Computer Networks 177 (2020) 107294

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

RICERCANDO: Data mining toolkit for mobile broadband measurements

Veljko Pejović*, Ivan Majhen, Miha Janež, Blaž Zupan[1]

Faculty of Computer and Information Science, University of Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Mobile broadband networks
Data mining
Network measurements
Anomaly detection

ABSTRACT

Increasing reliance on mobile broadband (MBB) networks for communication, vehicle navigation, healthcare, and other critical purposes calls for improved monitoring and troubleshooting. While recent advances in monitoring with crowdsourced and network infrastructure-based methods allow us to tap into a number of performance metrics from all layers of networking, huge swaths of data remain poorly explored due to a lack of tools suitable for fast, interactive, and rigorous MBB data analysis. In this paper we present RICERCANDO, a solution that enables rapid exploration of large heterogeneous MBB measurement data as well as the identification and explanation of unusual patterns detected in such data. RICERCANDO consists of a preprocessing module ensuring that time-series data is stored in the most appropriate form for mining, a rapid exploration module enabling iterative analysis of time-series and geomobile data to detect and single-out anomalies, and the advanced mining module that lets the analyst deduce root causes of observed anomalies. We implement and release RICERCANDO in open-source, and validate its usability on case studies from a pan-European MBB measurement testbed.

1. Introduction

In December 2018, after a glitch involving software certificates, up to 32 million O2 mobile network customers in the United Kingdom and some 30 million SoftBank network customers in Japan were left without access to data services for up to 24 hours [1]. Despite its relatively short duration, the incident prompted public outrage and lead O2 to compensate its customers and request "tens of millions of dollars" in damages from Ericsson, a network equipment manufacturer whose software caused the issue. The glitch was yet another demonstration of the value of mobile connectivity and the need to rapidly detect and understand the causes of mobile broadband network anomalies.

In the global connectivity landscape, mobile wireless communications play a particularly prominent role. The advent of mobile wireless communication had a tremendous impact on numerous aspects of our lives – from the way we navigate in unknown environments, communicate on the move, over the way we pay our bills, to the way we track our health and wellbeing. Underpinning and enabling all of this are mobile broadband (MBB) networks. These networks have witnessed rapid expansion recently – MBB subscriptions have grown more than ten-fold in the last decade and have reached 5.3 billion globally in 2018 [2]. Network performance is improving drastically – a few Mbps download speeds enabled by 3G technology at the break of the millennium appear ancient in comparison with a few Gbps delivered by today's 5G technology. Finally, MBBs are becoming more affordable – worldwide MBB access prices halved between 2013 and 2016 [3]. Together with the expansion of novel paradigms that depend on fast ubiquitous connectivity, such as the Internet of Things (IoT), e-Health, smart cities and factories, the above trends indicate that our reliance on MBB networks is to grow even further.

MBB networks have penetrated into virtually all aspects of our everyday lives, became the inseparable part of today's Internet, and ensuring MBB networks' reliability became a critical issue. Underpinning the efforts to ensure reliability are network monitoring and data analysis methods. Despite the advances in MBB performance measurement methods [4–8] the problem of the identification of performance anomalies and, even more, the identification of root causes of network anomalies remains unsolved. First, the sheer breadth of networks, both in terms of the number of devices as well as their geographic spread, requires consideration of multiple views of the same phenomenon before any conclusions can be made. Yet, frequent fine-grain measurements, necessary due to the networks' dynamic behaviour, result in tremendous amounts of data, rendering multifaceted/multigranular analysis a challenging task. Second, the networks' multilayered construction calls for a joint consideration of (meta) information from different levels, from physical layer information on signal strengths, over transport layer retransmissions, to packet delay and jitter. However, these data are collected by different probes and sensors, and providing a unified view

* Corresponding author.
E-mail address: veljko.pejovic@fri.uni-lj.si (V. Pejović).

[1] This work is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644399 (MONROE) through the open call project RICERCANDO. VP and IM are also supported by Slovenian Research Agency (grant no. N2-0136). BZ is also supported by Slovenian Research Agency (grant no. P2-0209).

https://doi.org/10.1016/j.comnet.2020.107294
Received 3 February 2020; Received in revised form 10 April 2020; Accepted 4 May 2020

of the data coming from different sources calls for novel intelligent data consolidation strategies. Finally, current approaches to explaining anomalies are rather ad-hoc and rely on networking experts' intuition. The increasing complexity of MBB networks prevents exhaustive search for potential reasons for network malfunctioning, while statistical and machine learning methods that could help pinpoint the causes of network anomalies often remain outside the network administrators' expertise and are challenging to apply within the existing network traffic analysis and visualisation tools.

In this paper we tackle the problem of *detecting and explaining* MBB network performance issues. We do that through RICERCANDO, a MBB network data analysis framework developed in tight collaboration of networking and data mining experts and designed to answer the above-listed challenges. RICERCANDO enables multi-staged and flexible data analysis. Our framework handles the first stage of the analysis through a data representation scheme that merges data of different types and from different sources, and adapts them to time series-based organisation suitable for querying with a different level of granularity. RICERCANDO then enables scalable interactive visual analysis of big network measurement data. Next, we devise anomaly detection methods that pinpoint measurements where network performance indicators significantly deviate from the expected values. Finally, through RICERCANDO's machine learning pipeline designed to help with the identification of key factors that might have caused the observed anomalies, we introduce rigorous statistical and machine learning methodology to MBB data analysis.

Specific contributions that RICERCANDO brings to the research area of network management include:

- Design of data merging and re-sampling method for agile data manipulation;
- Implementation of adaptable and multi-dimensional temporal and geographical visualisation of MBB measurement data;
- Implementation of various anomaly detection methods suitable for time-series data;
- Inclusion of support for modern data mining techniques in network data analysis.

Through a case study conduced on data collected through a pan-European MBB network measurement testbed we demonstrate RICERCANDO's ability to detect and explain network anomalies. Finally, we have released RICERCANDO as an open-source software and we invite the community to join our efforts towards supporting rapid MBB network measurement data analysis.

2. Related work

2.1. Monitoring MBB networks and measurement data management

A systematic method of monitoring is crucial for assessing the quality of service and troubleshooting in mobile broadband networks. Recently, a wide range of approaches for MBB measurements have been developed [9]. Approaches rely on either passive [5,8] or active measurements [10], or on a hybrid measurement methodology that combines both [4,11–13]. Passive measurements merely observe the existing network traffic, while active measurements inject own packets in order to evaluate performance metrics. The downside of active measurements is that the measurement process may impact the actual network under test.

In terms of the measurement point locations, certain approaches, especially those initiated by national regulators, use dedicated monitoring equipment and a small number of controlled nodes, while others rely on crowdsourced measurements conducted by a large number of often uncoordinated users [14]. The former have the benefit of being unrestricted by the provider, of viewing the network as users, and of covering wide geographical areas. OpenSignal, for instance, has more than 100 million users across the globe [15]. However, crowdsourced measurements suffer from unreliability due to the lack of control over the measurement equipment. A mobile app-based measurement software may be run on different phone models, with different implementations of the operating system, devices running different applications in parallel to the measurement app, different hardware issues (e.g. bent antennas), and devices placed in various locations during measurements (e.g. bag/pocket/hand), all of which may impact measurement results [16,17]. Recent commercial and research initiatives hence use crowdsourced-like approach with specialised equipment dedicated to network measurements [18].

Irrespective of the measurement approach, MBB measurement data is large-scale, temporal, heterogeneous, and shaped by a number of factors related to measurement methodology and equipment. Storing, processing and reasoning upon such data is challenging, and a number of solutions providing a structured approach to measurement data analysis have been developed. Svoboda et al. demonstrated the importance of using a well-defined methodology for packet delay measurement analysis in order to obtain meaningful interpretation of the results [11]. CoMo provides a structure for fast prototyping of network measurement mining applications [19]. Mostly concerned with data storage and flow, CoMo does not provide sufficient support for advanced analytics. Future efforts were aimed at either increasing scalability, usability, or the number of supported options for data analysis. ENTRADA, for instance, converts `pcap` log file to Apache Parquet and enables stream mining [20]. Similarly, DBStream was built to support rolling big data analysis [21]. The tool's utility has been demonstrated on a few use cases, including on the analysis of signalling and data transfer behaviour of different mobile device types and different operating systems [22]. Designed by networking experts, these systems usually provide solutions to network measurement data handling, yet stop at the point where advanced data mining is needed.

2.2. Mining MBB measurements

The complexity of MBB measurement data prompted networking researchers to resort to ad-hoc and task-specific approaches to data mining. Baltrunas et al. show that even simple correlation can help with network reliability estimates [23]. In order to profile network coverage in Norway, Lutu et al. perform hierarchical clustering of measurement data collected via train-mounted probes [24]. Narayanan et al. propose a feature distribution similarity graph to analyse spatio-temporal mobile measurement data [25]. The authors show the utility of the approach in a case study of profiling mobile users' behaviour from call detail records. ESkyPRO probe employs supervised classification to detect encrypted Skype traffic [26]. With RICERCANDO we go a step further and devise a rich framework focused on discovering general anomalies in MBB measurement data and identifying their root causes using unsupervised learning.

More advanced approaches try to automate the mining process, especially when it comes to anomaly detection, a key issue in network data analysis. An overview of statistical methods for anomaly detection for computer networking experts was presented by Callegari et al in [27]. A recent advancement in the area of automated detection is ADAM, a system that detects anomalies by estimating Kullback-Leibler divergence between the incoming and previously collected data [4]. Once an anomaly is detected, the system performs factor analysis to identify features exhibiting a similar abrupt change. RCA tool initially detects change points by measuring the entropy of considered features [28]. It then considers the full statistical distribution of the traffic features to characterise anomalies. Ricciato et al. suggested two approaches to bottleneck detection, the first one based on statistical analysis of the aggregate rate, and the second method based on TCP performance indicators [29]. Coluccia et al. proposed an anomaly detection methodology that identifies statistically significant deviations from the past behaviour using Maximum Entropy modelling [30]. In another study, the authors investigated distributions of multiple features to detect traffic anomalies, indicating that the alarm correlation across features may augment

the accuracy of the detector [31]. In [32] Li et al. describe a random forest-based approach for anomaly detection in passive measurements. While a clear intuition behind the rules of tree splitting provides a step towards interpretable machine learning, the approach does not yield a clear picture of which contextual parameter may have caused the anomaly. Furthermore, unlike RICERCANDO, work presented in [32] is not a full-fledged open-source software framework. Association rule mining is another popular approach for identifying potential causes of network malfunctions. Zargarian et al. present a method for mining association rules describing temporally and spatially correlated alarm events from a network log [33]. Similarly to RICERCANDO, this work aims to support networking experts by relieving them from the burden of big data analysis. RICERCANDO, however, integrates with Orange [34], a data mining suite that hosts a wide range of statistical tools, while also including association rule mining. Moreover, RICERCANDO provides tools for flexible data exploration enabling efficient visualisation of very large measurement datasets. Ahmed et al. identified network providers, locations, device types, and applications, or combinations of the above, that lead to performance degradation in a large 3G network [35]. The proposed method relies on iterative construction of regression models to detect underperforming measurements and association rule mining in order to single out the most prominent combinations. Intuitively, such an approach focuses on the most apparent, recurring anomalies. Our approach to anomaly detection (presented in Section 4) enables closer inspection and detection of even short-lasting deviations from the expected performance, while also providing statistical explanations for the discrepancy.

In summary, the existing work in the area of network measurement analysis primarily focuses on either measurement data storage and management [19–21], or on the development of methods for processing and profiling network measurements, and identifying anomalies in the data [4,25,27,30]. Furthermore, these tools often stop short of providing advanced data mining capabilities and instead rely on a networking expert's presence in the loop[2]. Finally, despite interactive live data visualisation likely being the most efficient means of harnessing expert knowledge [38], the presented tools seldom provide any advanced visualisation capabilities. Recognising the shortcomings of the above approaches as well as the limitations of existing visualisation tools (further elaborated in Section 4.2), in RICERCANDO we implement a suite of data processing, mining, and visualisation methods specifically tailored for MBB measurement data analysis.

3. MBB Measurement data characteristics and RICERCANDO analysis approach

A careful examination of the characteristics of MBB measurements represents a cornerstone of RICERCANDO. As discussed in the previous section, MBB measurements system can rely on passive or active measurements, and can be performed through well-planned installations or in an opportunistic crowd-sourced manner. Yet, certain properties characterise MBB measurements irrespective of the measurement system implementation.

We base our requirements analysis on the examination of the related work of MBB measurement mining (Section 2), but also on an in-depth analysis of a state-of-the-art MBB measurement platform – MONROE. MONROE is an open access hardware-based platform for independent, multihomed, large-scale experimentation in MBB networks [18]. The MONROE project aims to create a pan-national reliable open-access measurement platform for MBB networks[3]. The core of the system is a MONROE node, a custom-built device fitted with a Debian-based single board computer and up to three LTE modems connected to different providers. A centralised experiment scheduling system allows MONROE users to post custom-made experiments to distributed nodes and remotely collect measurement results. In addition, each node independently executes certain background experiments, such as periodic RTT measurements to MONROE servers. Finally, all the experiment data and meta-data are collected in a MONROE database implemented in Cassandra[4]. In 2018, the project operated 150 measurement nodes in four European countries, with more than a half of the nodes being mounted on buses, trains, and delivery trucks.

We have been conducting MONROE data analyses from the projects inception in 2016 and have obtained a thorough understanding of the characteristics of the measurement data. Similarly to other systems, MONROE measurement data are characterised by:

- Spatio-temporality: measurement nodes are geographically dispersed and often mobile;
- Multi-modality: multiple aspects of network performance (RTT, throughput, etc.) and meta-data (location, CPU load, etc.) are sampled in parallel;
- Heterogeneity of data exhibited through their varying granularity and the lack of synchrony among different measured features;
- Impact of the measurement methodology, hardware, and software on the measurement results;
- Lack of ground truth data.

A MBB data analysis tool has to cope with the above characteristics of the data on the implementation level. On the higher level, however, the tool has to enable comprehensive analysis, requirements of which have been discussed among the research community before. For instance, in 2006 Ricciato indicated that network traffic analysis should include statistical analysis that goes beyond simple ad-hoc solutions, visualisation and multidimensional exploration by networking experts and advanced machine learning algorithms, and should allow the data to be pipelined to other tools [39]. Recently, needs for additional higher-level inferences from MBB measurements, such as Net neutrality violation detection, have also been voiced [40].

We design RICERCANDO to take into account the unique characteristics of the MBB measurement data and directly answer the needs of the research community. In RICERCANDO, we explicitly support interactive analysis and put the user in the loop. Moreover, our data storage paradigm is adapted to support rapid visualisation and experimentation, so that the expert knowledge can be harnessed in the best possible way. Similarly, identifying a need for automated statistical analysis, we create a machine learning pipeline that automatically detects and suggests explanations for network anomalies. At the same time, the system's visual component maintains a close dialog with an expert enabling iterative investigation until the root cause of the issue is identified. Finally, recognising the uniqueness of each measurement setup and varying goals of those who analyse networks, we do not restrict RICERCANDO to particular mining techniques. Rather, we integrate it with the popular data mining suite Orange[5], allowing a wide range of current and future data mining approaches.

4. RICERCANDO framework

RICERCANDO is structured around modules that together create a data mining pipeline (Fig. 1). The framework assumes that the data is stored in a key-value database, such as Cassandra[6] used by the MONROE project. *Data Preprocessing* module (Section 4.1.1) transforms and stores the data so that it can be quickly retrieved along the temporal dimension. *Data Merging Interface* (Section 4.1.2) enables different views

[2] This is explicitly evident in Siekkinen et al. TCP RCA approach [36], but also through subtle issues related to data collection and interpretation process. For instance, Michelinakis et al. show how peculiarities of packet scheduling at an LTE base station impact capacity estimates inferred through measurements [37].

[3] http://www.monroe-project.eu

[4] http://cassandra.apache.org

[5] http://orange.biolab.si

[6] http://cassandra.apache.org/

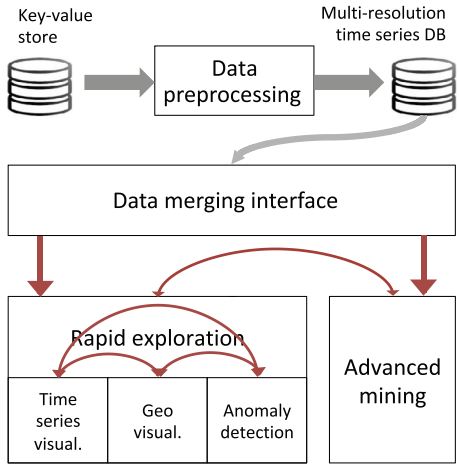

Fig. 1. An overview of RICERCANDO framework. Boxes represent framework's modules, while arrows represent data movement. Darker (red) arrows indicate that data is given in Python pandas format, suitable for interchange among different processing modules and tools. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

over the data. *Rapid Exploration* (Sections 4.2 and 4.3) module consists of three submodules that allow interactive visualisation of time-series data, geomobile data visualisation, and anomaly detection. Finally, *Advanced Mining* module (Section 4.4) interfaces with Orange data mining suite and enables sophisticated machine learning and additional data visualisation methods.

RICERCANDO implementation consists of a core `ricercando` Python library[7], data preprocessing scripts written in Bash and Python, Jupyter Notebooks for visual analysis, and an add-on for Orange data mining suite. All the code, together with the installation instructions is available on GitHub[8].

4.1. Data preprocessing and interfacing

4.1.1. Storage and re-sampling

MBB measurement data are often collected in relational or key-value databases, as they enable easy and efficient storage [4,21,41]. However, stored in such a manner, data are not suitable for rapid interactive exploration. This is especially true for data with a temporal dimension, which is common in MBB measurements – nodes move in space/time, RTTs are gathered with periodic pings, anomalies and glitches impact subsequent node behaviour. Key-value and traditional relational databases severely limit the performance and the flexibility of writing queries over time-series data. The volume of data and metadata gathered by MBB measurements can be large. For instance, RTT measurements from MONROE platform produce approximately 20 million entries per day. Data storage needs to support data sampling to allow zooming in and out on a selected chunk of data, or to support concurrent analysis of data coming from multiple nodes. MBB data comes from various sources, such as multiple nodes and multiple processes within a measurement node, and are often not aligned along the common time axis. Consequently, merging the data in order to enable multidimensional analysis is challenging.

In RICERCANDO we devise data transformation and data storage schemas to transform MBB data into minable representations. We use temporal data abstraction and feature engineering guided by domain-specific knowledge, and we construct scripts that implement various data transformation tasks. To solve the temporal data mining problem we transform the data to a time-series database[9] We store time-series data with the minimal temporal granularity determined by the measurement equipment time resolution (usually in millisecond range). We also sample and store the data at a different granularity (e.g. 1 s, 1 min, 30 min, etc.). This is crucial for enabling interactive visualisation – if a user requests to visualise a whole day of data, we fetch data of a coarser temporal granularity; for examining particular anomalies, we zoom in and provide fine-grain data. When sampling to low resolution the aggregation of values within the period depends on the type of data. Thus, with a few exceptions, for categorical variables we use *mode* function that returns the most frequently observed value in the considered time frame, while for numerical we use either *min, max*, or *mean*. The intuition for different aggregating functions stems from the diverse nature of the observed variables. For instance, RSSI values are often volatile even between subsequent closely-spaced measurements, thus their mean is usually considered [23]. On the other hand, for understanding network congestion, the minimum of the achieved throughput may be more informative than its mean. Finally, for the number of network users in a time period and for certain network resources (e.g. radio access bearer requests) the maximum value in a time slice may be the most appropriate aggregation function for network troubleshooting purposes [43].

4.1.2. Merging data from different sources

Data mining and modelling is performed on datasets consisting of *instances*, where each instance represents a data point in a multidimensional feature space. For example, a measurement of the GPS location, RTT, and the state of the measurement node at a point in time. As measurement data come from various non-synchronised sources, we often have to merge individual data streams along the same time axis. A sketch of the merging process we implement in the Data Merging Interface module is shown in Fig. 2 . For each of the time series (e.g. ping RTT, GPS coordinates, etc.) we find an intersection with a selected moment on the common time axis, and then apply a different strategy for inferring the value at the requested moment in time.

Similarly to the need for different aggregation strategies elaborated in Section 4.1.1, the need for a range of value inference strategies stems from the diverse nature of the observed variables. For instance, a change in a user's location is limited by the physical properties, such as the speed of movement. Thus, for GPS coordinates we perform interpolation between the last measurement before and the first measurement after the given moment in time. For RTT we take an average of the measurements recorded in a time window preceding the current moment. Note that the approach used with the GPS coordinates would not be appropriate here – network malfunctions or connection switches (e.g. from 3G to LTE connection) often result in sudden RTT changes, which would be masked by the simple interpolation approach. For features indicating discrete events we keep track of the node's state and assign the last observed state to the instance we are inferring the value for. For example, the last value of the indicator stating that an experiment is currently running at the node would be extrapolated to the currently considered time. Finally, RICERCANDO allows further tuning of the merging process, for example, by specifying the minimum freshness value of the data before it is included in a data instance – e.g. if no download speed measurements were taken in the last 60 s, the instance will contain a *null* value for *download speed*. This can be further extended to "tighten" the reliability of the inferred values – e.g. the larger the difference between the two GPS points we interpolate from, the less confident we become about the inferred value and we might consider replacing it with a *null* value. While we steer away from a fully automated merging and require

[7] `ricercando` is also available via `pip` installer

[8] http://github.com/ivek1312/ricercando/

[9] We use InfluxDB (www.influxdata.com) in our implementation; compared to popular alternatives, such as Elasticsearch, InfluxDB delivers 6.1x greater write throughput, uses 2.5x less disk space, and delivered 8.2x faster response times [42]; furthermore, InfluxDB supports time-series signal analysis out of the box. Nevertheless, RICERCANDO is not dependent on InfluxDB and alternative time-series databases can also be used.

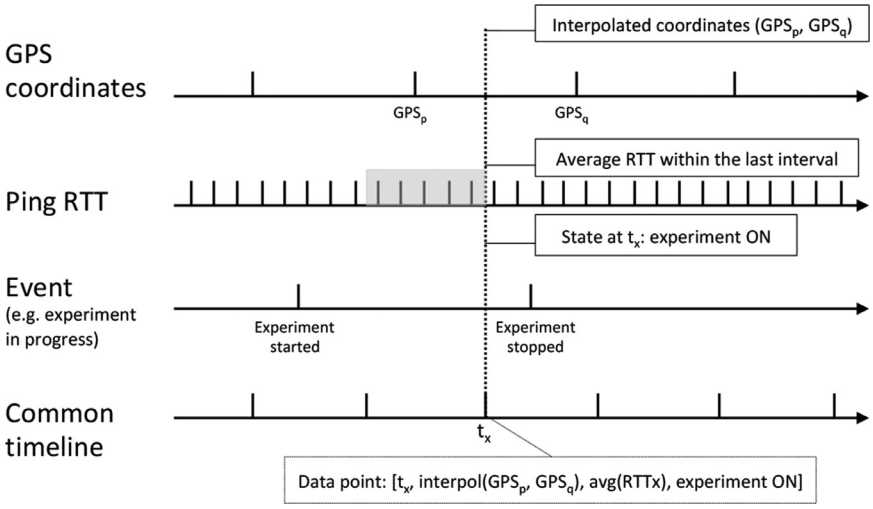

Fig. 2. Data merging along the common time axis in RICERCANDO.

input from a networking expert, this guarantees that the further analysis is done on truly meaningful data.[10]

4.2. Interactive visualisation of big MBB measurements data

Iterative examination of visualised data is crucial for network data mining [39]. These data, however, are multidimensional, temporal, and geo-mobile, and very large, exemplifying the common "three Vs" challenges in big data visualisation: volume, variety, and velocity [47]. Conventional data visualisation tools that come with data mining packages, such as WEKA or Orange, struggle with MBB data, moreover, the amount of data might even overburden specialised tools, such as Tableau [48]. Interactive Web-based visualisation frameworks, such as Bokeh[11], Holoviews[12] and Plotly[13], strive to tackle the above challenges, yet, they remain limited by the Web technology – it can handle only a limited amount of points before the Web browser chokes [49]. Grafana[14] represents a powerful and popular tool for data visualisation, yet, as in our framework data visualisation remains highly interactive and tightly connected with the data modelling pipeline, it requires a bespoke solution. For instance, our time-series and geo-visualisation tools perform adaptive sampling depending on the zoom level, interact with heterogeneous data merging and allow the selected data to be quickly funnelled into a Python Pandas Dataframe and forwarded to Orange for further inspection. Using an off-the-shelf solution for data visualisation would preclude such a flexible connection among different parts of our framework.

To visualise a large number of data points the existing solutions rely on methods, such as decimation and data-shading. Decimation resamples data in advance and displays only a predefined maximum number of data points. Data-shading plots rasterised images rendered to show the amount of detail appropriate for the current zoom level. While generally applicable, these methods are not suitable for interactive MBB data analysis – decimation omits random points, which may impact experts' interpretation of the observed measurements, and data-shading prevents interactivity since rasterised image disallow further data selection and forwarding to a machine learning pipeline. Finally, neither of the techniques tackles the problem of volume – how to automatically prepare the right amount of data for visualisation at query time.

RICERCANDO's original approach to data preprocessing (see Section 4.1) is naturally suited for tackling the "three Vs" challenge. The *volume* challenge is tackled by storing and sampling the data at different granularity, the *variety* issue is tackled with different merging/preprocessing techniques, and *velocity* is tackled with speed-optimised adaptable granularity queries. Based on the above approaches, we develop two modules for rapid interactive visualisation of MBB measurement data – one for time-series visualisation, the other for geographical data visualisation, both implemented in the form of Jupyter Notebooks. We opted for this environment, as opposed to custom stand-alone programs, as it allows quick prototyping and tweaking according to specific user needs and given datasets.

Time-Series Visualisation module for a selected network probe (node) and a time period plots a target key performance indicator (KPI) on a separate timeline for each of the node's interfaces. An additional dimension can be represented through the colouring of each of the points (Fig. 3). User is able to choose the preferred colour palette from various options. Finally, the tool enables hovering over a point, showing values of all the other dimensions associated with the same data point. A key property of the Time-Series Visualisation module is its adaptability to the amount of to-be-shown data. It relies on `getdf` function from `ricercando` Python module, which, for the given zoom level retrieves data from the database with an appropriate resolution, in order to preserve the interactivity of the notebook. For example, viewing a whole week worth of measurements might use data aggregated on 30 min intervals, whereas zooming into a particular RTT anomaly might fetch and show data with 10 ms granularity.

Geographical Data Visualisation module (Fig. 4) supports visualisation of a selected KPI of geo-referenced data from a measurement node on a separate map for each of the node's interfaces, for the given time period. Such visualisation is a key tool for the identification of issues affecting particular geographic regions. Similarly to the Time Series Visualisation module, hovering over a point shows values of all the other dimensions associated with the same data point. Geographical Data Visualisation module, too, relies on `getdf` function for adaptive data retrieval, so that the retrieved data resolution is adjusted to the current map zoom level.

RICERCANDO modules, such as Time Series Visualisation, Geographical Data Visualisation, Anomaly Detection, and Advanced Mining module are designed to fit into each other just like LEGO® bricks and allow flexible data analysis workflows. To support interoperability among modules we rely on Python pandas DataFrame (dark/red lines in Fig. 1). Indeed, each of the Jupyter notebook-based modules allows data selection (e.g. selecting a range of data points on a map) and storage (as a DataFrame on local storage), and retrieval from another module, e.g. in order to perform advanced mining in Orange.

[10] The inclusion of domain experts early on in the data preprocessing stage is often emphasised as a crucial step in modern data mining [44–46].
[11] https://bokeh.org/
[12] http://holoviews.org/
[13] https://plot.ly/
[14] https://grafana.com/

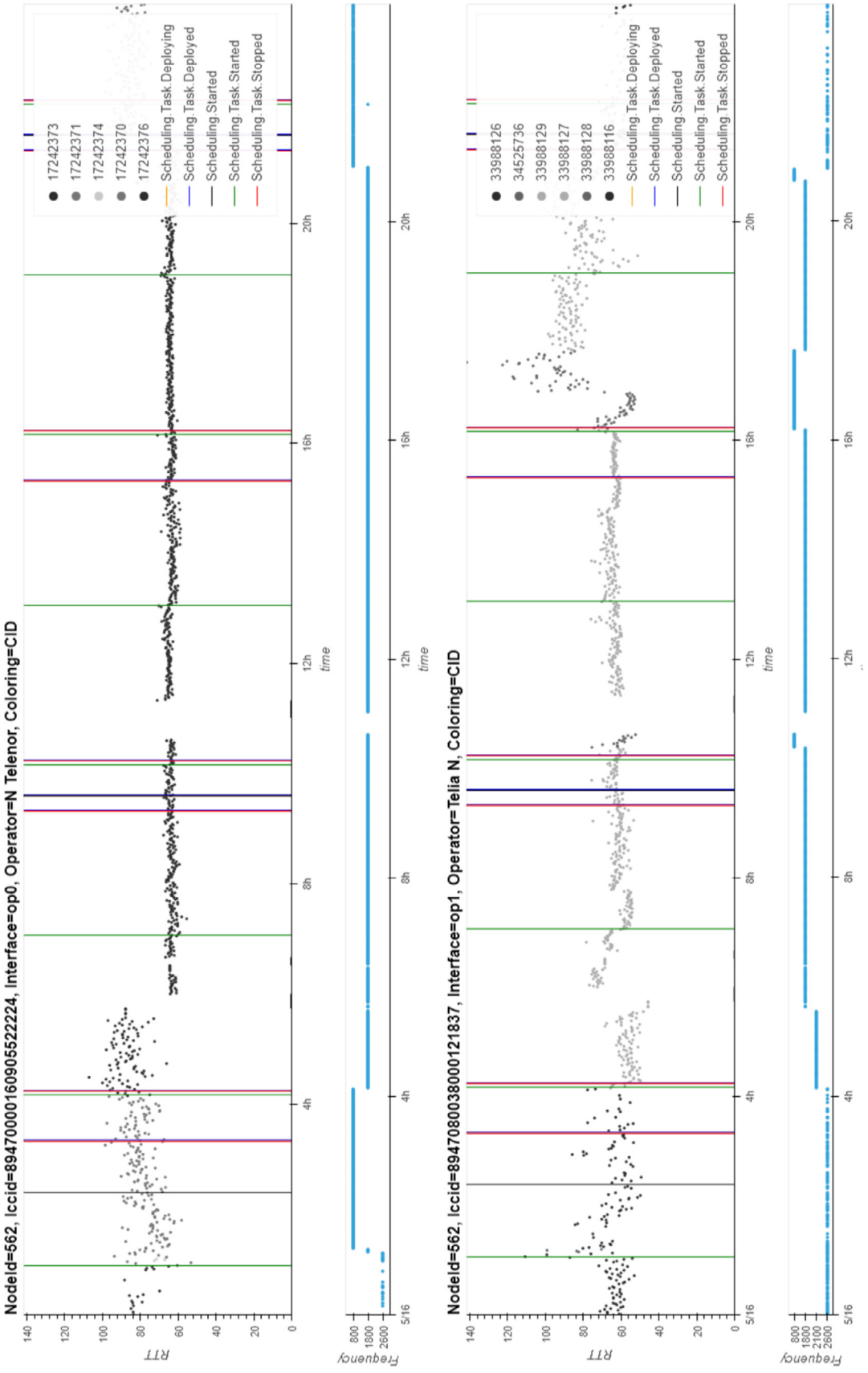

Fig. 3. Time-series visualisation in RICERCANDO. Y-axis represents RTT measured on each of the two interfaces of the same node, while colouring corresponds to the cell id (CID). Vertical lines represent MONROE experiment start/stop/loading moments. Plots below each of the RTT series show the frequency used by the interface. Figure shows node 562 with two interfaces. RTT in interface on top varied from 60 to 100 ms until 5h, while switching between different cell ids. From 5h to 20h the cell id does not change and also RTT stays almost constant at 60 ms. The operating frequency from 4h to 21h is 1800 MHz.

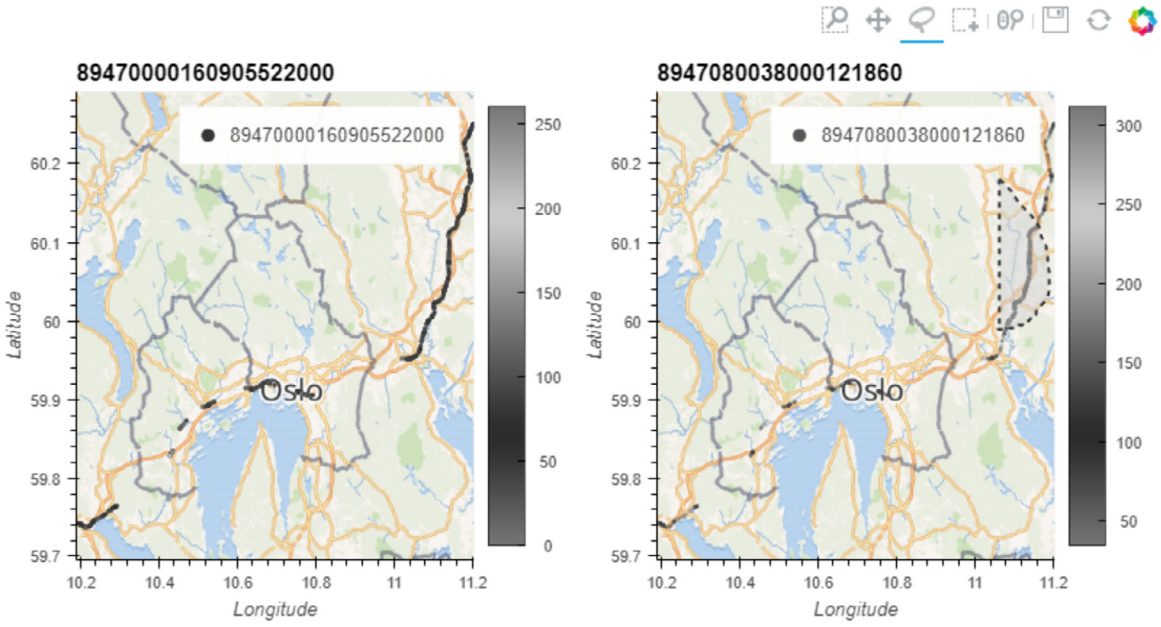

Fig. 4. Geographical visualisation of RTT measured on two interfaces of the mobile node travelling through Oslo. The shades of the trace correspond to different values of RTT. On the right image a selected region contains RTT data stored for further analysis.

4.3. Anomaly detection tool

Anomalies occur frequently in computer network measurement data and can be caused by anything from misconfigurations to cyber attacks [50,51]. Anomaly detection plays a central role in RICERCANDO. We implement a Jupyter Notebook that enables automatic detection and visual inspection of anomalies in the data (Fig. 8). Numerous detection methods relying on a range of techniques, from mining association rules [52], to modelling with Markov processes [53], have been proposed for anomaly inference (see [54] for a survey of anomaly detection approaches).

What is an anomaly? Without any knowledge of the underlying system that generates the data, an anomaly detection system aims to find "sufficiently different" measurements in a stream of data. Alternatively, the data are labelled as "anomalous" if they do not follow the patterns that a domain expert expects, based on her mental model of how the MBB network "should" behave. While the first definition leaves us struggling to find parameter values that would define "sufficiently different" behaviour in automated anomaly detection systems (Romirer and Ricciato have pondered on this question in the context of delay measurements in 3G networks [55]), the second definition is limited by the expert's (mis)understanding of the network phenomena. Thus, in RICERCANDO we aim to judiciously guide an expert in reasoning about the observed deviations. We implement methods for automated labelling of "sufficiently different" measurements, while at the same time the methods' parameters allow the experts, guided by an immediate visual feedback, to adapt the labelling to the currently considered situation. Further, we "encode" the underlying knowledge about the system to label as anomalous only those values that do not conform to a pre-constructed model, therefore, moving the automation closer to the "expert" side of the spectrum.

An anomaly is usually a previously unseen event and without substantial involvement of networking experts we cannot expect that a labelled training data set is available. Thus, lacking the ground truth, we use unsupervised machine learning for anomaly detection. Certain MBB measurements are clearly anomalous, characterised by rapid changes like sudden very high RTT. Consequently, our first approach to anomaly detection relies on a simple comparison of a signal with the previously observed data. Yet, observed changes in MBB measurements need not be anomalies, but reflections of natural changes in the underlying connectivity (e.g. a handoff from 4G to 3G). Thus, our second more advanced approach to anomaly detection relies on detecting deviations from the expected measurement values, where these values are predicted by a model that takes the underlying connectivity context into account. Finally, we augment our toolbox with an anomaly detection method that is founded in a statistical comparison between two sets of measured values.

In summary, RICERCANDO implements three anomaly detection methods:

- **Rolling mean** – a method based on a rolling window that compares data in the current window with a long-term mean of the measurements. Data points that are a number of standard deviations away from the rolling mean are regarded as outliers and a large enough cluster of outliers is identified as an anomalous region. The rolling analysis recognises fast and large changes of the values in a time series. The speed of change is related to the size of a rolling window, which in turn is related to the amount of data explored by the window. The number of standard deviations from the rolling mean determines the sensitivity of the method to the observed change. Different parameters for anomaly detection, including the rolling window size and the standard deviations threshold, can be set by the user. A networking expert can identify and fix the parameters for different applications, thus enabling subsequent automatic anomaly detection. With this method abrupt falls or rises (spikes) are treated as anomalies, while, for example, a gradually rising RTT due to increased network congestion would not be considered an anomaly.
- **Baseline comparison** – a detector that compares the actual value of a data point with the value predicted based on a pre-constructed model. Such a method can, for example, learn the expected RTT for a node using 4G technology experiencing a certain RSRQ in a certain area, and correctly attribute changes in RTT to either contextual changes – like fallback from 4G to 3G – or to an unexplained anomaly. Due to a large parameter space the observed data point might come with a previously unseen context. To cope with such a case, RICERCANDO builds the model using the quantile regression forest technique [56] that predicts the dependent variable value even

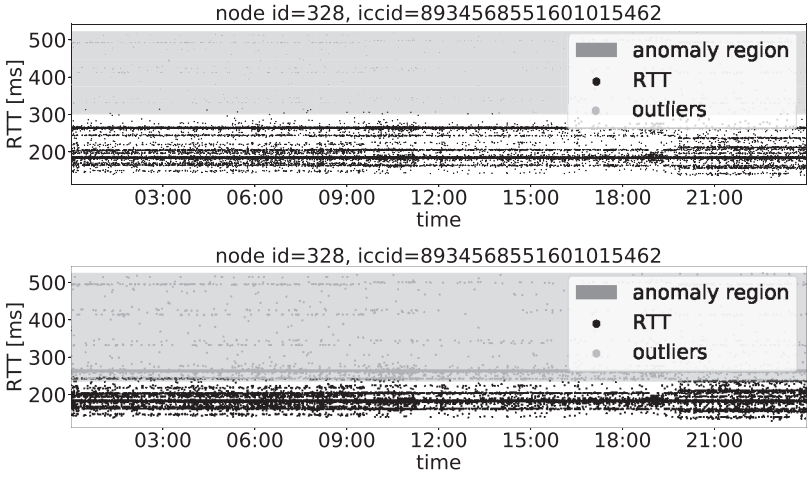

Fig. 5. Anomaly detector determined two different anomaly regions (higher RTT values shaded grey) within the same data by using distinct values of detection parameters in each case.

if the context has not been observed before. Furthermore, we build a model by using top N (by default 10) percent of the best performing measurements from a given context. This ensures that well performing points are not misclassified as anomalies.

- **Distribution comparison**[15] – a detector that empirically infers distributions of the same variable in different segments of the data using kernel density estimation technique, and then compares the distributions using Kullback-Leibler divergence. Significant difference between the previous and currently observed data distributions may indicate an anomaly.

The developed notebook allows the user to select a measurement node, a target KPI, and a time span in which the data is analysed. Additionally, the user can set a number of parameters that control the operation of the tool, including the sensitivity of anomaly detection. In the first step, the developed tool automatically detects the anomalies in measured data based on one of the above detection methods selected. Besides these methods, the tool supports a simple integration of new anomaly detectors. After one or more anomalies are detected, the tool enables informative visualisation of regular and anomalous data. Based on visual results a domain expert may adjust initial parameters to control the shape of the highlighted anomalies. This is demonstrated in Fig. 5 , where tuning of parameters produced two different anomaly regions within the same data. Descriptive visualisation also allows the experts to quickly find important aspects in the data. The data can then be saved so that anomalous regions are automatically labelled for further processing.

An important feature of the anomaly detection tool is concurrent anomaly detection. MBB data often contains measurements from a large number of nodes connected to a few different network providers, and detecting anomalies that simultaneously appear at all interfaces connected to the same provider is crucial for identifying whether the anomaly is isolated or affecting the whole network. In RICERCANDO we implement an optional concurrent analysis that takes into account all probes connected to a particular network. The output of the tool is a time diagram showing a cumulative count of anomalies over time for the selected network – moments when such a count is high indicate network-wide issues (see Section 5.4).

In all developed anomaly detection methods user can set various parameters. Identifying the relevant set of parameter values is indeed a complex task and must be done carefully by experts in order to enable automatic anomaly detection.

[15] This method is not suitable for streaming data analysis, therefore, we implement it in the Anomaly Detection module, but do not expose it through our GUI.

4.4. Advanced mining

Identifying root causes of the observed MBB behaviour is the final goal of data analysis. The existing tools for MBB data analysis were mostly developed by computer networking experts and support data preprocessing, visualisation, and simple statistical analysis [4,19,23,36]. RICERCANDO is developed in close collaboration with highly experienced data mining experts – one of the RICERCANDO authors is leading data mining research lab with more than 20 years of practical data mining experience in a range of domains. This synergy enables us to support advanced data mining for root cause analysis in RICERCANDO.

A key enabler of advanced mining in RICERCANDO is Orange – a popular GUI-based data mining toolbox where data processing workflows are constructed through visual programming by combing *widgets*. A widget is a computational unit with interactive visual interface that performs a particular function related to data preprocessing, visualisation, and modelling. Orange supports a range of machine learning methods, from unsupervised (clustering), to supervised (classifiers, regressions), from basic (e.g. naive Bayesian) to more complex state-of-the-art ones (e.g. neural networks). Fig. 6 depicts MONROE measurement data analysis using an Orange workflow of widgets.

Orange is limited in the amount of data it can handle. Thus, we use it as the last step of RICERCANDO analysis. We develop an Orange widget to import the data from RICERCANDO rapid exploration notebooks. Users can, thus, perform preliminary visualisation and analysis of a larger dataset in a Jupyter Notebook before selecting a particularly interesting dataset and analysing it further in Orange. In addition, we develop a widget for direct access to MONROE data stored in a time-series database.

One of the main questions a network analyst is interested in is *which factors may cause a particular anomaly?* [57]. To answer this, we develop an Orange widget that identifies *Significant Groups* of features that differentiate between regular and anomalous data. Note that a dataset containing labelled regular and anomalous data is automatically created by our Anomaly Detection module and imported to Orange via the iPython Connector widget. The main test implemented within the Significant Groups widget is the hypergeometric test. The test traverses all subsets of features and calculates the enrichment each subset brings to the anomalous data region. Sorting the subsets according to the enrichment, while also taking into account their significance levels, gives us a list of most probable causes for the detected anomaly. In addition, the

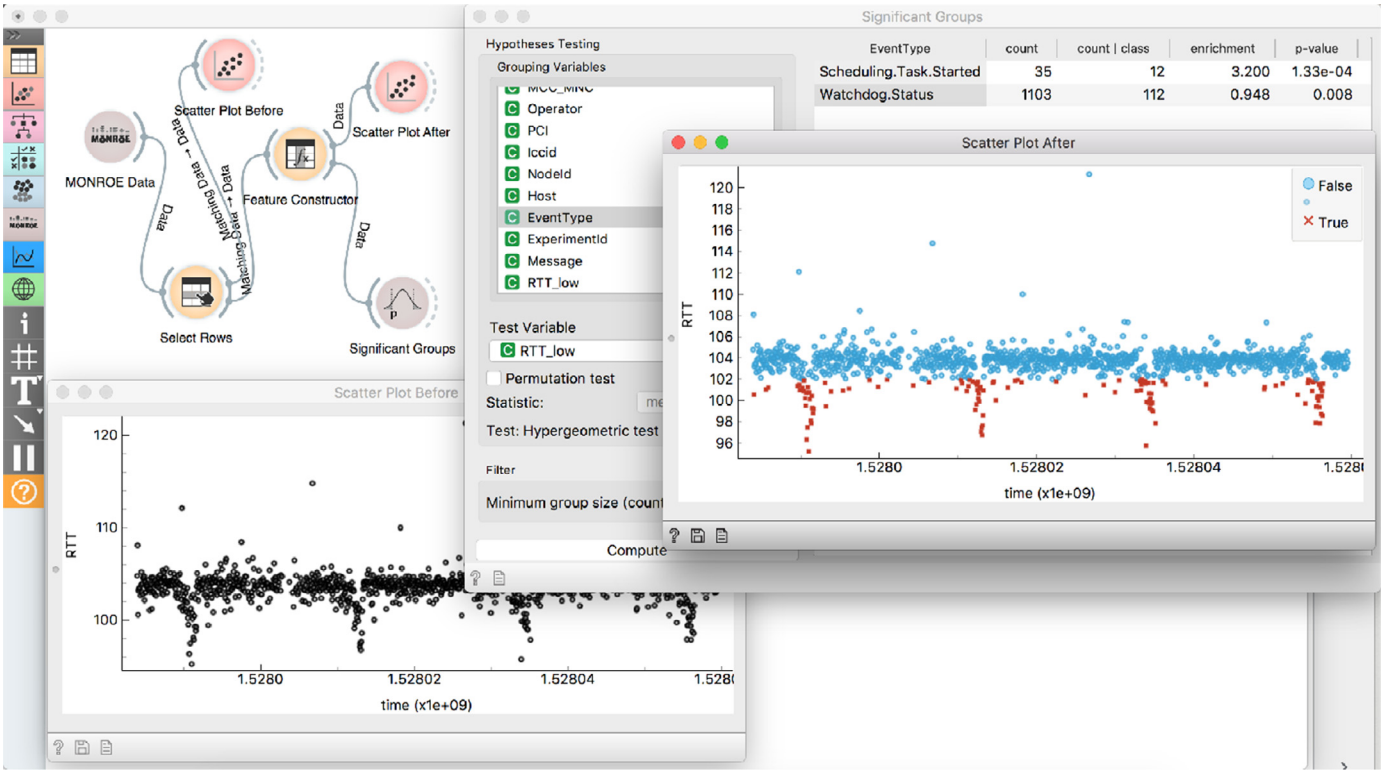

Fig. 6. MONROE data analysis in Orange. A workflow composed of Orange widgets is shown in the upper left corner. Each widget performs a specific function. A window corresponding to *Scatter Plot Before* widget (lower left) shows anomalous RTT behaviour. *Scatter Plot After* window (middle right) shows distinct RTT dips. *Feature Constructor* widget is used for splitting the data into groups with low and normal RTT. Finally, *Significant Groups* widget performs a hypergeometric test and identifies `Scheduling.Task.Started` event as a feature value that discerns between the two groups, indicating that a background experiment impacts the observed RTT.

widget supports other comparison tests that may help with root cause analysis, such as the permutation test and the *t*-test.

5. Case studies

The MONROE project provides large amount of data of various MBB network parameters. Irregular patterns in the data can quickly be spotted using the visualisations. However, to precisely define visually observed anomalies and to discover hidden anomalies that are not easy to illustrate, we developed an automatic anomaly detector. Beside identification of anomalies the computer tool also facilitates the determination of their root causes. Among multiple occurrences of anomalies that we found, selected case studies focused on RTT data are thoroughly described in this section.

5.1. Connection mode change

The first anomaly we identified by using our automatic detection tool's rolling mean method is depicted in Fig. 7 (top). The figure shows that on the given measurement node after 11:30 the RTT mean changes drastically from below 100 ms to approximately 250 ms. The anomaly detector automatically recognised the shift and marked it as an anomaly (grey region). Running the hypergeometric test and calculating the enrichment each feature subset brings to the anomalous data region, we found out that a change in the device's connectivity mode is the culprit. A switch from LTE to 3G perfectly coincides with the anomaly, as shown in Fig. 7 depicting the RTT and the interface's mode on the common time axis. Note that by automating the significant feature search we remove the need for comprehensive visual analysis.

This example shows the limitations of the automated approach relying on domain-agnostic data deviation detection (see discussion in Section 4.3). In Section 5.3 we present a model-based approach, which, armed with the knowledge based on the previously seen data, correctly considers the above example to be non-anomalous, as it can be easily explained through the network interface mode change.

5.2. Measurement system interference

In many instances we encountered sudden short-lasting drops in the measured RTT. Fig. 8 shows RTT measurements within two hours from 20:00 to 22:00 on one of the interfaces. The majority of measurements have values near 100 ms, but between 21:05 and 21:20 there is a concentrated group of measurements with values around 80 ms. The shaded area marks an anomalous group which was identified by our rolling mean detector. Many dispersed outliers can be seen in Fig. 8, yet only a cluster with a sufficient number of outliers composes an anomaly. In such situations the detection using the computer tool is more accurate than just a visual observation of data.

The significance analysis for this case shows that the root cause of this anomaly is the event `Scheduling.Task.Started` (Fig. 9), indicating that a start of an experiment on a node causes the anomaly. It seems that running an experiment on a node triggers a drop in measured RTT values.

We hypothesise that the cause of such behaviour is the discontinuous reception (DRX) mode. DRX allows interfaces to save energy by going to a low power mode when no data is being transmitted [58]. However, DRX may lead to the RTT increase if the ping packets, before the transmission, have to wait for the interface to go back to a high power state. MONROE platform pings are sent out with 1 second inter-packet time, while operators often set the DRX kick-in threshold to around 100ms. Consequently, we expect that most of the MONROE ping packets, unless an interface is already active because of an experiment, indeed have

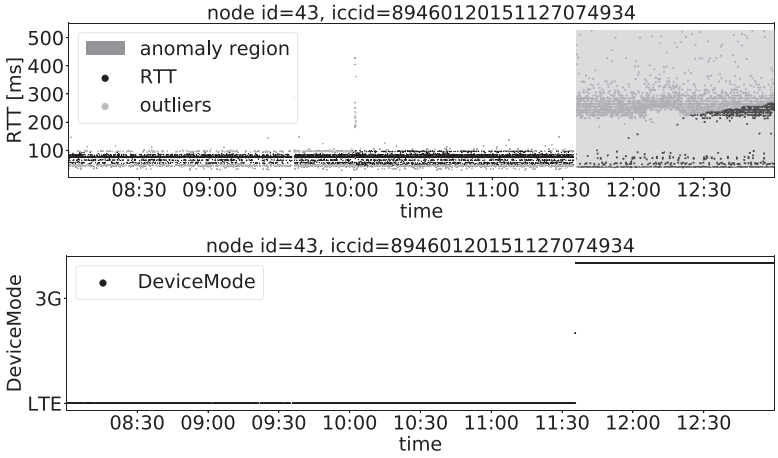

Fig. 7. The image on top shows the increase of RTT measurements after 11:30 and the bottom image shows how the shift correlates with the change of parameter DeviceMode. The vast majority of RTT values before 11:30 are around 100 ms, so the relatively rare outliers at that time do not form an anomaly.

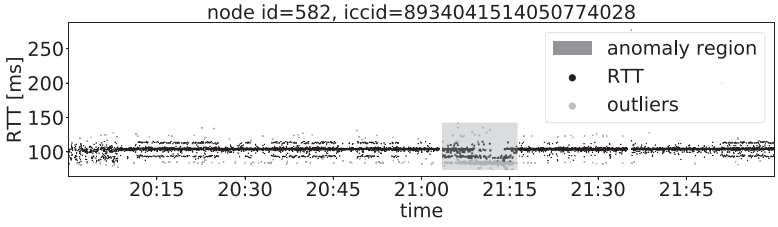

Fig. 8. The anomaly detected via rolling mean method is marked with a shaded area. Occasional outliers are coloured grey, however, they do not necessarily compose an anomaly.

Iccid,Variable	count	count | class	enrichment	p-value
8934041514050774028,EventType=Scheduling.Task.Started	1791	366	1.15067	7.0252e-21
8934041514050774028,Host=130.243.27.221	6289	372	1	0
8934041514050774028,Operator=YOIGO	6289	372	1	0
8934041514050774028,IP_Address=10.41.236.41	6289	372	1	0
8934041514050774028,Band=3	6287	372	1	0
8934041514050774028,CID=72209510	6289	372	1	0
8934041514050774028,DeviceMode=LTE	6289	372	1	0
8934041514050774028,DeviceState=connected	6287	372	1	0
8934041514050774028,Frequency=1800	6289	372	1	0
8934041514050774028,DeviceSubmode=unknown	6287	372	1	0
8934041514050774028,LAC=65535	6287	372	1	0
8934041514050774028,PCI=65535	6287	372	1	0

Fig. 9. Significance analysis determines the event `Scheduling.Task.Started` as the root cause of the anomaly shown in Fig. 8.

to wait for the interface to go to the high power state before the RTT measurements can be performed. To confirm the existence of DRX we conducted our own ping experiments on the MONROE platform with variable inter-packet times. As expected, once ping packets were sent out with a higher frequency, the measured RTT dropped.

5.3. Baseline model anomaly detection

The rolling mean anomaly detection method is limited in its ability to adapt to well understood changes in the observed variable. For instance, in the case examined in Section 5.1 the jump in ping RTT measurements is not unusual, having in mind the device connection mode change. On the other hand, the baseline method for anomaly detection uses a pre-constructed quantile regression tree model to infer the expected value of the observed parameter in the light of the given context, i.e. values of selected remaining parameters. Consequently, the method does not mark as anomalous those measurements that can be explained with the pre-constructed model. This greatly reduces the number of false positives, as an "anomaly" can, in fact, be explained by the model.

In Fig. 10 we show model-predicted values of ping RTT (black line) and the observed values (grey dots). The baseline model takes in to account RSSI (Received Signal Strength Indicator), RSRQ (Reference Signal Received Quality) and RSRP (Reference Signal Received Power) as independent variables. The prediction was created by quantile regression forests algorithm, taking into account the top percentile of predicted RTT values. Here the higher percentile indicates the better (i.e. smaller) RTT value. The outliers are the points distant from the baseline, meaning that their actual value highly disagrees with predicted value. In top image are two shaded anomaly regions formed by outliers high above the baseline.

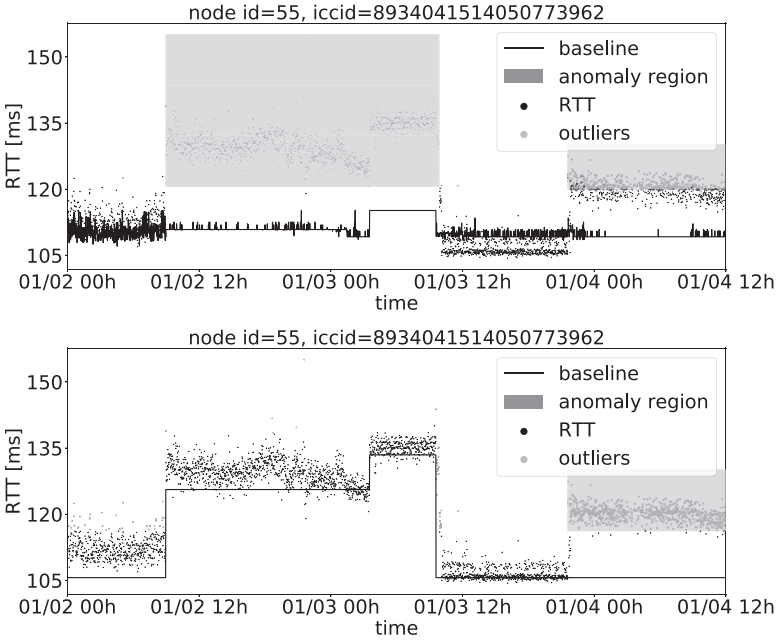

Fig. 10. In the image on top the baseline is the predicted value of RTT with respect to parameters RSSI, RSRQ, and RSRP. The detector marked two anomalies. The first anomaly is resolved by using CID to construct the baseline as shown on bottom image.

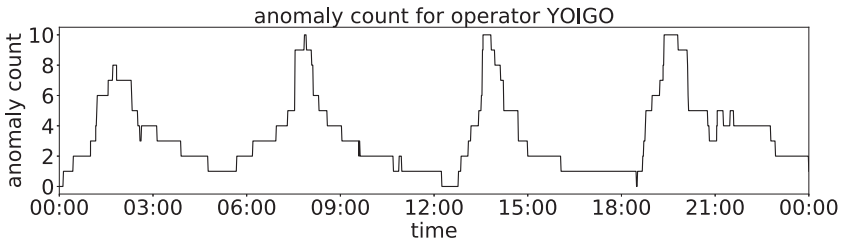

Fig. 11. Number of simultaneously occurring anomalies at all nodes connected to the same ISP.

The first anomaly can be explained by further refining the model via retraining with the information on the expected RTT at different cell IDs (CID) that a device connects to. This is clarified in bottom image in Fig. 10, where the baseline was constructed by quantile regression forests algorithm that predicts RTT values with respect to CID. The three steps in the baseline function in the bottom image correspond to three different cells that a device connected to. Therefore, the first anomaly constructed in top image is not considered an anomaly, if the CID parameter is taken into account. Note that the figure still does not explain why RTT measurements differ across different CIDs – this requires further investigation that goes beyond the capabilities of the collected dataset. The second anomaly on the right side of both images in Fig. 10, however, is not due to different CID, so its root cause is the variation in parameters other than CID, RSSI, RSRQ, or RSRP. In this way the baseline anomaly detector not only uncovers anomalies that are impossible to detect visually, but can also explain anomalies by choosing the appropriate independent variables for the quantile regression forests algorithm.

5.4. Network and system-wide anomalies

We are further interested to determine whether a certain anomaly appears only at a particular network interface or, perhaps, at a number of interfaces connected to the same Internet service provider (ISP), or even beyond – in a number of devices across the measurement system. Such a case could indicate a systemic cause of the anomalies, similar to the real-world example of network-wide outage from the opening paragraph of this paper. In order to study such examples we enhanced our anomaly detection tool to support concurrent anomaly detection over a number of interfaces – essentially, it counts all anomalies happening at the same time at nodes connected to the same ISP. Fig. 11 shows the number of anomalies that occurred simultaneously at all nodes connected to ISP YOIGO on June 2018. A pattern of periodic spikes can be observed. This anomaly is due to an RTT drop caused by a MONROE platform experimenter running heavy experiments, similarly to the case examined in Section 5.2. The large number of concurrent anomalies at spikes correspond to experiments scheduled to run on different nodes of the same operator at the same time.

We further examined potential network-wide anomalies. Through exploratory analysis at a few interfaces we noticed an anomaly caused by missing data. We then ran the concurrent anomaly detection tool for all the interfaces connected to a few different ISPs. In Fig. 12 we show the cumulative anomaly count for two different ISPs – Vodafone IT and YOIGO. We see that both operators exhibit simultaneous peaks that are more than two standard deviations above the mean anomaly count. The same peak is observable with other ISPs (not shown in the figure). This indicates a system wide anomaly, likely caused by a glitch in the measurement system.

6. Lessons learnt

Continuous experimentation and revising has marked the process of RICERCANDO design and development. Different prototypes have been developed, applied on the data, and evaluated, while at the same time the underlying measurement platform (MONROE) kept evolving, essentially making our goal a moving target. In this section we present some of the main lessons learnt through the development process.

Need for appropriate data preprocessing and representation. At the time RICERCANDO started in June 2016 the MONROE platform was producing only a modest amount of (meta) data from a limited number of

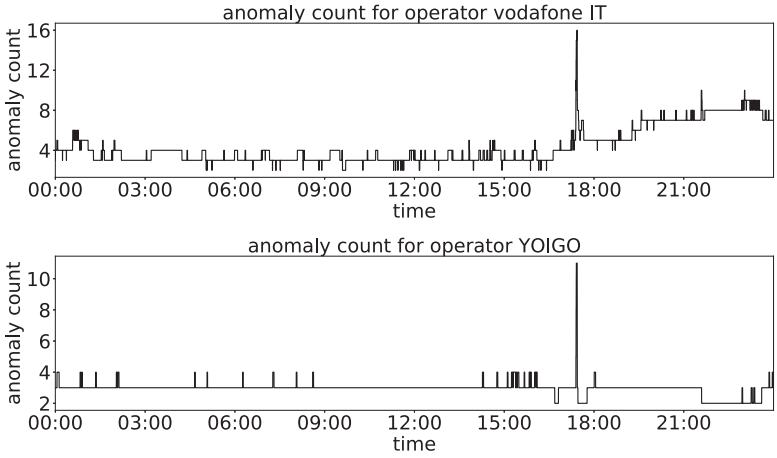

Fig. 12. A system-wide anomaly due to the missing RTT data at approximately 17:30 on January 1st 2018.

nodes. However, over the course of the project the amount of collected data grew both because additional nodes were deployed, as well as because of the additional information that was collected on each node (e.g. different background experiments). MONROE data are by default stored in a Cassandra no-SQL database. This, however, severely limits large-scale data mining of the platform data. While no-SQL databases enables easy storage of key-value pairs, they are inappropriate for mining temporal data. Most of the collected data indeed have a temporal dimension, thus time-based querying remains crucial. Another issue with no-SQL databases is that they often do not support data sampling. In MONROE, data are often collected with very fine granularity (e.g. a ping every second), which makes (visual) inspection over a larger time period impractical – there are simply too many points to be shown on a graph. In the early stages of RICERCANDO we tried to adapt to the given database. However, in the next step, in order to enable efficient temporal large data analysis we devised a solution that relies on InfluxDB, a database specifically designed for time-series data querying.

Joining tables over the common timestamp field is another challenge we have faced. Since timestamps are asynchronous, some tolerance on timestamp joining had to be accounted for. One solution was sampling data at rounded timestamps directly on the database, which we also used for visualisation. Another solution was provided by `mergeasof`, a function from the `pandas` module, which is similar to a left-join. Here, however, we use it to match on the nearest backward timestamp with a defined temporal tolerance between potentially merged instances. This helped us obtain more meaningful data points with fewer missing values. Data preprocessing and representation is usually the most difficult step, especially when dealing with large amounts of data. Our contribution, released in a form of processing scripts automates this step and streamlines further mining of MBB measurement data.

Available data imposes explanation capacity limits. The interpretation of some encountered anomalies eluded us. One of these is depicted in Fig. 13 . A drop in mean RTT value occurs around 7:00, similar to the case of ping experiment running on the node (Fig. 8). However, there were no scheduled experiments in the case in Fig. 13, so they are ruled out as root-cause of the anomaly. Also, the 2-hour extent of this anomaly is longer than the 10-minute duration of an experiment. Furthermore, the anomaly appeared only at one interface of the same node. The available data is simply insufficient for explaining this anomaly. More information, perhaps from specific operational logs of this particular device, are needed.

Effects of mobile broadband measurement system on the results. Uncovering the role of seemingly unrelated system design decisions on KPI values is one of the key observations we arrived to, as we tested RICERCANDO on MONROE data. For instance, after significant amounts of meta-data started arriving from MONROE nodes we discovered that RTT exhibits occasional spikes (going above 5X the usual value) interspersed with lost ping packets. Further analysis with our Rapid Exploration tools uncovered correlation between the observed anomaly and the node resource utilisation spikes, indicating potential executions of CPU-heavy experiments. Consequently, our suggestion to include experiment execution information in the metadata was implemented by the MONROE team, which later allowed us to pinpoint a particular experiment that resulted in the observed RTT behaviour. This is just one example where the measurement system, in this case through heavy resource usage by an experiment, resulted in anomalous measurements. The impact of the background traffic on RTT measurements via DRX mode toggling is another example of the coupling of the measurement methodology and the recorded result, and is explained in Section 5.

We further revealed that geographical and Internet coordinates of the measurement equipment impacts the observed measurement values. For instance, in our testbed, all nodes and all interfaces were sending ping probes to the same destination host IP of a server at Karlstad University, Sweden. We noticed that the nodes located in Norway and Sweden often had the mean RTT of the ping probes in the range between 40 ms and 60 ms, while it was not uncommon for the nodes in countries far from the destination host server to encounter mean RTT close to 100 ms. This observation precludes a cross-node anomaly detection approach. Thus, rather than comparing the absolute difference of feature values among distant nodes, we concentrated on individual modelling and detection of relative changes in feature values recorded at a single node.

MBB measurement data analysis requires multidisciplinary expertise. While we were already aware of the need for interdisciplinary expertise at the time we laid out plans for RICERCANDO, this need became even more evident as we progressed with development. First, MBB data is often analysed by computer networking domain experts. The need for expertise in data mining, in particular in data representation, statistical analysis, and geographical data analysis proved crucial and the data mining part of our team got several enquiries to help with other projects' data analysis issues. The two fields, data mining and computer networking, are seldom directly collaborating, and it is our hope that RICERCANDO results might facilitate this collaboration. Second, even when the general knowledge of networking is present, MBB measurement data mining requires in-depth knowledge of latest practices in broadband networks' implementation. Such knowledge is often available only with a close collaboration with relevant industrial players. Specifically, our identification of the DRX-related anomaly would not be possible without close collaboration with an industry professional experienced with LTE networks. Finally, visualisation of MBB measurement data, a crucial aspect of RICERCANDO, was based on lessons learnt from our data mining group's previous efforts in big data visualisation [59,60].

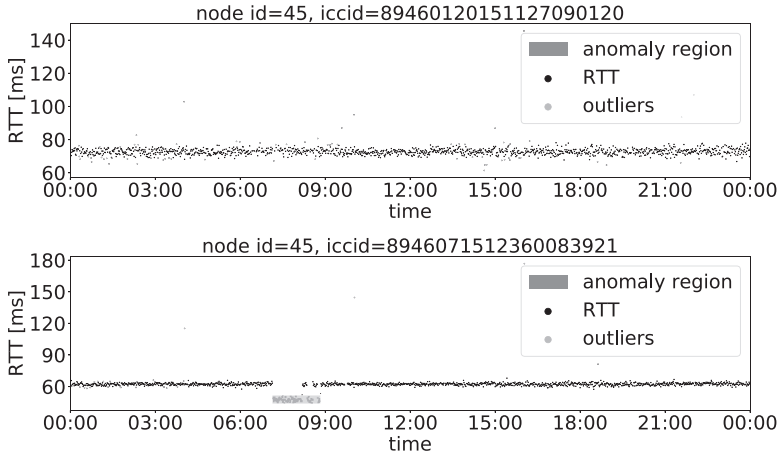

Fig. 13. Anomaly occurs only at one interface (bottom image) of the same node.

7. Conclusions

In this paper we presented RICERCANDO – an MBB measurement data mining toolkit developed in close collaboration of networking and machine learning experts. RICERCANDO goes beyond the existing tools by allowing rapid iterative visual analysis and rigorous advanced data mining of MBB data. The developed approach is founded in intelligent time-series data storage, re-sampling, and merging, followed by interactive visualisation methods that enable quick focus on a particular measurements of interest. Machine learning modelling then enables automated anomaly detection and root cause analysis via rigorous statistical methods.

Compared to the existing attempts at MBB data analysis, such as [23–25], RICERCANDO does not provide merely descriptive statistics about the underlying data and an implementation of a pre-selected mining technique. Rather, through integration with a full-fledged data mining suite, Orange, RICERCANDO enables a vast array of data mining techniques. The benefit of these techniques to not only identify, but also explain unusual network behaviour is evident in Section 5.2 where the significance analysis is used to pinpoint the reason for the change in RTT values. RICERCANDO's visualisation toolbox incorporates time-series and geo-based visualisation. The richness of options in RICERCANDO's visualisation toolbox is not on a par with the options provided by popular solutions, such as Grafana, yet, neither was the provision of such options one of our design goals. Instead, compared to visualisation-only solutions, RICERCANDO fully integrates with the data mining pipeline allowing interactive analysis through Pandas Dataframe-based communication. Finally, RICERCANDO can be compared with other MBB measurement data anomaly detection tools [29,30,32,36]. Most of these solutions focus on different methods for identifying unusual behaviour on a single measurement node, either through statistical means [30] or through in-depth knowledge of the underlying networking protocols [36]. In RICERCANDO we harness machine learning modelling (e.g. quantile regression forest [56]) and also provide a bird's eye view of the whole dataset. This is particularly evident in Section 5.4 where we demonstrate how network-wide anomalies can be detected with RICERCANDO.

RICERCANDO represents a holistic solution for network measurement analysis, yet, its modular design naturally supports framework expansion and evolution. Augmenting the anomaly detection module with deep learning (DL) techniques, something that we are already working on, demonstrates this expandability. Deep learning relies on large amounts of data in order to tune the models' numerous parameters. With the constant stream of new values sampled at high frequency, MBB traffic measurements are a great candidate for DL-based modelling. DL methods such as recurrent neural networks (RNNs) and deep Boltzmann machines (DBMs) have been used to model time series data and recognise anomalous events related to network security [61]. In our work, we will concentrate on the autoencoder (AE), a technique that relies on the dimensionality reduction to compress the representation of the usual network traffic. When this AE is then fed with new measurements, any failure to compress and reconstruct the measurements through the AE indicates an anomaly [62]. Compared to the approaches we have implemented in Section 4.3, the AE-based approach implicitly learns what the normal data should look like and is thus more likely to identify even previously unseen and unusual anomalies.

In this we present a number of use cases demonstrating the usability of the framework for anomaly detection and explanation. Although the framework was designed primarily for the analysis of data collected in MONROE testbed, its usability is by no means restricted to a particular dataset. We have already harnessed RICERCANDO for mining MBB measurement data gathered by the Slovenian Agency for Telecommunications (AKOS) with the goal of inferring Internet neutrality violations in Slovenia. Moreover, although targeting MBB measurements, certain parts of our framework could also be used in other environments, especially those characterised by heterogeneous measurements and measurements generated by a large number of probes. For instance, Internet Service Providers (ISP) usually manage large fixed access networks comprised of diverse sub-networks. RICERCANDO's geo-visualisation and anomaly detection tools can assist ISPs in rapidly detecting and localising issues within their network. Similarly, in datacenter environments, where different performance metrics (e.g. delay, throughput, packet loss, etc.) need to be tracked, RICERCANDO's anomaly detection module could provide support in detecting and explaining performance problems.

RICERCANDO toolbox has a great potential to assist commercial telcos and government regulators with monitoring and understanding MBB traffic, and we invite interested parties to download RICERCANDO[16], adapt it to their needs, enrich it with additional functionalities, and further contribute towards improved network measurement data analysis and understanding.

Declaration of Competing Interest

The authors declare that they do not have any financial or nonfinancial conflict of interests

Acknowledgements

The authors would like to thank Prof Fabio Ricciato for his guidance during the planing, execution, and writing about the work presented in

[16] http://github.com/ivek1312/ricercando/

this paper; the data mining team from the University of Ljubljana, including Jernej Kernc, Vesna Tanko, and Anže Starič for their contributions to RICERCANDO; to Janez Sterle for his help with the explanation of the observed network anomalies; and to David Modic for his feedback on an earlier draft of this paper.

References

[1] Guardian, O2 outage: more than 30m mobile customers unable to get online, 2018.
[2] Statista, Number of active mobile broadband subscriptions worldwide from 2007 to 2017 (in millions), 2018.
[3] I.T. Union, ICT Facts and Figures, 2017.
[4] P. Casas, P. Fiadino, S. Wassermann, S. Traverso, A. D'Alconzo, E. Tego, F. Matera, M. Mellia, Unveiling network and service performance degradation in the wild with mplane, IEEE Commun. Mag. 54 (3) (2016) 71–79.
[5] A. Finamore, M. Mellia, M. Meo, M.M. Munafo, P. Di Torino, D. Rossi, Experiences of internet traffic monitoring with tstat, IEEE Netw. 25 (3) (2011) 8–14.
[6] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, Z.M. Mao, Mobilyzer: an open platform for controllable mobile network measurements, Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2015, pp. 389–404.
[7] N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, N. Weaver, V. Paxson, Beyond the radio: Illuminating the higher layers of mobile networks, Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2015, pp. 375–387.
[8] N. Vallina-Rodriguez, A. Auçinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, J. Crowcroft, RILAnalyzer: a comprehensive 3G monitor on your phone, Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013, pp. 257–264.
[9] U. Goel, M.P. Wittie, K.C. Claffy, A. Le, Survey of end-to-end mobile network measurement testbeds, tools, and services, IEEE Commun. Surv. Tutor. 18 (1) (2016) 105–123.
[10] Ookla, SpeedTest Mobile Apps.
[11] P. Svoboda, M. Laner, J. Fabini, M. Rupp, F. Ricciato, Packet delay measurements in reactive IP networks, IEEE Instrument. Measur. Mag. 15 (6) (2012) 36–44, doi:10.1109/MIM.2012.6365543.
[12] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato, M. Rupp, A comparison between one-way delays in operating HSPA and LTE networks, in: 2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012, pp. 286–292.
[13] S. Muckaden, MySpeedTest: active and passive measurements of cellular data networks, Georgia Institute of Technology, 2013 Ph.D. thesis.
[14] M. Hirth, T. Hoßfeld, M. Mellia, C. Schwartz, F. Lehrieder, Crowdsourced network measurements: benefits and best practices, Comput. Netw. 90 (2015) 85–98.
[15] OpenSignal, Inc., OpenSignal Methodology.
[16] J. Ryckaert, P. De Doncker, R. Meys, A. de Le Hoye, S. Donnay, Channel model for wireless communication around human body, Electron. Lett. 40 (9) (2004) 543–544.
[17] C. Arthur, Steve Jobs solves iPhone 4 reception problems: 'don't hold it that way'.
[18] O. Alay, A. Lutu, D. Ros, R. Garcia, V. Mancuso, A.F. Hansen, A. Brunstrom, M.A. Marsan, H. Lonsethagen, MONROE: Measuring mobile broadband networks in Europe, in: Proceedings of the IRTF & ISOC Workshop on Research and Applications of Internet Measurements (RAIM), 2015.
[19] G. Iannaccone, Fast prototyping of network data mining applications, in: Passive and Active Measurement Conference, 2006, pp. 41–50.
[20] M. Wullink, G.C. Moura, M. Müller, C. Hesselman, ENTRADA: A high-performance network traffic data streaming warehouse, in: Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP, IEEE, 2016, pp. 913–918.
[21] A. Bär, A. Finamore, P. Casas, L. Golab, M. Mellia, Large-scale network traffic monitoring with DBStream, a system for rolling big data analysis, in: Big Data (Big Data), 2014 IEEE International Conference on, IEEE, 2014, pp. 165–170.
[22] P. Romirer-Maierhofer, M. Schiavone, A. D'Alconzo, Device-specific traffic characterization for root cause analysis in cellular networks, in: International Workshop on Traffic Monitoring and Analysis, Springer, 2015, pp. 64–78.
[23] D. Baltrunas, A. Elmokashfi, A. Kvalbein, Measuring the reliability of mobile broadband networks, in: Proceedings of the 2014 Conference on Internet Measurement Conference, ACM, 2014, pp. 45–58.
[24] A. Lutu, Y.R. Siwakoti, Ö. Alay, D. Baltrūnas, A. Elmokashfi, The good, the bad and the implications of profiling mobile broadband coverage, Comput. Netw. 107 (2016) 76–93.
[25] A. Narayanan, S. Verma, Z.-L. Zhang, Mining spatial-temporal geomobile data via feature distributional similarity graph, in: Proceedings of the First Workshop on Mobile Data, ACM, 2016, pp. 13–18.
[26] M.D. Mauro, C.D. Sarno, Improving SIEM capabilities through an enhanced probe for encrypted Skype traffic detection, J. Inf. Secur. Appl. 38 (2018) 85–95, doi:10.1016/j.jisa.2017.12.001.
[27] C. Callegari, A. Coluccia, A. D'Alconzo, W. Ellens, S. Giordano, M. Mandjes, M. Pagano, T. Pepe, F. Ricciato, P. Zuraniewski, A Methodological Overview on Anomaly Detection, in: Data Traffic Monitoring and Analysis, 7754, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 148–183, doi:10.1007/978-3-642-36784-7_7.
[28] P. Fiadino, A. DAlconzo, M. Schiavone, P. Casas, Rcatool-a framework for detecting and diagnosing anomalies in cellular networks, in: Teletraffic Congress (ITC 27), 2015 27th International, IEEE, 2015, pp. 194–202.
[29] F. Ricciato, F. Vacirca, P. Svoboda, Diagnosis of capacity bottlenecks via passive monitoring in 3G networks: an empirical analysis, Comput. Netw. 51 (2007) 1205–1231, doi:10.1016/j.comnet.2006.07.011.
[30] A. Coluccia, A. D'Alconzo, F. Ricciato, Distribution-based anomaly detection via generalized likelihood ratio test: a general maximum entropy approach, Comput. Netw. 57 (17) (2013) 3446–3462, doi:10.1016/j.comnet.2013.07.028.
[31] A. D'Alconzo, A. Coluccia, F. Ricciato, P. Romirer-Maierhofer, A distribution-based approach to anomaly detection and application to 3G mobile traffic, in: GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference, 2009, pp. 1–8, doi:10.1109/GLOCOM.2009.5425651.
[32] Y. Li, J. Sun, W. Huang, X. Tian, Detecting anomaly in large-scale network using mobile crowdsourcing, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 2179–2187.
[33] G. Zargarian, L. Vassio, M.M. Munafò, M. Mellia, Mining patterns in mobile network logs, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2019, pp. 1–6.
[34] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, et al., Orange: data mining toolbox in python, J. Mach. Learn. Res. 14 (1) (2013) 2349–2353.
[35] F. Ahmed, J. Erman, Z. Ge, A.X. Liu, J. Wang, H. Yan, Detecting and localizing end-to-End performance degradation for cellular data services based on TCP loss ratio and round trip time, IEEE/ACM Tran. Netw. (TON) 25 (6) (2017) 3709–3722.
[36] M. Siekkinen, G. Urvoy-Keller, E.W. Biersack, D. Collange, A root cause analysis toolkit for TCP, Comput. Netw. 52 (9) (2008) 1846–1858.
[37] F. Michelinakis, N. Bui, G. Fioravantti, J. Widmer, F. Kaup, D. Hausheer, Lightweight mobile bandwidth availability measurement, IFIP Network. Conf. (IFIP Networking) (2015) 1–9.
[38] P. Fox, J. Hendler, Changing the equation on scientific data visualization, Science 331 (6018) (2011) 705–708.
[39] F. Ricciato, Traffic monitoring and analysis for the optimization of a 3G network, IEEE Wireless Commun. 13 (6) (2006) 42–49.
[40] F. Li, A.A. Niaki, D. Choffnes, P. Gill, A. Mislove, A large-scale analysis of deployed traffic differentiation practices, in: SIGCOMM, Beijing, China, 2019, pp. 130–144.
[41] Ö. Alay, A. Lutu, R. García, M. Peón-Quirós, V. Mancuso, T. Hirsch, T. Dely, J. Werme, K. Evensen, A. Hansen, et al., Measuring and assessing mobile broadband networks with MONROE, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A, IEEE, 2016, pp. 1–3.
[42] C. Churilo, InfluxDB vs. Elasticsearch for time series data and metrics benchmark.
[43] J. Wu, P.P. Lee, Q. Li, L. Pan, J. Zhang, Cellpad: detecting performance anomalies in cellular networks via regression analysis, in: 2018 IFIP Networking Conference (IFIP Networking) and Workshops, IEEE, 2018, pp. 1–9.
[44] D.J. Hand, Principles of data mining, Drug Saf. 30 (7) (2007) 621–622.
[45] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI Magaz. 17 (3) (1996) 37–54.
[46] A. Ng, AI transformation playbook, Technical Report, Landing AI, 2018.
[47] A.S. Fiaz, N. Asha, D. Sumathi, A.S. Navaz, Data visualization: enhancing big data more adaptable and valuable, Int. J. Appl. Eng. Res. 11 (4) (2016) 2801–2804.
[48] Tableau Software, Tableau.
[49] J. Bois, High level plotting with HoloViews, 2019.
[50] A.K. Marnerides, A. Schaeffer-Filho, A. Mauthe, Traffic anomaly diagnosis in internet backbone networks: a survey, Comput. Netw. 73 (2014) 224–243.
[51] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, S. Savage, Inferring internet denial-of-service activity, ACM Trans. Comput. Syst. (TOCS) 24 (2) (2006) 115–139.
[52] D. Brauckhoff, X. Dimitropoulos, A. Wagner, K. Salamatian, Anomaly extraction in backbone networks using association rules, IEEE/ACM Trans. Netw. (TON) 20 (6) (2012) 1788–1799.
[53] I.C. Paschalidis, G. Smaragdakis, Spatio-temporal network anomaly detection by assessing deviations of empirical measures, IEEE/ACM Trans. Network. (TON) 17 (3) (2009) 685–697.
[54] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. (CSUR) 41 (3) (2009) 1–58.
[55] P. Romirer-Maierhofer, F. Ricciato, Towards anomaly detection in one-way delay measurements for 3G mobile networks: A preliminary study, in: International Workshop on IP Operations and Management, Springer, 2008, pp. 1–14.
[56] N. Meinshausen, Quantile regression forests, J. Mach. Learn. Res. 7 (2006) 983–999.
[57] A. Lakhina, M. Crovella, C. Diot, Characterization of network-wide anomalies in traffic flows, in: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, ACM, 2004, pp. 201–206.
[58] D. Vinella, M. Polignano, Discontinuous reception and transmission (DRX/DTX) strategies in long term evolution (LTE) for voice-Over-IP (VOIP) traffic under both full-dynamic and semi-persistent packet scheduling policies, Project Group 996 (2009).
[59] G. Leban, B. Zupan, G. Vidmar, I. Bratko, Vizrank: data visualization guided by machine learning, Data Min. Knowl. Discov. 13 (2) (2006) 119–136.
[60] M. Možina, J. Demšar, M. Kattan, B. Zupan, Nomograms for visualization of naive Bayesian classifier, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2004, pp. 337–348.
[61] D. Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, K.J. Kim, A survey of deep learning-based network anomaly detection, Cluster Comput. (2017) 1–13.
[62] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 665–674.

Veljko Pejović received his PhD in computer science from the University of California, Santa Barbara, USA in 2012 on the topic of resource-efficient wireless communication for rural areas. From 2012 to 2014 Dr Pejović worked as a research fellow at the University of Birmingham, UK in the area of mobile computing and sensing. His work on modelling users' movement and communication behaviour from mobile call records has won the 2013 Orange Data for Development Challenge, while his work on developing machine learning models of interruptibility based on sensor data resulted in the best paper nomination at the 2014 ACM UbiComp conference. Currently, he is an assistant professor at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia, where he works on mobile sensing and resource-efficient mobile computing.

Ivan Majhen received the Diploma degree in computer and information science from the University of Ljubljana, Slovenia. He has been working on WiFi-Direct implementation for Arduino platform at Jozef Stefan Institute, Slovenia in 2014. In 2018, he joined Computer Communications Laboratory at Faculty of computer and information science, University of Ljubljana. He is currently focusing on wireless sensors for human vital signs monitoring.

Miha Janež received the PhD in computer science from the University of Ljubljana, Slovenia in 2012 on the topic of circuit layout design. He is an assistant at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia. His research interests include the design of wireless sensor networks using accessible hardware components and the analysis of the data collected in large networks.

Blaž Zupan heads the bioinformatics lab at University of Ljubljana and is an Associate Professor at the Baylor College of Medicine in Houston. His research has focused on constructive induction, machine learning and epistasis approaches to reconstruction of gene networks, large-scale data fusion, and data visualizations. He believes that crafting simple tools that anybody can use to understand data is essential to advancements of humanity and democracy. His lab is developing Orange, a fully open-source, ever evolving data mining suite with a visual programming environment. He also enjoys writing scripts for YouTube videos to explain data science, and preparing courses that introduce data science.