Service-based Analytics for 5G Open Experimentation Platforms

Erik Aumayr[a], Giuseppe Caso[b], Anne-Marie Bosneag[a], Almudena Diaz Zayas[d], Özgü Alay[b,c], Bruno Garcia[d], Konstantinos Kousias[b], Anna Brünstrom[e], Pedro Merino Gomez[d], Harilaos Koumaras[f]

[a]*Ericsson Ireland Research Lab, Ireland*
[b]*Simula Metropolitan Center for Digital Engineering, Norway*
[c]*University of Oslo, Norway*
[d]*University of Malaga, Spain*
[e]*Karlstad University, Sweden*
[f]*National Centre of Scientific Research Demokritos, Athens*

Abstract

A scalable, flexible and reliable Analytics service has become a requirement toward building efficient Fifth Generation (5G) experimental platforms that can support a suite of end-user experiments and verticals. Our paper presents the challenges that come with designing such a service-based Analytics component, and shows how we have used it in the context of open experimental platforms in the *5GENESIS* project. Our Analytics service was designed both for enabling the efficient setup and configuration of the underlying platform, and also for ensuring that it provides useful insights into the experimentation Key Performance Indicators (KPIs) toward the end-user. Thus, Analytics proved to be a useful tool across several stages, starting from ensuring correct operation during the initial phases of the network setup and continuing into the normal day-to-day experimentation. Our experiments show how the tool was used in our setup and provide information on how to apply it to different environments. The Analytics component, designed as a set of microservices that serve several goals in the analytics workflow, is also provided as open source, being part of the *Open5Genesis* suite.

Keywords: 5G experimental platforms, Analytics frameworks, Microservices architecture, Open source

1. Introduction

Fifth Generation (5G) cellular systems are expected of being able to support different service types, i.e., enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC), in a seamless manner [1]. This is mainly achieved via radio plane enhancements and by enabling virtualization and dynamic deployment of system resources, which enable the needed system flexibility but also introduce further complexity in terms of system management. Throughout the different stages of a network setup and operation, a flexible, scalable, and reliable analytics service that can support network operations and give proper insights becomes a key system component, with the aim of supporting operational efficiency and optimising network performance [2].

Such an Analytics service is also vital for ensuring compliance with Service Level Agreements (SLAs). The SLA is usually defined as a set of end-to-end Key Performance Indicators (KPIs) that must be guaranteed to end-users and verticals. This leads to the need for automated experimentation and validation methods that use Monitoring and Analytics (M&A) techniques for identifying network bottlenecks and system configurations or malfunctions that hinder the compliance with SLAs. A reliable and efficient M&A framework should consider both end-users' and operators' perspectives, aiming at improving the Quality of Service (QoS) and users' Quality of Experience (QoE), while minimizing operators' management and operational costs. To utilize the monitored data, captured within such a framework, the Analytics service should support advanced data analysis methods, e.g., based on Machine Learning (ML) and Artificial Intelligence (AI), towards 5G system optimization [3, 4].

Within the above context and with such goals in mind, in this paper we design, implement and showcase a service-based Analytics module that can be easily embedded in 5G experimentation platforms and systems. In particular, the module is currently integrated in the open 5G experimentation facility implemented in the context of the H2020 *5GENESIS* project[1], which is formed by five 5G platforms with different underlying capabilities (e.g., standalone and non-standalone deployments, support for Internet of Things (IoT) and mission-critical services, satellite communications, etc.). The contributions of this paper can be summarized in terms of the following aspects:

- *General-purpose design*: We design an Analytics module with descriptive, diagnostic and predictive functionalities [3][4]. In particular, in the *5GENESIS* context, the component is being used for covering two broad use case areas, i.e., network-side system management and user-side performance analysis and prediction, with a focus on ensuring that 5G KPIs are properly measured and validated during

[1] `www.5genesis.eu`, Accessed on: November 2021.

the experimentation phases. The general-purpose design makes the proposed module different compared to most of the analytics frameworks proposed for networking systems, where the engines often focus on specific use cases, such as traffic analysis, prediction, and anomaly detection.

- *Modular implementation and flexible architecture*: We implement a component with a modular and flexible architecture. In order to provide different analytics methods (e.g., time series management, anomaly detection, correlation analysis, feature selection mechanisms, and prediction services, see Section 4), the module is based on *microservices*, designed as containerised modules, that make use of standard ML, visualisation, and reporting libraries, as well as well-defined interfaces. This development choice enables a straightforward reuse and extension of the component outside the *5GENESIS* facility, and makes it flexible, easily deployable, and scalable. The microservices can be instantiated on heterogeneous underlying infrastructures in a distributed fashion, with each container providing a specific service that can be used on its own or in combination with other services. New containers can also be easily developed and integrated in the module. Considering the ongoing standardization activities towards embedding ML/AI capabilities in 5G systems, the proposed module can be seen as a solution for some key analytics functionalities that are expected to be provided in the 5G Core (5GC), as defined by the 3rd Generation Partnership Project (3GPP), e.g., via Management Data Analytics Function (MDAF) and Network Data Analytics Function (NWDAF) (see [3, 4] and references therein).
- *5G-integrated and tested component*: We integrate and test the Analytics component in real 5G deployments. Hence, the module supports 5G experimentation and takes into account the peculiarities of the experiment definition and of the *5GENESIS* platforms. In this paper, we showcase integration and usage of the component by running dedicated experiments in the 5G infrastructure at the University of Malaga, i.e., one of the *5GENESIS* platforms.
- *Open source software and dataset*: Together with several other components developed in the context of the *5GENESIS* project, the Analytics module is open source and available as part of the *5GENESIS* Open Suite [5]. Moreover, the 5G dataset used in this paper for testing the module is also provided to the community, as a further contribution from *5GENESIS*.

Further details on the above contributions are provided in Section 2.3, following the analysis and discussion of related standardization and research activities in Sections 2.1 and 2.2. Altogether, component design, implementation, and testing allowed to derive the insights provided in this paper. These insights fall under several categories: (1) Considering the operational health of the system where Analytics is used, the services provided by the component enable a simplified troubleshooting of end-to-end deployments, by pinpointing nominal operations as well as possible malfunctions that need to be addressed; (2) In terms of the scenarios used to verify the Analytics services, the use of 3GPP-defined scenarios for typical network conditions is key for being able to extract insights on how well the experiments were running, e.g., by comparing expected vs. achieved performance; (3) Considering the deployment of the Analytics component, placement (e.g., in cloud or in edge) and configuration of its microservices play a key role for providing timely services, especially if data needs to be exchanged across microservices. This means that, while planning for the deployment mode of the component (e.g., all the microservices in the cloud or in the edge, or even a mixed placement), a user should be aware of data transfer rates between containers and time bounds on Analytics results. Indeed, we noticed that transferring large amounts of data between the containers might have a non-negligible impact on the timeliness of the results.

The rest of the paper is organized as follows: Section 2 provides a comprehensive overview of related work, including both standardization and research-related activities, and gives an in-depth explanation of the contribution of our work. Section 3 provides the background for this work in the context of the *5GENESIS* project, while Section 4 details the library of methods that we have implemented for the Analytics component, and explains how these methods relate to specific challenges in 5G experimentation. Sections 5 and 6 present our experimental setup in the Malaga platform, and the results obtained by using the Analytics component in the *5GENESIS* context. In Section 7 we discuss the experience gained by designing and deploying the Analytics component, and point out opportunities for future work. Finally, we provide examples of the Analytics Application Programming Interfaces (APIs) and corresponding responses in Appendix A, in order to provide a quick reference for readers and users.

2. Related Work and Contribution

As mentioned in Section 1, there is an increasing interest towards the application of data analytics, based on ML and AI, to the management and optimization of 5G (and beyond) systems [3, 4, 6, 7, 8, 9, 10, 11, 12]. The use of ML/AI allows to deal with the increased complexity of communication networks, and leads to a better management and exploitation of the large amount of data such networks generate. Data analytics can help minimizing the need for manual management and achieving higher system performance and user QoS/QoE, since it allows to derive and use optimized configurations and policies based on ML/AI models of system and user behaviours.

The trend towards ML/AI-based networks is reflected in several standardization and research activities, as summarized in the next two subsections. Our contribution in the context of these works is also explained in detail at the end of this section.

2.1. Standardization Activities

In Release 15, 3GPP has defined MDAF and NWDAF in the 5GC, aiming at providing data analytics to other Network Functions (NFs) [3, 4, 9, 10, 11]. MDAF provides services consumed by the Management and Orchestration (MANO) layer

of 5G systems. For example, MDAF can retrieve Operations, Administration, and Maintenance (OAM) data from different NFs and produce data-driven policies that can be used to recommend appropriate management actions to network operators. NWDAF, on the other hand, focuses on data plane services, and aims at providing data analytics for several analytics use cases, including the optimization of slice and Radio Access Network (RAN) configurations towards enhanced user experience. For example, the Policy Control Function (PCF) and Network Slice Selection Function (NSSF) in the 5GC can use the information provided by NWDAF on the load level of a slice instance, and derive an optimized tuning of slice components or the selection of a new slice that better suits service requirements [4][11].

During the activities that led to Release 16, 3GPP started a study item referred to as FS-eNA "Study of enablers for Network Automation for 5G", that analyzes how to enable network automation and extend NWDAF use cases including customized mobility management, load balancing, and network performance prediction, among others. On similar aspects, 3GPP also defined two further study items, i.e., "Study on Self-Organizing Networks (SONs) for 5G" and "RAN-Centric Data Collection and Utilization for Long Term Evolution (LTE) and New Radio (NR)", that mainly address data analytics for RAN optimization [4]. Such RAN-related activities nicely map onto the ongoing enhancements proposed by the OpenRAN Alliance (O-RAN)[2], toward establishing a common reference architecture for the implementation of next-generation RAN infrastructures, based on common hardware and shared definitions of components and interfaces. Among several functionalities, the O-RAN architecture includes specific components that allow the use of ML/AI for RAN optimization, e.g., real time and non real time RAN Intelligent Controllers (RICs) [11].

With regards to other standardization bodies, the European Telecommunications Standards Institute (ETSI) has created an industry specification group called Experiential Networked Intelligence (ENI)[3], that defines a ML/AI-based architecture to support operators in automating their systems based on user needs, environmental conditions, and business goals [10].

2.2. Research Activities

The research community is proposing several ML/AI mechanisms and showing the benefits of applying them to networking aspects. In the following, we analyze some reference literature, aiming at framing the state-of-the-art and discussing open challenges, in terms of a) which application methodologies, ML/AI techniques, and analytics use cases are under investigation, b) architectural and implementation aspects toward embedding ML/AI in 5G systems, and c) empirical analyses of 5G systems.

[2] `https://www.o-ran.org`, Accessed: November 2021.

[3] `https://www.etsi.org/technologies/experiential-networked-intelligence`, Accessed: November 2021.

2.2.1. Methodologies, techniques, and use cases

Comprehensive reviews on ML/AI methodologies, techniques, and use cases in communication networks are provided in [13][14]. The work in [13] defines a methodology for applying ML/AI to networks, in terms of a complete workflow where the main steps are data collection, feature engineering, establishment of ground truth, definition of performance metrics, and model validation. It also groups analytics use cases in traffic prediction, classification, and routing, management of resources, faults, and QoS/QoE, congestion control, and security. A similar workflow is proposed in [12], with steps limited to feature extraction, data modeling, and prediction (with online refinement), while similar analytics use cases are instead identified in [3], which also includes ML/AI-based network planning, load balancing, and beamforming, among others. The work in [14] gives a detailed description on ML/AI techniques applicable to networking scenarios (supervised, unsupervised, and reinforcement learning), and differentiates analytics use cases across system layers, i.e., from physical to application layers. A similar approach is also taken by [9], which however focuses on physical, Medium Access Control (MAC), and network layers. The highlighted analytics use cases include channel estimation, network planning, user association and scheduling, etc. Further discussions on ML/AI techniques and analytics use cases in 5G and beyond networks are provided in [6] [10].

2.2.2. Architectural and implementation aspects

Several investigations discuss architectural and implementation aspects toward embedding ML/AI in 5G systems.

Considering architectural aspects, several works focus on the 5G architecture, and specifically on MDAF and NWDAF. The work in [15] studies the application of regression and classification methods to several expected NWDAF functionalities, i.e., analysis of abnormal vs. expected behaviour for a group of User Equipments (UEs), and network load prediction. Test and validation are performed on a simulated 5G dataset, that includes a topology with a fixed number of cells and subscribers with different traffic patterns. Each cell is modeled using a set of features retrieved from other NFs, including transmitted bytes, subscriber categories, and subscriber identifiers. Anomalies in terms of unexpected data traffic patterns are included in order to make the dataset more realistic. The work in [16] analyzes how NWDAF can support the 5G Access Traffic Steering, Switching and Splitting architecture (ATSSS), defined in 3GPP Release 16 and providing mechanisms for selection and aggregation of 3GPP and non-3GPP access networks (e.g., 5G NR and WiFi), toward load balancing and QoS/QoE improvement at the user end [17].

The necessity for a RAN-dedicated analytics engine, referred to as RAN-DAF, is stressed out in [4]. In the proposed framework, RAN-DAF is decoupled from MDAF and NWDAF but can still communicate with them and with other analytics engines, referred to as Application Function (AF)-DAF and Data Network (DN)-DAF, which are possibly provided by 3rd parties and tailored on specific applications and services. Such analytics engines are deployed in a distributed fashion and exchange raw/processed data via inter-domain message buses,

while having common performance improvement goals. RAN-DAF is devoted to near real time radio resource management, as showcased in the proposed case study, where it takes advantage of UE traffic and mobility prediction performed by a dedicated AF-DAF to provide enhanced resource allocation schemes, which are tested in a Fourth Generation (4G) simulated environment. The need for RAN-DAF is also advocated in [11], where UE prototype trajectories and radio resource utilization patterns are used for showcasing enhanced handover and admission control mechanisms. Both analytics use cases take as a reference a dataset collected in 31 cells of an LTE network in an urban area of a European city, covering a period of two weeks with measurements taken every 15 minutes. In both works, the need for RAN-DAF is justified by considering that a) RAN and 5GC may need similar analytics but at different granularity (e.g., UE position with higher (RAN) vs. lower (5GC) resolution), b) RAN-DAF implementation improves scalability since faster decisions can be executed locally (e.g., in RAN co-located edge units), with no need for moving data towards cloud-based core networks, and c) business aspects. In the O-RAN context, recent examples of integration and usage of analytics engines can be found in [18] and [19].

The above works marginally discuss implementation aspects, i.e., how to practically implement a ML/AI engine to be easily embedded in 5G, which is instead the focus of the following investigations.

Data management, storage, and processing engines under the Apache Software Foundation[4] have been often proposed as starting implementation points. For example, [20] and [21] propose the use of Apache Hadoop[5] for storage and analysis of traffic data collected at either Internet Exchange Points (IXPs) [20] or inside cellular networks [21]. In both cases, data management and analytics functionalities are based on MapReduce [22]. Efficient data partitioning and querying are proposed and tested in [20], while traffic analytics functionalities (e.g., derivation of traffic statistics, and analyses on application-layer, web service provider, and user behaviour) are provided and tested in [21], which uses on a dataset from a Second Generation (2G)/Third Generation (3G) network.

Beyond traffic analysis and targeting service quality management in wireless networks, the work in [23] introduces an analytics module called Deep Network Analyzer (DNA) that is based on Apache Spark.[6] DNA performs anomaly detection and root cause analysis for the detected anomalies, and it is tested on two datasets from network operators, containing various undisclosed parameters referred to as KPIs and Key Quality Indicators (KQIs), with the latter representing the target features. Apache Spark is also used in [24] for designing CellScope, an analytics service that tries to mitigate the trade-off between latency in collecting data and accuracy of ML models trained on such data. CellScope primarily targets the optimization of RAN management; in order to achieve this goal, it employs a) intelligent grouping of data from different base stations composing the RAN and b) multi-task learning with hybrid offline/online model training/update. CellScope performance are tested by leveraging data collected on a large LTE operational deployment, for two analytics use cases, i.e., classification of connection drops (via Random Forest) and throughput prediction (via Lasso regression).

Finally, the Net2Vec solution proposed in [25] embeds ML and Deep Learning (DL) capabilities exploiting Python libraries, including SciKit-Learn and TensorFlow. Net2Vec is shown capable of handling a large amount of data (generated synthetically) for implementing and using a system that creates user profiles in a timely manner.

2.2.3. 5G empirical analyses

Due to initial deployments of 5G systems being only recently available, in terms of both dedicated testbeds and operational networks, current literature mostly uses simulations and/or older generation data (e.g., LTE datasets) for testing the proposed ML/AI schemes.

Initial performance measurements and statistical assessment on 5G operational networks have been recently presented in [26] and [27]. On the one hand, [26] analyzes a sub-6 GHz Non-Standalone (NSA) deployment in a dense urban environment in China, in terms of coverage, handover, UE energy consumption, and end-to-end throughput, latency, and application performance. On the other hand, [27] analyzes throughput, latency, and application performance over Millimeter Wave (mmWave) deployments of three US operators.

Aiming at reliable validation of 5G KPIs, our previous work in [28] provides the definition of testing procedures for reliable performance assessment in dedicated testbeds, and provides initial insights on the achievable throughput and latency performance on a 5G NSA deployment at 3.5 GHz.

One of the first ML-based analyses of 5G performance is given in [29] where, following the line drawn by similar investigations on 4G data (e.g., [30][31]), several ML schemes are used jointly with parameters collected at the UE side in order to assess the predictability of the throughput achievable in a mmWave urban deployment.

2.3. Contribution of our Analytics component

With regards to the application methodologies, we have generalized the analysis steps in [13] and based our work on a more flexible workflow that includes data collection and storage, data engineering and pre-processing, and data analysis as main steps. With this workflow in mind, along with the goals of the *5GENESIS* project, we have implemented a M&A framework, and integrated it on top of the *5GENESIS* Reference Architecture (see Section 3.1). As described and showcased in this paper, the Analytics component covers the last two workflow steps (data engineering and analysis), and has been designed and implemented in order to be easily deployed and used within the five 5G experimentation platforms that are part of the *5GENESIS* project. However, due to its flexible design and implementation, the component can be easily deployed and used in 5G platforms and networks outside the *5GENESIS* project, and thus we provide it open source to the community in [5].

[4] `https://www.apache.org`, Accessed: November 2021.
[5] `http://hadoop.apache.org/`, Accessed: November 2021.
[6] `https://spark.apache.org`, Accessed: November 2021.

Considering analytics use cases and ML/AI techniques, the current implementation of Analytics covers two broad use case areas, i.e., network-side system management and user-side performance analysis and prediction. This differs from most of previous work on the use of analytics frameworks in networks, where the proposed engines often focus on very specific analytics use cases (e.g., traffic analysis, prediction, and anomaly detection). Following the taxonomy in [3][4], our Analytics module provides descriptive, diagnostic, and predictive functionalities. Indeed, the component is being used in *5GENESIS* for network planning, (re-)configuration, troubleshooting, and user QoS/QoE analysis and prediction. This is achieved through a predominant use of statistical and supervised ML techniques, as described in Section 4. Moreover, the component can be easily extended for covering further analytics use cases. For example, within *5GENESIS*, decision-making and prescriptive analytics functionalities could be embedded through establishing coordinated operations between Analytics, Slice Manager, and policy engines at UE and network sides, i.e., NEAT [32] and APEX [33].

Considering architectural and implementation aspects, we provide a general-purpose Analytics component, that can be transparently used and integrated as part of MDAF, NWDAF, and even RAN-DAF, at least for some analytics use cases. Due to its descriptive, diagnostic, and predictive nature, the component specifically covers offline data analysis (e.g., it can be used for training a model of a KPI by using historical data); trained models can then be exposed and used by decision-making engines, e.g., non real time RICs in an O-RAN architecture.

As detailed in Section 4, Analytics functionalities are easily accessed by other NFs and components via REST APIs, the set of which can be extended toward providing further services for more specific analytics use cases. Moreover, such functionalities are decoupled from the underlying physical/virtual infrastructure, since they are provided as interconnected microservices, formed by Python-based data analysis code encapsulated in Docker[7] containers. This adds a further level of flexibility, since the microservices can be instantiated in a distributed fashion across the underlying infrastructure, and can be flexibly initiated, reconfigured, moved, and terminated on demand. Hence, in contrast to several previous works, we open source a component that can be easily integrated, used, and extended in heterogeneous physical/virtual 5G infrastructures, thanks to its flexible ecosystem formed by REST APIs and microservices.

Finally, considering empirical analyses of 5G systems, in this paper we extend [28] and validate the Analytics component by performing dedicated experiments in the real 5G NSA experimentation testbed at the University of Malaga. By doing so, we prove that the component can be used for better understanding of the correlations and causalities between system deployment, configurations, and user performance, which is in turn beneficial for optimized system and performance management.

[7] `https://www.docker.com/`, Accessed: November 2021.

3. *5GENESIS* Background

The Analytics component targeted in this paper has been developed within the context of the *5GENESIS* project whose goal is to bring together five 5G experimentation platforms from different European countries into one 5G experimentation facility, and allow experimenters to create and run experiments, as well as access data about KPIs related to the experiments. The five platforms of the *5GENESIS* Facility are: Athens platform, Malaga platform, Limassol platform, Surrey platform, and Berlin platform. For the experiments presented in this paper we use the Malaga platform, but the Analytics component is used within all *5GENESIS* platforms and constitutes an integral part of the *5GENESIS* experimentation facility. To provide the context for the Analytics component and our results, we introduce hereafter the *5GENESIS* reference architecture and then we provide a more detailed description of the *5GENESIS* M&A framework, in which the Analytics component resides.

3.1. 5GENESIS Reference Architecture

Figure 1 shows the reference architecture for the *5GENESIS* platforms [34]. The Analytics component is part of the upper layer, called the Coordination layer. The Coordination layer [35] and the Slice manager are cross-platform components that are instantiated across all the five *5GENESIS* platforms, formulating the Open *5GENESIS* Suite, i.e., an open-source framework for automated experimentation that *5GENESIS* has released. The lower layers are instead specific to the characteristics of each platform and therefore they can differ among the *5GENESIS* platforms.

The design and implementation of the Coordination layer is guided by the implementation of reference test cases devoted to the testing of KPIs in 5G networks and also for applications. A key part of these test cases is the definition of the testing scenarios, the measurement collection, and the execution of the tests. The Coordination layer allows to specify a common and sustainable interface that any test platform can instantiate in order to be able to apply the *5GENESIS* experimentation methodology [28]. The 5GENESIS experimentation methodology follows a modular approach to specify the input data and the network configuration required for executing an experiment, the workflow for running such an experiment, and the output results to collect for further analysis. The methodology includes the templates for the specification of the experiments as well as the specification and implementation of the components that orchestrate the execution of the experiments defined in such way. Such components include the Coordination layer and the Slice manager, as introduced previously. By adopting this methodology, different 5G platforms can perform automated experiments that yield comparable results which, in turn, allows the benchmarking of different technologies, applications and services. One key asset in the Coordination layer is the Experiment Life Cycle Manager (ELCM). The ELCM is the entity that performs the management, orchestration and execution of the experiments in the Open *5GENESIS* Suite, including three main components, namely the Scheduler, the Composer

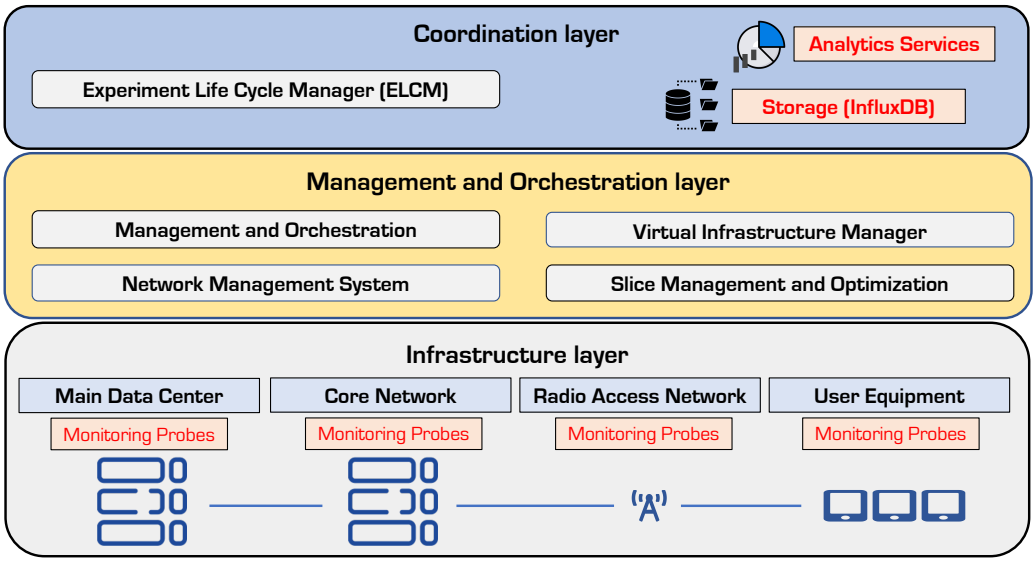

Figure 1: The *5GENESIS* reference architecture. The blocks written in orange represent the components part of the M&A framework.

and the Execution Engine (responsible for experiment automation), as well as several auxiliary elements. The Scheduler is responsible for managing the execution of the experiments on a higher level, by keeping track of the execution of the experiment stages (i.e., Pre-Run, Run, and Post-Run) for multiple experiments. The Execution Engine includes the logic for managing the execution of each experiment stage, by generating an independent Executor. The Composer is responsible for creating platform-specific configurations for the experiments to be executed. The Coordination layer also includes the Analytics module, whose architecture is explained in detail in Sections 3.2 and 4 and is the main focus of this paper.

The MANO layer is mainly responsible for the management of physical and virtual elements, including network slices. This layer contains the Network Function Virtualization (NFV) MANO, the Virtual Infrastructure Manager (VIM), and the Network Management System (NMS), responsible for the management of virtualized and physical resources, respectively. These components include monitoring probes for their specific domains, whose data is used also by the Analytics module.

The bottom layer is the Infrastructure layer, whose components essentially aim to constitute the end-to-end data path for the user traffic. We also have here the various probes used for monitoring the different elements in the network and the performance in the end-user equipment. As before, this data will be used by the Analytics module for the various analyses needed for 5G KPIs assessment and for performance optimization.

3.2. 5GENESIS Monitoring and Analytics Framework

As introduced above, the Analytics component is part of the *5GENESIS* M&A framework that is in turn integrated with the *5GENESIS* reference architecture. Before providing a detailed description of the Analytics component (Section 4), this subsection describes on a high-level the full-chain M&A framework.

The main goal of Analytics is to analyze heterogeneous data, that is, parameters related to the infrastructure status and to measured end-to-end KPIs, in order to a) verify the status of the *5GENESIS* facility during the experiments, b) allow for reliable assessment of 5G KPIs, and c) pinpoint possible issues causing performance losses that require new management policies and network configurations. Hence, from a functional perspective, the Analytics component is tightly coupled with the Monitoring system.

The Monitoring system is a distributed component that lies in the MANO and Infrastructure layers of the *5GENESIS* architecture. It includes two functional blocks, referred to as Infrastructure Monitoring (IM) and Performance Monitoring (PM). On the one hand, IM collects data exposed by the components and probes available at MANO and Infrastructure layers, such as, radio access and core networks, cloud/edge resources in the slices, and UE. On the other hand, PM focuses on measuring QoS/QoE KPIs, mainly at the UE side. Therefore, a large set of IM/PM probes is being developed, integrated, and used in the *5GENESIS* facility, including:

- *Network-side IM probes*: These probes retrieve metrics mainly through the exporters in the Prometheus software.[8] In particular, through the Node exporter[9], physical/virtual units forming MANO and Infrastructure layers can expose metrics on their memory load and consumption, among others. For example, as detailed in [36], the Slice manager uses Prometheus exporters to provide info on the status of the slice(s) instantiated during an experiment. Other exporters, based on Simple Network Management Protocol (SNMP), monitor further components, e.g., core and radio networks of specific technology providers;

- *UE-side IM probes*: These probes monitor physical layer parameters and configurations of the UEs during exper-

[8] `https://prometheus.io`, Accessed: November 2021.
[9] `https://github.com/prometheus/node_exporter`, Accessed: November 2021.

iment executions, e.g., coverage-related parameters, including Reference Signal Received Power (RSRP) and Signal to Interference plus Noise Ratio (SINR), which are key for better understanding the experienced QoS/QoE KPIs;

- *UE-side PM probes*: These probes measure 5G QoS/QoE KPIs, e.g., latency, throughput, and vertical-specific KPIs. The set of PM probes includes custom made Android apps that runs on Android phones as well as the MONROE Virtual Node (VN), that is, a Virtual Machine (VM) developed by reshaping MONROE physical nodes[10] and enabling a range of containerized experimentation on any general purpose Linux-based machine with a 4G/5G radio interface [37][38].

The Analytics component is hosted by the Coordination layer, along with the data Storage utility. The Storage is the infrastructure point where a) the Monitoring system redirects the data collected during experiment executions, and b) Analytics redirects its queries for retrieving specific data portions needed to run its analyses.

The *5GENESIS* facility uses InfluxDB for the creation of platform-specific instances of a long-term storage utility. InfluxDB is an open-source engine part of the InfluxData framework[11], and handles non-relational time series data. The ELCM is in charge of redirecting data collected by the different probes and the logging data available at the different components of the infrastructure to the InfluxDB instance, once the execution of an experiment in one of the *5GENESIS* platforms is terminated. It is also in charge of adding further metadata, e.g., the Experiment (Execution) ID, that might be useful for some of the Analytics functionalities. The integration of new probes and components is done by the development of plugins that will enable the communication of the new probe/component with the ELCM. The only requirement is that the probe/component provides a command line interface for its control and management and the retrieval of the measurements. The guidelines for integrating new probes and components are available at [39]. It is worth mentioning that InfluxDB also provides a lightweight integration with Prometheus and Grafana[12], this latter being used in the 5GENESIS web portal for visualizing raw collected data (in parallel to the Analytics visualization service that provides visual representation of the results of the executed analyses).

The connection between Analytics and the InfluxDB database is performed through REST calls to the InfluxDB server. Note that a pre-existing InfluxDB-Python client could have been used[13], but we noticed that direct HTTP queries enable faster connection and data retrieval, and thus we adopted this second approach. Once retrieved from the database, data is converted to Pandas[14] dataframes, goes through some pre-processing, if needed, and is finally provided as input to the micro-services in the Analytics component (cf. Section 4).

4. Service-based Analytics

In order to tackle the health and performance analysis of the network and experimentation infrastructure, we implemented a variety of analytics methods as microservices using Docker containers (see Figure 2). These include, among other things, anomaly detection and correlation services for health analysis purposes (e.g., is the experiment doing what it is supposed to do?), and predictive services for performance analysis purposes (e.g., are the network elements achieving the expected performance?). Our choice of algorithms and software was guided by the 5G experimentation requirements in the *5GENESIS* project, which we can summarize as follows:

- *Domain relevancy*: The Analytics component has to specifically focus on the analysis of mobile network experiments and be aware of the characteristics of these experiments. Hence, general-purpose solutions are not fit for use;

- *Data privacy*: Data collected during experiments in the *5GENESIS* facility has to remain on the local platform servers. Hence, cloud-hosted solutions are not fit for use in our case;

- *Accessibility and costs*: Analytics services has to be open source. Hence, proprietary solutions are not fit for this goal;

- *Usability*: It should not be mandatory to use complex queries or to know the database format for using the component. Hence, solutions not providing an easy-to-use Graphical User Interface (GUI) are not fit for use in our case;

- *Scalability*: Analytics services have to be lightweight and easy to instantiate/terminate on demand. Hence, containerised solutions should be preferred.

We further observe that a substantial collection of analytics solutions exist[15], but none of them fit all the aforementioned *5GENESIS* requirements. For example, tools such as Acumos[16] and Kubeflow[17] are powerful frameworks for the optimization and lifecycle management of ML/AI models. However, Acumos is a general solution that facilitates the manipulation and optimization of ML/AI models rather than providing a ready-to-use analytics framework. Similarly, Kubeflow is designed to make easier the deployment of ML models in Kubernetes. We could use Acumos and Kubeflow toolkits to support the creation

[10] `https://www.monroe-project.eu`, Accessed: November 2021.
[11] `https://www.influxdata.com`, Accessed: November 2021.
[12] `https://grafana.com`, Accessed: November 2021.
[13] `https://github.com/influxdata/influxdb-python`, Accessed: November 2021.
[14] `https://pandas.pydata.org`, Accessed: November 2021.
[15] `https://github.com/0xnr/awesome-analytics`, Accessed: November 2021.
[16] `https://www.acumos.org/`, Accessed: November 2021.
[17] `https://www.kubeflow.org/`, Accessed: November 2021.

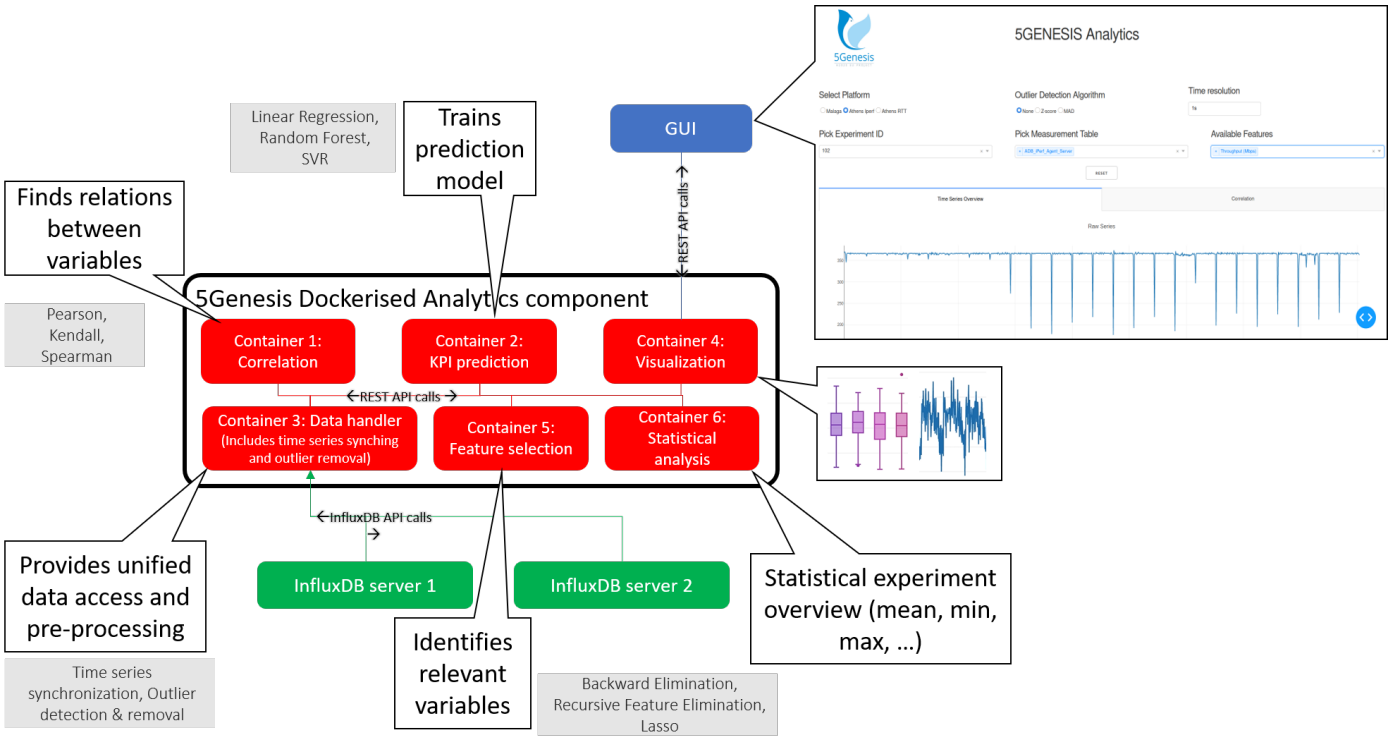

Figure 2: Overview of the containerised analytics microservices and their functions, including data handling, correlation, KPI prediction, statistical analysis, feature selection and visualisation. A graphical user interface (GUI) provides easy access to the different services.

and deployment of (parts) of the *5GENESIS* Analytics component. However, we believe that our standalone component results in a more compact, practical, and lightweight tool, since it specifically serves the analysis of KPIs in the 5G domain, and does not aim to be a general toolkit to be used in other contexts. The *5GENESIS* Analytics component was designed both for enabling the efficient setup and configuration of 5G platforms, and for ensuring that such platforms provide useful insights on the experimentation and analysis of end-user KPIs.

Our developed Analytics component is thus tailored on the *5GENESIS* requirements, but it is flexible enough to be applied and extended to other 5G experimentation platforms and use cases. Additional analytics microservices can be defined, based on specific additional needs to suit other goals for new 5G platforms. To ensure that the additional containers are properly built and deployed, their details and credentials must be included in the `analytics-stack.yaml` file and in the `install` script. A comprehensive `README` file is included in the online repository at `https://github.com/5genesis/Analytics`, explaining all the set up procedures for containers, which can be followed for new containers being added. If visualisation for the new containers is also required, then the visualisation microservice must be also modified to reflect this need. In the following sections, we describe the algorithms and services we implemented as part of our Analytics component.

4.1. Visualisation Service

The visualisation service allows for visual representation of the results from the other services. It provides a GUI that runs in the browser and lets the user see the output of the other containers.

The goal is to provide access to the results of the Analytics services as interactive visual results. After an experiment is run and its data is stored in the InfluxDB server, the user can browse and analyse the experiment results through the GUI shown in Figure 2 (top right). In the *5GENESIS* context, the user can select which experiment to analyze by providing an experiment ID in the dedicated field on the left hand side of the GUI. The main part of the GUI service is a tabbed environment, where the user can select one of the other Analytics services. In the current service release, the tabs contain (1) time series overview, (2) statistical analysis, (3) KPI correlation, (4) feature selection, and (5) KPI prediction.

4.2. Data Handler Service

The data handler service provides a unified access to data that is stored in an InfluxDB instance (e.g., the database of a specific *5GENESIS* platform). It also provides data pre-processing functionalities, including time series synchronisation, as well as outlier detection and removal. The following subsections describe these services in details.

4.2.1. Time Series Synchronisation

To run advanced analyses on the data collected during the execution of experiments, we need to merge the data coming from different monitoring probes activated during the execution. In *5GENESIS*, as in most distributed systems, the measurements may be recorded at slightly different times, often with a few

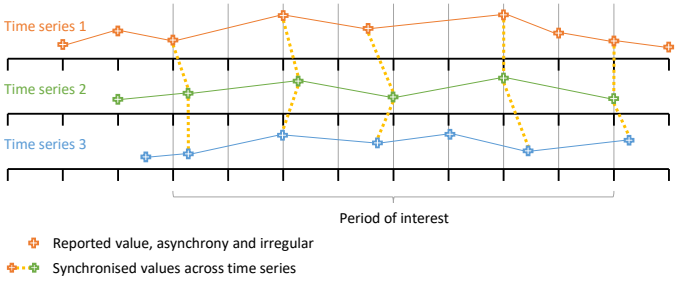

Figure 3: Time series synchronisation: Individual measurement points can be aligned by synchronisation (dotted lines).

milliseconds difference, or with a different granularity, as exemplified in Fig. 3. As an example in the scope of the *5GENESIS* M&A framework, IM and PM probes (see Section 3.2) may be nearly but not perfectly synchronised, since they are activated by the ELCM in consecutive steps. Moreover, they may collect data with different sampling periods. For example, targeting data collection scalability, IM probes collecting the CPU consumption of computing units may work at reduced rates compared to PM probes, which may instead require a more granular data collection in order to better monitor QoS/QoE KPI, e.g., user-experienced throughput.

Synchronisation is a way to align data points that are close in time, instead of using the exact time when they were recorded. In Figure 3, synchronisation is depicted via yellow dotted lines connecting data points from different monitored parameters, i.e., *measurements* from different monitoring probes, identified by red, green, and black time series. The synchronisation approach is feasible when, as in the *5GENESIS* context, the time difference between data points across measurements is usually much smaller (often in the order of milliseconds) than the time difference to the neighbouring data points in the same measurement (often in the order of seconds), which allows for accurate matching of data points across measurements. Another advantage of this method is that it uses the real values that are reported from the measurement probes and devices (as opposed to inferred values). In the current implementation of this service, a time granularity can be specified, on which the data points from all measurements will be synchronised. For example, if the granularity of one second is specified, the timestamp of each data point will be truncated to the full second. If there are multiple data points per second, the average is calculated. As a further note, we observe that the measurement rate depends on the probes. Moreover, the reporting period can be configurable or not. When the reporting interval can be adjusted, a reasonable value for it should reflect the purpose of the test and the radio propagation conditions during such a test. In this paper, the tests used for showcasing Analytics functionalities focus on measuring the achievable end-to-end user throughput at the application level (cf. Section 5.2). A reporting period of 1 second has been then configured because we did not observe high fluctuations in the preliminary phases used to setup the experiments. Note that if the focus of the test was, for example, to evaluate the performance of the MAC scheduler we would need a reporting period in the order of milliseconds and thus probes able to provide such measurement granularity.

4.2.2. Anomaly Detection

Another challenge that we must address before diving into further analyses is the occurrence of anomalies. Anomalies are values that were recorded but do not lie in the typical range of a variable (e.g., a 32-digit figure where the normal range is in the 3-digit area). A recorded number that is very far off from the expected values suggests an error in the measurement or reporting, which is relevant for the health monitoring of the experiment components, such as measurement probes. Two anomaly detection methods are currently integrated in the Analytics component: Z-Score and Median Absolute Deviation (MAD).

Z-Score uses the standard deviation of the given distribution of recorded data points in a measurement to determine whether a data point does not belong to that distribution. The Z-Score for a variable x_i is calculated as follows:

$$z_i = \frac{x_i - \bar{x}}{S}, \tag{1}$$

where $\bar{x}$ is the sample mean and S the sample standard deviation. Variables with a z_i value greater than 3 (at least three standard deviations away from the mean) are considered outliers and are thus removed before proceeding to other analyses.

MAD, also known as Robust Z-Score, uses the so-called MAD score to detect outliers. The score is defined, for each population sample x_i, as follows:

$$M_i = \frac{0.6745(x_i - \tilde{\boldsymbol{x}})}{\mathsf{MAD}}, \tag{2}$$

where $\mathsf{MAD} = \text{median}\{|x_i - \tilde{\boldsymbol{x}}|\}$, $\tilde{\boldsymbol{x}}$ represents the median of the sampled population, and the value of 0.6745 is derived under the assumption of normally distributed data, being the 75% percentile of the standard normal distribution. The samples for which $|M_i| > 3.5$ are considered outliers [40].

In our Analytics component, both methods are implemented using Pandas and NumPy vectorisation methods with a focus on performance and near real time application, and present comparable computational costs.

4.3. Statistical Analysis Service

This service provides a statistical overview of the data collected during the execution of an experiment. In particular, this service executes the KPI validation process defined in *5GENESIS* [41]. As detailed in [28], an experiment dedicated to the validation of a 5G KPI under specific network configurations and conditions, is executed in *5GENESIS* platforms as a repetition of a statistically significant number of consecutive trials (i.e., iterations). Therefore, the KPI validation process includes:

- A preliminary analysis of each iteration separately, where several indicators are evaluated and collected (e.g., min, max, median, and average values of the KPI under analysis observed during each iteration);

- The derivation of the same indicators at the experiment level, obtained by averaging the indicators of each iteration and adding a confidence interval that assesses the precision of the provided outcome.

In the Analytics component, this service relies on standard Python libraries (Pandas and NumPy) that allow for a straightforward evaluation of a full set of statistical indicators (e.g., min, max, mean, standard deviation, etc.)

4.4. Linear Correlation Analysis Service

This service provides state-of-the-art correlation algorithms, including Pearson, Kendall, and Spearman, to find linear relationships between variables collected during experiments.

The experimenter may be interested in observing the temporal behaviour of recorded variables, and evaluate the similarity between them. Therefore, in the *5GENESIS* and in other 5G experimentation frameworks, linear correlation is a fast and computationally efficient way to gain insights in terms of possible similarities between monitored parameters and KPIs. In the current implementation, the experimenter can specify the type of correlation they want to perform, choosing from the above mentioned correlation methods. The algorithms are provided via the Pandas library.

In particular, the correlation service currently provides two types of use cases:

- *Correlation between variables in the same experiment*: This allows the experimenter to compare parameters and KPIs collected during an experiment. For example, the experimenter can investigate the correlation between UE throughput and power consumption during an experiment executing in a urban environment under mobility;

- *Correlation between variables across different experiments*: This enables the evaluation of correlation between same variables collected during different experiments. For example, the experimenter may be interested in comparing the difference in trends for throughput collected during two experiments executed under the same network conditions but different configurations, in order to verify how the configuration changes impact the throughput experienced at the user end.

4.5. Feature Selection Service

The feature selection service uses algorithms, such as Backward Elimination (BE), Recursive Feature Elimination (RFE), and Least Absolute Shrinkage and Selection Operator (LASSO) to identify the most relevant variables with respect to target variable, e.g., a KPI.

Feature selection is an important step in data analysis. Given a dataset related to a particular experiment, feature selection can be primarily applied in quest of dimensionality reduction, e.g., to discard parameters collected by the monitoring probes that are not directly *correlated* with the KPI under analysis. The presence of these parameters may in fact hinder following analyses, e.g., prediction, as they may negatively impact the fitted model. In its current release, we implemented a service that focuses on numeric feature selection, that is, all categorical variables possibly collected during an experiment are not considered. Algorithms for numeric feature selection are traditionally divided in three main categories, that is, *filter*, *wrapper*, and *embedded* methods. Hence, we implemented one filter method, which basically reuses the linear correlation service described in the previous subsection, two wrapper methods, namely, BE and RFE, and one embedded method based on LASSO-regularized regression [42][43].

Being part of the wrapper category, BE adopts a ML algorithm to fit a specific model using the available features. It starts with the entire set, and iteratively removes features based on the model accuracy. More precisely, the same model is built at each iteration using the remaining features, and the p-value for each of them is evaluated. In our implementation, the features resulting in a p-value larger than 0.05 are removed. In its current implementation, an Ordinary Least Squares (OLS) model is adopted, since it is largely used to perform linear regression.

Similar to BE, RFE works by recursively removing features while building a model. It initially fits a model, e.g., linear regression, based on all features. Then, at each iteration, it evaluates feature coefficients and importance, ranks them on the basis of the linear regression accuracy, and finally removes low ranking features.

Across embedded methods, the ones based on regularization are quite popular. LASSO regularization deals with possible linear regression overfitting by penalizing unimportant features, assigning them coefficients up to zero. In this case, a regularization parameter, denoted α in our service, is needed in order to control the strength of shrinkage and in turn the selection of the relevant features.

Users can thus select one across the above methods and run the selection task. Multiple service queries with different algorithms make also possible to compare the results, i.e., the set of selected features obtained when different algorithms are used. The feature selection service in our Analytics component build on top of Pandas, NumPy, and SciKit-Learn libraries.

4.6. Prediction Service

The prediction service provides state-of-the-art ML algorithms to train prediction models on a given target KPI. The trained models can then be used for live predictions.

With the aid of KPI prediction, the experimenter can attempt to identify how the various network elements impact the targeted measurement KPIs, in order to understand how the network can be optimised to achieve a desired increase / decrease of the KPI. A typical application is to predict the resource requirements that are needed to achieve a desired throughput.

In the current release of the Analytics module, we implemented a number of algorithms to this end: Linear Regression, Random Forest and Support Vector Machines (SVM)-based regression algorithms, utilising freely available SciKit-Learn modules. Our selection was driven by a focus on faster trainable models that are served to the visualisation service in near real-time (linear regression) or with an acceptable waiting time

(random forest and SVM), for small and medium-sized data. A wider selection of prediction algorithms may be included in future releases, possibly including deep learning algorithms.

4.7. API Description

The Analytics dashboard provided by the visualisation service (described in Section 4.1) is the primary way of accessing the various Analytics services. The dashboard can be accessed through a web browser at `visualisation-URL/dash`.

All other services can be consumed programmatically through their respective REST APIs, with each request returning a JavaScript Object Notation (JSON) response. For the full API description for each service, we refer the reader to the *5GENESIS* open source code repository [5]. The `README` file includes descriptions of the APIs for each service and also gives examples of using the APIs and the corresponding responses. In this section we explain some of the common REST API calls for our Analytics services, and we also include full examples of these API usage and responses in Appendix A.

Data Handler API: The most common call to the data handler service is to retrieve (and preprocess) data. This will return a collection of KPIs and their values (as exemplified in Appendix A.0.1), where each data point is UNIX-timestamped. The data handler service also provides other endpoints, for example listing all available datasources, all available experiments, available experiments for a given measurement and available measurements for a given experiment.

Statistical Analysis API: The call to the statistical analysis endpoint returns test statistics about the experiment data. Experiment id, KPI and measurement parameters must be specified, as shown in Appendix A.0.2. The result returns statistics metrics such as percentiles, min, max and mean for each individual experiment iteration (0, 1, 2, ...) as well as for the whole experiment run, averaged over all iterations.

Linear Correlation Analysis API: The most common call to the linear correlation service is to request the correlation matrix for all fields/KPIs collected during an experiment. An optional parameter `remove_outliers` can also be specified, to exclude outliers from the data used for the correlation. The exact call and results are exemplified in Appendix A.0.3. The result shows the pairwise correlation values for all fields. The correlation service also offers another endpoint (`/correlate/experiments`) that allows the user to retrieve the correlation values for KPIs across different experiments.

Feature Selection API: The feature selection service offers one endpoint (`/selection`), which requires the experiment identifier and measurement parameters, as shown in Appendix A.0.4. The result contains three parts. "Features - Original" contains the full unfiltered list of features in the data, "Features - Selected" contains the subset of features that were chosen by the feature selection algorithm, and "Score" contains the scores that the algorithm assigned to each feature for more information about the selected features.

Prediction API: The prediction service offers an endpoint to train an ML model, where the desired algorithm and target KPI are specified in the path. At least one experiment id parameter must be specified as well, and we may also ask the prediction service to remove outliers by specifying the outlier removal method, as shown in Appendix A.0.5. The result shows the coefficient of the trained model (feature importance), the real and predicted values of the test set, and the results of the prediction on the test set, including R^2 and Mean Squared Error (MSE) (exemplified in Appendix A.0.5). A second endpoint (`/model`) is also available that allows to download the last trained ML model for future use.

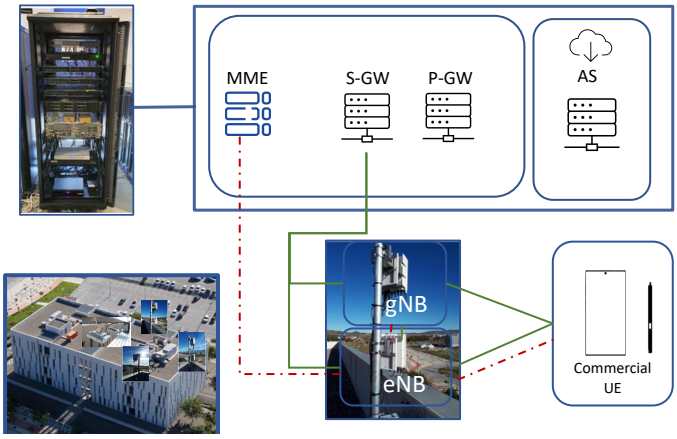

Figure 4: The 4G/5G experimental testbed at the University of Malaga.

3GPP technology	4G LTE+, 5G NSA
Core Network	Polaris Rel-15 EPC
Core Cloud	Openstack
Slice Manager	Katana
MANO	OSM v6
Automation Platform	OpenTAP
Infrastructure Monitoring	Prometheus

Table 1: Main infrastructure elements of the *5GENESIS* Malaga platform.

5. Experimental setup

In this section, we describe the *5GENESIS* Malaga platform and its configuration, the conducted experiments and the corresponding collected dataset.

5.1. Setup and Configuration

The testbed used in our experiments is a 5G NSA private network deployed at the University of Malaga. This private network, depicted in Figure 4, includes a RAN composed of four 5G gNBs and four 4G eNBs, a 3GPP Rel-15 Evolved Packet Core (EPC) as network core, and a main data centre. The setup also includes probes for monitoring radio parameters, IP traffic, and resource usage in the main data centre and the UE. More details about the infrastructure and the software components are provided in Table 1.

On top of the infrastructure shown in Figure 4, the Open *5GENESIS* Suite [5] has been deployed. This experimentation framework has been used for the execution of the experiments whose results are analysed in this paper.

Band	n78
Tx Mode	Time Division Duplex
Bandwidth	40 MHz
Carrier Components	1 carrier
MIMO layers	2 layers
Downlink MIMO Mode	2x2 TM3
Modulation	Adaptive (up to 256-QAM)
Beams	Single beam
Subcarrier Spacing	30 kHz
Uplink/Downlink Slot Ratio	2/8
Scheduler	Proactive

Table 2: 5G NR NSA configurations adopted during the experiments presented in this paper.

Band	B7
Tx Mode	Frequency Division Duplex
Bandwidth	20 MHz
Carrier Components	1 carrier
MIMO Layers	4 layers
Downlink MIMO Mode	4x4 TM4
Modulation	Adaptive (up to 256-QAM)

Table 3: 4G network configurations adopted during the experiments presented in this paper.

Regarding the radio configuration in the infrastructure, the data plane has been configured to use only the 5G data plane (data bearers are handled by gNB nodes), while 4G is acting as the anchor point for the control plane. Taking into account this configuration, only the 5G configuration parameters indicated in Table 2 should be taken into account to calculate the available bandwidth, according to [44]. The commercial UE used during the testing has been Samsung Galaxy Note 10 (Exynos chipset). The UE has been placed at two static locations: Location A is in the Line of Sight (LoS) of one of the four remote radio heads and in close proximity, in order to achieve the maximum theoretical throughput of 286 Mbps, according to the configurations in Table 2 and formula in [44]; Location B is in Non Line of Sight (NLoS) and closer to the edge of the cell, in order to quantify the degradation due to bad radio conditions. Finally, several tests were executed while walking and driving around the building, in order to analyse the correlations between traffic and radio parameters in pedestrian and vehicular scenarios.

Table 3 provides the details of the 4G configurations applied during our experiments, in order to complement the 5G configurations presented in Table 2. We further observe that the *proactive scheduling* functionality was activated for 5G in order to further reduce latency. This is a vendor-specific feature, which consists in a proactive allocation of grants in order to maintain the resource allocation for a user in case further uplink data arrives for transmission.

5.2. Executed Experiments and Collected Dataset

We have run three different measurement campaigns with different properties, resulting in three different datasets:

- Static LoS: The device was located close to a window, with a direct line of sight to the antenna pointing into the laboratory in order to provide good coverage within the building hosting the servers and the baseband unit. Very good propagation conditions are observed in this location.

- Static NLoS: The antenna pointing into the laboratory was deactivated. In these conditions the serving cell changed to one of the antennas pointing into the area in front of the building where the laboratory is located. Very bad conditions are observed in this scenario.

- Vehicular scenario near the building: The last dataset was collected during a driving test around the building where the antennas are deployed and also around other buildings in the area. Therefore, a high variability in the radio conditions is observed in this scenario, ranging from extremely bad to very good radio conditions.

The traffic bandwidth configuration to use depends on the test purpose of the experiment. For the experiments presented in this paper, the target is to characterise the maximum user data rate available. Therefore, the traffic bandwidth is set above the maximum that is available in the network scenario. In particular, the generated traffic was based on User Datagram Protocol (UDP) streams with a target throughput configured to a slightly higher value than the theoretical maximum data rate calculated according to the formula provided in [44] and for the network parameters described in Table 2.

Both Nemo Handy [45], a radio monitoring and drive test tool for Android devices, and 5GENESIS monitoring probes [46], a set of tools for traffic generation and automation based on iPerf[18], were used in the tests. Additionally, the Global Positioning System (GPS) location of the device was tracked during non-static tests.

The collected measurements include MAC Block Error Rate (BLER), Radio Link Control (RLC) BLER, RSRP, Reference Signal Received Quality (RSRQ), Received Signal Strength Indicator (RSSI), Rank indicator distribution, SINR, and throughput at IP, Packet Data Convergence Protocol (PDCP), MAC, and physical layer. Figures 5 and 6 are visualization examples (in space and time domain, respectively) of the PDCP throughput collected during the driving test.

In terms of analysing and troubleshooting the containers in the Analytics module, there are several tools available for container management and troubleshooting. Among those, we used Portainer[19] for observing and troubleshooting the containers' behaviour. We specifically focused on its observability features, which allowed us to get insights into the issue of container deployment impact onto the speed of getting results and also allowed us to inspect the logs for each container for any troubleshooting issues.

[18] `www.iperf.fr`, Accessed: November 2021.
[19] `www.portainer.io`, Accessed: November 2021.

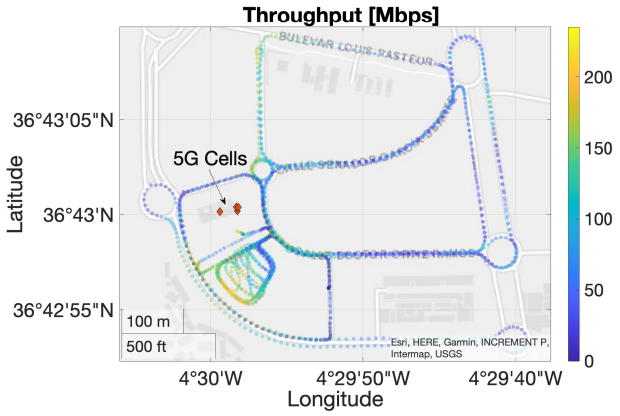

Figure 5: Geomapping of the PDCP downlink throughput [Mbps] collected during the mobile experiment.

6. Framework Evaluation

In this section, we leverage a selection of services provided by the Analytics module (Section 4), to analyze several aspects related to the 5G experiments executed in the Malaga platform. The reported analyses and results do not form an exhaustive list but rather serve as reference examples of use cases made available by the current release of the Analytics component. Therefore, we showcase different use cases for different experiments. We started by applying the statistical analysis service to investigate the performance in terms of downlink throughput obtained under static LoS and NLoS scenarios toward providing a reliable validation for the nominal KPI. We then apply correlation, feature selection, and prediction services to the mobile experiment, to investigate the behaviour of different parameters collected in a dynamic scenario, as well as the relationship of such parameters with downlink throughput.

6.1. Anomaly Detection

An essential step in data processing is the detection and removal of outliers. Two aspects need to be taken into account:

1. The presence of outliers is a key information, as it may indicate malfunctions in the setup adopted while running the experiments.
2. The removal of outliers avoids undesired skews in the data distribution that could lead to incorrect analyses and conclusions on the experiments, e.g., when further analysis methods provided by the Analytics component are used.

Outliers may appear for various reasons that require in-depth investigations. In the *5GENESIS* context, one possible explanation for the presence of outliers is the following: as anticipated in Section 4.3, experiments for KPI validation are conceived as a repetition of consecutive but independent iterations, with the ELCM restarting the components involved in the experiment, including the monitoring probes, at the beginning of each iteration. On the one hand, the presence of iterations is meant to increase data collection reliability and makes it possible to study experiment statistics on two different levels, i.e., per iteration (by isolating samples belonging to each iteration) and per experiment (by merging results from iterations). On the other hand, this start/stop functioning may provoke sudden malfunctions in the monitoring probes, particularly at the beginning/end of the iterations, leading to possible data outliers.

Figure 7 shows the presence of outliers in the downlink throughput collected during the static NLoS experiment executed in the Malaga platform, as well as its removal using Z-score. In this case, the presence of outliers leads to a misleading picture of the minimum and maximum throughput achieved in the adopted experimental setup. Indeed, the nominal range of data is narrower than the one the outliers would suggest, with a maximum throughput closer to 18 Mbps rather than the value of 45 Mbps collected in an isolated sample. Note that taking into account this value would also bias other statistical indicators, e.g., the average, therefore, it is a good norm to remove outliers before proceeding with further analyses.

All results presented in this section use outlier removal with the Z-Score method as described in Section 4.

6.2. Statistical Analysis

Figure 8 and Table 4 showcase the application of the statistical analysis service on the PDCP downlink throughput collected during (Figure 8a) static LoS and (Figure 8b) static NLoS experiments.

As discussed in Section 4.3, this service provides the results of the procedure defined in *5GENESIS* for a reliable KPI validation under well-defined network conditions and configurations. On the one hand, Figure 8 shows the statistics of the PDCP downlink throughput on a per-iteration basis.[20] On the other hand, Table 4 shows the results obtained on the entire experiment, by averaging the statistical indicators over iterations and adding a confidence interval to each indicator.

Per-iteration inspection is key for pinpointing possible malfunctions or sudden changes in network conditions and configurations, which may happen during a specific iteration. Hence, this Analytics service allows to pinpoint and help manage these situations, ultimately suggesting to discard and repeat anomalous iterations. In the reported examples, we instead observe a high stability of the throughput collected over the iterations, which is a key proof of the validity of all the iterations defining both experiments under analysis.

Per-experiment inspection allows to finalize the KPI validation procedure and, to do so, the statistical analysis service provides the results reported in Table 4. Among several possible observations, a straightforward comparison allows to highlight the significant impact of the network scenario on the achievable throughput. Indeed, a significantly higher throughput is obtained in the LoS scenario (about 271 Mbps on average, with a standard deviation of 3.55 Mbps) compared to its NLoS counterpart (about 15 Mbps on average, with a standard deviation of 1.91 Mbps).

[20] Note that a subset of 11 iterations are shown in Figure 8 for simplicity. 25 consecutive iterations are the minimum requirement defined by *5GENESIS* for obtaining a reliable and statistically significant KPI validation [28].

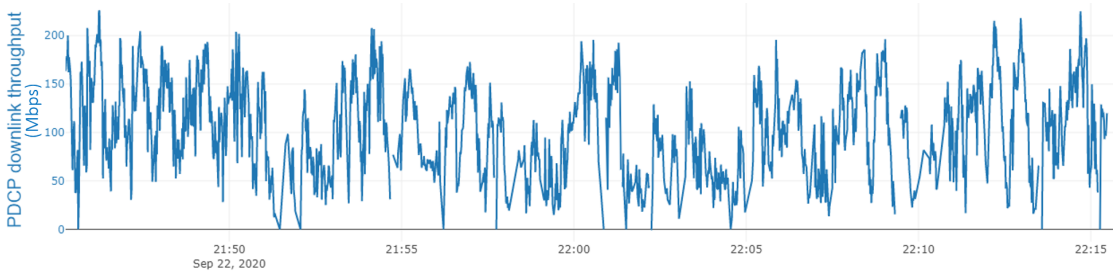

Figure 6: Time series visualization of the PDCP downlink throughput [Mbps] collected during the mobile experiment.

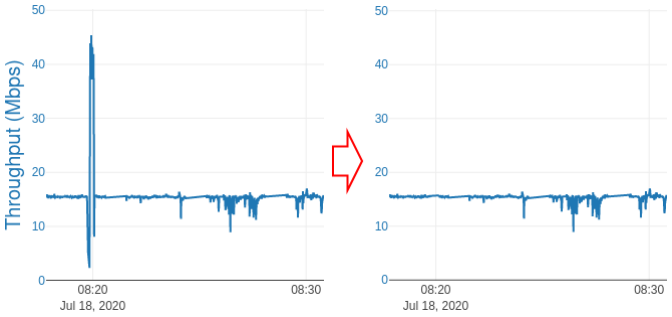

Figure 7: Z-score based outlier removal for the downlink throughput [Mbps] collected during the static NLoS experiment.

	static LoS		static NLoS	
Indicator	**Value**	**Conf. Int.**	**Value**	**Conf. Int.**
95% Percentile	277.27	0.30	16.71	0.10
75% Percentile	273.25	0.27	16.22	0.04
25% Percentile	269.12	0.27	15.35	0.25
5% Percentile	265.95	0.48	11.59	1.50
Max	283.24	4.58	19.26	0.73
Median	270.94	0.21	15.93	0.06
Mean	271.23	0.18	15.38	0.24
Min	258.27	4.72	3.80	1.36
Standard Dev.	3.55	0.29	1.91	0.39

Table 4: Test case statistics and corresponding confidence intervals for PDCP downlink throughput [Mbps], for static LoS and static NLoS experiments.

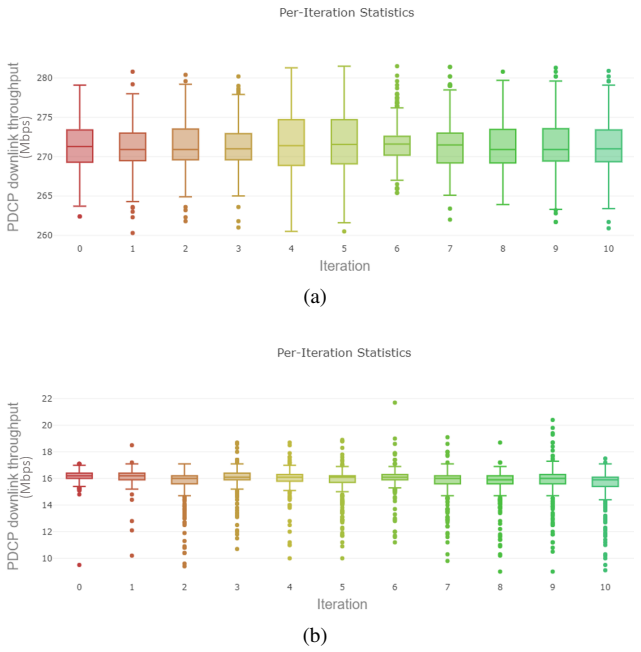

Figure 8: Statistical analysis of the PDCP downlink throughput KPI [Mbps]. Figure shows boxplots for the first 11 iterations of the static LoS (a) and static NLoS (b) experiments.

6.3. *Correlation Analysis*

A first approach for identifying the relationship between measured variables is to run a linear correlation analysis, which allows to see, in an easily interpretable way, to which degree the variables collected during an experiment are inter-dependent, might affect each other, or might be affected in the same way by other experimental variables.

Figure 9 depicts a heatmap representation of a correlation matrix, where the pairwise inter-dependency between several parameters collected during the mobile experiment is shown. In order to improve visibility, the matrix reports a subset of collected parameters selected via domain knowledge; This filtering functionality is made available by the Analytics dashboard.

In the reported example, we use Pearson correlation coefficients to evaluate the relation between each pair of parameters. On the one hand, values near 1.0 (color-coded in green in the heatmap) indicate a strong positive correlation, i.e., the measured parameters increase (or decrease) simultaneously. In particular, we see that, as expected in nominal situations, the downlink throughput observed at different layers of the protocol stack (e.g., MAC, PDCP, Physical Downlink Shared Channel (PDSCH), and RLC) positively correlated to variables related to the radio coverage experienced by the UE (e.g., RSRP, RSRQ, and SINR of the cell the UE is connected to). On the other hand, values near -1.0 (color-coded in red in the heatmap) indicate strong negative correlation, where measured values change in opposite increase vs. decrease directions. In the reported experiment, among others, the RLC downlink BLER data negatively correlated with the throughput and coverage-related parameters. We also observe strong negative correlations between

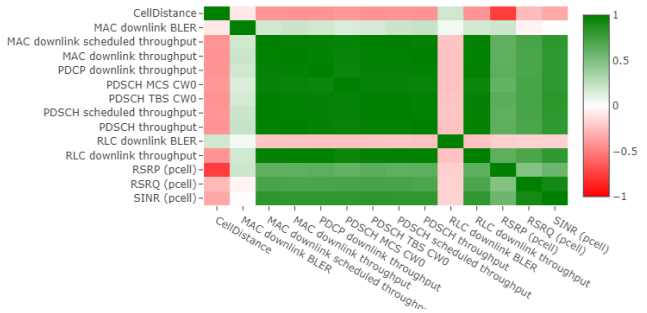

Figure 9: Correlation matrix evaluated for measurements collected during the mobile experiment. Measurements include a) throughput and other indicators at different layers (i.e, MAC, RLC, PDSCH, and PDCP), b) radio-related parameters (i.e., RSRP, RSRQ, and SINR) and c) the distance between collection points and the 5G serving cell (denoted *CellDistance* in the matrix).

the UE-cell distance (denoted as CellDistance in the correlation matrix) and both throughput and coverage-related parameters. These are again examples of a nominal behaviour in wireless scenarios: indeed, the error rate is expected to decrease (resp. increase) as the throughput increases (resp. decreases), while throughput and coverage are expected to decrease (resp. increase) as the UE-cell distance increases (resp. decreases).

From an experimental perspective, the correlation service provided by our Analytics component helps understand the inter-dependency between measured parameters. The lack of some expected correlations, as well as the presence of unexpected correlations between one or more parameters provide a key indication of the health of the experimental setup during the execution of experiments. Such indications may in turn be used to trigger troubleshooting and possibly needed reconfigurations. However, being based on linear correlation, this service is not able to pick up non-linear relationships between variables. Therefore, to gain additional insights, we need to follow a predictive analysis based route, as highlighted in the next sections.

6.4. Feature Selection

Feature selection is a precursor to the predictive analysis, and allows to focus on a subset of variables that affect a given target KPI. Furthermore, feature selection aims to minimize the size of the model that will be trained by the prediction algorithm, in order to get a simpler (and therefore faster) model without losing prediction accuracy.

To showcase the feature selection service provided by the Analytics component, we select the PDCP downlink throughput collected during the mobile experiment as our target KPI.

Aiming at revealing insightful but not too obvious relationships between the target KPI and other variables, we first apply our domain knowledge, as also done for the correlation analysis service, and exclude variables that are in essence synonyms of the target KPI, e.g., the throughput values obtained in the other layers of the 5G protocol stack (e.g., MAC and RLC).

We then adopt the LASSO algorithm to automatically select the parameters that would form a reduced set of prediction features for PDCP downlink throughput, as reported in Table 5. A

LASSO Feature selection
Elevation
MAC downlink BLER 2nd
MAC uplink throughput
PDSCH MCS CW0
PDSCH TBS CW0
PUSCH PRBs
PUSCH throughput
RACH access delay
RACH logical root sequence index
RACH pathloss
RLC uplink block rate
SINR (pcell)
Slot utilization DL

Table 5: LASSO-selected parameters for the PDCP downlink throughput collected during the mobile experiment. In total, 63 out of the 76 numerical features were discarded with a regularization parameter $\alpha = 0.1$.

total of 76 numerical features were collected during the experiment, thanks to the activation of PM/IM probes at the UE side, i.e., the *5GENESIS* iPerf-based PM probe and the Nemo Handy tool used as IM probe, as also discussed in Section 5.2. As reported in Table 5, the LASSO algorithm with regularization parameter $\alpha = 0.1$ selects 13 parameters as sufficient to build an accurate prediction model for the PDCP downlink throughput. Among other, the set of features include SINR alongside various PDSCH, Physical Uplink Shared Channel (PUSCH) and Random Access Channel (RACH) variables. Therefore, the feature selection service allows to drastically reduce the set of necessary features to predict a target KPI and build a simpler prediction model, which has advantages in terms of both performance (a simpler models is faster to train and use) and interpretability (a simpler model is easier to comprehend).

6.5. Predictive Analysis

In order to gain deeper understanding of the measured KPIs, as well as provide KPI models that could be used for further performance analysis and network optimization, the Analytics component also provides a predictive analysis service. We showcase this functionality by focusing again on the PDCP downlink throughput KPI and other parameters collected during the mobile experiment. The PDCP downlink throughput behaviour over time is shown in Figure 5, hence, providing an example of throughput experienced under mobility.

6.5.1. Prediction with LASSO-selected features

As a first example, we train a Random Forest regression for our target KPI by using the set of features obtained via LASSO selection (cf. Section 6.4). As reported in Section 4.6, Random Forest is one of the prediction algorithms available in the prediction service of the Analytics module.

To test the model's accuracy, we split the data into a training (80%) and a testing (20%) set. The results are reported in Table 6, where we observe a correlation coefficient between actual and predicted values of 0.99, and a Mean Absolute Error (MAE) of 3.72. This is a very small error, considering that

Metric	Train	Test
Mean	100.36	84.14
Standard deviation	47.18	67.08
R^2		0.99
MAE		3.72
MSE		49.83

Table 6: Accuracy results of the Random Forest prediction on the PDCP downlink throughput [Mbps] collected during the mobile experiment. The model uses LASSO-selected features (see Table 5) as input features.

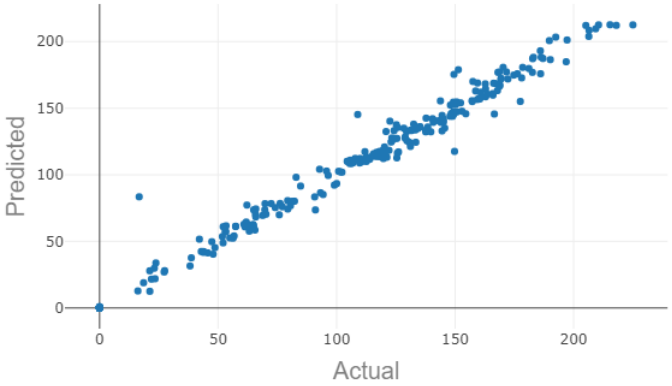

Figure 10: Visualization of actual (x-axis) vs. predicted (y-axis) PDCP downlink throughput [Mbps] for the mobile experiment. Prediction is performed via Random Forest, using LASSO-selected features (see Table 5) as input features.

the target KPI has a mean of 100.36 and a standard deviation of 47.18 in the training data. The low error also shows the feasibility of reducing the set of features to a much smaller set with little to no losses in model accuracy.

Since the error values do not paint a complete picture in a regression task, we also inspect the data points that the obtained model predicts for each actual PDCP downlink throughput value. This is shown in Figure 10, with an almost ideal distribution around the diagonal. This means that each predicted value is close to the actual PDCP downlink throughput value across the data.

In a performance monitoring use case, this model could now be used to estimate, for example, which Signal-to-noise ratio (SNR) level would be acceptable to reach a desired PDCP downlink throughput. There are many potential avenues for predicting network performance related KPIs. For example, a similar model could be trained on hardware resource aspects, such as CPU and memory, to estimate which hardware would be required to achieve good signal strength in an IoT scenario.

6.5.2. Prediction using radio coverage and distance features

In the previous predictive analysis, we obtained a highly accurate model that includes many features collected across the layers of the protocol stack. In some simpler settings, many of these parameters would be difficult to collect. It then makes sense to further reduce the set of available parameters and test the performance of the prediction service under limited data availability. Therefore, in this second example, we only take into account the parameters related to radio coverage, i.e., SINR, RSRP, and RSRQ, as well as the relative location of the UE (CellDistance), which is calculated from the latitude, longitude and elevation of the UE in relation to the 5G cell the UE is connected to.

Feature	Value
SINR	0.8038
RSRP	0.0958
RSRQ	0.0573
CellDistance	0.0431

Table 7: Feature importance values of the Random Forest model for the PDCP downlink throughput [Mbps] collected during the mobile experiment. The model only uses radio-related parameters (RSRP, RSRQ, and SINR) and UE-cell distance (CellDistance) as input features.

Metric	Train	Test
Mean	100.36	84.14
Standard deviation	47.18	67.08
R^2		0.86
MAE		14.83
MSE		609.58

Table 8: Accuracy results of the Random Forest prediction on the PDCP downlink throughput [Mbps] collected during the mobile experiment. The model only uses radio-related parameters (RSRP, RSRQ, and SINR) and UE-cell distance (CellDistance) as input features.

Table 7 shows the model learnt by the Random Forest algorithm on this reduced set of features. SINR results in the most impactful feature in this model, with RSRP and RSRQ taking a less important role. CellDistance is the least important feature and does not contribute much to the model, although we expect the distance between UE and cell to play a bigger role. In Section 6.3, we observed some correlation between CellDistance and PDCP downlink throughput, but the radio features, especially SINR, are clearly more indicative of the achieved throughput.

The accuracy of this model is described in Table 8 and confirms our expectations with respect to the lower performance when compared to the model that has access to a larger set of features. However, the model still shows acceptable performance, with 0.86 R^2 and a MAE of 14.83 Mbps, considering that data ranges between 0 and 230 Mbps. The scatter plot that contrasts the actual and predicted values in Figure 11 still exhibits an accumulation of data points around the diagonal, but with a less well-defined shape around the diagonal when compared to the same plot for the first experiment (Figure 10). Noteworthy is that this model does not predict anything above 200 Mbps although the real data achieves values up to nearly 230 Mbps. Despite the lower performance compared to the first model, this model would still be useful to predict throughput, as long as it is adopted for use cases where the obtained error margin is still acceptable.

7. Discussion and Conclusions

In this paper, we have presented the design and use of a flexible and scalable Analytics framework, based on microservices

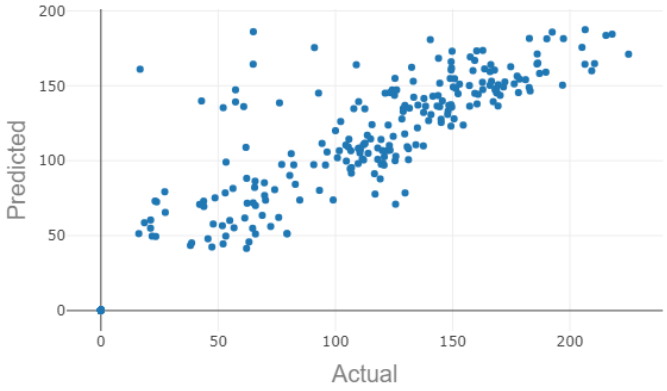

Figure 11: Visualization of actual (x-axis) vs. predicted (y-axis) PDCP downlink throughput [Mbps] for the mobile experiment. Prediction is performed via Random Forest, using radio-related features (RSRP, RSRQ, and SINR) and UE-cell distance (CellDistance) as input features.

that are open sourced and available to use by the community. While designing the Analytics framework, we catered for a diverse set of requirements supporting the needs of different platforms and their corresponding use cases. By doing so, we provided a homogeneous framework for analyzing and reporting the results of the experiments executed in quite heterogeneous platforms and service conditions. Furthermore, the Analytics framework enabled us to apply different algorithms with ease through different Analytics services. The ability to use these services on their own or in a pipeline give us different ways of analysing the datasets. We were able to observe how KPI trends evolve over the duration of the experiments, and also how different KPIs are related to each other. We could also extract correlations, detect outliers and apply advanced predictions, all done in a sustainable manner.

The Analytics framework was also of great importance while building the *5GENESIS* Malaga testbed, as we found it to be a very useful tool for troubleshooting. For example, we could gather insights for the radio deployment or identify anomalous iterations during the experimentation phase, or differences between PDCP and UDP traffic. Both of these results, when detected, warrant a more thorough analysis, as they could pinpoint to issues in the underlying platform.

Overall, our experiences within the *5GENESIS* project lead us to believe that this service-based Analytics component can be deployed in a flexible manner to suit the particularities of the underlying infrastructure, the specific needs for Analytics in the platform (i.e., which Analytics services do we need) and the performance of the Analytics service (i.e., being aware of data transfer rates between containers and time bounds on Analytics results). We have successfully used our Analytics service to derive insights both for network-based analytics to aid in the infrastructure configuration and running, as well as for providing insights towards the experimenters with regard to the service KPI trends and performance.

In terms of future work, we plan to collect feedback and, based on this, extend the Analytics service with additional capabilities, such as non-linear time series correlation and time-series forecasting. Moreover, we plan to make the APIs more generic and flexible, especially in term of retrieving and analysing results. Finally, we plan to use the Analytics component for the analysis of more heterogeneous and complex scenarios, by running experiments with multiple users possibly having different service requirements and KPIs. On this aspect, we highlight that the Analytics component can already handle multi-user/service use cases if a) the activated probes have a same reference time and b) the collected parameters are properly labelled, e.g., with user/service identifiers.

The analysis of these scenarios where, for example, eMBB and URLLC users may coexist on the same network deployment, can first help understanding and quantify the correlation between users' performance and network conditions and configurations; Then, a second step could be to extract relevant parameters and evaluate the predictability of the obtained performance (e.g., throughput and latency for eMBB and URLLC users, respectively) based on such parameters; Finally, the obtained data-driven models could be used for deriving and testing ML-based network optimization policies, e.g., in terms of eMBB/URLLC slice(s) configurations and/or radio resource management between heterogeneous users.

Acknowldgement

This work is funded by the EU H2020 research and innovation programme under grant agreement No. 815178 (5GENESIS).

References

[1] M. Series, Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond, Recommendation ITU (2015) 2083–0.

[2] A.-M. Bosneag, M. Wang, Intelligent network management mechanisms as a step towards 5G, in: 2017 8th International Conference on the Network of the Future (NOF), IEEE, 2017, pp. 52–57.

[3] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, F. Kojima, Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks, IEEE access 6 (2018) 32328–32338.

[4] E. Pateromichelakis, F. Moggio, C. Mannweiler, P. Arnold, M. Shariat, M. Einhaus, Q. Wei, Ö. Bulakci, A. De Domenico, End-to-end data analytics framework for 5G architecture, IEEE Access 7 (2019) 40295–40312.

[5] Open 5GENESIS suite. 2020.
URL `Availableonline:https://github.com/5genesis(Accessed:November2021)`

[6] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, L. Hanzo, Machine learning paradigms for next-generation wireless networks, IEEE Wireless Communications 24 (2) (2016) 98–105.

[7] S. Han, I. Chih-Lin, G. Li, S. Wang, Q. Sun, Big data enabled mobile network design for 5g and beyond, IEEE Communications Magazine 55 (9) (2017) 150–157.

[8] X. Cheng, L. Fang, X. Hong, L. Yang, Exploiting mobile big data: Sources, features, and applications, IEEE Network 31 (1) (2017) 72–79.

[9] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, J. C. Zhang, Artificial intelligence-enabled cellular networks: A critical path to beyond-5g and 6g, IEEE Wireless Communications 27 (2) (2020) 212–217.

[10] C.-X. Wang, M. Di Renzo, S. Stanczak, S. Wang, E. G. Larsson, Artificial intelligence enabled wireless networking for 5g and beyond: Recent advances and future challenges, IEEE Wireless Communications 27 (1) (2020) 16–23.

[11] R. Ferrus, O. Sallent, J. Perez-Romero, Data analytics architectural framework for smarter radio resource management in 5g radio access networks, IEEE Communications Magazine 58 (5) (2020) 98–104.

[12] Y. Liu, S. Bi, Z. Shi, L. Hanzo, When machine learning meets big data: A wireless communication perspective, IEEE Vehicular Technology Magazine 15 (1) (2019) 63–72.

[13] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, O. M. Caicedo, A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Journal of Internet Services and Applications 9 (1) (2018) 1–99.

[14] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, L. Hanzo, Thirty years of machine learning: The road to pareto-optimal wireless networks, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1472–1514.

[15] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, T. Tugcu, Intelligent network data analytics function in 5g cellular networks using machine learning, Journal of Communications and Networks 22 (3) (2020) 269–280.

[16] S. Barmpounakis, P. Magdalinos, N. Alonistioti, A. Kaloxylos, P. Spapis, C. Zhou, Data analytics for 5g networks: A complete framework for network access selection and traffic steering, International Journal on Advances in Telecommunications Volume 11, Number 3 & 4, 2018.

[17] 3GPP, System Architecture for the 5G System, TS 23.501, v16.4.

[18] S. Niknam, A. Roy, H. S. Dhillon, S. Singh, R. Banerji, J. H. Reed, N. Saxena, S. Yoon, Intelligent O-RAN for beyond 5G and 6G wireless networks, arXiv preprint arXiv:2005.08374.

[19] L. Bonati, S. D'Oro, M. Polese, S. Basagni, T. Melodia, Intelligence and learning in o-ran for data-driven nextg cellular networks, arXiv preprint arXiv:2012.01263.

[20] D. Sarlis, N. Papailiou, I. Konstantinou, G. Smaragdakis, N. Koziris, Datix: A system for scalable network analytics, ACM SIGCOMM Computer Communication Review 45 (5) (2015) 21–28.

[21] J. Liu, F. Liu, N. Ansari, Monitoring and analyzing big traffic data of a large-scale cellular network with hadoop, IEEE network 28 (4) (2014) 32–39.

[22] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Communications of the ACM 51 (1) (2008) 107–113.

[23] K. Yang, R. Liu, Y. Sun, J. Yang, X. Chen, Deep network analyzer (dna): A big data analytics platform for cellular networks, IEEE Internet of Things Journal 4 (6) (2016) 2019–2027.

[24] A. Padmanabha Iyer, L. Erran Li, M. Chowdhury, I. Stoica, Mitigating the latency-accuracy trade-off in mobile data analytics systems, in: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 513–528.

[25] R. Gonzalez, F. Manco, A. Garcia-Duran, J. Mendes, F. Huici, S. Niccolini, M. Niepert, Net2vec: Deep learning for the network, in: Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, ACM, 2017, pp. 13–18.

[26] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, H. Ma, Understanding operational 5g: A first measurement study on its coverage, performance and energy consumption, in: Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, 2020, pp. 479–494.

[27] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, Z.-L. Zhang, A first look at commercial 5g performance on smartphones, in: Proceedings of The Web Conference 2020, 2020, pp. 894–905.

[28] A. Díaz Zayas, G. Caso, Ö. Alay, P. Merino, A. Brunstrom, D. Tsolkas, H. Koumaras, A modular experimentation methodology for 5g deployments: The 5genesis approach, Sensors 20 (22). doi:10.3390/s20226652.
URL https://www.mdpi.com/1424-8220/20/22/6652

[29] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li, et al., Lumos5g: Mapping and predicting commercial mmwave 5g throughput, in: Proceedings of the ACM Internet Measurement Conference, 2020, pp. 176–193.

[30] J. Riihijarvi, P. Mahonen, Machine learning for performance prediction in mobile cellular networks, IEEE Computational Intelligence Magazine 13 (1) (2018) 51–60.

[31] K. Kousias, Ö. Alay, A. Argyriou, A. Lutu, M. Riegler, Estimating downlink throughput from end-user measurements in mobile broadband networks, in: 2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), IEEE, 2019, pp. 1–10.

[32] N. Khademi, D. Ros, M. Welzl, Z. Bozakov, A. Brunstrom, G. Fairhurst, K.-J. Grinnemo, D. Hayes, P. Hurtig, T. Jones, et al., Neat: a platform- and protocol-independent internet transport api, IEEE Communications Magazine 55 (6) (2017) 46–54.

[33] L. Fallon, S. van der Meer, J. Keeney, APEX: An engine for dynamic adaptive policy execution, in: 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), IEEE, 2016, pp. 699–702.

[34] H. Koumaras, D. Tsolkas, G. Gardikis, P. M. Gomez, V. Frascolla, D. Triantafyllopoulou, M. Emmelmann, V. Koumaras, M. L. G. Osma, D. Munaretto, E. Atxutegi, J. S. d. Puga, O. Alay, A. Brunstrom, A. M. C. Bosneag, 5genesis: The genesis of a flexible 5g facility, in: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2018, pp. 1–6. doi:10.1109/CAMAD.2018.8514956.

[35] A. Díaz Zayas, D. Rico, B. García, P. Merino, A coordination framework for experimentation in 5g testbeds: Urllc as use case, in: Proceedings of the 17th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 71–79. doi:10.1145/3345770.3356742.
URL https://doi.org/10.1145/3345770.3356742

[36] 5GENESIS project deliverable d3.6, monitoring and analytics (release b).
URL https://5genesis.eu/wp-content/uploads/2021/05/5GENESIS_D3.6_v1.0_FINAL.pdf,Apr.2021(Accessed:November2021)

[37] Ö. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, et al., Experience: An open platform for experimentation with commercial mobile broadband networks, in: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, ACM, 2017, pp. 70–78.

[38] M. Peón-Quirós, V. Mancuso, V. Comite, A. Lutu, Ö. Alay, S. Alfredsson, J. Karlsson, A. Brunstrom, M. Mellia, A. Safari Khatouni, et al., Results from running an experiment as a service platform for mobile networks, in: Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization, ACM, 2017, pp. 9–16.

[39] 5GENESIS project deliverable d5.4, documentation and supporting material for 5g stakeholders (release b).
URL https://5genesis.eu/wp-content/uploads/2021/08/5GENESIS-D5.4_v1.0.pdf,Aug.2021(Accessed:November2021)

[40] B. Iglewicz, D. C. Hoaglin, How to detect and handle outliers, Vol. 16, Asq Press, 1993.

[41] 5GENESIS project deliverable d6.1, trials and experimentation (cycle 1).
URL https://5genesis.eu/wp-content/uploads/2019/12/5GENESIS_D6.1_v2.00.pdf,Nov.2019Accessed:November2021)

[42] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.

[43] S. Le Cessie, J. C. Van Houwelingen, Ridge estimators in logistic regression, Journal of the Royal Statistical Society: Series C (Applied Statistics) 41 (1) (1992) 191–201.

[44] 3GPP, Technical Specification Group Radio Access Network;NR;User Equipment(UE) radio access capabilities, Technical Specification (TS) 38.306, 3rd Generation Partnership Project (3GPP), version 16.6.0 (09 2021).

[45] Nemo handy, keysight,2021. [online]. available.
URL https://www.keysight.com/es/en/product/NTH00000B/nemo-handy-handheld-measurement-solution.html(Accessed:November2021)

[46] Uma agents, university of malaga,2021. [online]. available.
URL https://gitlab.com/OpenTAP/Plugins/university-of-malaga/uma-adb-agents/-/tree/develop/Tap.Plugins.UMA.AdbAgents/Tap.Plugins.UMA.AdbAgents/AdbAgents/Agents(Accessed:November2021)

Appendix A. Analytics API examples

The following excerpts are examples of API calls and responses from each analytics service, including data handler, statistical analysis, correlation, feature selection and prediction.

Appendix A.0.1. Data Handler API

```
data-handler-URL/get_data/datasource/experimentId
```

Response from /get_data:


```
{
    "PDCP downlink throughput": {
        "1600811115000": 163.4,
        "1600811116000": 160.8,
        ...
    },
    "CellDistance": {
        "1600811115000": 44.9959961221,
        "1600811117000": 47.3855153703,
        ...
    },
    ...
}
```


Appendix A.0.2. Statistical Analysis API

```
stat-analysis-URL/statistical_analysis/database?
    experimentid=123&kpi=Throughput&measurement=
    throughput_measures
```

Response from /statistical_analysis:


```
{
    "Throughput": {
        "Iteration Statistics": {
            0: {
                "5% Percentile": 167,
                "25% Percentile": 176,
                ...
            },
            1: {
                "5% Percentile": 171.8,
                "25% Percentile": 177,
                ...
            },
            ...
        },
        "Test Case Statistics": {
            "5% Percentile": {
                "Value": 171.43958333333333,
                "Confidence Interval":
                    0.5810154812485544
            },
            "25% Percentile": {
                "Value":        175.890625,
                "Confidence Interval":
                    0.3156676239473986
            },
            ...
        }
    }
}
```


Appendix A.0.3. Linear Correlation Analysis API

```
/correlate/fields/datasource/experimentId?
    remove_outliers=zscore
```

Response from /correlate/fields:


```
{
    "CellDistance": {
        "CellDistance": 1,
        "PDCP downlink throughput": -0.382707002,
        "SINR": -0.336993597,
        ...
    },
    "PDCP downlink throughput": {
        "CellDistance": -0.382707002,
        "PDCP downlink throughput": 1,
        "SINR": 0.825896797,
        ...
    },
    "SINR": {
        "CellDistance": -0.336993597,
        "PDCP downlink throughput": 0.825896797,
        "SINR": 1,
        ...
    },
    ...
}
```


Appendix A.0.4. Feature Selection API

```
feature-selection-URL/selection/datasource/
    algorithm/target?experimentid=123&measurement
    =throughput_measures
```

Response from /selection:


```
{
    "Features - Original": {
        0: "MAC downlink BLER",
        1: "MAC downlink BLER 1st",
        2: "MAC downlink BLER 2nd",
        3: "PDCP downlink block rate",
        4: "RLC downlink throughput",
        ...
    },
    "Features - Selected": {
        0: "PDCP downlink block rate",
        1: "RLC downlink throughput"
    },
    "Score": {
        "MAC downlink BLER": 0,
        "MAC downlink BLER 1st": 0,
        "MAC downlink BLER 2nd": 0,
        "PDCP downlink block rate": 0.848355061,
        "RLC downlink throughput": 0.0458724611,
        ...
    }
}
```


Appendix A.0.5. Prediction API

Example that includes outlier removal:

```
prediction-URL/train/datasource/algorithm/target?
    experimentid=123&remove_outliers=zscore
```

Response from /train:


```
{
    "coefficients": {
        "MAC downlink throughput": 0.4974099716,
        "PDCP downlink block rate": 0.2324643848,
        "RLC downlink throughput": 0.1372971483,
        "MAC downlink scheduled throughput":
            0.0665096093,
        "PDSCH throughput": 0.0533150278,
        ...
    },
    "real_predicted_values": {
        "y_pred": {
            0: 109.8001233468,
            1: 141.7893004874,
            ...
        },
        "y_test": {
            0: 110.3,
            1: 130.7,
            ...
        }
    },
    "results": {
        "R2 score": 0.9956903516,
        "Mean Squared Error": 19.393652019,
        ...
    }
}
```