"5G for Drone-based Vertical Applications"

D2.3 Report on algorithms, mechanisms and tools for data analysis and visualisation

Document ID:	**D2.3**
Deliverable Title:	**Report on algorithms, mechanisms and tools for data analysis and visualisation**
Responsible Beneficiary:	**FRQ**

Topic:	**H2020-ICT-2018-2020/H2020-ICT-2018-3**
Project Title:	**Unmanned Aerial Vehicle Vertical Applications' Trials Leveraging Advanced 5G Facilities**
Project Number:	**857031**
Project Acronym:	**5G!Drones**
Project Start Date:	**June 1st, 2019**
Project Duration:	**42 Months**
Contractual Delivery Date:	**M24**
Actual Delivery Date:	**May 28th, 2021**
Dissemination Level:	**Public (PU)**
Contributing Beneficiaries:	**UO, THA, INV, HEP, NCSRD, AU, UMS, INF, NOK, RXB, EUR, DRR, CAF, FRQ.**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857031.

Document ID:	D2.3
Version:	V1.2
Version Date:	28.05.2021
Authors:	Farid Benbadis (THA), Agathe Blaise (THA), Youcef Salah (THA), Alan Branchereau (THA), Tanel Järvet (CAF), Maido Hiiemaa (CAF), Hamed Hellaoui (AU), Tarik Taleb (AU), Gokul Srinivasan (RXB), Ott Tahk (HEP), Piotr Dybiec (DRR), Paweł Korzec (DRR), Ilkka Kansala (NOK), Katri Pihlajaviita (NOK), Gregor Mogeritsch (FRQ), Syed Ali (UO), Antti Pauanne (UO), Jean Cerrillo Fernandez (UO), Vaios Koumaras (INF), Aggeliki Papaioannou (INF), Christos Sakkas (INF) Pradyumna Vyshnav (UML), Yannick Schaffer (INV), Pawel Montowtt (INV), Abderrahmane Abada (AU), Oussama Bekkouche (AU), Fofy Setaki (COS), George Lymperopoulos (COS), Stavros Kolometsos (NCSRD), Harilaos Koumaras (NCSRD), George Makropoulos (NCSRD), Anastasios Gogos (NCSRD)
Security:	Public (PU)

Approvals

	Name	Organization	Date
Coordinator	Jussi Haapola	UO	28.05.2021
Technical Committee	Farid Benbadis	THA	26.05.2021
Management Committee	Project Management Team	AU, CAF, NOK, FRQ.	26.05.2021

Document History

Version	Contribution	Authors	Date
V0.1	Version for Draft ToC	All	03.12.2020
V0.2	ToC input from partners	All	25.01.2021
V0.3	More ToC input from partners	All	10.03.2021
V0.4	Restructuring and updates	All	21.03.2021
V0.5	More restructuring, added contributions	All	25.03.2021
V0.6	Added more contributions	All	07.04.2021
V0.7	First internal review	All	27.04.2021
V1.0	Internal review	All	29.04.2021
V1.1	Improved version after internal review	AU, CAF,NOK, FRQ	17.01.2021
V1.2	Final Delivery Version	THA, FRQ.	26.05.2021

Executive Summary

The 5G!Drones project aims to trial use case scenarios of UAV on top of 5G facilities to evaluate 5G systems in supporting UAV vertical industry requirements. Four representative use cases with different scenarios have been selected to run on four different 5G facilities.

This document reports on activities related to data collection, analysis, and visualisation in the scope of WP2 work within the 5G!Drones project. It is one of multiple deliverables within WP2 and needs to be read in context with the other documents [1] [2] [3] as well as deliverables from WP3 and WP4 [4].

One key element of the 5G!Drones architecture is the ability to collect, manage and visualise KPI data during the trials. This allows appropriate analysis and reporting on the trials' results to give all partners the opportunity to generate learnings and move forward in their respective areas of development and research.

The document focuses on the practical data collection during the trials and its used components in chapter 3. Both sides, 5G facilities as well as the Trial Controller part are taken into consideration. Chapter 4 is dedicated to the data visualisation and analysis. Furthermore, in addition to examining the practical application during the trials, the partners also provide extensive insights on current state of the art and research in machine learning mechanisms.

Table of Contents

List of Figures

List of Abbreviations

3GPP	3rd Generation Partnership Project
4D	X, Y, Z and time
5G	5th Generation Cellular Technology
5GS	5G system
A2Vs	Advanced Aerodynamics Vessels
AAS	Active Antenna Systems
AGL	Above Ground Level
AKA	Also-Known-As
ANSP	Air Navigation Service Provider
ARC	Air Risk Class
ASU	Arbitrary Strength Unit
ATC	Air Traffic Control
ATM	Air Traffic Management
AWS	Amazon Web Services
BS	Base Station
BVLOS	Beyond Visual Line of Sight
C2	Command and Control
C3	Command, Control and Communication
CAA	Civil Aviation Authority
CAGR	Compound Annual Growth Rate
CEPT	Conférence Européenne des administrations des Postes et des Télécommunications (European Conference of Postal and Telecommunication Administrations)
CNN	Convolutional Neural Networks
COVID-19	Coronavirus disease 2019
CSP	Communication Service Provider
DAA	Dynamic Airspace Allocation
DBSs	Drone Base Stations
dFPL	drone Flight Plan
DGAC	Direction Générale de l'Aviation Civile (French Civil Aviation Authority)
DL	Downlink
DSA	Director/-ate Safety Airspace Airports and Information Services (Eurocontrol EATM)
DTT	Digital Terrestrial Television
Dx.y	Deliverable number y of the Work Package x
EASA	European Aviation Safety Agency
EC	The European Commission
ECC	Electronic Communication Committee (in CEPT)
eMBB	Enhanced Mobile BroadBand
eNodeB	E-UTRAN Node B, also known as Evolved Node B
ETSI	European Telecommunications Standards Institute
EVLOS	Extended Visual Line of Sight
eVTOL	Electric Vertical Takeoff and Landing
FAA	Federal Aviation Agency (in the USA)
FSS	Fixed Satellite Service (earth to space)
GCS	Ground Control Station
GDPR	General Data Protection Regulation

GPRS	General Packet Radio Service
GPS	Global Positioning System
GPT-C	GPRS Tunneling Protocol communication protocol
GRC	Ground Risk Class
HCAA	Hellenic Civil Aviation Authority
HD	High-definition (video)
HIDSes	Host-based Intrusion Detection Systems
HITL	Hardware In The Loop
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IMT	International Mobile Telecommunications
IoT	Internet of Things
ISS	International Space Station
ITU	International Telecommunication Union
ITU-R	ITU Radiocommunication Sector
KPIs	Key Performance Indicators
LAANC	Low Altitude Authorization and Notification Capability in the US
LIDAR	Light Detection and Ranging
LMF	Location Management Function
LTE	Long-Term Evolution
MAP	French “Manuel d’Activités Particulières” - Specific Activities Manual
MFCN	Mobile Fixed Communications Network
mMTC	massive Machine-Type Communication
MNO	Mobile Network Operator
NG RAN	Next Generation Radio Access Network
NIDSes	Network-based Intrusion Detection Systems
NS	Network Slicing
OBC	On Board Computers
OPEX	OPerating EXpense
QoE	Quality of Experience
QoS	Quality of Service
RC	Radio Communication
RPAS	Remotely Piloted Aircraft Systems
SAR	Search and Rescue
SESAR JU	The Single European Sky ATM Research Joint Undertaking
SITL	Software In The Loop
SLA	Service-Level Agreement
SORA	Specific Operations Risk Assessment
SPoC	Single Point of Contact
sUAS	small Unmanned Aerial System
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UE	User Equipment
UL	Uplink
uRLLC	Ultra-Reliable and Low Latency Communication

USSP	U-space Service Provider
UTM	Unmanned Traffic Management
VLOS	Visual Line of Sight
WP	Work Package

1 INTRODUCTION

1.1 Objective of the Document

This document is the deliverable to report on task T2.4: Tools for experiment data analysis and visualisation.

The goal of this document is to provide information on mechanisms for the management and analysis of the data that will be generated during the trials. These mechanisms will be applied in WP4 to allow visualisation and reporting which will be used both at trial execution time and for the post-trial evaluation of the results.

1.2 Structure of the Document

This Deliverable is structured in five chapters:

- Chapter 1 presents an introduction of the deliverable focusing on its objectives, structure, and target audience;
- Chapter 2 presents a quick general system overview and describes the general approach and key aspects to data collection, analysis, and visualisation;
- Chapter 3 gives an overview of tools and mechanisms used for data collection;
- Chapter 4 describes in detail different approaches on data visualisation and analysis including descriptions on machine learning principles;
- Chapter 5 concludes this deliverable.

Scope of T2.4., and therefore also for D2.3 according to the grant agreement [5] is:

Task 2.4: Tools for experiment data analysis and visualisation

The goal of this task is to provide sophisticated mechanisms for the management and analysis of the data that will be generated during the trials. These mechanisms will be applied in WP4. This task will face important challenges.

First, very large volumes of experimental data will be generated during the trials; these data pertain to both the UAV service level (e.g., video traces, sensor readings, etc.) and the 5G facility level (e.g., packet-level measurements, signal coverage reports, latency measurements, etc.). Second, these data are often unstructured, have multiple dimensions, and involve multiple KPIs to measure. The expected challenges pertain particularly to the management, analysis, and the visualisation of the experimental data and call for (i) big data management techniques, (ii) the application of data analytics and/or machine learning techniques for the analysis of trial results, (iii) development of visualisation tools which will be used both at trial execution time and for the post-trial evaluation of the results.

The work in this task place efforts on data analysis and the intuitive representation of trials results. This feature is becoming essential to process and understand the volumes of data generated by automated trial systems. This task will use and extend open-source tools (such as Elasticsearch, Logstash, Kibana, collectively known as the ELK stack for real-time actionable insights on any type of unstructured data. Notably, partners in 5G!Drones already have significant experience applying this solution and plan to extend these tools with new features, such as new visualisation plugins relevant to 5G parameters and advanced statistical data analysis, correlation techniques, and machine learning algorithms. The algorithms, mechanisms and tools developed in T2.4 will be reported in D2.3, while the related software will be released in D2.6.

1.3 Target Audience

This deliverable is mainly addressed to:

- The Project Consortium and Stakeholders to follow and validate project objectives.
- The contributing partners within WP2 as well as all partners involved in Trail execution as part of WP4.
- The general public as well as other partners interested in the 5G!Drones solution.

2 KEY ASPECTS OF DATA COLLECTION, ANALYSIS, VISUALISATION

2.1 Main Objectives

The main objective is to prepare and provide mechanisms for the management and analysis of the data that is generated during the trials to allow reporting and drawing conclusions from the experiments. First, diverse type of data will be generated during the trials; these data pertain to both the UAV service level (e.g., video traces, sensor readings, etc.) and the 5G facility level (e.g., packet-level measurements, signal coverage reports, latency measurements, etc.). The expected challenges pertain particularly to the management, analysis, and the visualisation of the experimental data.

In addition, it is important to keep up to date on latest state-of-the art methods in collection, analysis and visualisation to have the capability to decide on practical usages of new technologies and algorithms.

2.2 High-Level 5G!Drones System Architecture

Figure 1 show the general 5G!Drones high level architecture as agreed in previous deliverables [2] [3]. The scope of the deliverable covers T2.4 which addresses data analysis and visualisation aspect of this architecture.

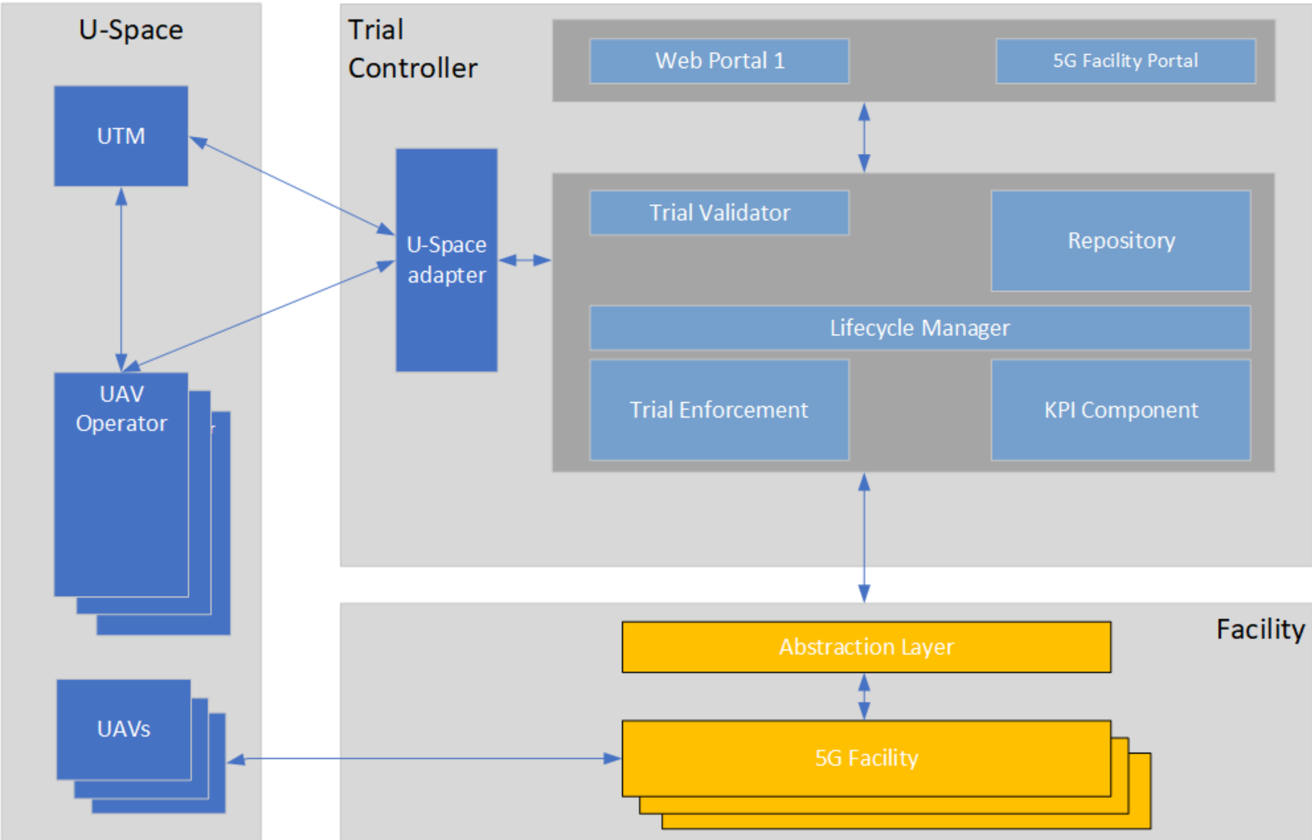

Figure 1: High-Level 5G!Drones System Architecture

2.3 Data Analysis Business Aspects and Impact

In the current information era, data, no matter if technical or not, is more available than ever before, and it is of particular importance to companies' operation. Each company gathers relevant information in the best possible way to increase productivity in business processes and functions since data efficient collection, processing and analysis can also help businesses become more effective and profitable, as well as forecast future business trends and introduce new business model. However, few businesses have really grasped the significance of effectively analysing data that contributes to company or industry growth, transition, or even creativity. Data patterns may disclose previously unknown details, allowing for the detection of issues and the creation of new methods of operation.

5G!Drones Data Analysis team except for its technical role has also focused on locating and analysing appropriate data and providing visualisation of results on values that will create business and market related impact in the drone industry through the use of 5G. It is important to map out the creation of business value by utilising analytics to establish a balance between the need for immediate technical value and business value leading to essential business impact and an exploitable competitive advantage.

To meet these requirements successfully, there is a close cooperation among WP2 and WP4 teams since the beginning of the project. During WP4 use case trials, it is essential to define and measure not only technical KPIs but also business KPIs, record their related data values and through data analysis provide results delivering tactical business value, and identify how these activities will be maintained and evolve in business impact for the drone industry (vital input to 5G!Drones T1.1 and T2.4 and their deliverables).

Within this framework, T2.4 and T4.2 teams, during the initial trial performed so far, have focused on identifying and proposing data values as well as KPIs that may have potential business impact in the drone industry and need to be further analysed and metrics for visualisation to be obtained. Use case and scenario leaders as well as drone business partners have been queried in order to assess the potential business impact from the data analysis results and link specific data analysis findings from the use cases to the project's data analytics, business processes and models (monitored in T1.1 and T2.4 respectively).

Although, this approach for defining and analysing business KPIs has already been initiated (mainly during the D4.3 Trials plan compiling process [4]), more processes will mainly take place at a future stage of the project and will be reported at future deliverables as T2.4 progresses and data analysis mechanisms become fully available/operational and matched to the WP4 use case trials. The Data Analysis team has already co-aligned its technical activities with this business scope (e.g. business impact), with the activities of T1.1 (Analysis of the UAV business and regulatory ecosystem and the role of 5G technology) and the use case trials (T4.2).

So far, specific data values of business interest have been spotted that may have business value/interest and impact on the proposed drone's industry business models and processes are:

- Low latency and drone responsiveness, efficiency and autonomy;
- 5G network connectivity, coverage and signal quality;
- 5G data rate and bandwidth and drone image/live video efficiency and quality;
- Number of connected drones/devices per unit area and new drone usages;
- Drone vertical can expect to have 5G coverage at normal operating altitudes of 20m to 120m AGL;
- Drone verticals can install their cloud native applications on the 5G MEC/Edge, which reduces the need to forward the entire data stream to a central server especially when using video analytics;
- Connection quality allows to use smaller safety buffer where space for flight corridors is limited asset in urban areas. Risk analysis requires a larger buffer area if there is a risk of network loss;
- Teleoperations for dedicated operations needs maximum total glass-to-glass latency of 300ms, incl. forwarding video: from drone´s camera to onboard 5G UE/communication device and to the MEC server and to the drone operator's computer.

  The 4K video stream via 5G allows it to be used for teleoperation with VR devices, computers, gamepads, joysticks etc. Higher quality video analysis (the better the original video quality, the more accurate the answers to the analysis of Computer Vision solutions or extracts for other applications). TV broadcasts also require at least 4K quality.

- To allow a very large number of IoT sensors to be placed in a small area. Monthly fee and battery lifetime of IoT sensor is much better than mobile phone.

2.4 Requirement and design phase

The requirement and design phase for data collection encloses defining the requirements and realisations of various characteristics in the different phases of a data collection pipeline. The design phase begins with defining the requirements for the data collection including the targets of the use case scenario. Once defined, the available tools and resources can be determined, leading to the definition of the data itself. The design phase is fundamentally important in multiple facets from further usage and refinement of the collected data to aspects such as data privacy and security.

2.4.1 Data sources and types

To initiate the design phase of data collection, the requirements for the data to be collected should be outlined. The requirements for data collection are heavily based on the requirements and needs to be defined for the outcome of the use case scenario. All data sources that are available for a scenario should be gathered and evaluated against the targets of the use case scenario in question. For the use case scenarios in 5G!Drones, the use case requirements and defined KPIs are presented in the D1.1 Use case specifications and requirements [6].

Defining the requirements initialises the basis for the next steps within the design phase. With properly specified requirements, the possible tools and the definitions of the data can be investigated and determined.

2.4.2 Tools and resources

Based on the defined data sources and other requirements, the layout of actualisation of the data collection is proceeded by gathering the tools and resources available. Depending on the definition of the requirements, a feasible way to fulfil and perform them should be determined. In practice, this phase consists of defining the tools that are needed and which of these tools are available for performing the data collection.

In addition to the tools required for collection of the data, the remaining steps of a data collection pipeline should be defined. The pipeline defined for 5G!Drones is presented in chapter 3 (Data collection) and chapter 4 (Data visualisation and analysis).

2.4.3 Data collection management

Regardless of the means of data collection or the collected data, there are aspects that should considered in order to achieve the most value from the data collection process. In essence, these aspects focus on the coordination and management occurring within the data collection pipeline.

To ensure the availability of the data intended to be collected, an important aspect in planning is to validate that relevant data is collected and that it is not corrupted during different processing phases. Another important aspect, which is crucial for the utilisation of the data, is to ensure that the requirements for adequate data collection speed and accuracy are met. Furthermore, it is necessary to, for example, determine and define the calibration level of measurement devices. These aspects are required to maximise the reliability of further usage and refinement of the data especially in analysation purposes.

Time stamp synchronisation with different measuring system is crucial for allowing analysis of possible deeper dependencies between actions and reactions in post-processing phase. In practice, clock synchronisation is needed before and during a trial, which for instance, can be achieved via implementation of Time-Sensitive Networking (TSN). TSN constitutes a set of profiles and standards, which offer guidelines and tools for implementation of TSN and enabling deterministic services through IEEE 802 networks. TSN standards focus on aspects including time synchronisation, transport, and quality of service, while TSN profiles offer complete specifications from protocols to configurations suited for a given TSN application [7].

A practical solution for clock synchronisation in a computer network is Precision Time Protocol (PTP), which is utilised in time sensitive networking. Within a local area network, utilisation of PTP can provide microsecond level accuracy, enhancing performance of a system and making it suitable for, for example, measurement and control purposes [8].

Modern measurement systems create massive amount of data and only fraction of it is interesting enough to be stored, which raises the question of how to separate valid data from spam data. One solution is a given possibility to pilot and experimenters to set different labelling values to KPI data, see sequence diagram of Figure 28. Similarly, the amount of data collected and stored can be reduced by identifying potential redundancies between data provided by different sources.

Data usage during and after project are defined in D6.1 Data Management Plan and quality and risk management plan [9].

2.4.4 Data model

After defining the targets and requirements of the data collection process and outlining the tools and resources available, another important step is to define the model or models of the collected data. The step is crucial especially in terms of further usage and refinement of the collected data. With a well-defined data model, the cost of time spent for the post-processing phase can be decreased while increasing the value achieved as a result.

The data as well as the data model are inherently determined by the sources and tools utilised to collect the data, however, the used and implemented tools can be utilised to modify the data model after the data has been collected. For timestamp-based data, calibration and synchronisation of the system within which the data is collected is crucial as discussed in the previous section. Further usage of the data that combines multiple data sources may benefit from unified characteristics of data models, which should often be striven for. An example of such usage that benefit from well-designed data is implementation of machine learning models.

2.4.5 Security and privacy

Data collection and gathering must be in a manner, which respects human right for data privacy as required by General Data Protection Regulation, see Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) [10]. In practice, a use case scenario must be designed in a way, where, for example, video stream with audio is collecting only needed information to achieve scenario specific target and not collecting any unneeded data. Secondly, how involved persons can see possible person information and personal data is properly deleted permanently from all valid mid-points in data flow chain, not only from end storage.

Data ownership as well as facility rights and responsibilities are defined before the trials, so the post-processing can be handled properly. In 5G!Drones Deliverable 4.3 [15] the details of the type of data is discussed; summarized it can be said that the Trials do only collect results of technical experiments, no personal data is stored.

2.5 Data protection and privacy concerns

The definition of personal data follows the definition of personal data in the article 4 paragraph (1) of general data protection regulation (GDPR). This includes information, such as, full name, email, phone number and social number, but also biometric information, such as facial images. In addition, data that can be used to track persons, such as GPS coordinates or IP addresses are personal data.

To comply with GDPR, personal data should be protected using appropriate organisational and technical safeguards. GDPR does not define exact safeguards that should be used because these depend on the risks caused by the data processing, but in general, the technical safeguards considered best practice are:

1) minimise data collection (only collect data that is required to provide the service);

2) anonymise or pseudo-anonymise data when appropriate;
3) remove old data (such as, analytics or logs) after a set period (e.g., one month);
4) encrypt data in transit and at rest;
5) use authorisation controls to restrict access to the data;
6) use immutable audit logging to monitor access to the data; and
7) enable backups, so data can be restored.

The article 35 of GDPR requires that Data Protection Impact Assessment (DPIA) must be conducted when the processing activity is likely to result in high risk to the rights or freedom of natural persons. The DPIA process is used to discover potential risks caused by the data processing activity and how these risks could be minimised. The article 35 paragraph (3) gives the following cases as an example when DPIA is mandatory:

- systematic and extensive evaluation of personal aspects relating to natural persons which is based on automated processing, including profiling, and on which decisions are based that produce legal effects concerning the natural person or similarly significantly affect the natural person;
- processing on a large scale of special categories of data referred to in Article 9 paragraph (1), or of personal data relating to criminal convictions and offences referred to in Article 10;
- or a systematic monitoring of a publicly accessible area on a large scale.

Regarding the 5G!Drones Project, previously defined security requirements address most of the data protection concerns, but in addition, data protection impact assessment is done within WP4 to minimise any potential risks of data protection issues.

3 DATA COLLECTION

This chapter describe the main components and mechanisms to collect data during the 5G!Drones trials. This includes facility specific mechanisms as well as components which are part of the Trial Controller architecture and therefore used during all trials. Delivery of the described software components is done with D2.6.

3.1 Facility Data Collection Tools

This chapter focuses on the data collection concepts used at the 5G facilities that are used to perform the Trials. See Figure 1: High-Level 5G!Drones System Architecture for the architecture overview.

3.1.1 Cell Performance Monitoring tool

There are several challenges that arise when conducting experiments and measurement tests using UEs. To name a few: possible use of different mobile applications/tools to access the information, differences in the log files which contain the results and sometimes the lack of visual representation which can assist the user in order to get a clearer picture of the experiment. To tackle these challenges, NCSRD and specifically the team from Media Networks Lab, has started developing the “Cell Performance Monitoring Tool” which can be a key application not only in performing tests but also in storing, organising and visualising these kinds of experiments.

Provided measurements

The measurements that will be supported by the application can be categorised into 3 main KPIs: Signal strength, Latency and Throughput, with the last two being currently under development. The application will be available as an Android App for mobile devices with an OS higher than version 9 and will also support the use of two different SIM cards (the user will be able to select specific SIM for data usage). A detailed list of the available metrics that the application can provide is presented below:

The part of the application that measures the signal strength has already been developed and each measurement is combined with a location and timestamp. Details towards the available measurements are listed below (detailed explanations of the abbreviations can be found at [11]):

- fcn (EARFCN, NRFCN, ARFCN…)
- cell-id
- enb-id
- asu-level
- type (LTE, NR, HSPA, HSPA+, EDGE ...)
- strength
- PLMN (Public Land Mobile Network) (MCC (Mobile Country Code), MNC (Mobile Network Code))

Extra measurements for NR:

- CSI-RSRP
- CSI-RSRQ
- CSI-SINR
- SS-RSRP
- SS-RSRQ
- SS-SINR

Extra measurements for LTE:

- RSRP
- RSRQ
- RSSI
- CQI
- RSSNR

Moreover, the application will support latency as well as throughput (uplink & downlink) and both functionalities are under development. The main screen of the application is depicted in Figure 2.

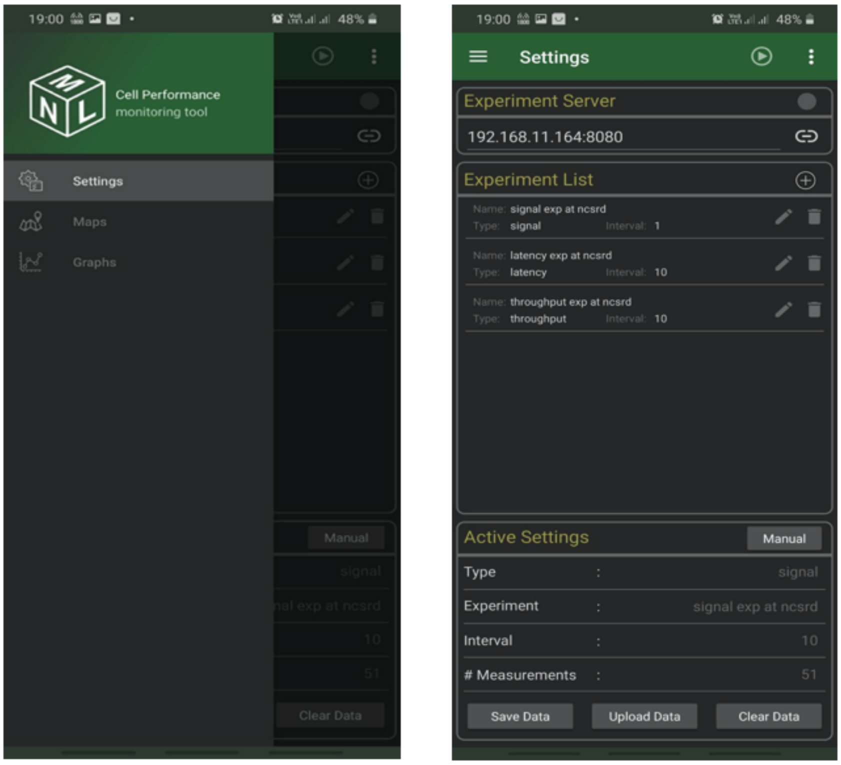

Figure 2: Cell Performance Monitoring Tool

Communication with the backend server

The app, which has been designed with the aim to work more efficiently with a backend server approach, is responsible for:

- creating new experiment IDs and storing them to a MongoDB;
- maintaining experiment lists with easy tagging and capabilities of adding details and metadata for the trial;
- sending a configuration list to the UEs to define the different types of measurements / experiment parameters that the user is able to perform;
- acting as an endpoint for the UE to connect and perform latency and throughput tests;
- storing the results of the experiments inside an ElasticSearch instance.

Even without the use of a backend server, the app will still be available to carry out measurements of signal strength and store them internally in the UE device. But, the full potential of the application is by placing this backend component inside the infrastructure and installing it as close to the UEs as possible. The competitive advantage of this approach is that the generated network traffic for the latency and the throughput tests will remain in an environment that is under control by the user. In that case much more credible results will be given, compared to the generated results by performing speed or latency tests with commercial apps, which use public endpoints and their traffic is directed through multiple hops over the Internet.

Figure 3 illustrates the maps that can be loaded within the app and how the measurements' data is visualised as the UE moves around a specific area. In the upcoming months effort will be put towards the following goals:

- To add the capability to generate bar-charts with the results of an experiment;
- To use Kibana together with ElasticSearch (see also 3.4.4) to generate graphs and visualisations with the experiments data (as already mentioned are also being kept inside the backend server);
- The application has been designed with the aim to have a clean and simple UI for the user when the UE is handheld, but there are also cases of specific trials that require the UEs to be attached to other devices (e.g., a flying drone) away from the user. This case creates the need for triggering specific tests remotely, without having to touch the device. Towards this goal, the UEs and the backend server will establish a bi-directional connection over Websockets and the experimenter will then be able to control the execution of tests remotely. This feature is already introduced and implemented, but it will bring more potential as the development of Latency & Throughput test mechanisms progresses.

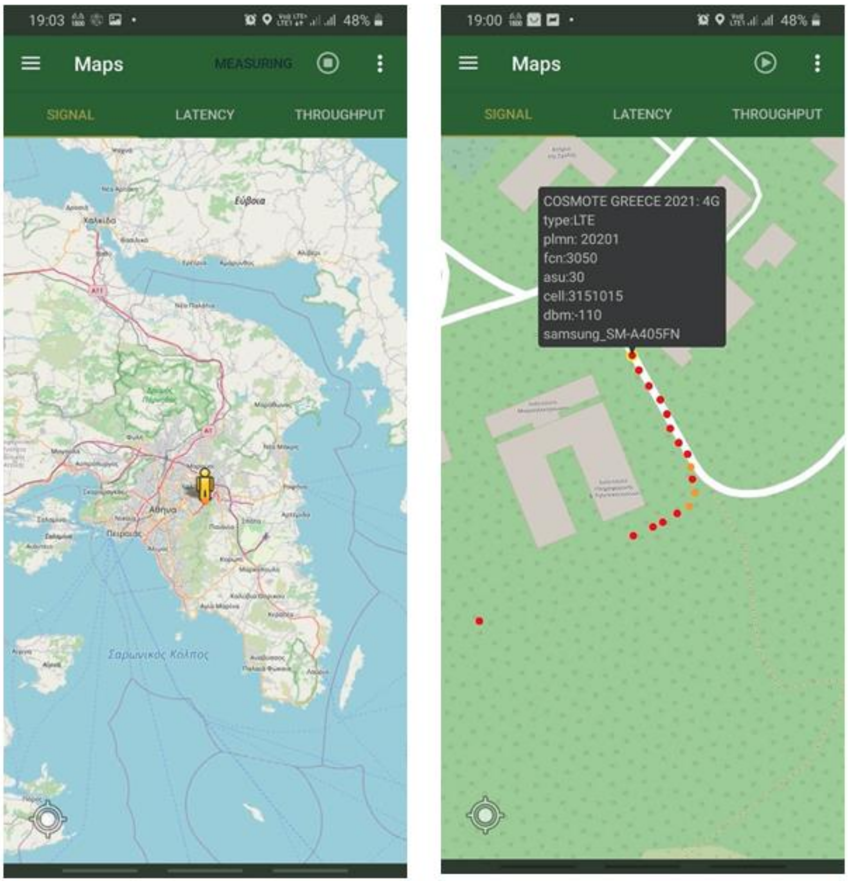

Figure 3: Cell Performance Monitoring Tool - Map

COSMOTools

The COSMOtools is an innovative, multi-purpose platform, aiming at network performance, mobility and customer experience (QoE) monitoring, in a dynamic, future proof, user-friendly, comprehensive and cost-efficient way. The COSMOtools constitutes a future proof platform, supporting the gathering and processing of bulk network measurements at a national level, in a cost-efficient way; involving OTE Group employees who could contribute by utilising their own (Android) smartphones. The main goals of the COSMOtools are achieved through:

- The depiction of access network related information (e.g., Cell-Id, RSSI/RSRP, SSRSRP/SSRSRQ/SSSINR, LAC/TAC, BS/eNBs/gNBs location/info – see [11] for network abbreviations) at mobile terminal and at desktop.
- The collection, depiction and uploading to a dedicated server of performance-related measurements such as, maximum upload bitrate, maximum download bitrate, latency, signal strength, for post processing and assessment.
- The identification of possible network issues (e.g., poor/no coverage, low throughput, high latency, unsuccessful handovers) via a graphical environment (WebGUI) even in real-time.

The COSMOtools platform is decomposed to client applications, server-side infrastructure and a graphical environment (WebGUIs), comprising a complete tool that can be utilised in multiple ways. More specifically, the COSMOtools platform components are the following:

1. Two (2) client applications running on Android devices, in “on-demand” mode, “on-event” mode or “periodically. These include:
   - The COSMO_QoE tool: It depicts 2G/3G/HSPA/HSPA+/4G/4G+/5G (and WiFi) network-related information (including BSs/eNBs/gNBs locations/capabilities/name, cell reselections locations/info, handover locations/info, etc.). On top of that, it collects, depicts and uploads –upon user demand-measurements related to quality that the end-user experiences (QoE), in terms of maximum upload bitrate, maximum download bitrate and latency, at the current user location (both indoors and outdoors) and while on the move. It uses Google Maps to present the gathered measurements to the user in real-time.
   - The COSMO_NetProbe5G tool: It is capable of measuring and uploading network performance related info (signal strength, latency, maximum download bitrate, maximum upload bitrate) periodically. The application may run as a background service.
2. Server-side infrastructure utilised to collect, store and process the related mobility/performance measurements.
3. A graphical environment (WebGUIs) with advanced filtering and presentation capabilities, enabling a complete network performance behaviour presentation, as well as the identification of existing or potential network design/performance issues (coverage gaps, low RSSI/bitrate areas, unwanted cell reselections, handover failures, etc.). A brief overview of the COSMOtools' features/capabilities is given below.

3.1.2 COSMO_QoE

The main features/capabilities of the COSMO_QoE tool are the following:

- Presentation of 2G/3G/HSPA/HSPA+/4G/4G+/5G (and WiFi) network related information (incl. BSs locations/capabilities/name, cell reselections/locations info, handover locations/info, etc.) in real time, over Google Maps.
- Collection, presentation and uploading (to dedicated servers) of QoE related measurements in real time, such as signal strength (RSSI, RSRP, SSRSRP, RSSNR SSSINR, RSRQ, SSRSRQ, etc.), latency, maximum download bitrate and maximum upload bitrate (in both numeric and graphical formats), both indoors and outdoors for both cellular and WiFi technologies.
- Coloured track/route (while on the move) depiction based on predefined measurement ranges (e.g., RSRP: from -85 to -100 dBm).
- Line connecting the user's current location with the serving BS; for easy identification of the location/name of the serving BS.
- Secure transfer of information between the terminal and the server/databases.

- Depiction on terminal screen (over Google) maps of past measurements.
- Push notifications (logging started, measurements were uploaded successfully, etc.).
- In-app application version update and BSs information update.
- Depiction of collected measurements via a user-friendly web environment (WebGUI).

Indicative snapshots of the COSMO QoE tool interface and measurements (at smartphone) are illustrated in Figure 4, Figure 5 and Figure 6.

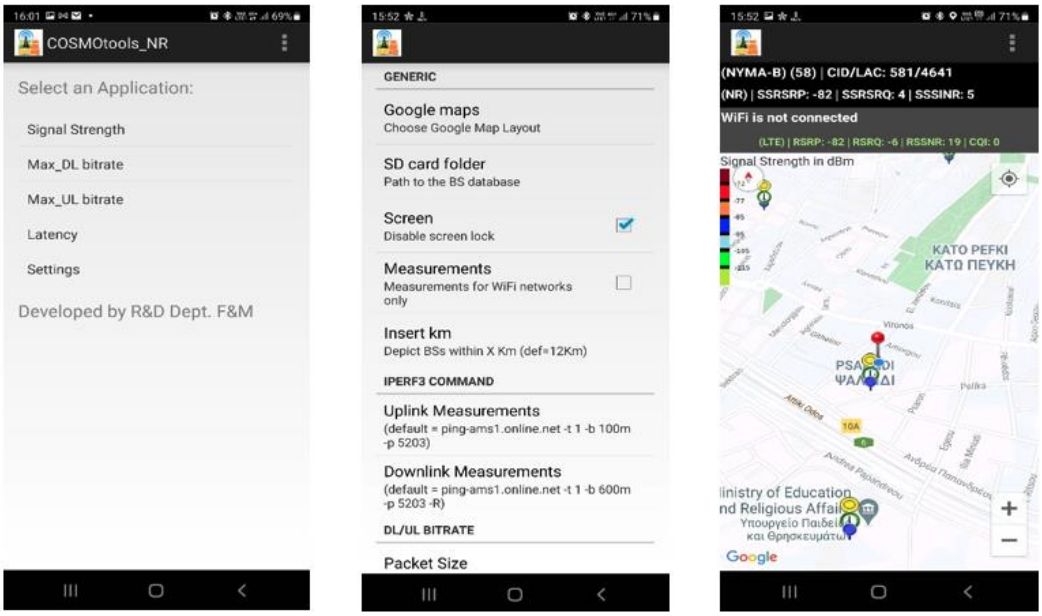

Figure 4: Cosmo QoE Tool - Example

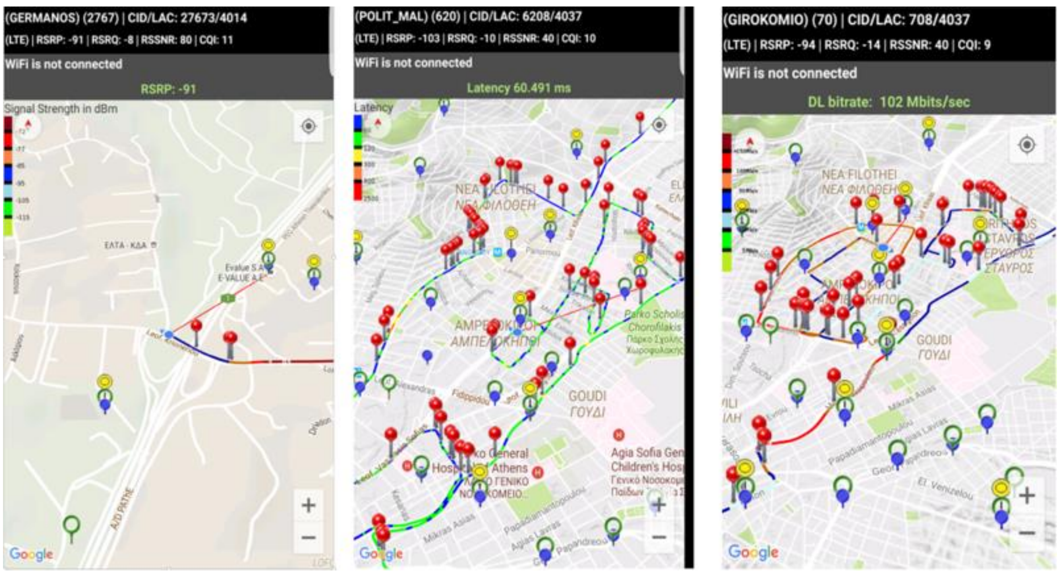

Figure 5: Cosmo QoE Tool - Map

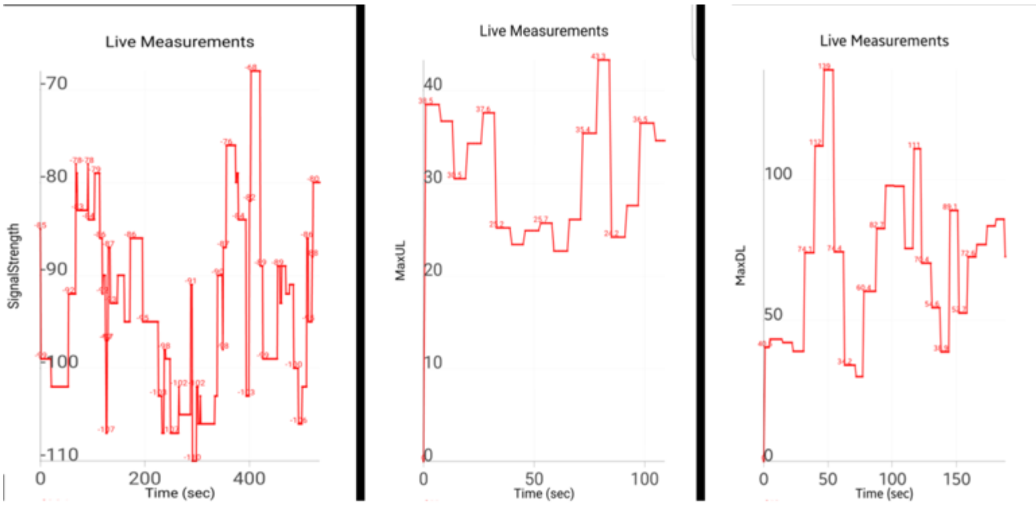

Figure 6: Cosmo QoE Tool - Measurements

COSMO_NetProbe5G

The aim of this tool is to (automatically) collect and upload network performance related info, such as signal strength (RSSI, RSRP, SSRSRP, RSSNR, SSRSSNR, SSSINR, RSRQ, SSRSRQ, CQI - [11]), latency, maximum download bitrate and maximum upload bitrate periodically (e.g., every X minutes). Indicative performance measurements are depicted in Figure 7. The COSMO_NetProbe5G App serves as "real-time" network probes, offering the user/MNO immediate info on network performance e.g., in cases of Self-Organised Networks, network changes, etc. The data gathering can be also visualised via a WebGUI. The application could be: 1) deployed by COSMOTE (or other NatCO), on own terminals distributed at specific locations -terminal operation could be remotely controlled and/or 2) offered as a commercial app (running in background- no mobile UI is required in this case).

Figure 7: Cosmo NetProbe

3.1.2.1 WebGUI

Through the Graphical Environment (under development), as depicted in Figure 8, the user may select to depict:

1) All Measurements (signal strength or latency or Max. upload and download bitrates);
2) Measurements for a specific range(s) (e.g., -66 … -75dBm);
3) Measurements collected for a specific time period (from date … to date);
4) Measurements collected for a specific IMEI/terminal/route;
5) The Serving BS at each measurement location ("linked-line" between measurement and Serving BS);
6) Details on each measurement collected (e.g., serving BS, measured value, Cell-ID, LAC, date/time, coordinates);
7) Cell reselections, handovers, no 3G/2G coverage locations (+info);
8) 2G/3G/4G BS/eNB locations & relevant info (Site name, SiteID, Coordinates, HSPA capability, etc.);
9) 3G Site, Cell-id, LAC specific "coverage";
10) Statistics e.g., #samples/range and #samples/technology, etc.

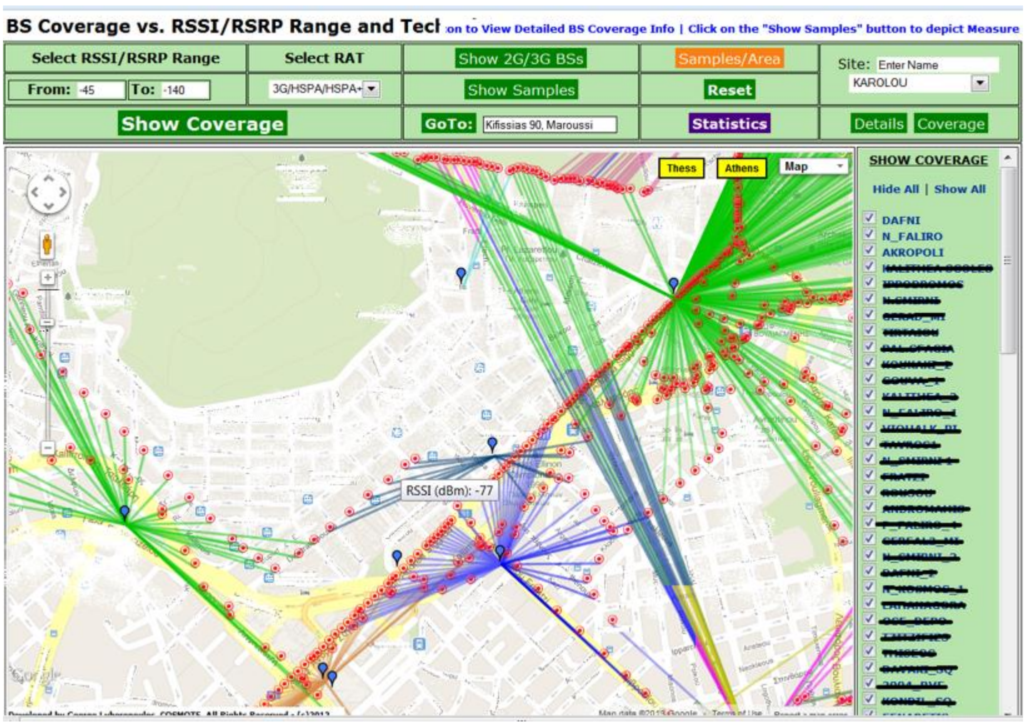

Figure 8: Cosmo QoE Webview

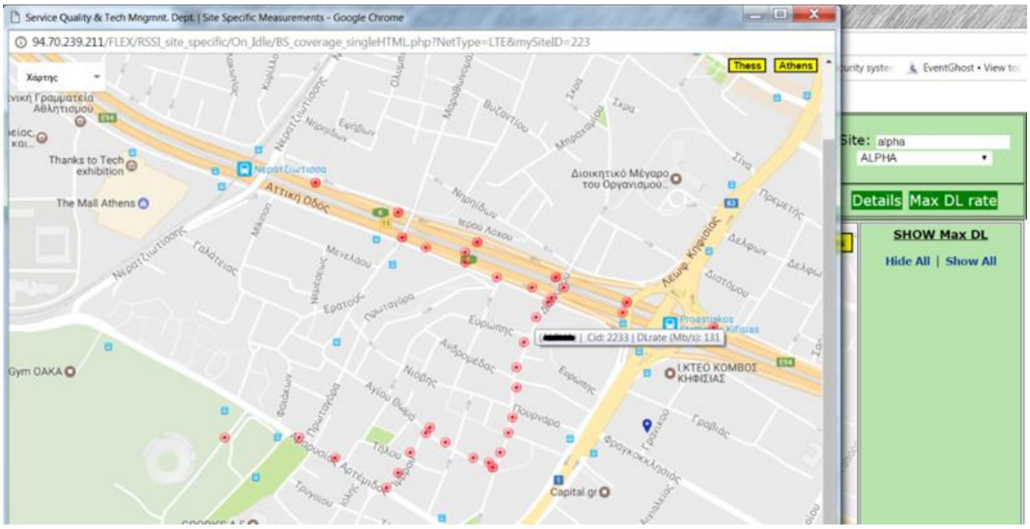

Figure 9: Cosmo QoE Map View

3.1.3 Infrastructure collector tools

To tackle the analysis of health and performance of the 5GENESIS Facility and platforms during the experiment executions, a variety of Analytics methods has been implemented. such methods have been developed as micro services, using Docker containers.

Figure 10 gives an overview of the implemented containers and their functions, including data handling, KPI correlation, KPI prediction, statistical analysis, feature selection and visualisation. A graphical user interface provides easy access to the different features. These features include anomaly detection and correlation analysis for health monitoring purposes (i.e., is the experiment doing what it is supposed to do?), and KPI prediction for performance analysis (e.g., to support potential performance improvement of certain network elements).

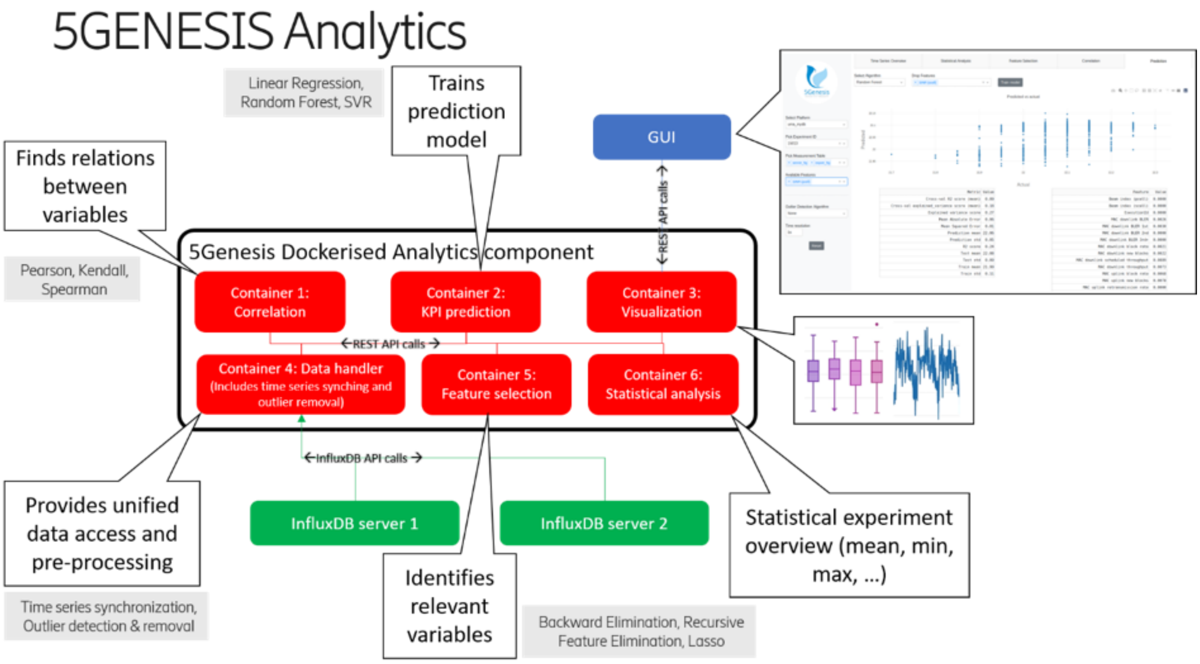

Figure 10: 5Genesis components

The following sections describe the algorithms and services in detail.

Visualisation Service

The visualisation service allows for visual representation of the results from the other services in an interactive fashion. The user can access this visualisation through a graphical user interface (GUI) that runs in the browser. After an experiment is run and its data is stored in the InfluxDB database, the user can browse and analyse the experiment results through the GUI shown in Figure 11. The user can select which experiment results to display on the left-hand side of the GUI, using the parameters of the experiment query. The main part of the interface is a tabbed environment, where the user can switch between the different Analytics services. From left to right, the tabs contain 1) a time series overview window, 2) a statistical analysis window, 3) a correlation window, 4) a feature selection window and 5) a KPI prediction window as described below.

Figure 11: Time Series overview window

After selecting the desired experiment and KPI(s), the time series overview window (Figure 11: Time Series overview window) displays a graph with the selected KPI(s). When multiple KPIs are selected for comparison, they are added to the plot with different colours and y-axis scales, as exemplified below.

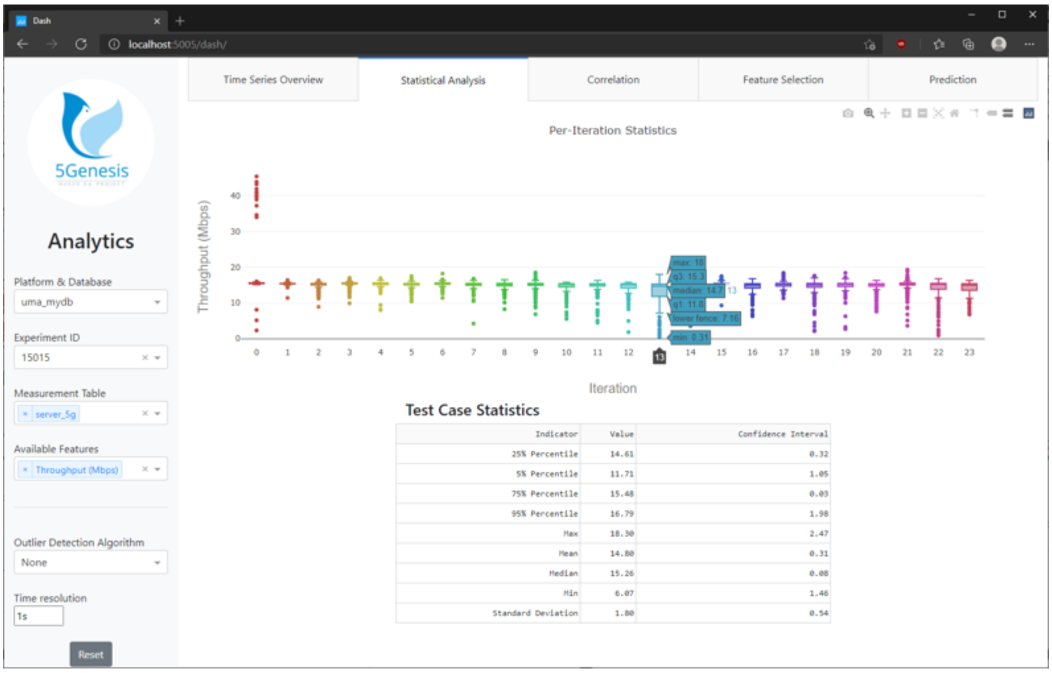

Figure 12: Statistical analysis window

The statistical analysis window shown in Figure 12: Statistical analysis window displays various statistical metrics for each iteration of the experiment, as well as agglomerated results on the entire experiment, following the 5GENESIS procedure for KPI validation. On mouse-over, the graph will show details for each box plot representing each iteration.

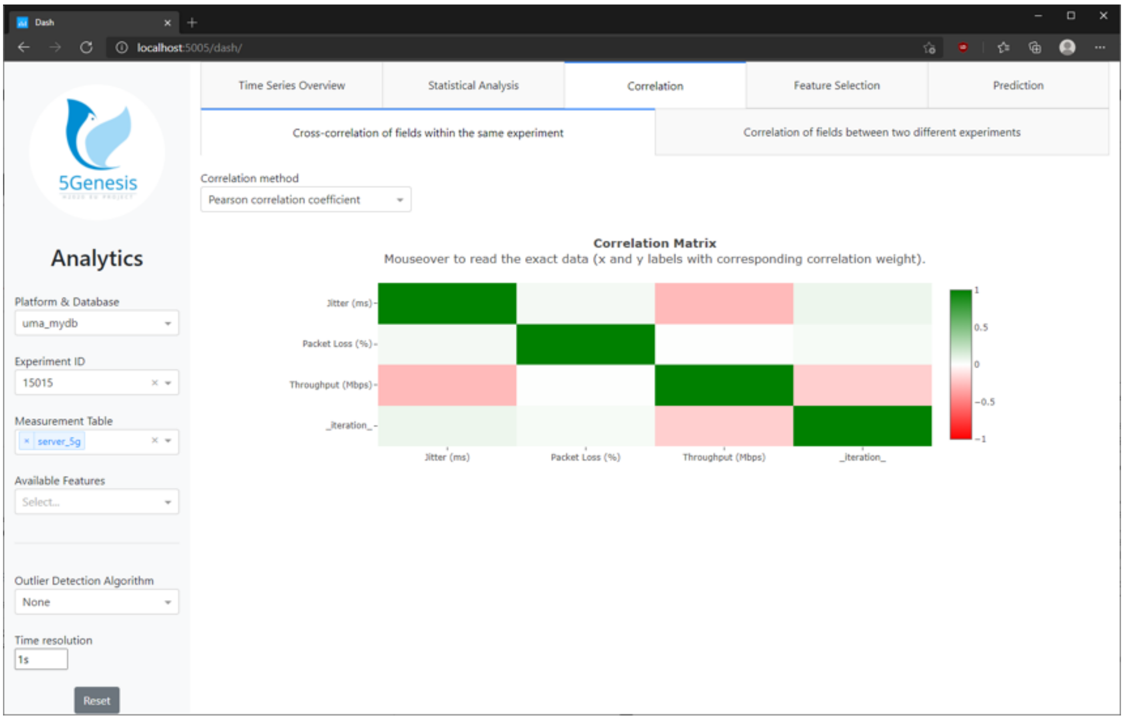

Figure 13: Correlation window showing the cross-correlation of fields

In the correlation window, the user can select between two types of correlation. The intra-experiment cross-correlation as presented in Figure 13: Correlation window showing the cross-correlation of fields, offers a correlation matrix of all numerical fields in the selected experiment.

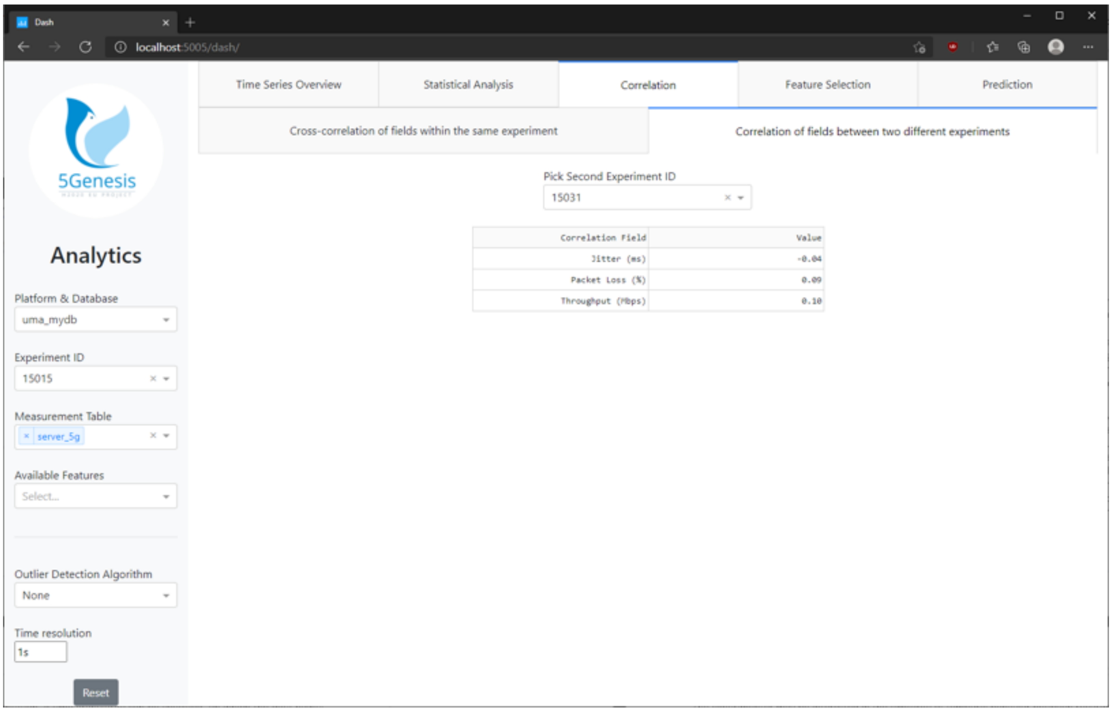

Figure 14: Correlation window showing the inter-experiment field correlation

The inter-experiment field correlation window shown in Figure 14: Correlation window showing the inter-experiment field correlation lets the user select a second experiment from a dropdown menu. All numerical fields in both experiments (the originally selected one and the added one) are then displayed in a table format.

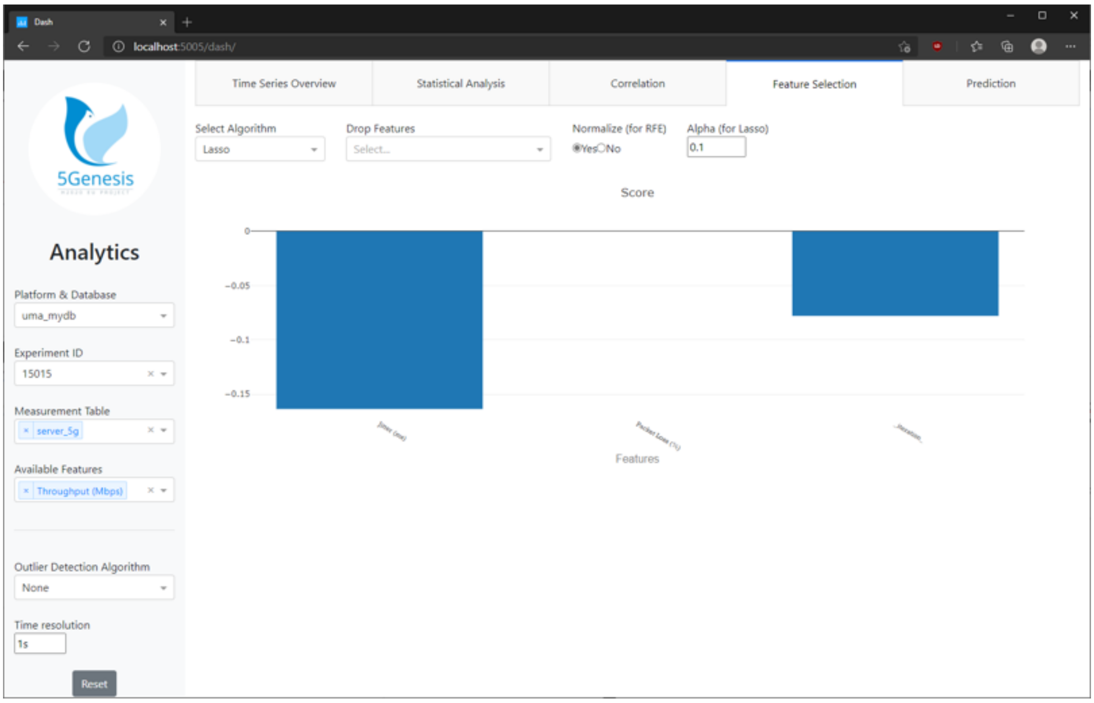

Figure 15: Feature selection window

The feature selection window shown in Figure 15: Feature selection window highlights the result of the selected feature selection algorithm, for which additional parameters can be specified and some features may be excluded from the feature selection.

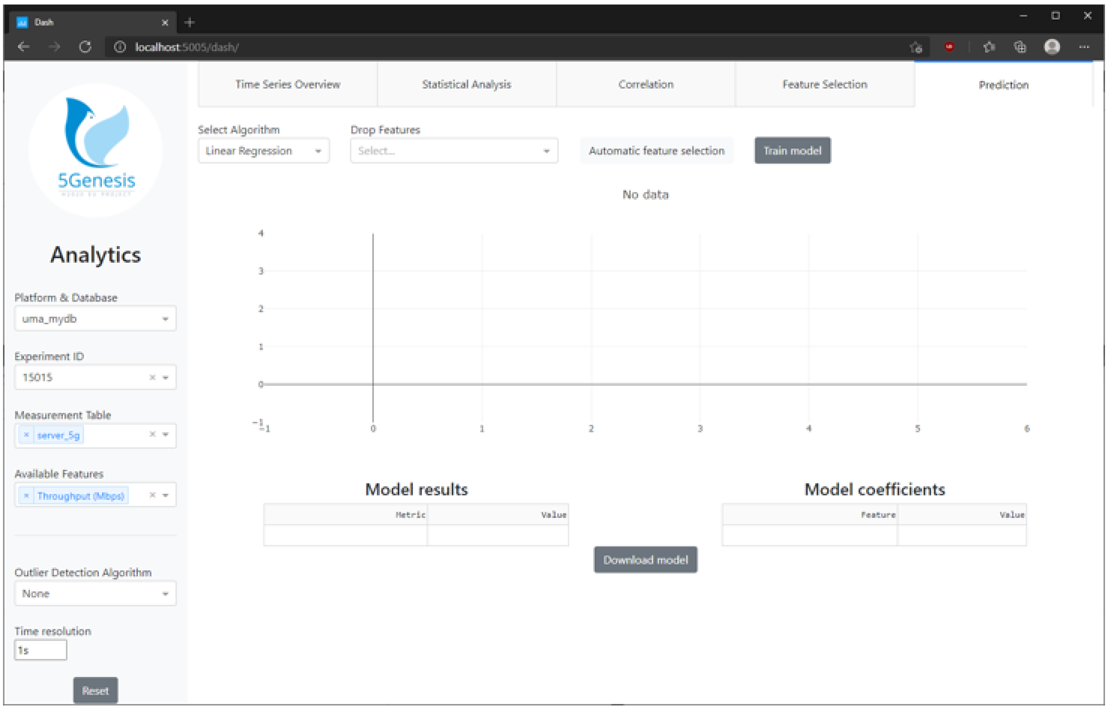

Figure 16: KPI prediction window

The last window displays the KPI model training (Figure 16: KPI prediction window). The user can select the ML algorithm to be used for the training and exclude features from the training. A button allows for automatic feature selection by calling the Feature Selection service in the background. After the "Train model" button has been clicked and the training concluded, the training results will be displayed, including a table with several error rate metrics, a scatter plot that contrasts actual vs predicted values, and a table the shows the model coefficients (in the case of the linear regression) or the feature importance values (in the case of the Random Forest or SVR algorithms).

For more visualisation topics please also see topics within chapter 4 DATA VISUALISATION & ANALYSIS.

3.1.4 Scenario specific tools

Some trials require additional tools to collect datasets specific to the individual scenario. Exemplary for that Use case 3 – Scenario 1 – sub-scenario 2 - long range power line analysis trials - is described in this chapter.

Due to weight and power constraints, UAV on board computers (OBC) are usually quite resource restricted. Hepta is the scenario leader for this scenario.

The drone's on-board computer has multiple parallel tasks:

1) Combining C2 and telemetry streams connecting multiple command and telemetry links with a single flight controller;
2) Collecting and forwarding lidar data;
3) Collecting and forwarding images from the camera.

In the future, real-time object detection and decision making are also envisaged to take place to ensure optimal data capture. To test and understand the limitations of such setups, it is important to be able to monitor and analyse OBC resource utilisation. Hepta is planning to use Promtail in the OBC to send generic logs to Loki situated in an AWS EC2 instance. All the OS metrics are grabbed by Telegraf which will be sent to InfluxDb also hosted in an AWS EC2 Instance. Both Promtail and Telegraf are compatible with ARM architecture. For visualisation and analysis, Grafana will be used. Grafana can visualise data from many different data sources e.g. Loki and InfluxDb. From Loki and InfluxDB all the metrics and data from logs can easily be queried and sent to any 3rd party.

3.1.5 Monitoring and Measurement of KPIs in 5GTN

3.1.5.1 Qosium Tool

Qosium is a real-time passive software-based solution for monitoring and measurement Quality of Service (QoS), traffic load, and Quality of Experience (QoE) without generating additional traffic [12] [13]. Qosium is a distributed measurement system composed of three components: 1) the measurement agent (Qosium Probe), 2) measurement controller (Qosium Scope), 3) and measurement results database system (Qosium Storage).

1) Qosium Scope is the main controlling entity software responsible to implement the QoS Measurement Control Protocol (QMCP) and control measurements in a manual manner [13]. It is a full-scale analyser software, which commands and controls Qosium Probes and their measurements, and depicts received results. Qosium Scope is a key enabler for manual QoS, traffic load, and QoE measurements.
2) The Qosium Probe is a lightweight measurement agent, which is installed to network devices from which traffic, QoS, and QoE are desired to be measured [13]. Network devices can be based on Linux, Android, or Windows systems to use Qosium Probes. Importantly, the Qosium Probe installed on Linux-based systems enables measurements of the latency-related KPIs (e.g., delay, jitter). The installed Probes can wait for the potential measurement points since it only listens for QMCP connection trials. Therefore, the Probes could be running all the time in the nodes where measurements can run eventually or constantly. The Precision Time Protocol (PTP) is a key enabler to attain better clock synchronisation accuracy among the measured network devices. Qosium Probe is a multi-thread system that can be involved in several measurement instances [13].
3) Qosium Storage is a dedicated database system that stores average results of measurements collected by the Qosium Probes [13]. Qosium Storage is especially useful when there are a lot of data results, and accessing the data is wanted to be centralised. Storage comes up with visualisation and access to measurements via its REST interface.

Measurements Results:

The basic results of Qosium provide QoS metrics, which tell how well the network is performing from the perspective of the monitored application(s)/service(s) [13]. Traffic statistics, flow statistics, radio level statistics and events, and QoE measurements are KPIs that can be obtained by Qosium. The results are monitored as averaged values over the user-defined Averaging interval [13]. The Qosium results come with a single timestamp used to organise and read the average statistics results. The results can be associated with the location to visualise in several types such as graphs, meters, and numeric values, and also draw heatmaps from any Qosium result.

Qosium can generate four categories of results:

- Average results - Average results are averaged over Averaging Interval, which is a parameter that can be defined in the user interface of Qosium Scope [13]. The average results comprise the most important statistic set with over 60 specific statistics to be evaluated.
- Packet results - Packet results provide accurate QoS statistics for every single measured data packet in a two-measurement points scenario (i.e., among two machines).

- Flow results - Flow results provide a simple view of who is communicating with who and how much traffic there is flowing [13]. It is able the flows detection of high bandwidth usage.
- Pcap results - These results provide optionally full packet captures of desired traffic in the Pcap format, allowing to analysis details with protocol analysers e.g., Wireshark [13].

Most of the Qosium's statistics are available for received (downlink) and sent (uplink) directions [12]. Hence, the average results are collected in both directions, meaning that sent and received traffic can be evaluated separately. The results can be averaged or evaluated accurately per packet. Among the most important average results, there are:

QoS statistics:

- Packet loss
- Delay (Latency)
- Jitter (delay variation)
- Connection break statistics

Traffic statistics:

- Throughput (bits/s)
- Traffic Load (in packets, bits, and bytes)
- Volume of data
- Packet sizes

Flow statistics:

- Traffic flows traversing measurement node
- Per-flow: load, duration, transmitted data, etc.

QoE (pseudo-subjective analysis):

- VoIP and video quality with PSQA in the MOS scale
- Generic QoE algorithm, which allows users to define their own QoS threshold values

Location-related statistics:

- Latitude
- Longitude
- Altitude
- Accuracy
- Heading
- Speed

Radio level statistics:

- RSSI
- SINR
- Cell ID
- Network Type
- BS MAC Address

Results Analysis:

Qosium generates statistics that can be evaluated in real-time, e.g., with Qosium Scope. The results can also be printed into files for later analysis. The four types of results files generated by Qosium Scope are in TXT and CSV formats, allowing import to external statistical tools (e.g., TensorFlow [14], Keras [15], Excel [16] , and Scikit-learn [17], PyTorch [18], etc). Importantly, the average statistics results generated during the measurement scenarios can be analysed using the REST interface of the Qosium Storage.

For analysis and troubleshooting situations, the Qosium Scope interface show measurement results in meters, plots, bars, and numerical formats. Depicted results are averaged over the desired time range. However, from average results, it is possible to drill into packet results where QoS results for every single packet are shown.

Plots depict how results have developed over time. Meters, on the other hand, depicts the current situation of measurements [12]. Results like delay, jitter, traffic load, packet loss, and over 20 other measures are measured and depicted always in both directions (i.e., received and sent traffic).

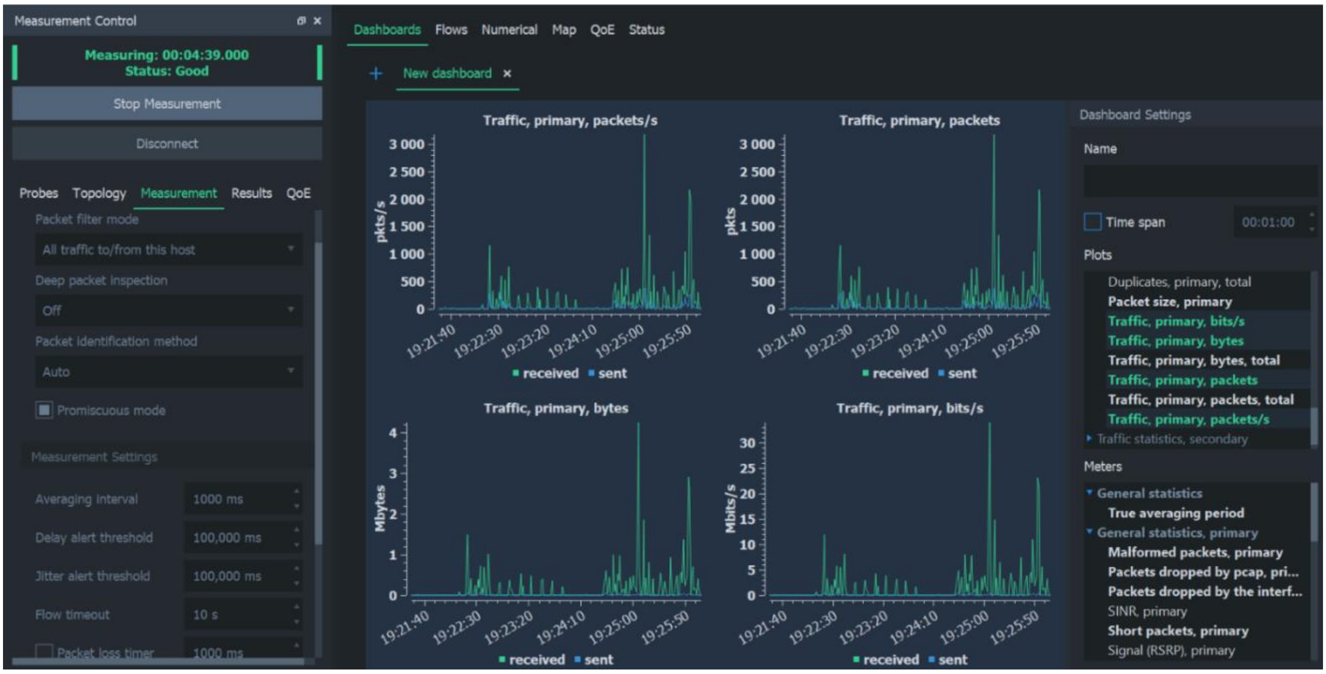

Figure 17: Dashboard Results - Plots - Qosium Tool

The graphical forms of KPI measurements can be customised in the dashboard, mainly among Plots and Meters as depicted Figure 17 and Figure 18.

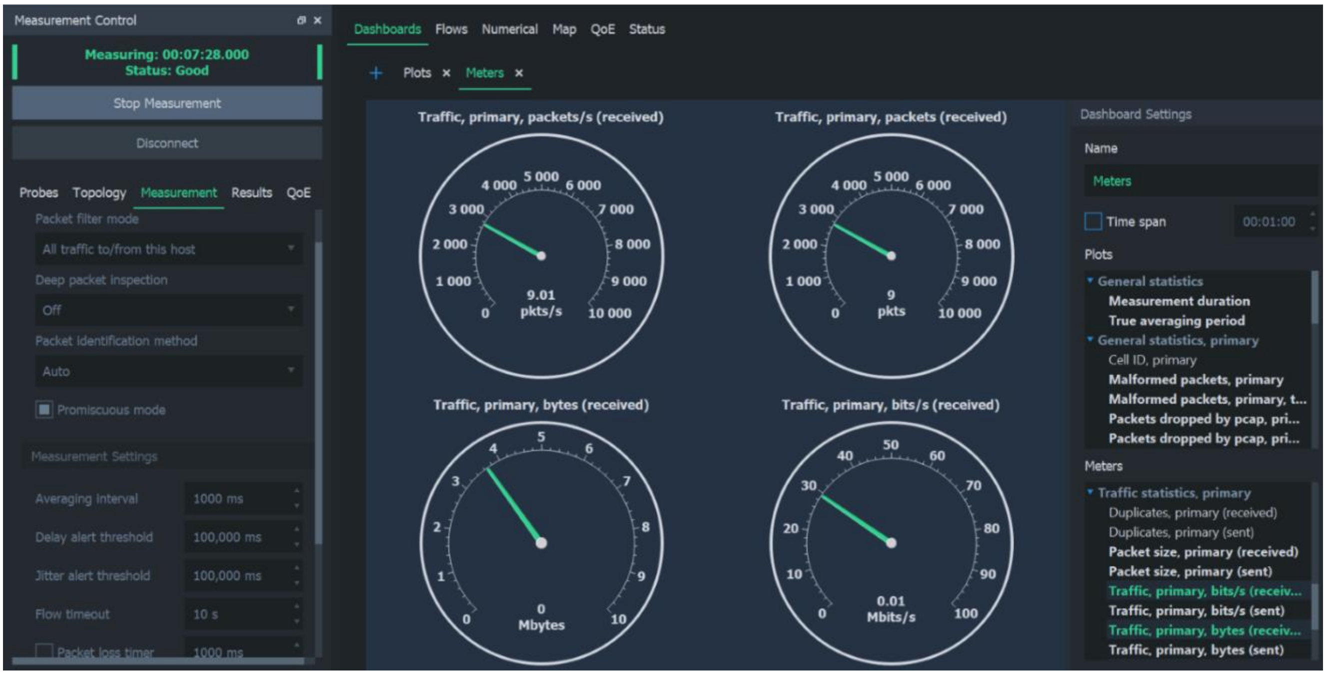

Figure 18: Dashboard Results - Meters - Qosium Tool

The QoS heatmap is one of the most interesting features of Qosium Scope, useful to analyse measurements of QoS or even QoE of a real geographical area. If Qosium Probe has location information during a measurement, Qosium Scope depicts any measure (e.g., delay) on a map even in real-time. Qosium Scope enables to find out the coverage area of a wireless network, not in terms of signal strength but terms of the actual quality experienced by end-users.

It is possible to also bring specific overlay images to the map of Qosium Scope. The image can be, for example, a building layout for indoor wireless measurements. Satellite positioning can also be given by using some positioning techniques.

Using Qosium Scope, heatmap visualisation can be parametrised to fulfil specific requirements. For example, the Figure 19 includes ruling colour coding, drawing precision, and map style parameters. The different colours represent signal strength coming from the wireless device where Green is excellent coverage, Yellow and orange represent the weaker but connectable signal, and Red means very poor signal strength.

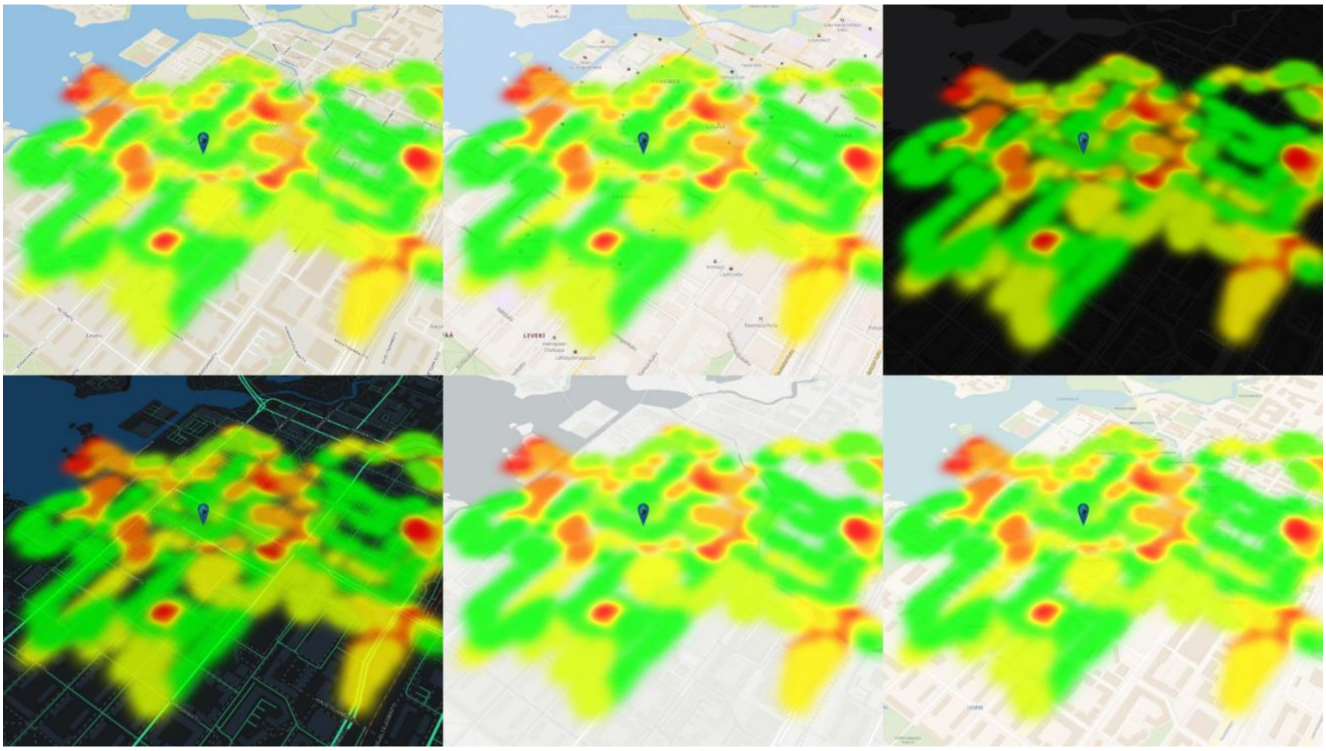

Figure 19: Heatmap Results [12] - Qosium Tool

In addition to QoS results, Qosium Probes extract overall network traffic to traffic flows. Flow analysis in Qosium Scope shows which network devices communicate with whom while the Qosium Probe is running. It is enabled to identify if interesting applications are traversing the Qosium Probe point.

Additional data analysis topics are also described within chapter 4 DATA VISUALISATION & ANALYSIS.

Storing the Results:

All KPI measurement results can be saved to a text or CSV file, which can later be loaded back to Qosium Scope or taken quickly to other analysis tools. On the other hand, the results of average statistics specifically are sent in real-time to the Qosium Storage database. In the Figure 20, the REST interface of Qosium Storage database list the measurement results for the specific time range, Service ID, User ID, Probe, Measurement Description and a Download option of a text format file with the average results.

#	Measurement	Stats	Started	Ended	Service ID	User ID	Probe	Description	Download
10	10-0-1618856263	0.0%	21:17 19.04.2021	21:30 19.04.2021	10	0	193.166.28.155	OpenStackServer↔EPCinstance-test13	Download
9	10-0-1618856150	0.0%	21:15 19.04.2021	21:29 19.04.2021	10	0	193.166.28.155	CentralMeas↔EPCinstance-test12	Download
8	10-0-1618854588	0.0%	20:49 19.04.2021	21:11 19.04.2021	10	0	193.166.28.155	5gtn-test11	Download
7	10-0-1618853890	0.0%	20:38 19.04.2021	21:16 19.04.2021	10	0	193.166.28.155	test1	Download
6	10-0-1618597407	100%	21:23 16.04.2021	21:33 16.04.2021	10	0	10.38.151.35	5gtn-test10	Download
5	10-0-1618595351	100%	20:49 16.04.2021	20:52 16.04.2021	10	0	10.38.151.35	5gtn-test9	Download
4	10-0-1618416615	100%	19:10 14.04.2021	21:28 14.04.2021	10	0	130.231.201.223	5gtn-test8	Download
3	10-0-1618406070	100%	16:14 14.04.2021	17:48 14.04.2021	10	0	130.231.201.223	5gtn-test7	Download
2	10-0-1617981910	10%	18:25 09.04.2021	19:21 09.04.2021	10	0	255.255.255.255	5gtn-test5	Download
1	10-0-1617971472	8.0%	15:31 09.04.2021	16:48 09.04.2021	10	0	255.255.255.255	unnamed	Download

Figure 20: Measurement Results List - Qosium Storage

The results of average statistics from Qosium Storage database become later in the data source for the 5G!Drones system, in specific for the Trial Controller. To achieve that, the Abstraction Layer by means of its component called Measurement Job Controller collects required KPIs in real-time from the Qosium Storage database to then provide such data in the same way to the Trial Controller.

In accordance with the measurement results located at the Qosium Scope and Qosium Storage, new mechanisms and algorithms can be proposed to validate 5G KPIs. In this sense, the Qosium tool capabilities to monitor and measure KPIs are aligned to the 5G!Drones project objective to validate the 5G network KPIs for UAV use cases.

3.1.6 Monitoring and measurement of KPIs in X-Network

X-Network makes use of open source and home-made tools for monitoring of KPIs. Two levels are distinguished: 1) deployed services which are at the virtualised resources level and 2) the QoS at the UE side.

The importance of monitoring the running services and keeping track of resources imposes using powerful tools working in a sophisticated way to guarantee the real time tracking without affecting the existing deployed system. In order to monitor resources and the deployed services, X-Network makes use of Prometheus, it comes with the capability of:

- Scraping data from deployment (pods).
- Scraping data from nodes, it uses a node exporter to retrieve information about the node resources.
- Pushing metrics from jobs which cannot be scraped, by dint of a Push Gateway.
- Defining rules for watching resources usage.
- Executing actions based on rules pre-defined (ex: send an email whenever a pod uses more than 2GB of memory).
- Using a database to store the scraped metrics.

Using this tool, a metric collection framework has been defined and implemented, which is fully compliant with the 5G!Drones system architecture and its abstraction layer (Jobs can be created to collected metrics). The architecture of metric collection framework is depicted in Figure 21.

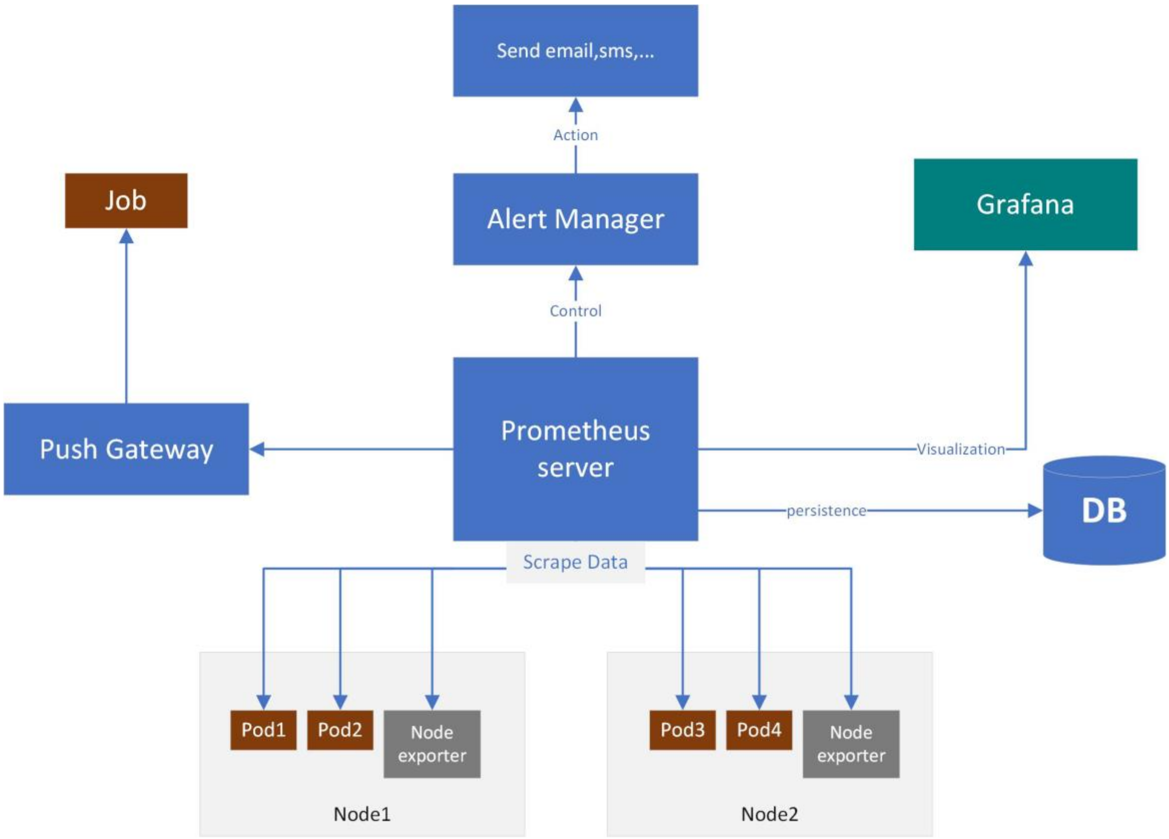

Figure 21: Metric collection architecture at X-Network

On the other hand, X-Network also makes use of home-made software for collecting QoS from the UE side. The collected metrics include RSSI, delay and throughput. X-Network is currently in the process of integrating this software with the measurement job logic.

3.2 Data Types

This chapter gives an exemplary overview of the different types of data that are collected and analysed during the trials. Each trial scenario may have additional, very use case specific data items, a specific scenario example is given in

In general, data types can be separated into:

A) Telemetry data about drone flight. It contains data on the drone's flight route, speed, velocity, waypoints, activities etc.

B) Payload data provided by onboard sensors and cameras.

While the telemetry data is very often in a standard format (e.g. Mavlink etc.), the payload sensor formats depend on the sensors and hardware used. One type of payload data is also the result of measuring the quality of the 5G network (latency, signal strengths, bandwidth, etc.) using the 5G UE (5G smartphone or 5G communication module) on board the drone. Photos are usually in .jpg or .png format, but the video format depends a lot on which compression technology to use (for example, Motion-JPEG etc.).

UAV Data Sources

UAV related data may be provided by very different means, mostly via “established” ATM related mechanisms or via very drone specific approaches.
INVOLI is providing traffic data information for manned aviation (different transponder types) and cooperative drones with INVOLI’s tracking drone hook-on-device, known as LEMAN.

For manned aviation, we collect and present data from different transponders: ADS-B, Mode S, Mode A/C and Flarm. **Error! Reference source not found.** shows which information is received from different d ata sources. Each type of transponder has its own set of data - i.e., for Mode A/C the aircraft position is not included, but INVOLI is able to calculate it, based on time of arrival of radio signal to several our receivers spread on the ground. Using the same method, INVOLI is also capable to validate the position of aircraft sent by transponders.

Table 1 Data types availability for different transponders types

	ADS-B	Mode S	Mode A/C	Flarm	LEMAN tracker
ICAO Address	Y	Y	N	Y/N	N
SQUAWK	Y	Y	Y	N	N
Call Sign	Y	N	N	N	N
Position message	Y	N	N	Y	Y
Position MLAT	Y	Y	Y	N	N
Altitude	Barometric, GNSS based	Barometric	Barometric	GNSS based	GNSS based
Speed	Y	N	N	Y	Y
Heading	Y	N	N	Y	Y
Climb speed	Y	N	N	Y	N
Signal strength	Y	Y	Y	N	Y

ADS-B system is very vast and carries much more information than specified in the table above, but INVOLI selected the set of data, which is the most important for traffic awareness and make it available for its output stream.

This data can be sent during mission from INVOLI central server and is available for 5G!Drones processing, storage and analysis.

In addition, it is possible to create and use the alarms for the situations, when two airborne objects are approaching too close each other.

3.2.1 Data Type Example – Use case 3 – Scenario 1 – SubScenario 2

This chapter provides an example of the scenario specific setup for the "Long range power line inspection".

a) Telemetry data is sent to the edge server using Mavlink(a communication protocol for unmanned vehicles, [19]), in the edge server the Mavlink data stream is forwarded to the operator's ground control station where it is being recorded into telemetry log files. From the edge server, a subset of telemetry information is sent to UTM via an API. An additional dataset is also logged to the SD card of the flight controller of Hetpa's unmanned helicopter in binary format as a .bin file.

b) Images - Hepta is utilising a camera and a LiDAR (Light Detection and Ranging) sensor for data collection. Camera is triggered to capture the images at defined locations during the mission. The images are then saved onto the onboard computer in a JPG format. The image files contain a timestamp, geographical coordinates, coordinate reference system information and gimbal angles information in Exif data to help visualise the image location and orientation on the map in Hepta's online infrastructure inspection software uBird.

c) Point cloud – Velodyne VLP-16 LiDAR is used to collect data for 3D mapping. The VLP-16 LiDAR sends data to the on-board computer through the Ethernet port as User Datagram Protocol (UDP) packets. The data itself consists of firing sequence, Laser ID, measured distances, azimuth angle, reflectivity of an object, model of sensor, laser return mode and timestamp. Further on, the measurement data is being converted from Spherical to Cartesian coordinates. The resulting point cloud is stored as a ROS message of type sensor_msgs/PointCloud2.

d) On-board computer (OBC) – System log files and metrics are recorded from the Ubuntu 18.04 OS running on the OBC. Standard Linux log files are read by Promtail and stored in Loki as chunks. Also, Telegraf sends metrics outputs in InfluxDB Line protocol format to InfluxDB.

e) Atmospheric weather conditions – visibility, precipitation and significant weather phenomena if any. Logged manually in text format, once per trial.

3.3 Simulated data

In addition to collecting data from real-field experiments data can also be collected using simulation. This can be considered when the target experiment cannot be performed (e.g., network is not operational, facility is not accessible, etc.) or when there is a need to reproduce and repeat experiments in a controllable environment. Simulated data is also important to evaluate experiments and network behaviour at a large scale. This subsection emphasises with collected data based on simulation.

3.3.1 QoS of UAVs connected to the mobile network

The quality of service perceived by UAVs connected to cellular network depends on several parameters. Real-field trials have demonstrated that a UAV can be subjection to interferences from the non-serving BS. This is a direct consequence of the close free space propagation characterising the aerial communication. Therefore, in order to collect simulated data that accurately reflects a real experiment, it is highly important to consider a realistic communication model. Wireless signal propagation is characterised by several propagation phenomenon, such as path loss, fast fading, and interference. A general system model for cellular UAVs is depicted in Figure 22.

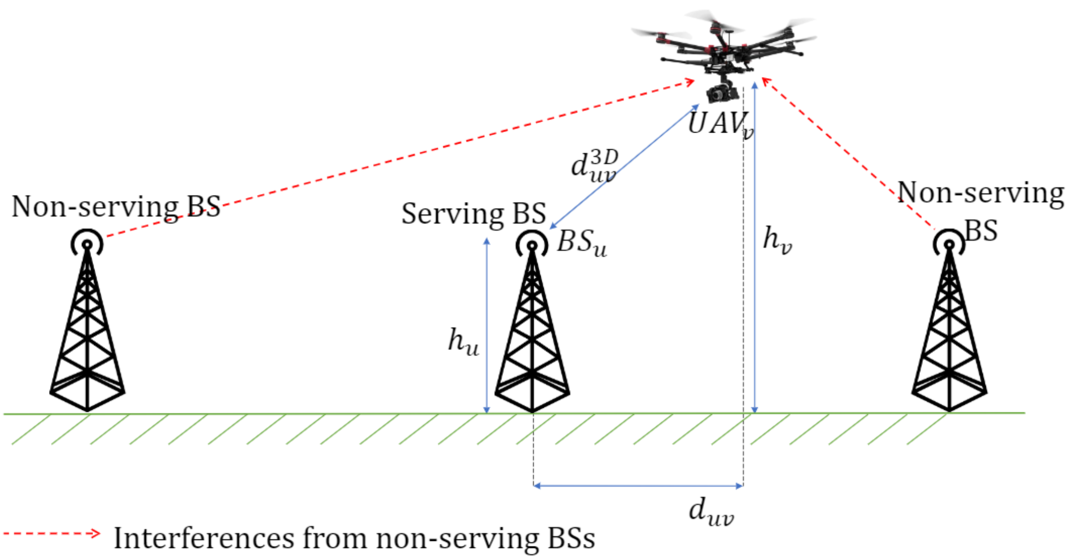

Figure 22: Modelling cellular communication for a UAV

The BSs employ an orthogonal frequency division multiple access (OFDMA) technique to serve the connected users. Consequently, intra-cell interference is neglected and interference can be caused only by non-serving BSs, as shown in the following figure. Let $\mathcal{V}$ and $\mathcal{B}$ denote respectively the set of UAVs and BSs. Let us also denote by u the serving BS and by v the receiving UAV. We also denote by $\mathcal{S}$ the set of sub-carriers used by the BSs. As a quality of service metric, we consider the outage probability defined as the probability that a UAV v fails in receiving packets from its serving BS u, and can be expressed as

$$P_{out,uv}^{UAV}(\gamma_{th}) = \sum_{j=1}^{m}\left(\beta_{1j}\frac{(-1)^j}{(j-1)!}\left(\frac{m}{A_{uv}}\right)^{-j}\left(\Gamma(j)+\sum_{t=1}^{N}\alpha'_t f_{j,1}(B_{tv}) - \sum_{t=1}^{N}\sum_{j'=1}^{m}\frac{\alpha_{t,j'}(-1)^{j'}}{(j'-1)!}f_{j,j'}(A_{tv}/m)\right)\right)$$
$$-\beta_{21}B_{uv}\left(1+\exp\left(-\frac{\gamma_{th}}{B_{uv}}\right)\left(\sum_{t=1}^{N}\frac{\alpha'_t}{\frac{\gamma_{th}}{B_{uv}}+\frac{1}{B_{tv}}} - \sum_{t=1}^{N}\sum_{j=1}^{m}\frac{\alpha_{t,j}}{\left(\frac{\gamma_{th}}{B_{uv}}+\frac{m}{A_{tv}}\right)^j}\frac{(-1)^j}{(j-1)!}\Gamma(j)\right)\right)$$

More details on the outage probability expression can be found in [19]. This expression has been derived taking into consideration most of propagation phenomenon known in wireless signals. This makes it more realistic. As depicted in Figure 23, simulating the QoS of UAVs is based on the parameters of the BSs (i.e., their positions, the used subcarriers and the employed transmission powers), the UAVs (i.e., their position, and their cell association), and also on the channel parameters.

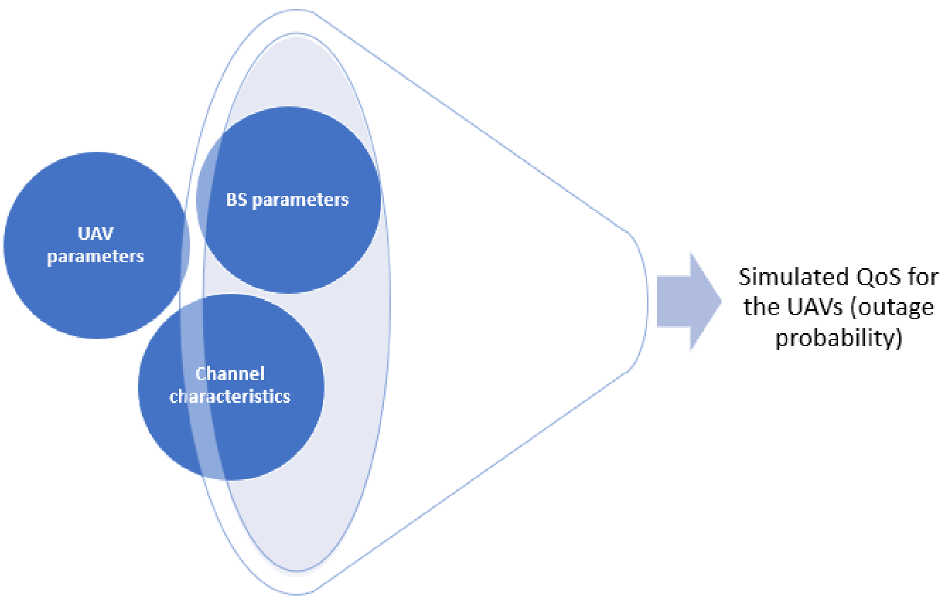

Figure 23: Parameters considered for simulating UAVs' QoS

3.3.2 Simulation Testbed

In the context of 5G!Drones Project, UMS team is currently building a hybrid simulation testbed (HITL+SITL) to remotely integrate and test APIs without the need to conduct physical drone flights thus reducing the need for travel during the present COVID19 times. It can further be used to measure some pre-defined KPIs over 5G networks. A set of KPIs that could be measured within the scope of the relevant use cases will be decided in due course after an initial validation of the testbed.

Purpose

The simulation testbed aims to implement interfaces and mechanisms by which simulated data from drones can be injected via APIs into other systems including UTM. A simulation platform will be used to enrich select use cases by utilising simulated UAVs to be able to demonstrate more complex scenarios and as a testing tool prior to live demonstrations.

The simulated UAVs will appear identical to real targets in the system. This objective is to provide data that is as close to a real flight as possible i.e., it should be indistinguishable from data that UMS would obtain with a physical UAV.

More specifically, for the use cases UMS is involved in i.e., UC2 S2 and UC4 S1, simulation testbed will be used to 1) Depending on the availability of resources, UMS can augment the selected scenarios that may be infeasible to run on physical devices due to a large number of UAVs required or due to a high degree of risk" 2) Develop, test, and integrate with APIs and 3) Validate and measure some pre-defined KPIs over 5G networks as a verification process before physical trials.

Benefits of a Simulation Testbed

It is important to understand that it is not the intention of the simulation testbed to replace the physical trials, but to act as a bridge allowing continuity in testing as preparations for real-life demonstrations. Two crucial benefits of developing such a testbed are:

- COVID-19: In the present scenario it is making it hard for people to travel due to restrictions across Europe. As physical trials require freedom to move between countries, there is a need to come up with a contingency plan to conduct trials in a safe manner remotely from their respective offices/labs. A simulation testbed is a very useful tool to validate certain parameters before live demonstrations are conducted. This serves also as part of a mitigation plan in the current situation of COVID-19 as it is limiting the ability to conduct physical trials.
- Permits: Any physical trials or demonstrations involving drones require flight permits from the relevant Civil Aviation Authority (CAA). Obtaining such permits was time consuming, tedious, and often a limiting process in the past. New EU legislations and regulations for drones have been introduced from the beginning of this year which aim to standardise regulations at a European level. These regulations are more receptive to innovation including automated drone flights. Since the new regulations have only just come into effect, the CAA is taking additional time to process flight permits. A simulation testbed will provide the opportunity to start tests to verify basic functionalities while waiting to obtain these permits.

Architecture

A preliminary architecture of the simulation testbed being developed along with the description of major components is provided in Figure 24.

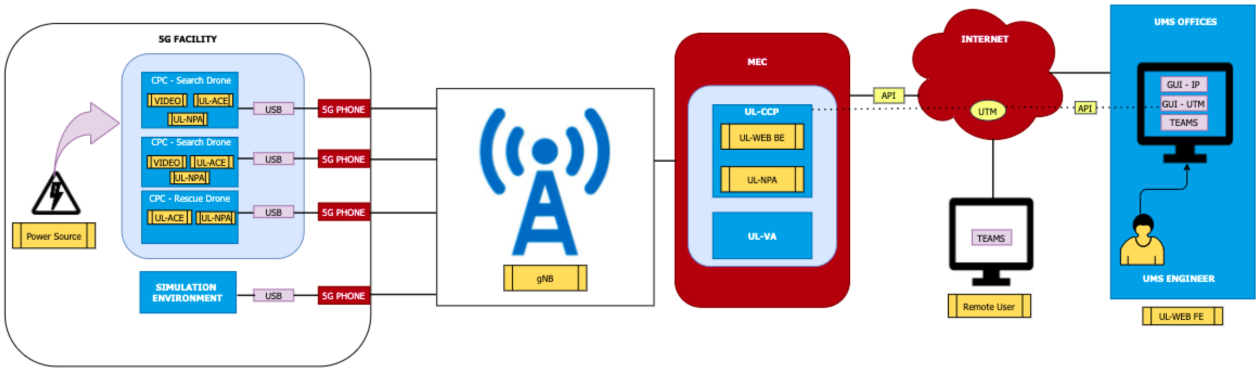

Figure 24: Simulation Architecture

UL-CCP

The UL-Central Command Platform (CCP) is a containerized application which is comprised of software modules and acts as the "central brain" for an UMS deployment, gathering data from connected UL-Autonomous Control End node (ACEs) to create a single context for devices in the fleet. UL-CCP integrates and orchestrates all the systems involved in the solution, processed via the Edge and/or cloud enabling the cloud-based Autonomy-as-a-Service operation.

UL-ACE

The UL- Autonomous Control End Node (UL-ACE) is a computing unit installed on a drone in addition to a flight controller. This computing unit provides higher processing power than the flight controller and enables higher-level autonomous control by sending C2 commands to the flight controller. The autonomous control endpoint also communicates with a central command platform (UL-CCP) that orchestrates the individual actions of a group of autonomous vehicles, hosted on a local server, on the network edge, or in the cloud. This communication can take place over Wi-Fi, 4G, or 5G.

UL-VA

The UL-Video Analytics (VA) module collects and processes the sensor data streamed from the drone for computer vision tasks. Typically, this module is located on the edge with the UL-CCP. For example, the focus is on the sensor type and analysis methods that can be applied for different system inspections. E.g., visual inspection will be camera based and will use AI techniques to analyse the camera data to perform object detection. Depending on the training data used during the learning process, the algorithm will be able to detect different types of objects. The modules can be custom made or a third-party plug-in to the UL-CCP. The results can be used to drive the mission and swarm planning or can be shared with third-party business information systems.

UL-NPA

The UL-Network Performance Analyser (NPA) module enables evaluation of performance and latency of various communication means like ROS 2, FastDDS, Connext DDS Micro, Eclipse Cyclone DDS and OpenDDS. Different performance metrics that are relevant for both performance and determinism are recorded, so that the system can be properly design and monitored. Each metric value is recorded for every second of the experiment duration.

Simulation Environment

The simulation environment is being developed in a framework known as Gazebo [21]. Gazebo offers the ability to simulate robots accurately and efficiently in complex indoor and outdoor environments. We used Gazebo with Robot Operating System (ROS), a toolkit/offboard API for automating vehicle control. The setup also contains a Pixhawk flight controller with PX4 autopilot system.

<u>Process Flow</u>

In order to understand the interaction between all the different components of the architecture, a process flowchart containing sequence of actions, services, and dataflow has been created that can be seen in Figure 25 below.

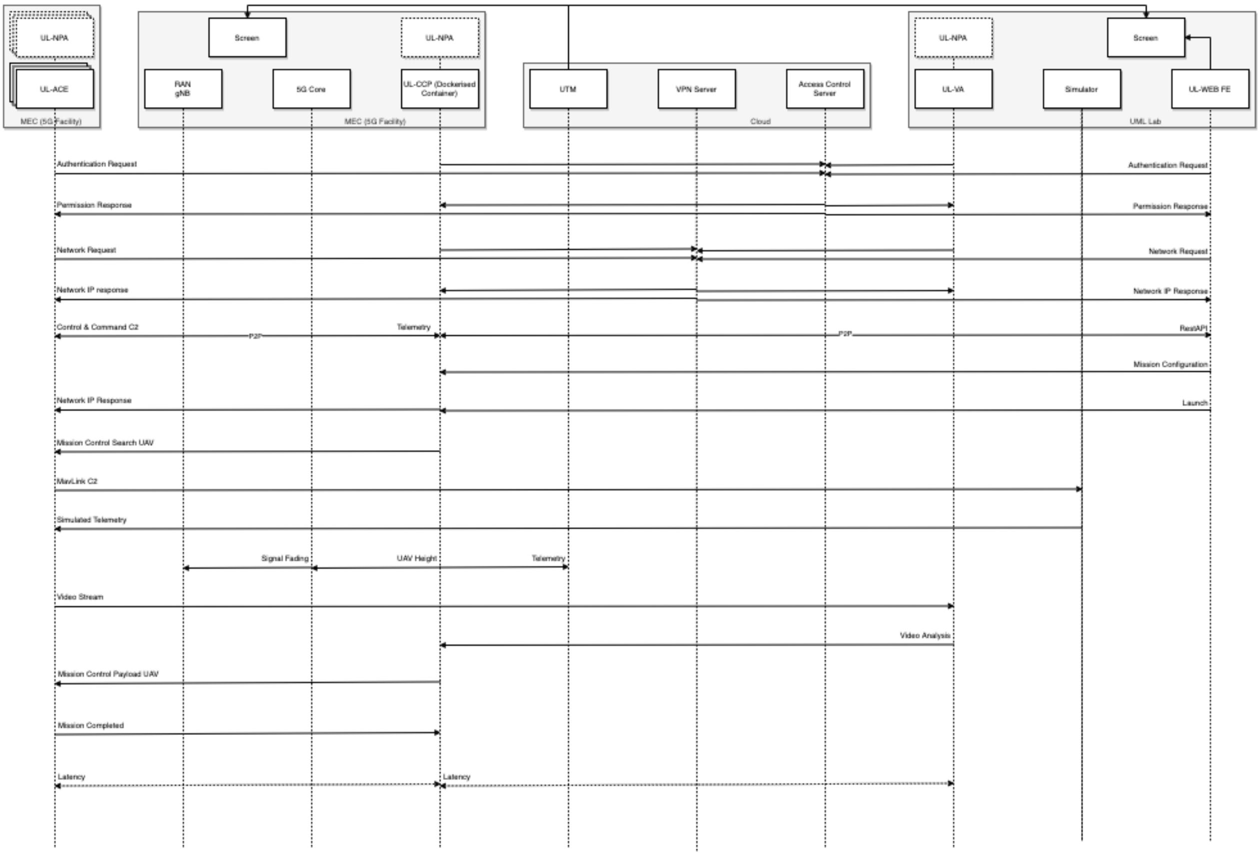

Figure 25: Simulation Process Flow

UMS is currently conducting simulation trials with support from NCSRD' and COS's teams in Athens, Greece. Initially, the objective of the tests will be to validate the functionality of all system components along with network connectivity and communication by deploying UMS modules on the MEC server hosted by NCSRD and COS.

In the next step, UMS will validate full simulation with end-to-end tests by deploying all its modules on the MEC infrastructure of 5G facilities at both Greece and France (EURECOM). This will also include integration with the UTM system. UMS is currently working with FRQ to enable this and has already completed preliminary integration. Partners will discuss the KPIs that needs to be measured during these tests along with the tools that could be utilised.

3.4 KPI Component

The Key Performance Indicator Component (KPIC), as part of the Trial Controller defined in [1], provides an agreed OpenAPI based REST Interface to receive KPI datasets defined by the sending component. The KPIC is responsible to storing and providing the KPIs for further analysis.

The current REST interface is available at:

https://5gdrones-kpi-endpoint.utm-labs-frequentis.com/swagger-ui/#/

A Kibana instance is available and exposed at:

https://5gdrones-kpi-kibana.utm-labs-frequentis.com/app/kibana#/home?_g=()

3.4.1 Architecture

The KPIC consists of 5 internal components as seen in Figure 26 The KPI endpoint in the REST interface which receives requests and provides output according to the agreed API. The ActiveMQ component receives data messages from the KPI endpoint to queue and distribute them.

The KPI Service instance consumes the queued message and prepares it for persisting it into the ElasticSearch datastore.

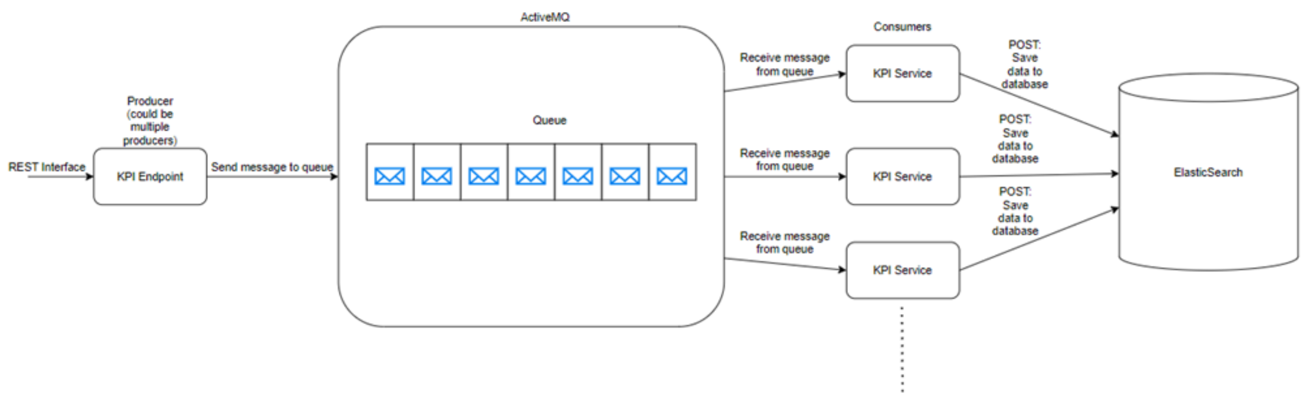

Figure 26: KPI Component - Architecture

3.4.2 API

The KPIC provides an OpenAPI based REST API which allows to provide KPI data elements.

The KPIC also allows to retrieve the status/last KPI items per Trial via the findByTrialID call.

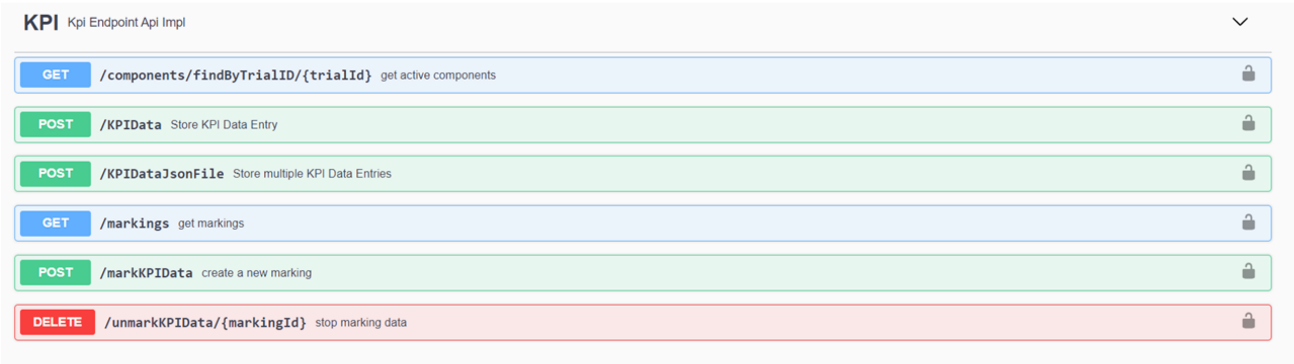

Figure 27: KPI Component - API Overview

3.4.3 Workflows

The KPIC Client reports its business KPIs via the KPIData message to the KPI Component (KPIC). The corresponding workflow is described by Figure 28: optionally the KPI Client may use the MarkKPIData call to set a tag to its reported KPI data messages. This allows easier analysis and reporting later on by marking specific business events during the Trials. Ending of the marking is possible by using unmarkKPIData.

In addition, the KPI Client may also use to use the KPIDataJsonFile call, this allows to provide KPI data packages via JSON file after the actual Trial, e.g. if no connectivity is available.

A KPIClient may retrieve current active reporting clients by using the getByTrialID call. The response consists of KPIC clients which were reporting for the defined TrialID. This allows to assure that all involved and needed clients are reporting KPIs.

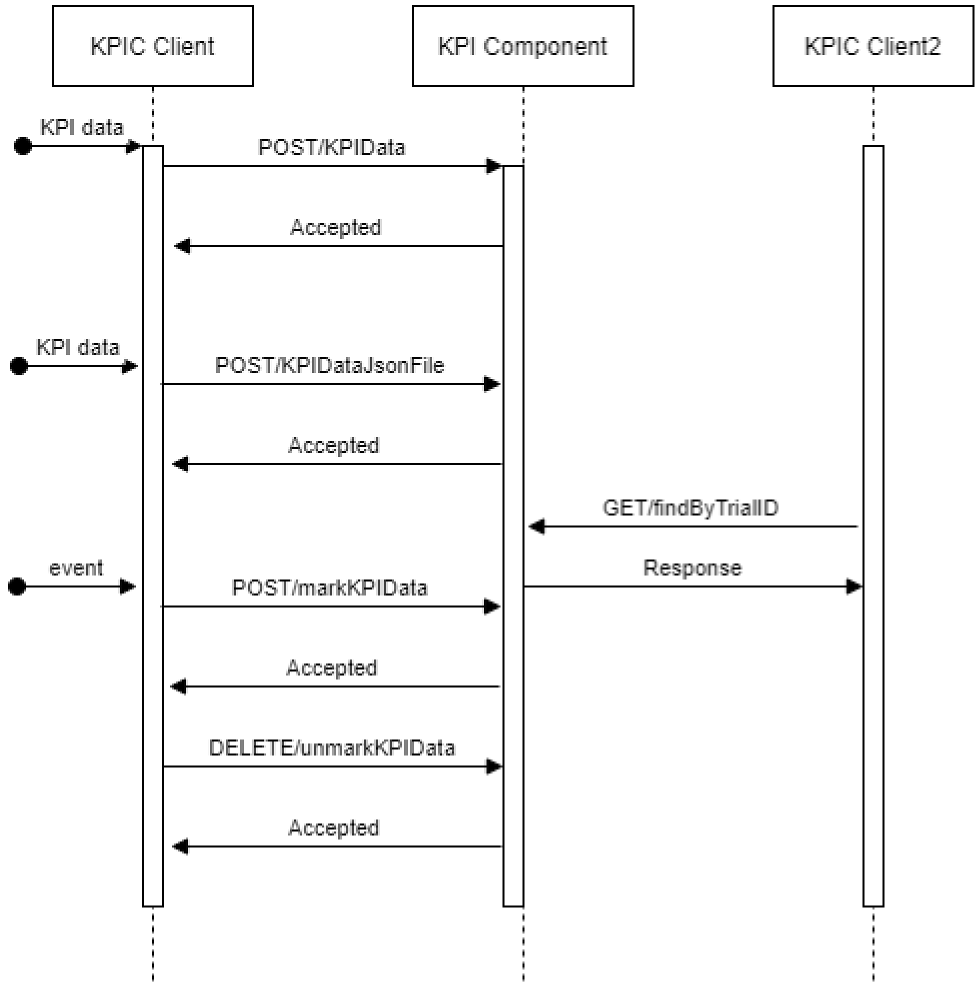

Figure 28: KPI Component Workflow

3.4.4 Elastic Search & Kibana

Elasticsearch is a search and analytics engine, part of the Elastic Stack.
The standard elastic stack features allow users to store their data in form of so-called indexes – much like tables in a conventional database. An entry in an index is a document which corresponds to a data record in a conventional database. Documents can have arbitrary fields and don't need to follow a predefined structure – this allows a flexible approach based on specific needs of different trial scenarios and environments. Querying an index is done using common fields all documents in an index share - these fields, their types and processing rules are defined in the index mappings.

Kibana is one of the open source products provided as part of the Elastic Stack. Its capabilities can be utilised to visualise data and analytics. Kibana functions on top of Elasticsearch and provides an effortless way to search and navigate data indexed in Elasticsearch. Moreover, Kibana is a powerful tool to visualise the saved logs, metrics and analysis results. Kibana offers a possibility create dashboards, which can also be shared and embedded in further applications.

4 DATA VISUALISATION & ANALYSIS

This chapter describes the data analysis and visualization methods collected during the 5G! Drones project. It is important to note that the widely used tools Excel, Google Earth and others are used for simpler analyses and visualizations. This chapter mainly describes the tools used for more complex analyses and visualizations.

For facility specific visualisation and analysis, please also check chapter 3.1 Facility Data Collection Tools

4.1 Machine Learning and Deep Learning technologies

Statistical and machine learning (ML) techniques can be leveraged for complex problems arising in network operation and management. Last years have witnessed a surge of such techniques, thanks to significant improvements in computing capacities and recent advances in data storing and processing. Applied to network security, data analysis has been widely explored to develop novel automation and detection techniques. We first describe the theory behind statistical learning models and their relevance to the networking problem, then we further review in detail the process for data analysis.

At the same time, we must take into account that ML techniques require a large amount of data and longer training. For many parameters with simpler functionality, the use of ML techniques is not practical - for example latency or UL / DL or for reliability measurements in the 5G!Drones project, there is no need to use ML techniques. Data collection, analysis and visualisation in 5G!Drones project preferably use software that is free to use, such as iPerf [20] and Nemo Handy [21]to measure network coverage quality.

4.1.1 Statistical learning

Originally, ML detection tools rely on statistical learning theories to build their model. There also exist unsupervised detection tools that use plain statistical approaches; their central assumption is that the phenomena the most rarely observed are the most likely to be anomalous. In statistical approaches, the fine analysis of the built statistical profiles of traffic allows one to understand how the detected anomalous instances differ from the usual behaviour; however, they work on a mono feature basis, thus do not correlate the different features by design. Moreover, statistical approaches work well combined with other algorithms as they are usually unable to provide additional information, such as the IP addresses of attackers or the attack root causes; in addition, they are of practically no computational complexity and easy to implement, which makes them a wise approach when the detection should operate with limited computational resources.

4.1.2 ML techniques: paradigms and addressed problems

We are now focusing on ML techniques, investigating the whole ML design pipeline, which consists in various steps: learning paradigms and ML techniques, data collection, features design, model evaluation, and ML applications.

4.1.2.1 Learning paradigms

First, ML techniques can be classified into four learning paradigms:

1. **Supervised learning** techniques learn from a labelled dataset what constitutes either normal traffic or attacks – there exist different techniques such as support vector machine (SVM)-based classifiers, rule-based classifiers, and ensemble-learning detectors.
2. **Unsupervised** approaches learn by themselves from unlabelled data, properties underlying the data, e.g., clusters, or rules describing the data. [22] [23]
3. **Hybrid or semi-supervised** approaches benefit from only a small part of labelled traffic, meant to be enough to learn from.
4. **Reinforcement learning** (RL) is an iterative process with agents that take actions in order to maximise the notion of cumulative reward. In the purpose of decision making, the learning is

traditionally based on exemplars from training datasets. The training data in RL constitutes a set of state-action pairs and rewards (or penalties).

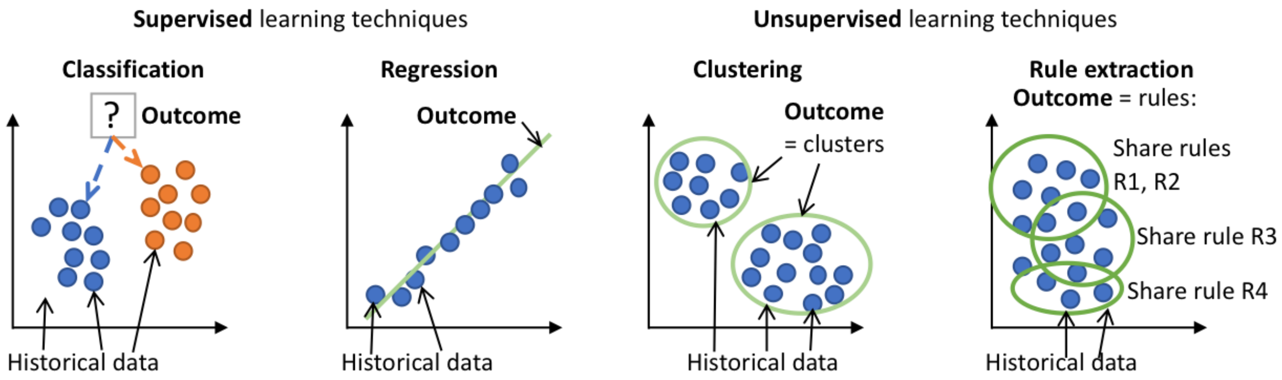

Figure 29: ML Learning paradigms

Figure 29 shows learning paradigms that benefit from machine learning: classification and regression for supervised learning, and clustering and rule extraction for unsupervised learning.

4.1.2.2 Problem categories

Four broad categories of problems can leverage ML, namely classification, regression, clustering, and rule extraction, as illustrated in Figure 29 . First of all, **classification** and **regression** are two supervised learning approaches; their objective is to map an input to an output based on example input-output pairs from labelled data. **Regression** approaches predicts continuous values output, whereas classification predicts discrete values, consisting in the different labels. Then, **clustering** and rule extraction are unsupervised learning techniques: clustering is the task of partitioning the dataset into groups, called clusters - the goal is to determine grouping among unlabelled data, while increasing the gap between the groups; **rule extraction** techniques are designed to identify statistical relationships in data, by discovering rules that describe large portions of the dataset. Note that the choice of the learning paradigm strongly depends on the training data. For example, if the dataset is not labelled, supervised learning cannot be employed, and other learning paradigms must be considered.

Concrete ML techniques applications

To illustrate the diversity in ML techniques applications, we provide concrete use cases for each of the aforementioned learning paradigms:

- **Classification** techniques are traditionally used in problems that contain labeled datasets, with two or more distinct classes. Botnet detection is but one example of such cases, where we can distinguish between malicious and benign flows; this way, the ML algorithm implicitly learns the inherent characteristics of a bot, and those of a benign host.
- **Regression** techniques are traditionally used for time series forecasting. The objective is to construct a regression model able to induce future traffic volume based on previous instances of traffic. Regression techniques are also employed to assess the impact of the global network condition on the QoS (Quality of Service) or QoE (Quality of Experience). Finally, monitoring Key Performance Indicators (KPI) in large-scale networks enables the quick detection of network outages and attacks.
- **Clustering** techniques are usually employed for outlier detection purposes. In network cyber-security, many intrusion detection schemes rely on data clustering to highlight significant deviations compared to usual end-user behaviours.

- Finally, **rule extraction** techniques, also named association rule mining, are commonly employed for personalised recommendations. Market Basket Analysis is one of the key techniques used by large retailers to discover correlations within sets of items. These techniques are also used by recommendation engines as for Netflix (for personalised movies recommendation) and Amazon (for suggestions of other articles related to the purchased one).

4.1.2.3 Data collection

The process of collecting data to apply and validate a given ML technique is an important step, but nonetheless difficult. Finding representative data, possibly without bias and labelled is a non-trivial task. Datasets also vary from one problem to another and from one time period to the next one. Data monitoring techniques are classified into active, passive, and hybrid techniques.

For more trial specific data collection topics, please check chapter 3 DATA COLLECTION

Active monitoring uses traffic measurement to collect relevant data and examine the state of networks. Such approaches commonly consist in a set of distributed vantage points hosting measurement tools like ping and traceroute; among them, RIPE Atlas [24] is a global network of over 10,000 probes that measure Internet connectivity and reachability, used for instance to identify data-centre collocation facilities in traceroute data from RIPE Atlas built-in measurements, then to monitor delay and routing patterns between facilities.

In contrast, **passive monitoring** collects existing traffic and infer the state of networks from it. Compared to active monitoring, it ensures that the inferred statistics correspond to real traffic and it does not introduce additional overhead due to bandwidth consumption from injected traffic. Passive monitoring data can be obtained from various repositories, given it is relevant to the networking problem being studied. Such traces include CDN (content delivery network) traces, darknets, and network telescope datasets. The latter consists of a globally routed, but lightly utilised network prefix - a /8 for the University of California San Diego (UCSD) Network Telescope Aggregated Flow Dataset and /20 for the Network Telescope dataset from LORIA. Inbound traffic to non-existent machines is unsolicited and results from a wide range of events, including misconfiguration, scanning of address space by attackers or malware looking for vulnerable targets, and backscatter from randomly spoofed DoS attacks. Other examples of passive data repositories include the Measurement and Analysis on the WIDE Internet (MAWI) Working Group Traffic Archive [25] and the CTU-13 dataset [26].

4.1.2.4 Feature design

Before applying an ML algorithm to the dataset, the collected raw data must be formatted to cover an adequate set of features. The first phase named Feature Extraction consists in cleaning the dataset that may contain missing values or noise. In addition, the collected raw dataset may be too voluminous to be handled. The need for dimensionality reduction is justified by multiple reasons. First, a large number of features may induce a high computational overhead. Also, a phenomenon called the curse of dimensionality refers to the sparsity in data increasing with the number of dimensions, which makes the dataset no more consistent. We first depict the features traditionally selected for network traffic analysis depending on the aggregation level. We then review the main strategies employed for feature extraction.

- **Common feature choice**

In reality, the feature choice directly depends on the problem formulation (i.e., the detection target) and thus on the granularity level. The taxonomy of aggregation levels for network traffic analysis includes payload-based, host behaviour-based, and flow feature-based techniques.

Payload-based analysis systems parse the packet payload looking for known application signatures. However, this incurs a high computational overhead and requires manual interventions from humans to monitor the alerts and regularly update the signatures database. In addition, the payload tends to be systematically encrypted due to privacy concerns. Payload-based systems usually employ features such as the payload size, but also more complex ones like specific byte sequences or particular key-words present in the payload that would execute malicious actions.

Host behaviour-based analysis systems compute per-host traffic features to model behavioural characteristics of hosts. Contrary to payload-based systems, it examines the inherent characteristics of hosts but also assesses graph-based features by considering hosts as nodes in a graph, to measure for example the centrality of nodes or the amount and frequency of traffic exchanged between the nodes. IDSes implemented at the host-level are named Host-based Intrusion Detection Systems (HIDSes), whereas those at the network-level are named Network-based Intrusion Detection Systems (NIDSes). Common features used by such systems include packet counts exchanged between nodes, service proximity, activity profiles, session duration, periodicity, and byte encoding or statistical characterisation of bytes, for each packet or each flow coming from a host.

Flow feature-based analysis systems aggregate communications on a per-flow basis, which consists of a 5-tuple made from the protocol, the source and destination IP addresses, and the source and destination port numbers. It is then a unidirectional exchange of consecutive packets on the network from a port at an IP address to another port at another IP address using a particular application protocol, including all packets pertaining to session setup and tear-down, and data exchange. A feature is an attribute representing unique characteristics of a flow, such as the number of packets in a flow, mean packet length, packet inter-arrival time, flow duration, entropy, to name a few. Entropy basically represents the traffic distribution predictability and enables one to detect volume-based anomalies such as DoS and DDoS. Flow feature-based techniques use flow features as discriminators to map flows to classes of interest.

- **Feature extraction**

Two main processes are usually employed to adequately select features.

The process of **Feature Selection** consists in removing the features that are not relevant or redundant in order to keep only a limited set of features. The filtering strategy (e.g. information gain), the wrapper strategy (e.g. search guided by accuracy), and the embedded strategy (selected features are added or removed while building the model based on prediction errors) are three techniques for feature selection.

The second process, named **Feature projection**, projects the data from a high-dimensional space to a space of fewer dimensions. The variance between each class is accentuated in the resulting space, removing redundancy in data. Both linear and nonlinear dimensionality reduction techniques exist. The main linear technique is named Principal Component Analysis (PCA), which finds the directions of maximum variance. The fraction of variance explained by a principal component is the ratio between the variance of that principal component and the total variance. The objective is to reduce the dimensionality while keeping a good amount of information, so that the cumulative explained variance ratio is close to 100%. PCA may also be used for traditional outlier detection. Using real traffic traces, one can demonstrate that normal traffic data can reside in a low-dimensional linear subspace and form a low-rank tensor. The anomalies (outliers) should stay outside this subspace. Therefore, tensor-based approaches try to recover the normal data by separating the low-rank normal data and outlier data from the noisy traffic data captured, and then detect anomalies by using the outlier data separated.

4.1.3 Deep learning techniques

Being a subset of Machine Learning (cf. Figure 30 from Microsoft documentation [27]), Deep Learning is usually defined by the assets it offers in regard to machine learning. Living in big data era, being able to make models from huge amount of data is tremendously important. Deep learning enables us to do so by using neural networks.

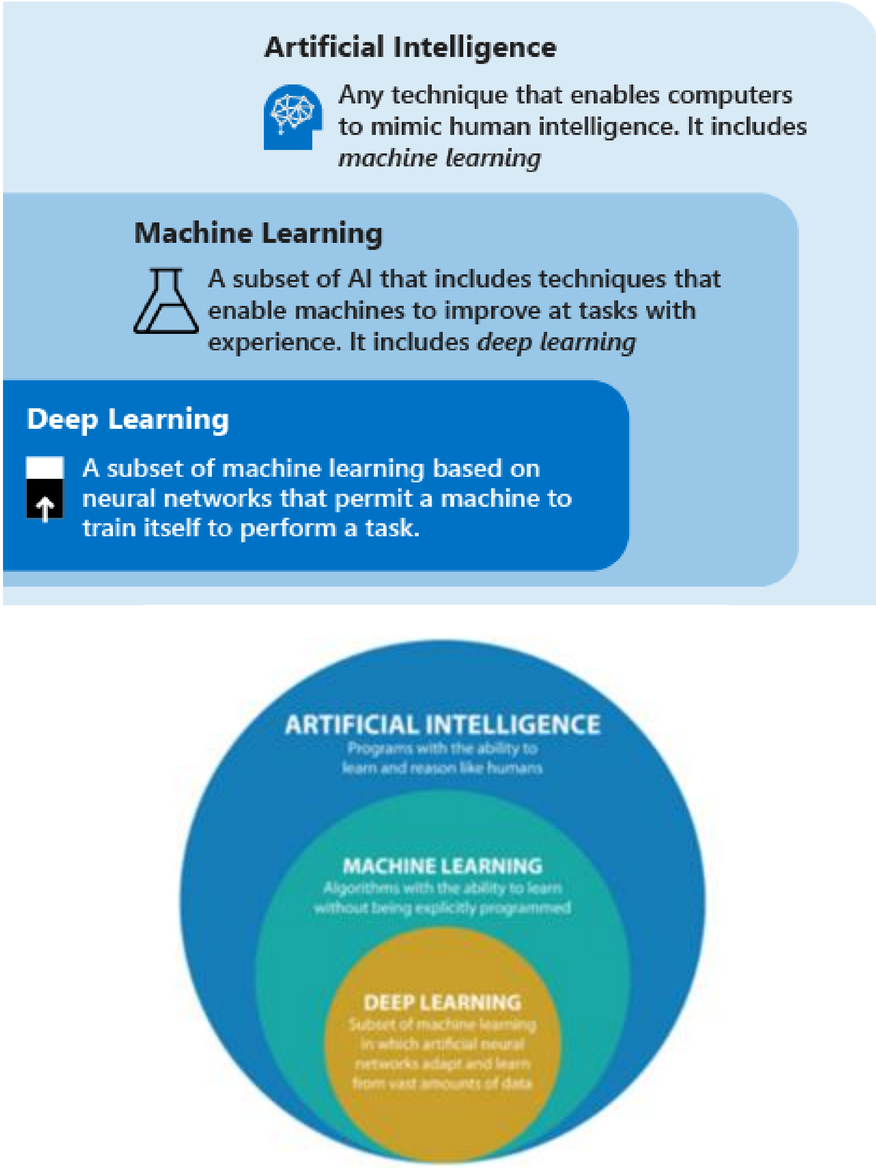

Figure 30: Deep Learning in Artificial Intelligence field

Neural networks are called that way because of the famous analogy between its base unit (the perceptron) and the nerve cells = neurons as illustrated in Figure 31 [28](

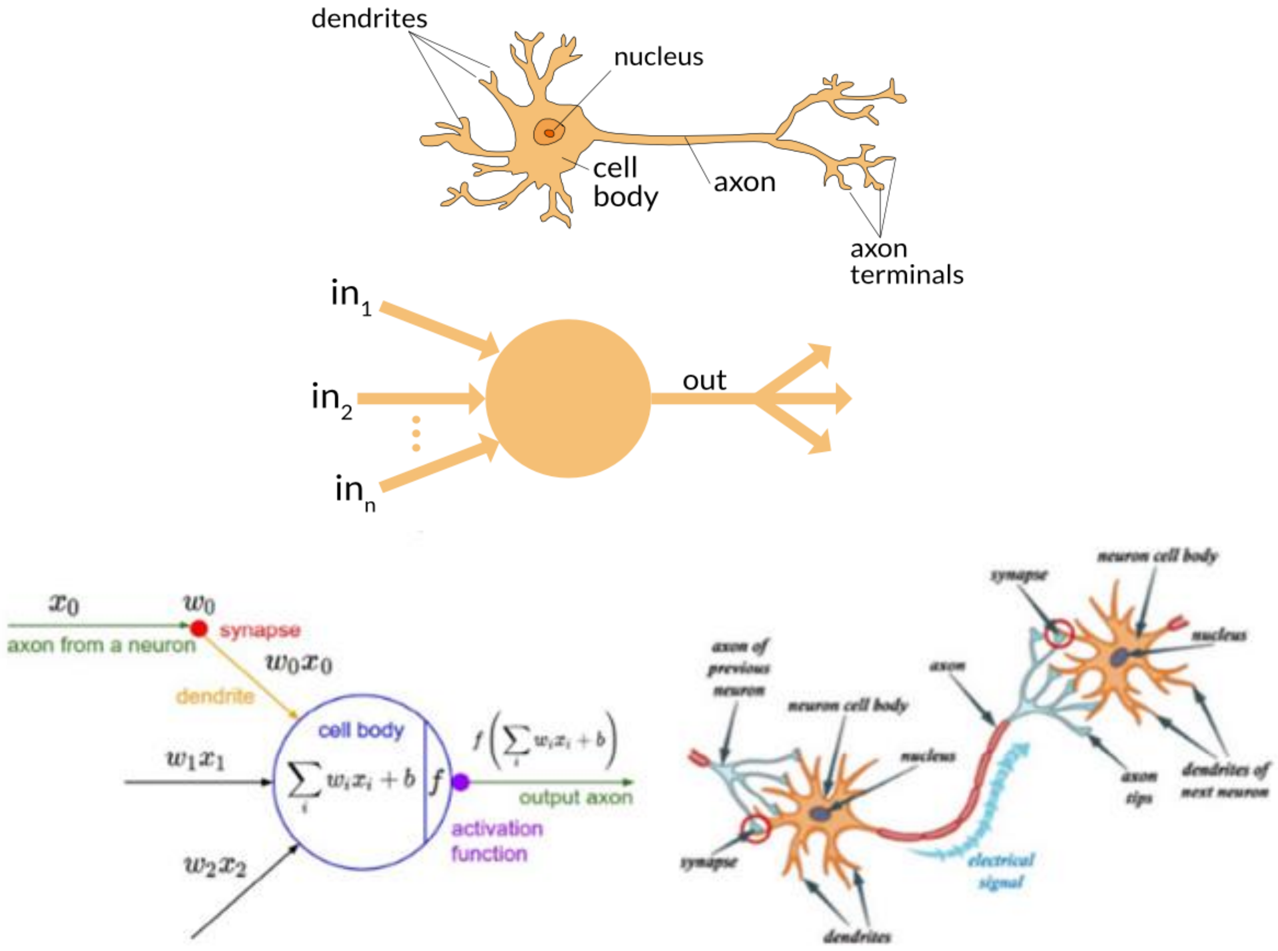

Figure 31: Percepton / Neuron analogy

If you take several perceptrons, organise them in layers with the first one connected to the input and the last one leading to a result that has the same format as the attended result (with an activate function if needed) you end up with a neural network (see Figure 32). The aim is to find the optimal weights vector **W*** to predict output from any input. The training set – as its name indicates – is used to find that vector based on the following process: going from the vector W_t, we compute the output and compare it to the real output **(FORWARD PROPAGATION)**. The observed error is then used for updating the **W vector** and obtaining W_{t+1} **(BACKWARD PROPAGATION).**

There are many update rules, the easier one is the gradient descent rule: $\boxed{\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}}$

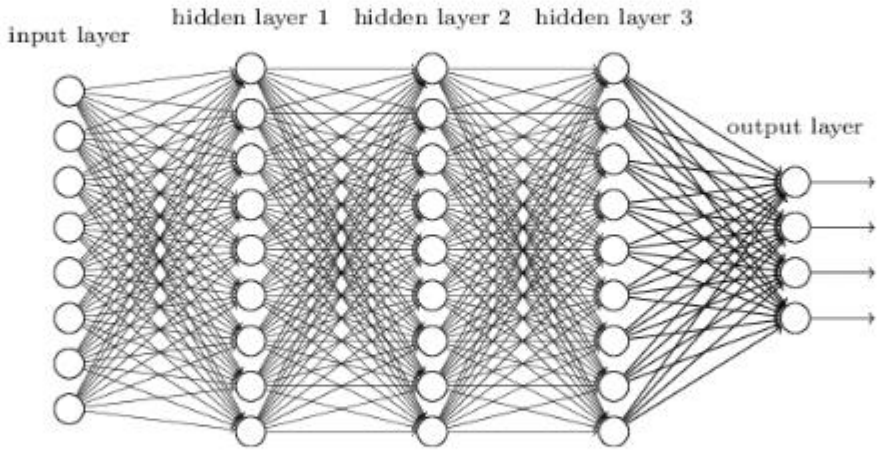

Figure 32: Neural network

Step by step, we tend to the optimal vector **W*** which defines our model.

Furthermore, we can tune the network by changing the number of layers, not connecting all the neurons to each other, adding convolutional layers (used when inputs are images), using other activation function (tanh, ReLU, leaky ReLU, softmax).

Indeed, deep learning can learn the hidden structure of huge amount of data while remaining very practical to manipulate: updating a saved model is achieved by running forward and backward propagation of new data with the already existing weights – this process is not very time consuming. Comparison with machine learning is made in Figure 33.

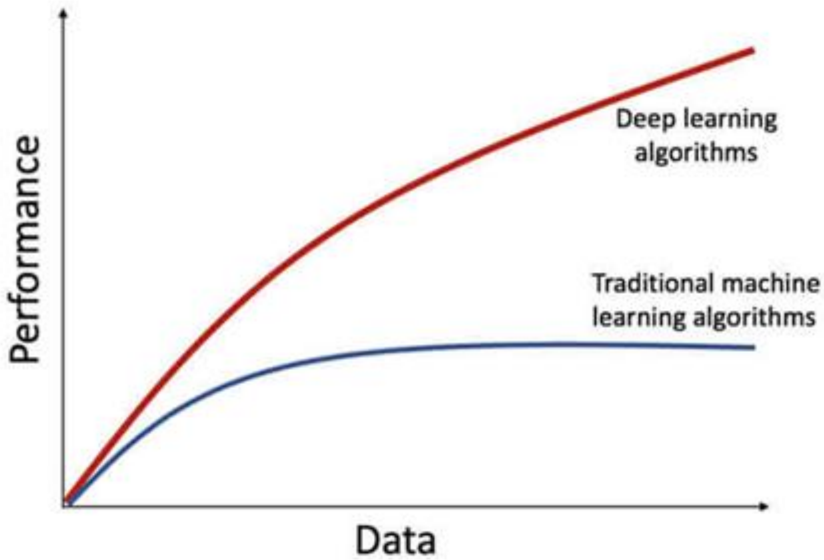

Figure 33: ML and DL comparison

4.1.4 Frameworks and APIs for ML and DL

Being tremendously useful in AI, we have to present ML/DL frameworks as you cannot pass by it. Tensorflow is one important framework which is a google automated learning platform that simplifies AI developers' work and gives the opportunity to create machine learning models **easily**.

Front-end API for building applications with the framework is in python, but those applications are executed in high-performance C++ which makes it **faster**.

4.1.4.1 Data pre-processing

For data prep-processing we use practical libraries. **Pandas** [29] is one library and it offers the very convenient pandas.dataframe format. Following tasks are simplified by pandas:

- Importing data sets: most of the common formats for data sets are readable with pandas (csv, json...).

- Queries over data sets: pandas.query function.

- Operations on columns: pandas.series format has useful dedicated functions.

scikit learn [17] is another library and it offers pre-processing toolbox, simplifying steps such as normalisation, data set split and scaling.

4.1.4.2 ML models handling

In order to create, modify and save machine learning models **scikit learn** is used**.** All types of machine learning algorithms are available on scikit learn and listed in Figure 34 extracted from Scikit-Learn documentation [30], whether it is Bayesian, ensemble learning classifiers (Random Forest, XGBoost), regressors or clustering (K-Means) algorithms.

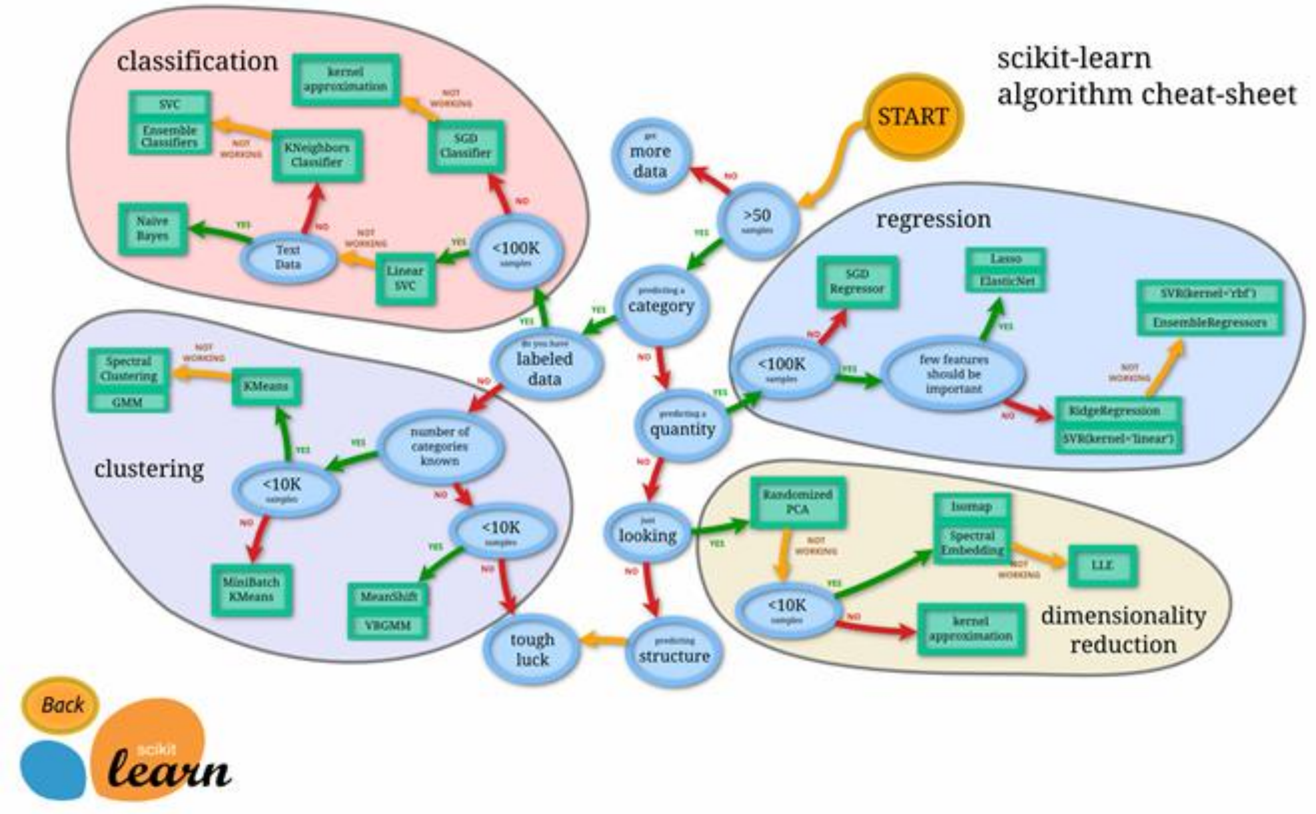

Figure 34: Available models on scikit learn

4.1.4.3 DL models conception

When it comes to neural networks, we use **Keras, Tensorflow's high level API.** Creating Neural Networks is simplified, we just have to describe each layer and its parameters, as demonstrated in Figure 35

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

Figure 35: DL models definition with keras

Moreover, we can have access to optimisers such as Adam or RMS_prop, simplified training function, pre-trained models for transfer learning: everything we need in deep learning.

4.2 ML-based data analysis for enhanced C2 (Command and Control) communication in cellular UAVs

Ensuring stable C2 communication is very important for cellular UAVs. This communication is reflected in the downlink scenario, in which packets are sent from the serving BSs (Base Station) to the connected UAVs. Enhancing the QoS experienced by the flying UAVs can be translated into optimising the transmission power employed by the BSs. This parameter also affects the interference amount pushed to the UAVs. In this subsection, we consider the outage probability expression defined in section 3.3.1 as a QoS metric. Therefore, reducing the outage probability for C2 communication considering the power optimisation problem can be formulated as

$$\min \max_{v \in \mathcal{V}} \quad P_{uv}^{out}(P_{u,s})$$

$$\textbf{s.t.}$$

$$\forall u \in \mathcal{U}, \forall s \in \mathcal{S}; 0 \leq P_{u,s} \leq P_{\max}.$$

In the above optimisation problem, $P_{uv}^{out}(P_{u,s})$ refers to the outage probability on the downlink for the link between the BS $u \in \mathcal{U}$ and the UAV $v \in \mathcal{V}$ being served via the sub-carrier $s \in \mathcal{S}$. $P_{u,s}$ refers to the transmission power employed by the BS u on the sub-carrier s. However, the above optimisation problem is not linear, which is due to the expression of the outage probability. Solving this problem takes huge time, especially for large scale networks. Therefore, such optimisation cannot be considered for online use.

4.2.1 ANN-based optimisation

The above optimisation aims to find the optimal allocation and can be considered in an offline environment. However, as solving the global optimisation takes time, this process is not adequate for online use. We therefore propose deep learning approach to optimise the allocation of powers on the downlink scenario. This can be achieved by establishing a prediction map defined as

$$\mathcal{F} : H \longmapsto \hat{P},$$

where H is the feature extracted from the deployed network, while $\hat{P}$ stands for the predicted power allocation (defined as $\hat{P} = \left(\hat{p}_{u,s}\right)_{u,s} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{S}|}$. A feature needs to be defined in a way to efficiently accommodate the scalability of the network. Indeed, the number of UAVs can change in practice and the prediction map needs to be trained to cope with this change. In the other hand, the number of the BSs and sub-carriers does not change frequently in practice. We therefore define a feature as $H = \left(a_{u,s}\right)_{u,s} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{S}|}$, where $a_{u,s}$ is related to the pathloss for the UAV served via the subcarrier s of the BS u.

For the ANN architecture, we consider a fully connected feedforward network. The input layer takes a realisation of a parameter of the vector H, while the outputs layer produces a power allocation $\hat{P}$ that estimates the optimal allocation P^*. a number K of hidden layers are considered between the input and output layers. A layer $k = 1, \ldots, K+1$ has L_k neurons, where each neuron l computes

$$\zeta_k(l) = \mathcal{G}_{l,k}(\theta_{l,k}^T \zeta_{k-1} + \delta_{l,k}),$$

where $\zeta_k = (\zeta_k(1), \ldots, \zeta_k(L_{k+1}))$ stands for $L_{k+1} \times 1$ output vector of layer k, $\theta_{l,k} \in \mathbb{R}^{L_{k-1}}$ denote the weights terms, while $\delta_{l,k} \in R$ refer to the biases terms. $\mathcal{G}$ is the activation function of neuron l in layer k. The combination of multiple neurons allows to perform complex tasks and emulate different functions.

In order to tune the weights and the biases in the network in a way to efficiently approach the target optimisation, the network needs to be trained using the training set (H, P^*). By exploring this set, the network can learn predicting the power allocation $\hat{P}$ that estimates the optimal assignment, even for new realisations. The training process will allow adjusting the weights and the biases to minimise the loss between the actual and the desired output, formulated as

$$\min_{\theta,\delta} \frac{1}{L_T} \sum_{L_T}^{l=1} \mathcal{L}(\hat{P}_l(\theta,\delta), P_l^*),$$

where $\theta = \{\theta_{l,k}\}_{l,k}$ and $\delta = \{\delta_{l,k}\}_{l,k}$. $\mathcal{L}$ stands for the error function. A widely used error function is the mean square error defined as $\frac{1}{\mathcal{L}_T} \sum(\hat{P}_l(\theta,\delta) - P_l^*)^2$.

4.2.2 Training and execution phases

The ML-based approach for optimising C2 communication is executed in two phases as depicted in Figure 36. The first level corresponds to the training phase in which a prediction map will be trained. Starting from an initial set of network deployments, the global optimisation will be applied. This allows to extract the optimised allocation of power, P^*, along with the corresponding feature values H. The constructed dataset of feature-label pairs will feed and train a prediction map. To do so, parallel models will be trained and evaluated in the same time. The model ensuring the lowest error will be selected to be used at the execution phase.

The second phase corresponds to the execution phase in which the prediction map will be used to predict the power assignments referred to us by $\hat{P}$. This phase is fast and can be used online.

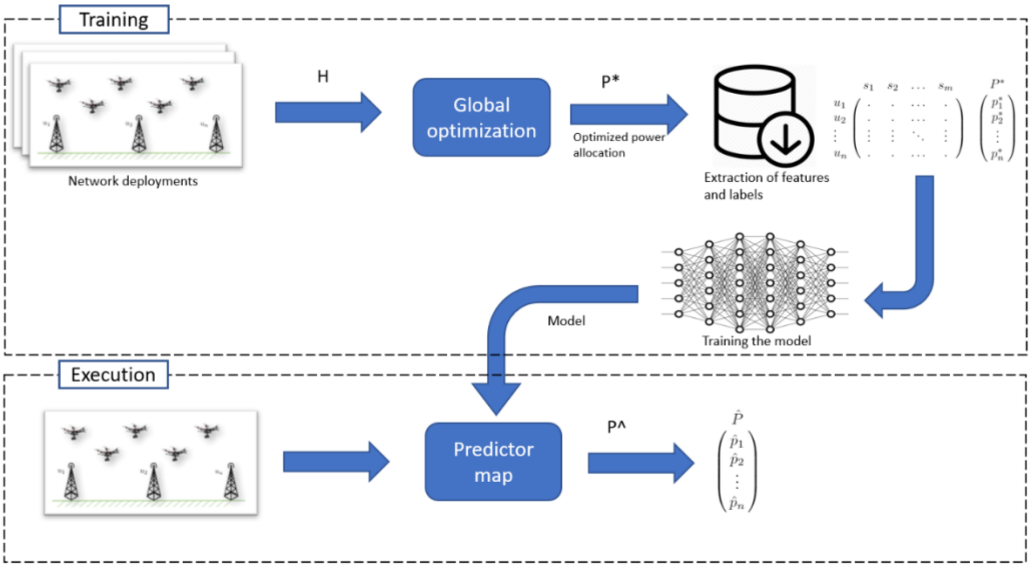

Figure 36: Training and execution phases for optimal power allocation.

4.2.3 Performance evaluation

To evaluate the ML approach, a simulation using python has been performed by AU. Pytorch [18] has been used to model and train the ANN. The considered dataset is composed of 10000 samples, meaning that the initial optimisation has been performed 10000 times. The evaluation considers 70% of samples for training and 30% of samples for validation. As hyper-parameters, we have considered a learning rate of 0.00001 and a batch size equal to 100.

Figure 37 shows the evaluation of the training and the validation loss. This evaluation has been considered throughout 100 epochs. As we can see, the execution of the ML approach is associated with a small loss. For the training phase, the loss was less than 0.073. Furthermore, the validation phase demonstrated a loss less than 0.084. This shows the effectiveness of the ML approach in predicting transmission powers that are near to the optimal values. Furthermore, unlike traditional optimisation method, the application of the predictor map is very fast and can be considered for online use. The performed evaluation showed that the execution time of the ML approach was less than 0.005s. It should be noted that the execution time for optimising the power considering the proposed approach includes extracting the features and reshaping their dimensions, as well as passing them via the trained network. We also note that these evaluations have been performed an x86_64 machine with 8 CPUs of 2397.224 MHz.

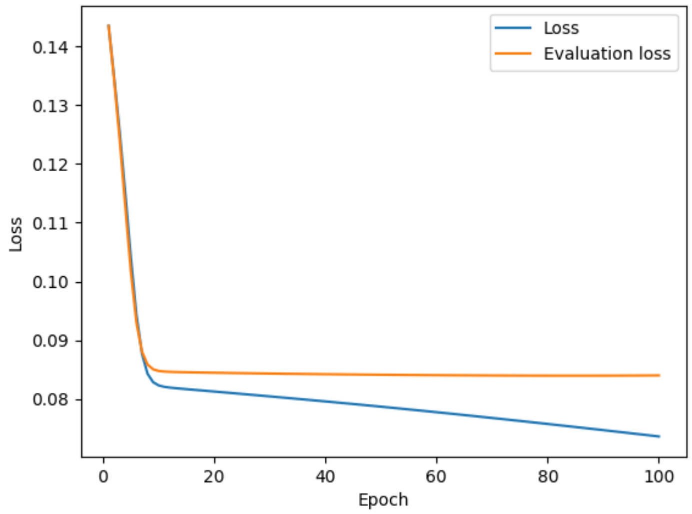

Figure 37: evaluation of the training and validation loss for the power optimisation solution

4.3 Photo/Video ML Analysis

Before the use of machine learning, image processing was based on one key idea: **performing convolution to extract feature from images**. From that, multiple operators have introduced: for instance, the **Sobel operator** approximates through convolution the derivatives of the original image. As a result, it detects the **edges** of the image, see Figure 38

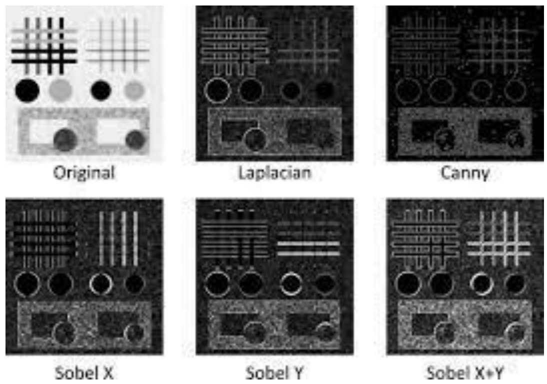

Figure 38: Some convolution operators in image processing

The same idea is used for **object detection,** as illustrated by the **histogram of oriented gradients** (HOG) in Figure 39 from OpenCV documentation [31]:

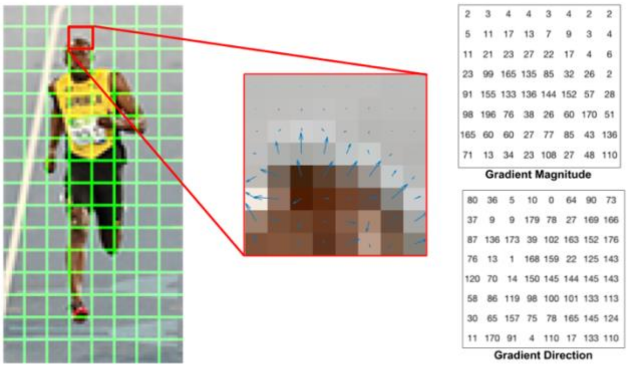

Figure 39: Histogram of oriented gradients

These techniques are not totally satisfying, as **computation time is high** and error rate is not negligible.

Machine learning algorithms entered the image processing field with the famous **Viola-Jones algorithm** in 2001 (Rapid Object Detection using a Boosted Cascade of Simple Features, P.Viola, M.Jones, 2001) which aims to detect faces while being pre-trained on a few thousand images data set. This algorithm represents all the assets of machine learning over other algorithms in image processing: **efficiency** (detects faces with almost no error) and **real time** use.

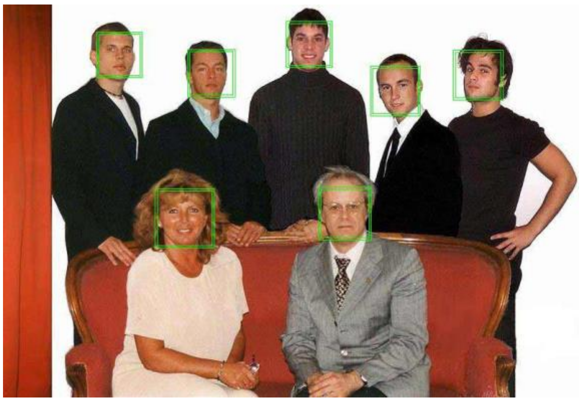

Figure 40: Viola-Jones face detection

Furthermore, as computer vision became a key issue, machine learning algorithms in image processing evolved to a more sophisticated form: **Convolutional Neural Networks** (CNN). CNNs combine the assets of neural networks (fast updates, perform multiple tasks in parallel, ability of approximating any function) and the assets of convolution (extracting features from images).

Since 2010, The ImageNet Large Scale Visual Recognition Challenge [32] is a competition of objects and face detection in which searchers from all over the world confront their algorithms. This challenge has become the most iconic example of the supremacy of CNNs in objects detection. Since 2012, all the winners have been using CNNs:

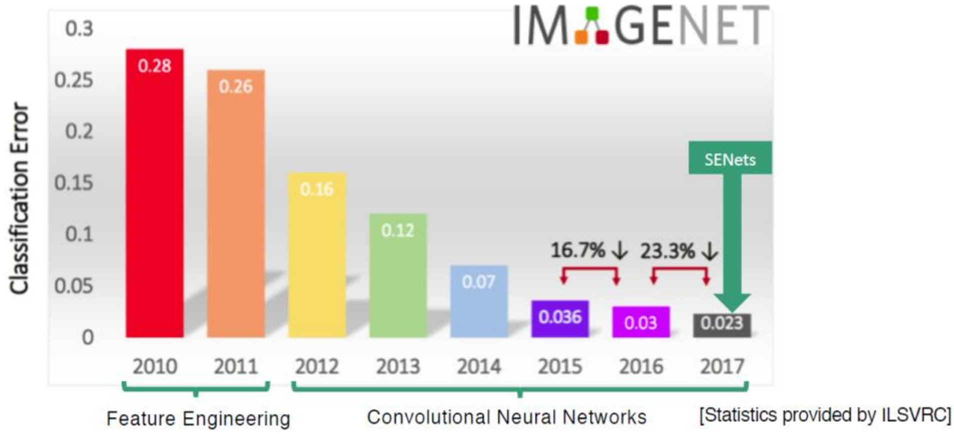

Figure 41: The ImageNet Large Scale Visual Recognition Challenge

4.3.1 Object Detection

For the long-range power line analysis scenario (UC3SC1SSC2), HEP will be using convolutional neural networks (CNN) for object detection from photos. The architecture will incorporate two parts – the first one detects features of interest, like insulators, then the second part determines which objects are faulty and which are not.

The architecture will be trained on a set of images captured from real infrastructure, which are pre-annotated by HEP. The trained models will be incorporated into Hepta's infrastructure inspection software uBird. Individual images captured in the trial are planned to be uploaded into uBird via an API as soon as they are captured. The detection model is then planned to be run on these images as batch jobs at set intervals.

As a result of the analysis, annotations are produced for the images with marking box coordinates, type of object, type of defect and its severity. Visualisation is done in uBird. A marking box is shown around the detected object with colour coding corresponding to the severity of the defect, as depicted in Figure 42.

Figure 42: Power Line Analysis

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take an input image, assign learnable weights and biases to various objects in the image and can differentiate one from the other. The pre-processing required for a ConvNet is lower compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/characteristics.

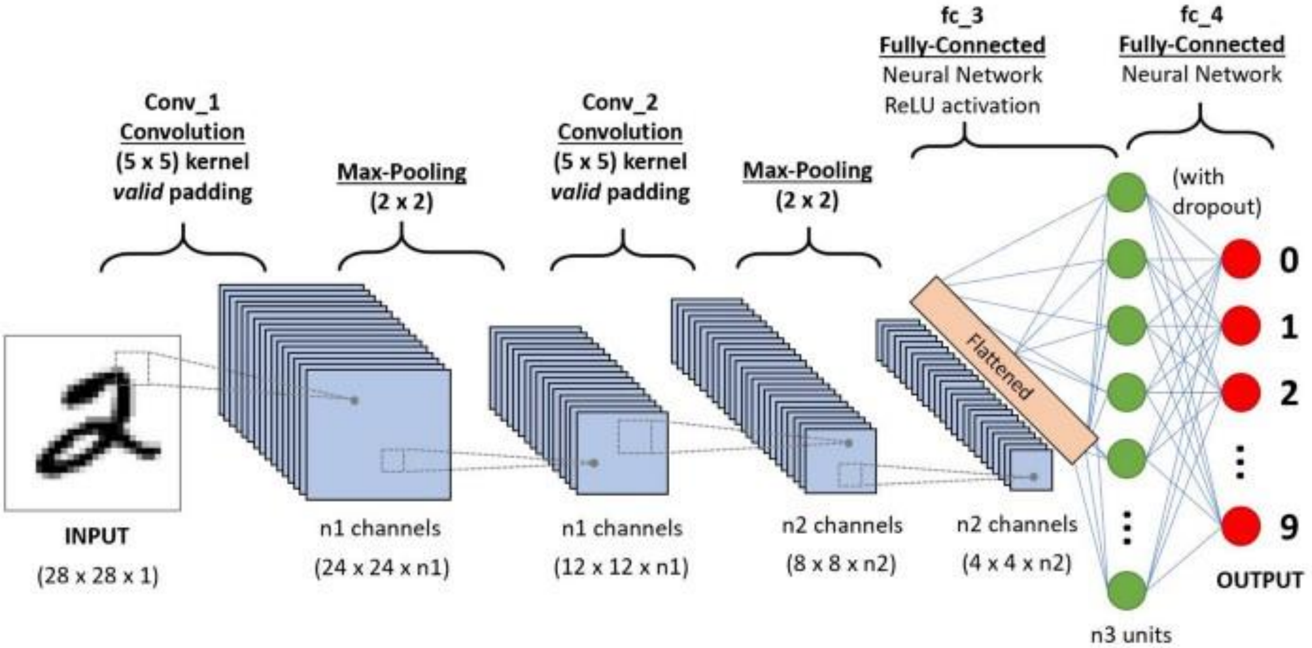

Figure 43: A CNN sequence to classify handwritten digits

Applying CNN and similar techniques for UAV based powerline inspection has been trending in the recent years, and significant research has been done in academia to explore various combinations of algorithms, architectures, and data models in order to make the process more efficient and optimum. One such example is shown in Figure 44. The structure of our power line detection network. The side outputs are generated by combining all the convolutional feature maps in each stage, which become coarser as the stride becomes larger. Exploiting the structured features, the final output presents excellent performance [33].

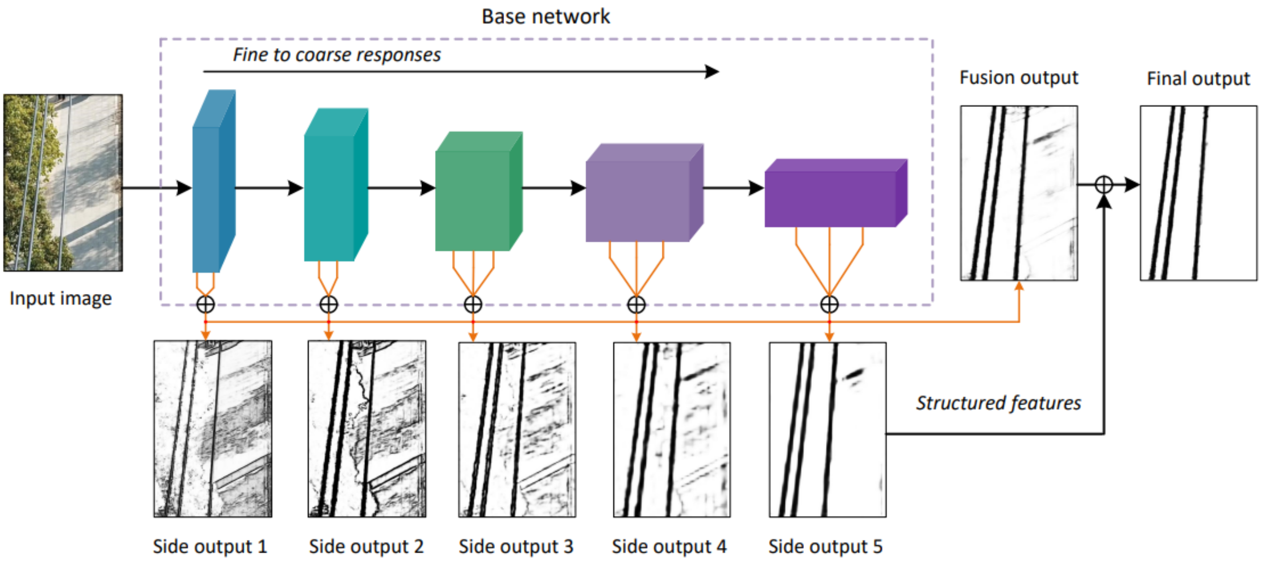

Figure 44: UAV based powerline inspection and detection structure example

In terms of the end-to-end process and structure of the UAV based powerline detection using methods like CNN, Figure 45 shows the breakdown of the stages involved and Figure 46 shows examples of results from employing methods such as CNN for powerline inspections using UAV. In the figure, results from several variants of CNN algorithm has been showcased, namely, (a,b) PorSTI-W and damper weights, (c,d) PorSTI-R and PolSTI, (e) spacer, (f) balisor, (g) LA and PorSTI-W, (h) sag adjusters and PorSTI-R, and (i,j) transmission towers, by the trained transmission line components detector. Every components appear with a random orientation, point of view, lighting conditions, shapes, colours and scales, yet the CNN-based detector is able to detect them robustly [34].

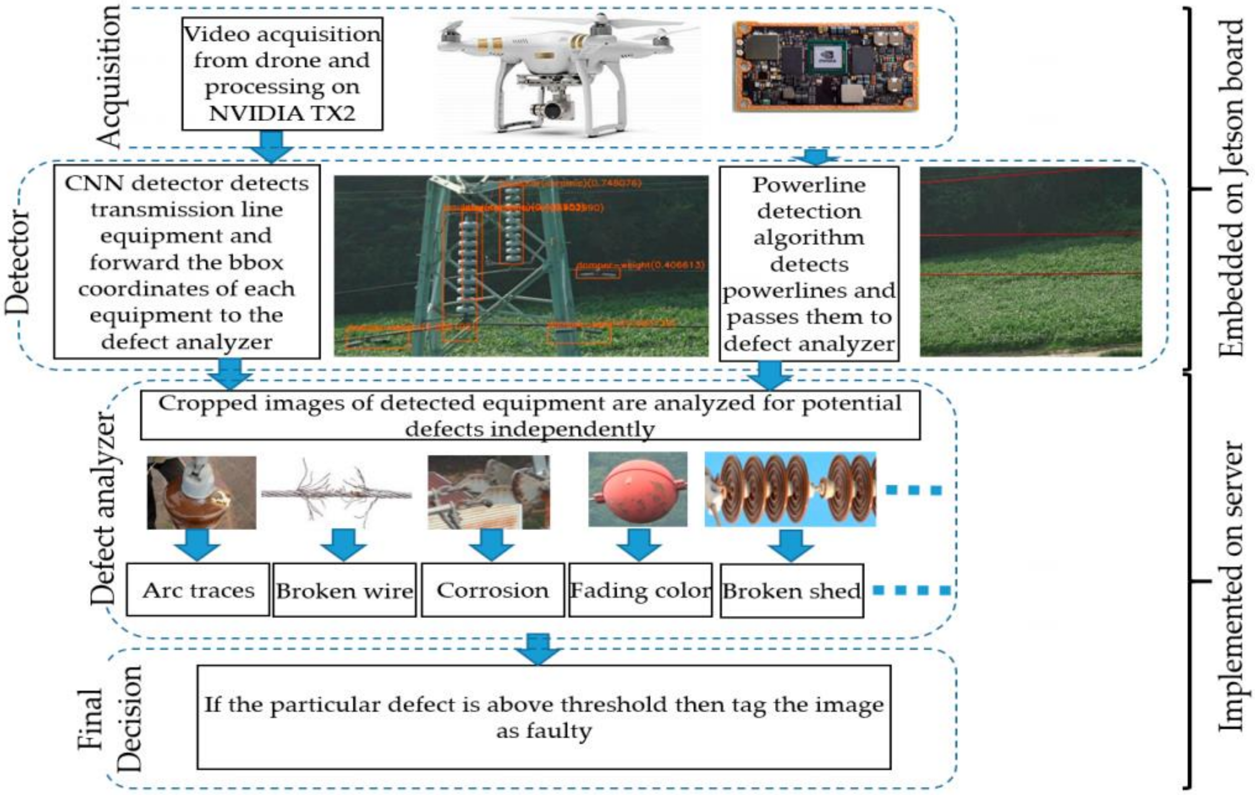

Figure 45: Overall system diagram for UAV based powerline inspection using CNN

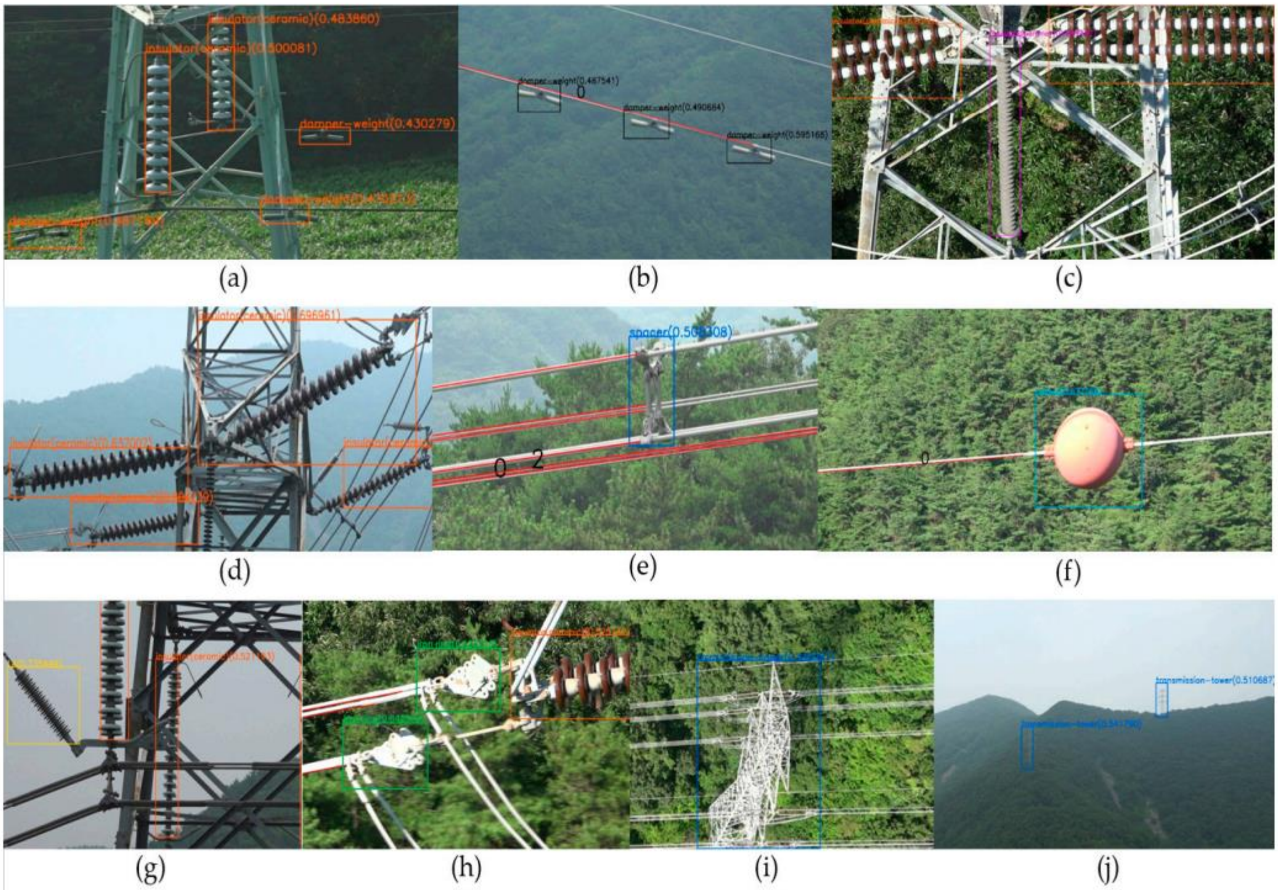

Figure 46: Detection results of the different types of electrical components

4.4 Data Analysis using Kibana

Kibana is a part of Elastic Stack [35], and it functions on top of Elasticsearch offering tools for searching, visualising, and analysing data indexed to Elasticsearch. With Kibana, the indexed data can be both viewed historically and streamed in near real time allowing utilisation of the software stack for monitoring purposes.

The core functionalities of Kibana include search tools and creation of individual visualisations and dashboards. Moreover, Kibana offers embedded visualisations, meaning that visualisations and dashboards created in Kibana can be viewed and interacted with within an external application independent of Elastic Stack.

4.4.1 Dashboards

With Kibana, a user can create customised dashboards to suit their needs for visualising the data that has been indexed to Elasticsearch. Dashboards are combinations of individual visualisations, which offer multiple ways to show the data and data aggregations. Examples of the visualisation types offered by Kibana include graphs, maps, charts, and tables. The visualisations and dashboards created in Kibana are interactive, allowing users to, for instance, filter the data shown in the visualisations.

Figure 47 shows an example view of a dashboard created in Kibana including various visualisations of data indexed in an Elasticsearch instance.

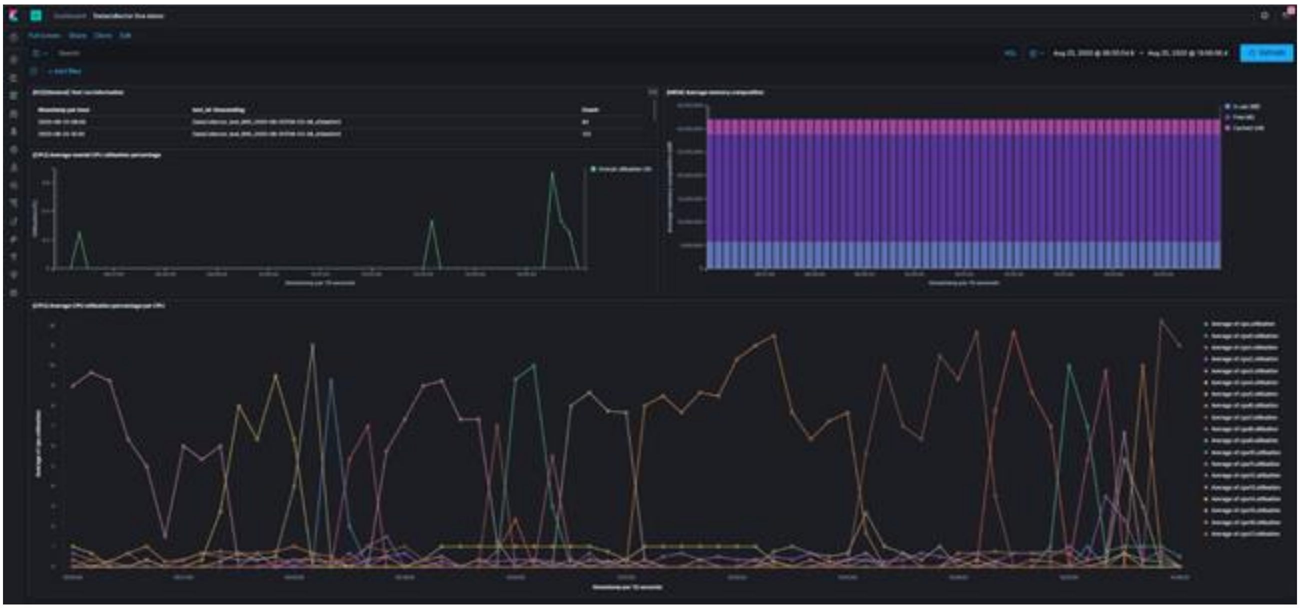

Figure 47: Example dashboard created in Kibana

4.4.2 GIS enhanced analysis

Kibana does also allow to use maps to visualise datasets as the elastic example in the following screenshots show. Of course the actual data sets need to contain geographical information to allow the according visualisation.

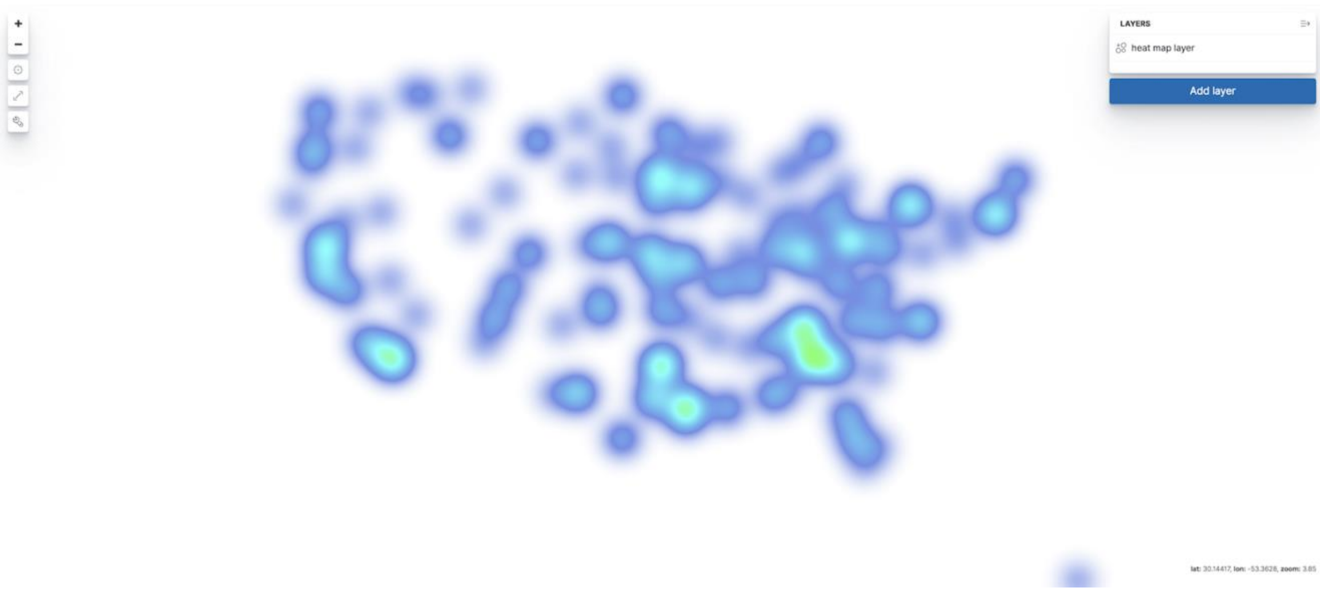

Figure 48: Kibana Heatmap Example

A heatmap as shown in Figure 48 can be used to depict locations with higher densities.

The path visualisation as show in Figure 49 can be used to visualize e.g. drone positions in combination with network performance information. This allows correlated collected KPIs to the geographical area of a trial scenario.

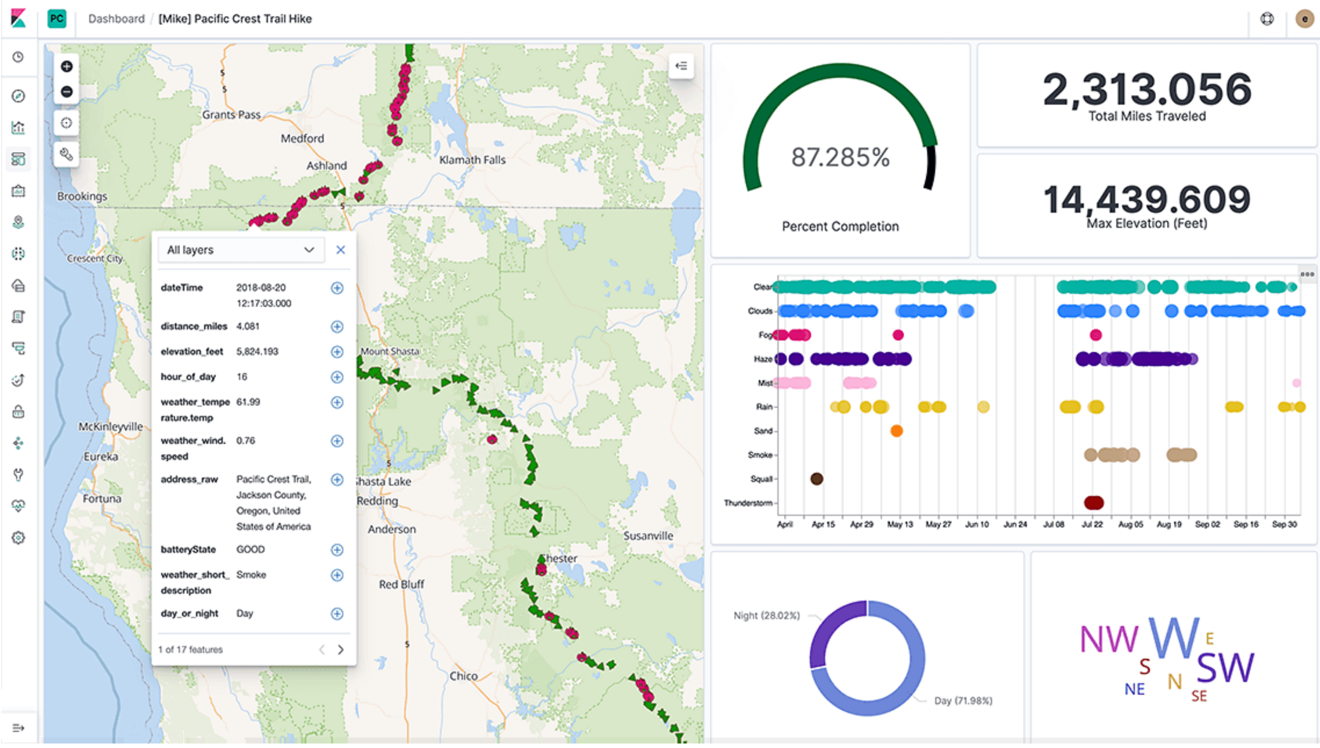

Figure 49: Kibana Path Visualisation Example

4.5 Network coverage visualisation in 3D

During the 5G!Drones project, feasibility tests were conducted in 2020. During the tests, the quality of the 5G network was measured. To plan cellular drone flights, it is important to know the quality of the 5G network in different locations and in 3D. Therefore, a Google Earth 3D map was used to visualise the measurement results (collected via nPerf [36]), as shown in Figure 50. More details can be found in the according WP4 trial reports.

Figure 50: OTE Academy-Speedtest with nPerf at levels 0-5-8-12-15-20-30 meter - 21 October 2020

4.6 Use Case Example: UAV traffic management

A common feature of all uses of cases, both those in the 5G!Drones project and in their practical use, is the need for graphic illustrations necessary for flight planning, supervision of the flight and post flight analyses.

Data visualisation is an interdisciplinary task that combines seemingly distant areas such as data transmission speed, latency, airspace configuration (e.g. that known from manned aviation) and, for example, population density. An additional element complicating the attempt to visualise the components of the system is its multidimensionality. While it is easy to imagine the visualisation of radio coverage in a 3D model, it is much more difficult to show the same space for different users who require different functionalities with different SLA (Service Level Agreement) for the same area. When building a commercial system, the effectiveness and efficiency of visualisation systems should be taken into account.

As a result of the experiments, the aspect of data analysis and visualisation should be divided according to Figure 51. It should be emphasised that data collecting, aggregating and analysing is a continuous process. Everything is changing, from population distribution affecting GRC-type risks, to airspace structures affecting ARC, to terrain affecting radio wave propagation, to the ever-increasing business demands of users.

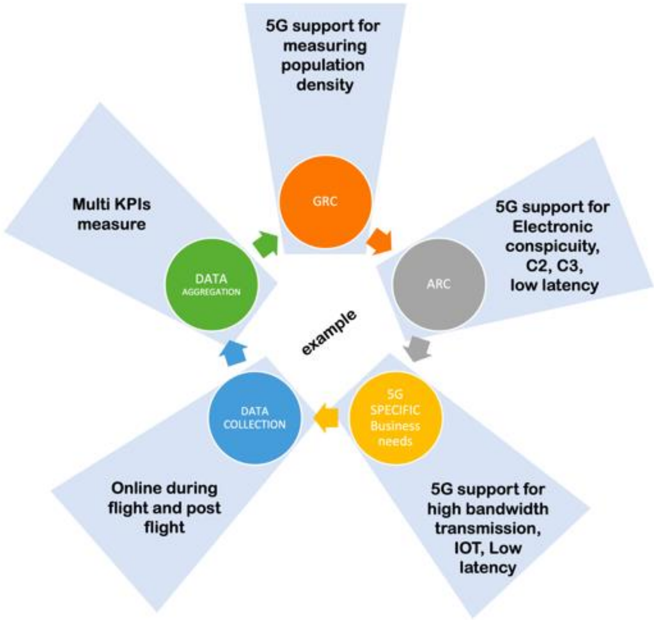

Figure 51: MNO "Self-propelling" data analyses and collection wheel

The process illustrates the self-propelling mechanism in which an increasing number of operations determine the boundary conditions for their performance. Please note that overall capacity of the system is influenced by many factors, including:

- the accuracy of data collection,
- the 4D (length, width, and height in space, plus time) precision of the flights (including algorithms for assigning designated routes - using GRC and ARC),
- effectiveness of anti-collision systems (DSA/DAA) supported by low latency high speed networks
- regulation requirements.

First, the data is collected during the 5G!Drones use cases. They are designed to determine the basic criteria necessary to draw KPIs and provide examples for data visualisation. Second, using the existing UTM systems provided by partners during the Trials (Smart SIS by Frequentis, PansaUTM by Droneradar), the data is illustrated individually. The next step will be an attempt to visualise data aggregated from many sources.

4.6.1 Data Collection

For the 5G!Drones project, the following data are collected, or should be considered during the test flights. Their exact specification will be successively supplemented until the end of the project.

- UAS specific
  - Telemetry data from UAS
  - C2/C3 link continuity
  - Photos and videos for specific scenarios (e.g. photos for creating 3D map etc.)
- 5G specific
  - Service provided (network coverage) on selected altitude / height
  - Cell handover time and mapping

  - Availability of the EDGE services
- GRC (Ground Risk Class) specific
  - People population distribution
- ARC (Air Risk Class) specific
  - Dynamic radio coverage reconfiguration (beam forming)

4.6.2 Data Analysis

The purpose of collecting, analysing and visualising data is ultimately the ability to achieve business goals. These, in turn, are determined by KPIs that allow problems to be categorised. Below is a list of KPIs combining technical challenges with the capabilities of modern telecommunications networks.

- Number of scenarios which require Edge
- Number of scenarios which require NS (Network Slicing)
- Number of scenarios which require 5G high bandwidth
- Number of scenarios which require low latency
- Number of scenarios which require high number of devices in certain volume
- Number of scenarios which require 5G network reconfiguration (beam forming)
- Post flight trajectories analyses, which will define demand for 5G coverage
- Number of cases where lost of 5G signal occurred in areas where 5G signal should be provided
- Number of cases where lost of 5G signal occurred do to overload
- Number of cases where NS for prioritisation is required
- Time needed to reconfigure 5G network according to business needs
- Time needed to reconfigure 5G network for life saving missions
- Time to re-establish 5G connection in the airspace
- Time to handover 5G connection in the airspace in relation to the speed of the drone
- Time to handover 5G connection in the airspace in relation of the speed of drone in case where no handover mapping was present
- Time to allocate NS for 5G high bandwidth in relation to the route distance
- Number of reported incidents do to 5G network failure
- Number of cases where 5G network could be overlapped with 3G/4G/ communication
- Number of cases where edge was needed to perform certain flights
- Number of planned and unplanned service outages
- Number of cases where 5G scalability limits has been faced
- Number of 3rd party data providers integrated

4.6.3 Data Visualisation

Data visualisation should be unambiguous and compliant with the rules of modern interface. The 5G!Drones project utilise the experience of many partners. Below are examples of illustrations of many topics, all of which are necessary for the preparation and execution of a flight. Please treat them as examples or inspiration.

ANSP (Air Navigation Service Provider**), UTM (**Unmanned Traffic Management and) **USSP** (U-Space Service Provider) **perspective**

The following pictures show examples of different visualisations from the UTM system.

In Figure 52 a 3-dimensional map view is shown which enhances the list display of Operation Plans in an Uspace.

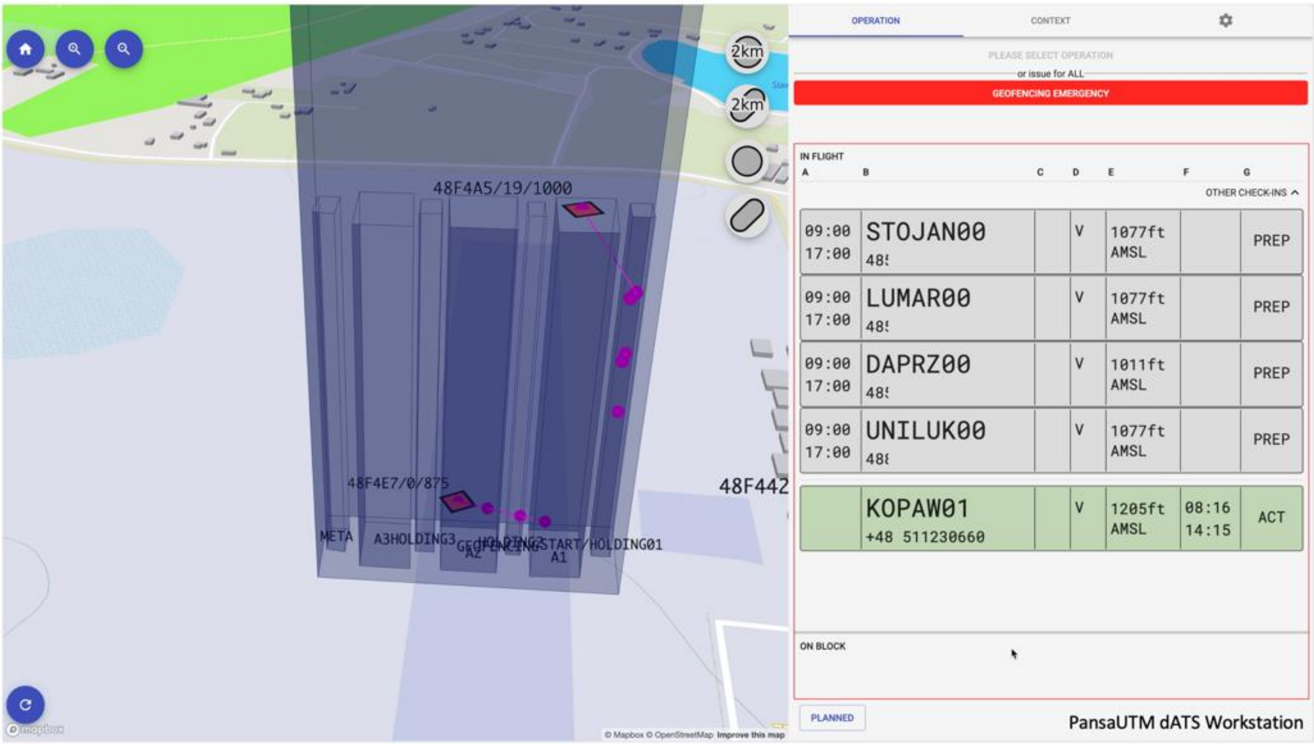

Figure 52: UTM 3D view

Figure 53 shows the visualisation of drone telemetry data on a map view. The rectangle is the actual area the drone should use, it can be seen on the bottom that the drone violated the assigned geofence – red path.

Figure 53: Example of geofencing violation view

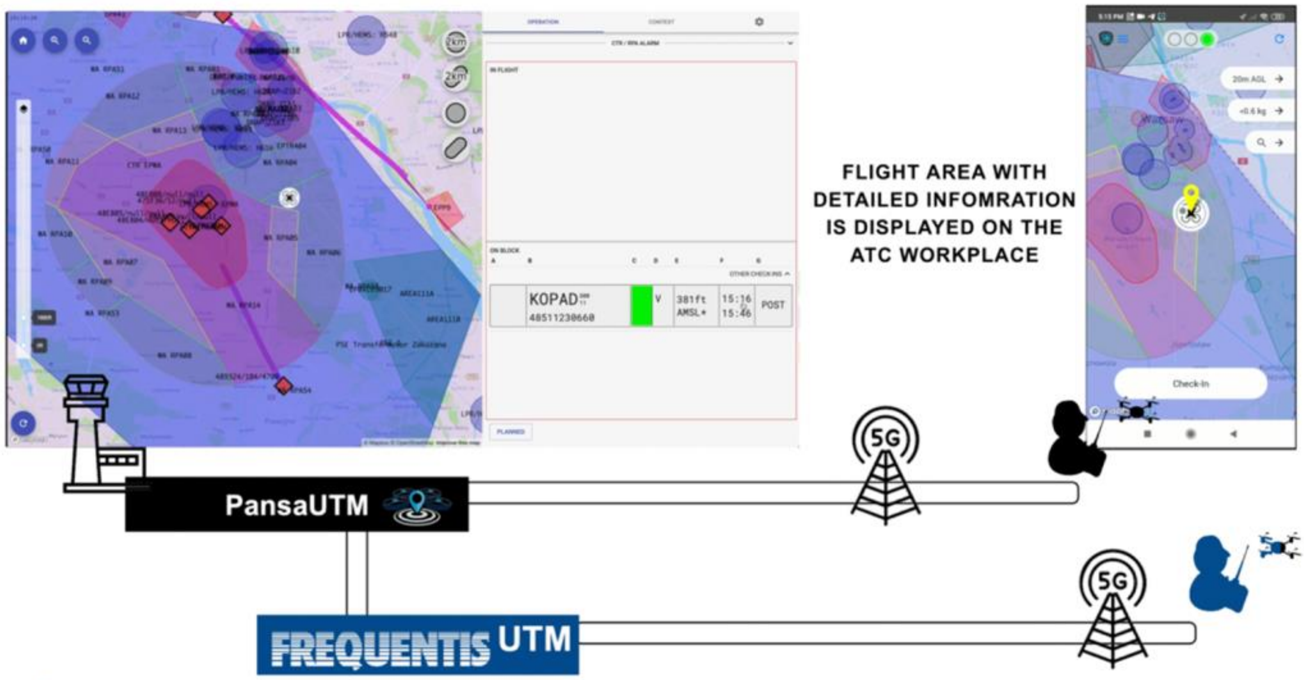

Figure 54: UTM interconnection and MNO support for inter USSP providers

Figure 54 shows the concept of two interconnected USSP providers sharing ATM/Uspace geo-information with end-users via 5G MNO. Also, in the figure is the graphical representation of geozones for both the end-user as well as the Air Traffic Controller. The full movie of the use case example is available **on online**. In Figure 55, the different visualisation of the same geozone dataset can be seen.

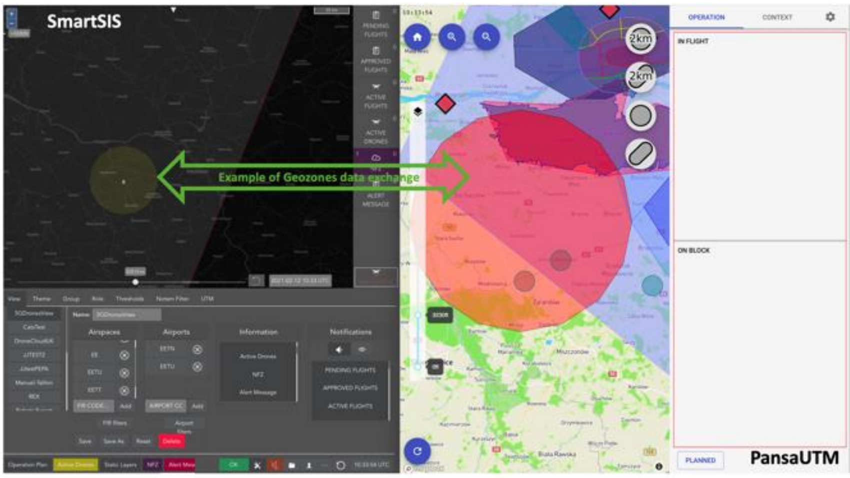

Figure 55: Example of Inter USSP connection - Dynamic Geozone visualisation

Figure 56 shows the map based visualisation of drone telemetry within two different USSP applications. The actual telemetry data is shared between the applications and kept in sync to have a common picture of the Uspace situation.

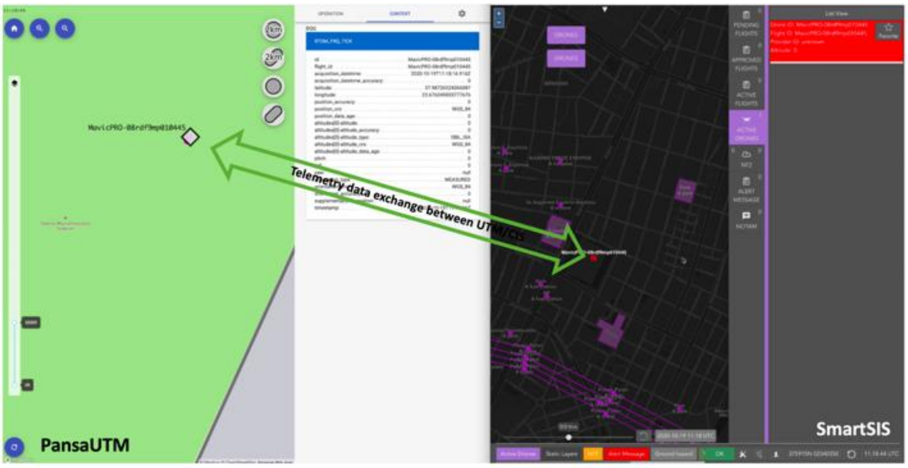

Figure 56: Inter USSP connection: telemetry data exchange (telemetry medium: 5G network)

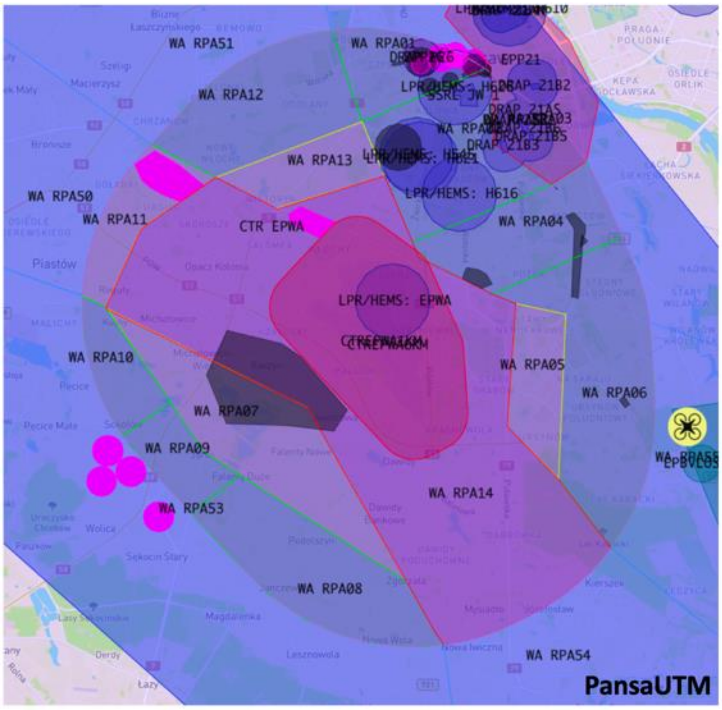

Figure 57: Example of dynamic airspace reconfiguration

Figure 57 show geozones including dynamic airspace reconfiguration in more detail - the magenta polygons represent NoFlyZones which are allocated dynamically. Grey polygons represent planned missions, the yellow icon depict an actual drone in operation.

ARC – Air Risk Class

In Figure 58: Example of geozone visualisation for UAS operator different air risk classes within a mobile phone application for UAS operators can be seen (source: Droneradar mobile app). This includes different geozones and air risk classes.

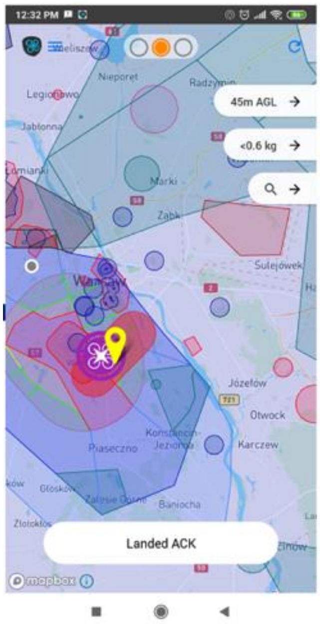

Figure 58: Example of geozone visualisation for UAS operator

Ground risk visualisation

Figure 59: Population density shows population density based on data derived from MNO. This data may be used to define ground risk classes.

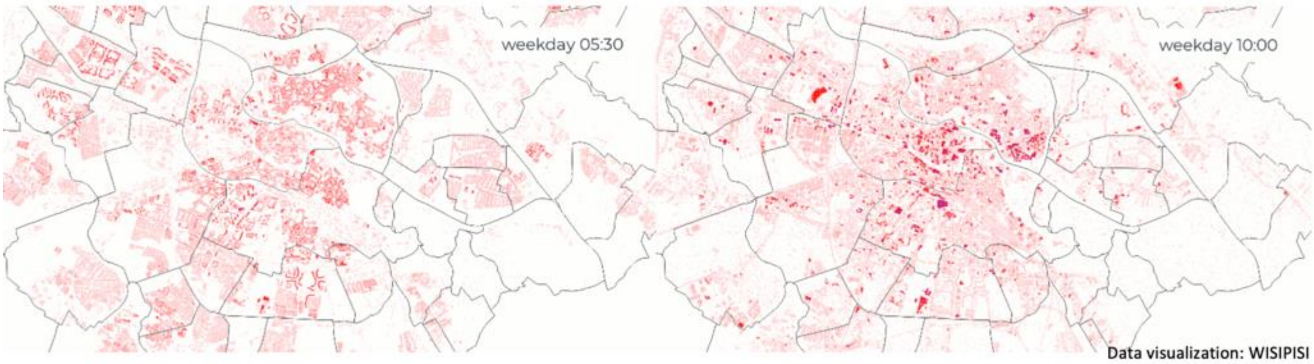

Figure 59: Population density from MNO

Figure 60: Statistical distribution example visualises statistical distribution compliant with the population table specified for JARUS SORA GRC Annex F [37].

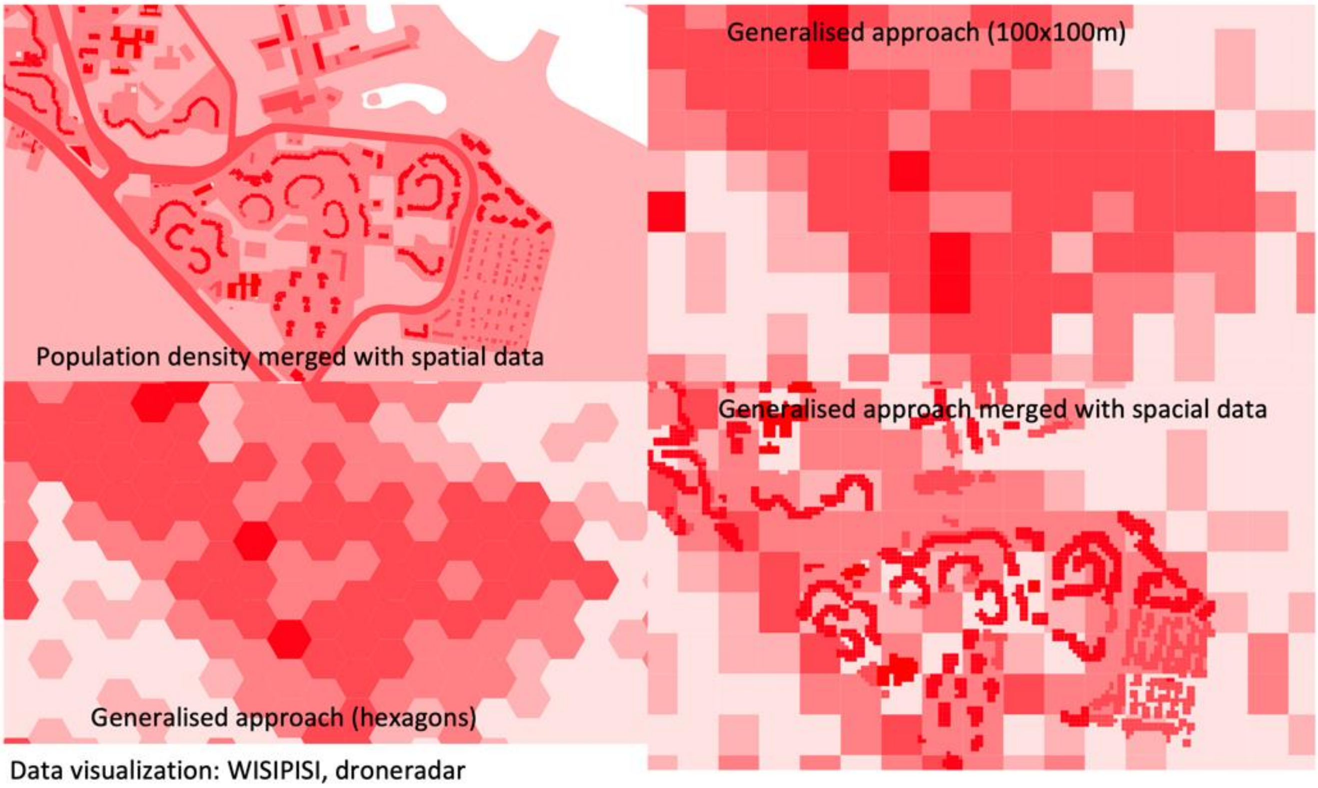

Figure 60: Statistical distribution example

Figure 61: Risk Heat Map - visualisation of the associated risks shows risk information already combined with UTM geozones.

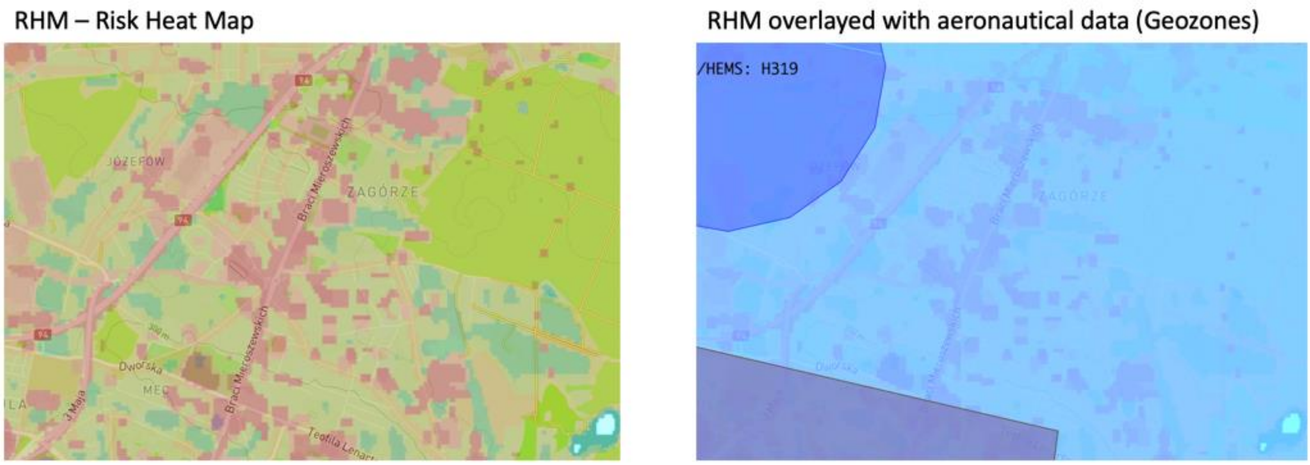

Figure 61: Risk Heat Map - visualisation of the associated risks

5G specific data visualisation

In Figure 62: Example of 5G radio coverage the visualisation of 5G radio coverage data is shown based on a specific selected altitude/height.

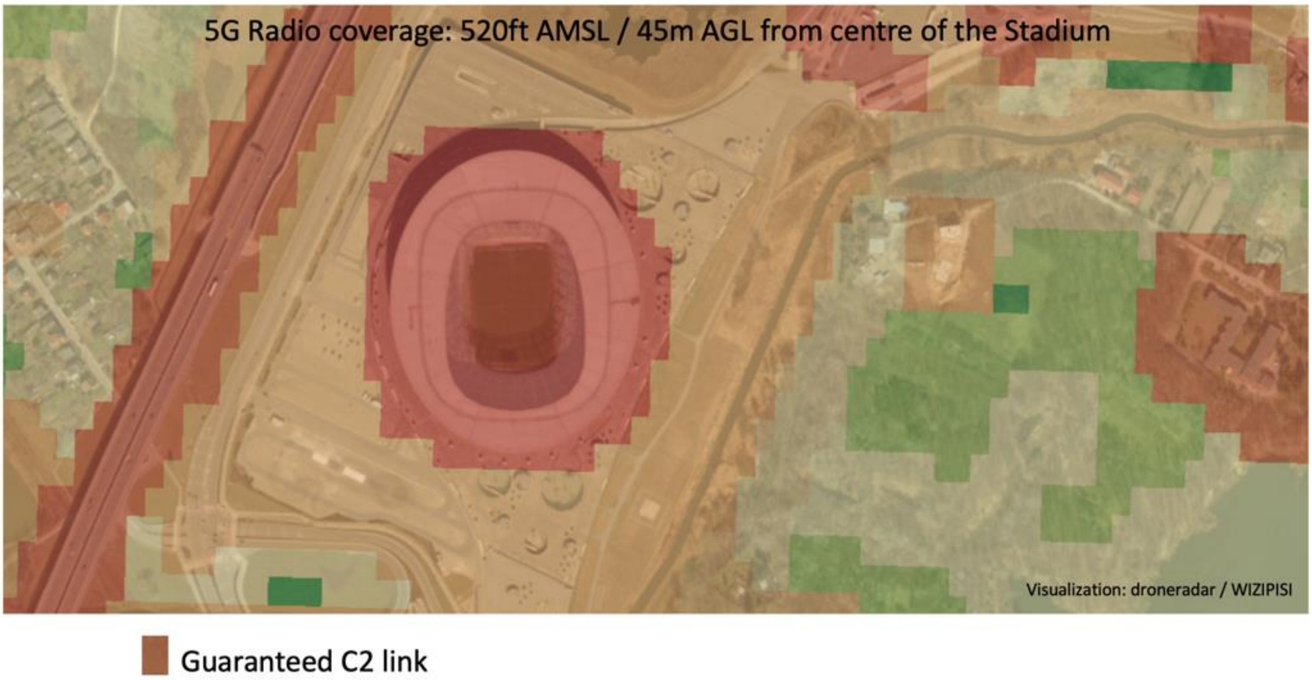

Figure 62: Example of 5G radio coverage

5 CONCLUSION

By finalising the description of the data collection concepts within task T2.4 and reporting it within this document, other tasks within WP2 and WP4 can continue to finalise the development of needed components and move on with integration to prepare successful drone trials.

The work within this task showed that the diversity of components and involved partners, coming from very different backgrounds and providing lots of different expertise, leads to very broad set of requirements and as a result to diverse approaches to fulfil these requirements. The inability of in-person meetings during 2020 and 2021 did increase this issue. Nonetheless, the partners managed to provide a complete and comprehensive solution to the goals set within this task. The presented mechanisms and concepts will make sure to allow collection and analysis of all needed parameters and test results for complete reports of the trials.

Chapter 2 starts with providing the reader with a general overview and continues describing business aspects as well as the data protection topic.

Chapter 3 focused on the Data Collection part by describing Facility Collection tools and the Trial Controller side of data collection.

Chapter 4, Data Visualisation and Analysis covered the academic side of data analysis with a special focus on machine learning topics as well as describing the practical approach of data analysis and visualisation via standard software. Also, an exemplary description for collection, analysis and visualisation of a UAV scenario was provided.

Of course, each of the individual chapters and subchapters would allow to go into much more detail, but the overall goal of the report to give a comprehensive overview of the data collection, analysis and visualisation topic within the 5G!Drones project is completed.

Please note, the software components described within this document will be released as part of D2.6.

6 References

[1] 5G!Drones D2.1, "Initial definition of the trial controller architecture, mechanisms, and APIs," H2020 5G!Drones project, 2020.

[2] 5G!Drones D2.2, "Initial implementation of the trial controller," H2020 5G!Drones project, 2021.

[3] 5G!Drones D2.4, "Definition of the trial controller architecture, mechanisms, and APIs," H2020 5G!Drones project, 2021.

[4] 5G!Drones D4.3, "Trial plan," H2020 5G!Drones project, 2021.

[5] 5G!Drones Grant Agreement, "Grant Agreement number 857031," H2020 5G!Drones project, 2019.

[6] 5G!Drones D1.1, "Use case specifications and requirements," H2020 5G!Drones project, 2019.

[7] IEEE 802, "Time-Sensitive Networking (TSN) Task Group," IEEE, [Online]. Available: https://1.ieee802.org/tsn/. [Accessed 2021].

[8] wikipedia, "Precision Time Protocol," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Precision_Time_Protocol. [Accessed 2021].

[9] 5G!Drones D6.1, "Data management, quality and risk management plan," H2020 5G!Drones project, 2019.

[10] europa, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016," [Online]. Available: https://eur-lex.europa.eu/legal-content/en/TXT/?uri=celex:32016R0679. [Accessed 2021].

[11] 3GPP, "Vocabulary for 3GPP Specifications," 2021. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=558. [Accessed 2021].

[12] Qosium, "Qosium web page," [Online]. Available: https://www.kaitotek.com/qosium. [Accessed 2021].

[13] Qosium, "Qosium Documentation," 2021. [Online]. Available: https://www.kaitotek.com/resources/documentation. [Accessed 2021].

[14] Google, "TensorFlow," 2021. [Online]. Available: https://www.tensorflow.org/. [Accessed 2021].

[15] Keras Special Interest Group, "Keras," 2021. [Online]. Available: https://keras.io/. [Accessed 2021].

[16] Microsoft, "Microsoft Excel," 2021. [Online]. Available: https://www.microsoft.com/de-at/microsoft-365/excel. [Accessed 2021].

[17] "scikit-learn," 2021. [Online]. Available: https://scikit-learn.org/. [Accessed 2021].

[18] pytorch, "pytorch website," [Online]. Available: https://pytorch.org/. [Accessed 2021].

[19] "MavLink," 2021. [Online]. Available: https://mavlink.io/en/.

[20] Hamed Hellaoui, Miloud Bagaa, Ali Chelli and Tarik Taleb, "Joint Sub-Carrier and Power Allocation for Efficient Communication of Cellular UAVs," *IEEE Trans. Wirel. Commun,* vol. 19, no. 12, pp. 8287 - 8302, 2020.

[21] "Gazebo," 2021. [Online]. Available: http://gazebosim.org/.

[22] ESnet, “iperf,” 2021. [Online]. Available: https://iperf.fr/.

[23] Keysight, “NTH00000B Nemo Handy Handheld Measurement Solution,” 2021. [Online]. Available: https://www.keysight.com/us/en/product/NTH00000B/nemo-handy-handheld-measurement-solution.html. [Accessed 2021].

[24] Fukuda Lab, “MAWILab,” 2021. [Online]. Available: http://www.fukuda-lab.org/mawilab/. [Accessed 2021].

[25] G. R. P. O. Juliette Dromard, “Online and Scalable Unsupervised Network Anomaly,” 2021. [Online]. Available: https://hal.laas.fr/hal-01406273/file/ORUNADA_Final_version.pdf. [Accessed 2021].

[26] RIPE, “RIPE Atlas,” 2021. [Online]. Available: https://atlas.ripe.net/. [Accessed 2021].

[27] MAWI, “MAWI Working Group Traffic Archive,” 2021. [Online]. Available: https://mawi.wide.ad.jp/mawi/. [Accessed 2021].

[28] Stratosphere Lab, “The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic.,” Image Net. [Online]. Available: https://www.stratosphereips.org/datasets-ctu13. [Accessed 2021].

[29] “Microsoft AI Documentation,” [Online]. Available: https://docs.microsoft.com/fr-fr/azure/machine-learning/concept-deep-learning-vs-machine-learning.

[30] F. Barreiro-Lindo, “Biologie : la muse des réseaux de neurones,” 2019. [Online]. Available: https://campus.hesge.ch/blog-master-is/biologie-la-muse-des-reseaux-de-neurones/.

[31] Pydata, “Pandas,” 2021. [Online]. Available: https://pandas.pydata.org/. [Accessed 2021].

[32] “Scikit-Learn Documentation,” [Online]. Available: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

[33] “OpenCV Documentation,” 2021. [Online]. Available: https://learnopencv.com/histogram-of-oriented-gradients/.

[34] Image Net, “Large Scale Visual Recognition Challenge,” 2021. [Online]. Available: https://www.image-net.org/challenges/LSVRC/index. [Accessed 2021].

[35] Zhang, Heng, Yang, Wen, Yu, Huai, Zhang, Haijian and Xia, Gui-Song, “Detecting power lines in UAV images with convolutional features and structured constraints,” *Multidisciplinary Digital Publishing Institute - Remote Sensing,* vol. 11, 2019.

[36] Siddiqui, Zahid Ali and Park, Unsang, “A Drone Based Transmission Line Components Inspection System with Deep Learning Technique,” *Multidisciplinary Digital Publishing Institute - Energies,* vol. 13, 2020.

[37] Elastic, “Elastic Stack,” 2021. [Online]. Available: https://www.elastic.co/elastic-stack. [Accessed 2021].

[38] nPerf, “nPerf Speed Test,” 2021. [Online]. Available: https://www.nperf.com/. [Accessed 2021].

[39] Jarus, “Jarus guidelines on Specific Operations Risk Assessment,” 2019. [Online]. Available: http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_v2.0.pdf. [Accessed 2021].

[40] 5G!Drones D1.4, “Report on UAV business and regulatory ecosystem and the role of 5G,” H2020 5G!Drones project, 2020.