cluster the GPS points, (3) sort the clusters, and (4) connect consecutive cluster centers to build a vector representing the route, which we use to define segments s_R of desired length along each route R. The output is a segment list $\mathbf{S} = \{s_R\}, \forall R$, which allows us to map any given GPS point to a particular route segment. This component only needs to be executed once during a time period in which the route structure does not change. Particularly, the component must be updated if new train routes are established by the transportation company, new trains are deployed, or a new MONROE node is installed on a train traversing a new route.

The purpose of the *coverage mapping* component is to present the technology coverage C_O of a MBB network O along all discovered routes, in segment granularity. This component requires the first 4 steps to be executed at least once, but can itself be run more often. For instance, where it is perfectly adequate to update the segment maps once every few months, or even every year, mobile network configurations might change more rapidly such that performance maps are rendered obsolete every few weeks.

In this study, we focus on the technology coverage along train routes for different network operators, but it is possible to extend the second part of our algorithm to use, for example network speed [20], [18] or latency measurements along train routes, so that other outputs including **mobile network performance maps** can be produced. Readers are referred to [3] for our sample implementation in R.

A. Part I: Segment Identification

Step 1. Identifying routes: We first inspect the cloud of GPS points plotted on a map, in order to group them into distinct routes which do not fork or bifurcate. While grouping, we consider the important train stations at big cities that are often the intersection of many different routes. We mark many latitude and longitude box cuts and using boolean logical operations between the box cuts. For our current dataset, this step yields 13 routes and they are listed in Table IV. Figure 3b shows a diagram of the box cutting, and Figure 3a compares our route prediction to an NSB schematic of the official routes from [21].



(a) Official (b) Identified

Fig. 3: Train routes.

Step 2. Clustering GPS points: After Step 1, routes are identified coarsely by their boxcut regions. However, they can only be visualized as clouds of GPS points, as shown in Figure 4a. The purpose of clustering is to go from this cloud of GPS points to a distinct set of representative points, which will mark the route segments later on. For each identified route, we cluster the cloud of GPS points belonging to this route by applying the k-means algorithm.

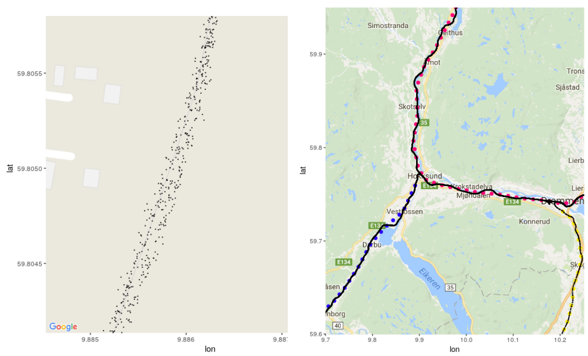
In the first iteration, we run the clustering algorithm coarsely with $k = 100$, to identify the route lengths (roughly). Table IV presents the estimated length of each route in terms of the number of segments, for $k = 100$. We go through Steps 3 and 4, and use the number of segments in each route, n_s (derived from Step 4) to run the algorithm for a second iteration. This time, we run the clustering algorithm with k proportional to the number of segments in each route, $k = c{*}n_s$. We conducted a sensitivity analysis to find a suitable c and the corresponding k per each route, and we observed that $c = 1$ provides enough granularity for the identified routes.

Note that, k-means is used for its efficiency and simplicity of implementation here. However, depending on the available dataset, different clustering algorithms can also be applied. Especially, if there is a significant difference in the density of GPS points along the routes, density based algorithms such as DBSCAN, could be used. This is a topic of our ongoing work, in an effort to generalize our algorithm further.

Step 3. Sorting clusters: The purpose of this step is to order the clusters along their associated route. Since the previous step yields an unsorted list of cluster centers, we need to put them in order of their geographic location to describe a route (directional path).

Step 4. Defining segments: The distance between the clusters identified in the previous step might not be uniform. We form vectors between consecutive cluster pairs and segment the vectors into equidistant intervals. For this study, we have chosen a total length of $1km$ for each segment in order to provide a fine granular coverage map. However, segment size can be configured to meet different needs.

Part I Output. Segment map(s): The algorithm takes the GPS point cloud as input, as illustrated in 4a, and as an output, we get a list of route segments, onto which any GPS point can be mapped. A sample segment map is illustrated in Figure 4. Segments are identified by the latitude and longitude of their beginning, their corresponding route, and ID (sequence number within route).[4]



(a) Cloud of GPS points (b) Segment map

Fig. 4: Input GPS data and output segment map.

B. Part II: Coverage Mapping

Step 5. Augmenting GPS data: We augment the given GPS dataset by adding two fields: `route ID` and `segment ID`. This is achieved by matching every GPS measurement to the nearest segment identified in Step 4. Note that in this step, any GPS dataset (not necessarily the one used in Part I) can be used.

Step 6. Group and merge After identifying the routes and associating GPS information to route segments, the GPS and modem data are merged. Note that multiple trains can be on multiple routes, we use a lookup table that has the train-node mapping (a list indicating which nodes are deployed on which trains), the measurements from which nodes can be used to asses the performance along which route. Furthemore, GPS and modem data from these nodes are grouped by operator. Finally, GPS and modem data per operator and route are merged with a 6 minute time window[5]. The merged dataset for each operator has the following columns: `time`, `node ID`, `device mode`, `longitude`, `latitude`, `train name`, and `segment ID`.

[4] Although train track infrastructures seldom change (except for new tracks being built every 5-10 years), in case of deviations, a particular train might be diverted to a different track. In this case, all measurements from this particular train would be associated with segments along the new track. We do not drop any measurements as long as they can be associated with an identified segment, within a given confidence level.

[5] The modem data is updated when there is a change of state or periodically with a $30s$ interval. However, metadata might not be updated at times due to hardware failures. This can cause a false matching of GPS data to modem data. The 6 minute time window is wide enough for the periodic measurements, but cuts off the matching in case the modem data is not updated

Step 7. Statistical analysis: At this point, for each operator O and per each segment s_R, an analysis of selected performance metrics can be conducted. One of the prominent methods is to consider the *maximum* of available technologies over all measurement points for a given segment and operator. An alternative is to use the *mode* of available technologies over all measurement points for a given segment and operator. Coverage maps can be generated using either statical representation. See Section V for a comparison of using *maximum* or *mode* on our dataset.

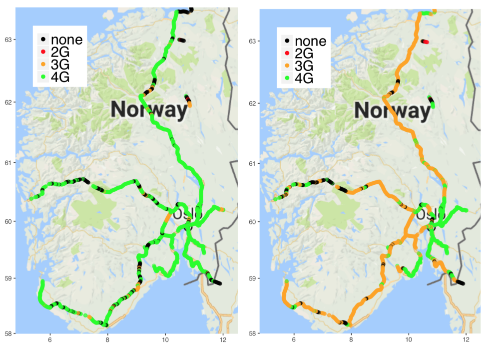
Part II Output. Coverage Map(s): As an output, we get a color-coded map of available technology along the 13 routes identified before, in segment granularity, for each operator.

V. Evaluation

In this section, we first discuss the metrics we selected for building the coverage maps, and then we present different ways we can leverage coverage maps.

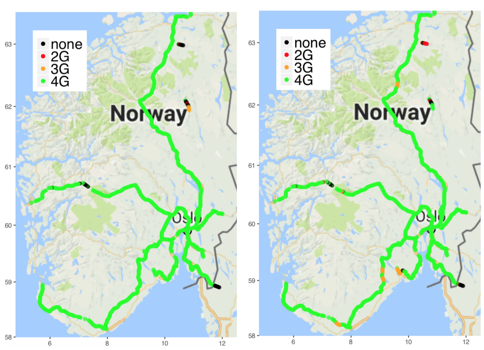
We choose the *maximum* and the *mode* as the statistical metrics. The statistical maximum is a measure of the *best* coverage provided by an operator within a given time period, regardless of any temporal effects, while the statistical mode is a measure of the *overall* coverage experienced by the end-users within the time period. For instance, if an operator suffers from a (temporary) loss of coverage in an area for some portion of the specified time period, this may be reflected in the statistical mode, whereas it will not affect the statistical maximum.

Figures 5a-6a and Figure 5b-6b illustrate the differences between the two chosen statistical metrics for the two operators. Note that, in these maps, we have not illustrated the northern most routes due to having very little data points in these routes. These coverage maps have been generated using the data described in Table III. We observe that although the best coverage provided by Telenor and Telia seem to be relatively similar, Telia users are spending more time on 3G than 4G on average. The underlying issue has been identified as the following: due to internal network configurations of Telia, when there is an active 3G connection (i.e. the connection is in active state, sending data), even though the 4G coverage is available, the network does not provide handover to 4G. The handover is only possible if the connection goes to an idle state and then become active again. Since MONROE nodes are constantly running a ping experiment in the background, and therefore keeping their network connection in active state, once a Telia SIM card on a node is on the 3G network, it will not connect back to 4G network even it is available (unless there is an explicit disconnect from the 3G network due to mobility or other reasons). Our finding has been discussed with and confirmed by Telia. We present the coverage map without modification, since we believe that it represents actual user experience of coverage. For instance, a user surfing the web on these routes would probably get the same coverage experience (heavy 3G domination), due to continuous activity over their connection. Similarly, although end users don't have a regular ping like MONROE, most users have background traffic that keeps their connection active.



(a) Telenor (b) Telia

Fig. 5: *Mode*: Coverage maps for Telenor and Telia along the train routes in Norway using statistical mode.



(a) Telenor (b) Telia

Fig. 6: *Maximum*: Coverage maps for Telenor and Telia along the train routes in Norway using statistical maximum.

In this paper, we have provided the coverage maps for all the data collected during one year period. However, the algorithm can also be used to track coverage changes in time over the routes. We have generated monthly coverage maps and observe the coverage evaluation during this time. Due to limited space, these results are not presented here but provided in the data repository [3].

VI. Conclusion

The current interest around MBB measurements, coupled with the emerging availability of measurement platforms, brings to light the problem of knowledge discovery in the current data-rich environment but information-poor settings. This usually implies working with complex datasets, which are often plagued by many issues, including high dimensionality, sparsity and the presence of categorical variables (e.g., RAT). The approach we propose in this paper helps to tackle some of the limitation of such datasets in the case of drive runs over railway paths. Clustering measurement samples around route segments allows us to transform our dataset and easily create coverage maps using high density of samples. In this study we look at the case of railways in Norway; however, our methodology can easily be generalized for running a similar study in other routes. Moreover, though here we focus on radio coverage, we plan to extend this analysis to produce additional performance maps for Quality of Service (QoS) and Quality of Experience (QoE) MBB metrics.

Acknowledgment

This work is funded by the EU H2020 research and innovation programme under grant agreement No. 644399 (MONROE), and by the Norwegian Research Council project No. 250679 (MEMBRANE).

References

[1] Ö. Alay *et al.*, "Experience: An open platform for experimentation with commercial mobile broadband networks," *MobiCom*, 2017.

[2] A. Lutu *et al.*, "The good, the bad and the implications of profiling mobile broadband coverage," *Computer Networks*, vol. 107, pp. 76–93, 2016.

[3] https://mosaic-simulamet.com/coverage-visualization/.

[4] C. Phillips *et al.*, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, 2013.

[5] M. K. Marina *et al.*, "Impact of indoor-outdoor context on crowd-sourcing based mobile coverage analysis," in *Proc. ACM SIGCOMM 2015 Workshop on All Things Cellular: Operations, Applications and Challenges*, 2015.

[6] T. Berisha *et al.*, "Benchmarking in-train coverage measurements of mobile cellular users," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.

[7] J. Garcia *et al.*, "Train velocity and data throughput - a large scale lte cellular measurements study," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–6.

[8] T. Berisha *et al.*, "Cellular network quality improvements for high speed train passengers by on-board amplify-and-forward relays," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, Sept 2016.

[9] Tektronix, "Reduce Drive Test Costs and Increase Effectiveness of 3G Network Optimization," Tektronix Communications, Tech. Rep., 2009.

[10] W. A. Hapsari *et al.*, "Minimization of drive tests solution in 3gpp," *Communications Magazine, IEEE*, vol. 50, no. 6, pp. 28–36, 2012.

[11] M. Molinari *et al.*, "Spatial interpolation based cellular coverage prediction with crowdsourced measurements," in *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*. ACM, 2015, pp. 33–38.

[12] M.-R. Fida *et al.*, "Zipweave: Towards efficient and reliable measurement based mobile coverage maps," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[13] S. Grimoud *et al.*, "Best sensor selection for an iterative rem construction," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*. IEEE, 2011, pp. 1–5.

[14] P. Newson *et al.*, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 336–343.

[15] C. Y. Goh *et al.*, "Online map-matching based on hidden markov model for real-time traffic sensing applications," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 776–781.

[16] M. P. Quirós *et al.*, "Results from running an experiment as a service platform for mobile networks," *MobiCom WiNTECH*, 2017.

[17] https://github.com/monroe-project.

[18] C. Midoglu *et al.*, "MONROE-Nettest: A configurable tool for dissecting speed measurements in mobile broadband networks," *INFOCOM CNERT*, 2018.

[19] A. Finamore *et al.*, "Experiences of internet traffic monitoring with tstat," *IEEE Network*, vol. 25, no. 3, pp. 8–14, May 2011.

[20] A. S. Khatouni *et al.*, "Speedtest-like measurements in 3G/4G networks: the MONROE experience," *ITC 29*, 2017.

[21] https://www.nsb.no/en/our-destinations/map-of-railway-lines.