PROJECT "LOCUS": LOCalization and analytics on-demand embedded in the 5G ecosystem, for Ubiquitous vertical applicationS

Grant Agreement Number: 871249
(https://www.locus-project.eu/)

DELIVERABLE D5.1

"Design and implementation of virtualization technologies and pattern recognition mechanisms for physical analytics", preliminary version

Deliverable Type:	Report
Dissemination Level:	Public
Contractual Date of Delivery to the EU:	31/12/2020
Actual Date of Delivery to the EU:	13/01/2021
WP contributing to the Deliverable:	**WP5 – Localization & Analytics for New Services**
Editor(s):	IBM, Joseph Antony
Author(s):	CNIT, Andrea Conti, Domenico Garlisi; IBM, Gabriele Ranco, Joseph Antony; INC, Athina Ropodi, Ioannis Filippas, Aristotelis Margaris; NEC, Gurkan Solmaz, Giuseppe Sircusano; NXW, Giacomo Bernini, Elian Kraja, Francesco Bocchi; SAMSUNG, Tomasz Mach; TEI, Marzio Puleri, Stefano Stracca; UMA, Emil J Khatib, Raquel Barco Moreno;

	VIAVI, Takai Eddine Kennouche.
Internal Reviewer(s):	CNIT, Domenico Garlisi; Nicola Blefari Melazzi INC, Kostas Tsagkaris; NEC, Gurkan Solmaz; IBM, Ramy Mohamed, Rosemary Devlin, Ryan Gallagher.
Short Abstract:	The goal of this deliverable is to describe the design and implementation of the virtualization techniques as well as the pattern recognition and machine learning algorithms for physical analytics involved in the deployment of the proposed new services.
Keyword List:	Spatio-Temporal Data Analytics, Machine Learning Pipelines, Virtual Network Functions, Deployment Functions.

Executive Summary

The goal of LOCUS is localization, together with analytics, and their combined provision 'as a service'. LOCUS will offer machine learning and/or deep learning-based solutions, and analytics leveraging the location and other contextual information. This deliverable (D5.1) covers the design and the implementation details of the different use cases involved in the localization and analytics for new services in LOCUS, as defined in the deliverable D2.1 [1], studied and developed in WP5.

The functionalities in WP5 are grouped under six use cases (NSE-UC1 to 6). The first use case (NSE-UC1) on efficient flow monitoring and management in large venues focuses on two functionalities: 1) to identify crowd mobility patterns to provide location analytics such as possible visitor paths, Points of Interest (indoor / outdoor / hybrid), and 2) to classify people movement behaviour relative to "geofenced" perimeters according to security/safety/other objectives in an indoor monitored area. The second use case (NSE-UC2) on crowd mobility analytics using multi-modal sensors evaluates two functionalities: 1) learning group mobility characteristics using wireless fingerprints, and 2) using multi-modal data for crowd mobility – COVID-19 as a special case. The third use case (NSE-UC3) on vulnerable road user (VRU) addresses two functionalities: 1) VRUs clustering, and 2) Time-to-Collision as a service in V2X. The fourth use case (NSE-UC4) evaluates logistics in a seaport terminal using Automated Ground Vehicles (AGVs). The fifth use case (NSE-UC5) provides analytics on crowd mobility profiles (e.g. Pedestrian, road traffic, railway routes) and predict the near-future traffic by assigning trajectory profiles per user. Due to the on-going COVID-19 pandemic, a new use case (NSE-UC6) on positioning and flow monitoring for controlling COVID-19 has been introduced in WP5. Two functionalities are defined in this use case: 1) Contact Tracing. The goal of this functionality is given an identified case of COVID-19 infection, it traces back the persons to have potentially been in proximity with the positive case within a certain number (to be set) of previous hours/days, 2) Monitoring epidemiological risk flow. The goal of this functionality is to estimate risk factors and their spatiotemporal evolution using epidemiological data combined with the flows of people moving from one area to another area.

This deliverable is a preliminary version, and the objective is to present the design, implementation details, and initial results from the various functionalities defined within each use case and the steps involved in the virtualization and deployment of each functionality. The second version of this deliverable (D5.2) will present the final achieved results, the virtualization infrastructure and the deployment aspects by the end of this project.

Table of Contents

List of Abbreviations

ABBREVIATION	EXPANSION
5G	Fifth generation technology standard for cellular networks
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
API	Application Programming Interface
BT	Bluetooth
CNN	Convolutional Neural Network
COVID-19	Coronavirus disease 2019
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
DTW	Dynamic Time Warping
ETSI	European Telecommunications Standards Institute
FR	Functional Requirement
GPS	Global Positioning System
GRU	Gated Recurrent Unit
HV	Host Vehicle
IoT	Internet of Things
IR	Infra-Red
k-NN	k-Nearest Neighbours
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory
ML	Machine Learning
NSE	New Services
PCA	Principal Component Analysis
POI	Point of Interest
QoE	Quality of Experience
Qos	Quality of Service
RACE	Rapid Action Coronavirus Earth observation tool

RFID	Radio Frequency Identity
RNN	Recurrent Neural Network
SVM	Support Vector Machine
SVR	Support Vector Regression
TULVAE	Trajectory User Linking via Variational Encoder
UC	Use Case
V2P	Vehicle to Pedestrian
VNF	Virtual Network Function
VM	Virtual Machine
VRU	Vulnerable Road User
WiFi	Wireless Fidelity
WP	Work Package

Table Index

Figure Index

1 Introduction

This report (Deliverable 5.1 or shortly D5.1) provides general guidance for exploiting all the aspects related to localization and analytics for new services on top of 5G. This report explains how different analytics methods could be used to discover meaningful patterns and trends from 5G network data, the proposed machine learning (ML) algorithms, selection and evaluation of the various machine learning and deep learning (DL) models, the essential pattern analytic functions required, and the various virtualization functions involved in the deployment of the proposed new services.
This report will be published in two stages:

- First as D5.1 in M13, as a preliminary version, describing the initial plans, design, and initial experimental results of the ML/DL methods, pattern recognition mechanisms, and virtualization technologies from the tasks T5.1, T5.2, and T5.3.
- Next as D5.2 in M26, the final version, describing the completed experiments, results and analysis, and implementation details of the ML/DL methods, pattern recognition mechanisms, and virtualization technologies from the tasks T5.1, T5.2, and T5.3.

This report summarizes the initial inventory of possible data sources, the envisioned analytics methods, proposed ML/DL pipelines for the functionalities of each use case, design and experiments, the initial results, and the implementation details. Describing the experimentation and findings regarding this new data-driven machine-learned paradigm and the virtualization technology involved is the main purpose of this deliverable.

This version describes the initial plans, design, and experiments that will be conducted, both to prove the premise of the data driven approach and as a base for future work. The structure and organization of this document is as follows:

- Section 2 – design of spatio-temporal analytics as virtual network functions. This section describes the design of various ML/DL pipelines, pattern analytic functions, and virtualization functions.
- Section 3 – implementation of spatio-temporal analytics as virtual network functions. This section provides the details about the ML/DL model(s) selection, experiments, evaluations and the implementations.
- Section 4 – Data Privacy. This section presents the data privacy and security problems associated with the location based services.
- Section 5 – Conclusion and next steps. This section will summarize the complete work involved and provide a lead to the future work.

1.1 WP5 Tasks

The activities and execution of the **localization and analytics for new services** are divided into three main tasks as defined in the project proposal:

Task 5.1: Physical Analytics and Pattern Recognition Components

Description: T5.1 will conduct traffic and mobility analysis, by developing and validating machine learning and pattern recognition mechanisms, in particular: unsupervised machine learning for creating knowledge on typical situations and for making predictions; supervised learning (pattern recognition) to identify and handle situations that are close to "known" ones. These mechanisms will be targeted to the generation of insights and predictions regarding the concentration and location of users (and potentially things) in space and time as well as dynamic map creation, anomaly detection and flow tracking utilizing multimodal data. To execute its work, Task 5.1 will leverage the virtualized information specified in T5.2. Moreover, it will deliver packaged algorithms and functions to Task 5.3 to define mechanism for implementing the API system behaviours and intent-based functions.

Task 5.2: Virtualization and ML Technologies for Location-based Services

Description: Task 5.2 will produce foundation functions for the location-based services, by relying on virtualization technologies in order to: (a) specify the appropriate data structures, which will virtualize the sensitive network information; (b) provide appropriate APIs towards the applications (which may be 3rd party ones); (c) create the mechanisms for enabling the update of the high-level data structures, based on the network information. This task outputs intermediate results to both Task 5.1 and 5.3; it will also collect requirements from WP2, WP6 and also other ones also from Tasks 5.1 and 5.3. Based on the API layer developed in T5.3, and on its backend analytics services, advanced applications will be developed, such as: (a) user guidance in the context of retail or even user safety applications in the context of smart venues; (b) warning points of interest (e.g., retail shops, etc.) about upcoming concentrations in time; (c) object localization; (d) sensor placement; (e) flow tracking.

Task 5.3: Virtualized platform for localization & analytics as a service

Description: Task 5.3 will expose to application developers an "API layer" capable of supporting abstract definitions of location related targets and related analyses/data processing actions to be automatically launched in the LOCUS platform (e.g. behavioural patterns analyses, trigger for action, etc.). First, an extensible library of analytics will be developed which covers multiple recurring application needs. These will include but are not limited to dynamic map creations and flow tracking, fusion of spatiotemporal data with multimodal information coming from other sources (e.g., data collected from the Internet), anomaly behaviour detection, and improvement of the raw location estimates provided by the previous two challenges above, and so on. Second, the analytics techniques will be

instantiated in the virtualized platform in common with WP4, by distributing the data collection and analytics functions at the edge/core and leveraging on composition/chaining approaches developed in WP4. The API layer will implement reliable access control mechanisms and federation techniques for analytics running within different domains, to permit exploitation and usage of the data by third-party business players. The API layer described above and on its backend analytics services will ease the development of advanced applications based on the insights and predictions produced by task 5.1 and the virtualized data of task 5.2.

1.2 Revisiting WP5 Use Cases

The functionalities in WP5 are grouped under six use cases as follows:

NSE-UC1: Flow monitoring and management in large venues and dense urban environment

Large venues and transportation hubs (airports, train stations, malls, stadiums, etc.) exhibit the gathering of large crowds, with complex mobility behaviour. The use case LEN-UC2 in WP3 Deliverable D3.1 [2] deals with providing signal processing solutions in such scenarios, NSE-UC1 provides solutions for an efficient flow and resources management to maximise QoE (Queues, logistics, staff, security, etc.).

By using the virtualized location information at the LOCUS localization platform [3], the LOCUS Analytics service will provide the best routes, smart notifications/recommendations adapted to individual preferences (walking, shopping, food, etc.) -when offered- and current flow and contextual information through a mobile application and/or smart panels.

Venue administrators and the transportation companies can be informed, by the LOCUS server, of their clients' situation in near real-time. They can use this information to optimise their embarking operations and reduce risk for last minute call to gate, flight delays because of one passenger, etc. Using location information, the venue will be able to elaborate statistics on people's flow to optimise their organisation and signalling to customers and passengers. By extension, 5G network operators can assess network performance and exploit the LOCUS Analytics service to improve QoS in relation to people's mobility and traffic patterns.

NSE-UC2: Crowd mobility analytics using mobile sensing and auxiliary sensors

LOCUS will perform crowd analytics in urban areas using limited auxiliary sensors such as cameras for benchmarking and improving the accuracy of the estimation, as well as machine learning with training from the historical data. Individual device location data can be leveraged to identify individualistic movement patterns using RNNs. Data manipulation, data fusion, and ML methods offered by LOCUS will be verified. Crowd sizes, group movement behaviours, people flow behaviours, and waiting time analyses will be conveyed via advanced

visualizations. The crowd analytics results aim to optimize smart mobility by improving decision making such as path planning of vehicles by humans or autonomous vehicles.

NSE-UC3: Vulnerable road user

This use case alerts the host vehicle (HV) of approaching Vulnerable Road User (VRU) in the road. HV approaches the VRU along roads that are defined by their lane designations and geometry. The HV should be able to avoid collision with the VRU. Analytics could provide more detailed insight into the characteristics of the system. For example, tracking 'Time to Collision' parameter for different type of VRU users in different weather conditions.

NSE-UC4: Logistics in a seaport terminal using Automated Guided Vehicles (AGVs)

The use case is related to logistics in a seaport terminal by using AGVs, it addresses the aspects related to the verification of the performances of the 5G positioning system in an operating context. The activities of a real seaport terminal are simulated with a virtual environment, created using the professional Unity 3D game engine. This VR environment can be also used as a digital twin of the real operations. The inputs from the 5G positioning system are used by the AGV navigation system to drive it during its shuttling operations.

In this use case, an expert system made using CLIPS is used to manage the terminal operation during the simulated loading/unloading operations. This is connected to a MySQL relational database, including the freights inventory and the AGVs data. The freights and AGVs data include also status and positional information that are used to define the vehicle path. The algorithm A* is embedded into the management control system as a co-function used to calculate on the fly the new trajectory that an AGV should follow to do its mission. The algorithm computes the shortest path avoiding all known obstacles, including also the freights already placed in the area. Once the task and the path to follow are received by the management system, the AGV local controller, using the received trajectory as reference, computes the next hop movement periodically based on the feedback provided by the 5G positioning system. The AGV implements a nonlinear fuzzy controller to guarantee a fast step response with a small overshoot both for long and short movements.

NSE-UC5: Transportation optimization based on identification of traffic profiles

This use case involves the abstraction of location information at a large scale in an outdoor area. Given this outdoor setting, where various high or low traffic streets, avenues and motorways as well as train routes and pedestrians exist, a variety of different mobility profiles emerge. This use case will enable flexible aggregation of low-level anonymized positioning and velocity information and will offer an abstracted view of location-based data with the purpose

of monitoring. The LOCUS platform will take the necessary steps to exploit its capabilities to satisfy use case requirements, offering services such as (a) identifying different mobility profiles through an augmentation and fusion process and (b) extraction monitoring options to users through an App/Dashboard. The App itself could potentially be used by state entities like traffic police in order to enhance decision processes and could be further expanded to include various functionalities, e.g. near-future predictions of traffic in the selected area, detection of any anomalies/incidents and so on.

NSE-UC6: Positioning and Flow Monitoring for Controlling COVID-19

We are living in a critical time due to the COVID-19 pandemic. In such a condition, the researchers of LOCUS have asked themselves "How can we help? Which tools can we offer to enhance the level of health safety in the restarting phase or to make our countries ready to prevent any future waves?" On March 19, 2020, the European Data Protection Board issued a statement emphasizing that member states should attempt to do what they can with anonymized data but notes that they have the power to "introduce legislation to enable the processing of non-anonymized location data where necessary to safeguard public security", including the location tracking of individuals, under strict privacy safeguards. As a matter of fact, each country is trying to react as fast as possible with ad-hoc solutions to track people movements in accordance with privacy rules.

Given the fact that the COVID-19 crisis will last months, and most countries are just entering the so-called Phase 2, we aim to propose solutions in this area with an effort to put together knowledge and expertise at different levels. With this spirit, LOCUS proposes the use case on location-awareness and analytics to control COVID-19, as described below. In addition, LOCUS believes that a better architectural design of cellular networks is needed for the general case of pandemic events. 5G has the opportunity to develop privacy-preserving secure contact tracing solutions and people flow monitoring solutions, using both LTE/NR 3GPP-based and non-3GPP based infrastructures (also referred to as RAT-dependent and RAT-independent). The proposed solution will prevent abuse of data through i) minimization contacts storage, ii) anonymization procedure, iii) encryption algorithm, and iv) dismantling after a period of time.

1.3 Review of the State of the Art

This section provides a brief review of the recent research highlighting the state-of-the-art methods related to the WP5 use cases.

1.3.1 NSE-UC1: Flow monitoring and management in large venues and dense urban environment

Monitoring and managing the crowd flow in an efficient way either in an indoor or outdoor environment is very challenging. There has been a lot of recent research on crowd mobility and urban sensing to study crowd dynamics, crowd trajectories/movement patterns, to estimate crowd count and density, and even to detect the crowd events and behavioural anomalies mainly using video data [4]. This use case mainly focuses on detecting the crowd mobility patterns and to provide advanced analytics based on the geolocation (spatio-temporal) data collected from the mobile network and the LOCUS platform. Many diverse techniques and data driven approaches using ML & DL algorithms have been proposed for mobile crowd sensing [5], urban flow monitoring and crowd mobility analytics [4]. The state-of-the-art for these focuses on spatio-temporal clustering methods [6] and trajectory predictions using RNNs with GRUs/LSTMs [7], [8], CNNs and hybrids [9], auto-encoders & variants [10]. For spatio-temporal clustering the well-known and successful techniques include k-means clustering and its variants such as k-medoids [11] and k-paths fast large-scale trajectory clustering [12] which reduces the time complexity for distance computations and improves the overall trajectories clustering efficiency. Also, density-based clustering algorithms such as DBSCAN and its variants like ST-DBSCAN [13] have achieved considerable success in spatio-temporal clustering.

RNNs are widely used for time series modelling and are designed to recognize the sequential characteristics and use patterns to predict the next likely scenario. However, they suffer from short-term memory due to the issue of vanishing gradients. GRUs and LSTMs are extensions of RNNs that have shown considerable success in spatio-temporal data predictive learning and representation learning [10]. For example, a predictive RNN [14] with a new spatio-temporal LSTM as its core (ST-LSTM) has achieved better results in spatio-temporal trajectory predictions. Social LSTM [8] is a highly successful LSTM model that is recently used to predict human trajectories in crowded spaces taking into account the social conventions and common-sense rules that humans typically utilize as they navigate in shared environments. LSTM models are good at handling sequence data while CNN models are effective when capturing the spatial correlation in the image like matrices. The hybrid model that combines RNN and CNN can capture both the spatial and temporal correlations of spatio-temporal data [10]. Another recent successful approach for pedestrian trajectory prediction [7] leverages the self-attention mechanism and transformer-based convolution mechanism and introduced a spatio-temporal graph transformer framework. Auto encoders and stack auto encoders are also used for crowd flow forecasting and spatio temporal predictions in particular to learn low-dimensional latent feature coding [10]. For example, TULVAE (trajectory user linking via

variational autoencoder) a generative model to mine human mobility patterns, which aims at learning the implicit hierarchical structures of trajectories has been successful in geo-tagged social media data [15].

1.3.2 NSE-UC2: Crowd mobility analytics using mobile sensing and auxiliary sensors

Crowd mobility analytics and mobile crowd sensing have reached an unprecedented growth and have seen new trends with the advent of smart communication devices and technologies such as internet of things (IoT) and big data. Recent studies report that along with mobile phones the ubiquitous technologies that include Bluetooth, Wireless fidelity (WiFi), Radio frequency ID (RFID), Global positioning systems (GPS), optical wireless communication (infrared or IR devices), video surveillance, and social media are employed as multimodal sensors to collect data on human activities from the urban space [4]. The acquired multimodal data are processed and mapped to provide advanced analytics on crowd mobility and urban sensing. A recent study [16] provides great insights on crowd mobility analytics and has analysed the crowd mobility models comprehensively for various scenarios such as pedestrian walk, with vehicle use, social network based, and human mobility in disasters. Many previous studies related to crowd mobility analytics (specifically group detection) fall into one of these three categories of detections: 1) video-based, 2) wireless activity-based, 3) using data from smartphone apps or social networks [17]. In video-based group detection, various studies leveraged video footages to extract crowd information and some studies focus on detecting groups. Hierarchical [18] and correlation [19] clustering approaches are proposed to detect groups by clustering movement trajectories extracted from video footage. There are recent studies related to “device-free” wireless detection of people in indoor environments using WiFi [20] [21] and Bluetooth scanners [22] [23]. There exist studies [24] [25] that use smart phone sensors for tracking group behaviours based on the fusion of multimodal data from accelerometer, barometer, and Wi-Fi location. Group In [17], a wireless scanning system to detect groups in indoor or outdoor environments is leveraged for this use case.

1.3.3 NSE-UC3: Vulnerable road user

There is an increasing attention towards research on Vehicle-to-Pedestrian (V2P) communication systems and these cater to different vulnerable road user groups (VRUs). Also, these V2P systems employ different communication technologies, and use different mechanisms to interact with the users [26].

As the initial research and standardization work on Intelligent Transport System (ITS) in ETSI, 3GPP and 5GAA, was mostly vehicle focused, user clustering was mostly explored in the context of the vehicle grouping. One example of such approach is vehicle platooning

application (see e.g. ETSI TR 103 298) [27] implemented by using vehicle-to-vehicle communications (V2V) means. In the platoon, first vehicle is driven manually or automatically and following vehicles are controlled by using V2V. Platooning use case has been discussed mostly for trucks where significant energy savings can be achieved through reduced air drag. Platooning reduces fuel consumption, which is particularly pronounced for heavy duty vehicles and research has shown that up to 15% of fuel can be saved on average over a three-truck platoon, implying a clear reduction in CO2 emissions. In the following ITS system evolution (e.g. Release 2 version of ETSI ITS standard), Vulnerable Road User safety was studied recently as an important part of the future connected car ecosystem. Considering the upcoming integration of V2X capability in the future mobile devices chipsets resulting in the increasing proliferation of VRU devices in the transport system, it is prudent to study grouping mechanisms allowing further performance improvements and VRU clustering could be a natural first step.

Expansion of cellular 5G technologies into new verticals such as automotive and connected vehicle applications enables new business opportunities (not directly available in previous generations of communication technologies due to their limited performance), including mission-critical services addressing road users' safety in the increasingly complex ecosystem including transport industry players, in addition to the legacy communication industry. Introduction of V2X communication capabilities and other 5G system improvements allows improved location accuracy, low latency, high reliability and data analytics to synergistically address new user requirements. New mission critical applications and services may address Business-to-Business (B2B) or Business-to-Business-to-Customer (B2B2C) relationships potentially introducing liability (and corresponding commercial penalty clauses) for Mobile Network Operators when unpredicted service quality issues are observed. Road safety domain and related new KPIs such as 'Time to Collision' parameter standardized in ETSI ITS is a good example of a measurable parameter which could be used to proactively estimate collision risk on the road, enabling new services and value generation (with support of new 5G capabilities) in related businesses such as traffic management, transport network planning and insurance industry.

1.3.4 NSE-UC5: Transportation optimization based on the identification of traffic profiles

Various Machine Learning (ML) and Deep Learning (DL) methods have been used for the analysis of spatio-temporal data (e.g., trajectory, velocity and other related information) in the context of transportation systems, i.e. for prediction of traffic and/or traffic congestion and other applications. This UC refers to the enhancement of transportation, whether by understanding the traffic patterns, e.g., common trajectories, velocities and/or user profile

(pedestrian/car and so on) or predicting said patterns for the near future. The applications may vary based on the datasets used and the type of information offered. Additionally, it should also be viewed as a UC that (a) will utilize information available to the LOCUS Platform, (b) investigate the advanced analytics that can be deployed within the LOCUS Platform and then be exposed to 3rd party applications, which in turn will utilize these results for traffic monitoring and optimization. For this reason, the analysis focuses on the true operator dataset that is available within the consortium, i.e. the OTE dataset, and with the anonymization restrictions that would apply in order to provide this service. Lastly, the techniques used could apply to other UCs as well.

Spatio-temporal traffic data can be highly complex and dynamic, especially for large-scale urban areas. Advances in data collection have inevitably led to the use of data-driven techniques, based on statistics, ML and ultimately DL. These techniques vary significantly, based on the data and applications. Indicatively, Asif *et al.* [28], employ unsupervised learning methods such as k-means clustering, principal component analysis, and self-organizing maps, to mine spatiotemporal performance trends and then use a Support Vector Regression (SVR) model for the prediction for a large interconnected road network and for multiple prediction horizons. Other exploratory analyses involved the use of using a modified nonnegative matrix factorization algorithm for identifying spatiotemporal traffic patterns [29]. Unsupervised techniques (being subject to interpretation) -and more commonly clustering techniques- have indeed been a significant part of dataset exploration, visualization and identification of traffic patterns and have been widely used and shown to give good results [6]. As an example, in [30] authors clustered data points based on spatial relations and then concatenated these clusters based on temporal relations. In the last years, however, DL techniques have been gaining ground. More specifically, in [31] deep Restricted Boltzmann Machine and Recurrent Neural Network (RNN) architecture is utilized to model and predict traffic congestion evolution based on Global Positioning System (GPS) data from taxis. Focusing mostly on trajectory prediction as a significant chapter of this UC, RNNs and Long Short-Term memory (LSTM) networks -made for sequence prediction- can fit the aforementioned data for transportation-related use cases. An LSTM model which can learn general human movement and predict their future trajectories is proposed in [8] with good results when tested with various datasets. Park *et al.* [32] analyzed the latent patterns in the past trajectories using an LSTM-based encoder and predicts the future trajectory/sequence using the LSTM based decoder. Lastly, Convolutional Neural Network (CNN)-based methods have been used, for example to learn traffic as images and predict network-wide traffic speed with a high accuracy [31].

1.3.5 NSE-UC6: Positioning and Flow Monitoring for Controlling COVID-19

Due to the pandemic situation, the European Commission is taking resolute action to reinforce public health sectors and mitigate the socioeconomic impact in the European Union. Digital technologies represent the principal asset to respond to the coronavirus and to keep people safe. European Commission digital strategy includes the monitor of the spread of the coronavirus through space earth observation and national contact tracing solutions [33].

1.3.5.1 Monitoring of coronavirus spread using satellite data

The EU Space Program, notably through its Earth Observation component, Copernicus, and its satellite navigation system, Galileo, offers free and open data/information that helps monitor the impact of the coronavirus outbreak. EU satellites have been monitoring traffic congestion and mapping medical facilities, hospitals and other critical infrastructures. Data that is collected from satellites, in combination with artificial intelligence, provides models to monitor the coronavirus spread. Earth Observation component are available from program RACE (Rapid Action Coronavirus Earth observation tool) [34]. The dashboard uses Earth observation satellite data to measure the impact of the coronavirus. It provides high-resolution image that collect images daily with a ground sample distance less than 1 meter around the globe [35].

Monitoring systems based on satellite images can support new indicators and new areas of interest with little effort, as all of them share the same database and input format [36]. These characteristics favour the adaptation and application of such systems based on automated Artificial Intelligence computer algorithms to extract this kind of information from them, without requiring extensive manual labour. For example, one of the strategies is to obtain information on vehicles' traffic flow. To map increases and decreases in the flow of vehicles over an area of interest, it is possible to combine open knowledge sources such as OpenStreetMaps3 [37], satellite images, and a machine learning based vehicle detector. Generally, a sequence of three stages composes the procedure: i) location sampling and region of interest extraction, ii) vehicle detection and iii) temporal analysis.

1.3.5.2 National contact tracing and warning apps

Contact tracing solutions have had success in reducing infection transmission in many epidemics. Contact tracing procedures via public health officials' interviews are used to advise exposed contacts to self-monitor for symptoms, self-quarantine or obtain medical evaluation and treatment. Digital contact tracing makes use of electronic information to identify exposures to infection, it has the potential to address limitations of traditional contact tracing, such as scalability, notification delays, recall errors and contact identification in public spaces.

Generally, digital contact tracing uses smartphone enabled technologies such as GPS, WiFi, and Bluetooth, to save locations and contact details of their owners, so that if they get infected.

Digital contact tracing solutions work in 3 phases: i) sensing, ii) collection and iii) reporting. In the sensing phase, individual users record their own location or the contact with other people. In the collection phase users record location data or close contacts. In the reporting phase, if an individual user is confirmed to be infected, he/she collaborates with some third parties to make her relevant location data available. Eventually, third parties could collect and aggregate the location data from infected individuals for any possible legitimate purposes.

The privacy requirement applies to both infected victims and other users. Except for disclosing the necessary information to the authorities, an infected victim might want to prevent any further disclosure to avoid social impact. An individual user may want to check their risk of being infected by matching their location data with that of infected individuals, but they may not want to disclose their data to the third parties or the infected individuals for any other purpose.

Any digital contact tracing system is specified by different technical specifications. The proximity measurement mechanism used. It can be absolute location data or relative location data:

- **Absolute location data** considers absolute users geolocation coordinate pair, for example, they are GPS location, or coordinate respect to static WIFI access points and Telcom cell towers.
- **Relative location data** considers relative users contact, for example the pairing of two Bluetooth enabled smart devices, the boarding on a transportation tool such as planes, buses, cars, or ships. In this case, we can have some reference description about the location, for example both persons are on the same flight on the day XYZ.

The data storage method, centralized storage (data is automatically stored on a central server), partially centralized (only data from infected individuals is transferred to a central server), decentralized (data is stored solely on user device). Decentralized models are designed to keep as much sensitive data on users' devices as possible. Methods are introduced to strictly control data flows in order to avoid accumulating any contact data on a centralized server. This means that a server exists but only to enable people to use their own devices to trace contacts. The server is not trusted with sensitive data at all and therefore is not vulnerable to function creep like all the other solutions.

The method of installation, manual installation (the user takes care of installing the application), default installation (installation is automatically carried out by the manufacturer and is not voluntary) [38] [39] [40].

The use of these specification has raised policy questions about privacy and data management, determining different approaches in different countries, especially between Asia and Europe. Encouraged (manual installation) or mandatory (default installation), and the possibility to share data represent the main conditions. Some countries, including South Korea and China, have adopted involuntary data collection systems [41] [42]. Applications developed in Europe, instead, are based on voluntary use, without a government obligation. These solutions, based on relative location, use Bluetooth signal strength to infer distance between smartphones to find exposure status. The proximity and duration details are shared confidentially between users that already installed the application.

StopCovid is the French application that follows the ROBERT (Robust and privacy-presERving proximity Tracing) protocol to trace contacts with a centralized architecture [43]. This application uses a centralized server, which is considered as a single point of failure, and would raise serious privacy and security concerns that may not be accepted by the users. The application sends an alert asking the user to contact his or her doctor and take a test.

The Belgian digital tracing application **Coronalert** is similar to the German application, **Corona Warn**. The Coronalert application uses Bluetooth as a proximity measurement mechanism. Like other European states Finland, Austria, Estonia, Italy, and Switzerland, Belgium has opted to use the Decentralized Privacy-Preserving Proximity Tracing (**DP-3T)** Protocol created by the European DP-3T consortium and the Temporary Contact Number (TCN) [44]. Since the adopted protocol opted for a decentralized storage system, Coronalert stores data on the users' smartphone. All interactions more than 15 minutes in duration are recorded in the form of a randomly encrypted code. The code is in the form of a 15-digit crypto-identifier (regenerated every 15 minutes) is stored for 14 days on the smartphone. The notification mode is binary, no risk level is reported. When an individual tests positive, the notification process is as follows: firstly, the individual asks the application to generate a 17-digit random code before undergoing the test and presents said code to the doctor; secondly, the doctor indicates the random code, the national registry number and the patient's telephone number on the form; thirdly, if the test is positive, server asks the user for permission to access their crypto-identifiers to inform other users.

The British government promotes **NHS Covid-19**, it is based on the decentralized model developed by Apple and Google. This is the **Exposure Notification** system originally known as the Privacy-Preserving Contact Tracing Project [45] [46] [47]. The protocol is similar to the DP-3T, but is implemented at the operating system level, which allows for more efficient operation as a background process. This leads to special privilege problem over normal apps, particularly on iOS devices where digital contact tracing apps running in the background experience significantly degraded performance. NHS Covid-19 includes the following features:

a perimeter risk alert system to warn about the level of risk around one's home; a QR code scan function available at the entrance of shops and public buildings so that a positive case can be notified

1.3.5.3 Decentralized Privacy-Preserving Proximity Tracing (DP-3T)

We briefly describe the DP3T infrastructure as it is specified in the current version [44]. The communication device is a Bluetooth-equipped smartphone running the DP3T App. The backend server acts as a repository for some data to be pushed by smartphones upon authorization by the authority.

At setup, the app creates a key SK0. Periodically (presumably, every day), the key expires and is replaced by a new one which is computed by:

$$SK_t = H(SK_{t-1}) \ for\ t = 1, 2, \ldots$$

The duration between the time SK_t is created and its expiration is called the crypto-period of SK*t* herein. These keys are kept in memory and erased after a set period (e.g. virus incubation). Each secret key generates *n* ephemeral identifiers $EphID_i$ of 128 bits by

$EphID_1 \,//\, \cdots \,//\, EphID_n = PRG\ (PRF\ (SK_t;$ "broadcast key"))

PRF is suggested to be HMAC-SHA256 while PRG could be AES-CTR or Salsa20. During the cryptoperiod of SK_t, the ephemeral identifiers are used in sequence, following a random order. Each $EphID_i$ becomes the current one during one *n*-th of the crypto-period of SK_t. The App regularly broadcasts the current $EphID_i$, as a beacon, via Bluetooth interface. Conversely, the app stores received beacons together with extra information such as the time, the proximity.

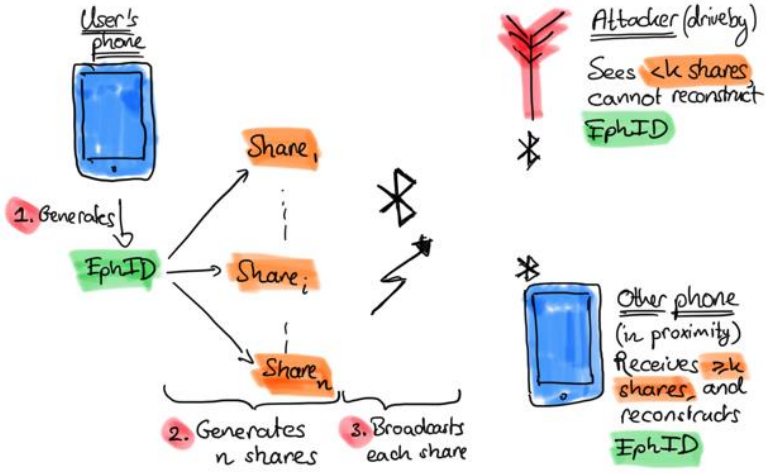

Figure* 1. *Broadcasts of the EphID, as a beacon, via Bluetooth interface [44]

Upon authorization by the authority, the server is fed by apps with pairs consisting of SKt's and their time of validity. They correspond to keys which were used by the App held by a user

who was reported by the authority (because of infection). New pairs are added every day and retrieved by the apps every day. With each pair, the App can re-generate the n ephemeral identifiers and check if they have been stored at the corresponding time. Based on that, the app can see how long and at which distance the infected person has been encountered, and can compute a risk. If the risk is above threshold, an alert is raised by the app.

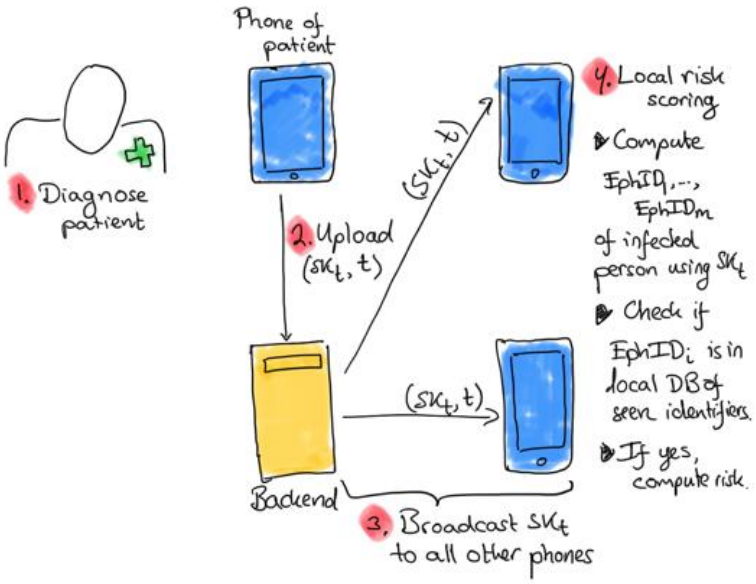

Figure 2. Backend workflow

1.3.5.4 Digital contact tracing privacy concerns

The constant and widespread monitoring by contact tracing apps, introduces privacy concerns. The collection of location data implies to obtain information about all individuals' movements. Even de-identified location data cannot be fully anonymized; paths of geographic coordinates can be cross-referenced with other public records to create probabilistic models of whom they belong to.

Despite the technological and health advantages provided by digital contact tracing, public decision-makers must take into consideration its impacts on privacy [39]. Several researchers argue that the adoption of digital contact tracing application could lead to the economic exploitation of private data and may also create a mass electronic surveillance system [48].

Thus, developed solutions should incorporate features to mitigate privacy concerns and ensure compliance with the European Union's General Data Protection Regulation. Such features include encryption of all personal data, restrictions on use of the data outside the public health responses to COVID-19, automatic deletion of data, and the option to delete data at any time. Use of an app should be voluntary and users should have the option to pause

contact detection, both to further protect privacy and to allow health care workers to disable monitoring when they are using appropriate precautions.

1.3.5.5 Attacks

In the attack scenario of this section, the goal of the malicious adversary is to make the app of a target victim raise false alerts. It could be disturbing and stressful for users to receive an alert. They could also be severely blamed by their neighbourhood or partner for being careless.

Attacks	Description
Untrusted server	The infected victims reveal the location data to the server, where the location data contain hashed network identifiers such as those of WIFI access points. Given that network identifiers are often static, this allows the server to recover the absolute location data points of the infected users. If a match has been found in the tracing phase, then user j's absolute location at some time stamps is also revealed to the server. In addition, in order to improve computational efficiency, time stamps are always disclosed to the server. The revelation of time stamps and absolute location implies serious privacy leakage and should be avoided.
Relay attack	An attacker can mount relay attacks, e.g. to relay the Bluetooth signal from Alice's smartphone to Bob's smartphone even when they are far from each other. Regarding the solutions employing Bluetooth for distance measurement, one attack deserves special attention. An attacker can place a Bluetooth range extender in a relatively populated place, such as a city square, and as a result it will make any pair of users a close contact to each other.
Backend Impersonation	Impersonating the backend server and send alert to the victim.
False Report	Instead of impersonating the backend server or the authority, the adversary could report his infection case.
Replay Attack	Consists of collecting existing *ephemeral identifiers* (EphIDi) and of replaying to several users that have not had contact with infected people.

1.3.5.6 Technology limit of the Bluetooth contact tracing solution

Second, the underlying technologies have measurement error, limiting the effectiveness of the apps in identifying contacts. For Bluetooth-based apps, there are several challenges in using signal strength to determine distance between devices: Bluetooth signal strength is hardware dependent, exhibits substantial fluctuations and is attenuated when people are between the transmitting and receiving devices [19-21]. Signal processing and hardware testing may mitigate some of these challenges. Bluetooth signals are also attenuated by walls and floors, which is advantageous in that it may reduce incorrect identification of exposure.

2 Design of spatio-temporal analytics as Virtual Network Functions

2.1 Machine learning Algorithms/ Pipelines

First, this section presents the list of ML/DL techniques and analytics (Table 1) proposed and studied for the various functionalities defined in WP5. Next, the design and development of each functionality are described in more detail following the taxonomy:

- Functionality description
- Inputs – Dataset, data format, pre-processing steps.
- Solution Design - Proposed and studied ML/DL algorithms/pipelines
- Outputs

Table 1. List of proposed ML/DL Techniques for each UC functionality

Use Case	Functionality	ML/DL Techniques and Analytics
NSE-UC1	Func-1: Identifying crowd mobility patterns	• K-means / K-medoids [11] / k-paths large scale trajectory Clustering [12] • ST Density based clustering – DBSCAN / HDBSCAN / T-DBSCAN [13] • RNNs with GRUs / LSTMs, CNNs & hybrids • Auto-encoders & variants [8], [9], [10] • ST Graph attention network [7]
	Func-2: Classifying people movement behaviour relative to geofenced perimeters	• Classification based on geofenced localizations using k-NN, SVM, Decision tree • Learning geofence models [49] • Smart geofencing using location sensitive product affinity [50] • Trajectory anomaly detection using normalizing flows [51]
NSE-UC2	Func-1: Learning group mobility characteristics using wireless fingerprints	Clustering algorithms for group detection [17] • DenGraph: Density based scanning • HCS: Clustering highly connected graphs • MaxClique: Clustering fully connected graphs • Girvan-Newman (GN), DBSCAN, MeanShift
	Func-2: Using multi-modal data for crowd mobility: COVID-19 as a special case	• Binary Classification (existence or non-existence of COVID-19 infection)

NSE-UC3	Func-1: Vulnerable road users clustering Func-2: Time to collision as a service in V2X (Also applicable to NSE-UC4)	• Classification of moving object trajectories using SVM, CRFs, Decision trees [52] • Classifying ST trajectories using CNNs [53] • Spectral Clustering in Graph-LSTMs [54]
NSE-UC4	Func-1: Logistics in a seaport terminal using AGVs	• Structured Analysis approach [55] • A* Algorithm [56] • Non-linear fuzzy controller [57]
NSE-UC5	Func-1: Transportation optimization based on identification of traffic profiles	• K-means clustering • Hierarchical clustering [58] • Density-based clustering: DBSCAN / OPTICS [59] • RNNs with LSTMs [8]

2.1.1 NSE-UC1: Flow monitoring and management in large venues and dense urban environment

The high concentration and flow rate of crowd in large venues like airports, train stations, malls, stadiums, etc. during rush hours can pose a prominent risk to passenger safety and comfort. Efficient crowd flow monitoring/control and resources management is essential to reduce these risks. For this, NSE-UC1 focuses on two functionalities: 1) Identifying crowd mobility spatio-temporal patterns, and 2) Classifying people movements relative to geofenced perimeters in a monitored indoor area. These functionalities can provide insights to measure footfall traffic, optimise visitor access, and detect possible congestions.

2.1.1.1 NSE-UC1-Functionality-1: Identifying crowd mobility patterns (spatio-temporal) – Location analytics such as possible visitor paths, POIs (indoor / outdoor / hybrid)

Identifying/recognizing crowd mobility patterns in cities is very important for public safety, urban planning, traffic and disaster management. This functionality addresses the general question of identifying patterns of individual or collective mobility behaviour, in terms of clusters, trends, densities etc in indoor/ outdoor/ hybrid locations.

Inputs: The expected input raw data to be consumed by this functionality is primarily trajectory information, consisting of timestamped longitude and latitude coordinates,

or timestamped X, Y. Additional information could also be coming as input, such as elevation/z coordinates, velocity, acceleration, user ids, etc.
Processed and aggregated data could also be consumed by this functionality, for instance flow-based data, Origin Destination tables, etc. The quality and performance of this functionality depends to a large extent on the input data, its structure, quality and scope in space and time.

Pre-Processing: Data coming from the real world will very likely be noisy, with irregular time steps, missing elements, etc. Hence pre-processing for this functionality is important. Pre-processing aims at converting the data from its raw representation to a more convenient one, for model consumption. This includes operations such as trajectory smoothing, correction and interpolation, timeseries resampling, outlier filtering, etc.
The data in question exhibits other inherent complex attributes, related to the complex mobility dynamics it represents at a microscopic resolution, spatio-temporal dependencies of its features, etc. With that said, this functionality requires clustering techniques that can discover spatio-temporal dependencies, unsupervised modelling and some degree of interpretability.
As an example, when relying on 5G based positioning in an indoor scenario, different positioning information originating for different 5G positioning techniques and possible auxiliary geolocation information (GNSS...etc) needs to be fused intelligently in a timely manner. And missing data, such as a resulting from system latency and unavailability of resources, needs to be detected and imputed.

Solution Design: Within this functionality deep learning approaches can be leveraged, such as models based on CNNs and RNNs, to support pipelines where feature selection and spatio-temporal modelling can be jointly optimized.
We will also investigate representative learning approaches to trajectory clustering, where CNN and RNN based autoencoding models will be trained to construct embedded representations of the locations timeseries, at a lower dimension. Classical clustering techniques such as K-means clustering, and Density-based clustering (DBSCAN/ OPTICS) will then be leveraged in the latent space representation of the data.
Interpretability is a challenge when working with deep learning models in general, and Autoencoders in particular, we will investigate the question of interpretability related to sequence-to-sequence autoencoders and investigate techniques such as Convolutional autoencoders and Manifold Learning. In the same context we will aim for the identification of good similarity metrics for trajectory and embedded trajectory data.

Outputs:

- Identified clusters or hotspots indicating crowd density and distribution.
- Description of the clusters in the form of meta-data, e.g. centroids.
- Spatio-temporal patterns indicating crowd movement such as possible visitor paths and POIs.
- Interpretable visualizations of the clusters indicating the crowd mobility patterns.

2.1.1.2 NSE-UC1-Functionality-2: In a monitored indoor area, classifying people movement behaviour relative to "geofenced" perimeters according to security/safety/other objectives.

People mobility is always tracked in security monitored indoor areas like shopping malls, airports or other venues, and these include authorized and unauthorized zones. Venue administrators/security can receive customized alerts for the existence of unauthorized persons in specific area and if there is any congestion in the POIs. This can help to quickly detect if there is any infringement/violation and to efficiently manage people traffic.
This functionality aims at implementing 'geofencing', which is the construction of a virtual perimeter around a geographic area, and the automatic detection of entry and exit of moving individuals/objects in this area.
Geofencing is regarded in this functionality as a means for mobility-based flow management, which can be further exploited to:

- deploying a specific network behaviour specific to the geofenced area (Advertisement, Content Serving, QoS…etc).
- implementing security/safety protocols where the presence of individuals/objects in the geofenced area in a specific pattern is undesirable and should trigger specific actions and policies, in near real-time.

The accuracy levels of this functionality in terms of classification, positioning, counting will vary, not only according to the specific deployment scenario under consideration, but also according to the availability and granularity of input data. The same with the timing constraints.

Inputs: This functionality will consume geolocation data coming either as raw timestamped trajectories, or in other aggregated form such as mobility flows, heatmaps, densities…etc. It will also consume contextual information necessary to assign virtual perimeters to geographic areas (e.g. geocodes, maps, environment features etc).

Pre-processing: The data quality assumptions and possible pre-processing steps are similar to those described in NSE-UC1-Functionality-1.

Solution Design: This functionality will investigate techniques to deploy accurate geofencing based on heterogeneous geolocation data, and environment-aware geofencing where specific landmarks and environment labels can provide assistance to maintain the geofences. This functionality will also investigate trajectory prediction and behaviour forecasting, to enable preventive flow management and geofence protection.

Geo-fencing is widely used for location-based marketing and customer-targeted advertisement. Broadly, there are two stages in implementing geo-fencing. The first stage, *geo-fence design* is about defining virtual boundaries (geo-fences) enclosing the selected key locations from an area of interest. The second stage, *real-time detection,* which is about detecting the presence of mobile devices present within the geo-fences in real time [50]. A similar approach to smart geo-fencing with location sensitive product affinity [50] can be leveraged for this functionality.

Another recent approach, highly successful for anomaly detections in trajectory data using normalizing flows [51] can provide greater insights to this functionality. A normalizing flow is a transformation of a simple probability distribution (e.g. Gaussian) into a more complex distribution by a sequence of invertible and differential mappings. In this approach [51], for each segment of the trajectory, model likelihood values are computed based on normalizing flows. Then the trajectory segments' likelihoods are aggregated into a single coherent trajectory anomaly score.

Outputs:

- Geofences – virtual boundaries enclosing the selected key locations in a given area of interest with reference to a venue map.
- Customised security alerts/ notifications for possible intrusions or detected anomaly.
- Interpretable visualizations of the geofences and crowd mobility patterns.

2.1.2 NSE-UC2: Crowd mobility analytics using mobile sensing and auxiliary sensors

LOCUS will perform crowd analytics in urban areas, using limited auxiliary sensors, such as cameras, for benchmarking and improving the accuracy of estimation, as well as machine learning with training from the historical data. Individual device location data can be leveraged to identify individualistic movement patterns using RNNs. Data manipulation, data fusion, and ML methods offered by LOCUS will be verified. Crowd sizes, group movement behaviours, people flow behaviours, and waiting time analyses will be conveyed via advanced visualizations. The crowd analytics results aim to optimize smart mobility by improving decision making such as path planning of vehicles by humans or autonomous vehicles.

2.1.2.1 NSE-UC2-Functionality-1: Learning group mobility characteristics using wireless fingerprints

NEC designs Group-In [17], a wireless scanning system to detect static or mobile people groups in indoor or outdoor environments. Group-In collects only wireless traces from the Bluetooth-enabled mobile devices for group inference. The key problem addressed in this work is to detect not only static groups but also moving groups with a multi-phased approach (see Figure 3) based on noisy wireless Received Signal Strength Indicator (RSSIs) observed by multiple wireless scanners without localization support. Group-In has new centralized and decentralized schemes to process the sparse and noisy wireless data, and leverage graph-based clustering techniques for group detection from short-term and long-term aspects.

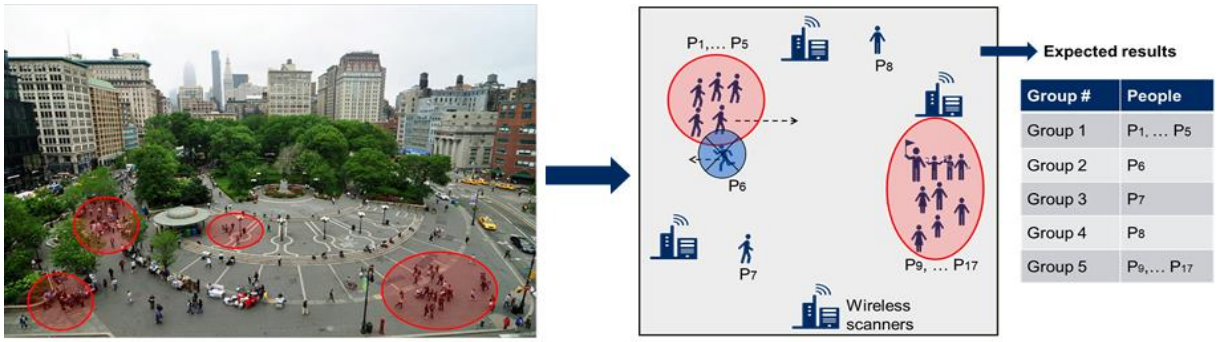

Figure 3. Detecting static or moving people groups in an urban area using wireless traces [17].

Inputs: This functionality will take geolocation data coming either as raw timestamped trajectories, or in other aggregated form such as mobility flows, heatmaps, densities, etc. Collection of user location and other essential data from sensors like Bluetooth, WiFi, etc will also be used. Figure 4 shows Group-In, multi-phased approach that includes the preprocessing steps involved such as Sampling and RSSI normalization.

Solution Design: A scientific paper related to the protoype system is published and presented in the ACM/IEEE Information Processing in Sensor Networks (IPSN'20) conference [17], which is a top-tier venue for fields such as sensing, cyber physical systems and IoT.
Group-In provides two outcomes: 1) group detection in short time intervals such as two minutes and 2) long-term linkages such as a month. To verify the performance, we conduct two experimental studies. One consists of 27 controlled scenarios in the lab environments. The other is a real-world scenario where we place Bluetooth scanners in an office environment, and employees carry beacons for more than one month. Both the controlled and real-world experiments result in high accuracy group detection in short time intervals and sampling times in terms of the Jaccard index and pairwise similarity coefficient.

Applications of the group inference are considered for crowd management, smart retail, evacuation modeling, social isolation, and social distancing.

For the group detection phase, following clustering algorithms used:

- DenGraph: Density-based scanning approach on graphs models
- HCS: Clustering highly connected subgraphs
- MaxClique: Clustering fully connected subgraphs

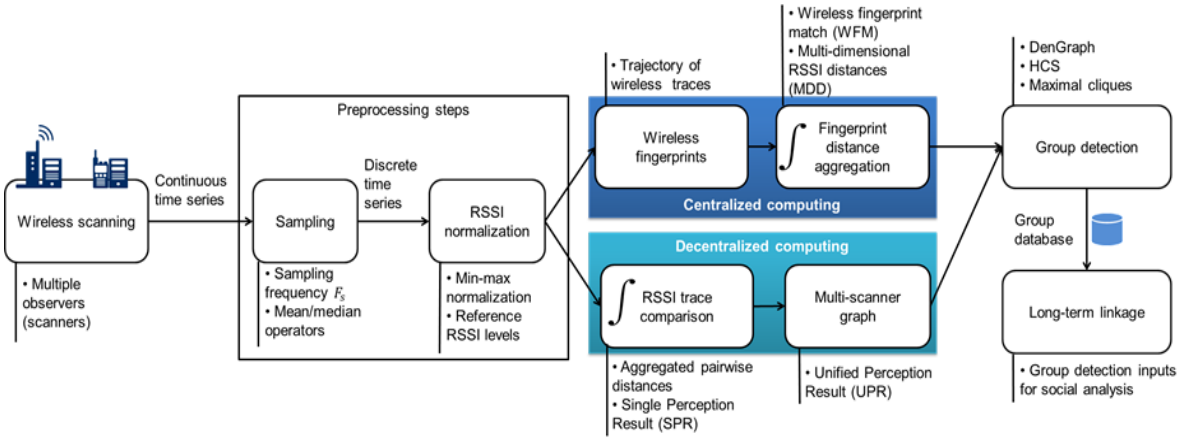

Figure 4. Group-In multi-phased pipeline approach with pre-processing, centralized/decentralized computing, and clustering [17].

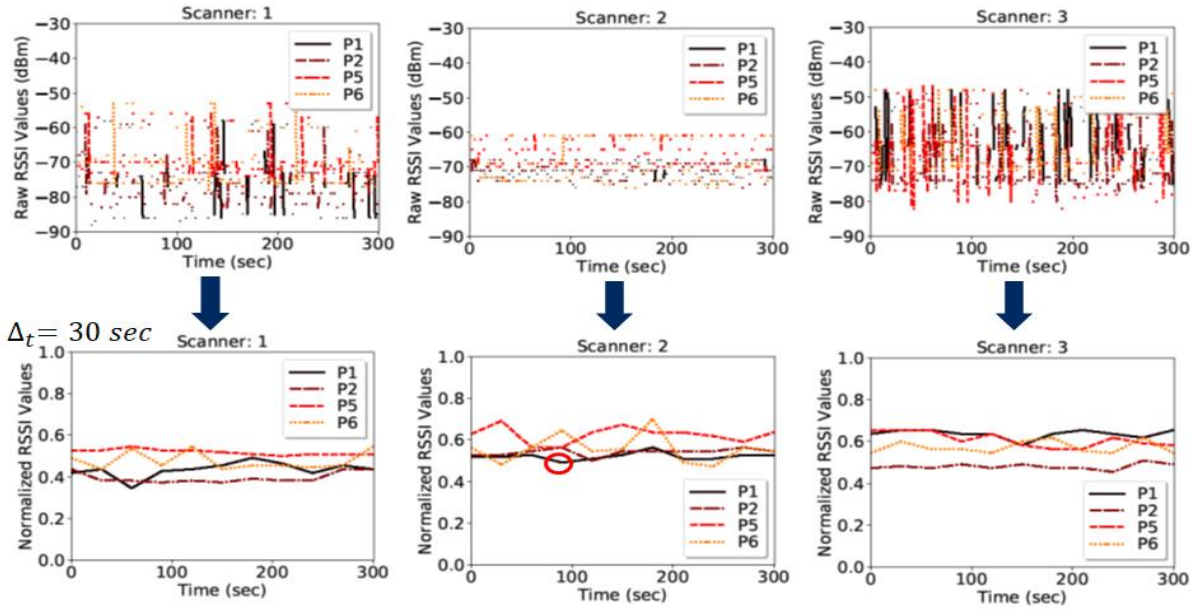

Figure 5. Result of the preprocessing phase: Top: Before preprocessing, bottom: after preprocessing [17].

Figure 5 shows the outcomes of initial tests conducted in the lab setup using three scanners (Scanner 1,2, and 3). The top shows the initial RSSI measurement (before preprocessing) from

the same beacons by different scanners. The bottom shows the same measurements after the preprocessing steps with 30sec sampling time.

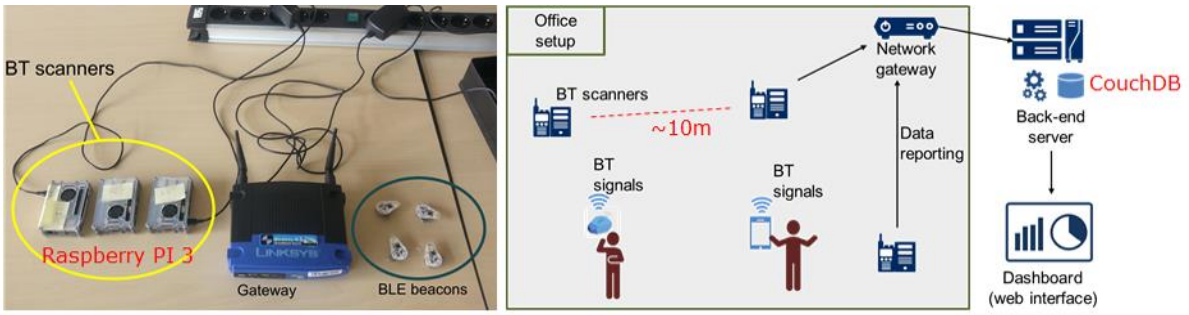

Figure 6. Experimental setup and basic setup in the office environment.

Figure 6 (left) shows some of the devices used in the initial experiments such as Raspberry PIs, BLE beacons, and wireless gateway. Figure 6 (right) illustrates the basic system setup for the experiments.

Outputs:

- Identified clusters as the inference of static or mobile groups.
- Estimate of group size.
- Group detection in short time intervals such as 2 minutes.
- Group detection in long-term linkages such as a month.

2.1.2.2 NSE-UC2-Functionality-2: Using multi-modal data for crowd mobility – COVID-19 as a special case

We plan to work on another functionality which would potentially help COVID-19 situation during and after the pandemic. The functionality targets understanding and automatically extracting “situations” in the real-world that may lead to the spread of viruses including COVID-19. For instance, some environments may lead to more spread, e.g., indoor areas without good air ventilation or setups which cause people to physically interact with each other. This functionality aims to recognize such scenarios using multi-modal data including camera data as well as various IoT sensors such as Bluetooth, Wi-Fi, accelerometer, gyroscope, infrared sensor, and so on. One possibility is to extend the existing NSE-UC2-functionality-1 which already aimed for group monitoring. The functionality would improve sensors other than wireless scanners such as image/video feeds.

Inputs: The input dataset should include the multi-modal sensing types from mobile sensing data (e.g., smartphone data), video feeds, as well as passive sensing data (through scanners or sensors placed in the environment). The data is planned to be formatted as data frames using the Pandas framework in Python language. The raw data will have formats of AVI, MP4, CSV, or JSON. Preprocessed data will have JSON or Pandas data frames format.

Pre-processing: Preprocessing step includes parsing all the datasets and combining them together in larger tables. Functionality 2 considers another pipelined approach which uses pre-trained ML models and enhancing the raw video data along with the annotated data from the pre-trained models such as object detection and pose detection engines. Moreover, we consider applying various clustering and ML methods such as K-Means clustering and Random Forest model, as well as more advanced neural network models such as DNNs, more specifically LSTMs, AutoML, data programming and others. The selection of the set of ML techniques is based on the need of the given scenario and datasets.

Solution Design: The solution is going to create warnings/alerts for the scenarios, settings, or dynamic events that may lead to the virus spread for a given environment. These situations are considered to be produced through the binary classifications (i.e., existence or non-existence of a virus possibility). The solution is considered to be implemented using Python language with TensorFlow and Keras frameworks, and Pandas library.

Feasibility Study: As the second functionality is defined recently (after the pandemic occurred) due to the unexpected COVID-19 situation in line with the COVID use case, UC2 considers initially conducting a feasibility study before actually developing advanced ML solutions/framework for the second functionality. The main goal is to understand the applicability of ML and sensing on the detection of proneness to COVID at a given situation in an indoor environment. The feasibility study includes leveraging existing datasets (e.g., open datasets or datasets obtained through collaborations with external partners) for simple setups, where data from sensors in an indoor environment are used for data collection. The goal in this study to first show that we could create simple COVID-19 labels using sensors that are carried by people or room sensors that are deployed in different parts of the rooms (e.g., ceiling). These simple labels would be used for initial visualizations, understanding of the data sources, and understanding correlations between various data sources. Later, the feasibility study considers employing off-the-shelf simple ML algorithms for preliminary training and testing.

Plan for Experiments: In the case of not being able to find suitable and usable open datasets or further testing different hypothesis, UC2 considers short data collection campaigns for controlled and/or real-world setups where various IoT sensors including wireless sensors and auxiliary sensors (e.g., cameras) are considered.

Future Work: The initial feasibility study is considered simple labels for COVID proneness using open dataset(s). In the later phase, a joint development of a more advanced ML pipeline is considered for the special COVID-19 monitoring and classification.

Outputs:

- Similar to the outputs from NSE-UC2-Functionality-1
- Customised warnings/alerts

2.1.3 NSE-UC3: Vulnerable road user

This use case alerts the host vehicle (HV) of approaching Vulnerable Road User (VRU) in the road. HV approaches the VRU along roads that are defined by their lane designations and geometry. The HV should be able to avoid collision with the VRU. Analytics could provide more detailed insight into the characteristics of the system e.g. tracking Time to Collision parameter for different type of VRU users in different weather conditions.

2.1.3.1 NSE-UC3-Functionality-1: Vulnerable Road Users Clustering

In future Cooperative-ITS V2X systems the number of VRUs operating in a specific area could be significant (e.g. up to hundreds of pedestrians crossing a road in a busy area) or the VRU can be combined with used VRU vehicle (e.g. rider on a bicycle). In order to reduce the amount of Vehicle-To-Pedestrian (V2P) communication and thus the resource usage to improve road safety, VRUs should be grouped together forming VRU clusters (see Figure 7).
These clusters can be homogeneous VRU clusters (group of pedestrians only) or heterogeneous VRU clusters (groups of pedestrians and bicycles with person). A VRU cluster is a set of two or more VRUs (e.g. pedestrians) such that the VRUs move with similar mobility pattern, i.e. with coherent velocity or direction. These clusters are considered as a single entity and only the cluster head VRU will continuously transmit the corresponding V2P safety messages for the whole group.

Inputs: Collection of host vehicle and VRU location information from device sensors at regular intervals, collection of user profile metadata from app registration, and collection of external meta-data related to roads, public transport routes etc.

Solution Design: VRUs could dynamically form, disband, join or leave the cluster. A VRU device determines whether it can join or should leave a cluster by comparing its measured position and speed with the position and speed indicated in the Vulnerable Road User Awareness

Message (VAM, see Table 2 for message format details) of the cluster head VRU. If the compared information fulfils certain conditions, e.g. less than 3 - 5 meters away and speed difference less than 5 % of own speed, the VRU device can join the cluster. VRU cluster is characterized by the number of participants, dimensions, speed and position. Detailed technical requirements for VRU cluster definition, concept, VAM and operational requirements are defined in ETSI TS 103 300-2 V2.1.1 (2020-05) standard.

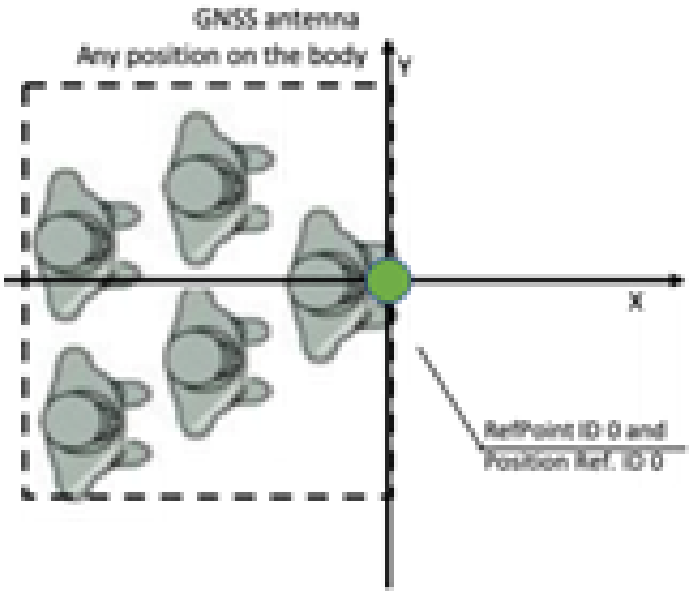

Figure 7. Cluster of pedestrians with a reference position example (adopted from ETSI TS 103 300-2)

Table 2. VRU Awareness Message format (adopted from ETSI TS 103 300-2)

Parameter		Comments
VAM header including VRU identifier	M	
VRU position	M	
Generation time	M	
VRU profile	M	
VRU type	M	e.g. VRU profile is pedestrian, VRU type is infant, animal, adult, child, etc.
VRU cluster identifier	O	
VRU cluster position	O	
VRU cluster dimension	O	geographical size
VRU cluster size	O	number of members in the cluster
VRU size class	C	mandatory if outside a VRU cluster, optional if inside a VRU cluster
VRU weight class	C	mandatory if outside a VRU cluster, optional if inside a VRU cluster
VRU speed	M	
VRU direction	M	
VRU orientation	M	
Predicted trajectory	O	succession of way points
Predicted velocity	O	including 3D heading and average speed

Heading change indicators	O	turning left or turning right indicators
Hard braking indicator	O	
NOTE: "M" stands for "mandatory" which means that the data element shall be always included in the VAM message. "O" stands for "optional" which means that the data element can be included in the VAM message. "C" stands for "conditional" which means that the data element shall be included in the VAM message under certain conditions.		

Outputs: Customised alerts / messages to HV that a VRU is approaching.

2.1.3.2 NSE-UC3-Functionality-2 (also applicable to NSE-UC4-Functionality-2): Time to collision as a service in V2X

Both NSE-UC3 and NSE-UC4 use cases relate to the vehicle mobility and describe scenarios with moving physical objects such as cars and VRUs creating potential road safety risk with their operational information e.g. location, speed and heading. Such operational information will be shared by V2X messages in future Cooperative-ITS systems.

Inputs: Continuous stream of historical geo-location data as generated from localization functions, Geo-spatial location information of the areas under investigation.

Solution Design: In Europe there is currently ETSI Intelligent Transport Systems standard (EN 302 637-2) which defines periodically sent (typically between 0.1 -1 second) V2X Cooperative Awareness Messages (CAM) including vehicle operational info such as vehicle heading, speed and acceleration (see Table 3). There is a similar ETSI ITS standard (TS 103 300-2) defined for Vulnerable Road User Awareness Messages (see Table 2).

Table 3. Cooperative Awareness Message format (adopted from ETSI EN 302 637-2)

Data Elements	Type	Typical Size (Bytes)	Description
Header	Mandatory	8	Protocol version, message type, sender address, and time stamp
Basic Container	Mandatory	18	Station type (e.g., lightTruck, cyclist, pedestrians, etc.) and position
High-Frequency (HF) Container	**Mandatory**	**23**	**All fast-changing status information of the vehicle, i.e., heading, speed, acceleration, etc.**
Low-Frequency (LF) Container	Mandatory (every 500 ms)	60 (7 path history points)	Static or slow-changing vehicle data mainly path history. The path history is made up of a number of path history points. Typically, 7 path history points are sufficient to cover over 90% cases based on extensive testing whereas up to 23 path history points can be contained. Each point is approximately 8 bytes [1].

Special Vehicle Container	Optional	2 ~ 11	Specific vehicles role in road traffic (e.g., public transport, vehicles realizing a rescuing operation, etc.).

Such operational moving object information may be collected and used by new analytics algorithms and could be used to create novel advanced localization and analytics-based services for road safety applications applicable both to vehicle and VRU safety use cases. One example of such application which leverages vehicle or VRU location, speed and heading information could be 'Time to collision as a service in V2X' based on the Time to Collision (TTC) parameter definition in ETSI ITS standard TS 101-539-3 V.1.1.1 (2013-11) which defines the time period before the physical collision of one moving object with another one with a conflicting movement trajectory. This parameter is typically used to decide the nature and urgency of the required collision avoidance action (see Figure 8 for details). TTC calculation example between a vehicle and VRU is provided in Figure 9.

It is expected that in the future, 'Time to collision as a service in V2X' could be a location analytics-based enabler for new business applications in connected and automated mobility such as

- Driving risk evaluation for individual vehicle or VRU
- Traffic management for vehicles in a specific region
- Transport network planning for a specific road network section.

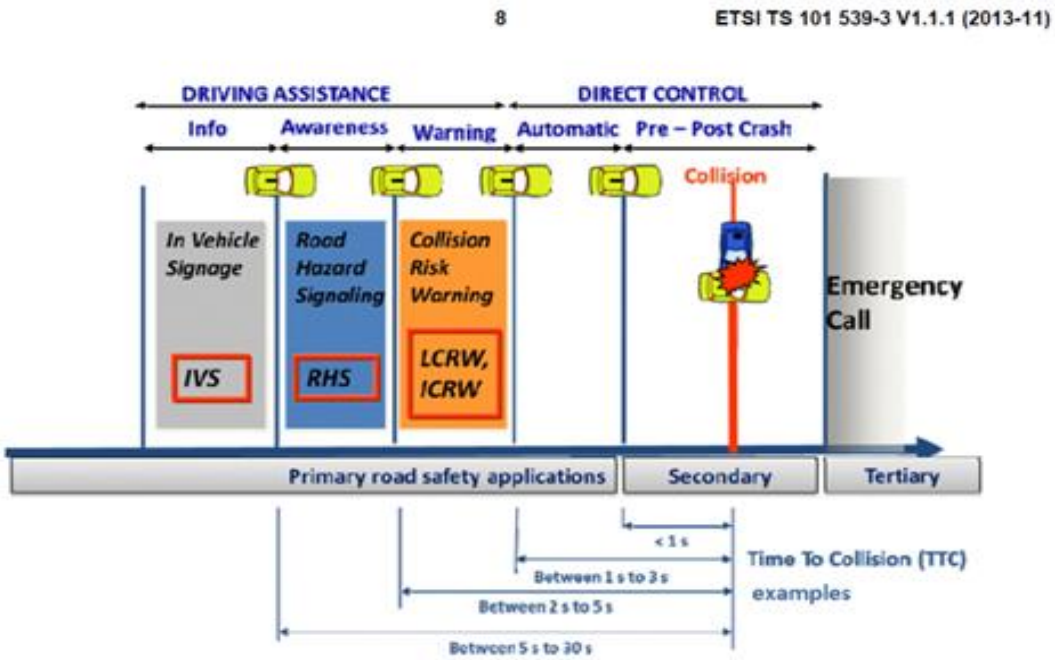

Figure 8. Time to Collision parameter definition (adopted from ETSI TS 101 539-3)

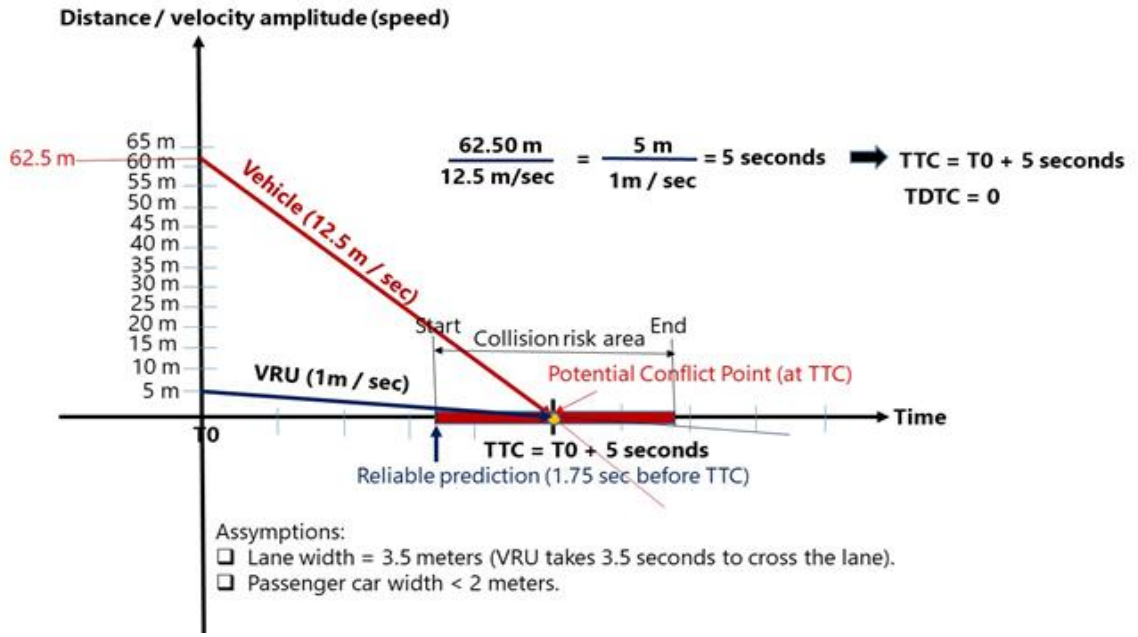

Figure 9. Example of Time to Collision calculation between a vehicle and VRU (adopted from ETSI TS 103 300-2)

Outputs: Estimation of Time to Collision and Likelihood of Collision

2.1.4 NSE-UC4: Logistics in a seaport terminal using AGVs

Logistics is a key element in industrial and seaport operations. Flexibility required by production in Industry 4.0 and deep automation of seaport activities require the massive introduction of automated transport systems like AGVs (Automated Guided Vehicle) that require an accurate and real-time localization to implement a high-performance navigation of the autonomous vehicles.

In outdoor logistics, like seaport, the position accuracy required by AGVs shuttling freights between storage and loading areas is less than 1 meter. In this case there is a continuous interaction between the AGV and humans in correspondence of the loading and unloading areas. So, a very high precision is not required, since the workers will compensate small errors. For instance, when an AGV reaches the loading area a crane, driven by a man, will pick the objects to be loaded on the ship. So, he can take care of small positioning errors.

Since 5G connectivity is going to be introduced in industrial and logistics environments for lowering OPEX and reduce operation times, while increasing safety and efficiency, 5G could be a proper mean for providing an accurate positioning of AGVs both in indoor and outdoor contexts. If sufficiently accurate, it could be a valid replacement of magnetic tapes on the floor, LIDAR and camera sensors used nowadays for AGV navigation purposes. The use case intends to demonstrate the feasibility of introducing a sufficiently accurate positioning system

based on 5G technology. The 5G network should detect the positioning of AGVs operating indoor and be a valid replacement of GPS in outdoor cases,

Inputs: The real time position data of the AGVs collected at regular intervals, map of the shuttling area is considered already available to the navigation system.

Solution design: In this UC a mission/navigation system that control the movement of AGVs in a seaport environment will leverage on the accuracy of positioning information provided by the LOCUS platform to achieve high precision real time control of AGV operation. The basic structure of the mission/navigation control system for AGV is described in the following picture:

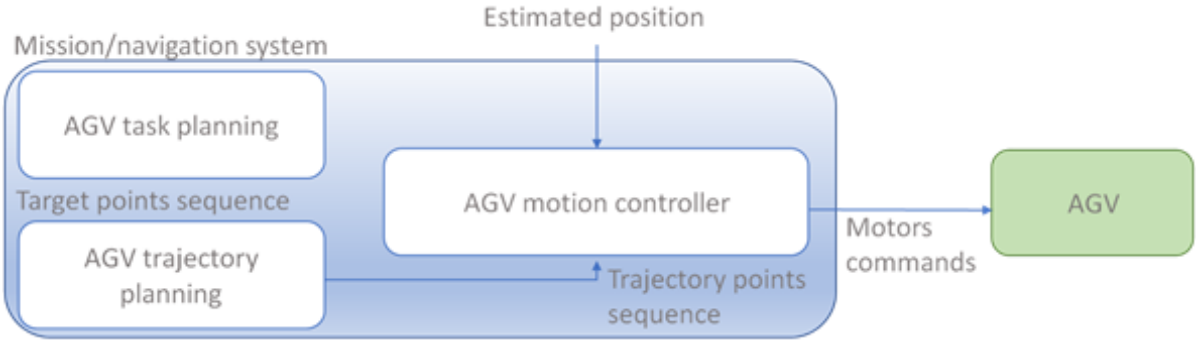

Figure 10. Mission/Navigation control system for AGV

Main functional blocks are:

- AGV task planning function: This function coordinates the activities in the seaport warehouse area and takes care of assigning the vehicles to the different missions depending on their status and their current position respect to the freight to shuttle. This function handles also the relational DB containing the freights inventory and the vehicle data. It will make use of a rule-based expert system using CLIPS and a relational DB based on mySQL.
- AGV trajectory planning function: This function is used to determine the path the vehicle must follow in the test area first to reach the freight to pick and then the destination. It is based on the A* algorithm. The algorithm will receive a map reporting the areas were the vehicle can and can't go through. These areas include also the other freights placed in the seaport area whose position is known. The map is built on the fly based on information about static object in the area and position and size of freights contained in the inventory in the relational DB
- AGV motion controller function: This function controls the navigation and movements of the vehicle in the area. It receives information from the trajectory planning function

and from the positioning system and, based on this information, it computes the next movement step. The controller is a non-linear fuzzy controller. This kind of controller was chosen because its non-linear behavior allows for fast response and high accuracy both for short and long movement steps.

Outputs:

- AGV motion controller provides the predicted next movements of AGV as trajectories.
- AGV task planner coordinates the activities in the seaport warehouse and takes care of assigning the vehicles to different missions based on their status.
- AGV trajectory planner determines the path the vehicle must follow in the test area. Map manager builds a map on the fly based on the information about the static object in the area, and position and size of freights contained in the inventory as per the relational DB.

2.1.5 NSE-UC5: Transportation optimization based on identification of traffic profiles

This use case involves the abstraction of location information at a large scale in an outdoor area. Given this outdoor setting, where various high or low traffic streets, avenues and motorways as well as train routes and pedestrians exist, a variety of different mobility profiles emerge. This use case will enable flexible aggregation of low-level anonymized positioning and velocity information and will offer an abstracted view of location-based data with the purpose of monitoring. The LOCUS platform will take the necessary steps to exploit its capabilities to satisfy UC requirements, offering services such as (a) identifying different mobility profiles through an augmentation and fusion process and (b) extraction monitoring options to users through an App/Dashboard. The App itself could potentially be used by state entities like traffic police in order to enhance decision processes and could be further expanded to include various functionalities, e.g. near-future predictions of traffic in the selected area, detection of any anomalies/incidents and so on.

NSE-UC5-Functionality-1: **Analytics on crowd mobility profiles (e.g. Pedestrian, road traffic, railway routes) and predict the near-future traffic by assigning trajectory profiles per UE.**
Analytics on crowd mobility profiles by understanding trajectory profiles and prediction of near-future positions/trajectories for the determination of future traffic. The goal of this functionality is to prove the feasibility and exploitability of location information through time for smart city traffic management.

Inputs: The analysis focuses on the true operator dataset that is available within the consortium, i.e. the OTE dataset, and with the anonymization restrictions that would apply in order to provide this service. Geolocation UE information at regular intervals, therefore trajectory information will be used. Complementary information like e.g. mobility labels (if available) or velocity can be employed (or calculated indirectly through the position information in the case of velocity).

Pre-processing:

The dataset preprocessing and cleaning procedures may vary depending on the very nature of the data, the purpose/application, as well as the algorithm used. It is also part of a trial-and-error process and therefore it may change based on the results. In this context, we present an initial exploratory analysis of the dataset and the various methods used.

Step 1: Data Cleaning/Manipulation

- Select columns with Time, IMEI (user identifier), Latitude, Longitude and drop rows containing Nan values.
- Drop rows containing irrelevant values (e.g “ZAKINTHOUC4” in Latitude column).
- Set latitude, longtitude to float and IMEI type to categorical.
- Optionally and for convenience, time values/slots can be set to integers or continuous values (e.g. '24-02-2020 19:00:00' -> 0, '24-02-2020 19:15:00' -> 1 etc.).
- Users that have less than 3 points in the dataset are considered short for trajectory-related analytics and therefore are drop.
- For users that do not appear for many consecutive time slots (currently set to 2 hours), we consider the new appearance a new independent trajectory, assigning a new user identifier.

Step 2: Data Conversion from coordinates to meters (optional step to allow for future calculations and model training to be done in meters)

- Calculate distances, angles to all data points from a global starting point and find new x, y values given the calculated distance, angle (using sine, cosine).

Step 3: Preprocessing interpolation

- Due to the multiple measurements per timeslot that vary per IMEI/id, we group data by id and for each id we assume at least 3 points per time step (3-point “tracklet”) through linear interpolation. As this is an initial exploratory analysis for understanding and predicting trajectories, we further interpolate if trajectory length is small so that for each user will have at least 20 points per trajectory.

Step 4a: Preprocessing for clustering methods

- Following step 3, the trajectories are divided in smaller 3-point sequences (called tracklets), using a time-shifting window of length 3.

Step 4b: Preprocessing of data for sequence/trajectory prediction

- Use a time-shifting window of length 15 (12 points for training and 3 for prediction) to create shifting trajectories of equal length for model training/testing
- "Min-max" scaling for the global minimum, maximum x, y values in trajectories.
- Split data to training set as 90% of the original dataset. For early stopping purposes, the remaining set is further split in half in the final test set and a validation set (for LSTM model tuning).

Solution Design:

List of unsupervised learning algorithms to group trajectory patterns per profile and perform exploratory analysis of the mobility patterns

- K-means clustering
- Hierarchical clustering
- Density-based clustering (DBSCAN/ OPTICS)

Supervised approaches (Classification algorithms) to assign profile/ trajectory pattern.

- Recurrent neural networks (RNNs) – LSTMs

Outputs:

- Trajectory predictions for the near future; Aggregation of predictions to provide future UE density, e.g., for congestion.
- Real-time map visualization of traffic density in a given area.

2.1.6 NSE-UC6: Positioning and Flow Monitoring for Controlling COVID-19

This use case focuses on developing efficient tools to enhance the health safety in the restarting phase after COVID-19 pandemic or prevent any future bouncing waves of this pandemic. This use case underpins the 5 user stories:

- User Story 1: A person is tested positive to the virus and it is needed to trace back the persons he/she has potentially been in proximity within a certain number (to be set) of previous hours/days.
- User Story 2: Risk factors based on epidemiological data are associated to flow of people moving from one area to another area.

- User Story 3: A person is tested positive to the virus or is one of the categories at risk (each elderly people). It is required to identify if he/she is moving outside a quarantine area.
- User Story 4: Automatic control of non-allowed grouping person, or above a certain group size, and in certain locations.
- User Story 5: User is informed about infection probability per given public area (e.g. metro).

All these user stories require certain degrees of privacy that might vary with the Country.

2.1.6.1 NSE-UC6-Functionality-1: Contact tracing

The goal of this functionality is given an identified case of COVID-19 infection, it traces back the persons to have potentially been in proximity with the positive case within a certain number (to be set) of previous hours/days.

Inputs: Historical data of past contacts of an individual related to their COVID-19 situation, crowd mobility data from the area under surveillance,

Solution Design:
There are two main approaches envisaged: one based on 3GPP operators' data and one relying only on an app that derives proximity/contact data from non-cellular technologies (e.g., Bluetooth). While the first suffers of limited location accuracy, the second suffers of possible limited number of people that will really install such app (for various reasons such as privacy and security concerns, digital divide issues, fragmentation of the market with several apps offering incompatible solutions). In addition, contact tracing system can follow two different models: i) centralized, in which the generation of identifiers and generation of contact graphs are done on a central server; and ii) decentralized, which avoid accumulating any contact data on a centralized server.

Outputs:

- Classification of the mobility behaviour in accordance with current quarantine and health protocols.
- Trigger a proximity event whenever proximity/contact criteria are detected by using the available geolocation information.
- Provide timely updates on current quarantine and health protocols, and trigger a violation event when-ever a quarantine is violated or a mobility behaviour is classified as risky, illegal, etc.

2.1.6.2 NSE-UC6-Functionality-2: Monitoring epidemiological risk flow

The goal of this functionality is to estimate risk factors and their spatiotemporal evolution using epidemiological data combined with the flows of people moving from one area to another area.

Inputs: People flow data from area to area (area dimension depending on the available data, ranging from tiles to census areas) on time frame basis (ranging from every hour to daily basis depending on the available data), and aggregated mobility data from the census areas with epidemiological data in those area.

Solution Design:
As cases of COVID-19 are being reported, epidemiologists are conducting public health surveillance: the systematic collection, analysis, and interpretation of health data [60]. Surveillance allows epidemiologists to calculate: **incidence** (number of new cases reported over a specific period of time), **prevalence** (number of cases at one specific point in time), **hospitalizations** (number of cases resulting in hospitalization) and **deaths** [60].

There is a recent study on developing mathematical models to quantify epidemiological risk and personal infection risk [61]. In this study, a high-resolution spatio-temporal model, namely HiRES is proposed for the risk assessment of epidemic disease with human-to-human transmission based on trajectory data and mean field theory. Based on the epidemic risk maps produced by HiRES model, a person infection risk scoring model is used to obtain quantified risk of infection for every authorized individual. Based on these risk scores, we can develop statistical inference and machine learning approaches respectively to detect early infection of suspected cases.

Outputs:

- Provide a spatiotemporal evolution of epidemiological risk based on data aggregation
- Predictions for the risk of contagion in different areas

2.2 Spatio-temporal analytics virtualized functions

LOCUS offers a platform that expose localization analytics as services that can be leveraged by 3rd party vertical applications to fulfil their specific logics. In practice, LOCUS implements a set of localization analytics and ML functions that can be composed to build high level localization services to be consumed by external applications. In the context of applications for new services (NSE), i.e. for 3rd party verticals, LOCUS exposes a set of localization analytics functionalities as described in section 2.1. Such functionalities refer to ML algorithms and

pipelines targeting the NSE use cases defined in D2.1, together with analytics functions for pattern recognition. These localization analytics functionalities are considered as LOCUS virtualized functions (with reference to the terminology defined in D2.4) and are managed by the LOCUS Management and Orchestration (MANO) for their on-demand deployment in the LOCUS virtualized platform developed in WP4, as virtual functions.

The following subsections provide further details for designing and developing the functionalities described in section 2.1 for ML and localization analytics as virtual functions, including how to package them for being ready to be managed and orchestrated by the LOCUS MANO.

2.2.1 Packaging of spatio-temporal analytics functions

As specified in deliverable D2.4, LOCUS follows a Service Based Architecture approach, where the LOCUS functions provide atomic and loosely coupled localization and analytics related services that can be chained to realize more complex localization analytics services. For this, each LOCUS function (i.e. localization enablers, analytics functions, ML algorithms) is subject to be packaged as a virtual function with the aim of being deployed on-demand in multiple instances to satisfy the requirements of the various LOCUS applications running on top of the platform. This allows to implement a localization analytics platform where elementary functions can be combined in different ways to build heterogeneous pipelines producing analytics services that match specific 3rd party vertical applications localization requirements. In practice, this allows to re-use (and possibly share) the same function for different purposes, by creating multiple instances starting from the same virtual function package.

Indeed, in this case, packaging a LOCUS function means to provide a common deployment template, including at least a description of the function and the reference to the software image to be run to realize the function itself. This means that the various analytics and ML functions described above need to be packaged following a unified format in order to be managed in a unified way by the LOCUS Management and Orchestration part of the architecture. According to D2.4 [3], LOCUS could follow the ETSI NFV approach, in which case the LOCUS functions could be packaged in the form of Virtual Network Functions (VNFs) packages, allowing their combination and interconnection as localization analytics services modelled as Network Service Descriptors (NSDs). This unified packaging approach allows the localization and analytics functions and services for both Smart Network Management and new 3rd party vertical application to share the virtualized platform where they run and be managed by a common LOCUS MANO framework.

The packaging of the LOCUS functions as virtualized functions, however, includes two main principles. The first one relates to the common deployment format just described above, that

is the VNF Package that follows the ETSI NFV management and orchestration principles, and it is an archive that contains all of the required information for the LOCUS MANO to manage the lifecycle of LOCUS virtual functions.

Figure 11. A VNF Package content (ref. ETSI NFV GS SOL 004)

A VNF Package, as specified in D2.4 [3] and shown in Figure 11, includes:

i) the VNF Descriptor that defines metadata for VNF lifecycle management, in terms of virtualized resource needs, connectivity, supported operations, size-bounded deployment and capacity configurations

ii) the software images required to run the VNF

iii) a manifest file that provides package integrity and authenticity,

iv) optional additional file for specific lifecycle management needs (e.g. vendor-specific files, configuration scripts, etc.).

This format can be considered valid for both LOCUS functions implemented as microservices based on containers, as well as more traditional telco-oriented Virtual Machine based VNFs. Indeed, ETSI NFV has recently expanded its management and orchestration functionalities towards the so-called Containerized Network Functions (CNFs), which are basically VNFs implemented as containers for cloud native NFV deployments.

The second packaging principle is more related to how the function itself is run within the virtualized platform, in terms of type of software image referenced in the package itself. LOCUS supports two types of software images for its localization and analytics related

functions: containers and Virtual Machines. For each of them, a different software packaging approach is designed and used in LOCUS, as described in the next two subsections.

2.2.1.1 LOCUS spatio-temporal analytics functions as Docker containers

Docker [62] is the de-facto standard for building and sharing applications running in containers. A container allows to package an application with all of its dependencies into a standardized unit for software development, offering a logical packaging mechanism in which applications can be abstracted from the environment where they actually run. This way, container-based applications can be deployed easily and independently from the target environment, being it a private or public cloud. Containers are the base foundation of cloud-native technologies, and unlike Virtual Machines they do not required operating system virtualization (and thus introducing high overhead) while they enable a much more efficient usage of the underlying system and resources, that translates into lightweight applications and faster deployment and termination times.

LOCUS supports localization and analytics related functions implemented as containers, enabling their deployment in cloud-native environments, e.g. based on Kubernetes [63]. As said, the LOCUS containers refer to one software image type supported in the VNF package.

With Docker, the process for containerizing a LOCUS functions follows a standard procedure, which is depicted in Figure 12. This procedure is built around three main fundamental items:

- *Dockerfile*: it is a file that provides the main set of instructions required to build a new Docker image. The instructions are executed by the Docker engine in the order they are listed *Docker Image*: it is an organized collection of files, configurations and installed programs, as well as a set of instructions (from the Dockerfile) for the execution of the various programs. It is built from the Dockerfile by using the Docker command: "*docker build*". Docker images do not have a state and are immutable, and typically contains layered filesystems. *Docker Container*: it is a running instance of a Docker image. It is possible to access the container, write in its filesystem and make changes, but this does not affect the image. When the container is terminated, the changes applied are deleted with the terminated instance. In LOCUS, the containers are instances of the LOCUS elementary functions and are managed by the LOCUS MANO, which takes care of their lifecycle management (i.e., instantiation, configuration, termination) according to the requirements of the localization analytics service they are instantiated for.

Docker enables a semi-automated process for container images creation and exposure starting from the raw software code of a given LOCUS spatio-temporal analytics function. As depicted in Figure 12 indeed, once a function is developed, a Dockerfile can be prepared

(typically as a human based action) to create a Docker image through an automated build process. Once the Docker image is ready, it can be automatically published to a Docker registry. The Docker registry is a repository that hosts all the available container images that can be used for cloud-native deployments of the developed functions. In LOCUS, it is linked to the Functions Catalogue that is part of the LOCUS MANO, as described in deliverable D2.4. This catalogue indeed maintains the list of localization analytics functions that can be deployed in the LOCUS virtualized platform (representing the whole set of analytics capabilities of LOCUS) and stores the related software images required to deploy a new instance of each function.

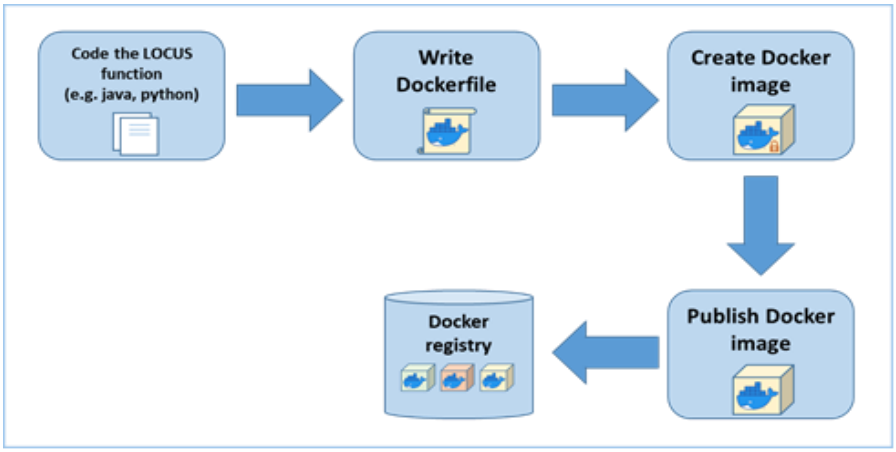

Figure 12. LOCUS functions packaging into containers

2.2.1.2 LOCUS spatio-temporal analytics functions as Virtual Machines

While Docker containers enable cloud-based deployment that are natively flexible, agile and highly scalable and suitable for lightweight analytics applications and logics to be run, e.g., at the edge of the network where computing resources are limited, VMs implement a heavyweight abstraction of physical servers with full operating system features. This makes VM deployments more suitable for long-lived analytics applications that can run at scale in core data centres. As said, LOCUS aims to support both options as the requirements for the localization analytics functions developed in WP5 (and WP4 as well) are heterogeneous and in general any solution is suitable for all of the cases.

VMs are a specific case of software images that provide a full package including operating system, application logics and the required binaries and dependencies (such as libraries and other applications) to run the given main applications. With this approach, the process for creating a LOCUS spatio-temporal analytic function VM follows a fully manual and human-driven based on a de-facto standard procedure suitable for Infrastructure as a Service (IaaS)

platforms such as Openstack [64]. As depicted in Figure 13, after the function is developed and related software code is ready, a clean VM instance is created, e.g. with dedicated software tools for virtualization such as Qemu [65] starting from a base image (e.g. Ubuntu cloud image [66]. Once the clean VM instance is created, the LOCUS function can be installed in the VM. This is still a manual process that can be facilitated through some tools like Ansible [67], which help sorting out the various application dependencies. Once this installation process is completed, the VM instance can be exported as a new VM image (e.g. with Qemu this is a transparent process) that is the final version to be uploaded in the image service of the reference virtualization platform. In the case of LOCUS, Openstack is used so the new VM image is uploaded to the Openstack Glance image service. This image service, as it happens for the Docker registry, is linked to the LOCUS MANO Functions Catalogue.

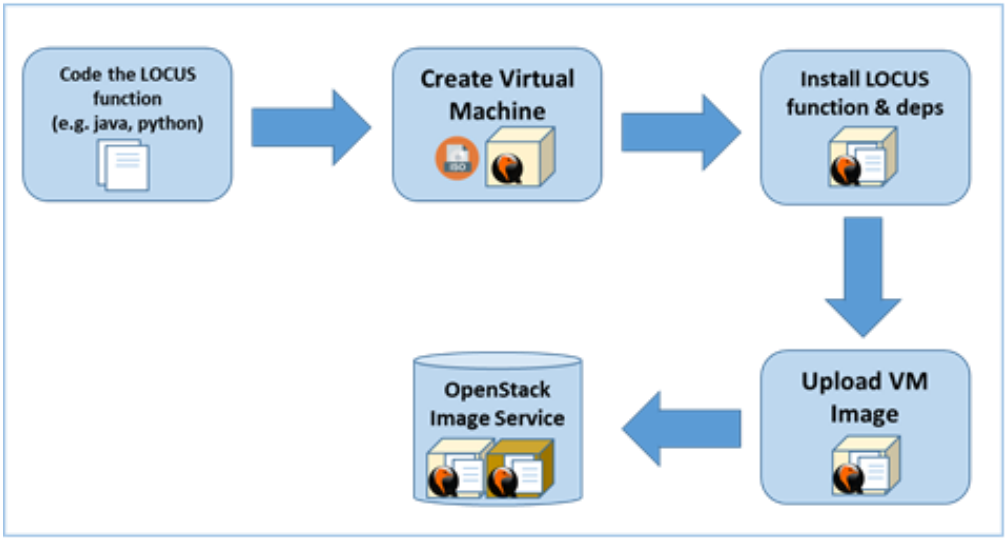

Figure 13. LOCUS functions packaging into VMs

2.2.2 Deployment of LOCUS spatio-temporal analytics virtual functions

The LOCUS spatio-temporal analytics virtual functions are a specific type of LOCUS functions which deal with the data analytics capabilities required by the 3rd party applications running on top of the LOCUS platform. As detailed above and specified in D2.4 [3], they can be properly packaged as virtual functions to be managed by the LOCUS MANO and deployed on-demand in the virtualized platform as part of localization analytics services which are offered by the LOCUS platform towards the Smart Network Management and 3rd party vertical applications. In particular, in the case of analytics for 3rd party vertical applications, a localization service is practically the implementation of one of the NSE UC functionalities specified in section 2.1, and it can be seen as the chaining of analytics and ML functions to realize a specific localization

related service objective (e.g. the identification of clusters for crowd mobility). These services have specific requirements in terms of input data (e.g. positioning data) and generate specific output data depending on the service objective. These localizations services together with the output they generate, as specified in D2.4, are exposed through the localization analytics as a service APIs, which regulate the interaction of the 3rd party vertical applications with the whole LOCUS platform. Full details of this API layer design principles and specification will be provided in deliverable D5.3 [68].

On the other hand, it is important to highlight that the virtualized platform (and related management and orchestration capabilities) required to run the analytics functions for the NSE UCs described in section 2.1 is implemented in LOCUS as a common framework to host localization virtual functions and services for both Smart Network Management and 3rd party applications. In addition, the deployment of the machine learning and analytics functions for the various LOCUS NSE UCs is also managed and orchestrated by the common LOCUS virtualization platform that will be developed in WP4 and reported in D4.3 [69].

However, for the specific case of the LOCUS NSE machine learning pipelines described in section 2.1, the following subsection provide more details on the virtualization approach followed to deploy and run them as virtualized services.

2.2.2.1 Virtualization of the NSE UCs machine learning pipelines

A machine learning pipeline can be seen as a workflow where various data manipulation steps and operations are involved, e.g. including data cleaning, data pre-processing, data filtering, etc. Following the approach defined in section 2.2.1 above, each pipeline component can be developed or in any case become a self-contained set of user code, and in turn packaged as a stand-alone virtual function. In the example depicted in Figure 14, each pipeline step is packaged as an independent Docker image, that provide that specific pipeline capability with its requirements in terms of input data and generation of output data.

At the deployment phase, each machine learning pipeline component can be therefore executed as a virtual function that requires to be linked and chained with the other steps according to their input and output data requirements. In the case of Figure 14, this translates into their deployment as container POD instances in Kubernetes.

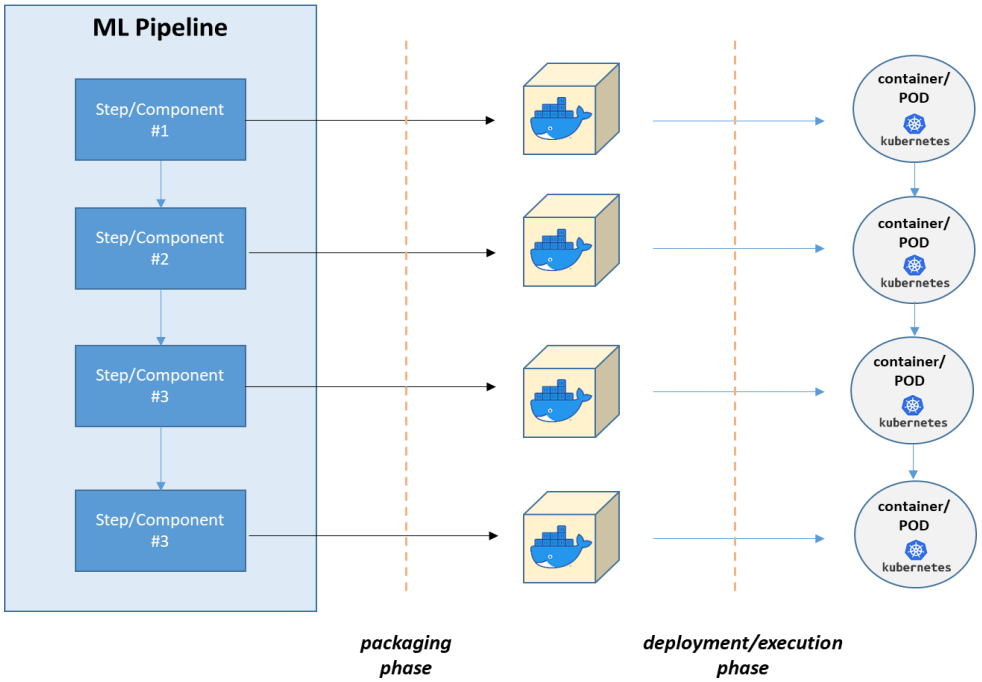

Figure 14. LOCUS machine learning pipeline virtualization approach

The role of the common LOCUS virtualization platform implemented in WP4, as described above, is exactly to provide those localization service coordination features that makes the chaining of the various machine learning steps possible, thus guaranteeing the requirements of each machine learning component are fulfilled. In this sense, the collaboration and integration with the LOCUS virtualized platform and MANO functionalities developed in WP4 becomes essential for the proper management and deployment of the NSE UC functionalities and machine learning pipelines.

Beyond the coordination and orchestration functionalities provided by the LOCUS MANO, different approaches are possible to make a virtualized machine pipeline work in terms of chaining between the various steps and components. In the initial investigation and design of machine learning pipeline virtualization approach, which will be continued as part of T5.2 and T5.3 activities and further reported in next WP5 deliverables, four different models have been identified to make the virtualized machine learning pipeline components exchange the required data. These are briefly described below, taking as reference the machine learning pipeline for the NSE UC1 functionality 1 (for crowd mobility patterns) inference, and assuming the use in the LOCUS platform of a common (logically centralized) data store where data generated and outputted by the various LOCUS functions (including localization enablers and the various analytics functions).

Figure 15 shows the first proposed model, which is based on a direct interaction among the various steps, with dedicated data loading functions to retrieve the pipeline input data and store the pipeline output. Therefore, the interaction with the common LOCUS data store is restricted to input and output pipeline data. While this approach can lead to ad-hoc interfaces between the pipeline steps (thus limiting the re-use and share of common steps across different virtualized pipelines), it highly simplifies the management and orchestration at deployment/execution phase as the pipeline configuration carried out the LOCUS MANO would mostly imply proper selection of input data.

The second proposed model is depicted in Figure 16. With respect to the previous approach, in this case the interaction among the various machine learning components is mediated by the common LOCUS data store, so each virtual function composing the virtualized machine learning pipeline read and write to the data store, making each function more independent. On the other hand, the way pipeline input data is loaded, and the output data is stored is similar to the previous case. This approach is expected to improve the possibility to reuse and share of common machine learning virtual functions, as in this case they can be considered as stand-alone functions with loose direct dependency with other steps. However, for each machine learning pipeline virtual function, the management and coordination aspects for function configuration at deployment time can become more complex as it should take care of input and output data requirements of each step.

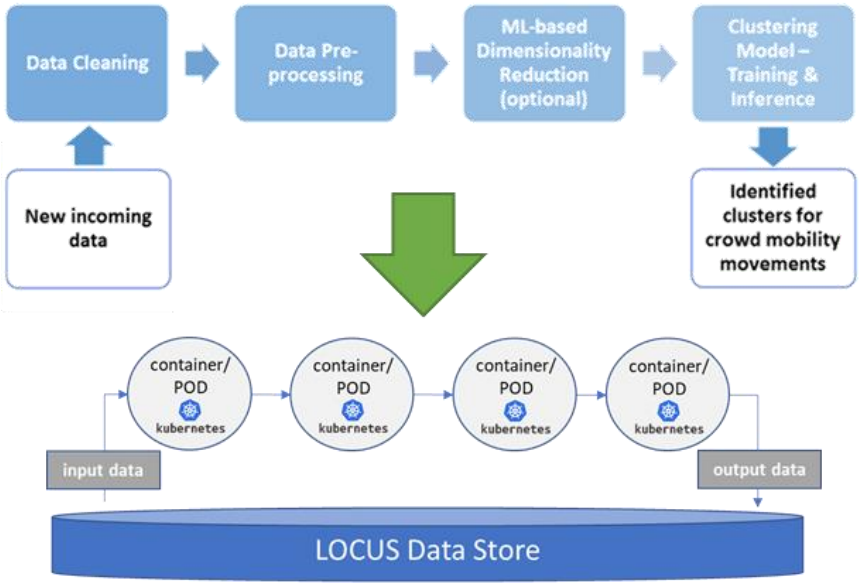

Figure 15. Machine Learning virtualization approach: direct communication

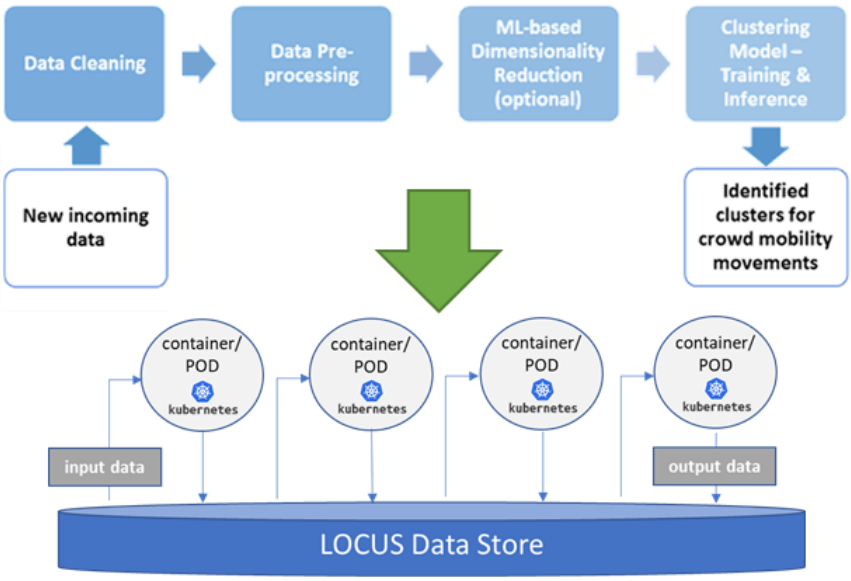

Figure 16. Machine Learning virtualization approach: communication through common data store

Figure 17 shows a third virtualization approach, where the NSE UC machine learning pipelines virtual components interact through a local data store which is deployed as part of the virtualized machine pipeline service. This case represents an evolution of the previous approach, where local (and possibly temporary) pipeline intermediate data is not maintained in the common LOCUS data store. This can be useful in those cases where intermediate data shall not be shared with other pipelines or in general analytics functions not belonging to the given localization service.

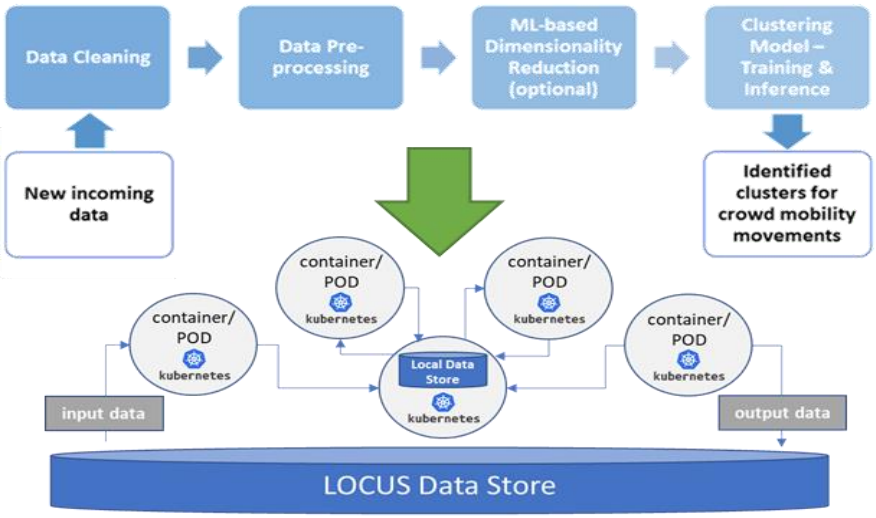

Figure 17. Machine Learning virtualization approach: communication through local data store

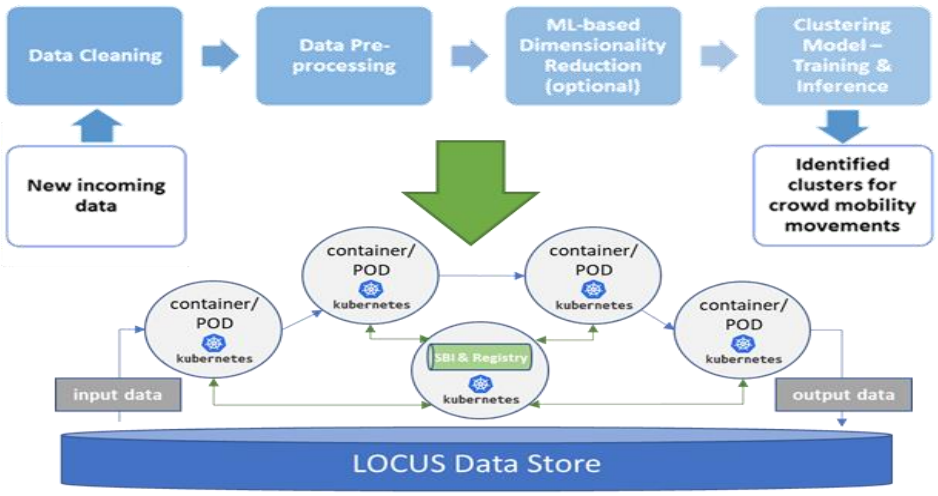

Figure 18. Machine Learning virtualization approach: service-based interface

The last proposed model is depicted in Figure 18, and it is based on a service-based approach for the interaction among. In practice, in this case the virtualized machine pipeline includes a specific virtual function which provide service discovery and registry features to let the pipeline components interact through a service-based model. As a generic principle, this approach is similar to the first one, as it enables direct interactions among the various pipeline steps. However, in its implementation there is a substantial difference that makes it more flexible, improving re-use and share of functions. Indeed, the virtual function providing the service-based interface and registry function can be considered as a common function to be re-used in different pipelines, that enables unified interfaces and interactions among the pipeline virtualized steps.

As said, these four models are the result of a preliminary design work for the virtualization of NSE UC machine learning pipelines. More details on their analysis will be provided in next WP5 deliverables as part of the T5.2 and T5.3 activities related to data virtualization and management (with the definition of a LOCUS unified approach for data storage and exposure), as well as to virtualization of NSE UC functionalities and their exposure as localization analytics services towards 3rd party vertical applications.

3 Implementations and Results

This section presents the machine learning analytics involved, implementation details, some initial results and indicative solutions for the different UCs, and the virtualization steps.

3.1 Machine Learning Models - Selection and Evaluation

In this Section, we investigate different ML models for the UC functionalities and present the implementation details, some initial results and the next steps.

3.1.1 NSE-UC1: Flow monitoring and management in large venues and dense urban environment

For this use case, we started our experiments with an initial exploratory analysis on open PFLOW [70], an open dataset for typical crowd movements through different transportation modes in urban areas. To start with, we investigated different trajectory clustering and trajectory forecasting techniques to identify crowd mobility patterns, and we report our preliminary results.

Recently, deep learning models based on CNNs and RNNs have been shown to be highly successful in hierarchical feature learning ability in both spatial and temporal domains [10].

We started our exploratory analysis on the open PFLOW dataset with simple k-means clustering and its variants such as k-medoids [11], k-paths large scale trajectory Clustering [12]. K-medoids algorithm is robust and less sensitive to noise and outliers, compared to k-means as k-medoids minimizes the sum of dissimilarities between the designated medoid (cluster centre) and the various labelled data points in the cluster. We observed both k-means and k-medoids are more suitable for spatial clustering. As a next step for effective trajectory data clustering, we tested k-paths clustering. Given a set of trajectories, k-paths aims to partition the trajectories into clusters to minimize an objective function based on trajectory distance measures [12]. K-paths clustering has two advantages when compared to k-means such as trajectories can be of varying lengths instead of fixed-length vectors in Euclidean space, the centroid path is not only based on the mean value of all the trajectories in the cluster, and a trajectory distance measure for two trajectories can be defined [12]. After testing different partition-based clustering approaches like k-means, k-medoids, and k-paths, we tested successful density-based Spatio-temporal trajectory clustering techniques such as DBSCAN and its variants HDBSCAN and T-DBSCAN [13]. We observed for large scale trajectory clustering all the tested techniques are inefficient and also computationally expensive for longer sequences due to the complexity involved in the distance measure computations.

Even though RNNs are widely used for time series modelling and are designed to recognize the sequential characteristics and use patterns to predict the next likely scenario, they suffer from short-term memory due to the issue of vanishing gradients. GRUs and LSTMs are extensions of RNNs that have shown considerable success in spatio-temporal data predictive learning and representation learning. LSTM models are good at handling sequence data while CNN models are effective when capturing the spatial correlation in the image like matrices. The hybrid model that combines RNN and CNN can capture both the spatial and temporal correlations of spatio-temporal data. Therefore, we started investigating trajectory predictions/forecasting using RNNs with GRUs/LSTMs [7] [8], CNNs and hybrids [9], auto-encoders & variants [10] for this use case.

3.1.1.1 **Input Data**

The open PFLOW consists of continuous spatio temporal locations data for an individuals' movement and these data can be aggregated and processed to investigate mobility patterns or link traffic volume to show mass people movement [70]. The PFLOW dataset is an aggregation of person trip surveys from 25 metropolitan areas in Japan stored on a minute-by-minute basis in a spatio temporal database with tracking id, latitude, longitude, time stamps, transportation mode, and magnification factor. This is a huge dataset that contains more than 660 million data records. Even though PFLOW is a simulated dataset, it is close to real world data and it comes with anomalies such as noise and irregular sequence lengths. Our pre-processing steps to mitigate these anomalies included filtering and truncating the irregular trajectory sequences to properly handle long trajectories by restricting them to a predetermined sequence length, scaling the sequences using a minmax scaler, transforming the GPS world geodetic system (WGS84) coordinates to a metric XY grid.

In the following analysis, we used a sample (approximately $1/6^{th}$) of the open PFLOW dataset for our initial testing,using GRU/LSTM based models for single step prediction, and sequence-to-sequence Autoencoder models for multiple-steps prediction and spatio-temporal clustering based on the Autoencoder latent space.

The pipelines developed are as shown in Figure 19 and Figure 20 for prediction and clustering. The preprocessing steps included loading the trajectories as spatio-temporal data, filtering the samples and restricting them to a predetermined sequence length, scaling the sequences using a minmax scaler. The raw data is represented in GPS world geodetic system (WGS84), we make sure to transform it to metric XY.

In the case of trajectory prediction, the models ingest fixed length 'tracklet' (sub-trajectory) configured with a 'look-back' parameter, and predicts a fixed length 'tracklet' configure with a 'look-forward' parameters (equals 1 for single-step and more than 1 for multiple-steps). In

the case of clustering, only the 'look-back' parameter is relevant since the autoencoder learns to reconstruct the input as good as possible.

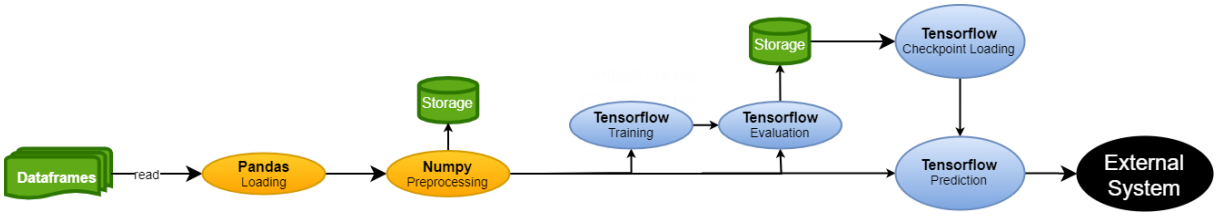

Figure 19. Prediction Pipeline

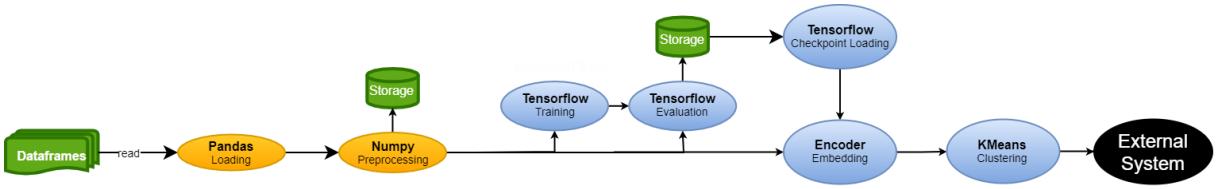

Figure 20. Clustering Pipeline

3.1.1.2 Single Step Trajectory Predictions

Figure 21 shows an instance of single step prediction trajectory prediction using simple RNN with GRU or LSTM models. The goal is to predict the next point given a random anonymous trajectory with spatial and time stamp information. The RNN models overfit for longer sequences even with drop out regularization. When dropout is used for RNNs, the 'disturbance' it generates at each time step propagates over a long interval, thereby decreasing the network's ability to represent long range dependencies. Including one dimensional convolution filters along with GRU/LSTM (as shown in Figure 22) provided better results especially for long look backs.

Figure 23 shows an example of an original trajectory, and the reconstruction of original trajectory by single step predictions using GRU, LSTM. The models were trained with a tracket length of 240 equals to 4 hours (data comes with 1min temporal spacing). We observed that a GRU based models perform better than their LSTM counterparts, we should note that when it comes to training performance, models with an initial 1D Convolutional layer perform the best, this is explained by the fact that Convolution layers are know to surpass recurrent layers in benefiting from CUDA acceleration supporting the Tensorflow API using for this work.

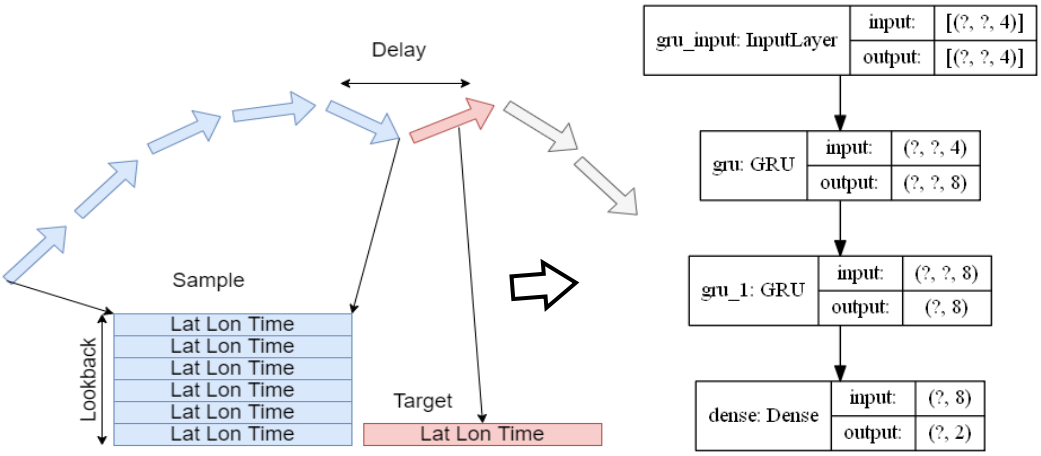

Figure 21. Trajectory prediction using simple RNN (GRU / LSTM) models

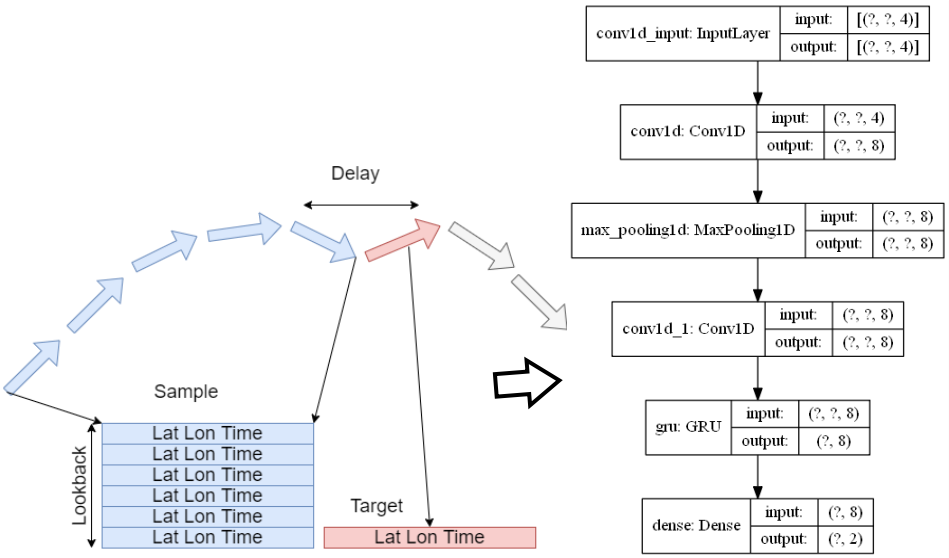

Figure 22. Trajectory predictions using Convolution + GRU/LSTM models

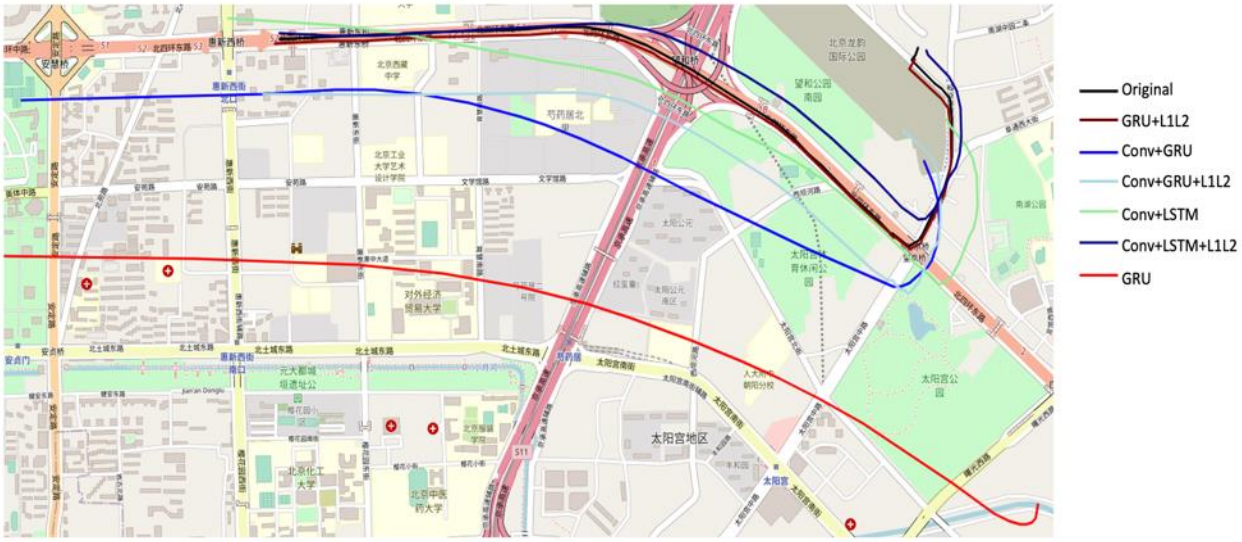

Figure 23. Trajectory prediction and reconstruction (sample) using different DNN models

3.1.1.3 Multiple Steps Trajectory Predictions

We also investigate multiple steps trajectory prediction, using a Sequence to Sequence Autoencoder Model as shown in Figure 24, and implementing the pipeline in Figure 19.

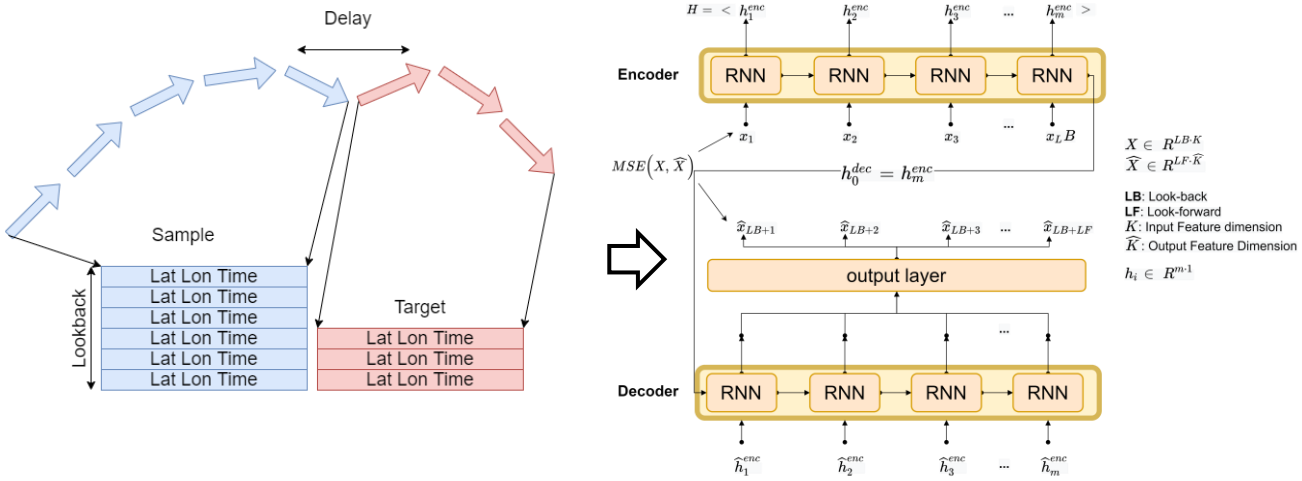

Figure 24. Multiple Steps Prediction

The trained model is evaluated using the Average Displacement Error (ADE): defined as the average of the root mean squared error between the ground truth and the predicted

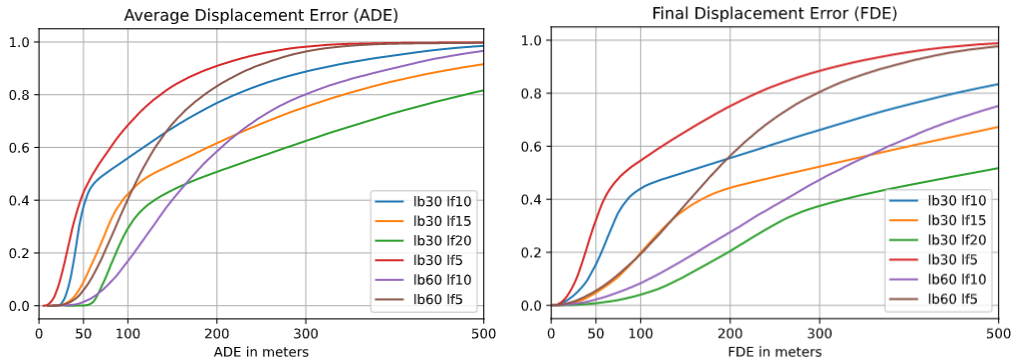

Figure 25. ECDF of ADE and FDE evaluations

trajectory, and Final Displacement Error (FDE): defined as the ADE taking into account only last positions in the ground truth and predicted trajectories.

In Figure 25 Empirical Cumulative Distribution Functions are plotted for both ADE and FDE, for evaluations of models trained with varying look-back (lb) and look-forward (lf) parameters. As expected, the best prediction performance is with short look-forward value, in other words the longer in the future the model tries to predict the higher the error it is going to make.

It is also noticeable that performance does not increase with increasing lookback parameter, in other words providing the model with more information does not yield better learning. This can be explained by the fact that temporal spacing of 1 min means the further we go back in the past, the less likely is the information relevant to the current mobility decision, higher temporal resolution is more likely to yield better performance for the same sequence length.

Another key insight from this result is that to gain further performance gains in prediction, we should look beyond the sequential information represented in the trajectories themselves and consider contextual information and inter-personal dynamics between moving entities. This point will form the basis for the further development of this work.

3.1.1.4 Trajectory Clustering

We tested sequence to sequence (seq2seq) models based on LSTM and GRU for representation learning, and clustering in the Autoencoder's latent space. We implemented the pipeline described in Figure 20.

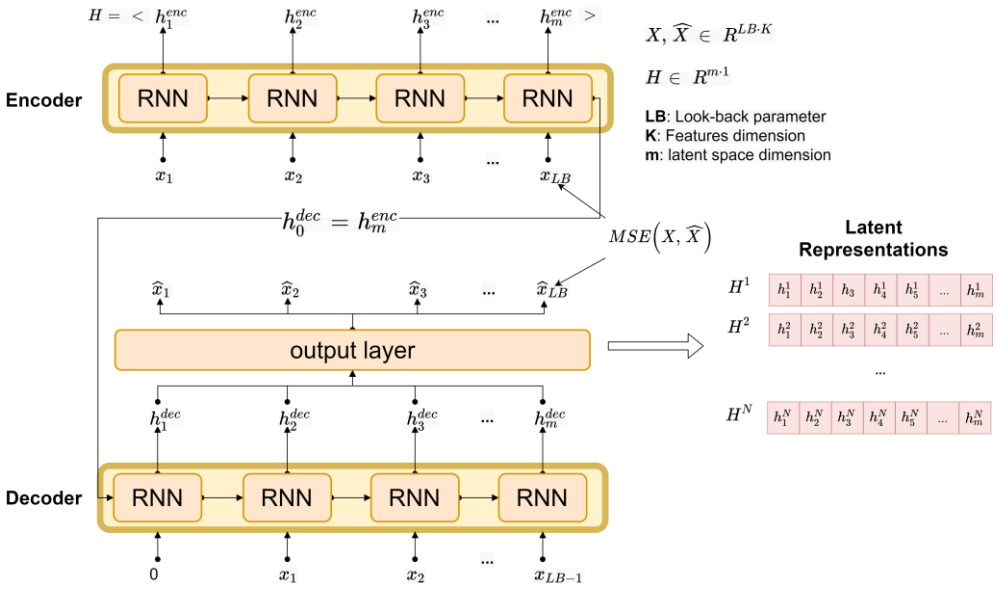

Figure 26. RNN (LSTM/GRU) Encoder-Decoder Architecture.

Figure 26 shows the steps involved in a basic seq2seq flow of an autoencoder: i) The encoder generates the fixed size vector H from the input sequence X, ii) decoder reconstructs the sequence X from H. H is a low dimensional representation, that is the result of the model learning to squeeze the most essential information from input trajectories, that is sufficient for good reconstruction.

Using the model in Figure 26, we conducted training on the OpenPFLOW dataset, and extracted the low dimensional representations H^i. We conduct principle component analysis on the latent representations, and we target an explained variance of 95%. Kmeans algorithm is applied to the data represented by the reduced components. Figure 27 Show this process,

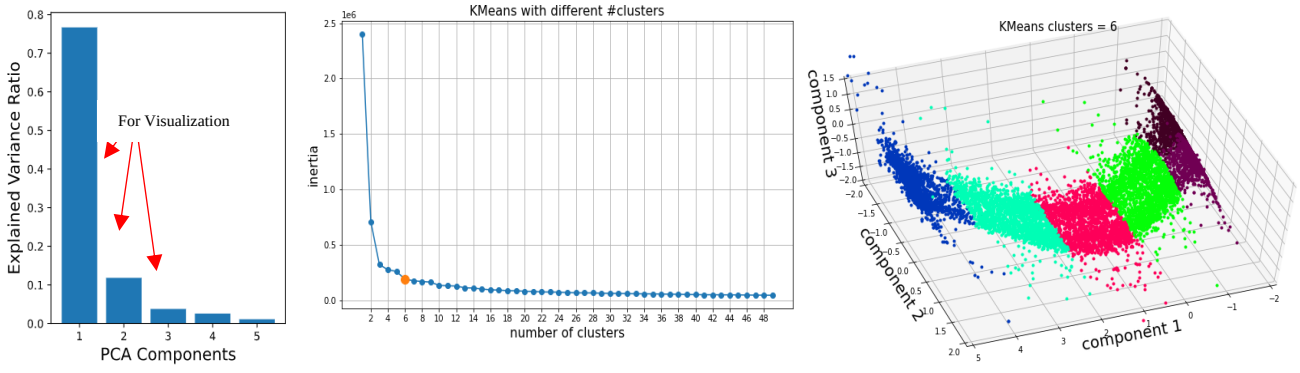

Figure 27. Clustering Analysis.
From left to right: PCA analysis, elbow detection of #clusters, visualization.

where PCA analysis resulted in 5 components that explain at least 95% of the data, from which we use the first 3 for visualization. To select the cluster size for Kmeans, we use the elbow technique where we chose the point of the highest curvature for Kmeans inertia as function of cluster size, which yields an optimal cluster size of 6 for the used data sample.
We map the discovered clusters in the latent representation to their corresponding original tracklets and we reconstruct the original trajectories knowing their designated clusters. The result is plotted in Figure 28.

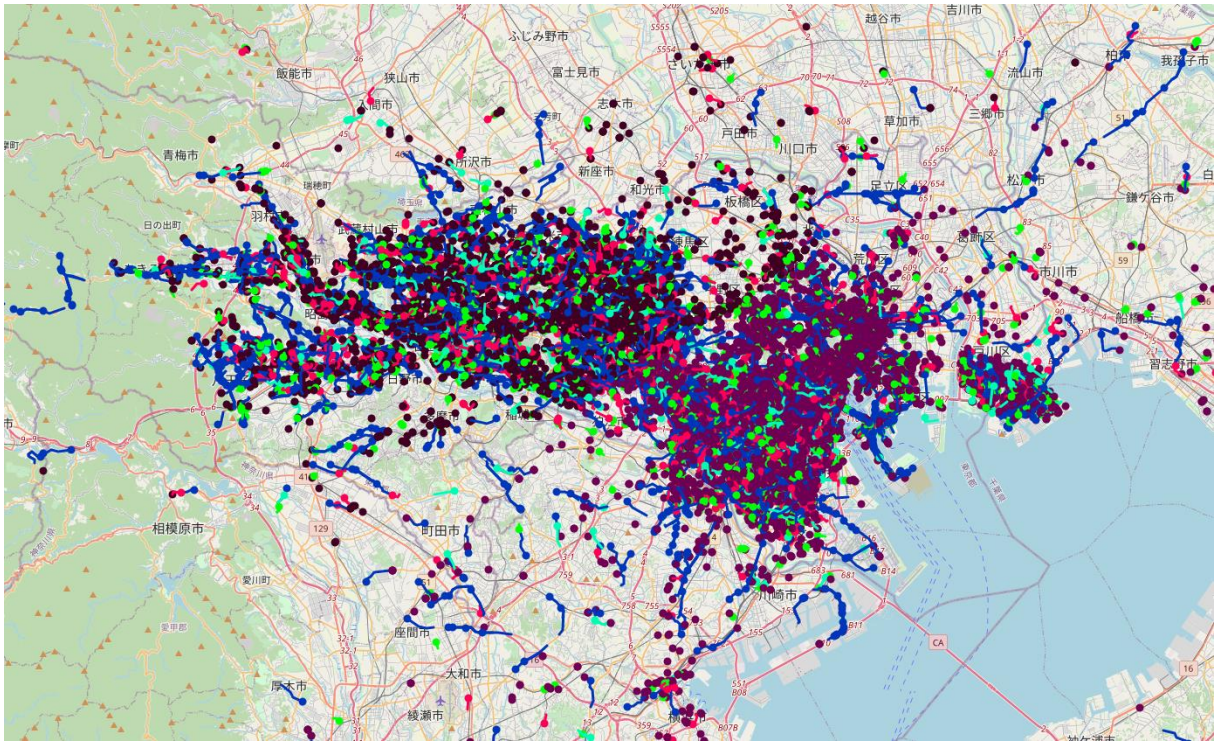

Figure 28. Visualization of Kmeans clusters on a map

The outcome of this analysis shows that latent representation based clustering, allows for the discovery of interesting regional, sub-regional and cross-regional mobility trends across time. The analysis is also very dependent on the size of the data sample used for training, and on the look-back parameter that dictates how much sequential information to digest by the model. The fact that the analysis is unsupervised poses a challenges in terms of evaluation of the accuracy and interpretability of the results. These insights form the basis for our future development of this work.

3.1.1.5 Plan for Future work

The future work will focus on improving mobility prediction, by looking for datasets with higher temporal resolution, and incorporating inter-personal mobility dynamics for short

range predictions. We also also investigate models that incorporate contextual information (geographic, landmarks, semantics...etc) to enhance long range predictions. As well as applying prediction models to other representations of mobility information, such as flow-based representation, group mobility, with a focus on crowded environment commonly found in indoor and dense urban areas.

For clustering, we will investigate further the possibility of acquiring or creating a labelled data set, to assist in evaluating the accuracy of the unsupervised clusters detection, and add some degree of interpretability and confidence in the output of the analytics pipeline relying on this functionality.

We already tested mean squared error loss functions. Further, we want to explore metrics and loss functions such as Dynamic Time Warping (DTW) and shape and time distortion loss [71] and other performance metrics suitable for trajectory forecasting. Another recent successful approach for trajectory prediction [7] leverages the self-attention mechanism and transformer-based convolution mechanism and introduced a spatio-temporal graph transformer framework. Our future work will explore graph attention networks for trajectory representation learning and forecasting.

3.1.2 NSE-UC2: Crowd mobility analytics using mobile sensing and auxiliary sensors

3.1.2.1 NSE-UC2-Functionality-1: Learning group mobility characteristics using wireless fingerprints

Details of the Experiments:

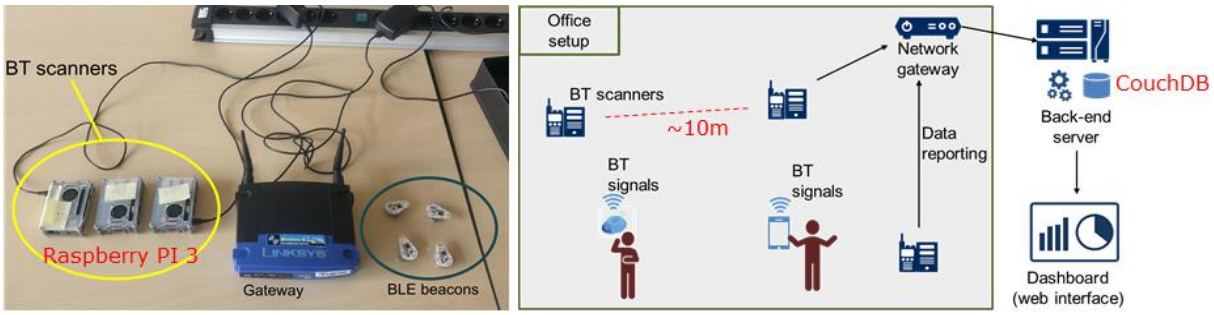

Figure 29. Experimental setup and basic setup in the office environment.

Figure 29 (left) shows some of the devices used in the initial experiments such as Raspberry PIs, BLE beacons, and wireless gateway. Figure 29 (right) illustrates the basic system setup for the experiments. The experimental setup included three wireless scanners that are placed 10 meters apart. The wireless scanners are able to capture wireless packets (Bluetooth advertisement packets) from BLE beacon devices. Wireless scanners (Raspberry PIs) can perform simple processing on themselves and create messages and send the packets to the

back-end server through the illustrated network gateway. The back-end server places the data into a NoSQL database that provides fast access to the data through key-value pairs (JSON objects). Lastly, the back-end server of the Group-In process the raw data through its analytics modules and visualizes the analytics results on a web dashboard as shown below in Figure 30.

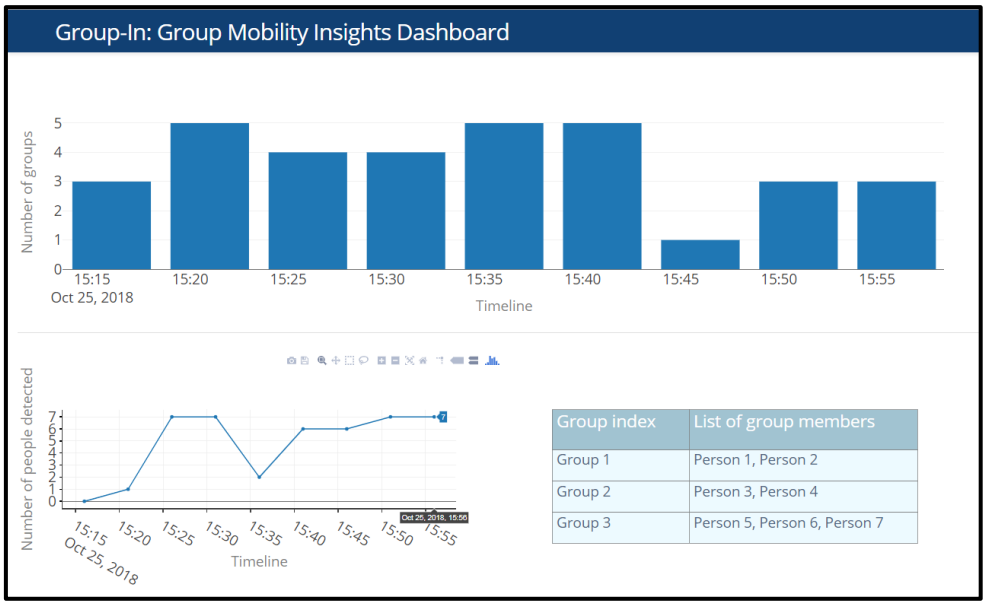

Figure 30. Group-In Web dashboard showcasing the crowd mobility analytics analytics results (from [17])

Reasoning / Justification of the selected model

Group-In selected to leverage three clustering-based approaches as the group monitoring problem through wireless fingerprinting is a specific clustering problem. Moreover, we considered the application of Group-In in many setups without much training and testing. The main reason is applicability in real (wild) scenarios where the data collection can be costly and not feasible due to various reasons such as privacy or others (e.g., smart campus, smart city, and smart office). The main idea was to have an easy to deploy system which generates group monitoring insights from the deployment environment without much configuration changes or long data collection. The model selection focused on using *graph clustering approaches* as the data can be modelled easily as a social network, where nodes represent mobile wireless devices (or people) and edges represent the aggregated distance between these devices. The graph clustering would give set of clustered nodes which would represent the people groups in real-world. Various graph clustering algorithms are tested and showcased in the results. The model selection has been made empirically, where three graph clustering algorithms performed best various scenarios. There is no clear winner graph clustering algorithm as from scenario to scenario on of the three models may perform better. Thus, at the moment Group-In employs the following three graph clustering methods: DenGraph, HCS, and MaxClique.

Results:

Figure 31 shows the initial experimental results of Group-In for controlled and real-world setups. Group-In achieves close to 98% accuracy for 30 sec sampling time and 2 minutes time interval, whereas in the real-world setup the accuracy reaches close to 90% for both metrics. Girvan-Newman (GN), DBScan, MeanShift clustering algorithms are used for comparison. These algorithms are applied directly after the preprocessing phase (replacing centralized or decentralized computing).

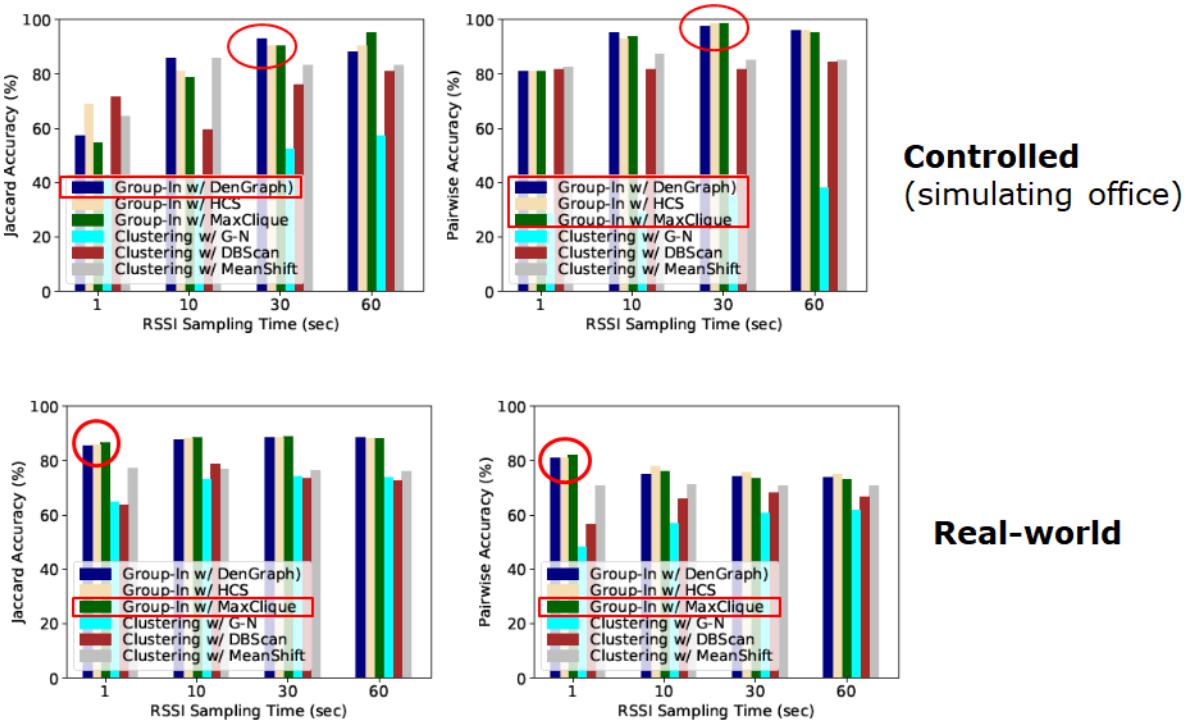

Figure 31. Experimental results for controlled and real-world office scenario setups. Using the centralized computing approach (from [17]).

Figure 32 shows another set of results of Group-In only for controlled setups. In this experiment, the BLE beacon devices are artificially placed together as two sets of devices, representing two groups separated from each other scenario. These two groups of devices are placed with 10 meters distance from each other. Later, the groups are gradually placed closer to each other until they are only 1 meter apart. The groups are statically placed during the experiments. The results in Figure 32 shows that as long as the groups have around 4 meters distance (even though they are static), Group-In can differentiate the people in different groups from each other with about 80% Jaccard accuracy and pairwise differentiation accuracy. On the other hand, in the scenarios where two groups are too close to each other (e.g., with 1- or 2-meter distance) statically or dynamically (groups walking in parallel to each

other), Group-In produces limited group monitoring performance. This experiment shows the feasibility of application of group monitoring through wireless fingerprinting using Bluetooth in various use cases in real scenarios such as in city squares or shopping streets. More extensive results about the functionality 1 can be seen in the full paper which was included in ACM/IEEE IPSN'20 proceedings [17].

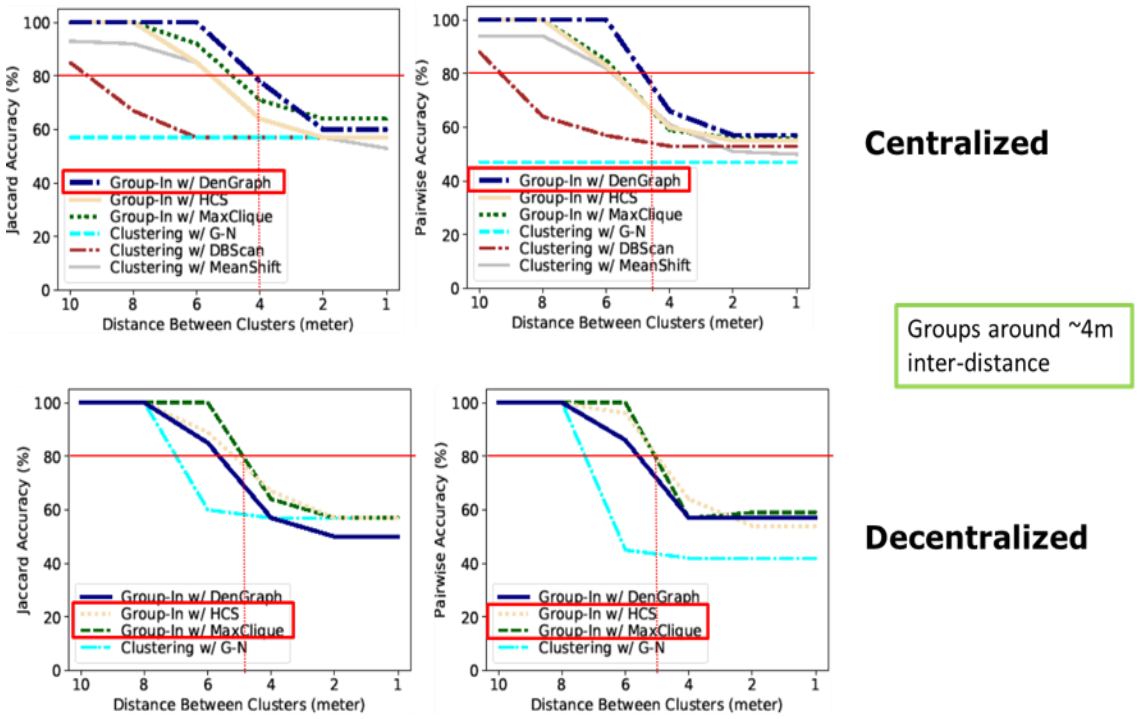

Figure 32. Group monitoring performance of Group-In based on group inter-distances (from [17]).

3.1.2.2 NSE-UC2-Functionality-2: Using multi-modal data for crowd mobility – COVID-19 as a special case

Details of experiments:

For the experiments, we explore the possibilities of using open datasets as data collection may take time and it may be not feasible given the current quarantine conditions. We have searched and found a few open multi-modal datasets which would be fitting to the purpose of the second functionality. We aim towards data collection through short-data collection campaigns where various wireless data sources (e.g., Bluetooth) and auxiliary sensors can be leveraged. For these possible data collection campaigns, we consider both controlled and real-world uncontrolled experiments.

Reasoning / Justification of the selected model

We consider various classification-based ML approaches such as deep learning, random forest, and other classifiers as the target is to detect situations which may lead to the COVID spread. Although the final ML model has not been selected, we consider various options which may include more advanced ML models such as different computer vision frameworks, data programming, recurrent neural networks (RNNs), or AutoML.

Performance analysis

We explore possibility of using real data from sensors of people or room sensors to make a preliminary performance analysis on the feasibility of the functionality 2. We consider the performance analysis for both simulated and real ground-truth data.

Benchmarking

Benchmarking can be done in two ways for functionality 2: 1) Simulation-based, 2) Using real ground-truth data. In some of the datasets, there is no real indication or possibly correlated ground-truth data to COVID proneness. On the other hand, we consider creating probabilistic functions to simulate the ground-truth based on given situations or measurements from the auxiliary sensors (e.g., cameras). In some other datasets, there might be ground-truth data about the people's movement or interactions with each other. For these datasets, the benchmarking is considered for the given ground-truth data. For the data UC2 may collect, we consider collecting ground-truth data for controlled setup and real-world data. Benchmarking would be done in comparison to the existing ML approaches developed by the UC2 or leveraged by the UC2.

3.1.3 NSE-UC4: Logistics in a seaport terminal using AGVs

The activities related to the UC4 on logistics in a seaport terminal using AGVs, that were performed, refer to the feasibility study, the implementation and some preliminary experimentations.

In the feasibility study, the structured analysis approach was followed. In software engineering, structured analysis is a method for analysing systems and developing specifications, data flows and procedures describing a complex system.

The methodology requires the definition of a hierarchical set of data flow diagrams (DFDs) describing the processes on data, the data flows and the data structures used into the system. The description starts with a context diagram (the highest level DFD), depicting the actors communicating with the system and the exchanged data. Then, the process reported in the context diagram is detailed with sub-DFDs. Each dataflow and file or DB used in the system is defined in the data dictionary where it is formally described.

The elementary processes at the end of the decomposition are described in structured English by means of a flowchart. They represent the processing part of the program.

Table 4. DFD legends

Graphic element	Meaning
Green box	Actor interacting with the system. It could be a machine or a human being
Orange border rounded box	Process describing the transformation of the input data flows into the output ones
Magenta bar	File or database
Arrow	Data flow describing the information exchanged between two processes

Data flows and data structures are defined formally in the data dictionary made using an excel file. The hierarchy of the processes starts with the context diagram as shown in Figure 33.

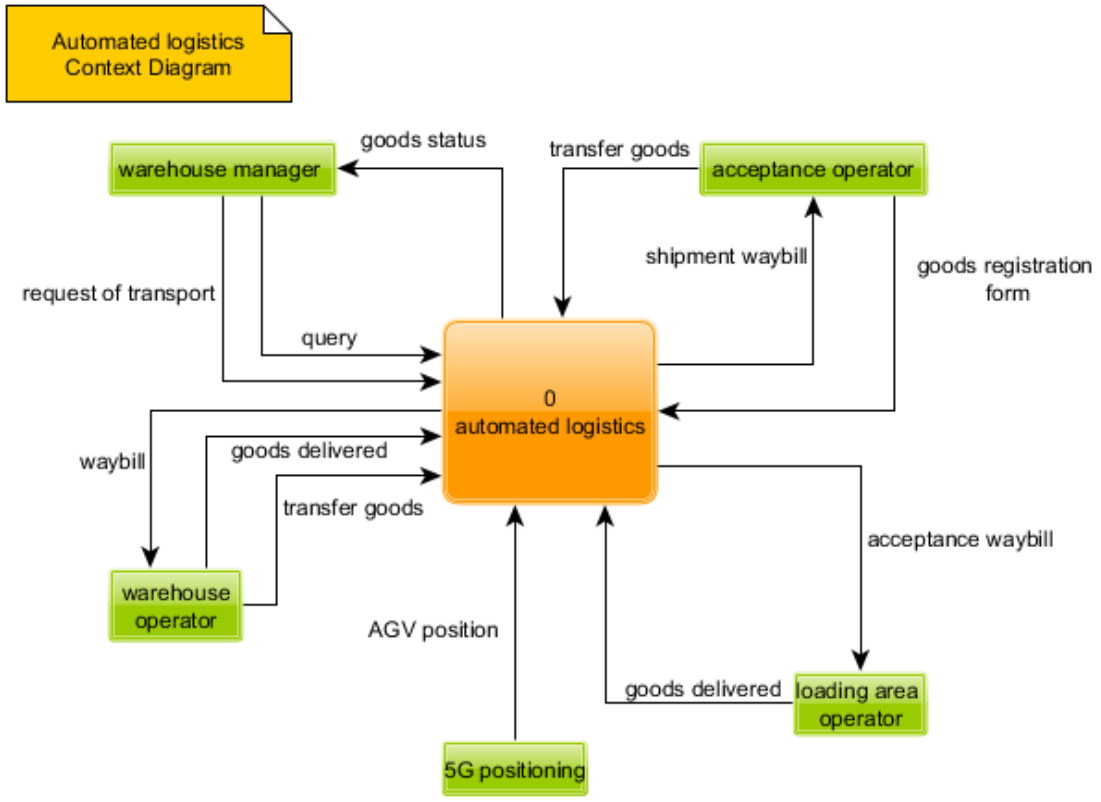

Figure 33. Context Diagram

Starting from the input and output data flow, the hierarchical functional architecture (Figure 34) was defined with all the data sub-flows used between the identified functional elements. In structured analysis the required functions are defined based on the data transformations required to perform the task.

The process #0 is exploded in a set of interacting sub-processes. It includes a file and a DB. The pending service requests made by the terminal operator are placed into the file “pending requests” (a queue). The DB “AGV status” stores the information regarding each AGV in the fleet.

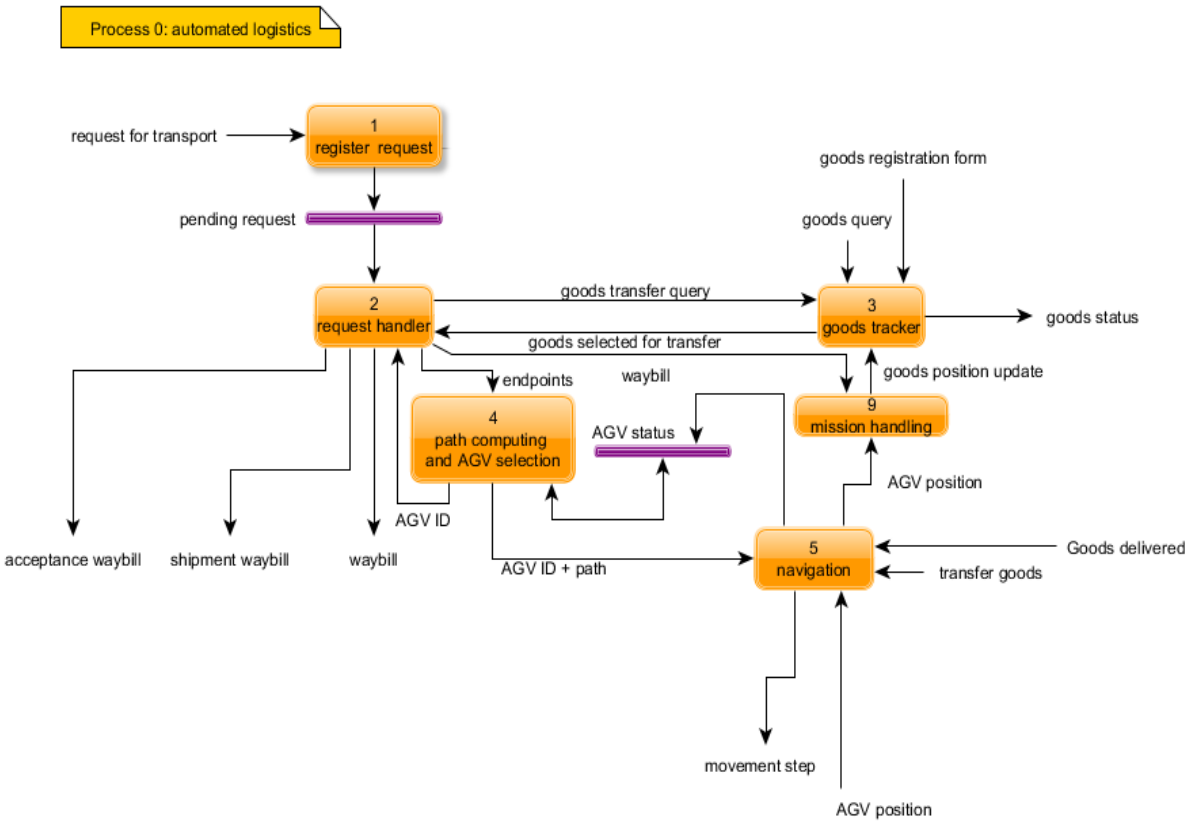

Figure 34. Hierarchical functional architecture

Process #0: automated logistics

Process #1 has the purpose to register an incoming request into the requests file.

Process #2 handles the next request to serve. It checks in the goods DB the objects to pick, determines the endpoints of the travel for the AGV, prepares and sends the waybill to the involved parties to inform them about the transfer.

Process #3 manages the goods DB.

Process #4 computes the possible routes for the available AGVs knowing the endpoints and selects the AGV that has to run the shortest route.

Process #5 controls the navigation of the AGV along its whole route. It uses route information together with the positional ones provided by 5G and defines the next movement step required to follow the predefined trajectory.
Process #9 handles the data related to the mission asking the relational DB for proper updates of the vehicle status.

Implementation

The management system was made using an expert system based on CLIPS that was embedded in a master program made using python. A set of rules were defined for managing the different phases shown in the previous diagrams. Each rule in the expert system is associated to specific co-functions for implementing the computational parts. The co-functions are implemented in python. The set of rules is a basic set that can be extended to cover more complex operations. All rules operate in parallel exploiting the facts that are introduced or created by the rules at runtime. So, choices depend on the current system situation and policies that are introduced in the system to handle the different events.
The A* algorithm for the AGV path computing exploits a specific python library and is interfaced as a co-function with the expert system. It is called every time a new mission is assigned to an AGV.
The computed path is provided to the AGV navigation/control system that manages the movements of the AGV based on the 5G positional data and the path provided by the expert system. The Control system of the AGV, is a nonlinear fuzzy controller. It first normalized both the desired target position (provided by the path computed by the A* algorithm) and the 5G positional data in the range [0,1]. Then, these data are fed to the fuzzy rules. Three sets of fuzzy rules are defined to handle long, medium and short-range movement. In this way it is possible to optimize the vehicle, reaction optimizing the movement accuracy and the AGV responsiveness. The defuzzier is of CoM type. It determines the movement step in terms of acceleration/deceleration and direction for the next hop.
The relational DB including the information about the status and position of the AGV and the freights data (type, weight, size and position) was implemented using MySQL as DB. The DB can be updated and queried from the management system. A specific set of queries were defined and the interface between the system manager and the relational DB was defined and implemented.
The VR environment used for verifying the behaviour of the vehicle is implemented using the Unity 3D game engine. The simulator control functions are implemented in c#. The environment models a real seaport terminal in a realistic way. AGVs are modelled as automated forklifts able to shuttle freights (boxes). The VR environment communicates with

the AGV control system using dedicated sockets connecting the simulator to the python functions implementing the AGV controller. It will be possible to track the positional error runtime both in a numerical and a visual way. In the VR, an "avatar" of the AGV will be also shown to allow the visual check of the accuracy that can be achieved in positioning respect to the optimal path computed by the A* algorithm.

Plan for future work

The next steps relate to the integration and optimization of the different functions including the management system controlling the missions, the relational DB hosting the information on freights, ships and AGVs, the AGV controller and the VR environment.

As soon as the model of the 5G positioning will be available, it will be also integrated and preliminary validated.

After these steps the performance of the 5G positioning system to guide an AGV in the seaport will be evaluated in a series of test conditions. These data will provide the final outcome of this activity.

3.1.4 NSE-UC5: Transportation optimization based on identification of traffic profiles

Exploratory analysis using clustering techniques

In D4.1 [69], the application of different clustering techniques for the detection of Points/areas Of Interest (POIs) was successfully tested. More specifically, Hierarchical Cluster Analysis [58], and two density-based techniques, namely Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [59] and Ordering Points To Identify the Clustering Structure (OPTICS) [72] were employed in order to determine areas through the grouping of UE positions at the various intervals, with satisfactory results.

In this deliverable, we use the same methods exploiting a different type of input, going from single points to small-scale trajectories. This way, we attempted to group at a small-scale level the individual movements in the area under investigation. By grouping these small trajectories, we allow for the understanding of similar but not in fact identical routes. For example, a pedestrian walking on a main avenue and turning right to a road, can have a similar to another pedestrian trajectory walking in an opposing sidewalk or with a slightly higher speed. Similar approaches, of grouping tracklets has been used previously in [30]. Given the scale of timeslots (15 mins) in the original dataset, we do not expect grouping of motions at a micro level, however some basic group of movements are expected to be found. Indeed, in order to focus in the significant more common and representative "trajectory clusters", in the following text we present density-based techniques, as these methods allow the exclusion of

"outliers" and may focus more on the common trajectories, excluding the less common/sparse ones.

Two different approaches were explored:

- Clustering the coordinate values of the three consecutive points of the trajectories/tracklets, as a standard baseline approach.
- Employing LSTM autoencoders in order to derive a latent representation of the tracklets and then cluster.

For model tuning, a hyperparameter search was performed in both cases:

- For DBSCAN, the maximum distance between two samples (eps) was explored (smaller eps resulting in smaller and denser clusters).
- In OPTICS, the search explored (a) the number of samples in a neighborhood for a point to be considered as a core point (min_samples), (b) the maximum distance (max_eps) between two samples for one to be considered to be in the neighborhood of the other (again smaller values resulting in smaller and denser clusters), and (c) the minimum steepness (xi) on the reachability plot that constitutes a cluster boundary.

The results were inspected visually, however the Silhouette Coefficient [73] as also employed as a metric that can account for both intra-cluster and nearest-cluster distances for each sample for clustering quality verification. Indeed, the results achieved high values of average Silhouette scores especially for smaller very dense clusters.

Indeed, depending on the hyperparameters chosen, clusters of different density and scale are determined. For instance, as shown in Figure 35 (a) more than 800 clusters were detected using OPTICS when clustering the trajectory coordinates, and an average silhouette score of over 0.95 was achieved. Similarly, by increasing the min_samples and max_eps values, fewer clusters, but more clearly defined can be seen with a slightly lower Silhouette score but again over 0.9 Figure 35 (b).

As is evident in Figure 35, in the second case, a number of points are not shown in order to improve visualization, as they are outliers. Indeed, the clusters are more distinct (as the criteria chosen requires a larger number of trajectories to form a cluster) and are of greater scale (as the distance criterion has been increased).

Figure 36 presents a comparison of the clusters determined using (a) clustering of trajectories and (b) using the LSTM autoencoder and then clustering the latent representation of the trajectories. It should be noted that, in this case, major differences in the clusters determined occur and different points are removed as outliers. However, it is clear that in the case of Figure 35 (a) clusters are very broad and concern a wider area, while in Figure 35 (b) clusters

are more detailed. The silhouette scores in these cases are similar, i.e., 0.79 for (a) and 0.71 for (b) respectively. However, their values are acceptable.

Indeed, this proves that, as expected, the LSTM autoencoder provided a latent representation of the tracklets/ trajectories that was intrinsically able to help the clustering method differentiate and group tracklets in a more detailed manner, possibly including in this latent representation nuances not taken into account in the first approach, i.e. other similarities in trajectories that cannot be described based on a simple distance criterion.

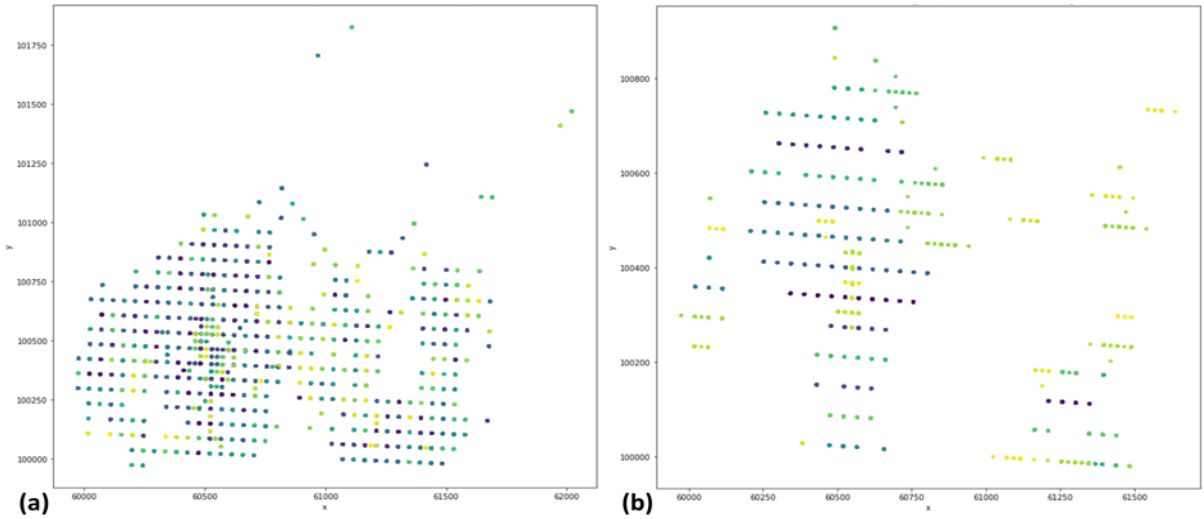

Figure 35. Example trajectory clustering (OPTICS): (a) min_samples=25, max_eps=15, xi=0.005, (b) min_samples=125, max_eps=55, xi=0.005. Outliers are not shown for improved visualization

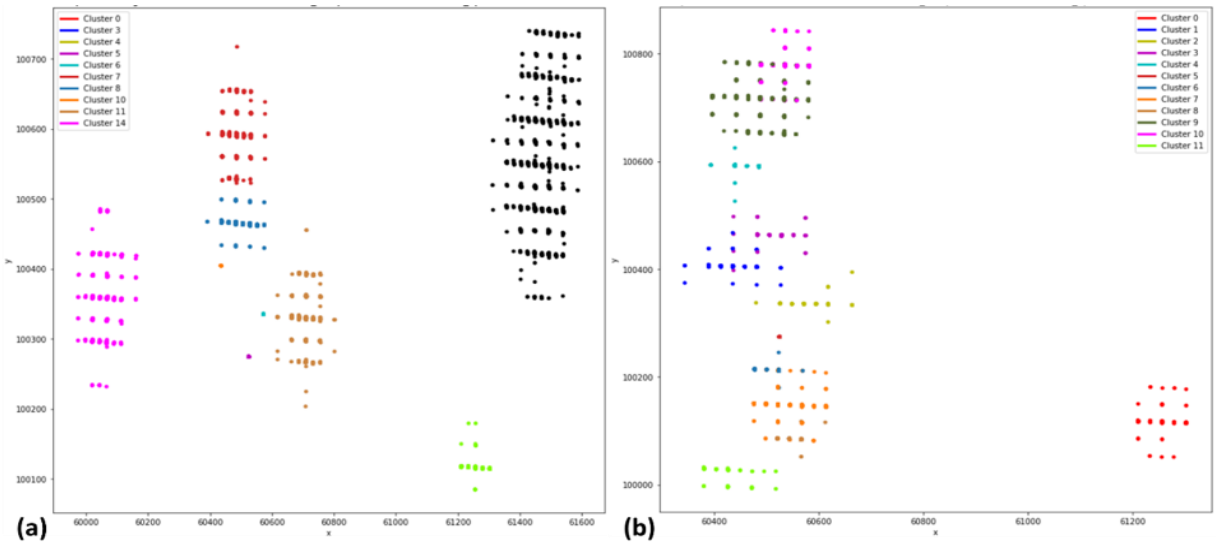

Figure 36. Example comparison of (a) density-based clustering and (b) LSTM autoencoder followed by density-based clustering, zooming in a specific subarea. Outliers are not shown for improved visualization.

Trajectory prediction using LSTMs

Long short-term memory (*LSTM*) artificial neural networks were used for trajectory prediction, i.e. so as to predict the UEs next positions. In order to avoid padding effects in the model training process, having ensured that the same trajectory length through interpolation and a time-shifting window, 12 points were used as model input and the next 3 points were used as the targeted positions to be predicted.

While the LSTM tuning process is ongoing, some initial results are available and are included in this section. After some testing, the network was empirically set to have two LSTM layers with 128 neurons and a single dense output layer for the prediction of the coordinates of the 3 positions. In addition to this, different learning rates with decaying values based on model improvement after a (variable) number of epochs were tested. Early stopping criteria were also set to avoid overfitting. In detail:

- *Learning rate scheduling:* Learning rate was set to 0.001, 0.0005 or 0.0001 and was reduced by a factor of 0.2, 0.5 or 0.7, if there was no improvement in validation set loss for 15 epochs. A minimum value for the learning rate was set based on the initial value.
- *Early stopping:* The model stops training when there is no improvement in the validation set loss for more than the set number of epochs, i.e. in this case 30 or 50.
- *Dropout rate:* Set to 0 or 0.3.
- *Optimizer:* Adam.
- Loss function: Mean Squared Error (MSE).

Table 5. Top-10 LSTM models: Error results

Model No	Test set					Validation set				
	P1 error (m)	P2 error (m)	P3 error (m)	Average error (m)	Frechet distance (m)	P1 error (m)	P2 error (m)	P3 error (m)	Average error (m)	Frechet distance (m)
1	31.98	54.19	77.89	54.69	83.60	31.25	54.55	78.54	54.78	84.03
2	33.69	55.31	79.26	56.09	85.61	33.31	55.89	80.48	56.56	86.44
3	36.39	58.04	81.48	58.64	88.66	35.27	57.60	82.38	58.42	89.04
4	36.32	56.75	83.78	58.95	89.47	35.87	57.85	85.63	59.79	90.86
5	40.12	54.89	82.51	59.17	88.09	39.41	55.45	83.60	59.49	88.94
6	37.88	57.99	82.01	59.29	88.30	36.91	58.67	82.93	59.51	88.92
7	37.81	58.86	81.25	59.31	87.57	37.74	60.53	84.40	60.89	90.30
8	38.52	58.71	82.18	59.80	89.97	38.50	60.12	84.16	60.92	91.45

9	38.70	57.55	83.51	59.92	90.81	38.27	58.35	85.41	60.68	92.40
10	39.06	60.65	81.53	60.41	88.51	38.06	60.84	81.52	60.14	88.34

Lastly, results were explored based on the error Euclidean distance for the 1st, 2nd and 3rd point (P1, P2, P3), their average value and discrete Fréchet distance (F) [74] – a metric used to measure distances between curves- for train, validation and test datasets.

In . , the best models in terms of average 3-point test error are presented, while Figure 37 shows some example results (trajectory inputs in yellow, the predictions in red and the true values in blue).

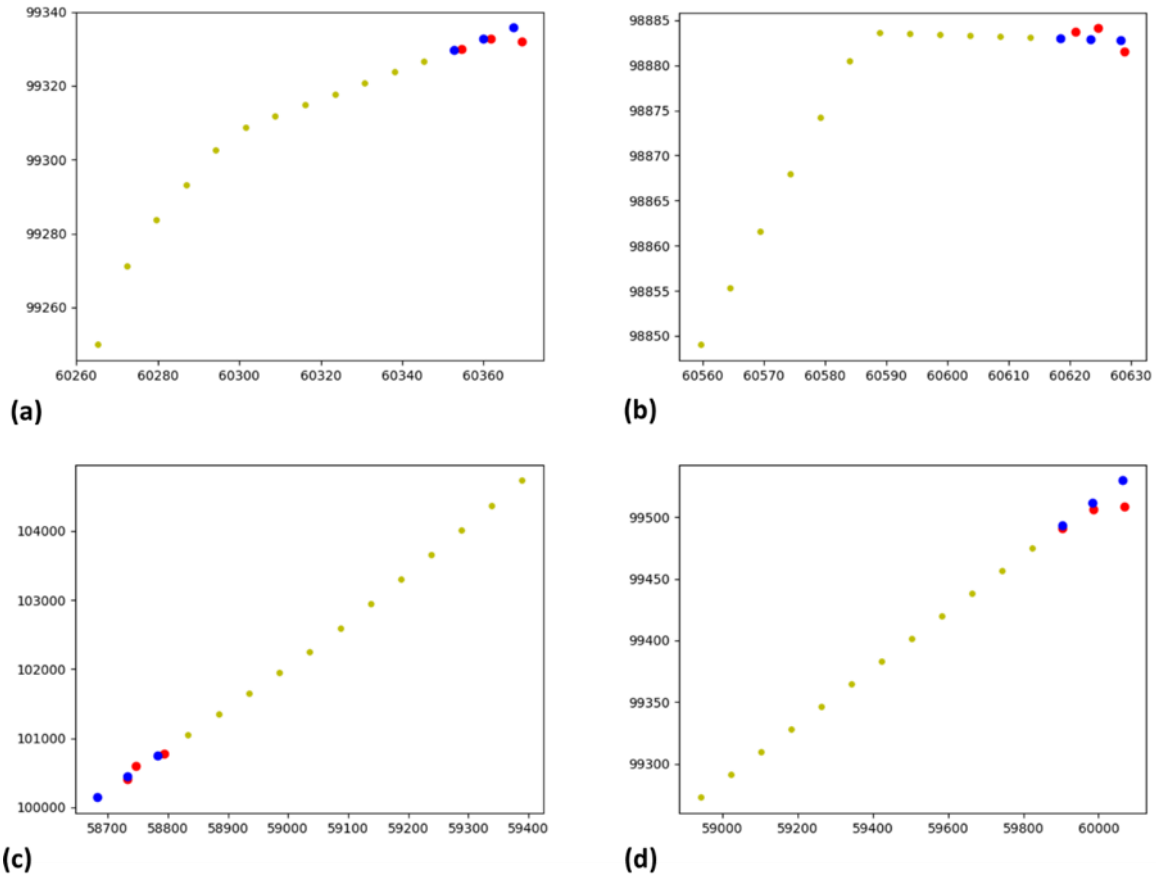

Figure 37. Example trajectory predictions with best LSTM model. Trajectory input presented in yellow, predicted values in red and true next points in blue.

Overall, the balance between the validation and test set error shows a satisfactory tuning process, i.e. overfitting was successfully avoided. Furthermore, the error for the first point prediction is smaller than in the cases of points 2 and 3 and close to 32m. This is expected, as the further into the future we predict, the more uncertain the prediction is. Figure 38 shows the distribution of errors for P1, P2, and P3. Indeed, the majority of cases have an error smaller

than 25m. It should also be noted that in the case of the test set, more that 84% of the P1 prediction error is less than 45m.

Plan for Future Work

As a next step, some extreme values (very few) will need to be further explored in order to determine any outlying behavior. However, the dataset itself and the assumptions made during the dataset preprocessing phase must be considered, as they may also have an effect on the overall error. In fact, the measurement timeslots were set to 15 minutes, allowing for ample time for the UE to move through the area in varying speeds. Furthermore, the assumption made that the multiple measurements within the timeslot are equally distributed through time and the interpolation process chosen may have a significant effect in the actual position coordinates chosen for the training phase, as well as for measuring the prediction error. In this context, the error achieved can be considered within an accepted range. However, in the future, the model building and tuning as well as the preprocessing procedures will be further investigated and enhanced in order to achieve better results, including the application of Sequence-to-Sequence Prediction models [32].

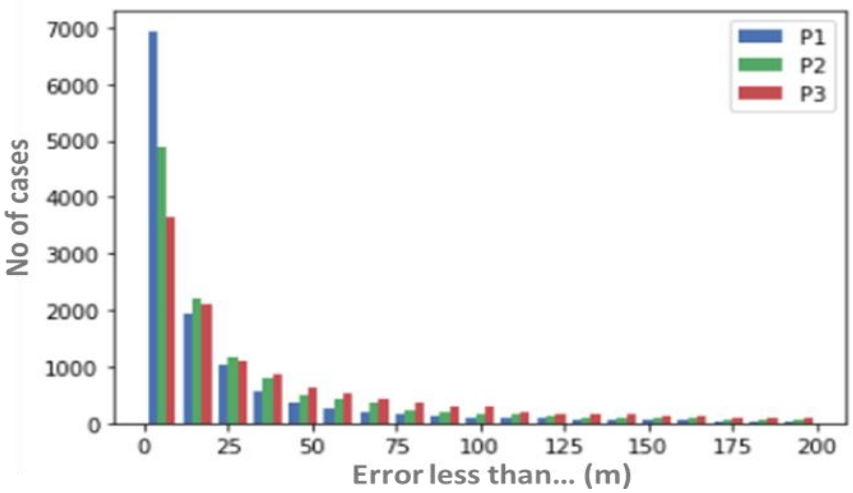

Figure 38. Error (m) distribution for predicted points P1, P2, P3

3.1.5 NSE-UC6: Positioning and Flow Monitoring for Controlling COVID-19

3.1.5.1 Implementation details and Initial Results

This section summarizes the initial results obtained from crossing the mobility data of the operators with the contagion data. In this initial exploration, the data of total population flow between regions and the number of total cases detected have been observed.

Mobile operator data

The operator data, processed by the National Statistics Institute (INE) of Spain, contains information about cells of approximately 5000 users. In the original study, user mobility was tracked over several days in 2019, calculating the number of users (flow) moving between pairs of cells. The analysis was carried out with data from November 2019, under the hypothesis that they are similar to the mobility prior to the confinement in March 2020, which is believed to have spread the virus throughout the whole province of Málaga.

The data comes in two different datasets:

- Absolute mobility per population cell: for each cell in the province, the number of people leaving to any other cell and the number of people coming are measured, along with the total population.
- Pair-wise mobility: for a given pair of origin and destination cells, the total number of people moving is measured, as long as it is higher than 100 (for privacy preservation)

Contagion data

Contagion data is aggregated per cell, giving the total number of detected cases up to March 23, 2020. The time span covers cases that most likely originated before the confinement (March 13, 2020), as a direct consequence of the mobility of previous days.

Absolute mobility results

In these first results, the absolute mobility data per cell will be crossed with the contagion data. The initial hypothesis is that the greater the mobility, the greater the probability of contagion. Figure 39 shows the density of cases in each cell. To verify the hypothesis, we will superimpose the number of cases with the absolute mobility. In the case of mobility, three types of mobility can be considered: incoming (i.e. the number of visitors, Figure 40), outgoing (i.e. the number of people leaving a cell, Figure 41) and combined (i.e. the total of the previous two, Figure 42).

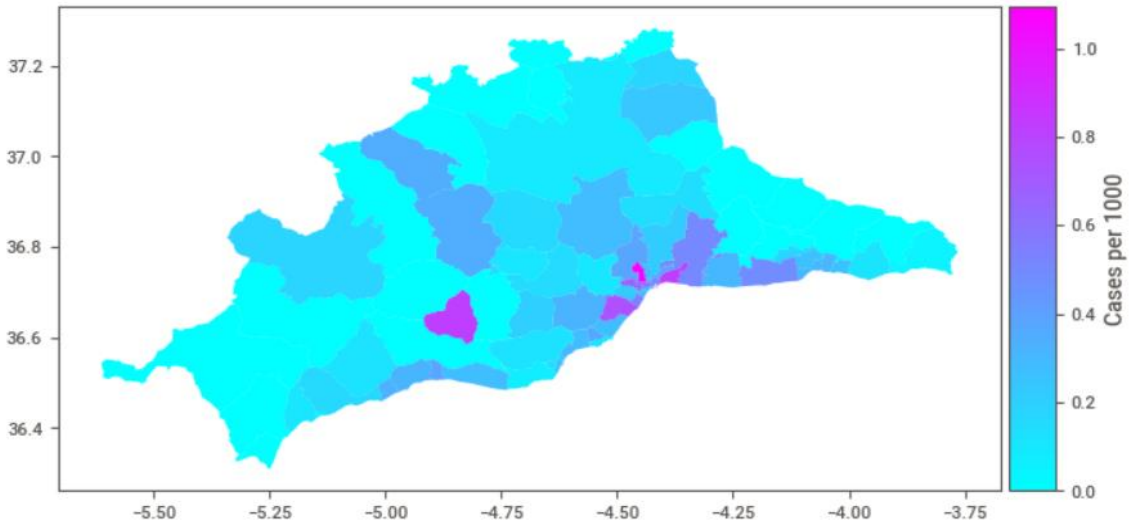

Figure 39. Case density, given as number of cases per 1000.

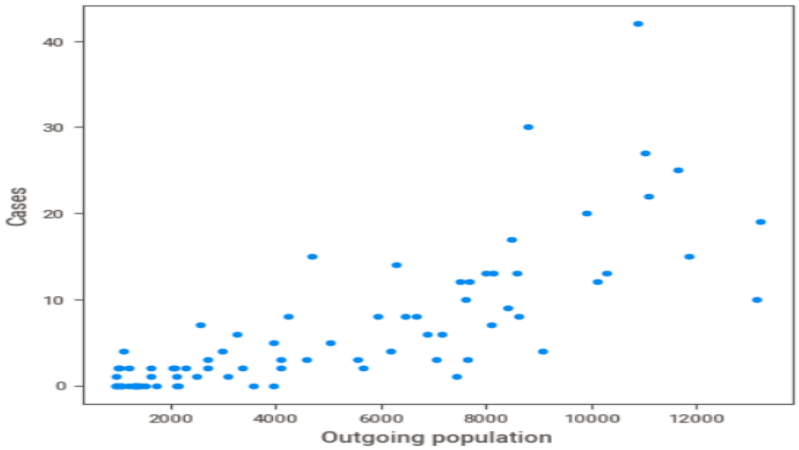

Figure 40. Number of cases vs outgoing population.

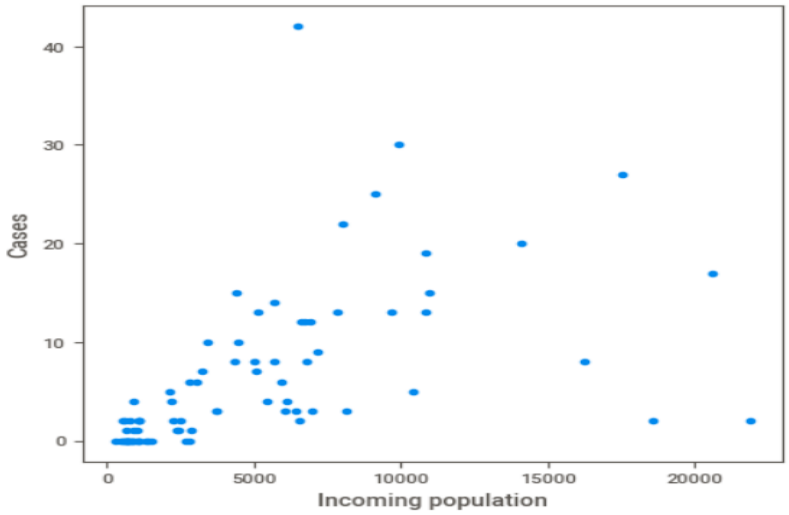

Figure 41. Number of cases vs incoming population.

It can be seen how there is no strong correlation between mobility and cases, so the original hypothesis is not verified. However, it is observed that low mobility seems to guarantee a low rate of infections, conforming a boundary of maximum number of cases based on mobility, as shown in Figure 43. In other words, the higher the mobility, the higher maximum number of possible cases, although apparently the number of them depends on other factors.

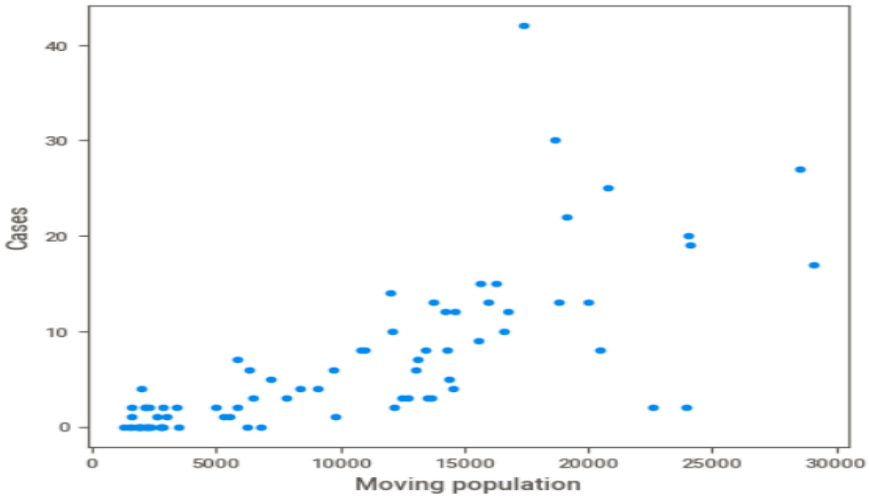

Figure 42. Number of cases vs total flow of the cell.

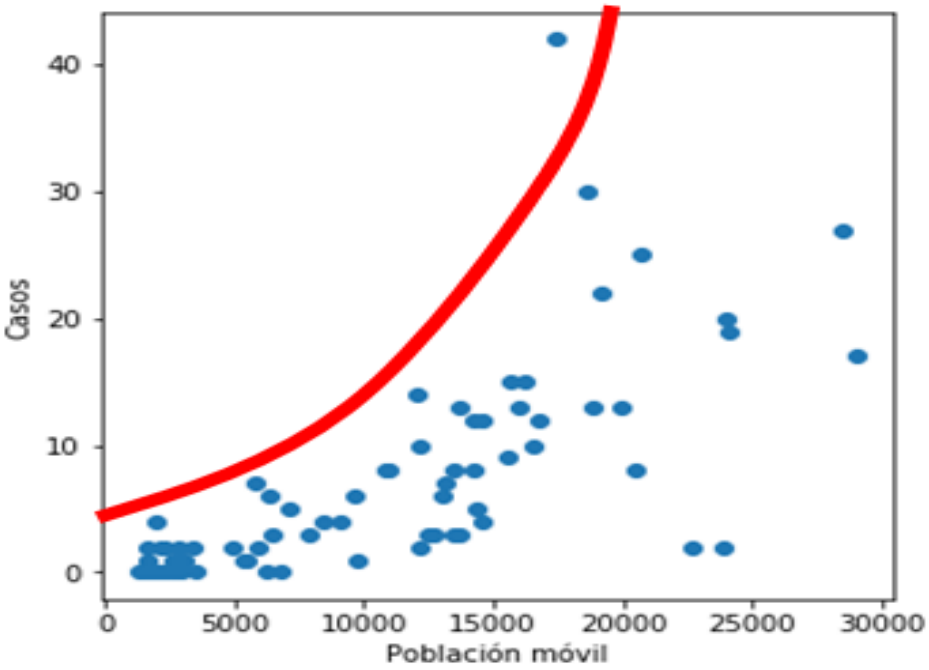

Figure 43. Lower mobility caps the number of cases.

Mobility analysis with hot zone discrimination

In these results, a classification of cells by quartiles has been done according to the number of infections. This classification can be seen in Figure 44.

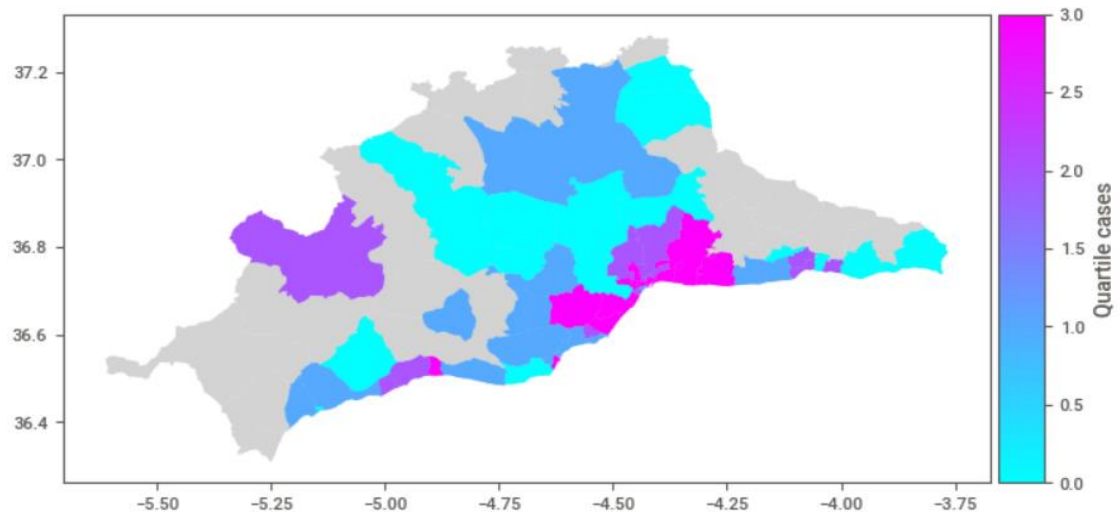

Figure 44. Cells classified in quartiles.

It is important to note that a logical deduction that could be made is that the distribution in quartiles is highly influenced by the population. To eliminate any possible bias, the density of cases per population will be used (Figure 45).

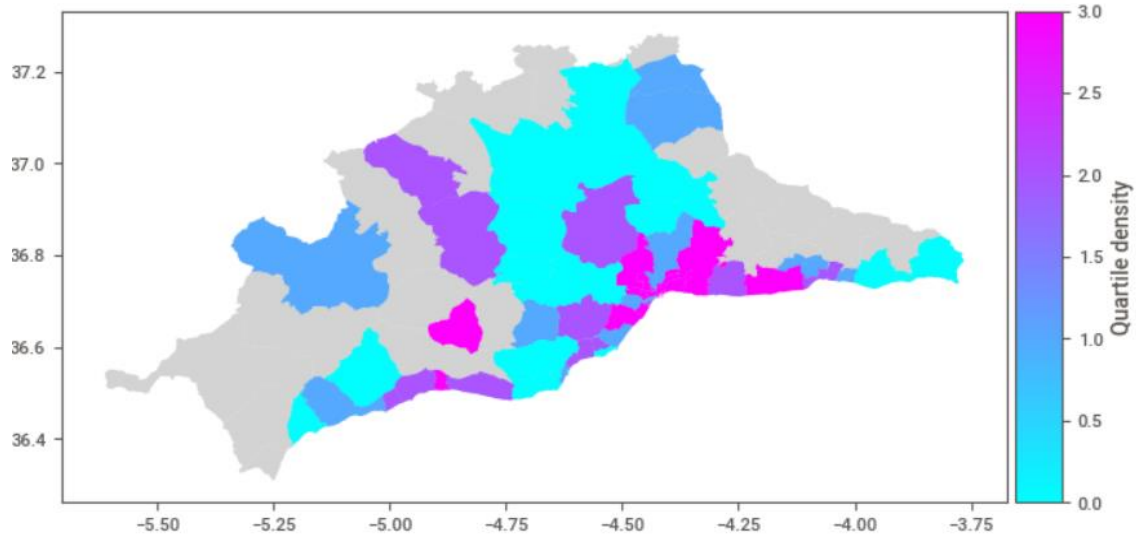

Figure 45. Cells classified in quartiles by the density of cases.

The initial hypothesis is that cells with more cases will have more traffic with cells from the first quartile. To test this hypothesis, we represent the number of cases as a function of the traffic distribution with cells from each quartile, as shown in Figure 46 to Figure 51.

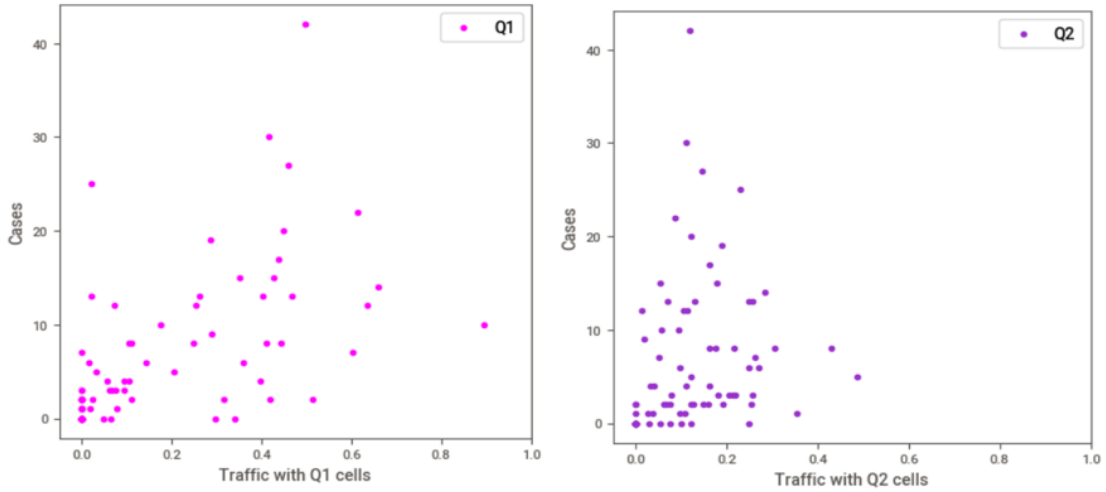

Figure 46. Number of cases as a function of traffic with cells from the first (left) and second (right) quartiles.

Figure 46 shows a trend similar to the general case: the proportion of traffic with cells in Q1 and Q2 increases the maximum number of cases.

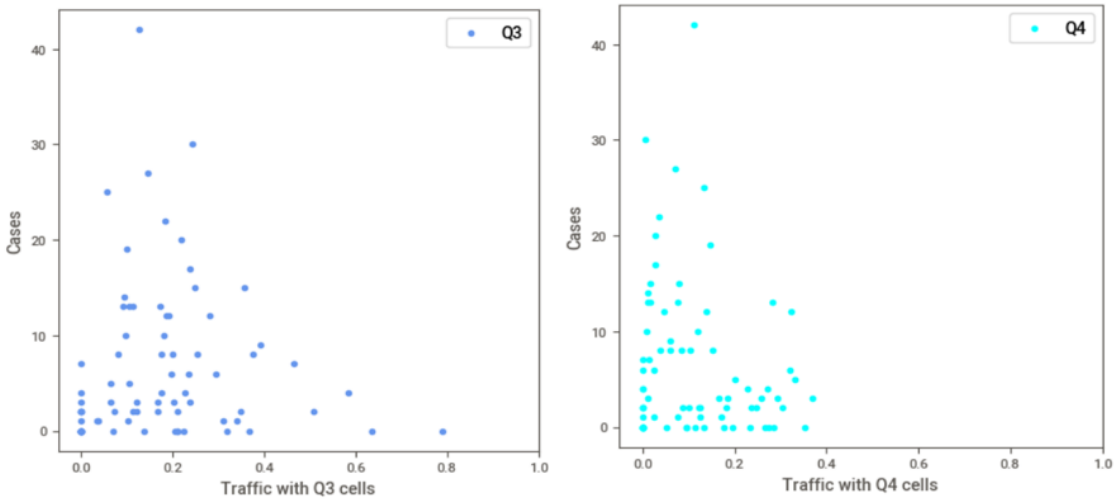

Figure 47. Number of cases as a function of traffic with cells from the third (left) and fourth (right) quartiles.

Figure 45 and Figure 46 show that as the proportion of relationships with cells in a high quartile increases, fewer cases are seen. This trend is seen more clearly in Figure 47.

From these results it can be concluded that, although a high traffic exchange with hot cells has an influence on the probability of having many cases, they are not the determining factor. It should be noted that this analysis has only been carried out on cells that have cases.

Anomalous cells

In this section, the results of the abnormal cells will be shown, that is, those of the first quartile of infections and with little mobility, and those of the fourth quartile with high mobility. Detecting these extreme cases can lead to their analysis in greater detail to see why the former perform poorly and the latter perform well.

In Figure 48 the abnormal cells are shown; those of Q1 with low mobility are marked in red, and the ones of Q4 with high mobility in green. Figure 49 shows the traffic distribution of cells marked in green and Figure 50 that of cells marked in red.

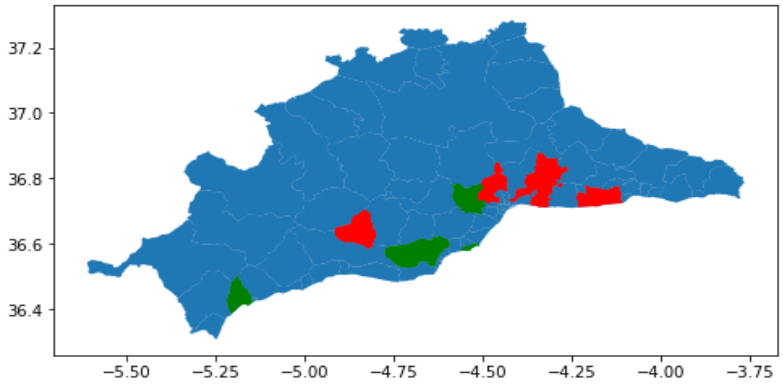

Figure 48. Anomalous cells.

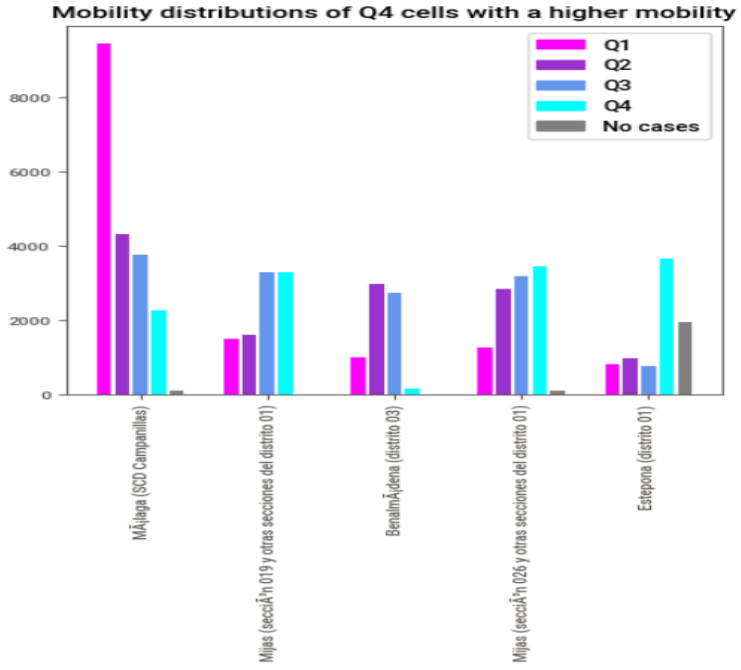

Figure 49. Mobility distributions of the cells with the best performance.

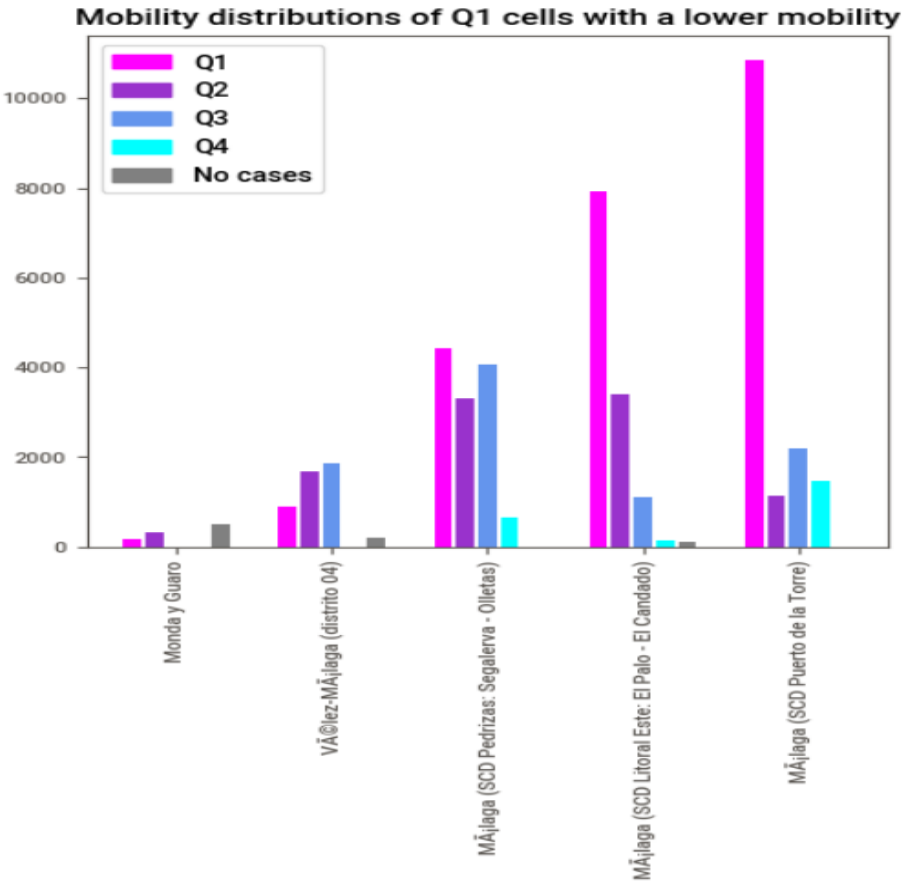

Figure 50. Mobility distributions of the cells with the worst performance.

At first glance, we can see in Figure 51 that the traffic distribution of most of the best cells excludes the traffic with cells from Q1. However, in the case of *Málaga (SCD Campanillas)* we observe a strong component of traffic exchanged with cells from Q1. In Figure 50 we see cells of Q1 that have little mobility and many cases. In three of them, the component of exchange with Q1 cells is very important. An interesting exception is the *Monda y Guaro* cell, which has a very low mobility. It should be noted that this may be due to the fact that exchanges with cells in which the traffic is less than 100 are not collected. Figure 51 shows the same information for cells without cases and with greater mobility. A variety of situations are observed in this case; from cells with considerable interchange with cells Q1 (*Coín* and *Ojén e Istán*), as well as cells like *Casares* that do not have any mobility with cells Q2 or higher. However, unlike what was observed for the best Q4 cells, the total mobility in these cases is lower.

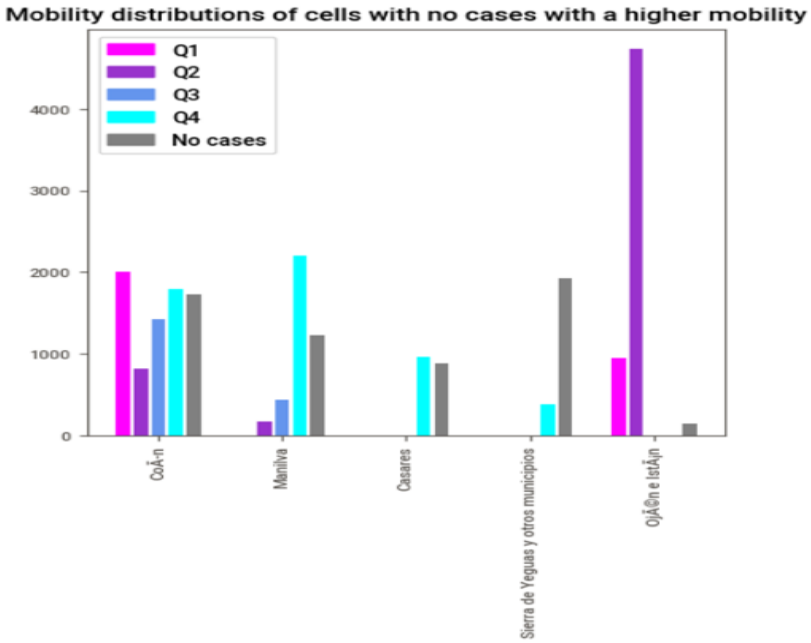

Figure 51. Mobility distribution of cells without cases and a high mobility.

Conclusion

Throughout this study, various mobility factors and their influence on the number of cases have been explored. Some weak relationships have been observed, which in general reflect what was expected: that with less mobility, fewer cases occur. The opposite is not necessarily true; although the probability distribution's higher limit increases, it does not guarantee a high number of cases. Likewise, the relationship with cells with a high density of cases has a similar effect; a high proportion of relationships with cells with many cases increases the maximum possible number of cases.

3.1.5.2 A summary of the activities and progress updates of SAMSUNG as a rapporteur in Europe for E4P developing COVID-19 proximity tracing solutions

"Europe for Privacy-Preserving Pandemic Protection" (E4P) Industry Specification Group was created by ETSI as a global initiative. E4P supports the ultimate objective to save lives by contributing to break COVID-19 transmission chains. E4P aims to define a global framework for digital contact tracing systems that would facilitate data privacy, security, scalability and interoperability. Participants to the E4P meetings are representatives of global organizations, with members ranging from government and EC representatives, vendors, operators and research bodies to ethics, legal and cybersecurity players. E4P standardization scope is partially related to COVID-19 related work in LOCUS as described below.

In E4P work roadmap three priorities (stages) related to Synchronous Contact Tracing Systems have been defined

- **PRIORITY 1** - To specify system and facilitate international interoperability using smartphone-based Bluetooth low energy in Europe based on DP-3T/GAEN and Robert/Desire digital contact tracing protocols. System should be compliant with GDPR and EC guidelines.
- **PRIORITY 2 -** To improve the systems specified by adding new functions or devices e.g. Token, new API and Operating Systems, new proximity detection measurements Starting from their description in group report to derive related requirements and consequently the related technical solutions.
- **PRIORITY 3 -** To explore and develop new systems that would better fit the need e.g. cellular network based solution.

Priority 1 work started in June 2020, five work items have been approved and the development of the related documents has progressed since. Priority 2 and 3 work is planned to be started in Q1 2021.

E4P hosts plenary and drafting session meetings every 2 weeks to support the delivery targets. These documents which are currently (Dec 2020) being finalized as part of Priority 1 (draft versions are available at https://docbox.etsi.org/ISG/E4P/Open) include

- **Comparison of existing pandemic tracing systems** (Group Report)
- **Requirements for pandemic contact tracing systems using mobile devices** (Group Specification)
- **Device-based mechanisms for pandemic contact tracing systems** (Group Specification)
- **Back-End mechanisms for pandemic contact tracing systems** (Group Specification)
- **Pandemic proximity tracing systems: Interoperability framework** (Group Specification)

In E4P Samsung has been a rapporteur of the 'Device-based mechanisms for pandemic contact tracing systems' Group Specification (GS) and as part of this role organized drafting sessions and attended plenary, rapporteur and drafting sessions of complementary specifications. On the technical side, so far Samsung contributed to the development of the Device-based mechanisms GS and also contributed to E4P reference pandemic contact tracing systems architecture work, Back-End Task Force and high level and interoperability requirements definition. In addition, Samsung provided E4P status and roadmap presentations to LOCUS WP5 and suggested directions for its ongoing COVID-19 use cases work, identifying potential LOCUS contributions to E4P in the future.

3.2 Virtualized Machine Learning Pipelines

This section provides a brief description of the initial experimental work carried out for the virtualization of the NSE UCs machine learning pipelines. Starting from the design principles described in section 2.2, this work is being performed following an incremental approach in two main steps:

1. Deployment and test of NSE UC virtualized machine pipelines on existing opensource AI/ML frameworks
2. Integration of the NSE UC virtualized machine pipelines with the LOCUS virtualization platform developed in WP4

The first step aims at carrying out an initial validation of the NSE UC machine learning pipelines in virtualized environments, leveraging on opensource frameworks that aims at delivering pipelines as services following the virtualization principles described in in section 2.2. In particular this work is performed with Seldon [75] and Kubeflow [76]. On the other hand, the second step represent the final target for the full integration of the NSE UC machine learning pipelines in the LOCUS platform, that allows to expose them as on-demand localization analytics services towards the 3rd party vertical applications.

The first exemplary NSE UC functionality experimented (and reported in this deliverable) is the UC1 Functionality-1 for identifying crowd mobility patterns. In particular, the available machine learning pipeline software code has been preliminarily deployed and validated according to the first experimentation step above on existing opensource AI/ML frameworks, as described in the following sub-section.

3.2.1 Experimental scenario and setup

This section describes the initial experimental and validation work of virtualized ML pipelines, as a preliminary activity towards the realization of the LOCUS localization analytics as a service virtualization platform. This first step targets as final objective its integration with the LOCUS cloud-native and NFV-oriented platform able to support the lifecycle management of ML pipelines.

Figure 52 shows the reference ML pipeline for this experimental work, which is related to the NSE UC1 Functionality-1 and involves several steps like data preparation and ML and Data Mining workflows.

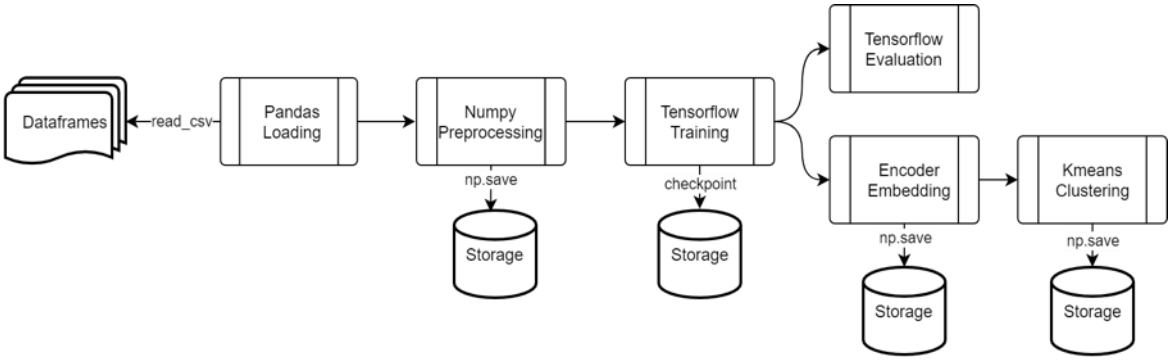

Figure 52. LOCUS NSE-UC1 Functionality-1 pipeline

In the experimental scenario, the logic and runtime environment of these individual steps is encapsulated in independent containers, which are then managed through an AI/ML framework. In particular, Kubeflow [76] is used, as open-source solution to test, deploy and maintain a machine learning pipeline in a microservice environment. Kubeflow is a scalable ML platform that runs on Kubernetes and exploits all its capabilities to facilitate the development and deployment of virtualized ML solutions.

A pipeline in Kubeflow is a graph description of an ML workflow, namely a set of pipeline components and how they combine. Thus, once the final user has translated the ML workflow into the python file that represents it in the Kubeflow format, the pipeline can be deployed and maintained in a fully automated way, exposing the user the needed API. Kubeflow also includes several components for model serving, like Seldon core [75] and Tensorflow (TF) serving [77]. Such components can serve a trained model available in a local or remote repository as a REST microservice in Kubernetes. Another key tool is Argo Workflow [78], a Kubernetes-based workflow engine that enables the composition of containers as steps of a pipeline or nodes of a Directed Acyclic Graph.

Figure 53 shows a comprehensive list of the Kubeflow components, while Figure 54 shows how they can be mapped in the UC1 pipeline. In particular, the proposed experimental scenario handles the definition and lifecycle management of the pipeline's steps with Argo, while the model training and serving are carried out by the suitable Kubeflow components like TensorflowJob operator [79] or Seldone core.

This experimental scenario is deployed in an NFV-oriented environment with Kubernetes as a container orchestrator and an OpenStack Queens instance as IaaS infrastructure manager.

A preliminary set of tests and validations of the NSE UC1 Functionlity-1 model serving and REST interaction with it have been carried out in a Minikube instance running in a Openstack VM with 24 CPUs, 40GB of RAM and 50GB storage. The Minikube instance hosts the Kubeflow components needed to serve a model and handle the LCM of the corresponding containers: Seldon core and TF Serving. Alongside with the model serving components, the testbed

includes Istio [80], an ingress Kubernetes gateway used to wire, and Minio [81], a Kubernetes-native object storage solution to save and retrieve the trained model. Figure 55 shows the logical view of the testbed components, while Figure 56 shows a high-level description of the steps involved in the deployment of the demonstration platform.

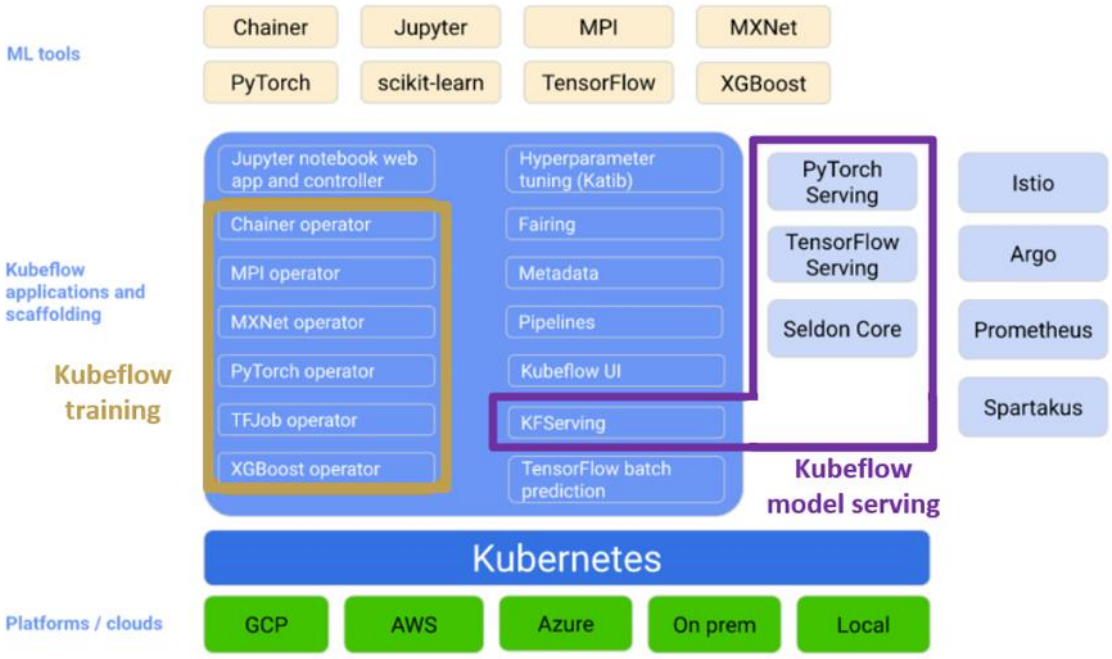

Figure 53. Kubeflow components.

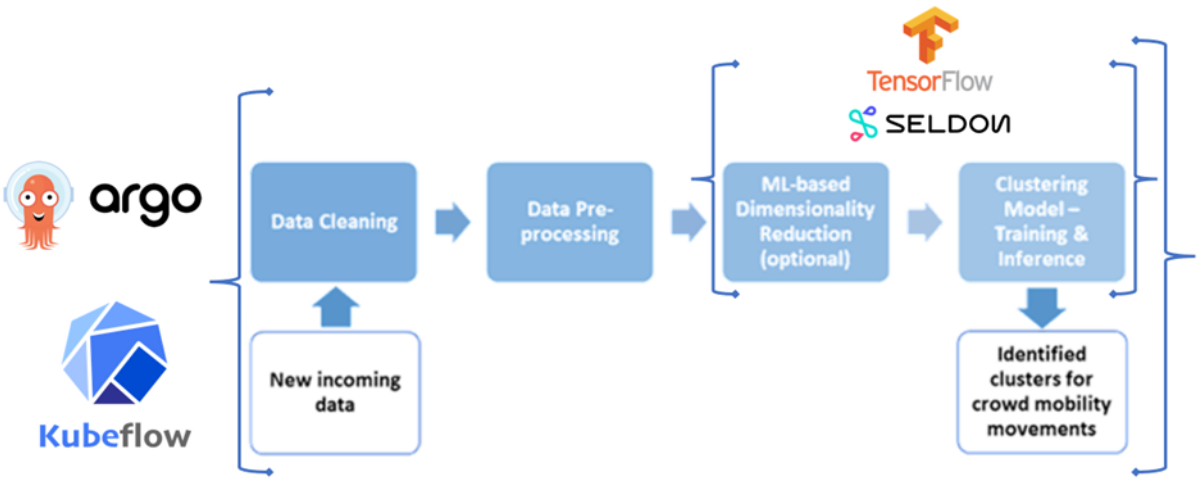

Figure 54. Mapping of Kubeflow components into the LOCUS NSE-UC1 pipeline.

The deployment of the testing platform starts with the setup of a Minukube instance that runs a single-node Kubernetes cluster. Once the cluster is up, the ingress controller, the object storage and the serving components are initialized. At this point, in order to serve a model as

a REST containerized application, an instance of the trained model should have been saved in a Minio bucket and Seldon has to be configured to access and deploy the model.

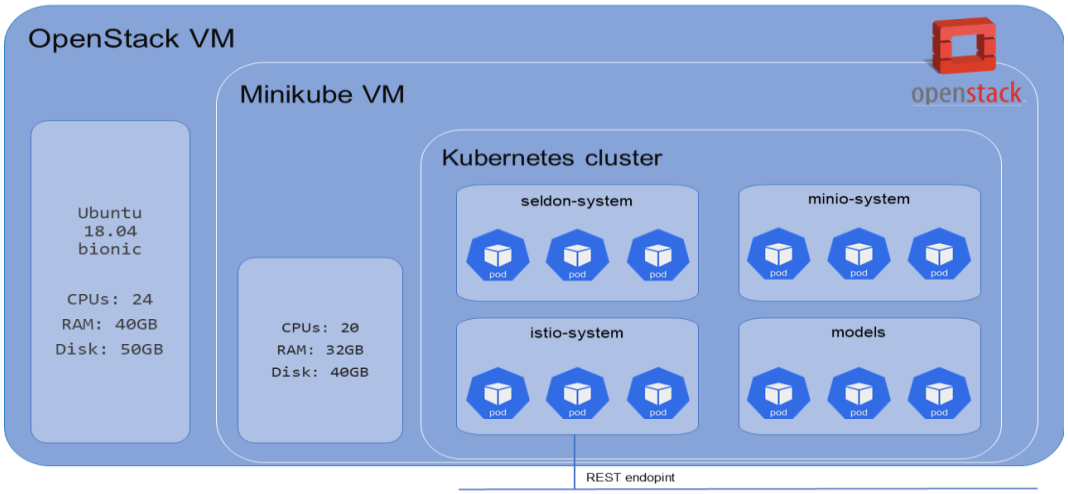

Figure 55. Logical view of the testbed's components.

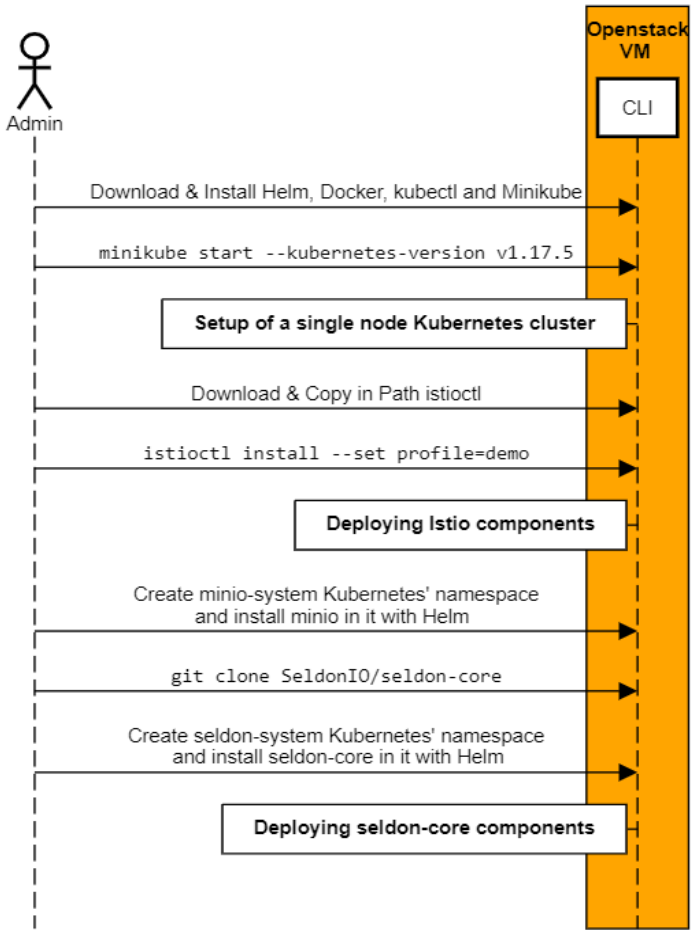

Figure 56. High-level sequence diagram of the testbed deployment.

To deploy a model in Seldon a *SeldonDeployment* Kubernetes custom resource has to be defined. This *SeldonDeployment* file is a JSON or YAML file that allows the Seldon user to define a model as an inference graph of Kubernetes PODs. The descriptor defined for the preliminary experimental test of the NSE UC1 Functionality-1 pipeline includes an inference graph with only one node, as follows:

```
apiVersion: machinelearning.seldon.io/v1alpha2
kind: SeldonDeployment
metadata:
  name: locus-encoder
  namespace: models
spec:
  name: locus-encoder
  predictors:
  - graph:
      children: []
      implementation: TENSORFLOW_SERVER
      envSecretRefName: seldon-init-container-secret-minio
      modelUri: s3://locus-bucket/locusencoder
      name:  embeddings-locus
      parameters:
        - name: model_signature
          type: STRING
          value: serving_deafult
        - name: model_name
          type: STRING
          value: embeddings-locus
    name: default
    replicas: 1
```

The *SeldonDeployment* file can then be used to deploy the model through the following command:

```
kubectl apply
```

After this last command, assuming the Istio gateway is available at *istioGateway* and with a Seldon deployment name called *deploymentName* in namespace *<namespace>,* a REST endpoint is exposed at:

```
http://<istioGateway>/seldon/<namespace>/<deploymentName>/api/v1.0/predictions
```

Finally, an HTTP POST request should be forwarded to this endpoint in order to query the model for a prediction. Both the request and the result of a prediction are shown below:

```
$ curl -X POST http://$INGRESS_HOST/seldon/models/locus-
encoder/api/v1.0/predictions     -H 'Content-Type: application/json' -d '{ "data":
{ "ndarray": [[[1,1]]] } }'


RESPONSE:
{
  "data": {
    "names": ["t:0", "t:1", "t:2", "t:3", "t:4", "t:5", "t:6", "t:7"],
    "ndarray": [[[0.0527073741, -0.0368852802, 0.0849082693, -0.0666406527,
0.0645413473, -0.0348753221, 0.049843248, 0.0899739489]]
    ]
  },
  "meta": {}
}
```

As expected, the encoder of the LOCUS pipeline returns an eight-dimensional representation of the trajectory in input, that is a tuple of tuple of couples - in this case the trajectory is just a point.

3.2.2 Next steps

As said, the experimental work reported in this deliverable is a preliminary work in the context of NSE UCs machine learning pipeline virtualization and delivery of localization analytics as a service. Following the incremental approach described above, the plan for the coming months (in both T5.2 and T5.3) is to continue the integration and experimentation with AI/ML frameworks like Seldon and Kubeflow, also considering other NSE UCs functionalities, taking as priority those that will feed the LOCUS proof-of-concepts. As a second step, these virtualized NSE UCs machine learning pipelines will be further integrated with the LOCUS MANO developed in WP4 for full compatibility with the LOCUS platform. In this direction, the overall experimental work on virtualized pipeline deployment is expected to provide insights and relevant inputs for the implementation of service optimization strategies (e.g. in terms of edge/core placement options and resource consumption dimensioning) in the LOCUS MANO.

4 Data Privacy

Location Based Services (LBS) exhibit a great potential for the operators, as they can exploit them to offer new useful services to the customers. However, significant privacy and security concerns are raised by LBS. To this purpose, protecting privacy has become one of the primary concerns in 5G networks, as risks can have high consequences. This section presents the implication of the LBS on the privacy issues and how the data processing needs to be implemented in order to exclude them. We primarily introduce privacy definition and after we present how this issue is addressed by LOCUS platform.

Basically, wireless security can be categorized into two aspects: confidentiality and privacy. **Confidentiality** protects data transmission from passive attacks by limiting the data access to intended users only and preventing the access from or disclosure to unauthorized users. For example, data encryption has been widely used to secure the data confidentiality by preventing unauthorized users from extracting any useful information from the broadcast information.

Privacy prevents controlling and influencing the information related to legitimate users, for example, privacy protects traffic flows from any analysis of an attacker. Privacy in general deals with the protection of personal information which may reveal or may lead to hinting any details of personal information/activities regarding a specific user.

There is a privacy threat whenever an adversary can associate the identity of a user to information that the user considers private [82]. In the case of LBS, this sensitive association can be possibly derived from location-based requests issued to service providers. The identity and the private information of a single user can be derived also from requests issued by a group of users as well as from available background knowledge.

4.1 State of the art

To mitigate these privacy problems in LBS, many Location Privacy Protection Mechanisms (LPPMs) have been proposed in the literature. Their goal is to protect location privacy of users while still allowing them to enjoy geolocated services. Some of them are rather generic and can adapt to a lot of situations while others are very specific to a single use case.

Privacy in 5G must been considered from the beginning, many necessary features must be available built-in in the system. A hybrid cloud-based approach is required where mobile operators are able to store and process high-sensitive data locally and less-sensitive data in public clouds. In this way, operators will have more access and control over data and can decide where to share it. Similarly, service-oriented privacy in 5G will lead to more viable solution for preserving privacy [83].

In release 15, 3GPP defines a security framework, architecture and possible operations for the 5G systems. The architecture introduces of several security entities, such as AUSF, Authentication Credential Repository and Processing function (ARPF) and Security Anchor Function (SAF). Regarding the security on location, different issues are introduced in the Technical Report (TR) 33.814, the scope of this TR is to analyse the security aspects of location service in 5G system and ensure the security solutions are aligned with 3GPPP specifications. The work includes, the study of the security key issues, threats and requirements of location service in 5G system. And the elaboration of the potential security solutions to cover these requirements.

The literature presents several algorithms, schemes, and protocols that will protect as much as possible user information. For location privacy, anonymity-based techniques must be applied where the subscriber's real identity could be hidden and replaced with pseudonyms. Encryption-based practices are useful in this case; for instance, a message can be encrypted before sending to a LBS provider. Techniques such as obfuscation are also crucial, where the quality of location information is reduced in order to protect location privacy. Moreover, location-cloaking-based algorithms are quite useful to handle some major location privacy attacks such as timing and boundary attacks. Considering an obfuscated mobility dataset and a set of user profiles learnt from users' past mobility, a user re-identification attack tries to re-associate a portion of the obfuscated data to its originating user. The aim of these attacks is to break user anonymity by re-associating data obfuscated using a given LPPM with user profiles built from user past mobility. User re-identification attacks aim at linking user obfuscated data to her former mobility data. It is worth mentioning that the terminology de-anonymization can be found in place of re-identification [84]. We would rather use de-anonymization to describe the process of finding a user real identity (e.g., name, address) while re-identification describes the process of recovering a user ID in the system.

To provide privacy preserving functions to the LOCUS platform we consider technics in part exploits in a tool for cyber security sharing that mitigate the risk of privacy diffusion and secondary privacy damage [85] and data query correlation [86].

4.2 Data flow and functions

The main question is how location data will be collected, stored and used without disclosing the private data of individuals. Moreover, key privacy properties can be used in 5G network, as follows. A first approach is to hide the identity of the user since user anonymity would guarantee also the user privacy. Even assuming that we adopt effective anonymization techniques for general information contained in the LBS queries, the spatiotemporal data contained in the queries can sometimes re-identify the user. Thus, anonymization is difficult

to achieve, a possibility to prevent private data leakage is to restrict the amount of private information being released while interacting with the LBS.
In this section we first present the data location flow and therefore the list of the functions that will be implemented in the LOCUS platform.
From a general overview, the main input is represented from raw location data generated from the users, and the output is a database that contains the sanitized and clean location data. The general architecture consists of two main layers: SANITIZATION layer and PRIVACY POLICY layer. The sanitization layer processes raw location data in order to represent them through an anonymization space. The second layer, the privacy policy, confirms that the sanitized trace properties are exposed according with k-anonymity and obfuscation user policy.

Figure 57. Data flow privacy preserving

The general data architecture consists of two main parts. LOCUS client which runs locally on a user's environment and generate location data, a LOCUS privacy module which collects and disseminates service's queries and replies. Figure 57 depicts these components and how they interact. User environment (left side of the Figure 57) run the client program. The client collects the location data via its recorder module. The location data pass through a SANITIZATION module to remove Personally Identifiable Information (PII). Client recorder module also forwards privacy policy to the LOCUS privacy module.
LOCUS privacy module includes a Query Handler module. Whenever LOCUS platform receives location data request, it processes the query according to the query policy of all the involved users and it provides a response. Query request can be produced by external or internal LOCUS platform services and aggregate results are considered.

Table 6 reports the functions used to implement privacy preserving system model to the LOCUS platform.

Table 6. LOCUS Privacy Functions

Function name	Description
Sanitization	This function implements the process of removing user sensitive information from stored location data, so that the data may be distributed to a third-party entity. Sanitization attempts to reduce the data classification level. The goal of this function is also related to the data anonymization.
***k*-anonymity**	This function implements data process to produce that individual in the database is indistinguishable, with respect to the quasi-identifiers, from k - 1 other individuals. There are various methods to achieve k-anonymity: generalization of an attribute (for example postal addresses can be generalized to the street or to the city, depending on how much we need to generalize), suppression of an attribute, or addition of dummy records.
Obfuscation	This function implements techniques aim at blurring or perturbing the location information contained in LOCUS persistence entity because of its potential sensitivity. As an example, the precision of location information can be decreased by translating precise point coordinates to geographic regions; Analogously, the precision of the temporal information usually associated with location can be decreased by converting precise timestamps into time intervals.
Policy definition	In order to manage different level of privacy and security, the LOCUS platform provide the possibility to define user/device policy. This function implements the setting of the policy privacy. It describes a fine-grained sanitization policy developed for their private data, to facilitate their release to public, and a sanitization tool that applies the policy. This policy works in the following way: i) query result restriction specifications that list fields that must have L unique values in the result set. ii) operation restriction specifications that deny access to a list of fields and variables except via the list of utility functions. This function has implication on filtering, aggregation, anonymization, and obfuscation procedures.
Result Aggregation (query)	LOCUS platform protects data location through a secure query framework which only releases aggregate and prevalent results based on work done by [87] and [86]. Thus, the privacy of a user/device is further protected by using a “hiding in a crowd” approach thirty party can query on any features which interest them, but they only receive aggregate responses (counts, histograms, etc.) to address data query correlation. These responses are synthesized from user/device individual responses, after applying their query policies to ensure.

4.2.1 Privacy preserving modules in LOCUS architecture

Based on the requirements just listed and the problems discussed, an architecture for consider the privacy issues is introduced in this section.

The proposed architecture is shown in Figure 58, depicts that basic process for data collection and storage. Location data are recorded and stored in the LOCUS persistent entity locally. The data collection process takes place locally on user environment and considers all the possible sources present in LOCUS. Users can choose the privacy level that will be used for the own location data sharing.

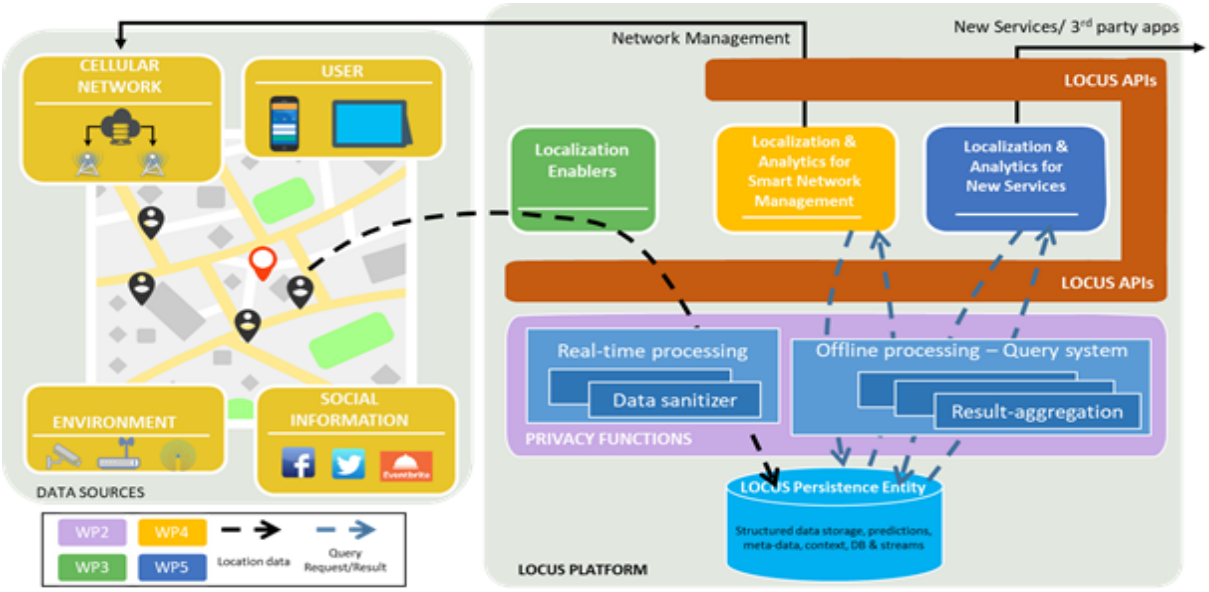

Figure 58. Privacy preserving modules in LOCUS architecture

We distinguish between two classes of LPPMs, as illustrated in Figure 58 (privacy functions box). **Real-time LPPMs** process data during the generation. **Offline LPPMs** process data when a request from network or service are received. Instead of protecting locations on-the-fly, offline LPPMs protect whole mobility datasets at once, possibly leveraging the knowledge of the behaviour of all users in the system to apply more efficient and indirect schemes.

Real-time LPPMs implement anonymization functions, while offline LPPMs implement the query system, it manages the mechanism for queries. Queries are pulled by modules related to the WP4 and WP5. To protect from data query correlation, we aggregate responses from clients through our secure query system. These responses are aggregated on the server before being returned to a service. Since we release only aggregate responses, many active and passive attacks that work against data sets such as sanitized data are ineffective in our context.

An overview of the querying process in three steps:

1. A service submits a query via the LOCUS APIs. The queries are processed by the WP4 modules if the service is related to the network configuration, differently, the queries are processed by the WP5 if the service is related to external information.
2. The service query is sent to query system.
3. Offline LPPMs process the query, if the query is policy permits, it returns results along with information on how users wants its response aggregated.

We take a “hiding in a crowd” approach for privacy protection by enforcing k-anonymity criteria before any result is returned to a service. k-anonymity ensures that the returned result is a set of grouped responses such that each group has a single aggregated value representing at least k different users replies. For example, if k = 5, the result might be: “5 users visited the point of interest around 5 times and 6 users visited the website around 10 times”, but the result would never be: “1 user visited the point of interest 2 times per day” (since k-anonymity would not be met with k = 1). We determine the membership of each group of counts by first ordering values and grouping responses with the same value together. If the number of responses for a value is less than k we either drop this value from the response or merge this group of responses with the adjacent group. We then check this merged group to see if its membership is at least k. If it is not, we repeat the process, otherwise we continue with the next group. For now, the value associated with a merged group is the maximum value in that group, however we plan to investigate other methods of value assignment. For our initial deployment we have fixed k at the low values of k = 2.

5 Conclusion and Next Steps

As a first step, functionalities for each use case in WP5 were derived and defined. For each functionality the inputs, intended outputs, the pre-processing steps involved, and solution design for implementation have been identified and defined in this deliverable. All the WP partners have started further investigations and implementations. The development of each functionality and the initial research carried out are presented. The summary of work progress and activities of each functionality with respect to Task 5.1 are as follows.

NSE-UC1-Functionality-1: Identifying crowd mobility patterns (spatio-temporal) – Location analytics such as possible visitor paths, POIs (indoor/outdoor/hybrid). This functionality addresses the general question of identifying patterns of individual or collective mobility behaviour, in terms of clusters, trends, densities etc in indoor/outdoor/hybrid locations. We identified suitable open datasets for this functionality and completed initial exploratory analysis. We started investigating representative learning approaches for trajectory clustering and predictions using deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Classical clustering techniques such as K-means clustering, and Density-based clustering (DBSCAN/ OPTICS) will then be leveraged in the latent space representation of the data. Further, we will investigate sequence-to-sequence autoencoders, convolutional autoencoders, and LSTM autoencoders for this functionality. We will also aim for the identification of good similarity metrics for trajectory and embedded trajectory data.

NSE-UC1-Functionality-2: In a monitored indoor area, classifying people movement behaviour relative to "geofenced" perimeters according to security, safety, and other objectives. This functionality aims at implementing 'geofencing', which is the construction of a virtual perimeter around a geographic area, and the automatic detection of entry and exit of moving individuals or objects in this area. This functionality will investigate techniques to deploy accurate geofencing based on heterogeneous geolocation data, and environment-aware geofencing where specific landmarks and environment labels can provide assistance to maintain the geofences. This functionality will also investigate trajectory prediction and behaviour forecasting, to enable preventive flow management and geofence protection.

NSE-UC2-Functionality-1: Learning group mobility characteristics using wireless fingerprints. For this functionality, Group-In, a wireless scanning system to detect static or mobile people groups in indoor or outdoor environments has been designed. We have been exploring several open datasets for this use case. We will investigate ML techniques based on clustering algorithms to identify static and dynamic groups.

NSE-UC2-Functionality-2: Using multi-modal data for crowd mobility – COVID-19 as a special case. The functionality targets understanding and automatically extracting “situations” in the real-world that may lead to the spread of viruses including COVID-19. For instance, some environments may lead to more spread, e.g., indoor areas without good air ventilation or setups which cause people to physically interact with each other. This functionality aims to recognize such scenarios using multi-modal data including camera data as well as various IoT sensors such as Bluetooth, Wi-Fi, accelerometer, gyroscope, infrared sensor, and so on. One possibility is to extend the existing NSE-UC2-functionality-1 which already aimed for group monitoring. The functionality would improve sensors other than wireless scanners such as image/video feeds. The plan is to investigate ML techniques and develop a pipeline to create warnings/alerts for the scenarios, settings, or dynamic events that may lead to the virus spread for a given environment.

NSE-UC3-Functionality-1: Vulnerable road user. This use case alerts the host vehicle (HV) of approaching Vulnerable Road User (VRU) in the road. HV approaches the VRU along roads that are defined by their lane designations and geometry. The HV should be able to avoid collision with the VRU. Analytics could provide more detailed insight into the characteristics of the system e.g. tracking Time to Collision parameter for different type of VRU users in different weather conditions. For this functionality, we will investigate ML techniques such classification of moving object trajectories using SVMs, CRFs, decision trees, and classifying spatio-temporal trajectories using CNNs.

NSE-UC3-Functionality-2: Time to collision as a service in V2X. Both the use cases, NSE-UC3 and NSE-UC4 relate to the vehicle mobility and describe scenarios with moving physical objects such as cars and VRUs creating potential road safety risk with their operational information e.g. location, speed and heading. Such operational information will be shared by V2X messages in future Cooperative-ITS systems. The goal is to collect operational moving object information and use them to develop new analytics algorithms and to create novel advanced localization and analytics-based services for road safety applications applicable both to vehicle and VRU safety use cases. One example of such application which leverages vehicle or VRU location, speed and heading information could be ‘Time to collision as a service in V2X’ based on the Time to Collision (TTC) parameter definition in ETSI ITS standard TS 101-539-3 V.1.1.1 (2013-11) which defines the time period before the physical collision of one moving object with another one with a conflicting movement trajectory.

NSE-UC4-Functionality-1: Logistics in a seaport terminal using AGVs. This functionality focuses on deep automation of seaport activities that require the massive introduction of automated transport systems like AGVs (Automated Guided Vehicle), and these in turn require an accurate and real-time localization to implement a high-performance navigation of the

autonomous vehicles. In this UC a mission/navigation system that control the movement of AGVs in a seaport environment will leverage on the accuracy of positioning information provided by the LOCUS platform to achieve high precision real time control of AGV operation. An initial structure of mission/navigational system for AGV has been proposed and is under development with main functional blocks as AGV task planning function, AGV trajectory planning function, AGV motion controller function.
NSE-UC5: Analytics on crowd mobility profiles (e.g. Pedestrian, road traffic, railway routes) and predict the near-future traffic by assigning trajectory profiles per user. The goal of this functionality is to prove the feasibility and exploitability of location information through time for smart city traffic management. Initial work has focused on using unsupervised ML algorithms to group trajectory patterns per user profile. The plan is to investigate deep learning based approaches to assign profile/trajectory patterns.

Due to the on-going COVID-19 pandemic, a new use case NSE-UC6: Positioning and Flow Monitoring for Controlling COVID-19 has been introduced in WP5. This use case focuses on developing efficient tools to enhance the health safety in the restarting phase after COVID-19 pandemic or prevent any future bouncing waves of this pandemic. Furthermore, two functionalities are defined for NSE-UC6.
NSE-UC6-Functionality-1: Contact Tracing. The goal of this functionality is given an identified case of COVID-19 infection, it traces back the persons to have potentially been in proximity with the positive case within a certain number (to be set) of previous hours/days. We will investigate ML techniques for classification of the mobility behaviour in accordance with current quarantine and health protocols. The developed mechanism will trigger a proximity event whenever proximity/contact criteria are detected by using the available geolocation information. The goals is to provide timely updates on current quarantine and health protocols, and trigger a violation event when-ever a quarantine is violated or a mobility behaviour is classified as risky, illegal, etc.
NSE-UC6-Functionality-2: Monitoring epidemiological risk flow. The goal of this functionality is to estimate risk factors and their spatiotemporal evolution using epidemiological data combined with the flows of people moving from one area to another area. We will investigate ML techniques and develop a pipeline to provide a spatiotemporal evolution of epidemiological risk based on data aggregation. The developed system from this functionality will provide predictions for the risk of contagion in different areas.

Task 5.2 and Task 5.3 focus on the development of localization analytics as a service framework to be integrated in the LOCUS platform and have an overlap in the initial stages of

development, mostly related to the definition of a common and unified approach for data virtualization, management and exposure of ML pipelines and analytics functions as services. These tasks rely on the ML pipelines developed for the location-based services in Task 5.1. Hence, both these tasks commenced at a later stage after the initial developments in Task 5.1. The summary of work progress as follows.

- Preparation and consolidation of a data model survey to collect and maintain updated information on expected data formats and types of NSE Use Cases and analytics functions to be managed in the API layer for localization & analytics as a service
- Consolidation of data model survey through interactions with partners, further categorization of expected input/output data and definition of data type categories.
- Investigation and study of intent-based APIs and approaches in the state-of-the-art (EU projects, ETSI ZSM standard, SDN controllers)
- Work on definition of the localization as a service analytics approach and its intent-based exposure through LOCUS APIs
- Investigation of data virtualization and management tools suitable for ML & analytics data store and exchange (teeid, irods, opendatahub)
- Identification of Analytics Functions from various WP5 UCs, to be exposed as location services.
- Initial work on packaging options for the LOCUS ML & analytics functions developed in WP5.

References

[1] “Deliverable from WP2 - D2.1: "Scenarios, Use Cases, Requirements",” 2020. [Online]. Available: LOCUS Project, url: https://www.locus-project.eu/results/deliverables/. [Accessed January 2021].

[2] *Deliverable from WP3 - D3.1: "5G-based localization solutions, preliminary version",* LOCUS Project, url: https://www.locus-project.eu/results/deliverables/, 2020.

[3] *Deliverable from WP2 - D2.4: "System Architecture, Preliminary Version",* LOCUS Project, url: https://www.locus-project.eu/results/deliverables/, 2020.

[4] M. S. Kaiser, K. T. Lwin, M. Mahmud, D. Hajializadeh, T. Chaipimonplin, A. Sarhan and M. A. Hossain, “Advances in crowd analysis for urban applications through urban event detection,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 10, pp. 3092--3112, 2017.

[5] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich and P. Bouvry, “A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities,” *IEEE communications surveys & tutorials,* vol. 21, no. 3, pp. 2419-2465, 2019.

[6] Z. Shi and L. S. Pun-Cheng, “Spatiotemporal data clustering: a survey of methods,” *ISPRS international journal of geo-information,* vol. 8, no. 3, p. 112, 2019.

[7] C. Yu, X. Ma, J. Ren, H. Zhao and S. Yi, “Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction,” *arXiv preprint arXiv:2005.08514,* 2020.

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 961-971, 2016.

[9] S. Deng, S. Jia and J. Chen, “Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data,” *Applied Soft Computing,* vol. 78, pp. 712-721, 2019.

[10] S. Wang, J. Cao and P. Yu, “Deep learning for spatio-temporal data mining: A survey,” *IEEE Transactions on Knowledge and Data Engineering,* no. IEEE, 2020.

[11] J. Newling and F. Fleuret, “K-medoids for k-means seeding,” *Advances in neural information processing systems,* vol. 30, no. 5195-5203, 2017.

[12] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis and X. Qin, “Fast large-scale trajectory clustering,” *Proceedings of the VLDB Endowment,* vol. 13, pp. 29-42, 2019.

[13] D. Birant and A. Kut, “ST-DBSCAN: An algorithm for clustering spatial-temporal data,” *Data & knowledge engineering,* vol. 60, no. 1, pp. 208-221, 2007.

[14] Y. Wang, M. Long, J. Wang, Z. Gao and P. S. Yu, “Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms,” *Advances in Neural Information Processing Systems,* vol. 30, pp. 879-888, 2017.

[15] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong and F. Zhang, “Trajectory-User Linking via Variational AutoEncoder,” *IJCAI,* vol. 1, pp. 3212-3218, 2018.

[16] G. Solmaz and D. Turgut, “A survey of human mobility models,” *IEEE Access,* vol. 7, pp. 125711-125731, 2019.

[17] G. Solmaz, J. Furst, S. Aytac and F.-J. Wu, "Group-In: Group Inference from Wireless Traces of Mobile Devices," *19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 157-168, 2020.

[18] W. Ge, R. T. Collins and R. B. Ruback, "Vision-based analysis of small groups in pedestrian crowds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, pp. 1003-1016, 2012.

[19] F. Solera, S. Calderara and R. Cucchiara, "Socially constrained structural learning for groups detection in crowd," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 995-1008, 2015.

[20] F. Adib, Z. Kabelac and D. Katabi, "Multi-person localization via RF body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation NSDI 15*, 2015.

[21] X. Guo, B. Liu, C. Shi, H. Liu, Y. Chen and M. C. Chuah, "WiFi-enabled smart human dynamics monitoring," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017.

[22] V. Kostakos, E. O'Neill, A. Penn, G. Roussos and D. Papadongonas, "Brief encounters: Sensing, modeling and visualizing urban mobility and copresence networks," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 17, no. 1, pp. 1-38, 2010.

[23] J. E. Larsen, P. Sapiezynski, A. Stopczynski, M. Morup and R. Theodorsen, "Crowds, bluetooth, and rock'n'roll: understanding music festival participant behavior," in *Proceedings of the 1st ACM international workshop on Personal data meets distributed multimedia*, 2013.

[24] K. Jayarajah, Y. Lee, A. Misra and R. K. Balan, "Need accurate user behaviour? pay attention to groups!," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, 2015.

[25] R. Sen, Y. Lee, K. Jayarajah, A. Misra and R. K. Balan, "Grumon: Fast and accurate group monitoring for heterogeneous urban spaces," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014.

[26] P. Sewalkar and J. Seitz, "Vehicle-to-pedestrian communication for vulnerable road users: Survey, design considerations, and challenges," *Sensors*, vol. 19, no. 2, p. 358, 2019.

[27] *ETSI TR 103 298 V0.0.1 (2016-06) Intelligent Tranport Systems (ITS); Platooning; Pre-standardisation study*, 2016.

[28] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic and P. Jaillet, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 794-804, 2013.

[29] X. Ma, Y. Li and P. Chen, "Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm," *Journal of Traffic and Transportation Engineering (English Edition)*, 2018.

[30] S. Heldens, N. Litvak and M. Van Steen, "Scalable detection of crowd motion patterns," *IEEE transactions on knowledge and data engineering*, no. IEEE, 2018.

[31] X. Ma, H. Yu, Y. Wang and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PloS one*, vol. 10, no. 3, pp. 19-44, 2015.

[32] S. H. Park, B. Kim, C. M. Kang, C. C. Chung and J. W. Choi, "Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture.," *IEEE Intelligent Vehicles Symposium (IV)*, no. IEEE, pp. 1672-1678, 2018.

[33] European Commission, "Digital solutions during the pandemic," 2020. [Online]. Available:] https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/digital-solutions-during-pandemic_en.

[34] "Rapid Action on Corona Virus and EO," 2020. [Online]. Available: https://race.esa.int. [Accessed December 2020].

[35] ""WorldView-3 Satellite Sensor," Satellite Imaging Corporation," 2020. [Online]. Available: http://www.satimagingcorp.com/satellite-sensors/worldview-3.html. [Accessed December 2020].

[36] R. Minetto, M. P. Segundo, G. Rotich and S. Sarkar, "Measuring Human and Economic Activity from Satellite Imagery to Support City-Scale Decision-Making during COVID-19 Pandemic," *arXiv preprint arXiv:2004.07438*, 2020.

[37] "Open Street Map," [Online]. Available: https://www.openstreetmap.org. [Accessed December 2020].

[38] S. Jacob and J. Lawaree, "The adoption of contact tracing applications of COVID-19 by European governments," *Policy Design and Practice*, pp. 1-15, 2020.

[39] H. Alsdurf, Y. Bengio, T. Deleu, P. Gupta, D. Ippolito, R. Janda, M. Jarvie, T. Kolody, S. Krastev, T. Maharaj and others, "COVI White Paper," *arXiv preprint arXiv:2005.08502*, 2020.

[40] W. Beskorovajnov, F. Dorre, G. Hartung, A. Koch, J. Muller-Quade and T. Strufe, "ConTra Corona: Contact Tracing against the Coronavirus by Bridging the Centralized-Decentralized Divide for Stronger Privacy.," *IACR Cryptol. ePrint Arch.*, p. 505, 2020.

[41] Y. Park, Y. Choe, O. Park, S. Park, Y. Kim, J. Kim, S. Kweon, Y. Woo, J. Gwack and S. a. o. Kim, "COVID-19 National Emergency Response Center, Epidemiology and Case Management Team. Contact tracing during coronavirus disease outbreak, South Korea," *Emerg Infect Dis*, vol. 26, no. 10, pp. 2465-2468, 2020.

[42] Y. a. Z. J. Yang, "Coronavirus brings China's surveillance state out of the shadows," *Reuters News. Reuters*, 2020.

[43] C. Castelluccia, N. Bielova, A. Boutet, M. Cunche, C. Lauradoux, D. Le Metayer and V. Roca, "ROBERT: ROBust and privacy-presERving proximity Tracing," 2020.

[44] C. Troncoso, M. Payer, J.-P. Hubaux, M. Salathe, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, D. Antonioli and others, "Decentralized privacy-preserving proximity tracing," *arXiv preprint arXiv:2005.12273*, 2020.

[45] "Exposure Notification, Apple Developer,," 2020. [Online]. Available: https://developer.apple.com/documentation/exposurenotification. [Accessed December 2020].

[46] "Exposure Notifications: Using technology to help public health authorities fight COVID-19," 2020. [Online]. Available: https://www.google.com/covid19/exposurenotifications/. [Accessed December 2020].

[47] “Contact Tracing - Bluetooth Specification,” 2020. [Online]. Available: https://blog.google/documents/58/Contact_Tracing_-_Bluetooth_Specification_v1.1_RYGZbKW.pdf. [Accessed December 2020].

[48] N. Martinez-Martin, S. Wieten, D. Magnus and M. K. Cho, “Digital contact tracing, privacy, and public health,” *Hastings Center Report,* vol. 50, no. 3, pp. 43-46, 2020.

[49] T. G. O'neill and B. P. Williams, *Learning geofence models directly,* Google Patents, US Patent 9,349,104, 2016.

[50] A. Garg, S. Choudhary and P. Bajaj, “Smart Geo-fencing with Location Sensitive Product Affinity,” in *Proceedings of SIGSPATIAL’17, ,* Los Angeles, CA, USA, November 2017.

[51] M. L.D. Dias, C. Lincoln C. Matoos, T. L.C. Da Silva, J. Antonia F. de Macedo and W. C.P.Silva, “Anomaly Detection in Trajectory Data with Normalizing Flows,” in *Proceedings of IJCNN*, Glasgow, UK, 2020.

[52] C. Du Mouza and P. Rigaux, “Multi-scale Classification of Moving Objects Trajectories,” in *SSDBM*, 2004.

[53] P. Wang, Z. Li, Y. Hou and W. Li, “Action recognition based on joint trajectory maps using convolutional neural networks,” in *Proceedings of the 24th ACM international conference on Multimedia,* 2016.

[54] R. Chandra, T. Guan, S. Panuganti, T. a. B. U. Mittal, A. Bera and D. Manocha, “Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms,” *IEEE Robotics and Automation Letters,* vol. 5, no. 3, pp. 4882-4890, 2020.

[55] H.-S. Roh, C. S Lalwani and M. M. Naim, “Modelling a port logistics process using the structured analysis and design technique,” *International Journal of Logistics Research and Applications,* vol. 10, no. 3, pp. 283-302, 2007.

[56] L. Heilig and S. Voss, “Information systems in seaports: a categorization and overview,” *Information Technology and Management,* vol. 18, no. 3, pp. 179-201, 2017.

[57] I. Harmati, G. Orban and P. Varlaki, “Takagi-Sugeno fuzzy control models for large scale logistics systems,” in *2007 International Symposium on Computational Intelligence and Intelligent Informatics*, 2007.

[58] H. Trevor, R. Tibshirani and J. Friedman, *Hierarchical clustering. The Elements of Statistical Learning,* New York, Springer, 2009.

[59] M. Ester, H.-P. Kriegel, J. Sander, X. Xu and others, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, 1996.

[60] “Centers for Disease Control and Prevention (CDC),” 1 July 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/about-epidemiology/monitoring-and-tracking.html. [Accessed December 2020].

[61] C. Zhou, W. Yuan, J. Wang, H. Xu, Y. Jiang, X. Wang and Q. H. a. Z. P. Wen, “Detecting Suspected Epidemic Cases Using Trajectory Big Data,” *arXiv preprint arXiv:2004.00908,* 2020.

[62] “Docker,” [Online]. Available: https://www.docker.com/. [Accessed 2020].

[63] “Kubernetes,” [Online]. Available: https://kubernetes.io/. [Accessed 2020].

[64] “Openstack,” [Online]. Available: www.openstack.org. [Accessed 2020].

[65] “Qemu,” [Online]. Available: https://www.qemu.org/. [Accessed 2020].

[66] “Ubuntu Cloud Image,” [Online]. Available: https://cloud-images.ubuntu.com/. [Accessed 2020].

[67] “Ansible,” [Online]. Available: https://www.ansible.com/. [Accessed 2020].

[68] “Deliverable from WP5 - D5.3: "Design of the localization & analytics as a service solution",” 2020. [Online]. Available: Locus Project, url: https://www.locus-project.eu/results/deliverables/.

[69] “Deliverable from WP4 - D4.1: "Implementation of the Virtualization platform for network control and management, preliminary version",” 2020. [Online]. Available: Locus Project, url: https://www.locus-project.eu/results/deliverables/.

[70] T. Kashiyama, Y. Pang and Y. Sekimoto, “Open PFLOW: Creation and evaluation of an open dataset for typical people mass movement in urban areas,” *Transportation research part C: emerging technologies*, vol. 85, pp. 249-267, 2017.

[71] L. a. T. N. Vincent, “Shape and time distortion loss for training deep time series forecasting models,” in *Advances in neural information processing systems*, 2019, pp. 4189--4201.

[72] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, “OPTICS: ordering points to identify the clustering structure,” *ACM Sigmod record*, vol. 28, no. 2, pp. 49-60, 1999.

[73] R. C. De Amorim and C. Hennig, “Recovering the number of clusters in data sets with noise features using feature rescaling factors,” *Information Sciences*, vol. 324, pp. 126-145, 2015.

[74] S. Har-Peled and other, “New similarity measures between polylines with applications to morphing and polygon sweeping},” *Discrete & Computational Geometry*, pp. 535-569, 2002.

[75] “SeldonIO,” [Online]. Available: https://github.com/SeldonIO. [Accessed 2020].

[76] “KubeFlow,” [Online]. Available: https://www.kubeflow.org/docs. [Accessed 2020].

[77] “TensorFlow,” [Online]. Available: https://www.tensorflow.org/tfx/guide/serving. [Accessed 2020].

[78] “Argo work flow,” [Online]. Available: https://argoproj.github.io/argo/. [Accessed 2020].

[79] “TensorflowJob,” [Online]. Available: https://www.kubeflow.org/docs/components/training/tftraining/. [Accessed 2020].

[80] “Istio,” [Online]. Available: https://istio.io/. [Accessed 2020].

[81] “Minio,” [Online]. Available: https://min.io/. [Accessed 2020].

[82] C. Bettini, S. Mascetti, D. Freni, X. Wang and S. Jajodia, “Privacy and Anonymity in Location Data Management,” in *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*, 2010, pp. n. 10.1201/b10373-13.

[83] “Huawei 5G Security: Forward Thinking,” 2016. [Online]. Available: http://www.huawei.com/minisite/5g/img/5G-SecurityWhitepaperen.pdf. [Accessed 2020].

[84] S. Gambs, M.-O. Killijian and M. N. del Prado Cortez, “De-anonymization attack on geolocated data,” *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1597-1614, 2014.

[85] B. Krishnamurthy and C. Wills, “Privacy diffusion on the web: a longitudinal perspective,” in *Proceedings of the 18th international conference on World wide web*, 2009.

[86] V. Sharma, G. Bartlett and J. Mirkovic, “Critter: Content-rich traffic trace repository,” in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, 2014.

[87] J. Mirkovic, “Privacy-safe network trace sharing via secure queries,” in *Proceedings of the 1st ACM workshop on Network data anonymization*, 2008.