



# A Cloud-native Approach for Processing of Crowdsourced GNSS Observations and Machine Learning at Scale: A Case Study from the CAMALIOT Project

Grzegorz Kłopotek<sup>a,\*</sup>, Yuanxin Pan<sup>a</sup>, Tobias Sturn<sup>b</sup>, Rudi Weinacker<sup>b</sup>, Linda See<sup>b</sup>,  
 Laura Crocetti<sup>a</sup>, Mudathir Awadaljeed<sup>a</sup>, Markus Rothacher<sup>a</sup>, Ian McCallum<sup>b</sup>,  
 Steffen Fritz<sup>b</sup>, Vicente Navarro<sup>c</sup>, Benedikt Soja<sup>a</sup>

<sup>a</sup>*Institute of Geodesy and Photogrammetry, ETH Zürich, Zürich, Switzerland*

<sup>b</sup>*International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria*

<sup>c</sup>*European Space Astronomy Centre (ESAC), European Space Agency (ESA), Madrid, Spain*

Received 15 January 2024; received in revised form 19 February 2024; accepted 27 February 2024

## Abstract

The era of modern smartphones, running on Android version 7.0 and higher, facilitates nowadays acquisition of raw dual-frequency multi-constellation GNSS observations. This paves the way for GNSS community data to be potentially exploited for precise positioning, GNSS reflectometry or geoscience applications at large. The continuously expanding global GNSS infrastructure along with the enormous volume of prospective GNSS community data bring, however, major challenges related to data acquisition, its storage, and subsequent processing for deriving various parameters of interest. In addition, such large datasets cannot be managed manually anymore, leading thus to the need for fully automated and sophisticated data processing pipelines. Application of Machine Learning Technology for GNSS IoT data fusion (CAMALIOT) was an ESA NAVISP Element 1 project (NAVISP-EL1-038.2) with activities aiming to address the aforementioned points related to GNSS community data and their exploitation for scientific applications with the use of Machine Learning (ML). This contribution provides an overview of the CAMALIOT project with information on the designed and implemented cloud-native software for GNSS processing and ML at scale, developed Android application for retrieving GNSS observations from the modern generation of smartphones through dedicated crowdsourcing campaigns, related data ingestion and processing, and GNSS analysis concerning both conventional and smartphone GNSS observations. With the use of the developed GNSS engine employing an Extended Kalman Filter, example processing results related to the Zenith Total Delay (ZTD) and Slant Total Electron Content (STEC) are provided based on the analysis of observations collected with geodetic-grade GNSS receivers and from local measurement sessions involving Xiaomi Mi 8 that collected GNSS observations using the developed Android application. For smartphone observations, ZTD is derived in a differential manner based on a single-frequency double-difference approach employing GPS and Galileo observations, whereas satellite-specific STEC time series are obtained through carrier-to-code leveling based on the geometry-free linear combination of observations from both GPS and Galileo constellations. Although the ZTD and STEC time series from smartphones were derived on a demonstration basis, a rather good level of consistency of such estimates with respect to the reference time series was found. For the considered periods, the RMS of differences between the derived smartphone-based time series of differential zenith wet delay and reference values were below 3.1 mm. In terms of satellite-specific STEC time series expressed with respect to the reference STEC time series, RMS of the offset-reduced differences below 1.2 TECU was found. Smartphone-based observations require

\* Corresponding author.

E-mail address: [grzegorz@klopotek.org](mailto:grzegorz@klopotek.org) (G. Kłopotek).

special attention including additional processing steps and a dedicated parameterization in order to be able to acquire reliable atmospheric estimates. Although with lower measurement quality compared to traditional sources of GNSS data, an augmentation of ground-based networks of fixed high-end GNSS receivers with GNSS-capable smartphones would however, form an interesting source of complementary information for various studies relying on GNSS observations.

© 2024 COSPAR. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

*Keywords:* GNSS; Troposphere; Ionosphere; Smartphones; Machine learning

## 1. Introduction

The Global Navigation Satellite System (GNSS) has revolutionized the area of positioning, navigation and timing. GNSS is also a key tool in many areas of science, such as the quantification of global-scale geodynamical phenomena (Hammond et al., 2016; Beutler et al., 2020), or realization of the International Terrestrial Reference System (Altamimi et al., 2023). Due to its all-weather capability and the high spatio-temporal resolution of the estimates that this technique can provide, GNSS is also considered to be an invaluable tool for geophysical research and monitoring of complex large-scale phenomena occurring within the Earth's system, such as those connected to the ionosphere or troposphere. The latter two domains are being thoroughly investigated with the use of GNSS observations as a large number of observations collected from various parts of the globe as well as the sensitivity of the GNSS technique to changes in the ionospheric and tropospheric states allow the phenomena driving changes in these domains to be quantified (Flores et al., 2000; Takahashi et al., 2016; Yu et al., 2017; Bosser et al., 2021). This is reflected in the observed variations in total electron content (TEC) or water vapor content, where the latter is deduced based upon information contained in the time series of zenith wet delay (ZWD) and tropospheric gradients. Usually, the latter two are set up as solve-for parameters in the GNSS data analysis, but this can be also extended to vertical TEC (VTEC) when considering the raw (uncombined) observation approach (Strasser et al., 2019).

Precipitable water vapor (PWV) overlying the GNSS receiver can be determined from the derived time series of Zenith Total Delay (ZTD) if certain constants related to the refractivity of moist air and of the weighted mean temperature of the atmosphere are known (Bevis et al., 1992; Bevis et al., 1994; Ning et al., 2013). Thanks to continuously operational GNSS, long-term PWV trends can be acquired globally, at a high frequency, allowing changes in the hydrological cycle, atmospheric radiation, and climate to be studied. The use of GNSS-derived PWV for atmospheric studies is now a well-established field of research and the GNSS-derived PWV estimates are in good agreement with the conventional instrumentation (radiosondes or microwave radiometers) that is used for this purpose (Van Malderen et al., 2014). Atmospheric water vapor estimates from static ground-based GNSS receivers

are nowadays operationally assimilated into numerical weather prediction models (NWP), either in the form of PWV or ZTD (Uppala et al., 2005; Bennitt and Jupp, 2012; Guerova et al., 2016).

GNSS can also provide a very precise estimation of the slant total electron content (STEC), which is the linear integral of the electron density along any satellite-receiver ray path (Davies and Hartmann, 1997), with units of electrons per square meter, where  $10^{16}$  electrons/m<sup>2</sup> corresponds to 1 TEC unit (TECU). The satellite-receiver-specific ionosphere TEC at the line of sight is compressed at a point referred to as the ionospheric pierce point (IPP), and mapped into the vertical direction with the use of the ionosphere mapping function. Commonly one utilizes dual-frequency GNSS signals and their geometry-free linear combination to acquire STEC (Ciraolo et al., 2007). When satellite-specific VTEC is obtained from many globally distributed GNSS receivers, such as with those comprising the International GNSS Service (IGS) network, they can be used to generate global ionosphere maps (GIM, Hernández-Pajares et al., 2008), which represent snapshots of global VTEC that find many applications in single-frequency positioning or studies of the ionosphere. In an alternative approach, commonly referred to as uncombined precise point positioning (UC-PPP), one leverages the uncombined observations recorded at different frequencies and utilizes them directly in the analysis in order to derive ZTD and VTEC (Zhang et al., 2012; Wang et al., 2020). Single-frequency PPP (SF-PPP) can in principle be employed for the same purpose (Zhang et al., 2018; Zhao et al., 2019).

Over the past years, GNSS has been developing significantly and rapidly, with the dawn of the fully operational Galileo or BeiDou Navigation Satellite System as well as continuous developments of the space segments of the well-established GPS (Global Positioning System) and GLONASS systems. Besides the higher level of interoperability and consistency between satellite systems present nowadays, the most important advance in terms of GNSS technology has been the introduction of modern signals of improved quality as disseminated in the L5/E5 band. GNSS infrastructure has also been expanding significantly and rapidly in recent years, not only in space but also on the ground. Nowadays, the group of devices with GNSS-capable receivers also includes a wide range of mass-market sensors that are utilized as static or kinematic sen-

sors for multiple applications ranging from self-driving cars to unmanned aerial vehicles, deformation monitoring (Poluzzi et al., 2020), or land surveying (Odolinski and Teunissen, 2019). Besides providing coordinates of a satisfactory quality, such a network of low-cost dual-frequency GNSS can also be used for atmospheric sounding with the quality of the troposphere products having comparable accuracy to the same estimates acquired with the use of the geodetic equipment (Stępnik and Paziewski, 2022). When treating ionospheric contributions as an estimable parameter, single-frequency multi-GNSS low-cost receivers can be also applied for retrieval of VTEC and PWV with the use of the PPP approach (Zhang et al., 2018; Zhao et al., 2019). Mass-market receivers, offering an acceptable trade-off between performance and affordability, can already be considered a cost-effective solution for the improvement of the spatial resolution of GNSS observations that are collected with the use of the conventional permanent GNSS networks. Such a tendency will also not likely change in the future.

The era of modern smartphones, running on Android version 7.0 and higher, facilitates nowadays the acquisition of multi-constellation GNSS observations from such devices through the related application programming interface (API). This paves the way for the GNSS community data to be potentially exploited for mass-market precise positioning, GNSS reflectometry, or in various geoscience applications, as smartphone use brings low weight, low power consumption, and portability benefits. Smartphones have already been evaluated in terms of their capacity to determine precise coordinates based on GNSS observations (Darugna et al., 2020; Paziewski et al., 2021). Recent studies with such devices also include their utilization in VTEC determination through the carrier-to-code leveling (CCL) approach (Xu et al., 2022). However, currently there are a few technical aspects that limit the accuracy of smartphone-based positioning (Paziewski, 2020) or the quality of the solve-for parameters set up during the GNSS analysis. As smartphones are characterized by an increased sensitivity to the reflected GNSS signals, multipath is the main error source hindering high-precision GNSS-based positioning with such devices. In addition, smartphones are characterized by diverse performance in terms of GNSS signal tracking, which results in model-dependent performance (Paziewski et al., 2021). An additional factor in this regard is also the orientation of the smartphone while taking the measurements, where an upward (vertical) orientation tends to be the most beneficial in terms of the quantity and quality of the GNSS observations that can be available for the subsequent analysis (Yong et al., 2021; Li et al., 2022a). Nevertheless, the continuous progress in the area of GNSS and smartphones may result in the near future in a performance that is comparable to the GNSS receivers employed for atmospheric sensing.

Although with lower measurement quality compared to traditional sources of GNSS data, an expansion of ground-

based networks of fixed high-end GNSS receivers with GNSS-capable smartphones would form an interesting source of complementary data for various studies relying on GNSS observations. The latter are potentially available at large scale from affordable smart devices and hundreds of low-cost single-frequency/dual-frequency GNSS receivers, yet such data is currently far from being fully exploited for science and society. Access to such observations is also currently limited, the means for their collection is missing, and the data processing itself is rather challenging. The continuously expanding global GNSS infrastructure along with the enormous volume of prospective GNSS community data bring therefore major challenges related to data acquisition and storage as well as subsequent processing in conjunction with conventional GNSS observations for deriving various parameters of interest such as those related to precise positioning, the troposphere, or ionosphere. Such large data sets cannot be managed manually anymore, leading to the need for fully automated and sophisticated data processing pipelines. The same applies to the analysis and fusion of huge amounts of heterogeneous auxiliary data and models, requiring a dedicated approach in order to exploit this type of data in a thorough manner and fully benefit from such a concept.

Machine Learning (ML) in high-performance computing environments tends to be an appropriate solution in terms of data fusion, classification or forecasting tasks. The additional benefit is that the large volume of heterogeneous data can be exploited and introduced into the modeling related to both spatial interpolation and forecasting. ML or Deep Learning (DL) facilitate incorporation of information from different and rather complex domains (e.g. solar data), which tends to be a challenging task as a direct physical relation is usually difficult to formulate mathematically. ML and DL can also facilitate overcoming systematic biases and different spatio-temporal resolution of datasets (Zhang and Yao, 2021). In terms of GNSS-based ionospheric observations, ensemble methods and DL architectures have been utilized for single-point, regional, and global TEC modeling and forecasting (Cesaroni et al., 2020; Kaselimi et al., 2020). Similar remarks apply to PWV or Zenith Wet Delays, as derived from various techniques, which can be exploited by means of ML/DL and in conjunction with parameters from the meteorological domain (Benevides et al., 2019). In this context, interpolation carried out with ML can be considered as an alternative to different approaches that have been suggested such as iterative tropospheric decomposition (Yu et al., 2017), a seasonal Gaussian function (Hu and Yao, 2019), or least-squares collocation (Wilgan and Geiger, 2018).

The ESA NAVISP Element 1 project (NAVISP-EL1-038.2) entitled *Application of Machine Learning Technology for GNSS IoT Data Fusion* (CAMALIOT) focused on a set of activities covering the aforementioned topics with a

proof-of-concept cloud-native software and community data collection campaign in order to investigate the potential and increase the usability of IoT (Internet of Things) GNSS data for scientific purposes. The quantifiable goals of this project concerned the integration and utilization of heterogeneous GNSS datasets for modeling and prediction of parameters describing the change in the ionospheric and tropospheric states with the use of the chosen ML algorithms and the proposed technology stack. The proof-of-concept cloud-native software (SW), hereafter referred to as the CAMALIOT SW, encompassed acquisition, ingestion, and pre-processing of smartphone-based GNSS observations, the development of a crowdsourced GNSS repository, GNSS code and carrier-phase processing of the collected assets, and the inclusion of the components related to ML, all to be seamlessly integrated into the existing infrastructure of the GNSS Science Support Centre (GSSC) of ESA. The main goal of the GSSC is to integrate various sources of information from all GNSS domains into a single platform while providing comprehensive processing assets to deliver advanced analysis services concerning continuously increasing GNSS data sets and products (Navarro et al., 2021).

This contribution focuses on providing an overview of the proof-of-concept architecture designed and implemented in relation to the CAMALIOT project with information on the SW functional architecture, collection and processing of community GNSS data at scale, and related GNSS analysis approaches that were established in order to process observations collected by both geodetic-grade static receivers and smartphones. In Section 2, the CAMALIOT SW is introduced with the description of its sub-components and information on ML and GNSS processing at scale as exploited in relation to the project goals. Given the highlighted approach and developed software, example processing results related to ZTD and STEC based on the analysis of observations collected by permanent GNSS stations and from local (controlled) measurement sessions involving Xiaomi Mi 8 are presented in Section 3 in order to showcase the feasibility of our strategy. The general discussion concerning the obtained results is given in Section 4. The summary and outlook given in Section 5 conclude this contribution.

## 2. Methods

GNSS processing and application of ML for a robust, scalable and automated collection, processing, and combination of the aforementioned data had to be properly accommodated in the proposed software architecture, with the technology stack that would be sufficient to address multiple challenges related to this pilot project. In addition, the aim was to secure GNSS data from the modern generation of smartphones, exploit their usefulness for this project, and utilize it in conjunction with observations from high-grade GNSS stations through ML/DL. The CAMALIOT SW had also to be compliant with the current architecture of

the GSSC, which is a cloud-native system built upon the concept of micro-services and utilizing Docker and Kubernetes technologies. Due to the limited scope of the study, this section concentrates on the high-level overview of the developed SW and its features as well as providing information on the GNSS data analysis approach applied to meet the goals of the activity. The detailed information concerning ML in relation to GNSS-based atmospheric estimates or the implementation details of the developed Android application are not discussed here.

### 2.1. CAMALIOT Software Architecture

The proposed software has been designed to run on a Kubernetes cluster that can be understood as a set of physical or virtual machines that run containerized applications. Kubernetes is the new operating system for the cloud and it has grown to become a powerful and flexible tool that can be run on a variety of cloud platforms and on-premises. On Kubernetes, all programs (developed code) run in containers so that they can be isolated from each other and can be easily developed, configured, deployed, monitored or removed without affecting any of the other containers comprising an application. Kubernetes addresses many of the manual processes associated with the deployment and scaling of the containerized applications as well as issues including high availability (HA) and load balancing. Kubernetes can therefore be understood as a container orchestration tool that provides an additional level of coordination for combining individual containers (software components, separate applications) into a cohesive entity.

This proof-of-concept cloud-native application has been designed and implemented as a set of services that carry out specific tasks and are connected via the hypertext transfer protocol (HTTP). The CAMALIOT SW can be perceived as a self-contained modular system that is designed to interact with numerous components of the GSSC through its external interfaces, both realized with the use of the implemented Representational State Transfer (REST) API (also known as RESTful API). The functional architecture of the CAMALIOT SW is depicted in Fig. 1. The proposed architectural pattern allows services to be developed and deployed independently depending upon current needs. The service-based architecture (or a microservice-based architecture) also allows the individual services to be scaled horizontally (increasing the number of instances of the same service), which provides resource and cost optimization benefits. A common practice is also deployment of multiple service replicas in order to ensure high-availability and fault-tolerance of the cloud-native applications. In such a case, a single SW component running on Kubernetes contains in most of the cases at least one deployment and service definitions. The former is used to establish the components (applications) on the cluster, whereas the service defines a logical set of component replicas (pods) and a policy by which to access them. Then mul-

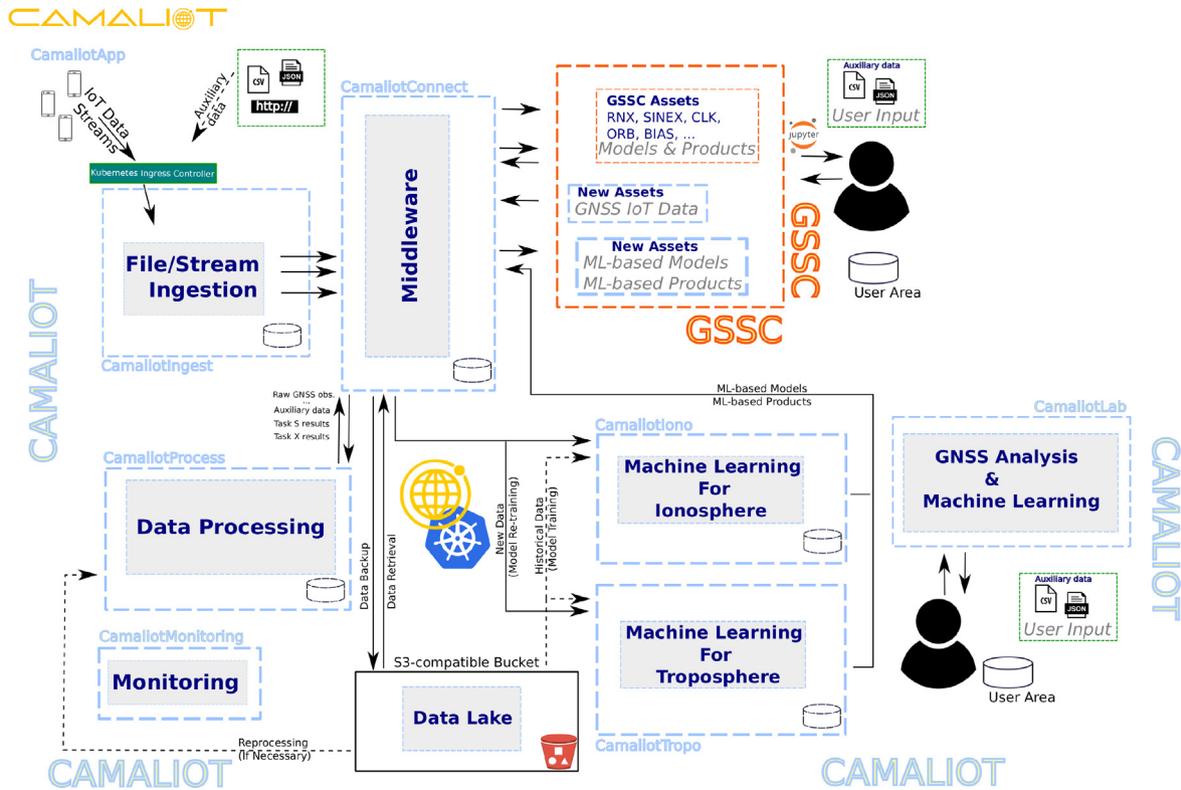


Fig. 1. SW functional architecture designed and implemented for CAMALIOT. The names of the software components are given in light blue.

tuple (three or more) pods are always created. In case one node of the Kubernetes cluster fails, the data processing is not disturbed as the unaffected replicas of the applications are still available to receive the incoming files or messages.

The incoming data within the CAMALIOT SW undergoes many different steps including validation, windowing, aggregation and conversion of the crowdsourced GNSS observations to hourly RINEX files, quality control, and GNSS processing for ionosphere and troposphere. The processing pipeline is complemented with ML-related tasks (pre-processing, model training, model validation, model prediction). Therefore, many open-source SW packages/libraries constitute the CAMALIOT SW, with some of the parts covered by the code that was developed for the purposes of this activity. Among the chosen SW packages, this included Apache Kafka, Apache Spark 3, FastAPI, TensorFlow2 (TF2), Open MPI, scikit-learn, joblibspark, Horovod, JupyterLab, Prometheus, and Grafana. The utilized programming languages included C/C++, Python, and Java. The continuous integration (CI) and continuous deployment (CD) were arranged with the use of a Jenkins server and a private image repository. Both components were hosted externally, but were accessible from the target computing infrastructure (Kubernetes). The latter was established along with the object storage on the resources provided by Exoscale<sup>1</sup>.

Containerized applications are immutable implying that all data created during their lifetime is lost if they crash or are deleted. This might be convenient for some applications, but in most of the cases the containerized services need to preserve some information or share information with other applications. For the CAMALIOT SW, two types of storage were used, i.e., a persistent storage and object storage. Concerning the former, Kubernetes persistent volume (PV) is a storage resource that is established within the Kubernetes cluster and employed to retain data for long periods of time. PV allows a storage unit to be mounted to a Kubernetes node and also share information between nodes. The object storage systems allow, on the other hand, retention of massive amounts of unstructured static data, also commonly referred to as data lakes. Object storage is an important element in relation to cloud applications as it offers optimization of resources, reduction in costs, essentially infinite scalability, or faster file retrieval compared to traditional solutions. In this activity, Longhorn<sup>2</sup>, a cloud-native distributed block storage for Kubernetes, was used for creation of PVs that were utilized by the components of the CAMALIOT SW for persistence of intermediate or temporal files that needed to be shared across pods. In terms of fault-tolerance, persistent volumes are usually replicated (replication factor of three) and backed up to ensure that no data is lost in case one of the nodes of the Kubernetes cluster fails. The crowd-

<sup>1</sup> [www.exoscale.com](http://www.exoscale.com)

<sup>2</sup> <https://longhorn.io>

sourced community data, collected assets as well as the assets produced within the CAMALIOT SW were uploaded to the dedicated bucket (object storage) for backup and processing/re-processing purposes.

## 2.2. High-level Overview of the Software Components

In order to meet the requirements of the CAMALIOT project in relation to securing smartphone GNSS community data through dedicated crowdsourcing campaigns (See et al., 2023), an Android application<sup>3</sup> was designed, developed, published, and assessed throughout the course of the project. The Android application was created by consortium members from the International Institute for Applied Systems Analysis (IIASA). The target audience in this case were researchers working in the area of GNSS, who might be interested in using the data for research purposes, and in general volunteers motivated to contribute to a citizen science project. In order to encourage participation from many people with various backgrounds and experience with GNSS, the application was intentionally designed to be simple to use. An example of the pages from the user interface (UI) of the CAMALIOT Android application is shown in Fig. 2.

During the course of the two consecutive crowdsourcing campaigns, spanning from March to November 2022, the registered users were asked to collect GNSS observations and then upload them to the server, i.e., the CAMALIOT SW. This was not mandatory as with the use of the list of logged sessions the user could decide which of the generated files can be removed from the phone, uploaded to the server, or converted to RINEX-3 files for their own use. The application can also work in a continuous mode, where there is no need for the user to interact with the application as the file upload is carried out in an automatic fashion and based on a pre-defined observation window, as visible in Fig. 2. This implies that instead of submitting a single file with observations per logging session, which can last a few to several hours, the files can be split automatically into hourly files and then submitted (also automatically) to the server.

The primary output of the developed Android Application, hereafter referred to as the *CamaliotApp*, are compressed text files with the format similar to the one applied by the GnsLogger App, an Android application developed and maintained by Google. Apart from their compression to a zip format, no additional modifications of the files are made before uploading. Therefore, rather than collecting the RINEX files as a result of the in-phone conversion, the users were uploading GNSS observations in the form of comma-separated-values (CSV) files, which were subject to subsequent processing at the server side. As a consequence, other sensor data can be submitted in the same file besides having the possibility of preserving

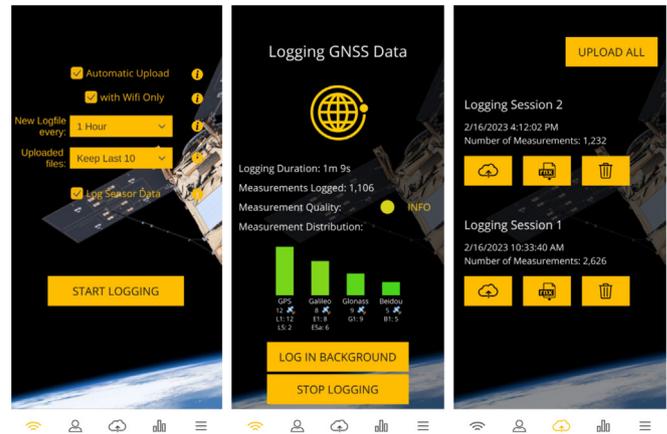


Fig. 2. CAMALIOT Android Application: User Interface (ver. 0.5.9.8). Shown is an interface for configuration of the logging session (left), an interface displayed while taking measurements (center), and a subpage for data upload and RINEX conversion (right). Besides GNSS observations, the developed application allows also to collect data from environment and motion sensors.

original GNSS observations. The file uploading itself was realized via a POST command that was made to the dedicated endpoint connected to the component responsible for their ingestion (*CamaliotIngest*). Although possible due to the utilized software stack (currently apart from the GNSS engine), real-time data streaming of observations from the *CamaliotApp* to the server has not been implemented and this activity focused on a quasi-continuous collection of GNSS observations from smartphones.

The compressed zip files from the *CamaliotApp* instances reached the *CamaliotIngest* component, where they were processed in an asynchronous manner. The *CamaliotIngest* component was directly connected to an Ingress Controller, which is a service that routes the traffic from outside of the cluster to the services and pods inside the cluster. At that stage, the received files were forwarded to object storage (backup of raw observations) as well as uncompressed and parsed for subsequent processing.

*CamaliotConnect* represents middleware that, when utilized as one of the SW components, keeps the data integration simple with the option to seamlessly integrate new micro-services or services. Via this middleware, e.g., the compressed log files (as produced by *CamaliotApp*) from *CamaliotIngest* are continuously uploaded to the dedicated bucket (in the object storage), whereas the serialized data from *CamaliotIngest* undergo further processing using the remaining CAMALIOT SW components that might need to process the same data differently. Connection of the CAMALIOT SW components, communication with external services (object storage or GSSC) and the support concerning the streaming data paradigm, where data are represented as streams of events, was implemented with the use of the Apache Kafka ecosystem (and by leveraging the AVRO serialization format), its open-source extensions, schema registry as well as the custom Python-based REST API in order to meet specific tasks related to data

<sup>3</sup> Camaliot: available on Google Play Store

backup/retrieval or triggering of the processing to be carried out by the components of the CAMALIOT SW.

The *CamalioProcess* component, the multi-purpose processing layer, concerned batch and stream processing of the collected data where the required assets are provided through the *CamalioConnect* component. Tasks associated with this component included data enrichment, aggregation, and creation of RINEX-3 files with GNSS observations acquired with the use of *CamalioApp*. In terms of scalability, batch/stream processing required for the purposes of this activity was addressed with the use of Apache Kafka and Apache Spark, open-source frameworks commonly used in modern platforms working with huge volumes of data. More details concerning the processing steps are given in Section 2.3.

ML-related parts for spatial interpolation and forecasting of time series related to the troposphere and ionosphere were implemented as separate components, see Fig. 1. The subset of libraries mentioned in Section 1 formed a proof-of-concept setup that was tested throughout the course of the project for ML and DL at scale. The output of such components are ML-based models and predictions, where both can be treated as the final output of the CAMALIOT SW. More information concerning the ML/DL framework as implemented in the CAMALIOT SW are given in Section 2.5.

*CamalioLab* represents a web-based sandbox component that has been established for demonstration purposes allowing to experiment with GNSS and ML. The user performs analysis tasks with the use of JupyterLab, the provided inputs and/or stored assets.

Software component monitoring plays an important part in assuring high availability of any software running in production and when aiming for minimal down-times. For the CAMALIOT SW, a component dedicated to monitoring (*CamalioMonitoring*) was built upon three entities, i.e., Prometheus Operator, Prometheus, and Grafana. Prometheus is a tool that collects metrics from the monitored applications (components and subcomponents) and stores it with the timestamp at which it was recorded. The metrics can be of various kinds as different types of applications can be monitored (databases, Ingress controllers, Apache Kafka). Such metric scraping occurs over HTTP via a pull model. In other words, the SW components and sub-components expose the metrics at a certain port so that the Prometheus server can collect this data. Subsequently, such time series data allows the performance of the applications to be investigated, and, if needed, utilized to improve their performance. However, a more important aspect is the capability of a quick response in case one of the SW components fails. Moreover, Horizontal Pod Autoscaling (HPA), that is used in conjunction with the scraped metrics, can be used in the Kubernetes environment for an optimal resource management by scaling up or scaling down (horizontally) specific micro-services based on specific metrics, e.g., those related to the data load or the number of files in the processing queue. Finally,

Grafana is an open-source analytics web application (with built-in support for Prometheus) providing charts, graphs, and alerts in order to visualize and interact with metrics of various kinds. An example of the Grafana web-based interface with metrics related to the *CamalioConnect* component is shown in Fig. 3.

### 2.3. GNSS Data Pre-processing

A number of processing steps occur within the *CamalioProcess* component. Due to the variety of tasks, several subcomponents comprise that part of the CAMALIOT SW.

GNSS-related serialized data present in the Kafka topics are utilized in the *CamalioProcess* component to create device-specific hourly RINEX-3 files. The RINEX converter developed by IIASA (and present also in *CamalioApp*) is used to convert the incoming data into RINEX-3 files. That custom converter was implemented for an efficient and automated conversion of the available files while ensuring receiver clock consistency between the pseudo-range and carrier-phase measurements in the created files. At this stage, the 9-character device name is either generated or retrieved in order to create such RINEX files. To diminish the size of the files while including an additional validation step, the created RINEX files are subsequently converted to Compact (Hatanaka) RINEX (CRX) files and then compressed to gzip files. The latter are then uploaded to *CamalioConnect* for their backup on the object store, where they are partitioned by year and day of year, based on the timestamp present in the filenames. Such files are also available for further processing, in conjunction with RINEX files that contain observations from geodetic stations.

In order to save processing time and minimize the impact of poor quality GNSS observations on the quality of the derived products and models, it is proposed that, at this stage, the RINEX files undergo the screening process in the subcomponent dedicated to quality control. Only the hourly files with measurements of proper quality are utilized at a later stage to form a complete observation session. Based on the utilized measures, RINEX files containing observations of low quality are therefore not included in the subsequent processing.

The available RINEX files form an input to the GNSS processing subcomponent that was implemented as a micro-service with the dedicated API for receiving RINEX-2/RINEX-3 files, configuration files or calls from other SW (sub) components. At this stage, GNSS products, auxiliary information and collected RINEX files are used in the GNSS processing utilizing the Extended Kalman Filter (EKF) that is present in the dedicated GNSS engine, hereafter referred to as *CamalioGNSS*. Parallel processing of RINEX files is implemented based on the concept of a processing queue that is shared across the micro-service's replicas. Depending upon the needs, the GNSS micro-service can be then scaled horizontally in order to accom-

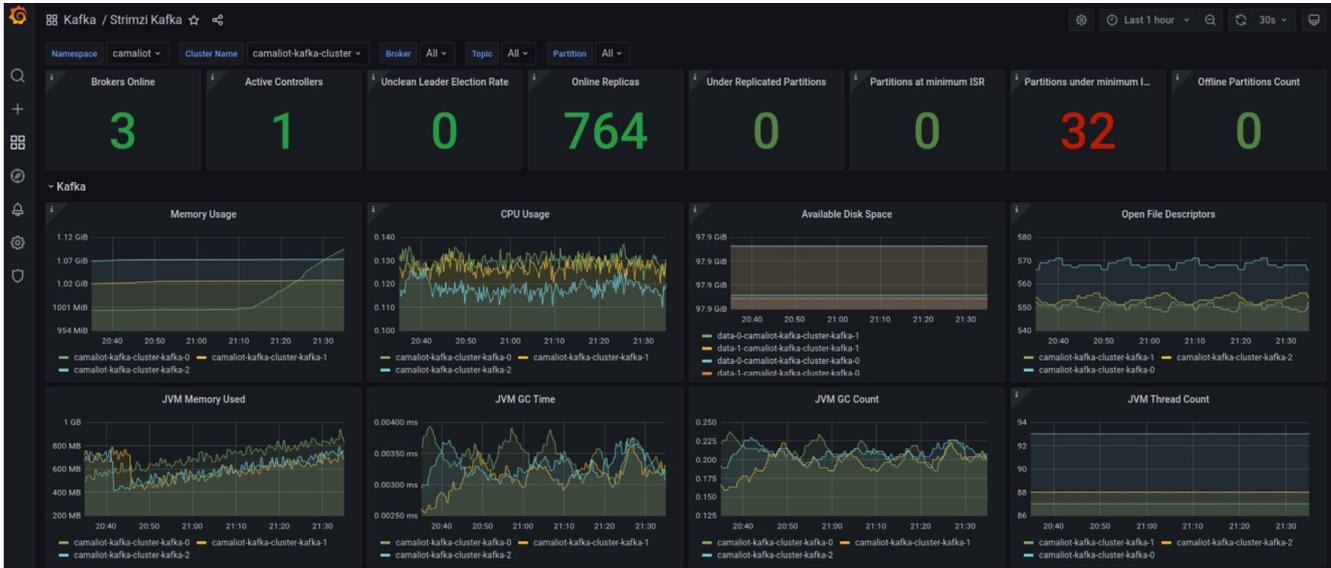


Fig. 3. CAMALIOT SW: An example of the utilized Grafana dashboard (based on the Strimzi template) displaying metrics from the *CamaliotConnect* component and related to the Kafka cluster.

modate an increased amount of the incoming RINEX files. Once the processing is finished, the troposphere-related estimates are separated into files in the SINEX TRO format<sup>4</sup>, whereas the acquired ionosphere-related satellite-specific VTEC/STEC estimates are stored in the SINEX ION files. In the last step, both SINEX TRO and SINEX ION files can be submitted via *CamaliotConnect* for backup and their further utilization either as new assets of the GSSC or as an input to the *CamaliotIono* and *CamaliotTropo* components. The SINEX ION format was introduced for this activity and corresponds closely to the SINEX TRO format in the way how the estimates are represented. Example SINEX ION files can be found in the related data repository, see Acknowledgments.

#### 2.4. GNSS Engine

As a part of this activity, a dedicated GNSS engine has been developed that allows diverse static/kinematic multi-GNSS observations to be processed, either single-frequency or dual-frequency, such as those collected by high-grade IGS stations, low-cost receivers or crowd-sourced GNSS observations. The motivation was the possibility of deriving a consistent set of ionospheric and tropospheric estimates from various GNSS sources to be used at a subsequent stage that is related to ML/DL.

The analysis approaches relevant for this activity include CCL, differential GNSS (known as relative positioning or baseline processing) and PPP with the latter introduced by [Zumberge et al. \(1997\)](#) for static GPS and later used for kinematic GPS ([Kouba and Héroux, 2001](#)).

RTKLIB ver. 2.4.3 b34 ([Takasu and Yasuda, 2009](#)) formed the base of *CamaliotGNSS*, which has been extended with additional features (e.g. multi-GNSS support, flexible GNSS observation type selection, SINEX interface, CCL module) tailored to the considered use cases. Parameter estimation is performed using EKF with the target and nuisance parameters expressed as EKF states (epoch-wise state vectors and covariance matrices). In the case of post-processing, the parameter estimation is possible with either a forward-in-time filter, backward-in-time-filter or via the combined solution taking advantage of both forward and backward filter solutions in order to address the initialization periods of the KF, apart from improving the quality of the solve-for parameters through smoothing (in essence a weighted average). For baseline processing (BP), the carrier-phase integer ambiguity resolution ([Teunissen et al., 1997](#)) is available. In relation to PPP, the carrier-phase integer ambiguity resolution (AR) step that underpins the concept of PPP-RTK ([Teunissen and Khodabandeh, 2015](#); [Li et al., 2022b](#)), or also known as PPP-AR ([Glaner and Weber, 2021](#); [Chen et al., 2021](#)), has not been implemented so far.

The developed engine allows satellite-specific VTEC time series to be extracted based on the KF-based CCL approach, in which leveling biases are separated from VTEC on a satellite-by-satellite basis and reinitialized for each new satellite arc or whenever a cycle slip occurs. Currently, differential code biases (DCBs) cannot be estimated and external products need to be used instead. For IGS stations, the DCB information can be provided to the engine as either files with GNSS bias products as produced by CODE (DCB files), which applies only to GPS and GLO-NASS observations, or by using files in the BIAS SINEX (BSX) format, where the same parameters are available

<sup>4</sup> [https://files.igs.org/pub/data/format/sinex\\_tro\\_v2.00.pdf](https://files.igs.org/pub/data/format/sinex_tro_v2.00.pdf)

but for multiple GNSS constellations. The latter option allows also different pairs of observations and frequencies to be utilized to form a geometry-free linear combination of pseudorange and carrier-phase observations.

#### 2.4.1. Analysis Setup

The GNSS processing strategy in relation to this activity can be divided into two steps. In the first step, RINEX observations from geodetic-grade stations are processed in order to obtain precise coordinates, SINEX TRO files and SINEX ION files. The parameterization of the solve-for parameters estimated in our analysis approach and in relation to that step is highlighted in Table 1 and Table 2 for extraction of ZTD and VTEC, respectively. The subsequent step concerns alternative sources of GNSS observations (in this case smartphone data), which are processed mainly in the baseline mode in order to derive ZWD in a relative manner and using (so far) CCL to extract VTEC. The parameterization applied in that case is shown in Table 1 and Table 2 for extraction of differential ZWD and VTEC, respectively.

Relative positioning, exploiting double differencing of measurements, on short baselines allows single-frequency GNSS observations to be utilized for deriving rover position and differential ZWD (Webb et al., 2016; Fermi

et al., 2019) without the need for handling of the satellite/receiver clock contributions (the double-difference approach virtually eliminates clock errors) and differential ionospheric delays as their impact in this case is negligible (due to the spatial behavior of the ionosphere). The premise in this case is that the neglected ionospheric effects on relatively short baselines do not corrupt the solve-for parameters. The important prerequisite is that the ZWD time series of the reference station are available and not estimated during the processing as almost parallel signal paths of the receivers to the same satellite prevent the simultaneous estimation of ZWDs for each station forming short or moderate baselines (same situation applies to local multi-station GNSS networks). As a result, baseline processing provides a relative estimate of the ZWD that is expressed with respect to the value established for the reference station. In our case, the ZTD estimates from step one and stored as SINEX TRO files can be used to supply the relative positioning with tropospheric estimates for the base station. As this is a local technology (range of tens of kilometers), the proposed single-frequency baseline approach requires, however, a dense network of reference stations to provide regional coverage for processing of smartphone observations. This is however, not an issue when considering alternative GNSS data as a complemen-

Table 1  
GNSS data processing strategy in CamalioTGNSS for IF-PPP and single-frequency (L1/E1) baseline processing (the double-difference approach) based on high-grade and smartphone-based GNSS observations. Process noise abbreviated as PrNo.

Parameter	Parameterization		
Processing method	IF-PPP	baseline processing	
Data type	high-grade obs.	high-grade obs.	smartphone
Solution type		EKF combined run	
Obs. resolution [s]		30	
Constellations		GPS & Galileo	
Cut-off elevation angle		7°	
SNR mask [dB/Hz]	35	rover: 35 base: 35	rover: 25 base: 35
A priori noise (raw obs.)	code: 0.3 m phase: 3 mm	code: 0.3 m phase: 3 mm	code: 1.0 m phase: 6 mm
Weighting		3 mm / sin $el$	
Mapping function		GMF	
Ionosphere	ionosphere-free combination	eliminated with double differencing (short baselines)	
Troposphere:			
- Parameterization	ZTD (random walk), no grad.	differential ZTD (random walk), no grad.	
- PrNo std. dev. [m/√s]	$5 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	
- State std. dev. [m]	0.3	0.1	
Ambiguity:			
- Resolution mode:		float	
- State std. dev. [m]	60.0	10.0	
Receiver clock:			
- Parameterization	white noise	-	
- State std. dev. [m]	60.0	-	
Inter-system bias	constellation-specific receiver clock	eliminated with double differencing	
Satellite orbit & clock	5-minute & 30-second ESA precise products	broadcast ephemerides	
Earth rotation parameters	ESA	-	

Table 2  
GNSS data processing strategy in CamalioTGNSS concerning the KF-based CCL for VTEC extraction applied to high-grade GNSS observations and smartphone data. Process noise abbreviated as PrNo.

Parameter	Parameterization	
Data type	high-grade obs.	smartphone
Solution type	EKF combined run	
Obs. resolution [s]	30	2
Constellations	GPS & Galileo	
Cut-off elevation angle	15°	30°
SNR mask ( $f_1/f_2$ ) [dB/Hz]:	35/35	25/25
Carrier smoothing of code observations	-	10 samples
A priori noise (raw obs.)	code: 0.3 m	1.0 m
	phase: 3 mm	6 mm
Weighting	3 mm / $\sin \epsilon_l$	
Ionosphere:		
- Parameterization	vertical delay & delay rate	
- PrNo std. dev. [m/ $\sqrt{s}$ & m/s/ $\sqrt{s}$ ]	$1 \cdot 10^{-4}$	
- State std. dev. [m & m/s]	1.0 & 0.01	
Mapping function	single-layer model	
Ionosphere height [km]	450.0	
Leveling bias:		
- State std. dev. [m]	30.0	
Differential code biases	BSX products from CAS	

tary source of information that could be potentially used to densify the current network of high-end receivers commonly used for atmospheric monitoring.

The analysis strategy applied to smartphone data differs from the approach including geodetic-grade stations in two aspects. In general, the utilized SNR mask needs to be lower in order to not exclude a large number of decent observations. The latter are characterized on average by approximately 10 dB-Hz lower carrier-to-noise density ratio (C/N0) than those collected with high-end antennas and receivers (Zhang et al., 2020). In addition, different values are applied concerning observation weighting, but the approach utilizing elevation-dependent weighting remains unchanged.

In terms of ionosphere extraction from smartphone data, some changes with respect to the parameterization applied for geodetic-grade stations are made mainly to reduce the negative impact of potentially poor observations on the precision and trend of the STEC time series. Apart from the different stochastic models used and a lower SNR mask applied, no observations below 30° are included in the processing. By default, the smartphone-related observations are recorded with a sampling rate of a single second. In our case, the observations are down-sampled to two seconds (to save computing time) and carrier smoothing of code observations is introduced in order to diminish the negative impact of outliers in the pseudorange measurements at both L1 and L5 frequency bands.

## 2.5. Machine Learning with the CAMALIOT SW

The most computationally demanding stage in the case of modern ML/DL pipelines is the model training part, with the training data size and the complexity of the employed ML model architectures as the main contributing factors. In order to address that, the developed ML/DL framework utilized across the components of the CAMALIOT SW leverages the combination of Apache Spark, TF2, Horovod, and Open MPI to support model prototyping with the use of either scikit-learn, which covers standard ML models, or TF2, which is utilized commonly for DL. In both cases, distributed/parallel training is performed on Spark executor nodes as SparkApplications. Such a combination of open-source libraries (Apache Spark, TF2 and Horovod) has several benefits such as high-throughput batch/continuous processing or distributed model training. In order to train the models prototyped with scikit-learn, the joblib-spark library serves as a means for executing parallel jobs on Spark executor nodes. Such an approach unifies data processing and model training with the use of various libraries and encapsulates them within a single ML pipeline. Based on the consistent set of GNSS-related estimates such as VTEC or ZWD with the simultaneous use of auxiliary variables related to troposphere and space weather as input features, the goal behind the development of the dedicated ML/DL models is the spatial representation of those two variables with utmost accuracy as well as short- to mid-term forecasting of those parameters. In this regard, example studies related to the CAMALIOT project are shown, e.g., by Crocetti et al. (2022) or Soja et al. (2023). The same concept is investigated by Mao et al. (2023) to generate ML-based GIM using multi-GNSS data and the KF-based CCL approach.

## 3. Results

This section provides information on the quantity of the GNSS data collected throughout the course of the conducted crowdsourcing campaigns (an intrinsic part of the activity) with the use of the developed Android application. This is followed by example processing results related to ZTD and STEC derived with the use of the developed GNSS engine and based on the GNSS observations from permanent GNSS stations and data acquired with the use of Xiaomi Mi 8 during local measurement sessions.

### 3.1. Quantity of Crowdsourced GNSS Observations

The two conducted crowdsourcing campaigns resulted in a significant amount of GNSS observations that were collected per day. The daily quantity and volume of the observations as collected with the use of the *CamalioTApp* instances and submitted to the CAMALIOT SW (the *CamalioTIngest* component) are shown in Fig. 4

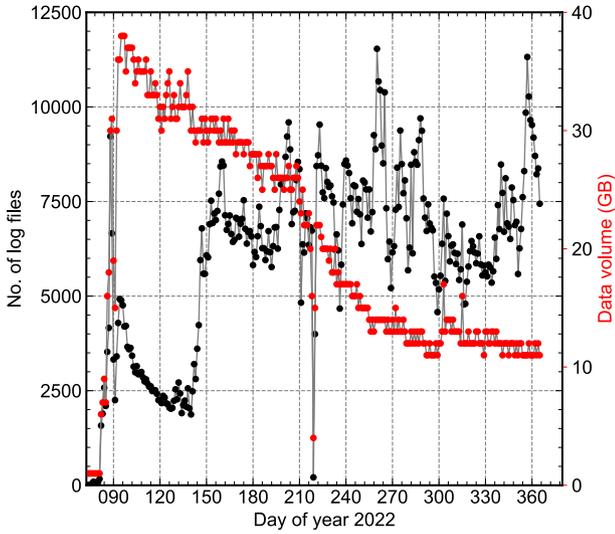


Fig. 4. Quantity and volume of community data collected as zip archives throughout the course of activity with the use of the developed CAMALIOT Android application. The number of log files corresponds to the quantity of the compressed text files (CSV files containing raw GNSS observations) that were uploaded by the registered users. The crowdsourcing campaigns concern days 76–334 (inclusive) of 2022.

Over twelve thousand registered application users from various parts of the globe were interested in contributing to this activity, some of who have retained an ongoing interest in contributing to data collection (See et al., 2023). As the crowdsourced observations were collected in an uncontrolled manner, initially the size of the incoming files varied significantly until the continuous mode was introduced to control the data flow in this regard and achieve quasi-continuous and stable data ingestion at the server side. The introduction of the continuous mode in *CamaliotApp* is visible in Fig. 4 as a radical change in the number of collected files that occurs around day 140 of the year.

### 3.2. GNSS Processing

#### 3.2.1. High-Grade GNSS Observations

This subsection is dedicated to the example tropospheric and ionospheric estimates derived with CamaliotGNSS and using multi-constellation GNSS observations from selected IGS stations. In terms of ZTD, the applied processing strategy concerning IF-PPP and baseline processing is summarized in Table 1. For the sake of comparison with smartphone-related estimates, carrier-phase integer ambiguity resolution, which is applicable in the presented context only in relation to baseline processing, was not applied in the analysis.

ZTD time series from the IF-PPP approach for stations WTZR00DEU and AIRA00JPN for a single day from the year 2023 are shown in Fig. 5, complemented with external and publicly available ZTD estimates. The latter include time series produced by the Nevada Geodetic Observatory

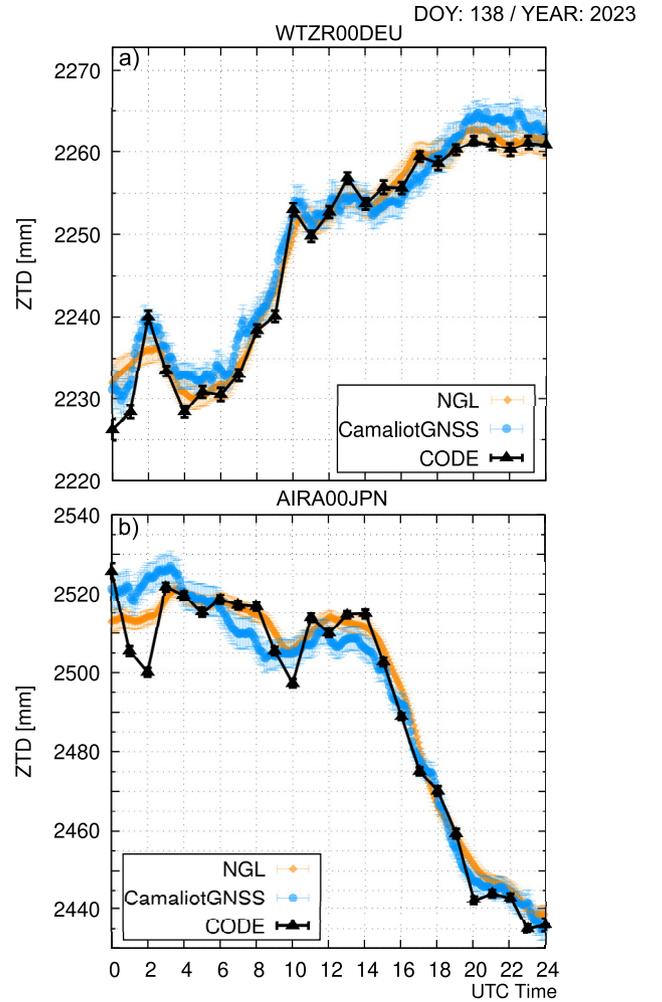


Fig. 5. ZTD time series (and their formal errors) at (a) WTZR00DEU and (b) AIRA00JPN for 18.05.2023 (doy 138). The ZTD time series from CamaliotGNSS (blue) derived with IF-PPP and based on both GPS and Galileo observations are shown along with equivalent estimates by NGL (orange), and CODE (black). Here, estimates from the CamaliotGNSS engine obtained through the EKF combined run with 30-s observation sampling are shown with a 5-min resolution.

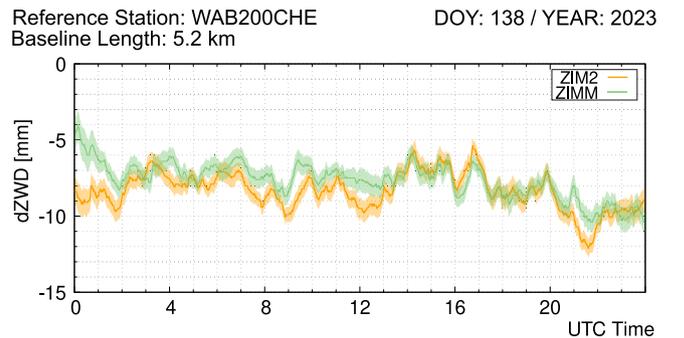


Fig. 6. Baseline processing results for the ZIM200CHE-WAB200CHE and ZIMM00CHE-WAB200CHE baselines. Shown are dZWD time series with formal errors (confidence interval of 68%) over the period of 24 h for May 18th, 2023. Unlike ZIM200CHE, ZIMM00CHE does not support Galileo observations and only GPS observations were used in the solution. ZIM200CHE and ZIMM00CHE abbreviated as ZIM2 and ZIMM, respectively.

(NGL) with the use of GipsyX (Bertiger et al., 2020), and CODE based on the solutions from the Bernese GNSS Software (Dach et al., 2015). An example tropospheric corrections (dZWD) and their formal errors at ZIM200CHE and ZIMM00CHE derived from baseline processing at L1/E1 are shown in Fig. 6 for a 24-h period from the year 2023. The time series of the co-estimated positions (for the same period) are given in addition in Appendix A. In both cases, WAB200CHE station was used as a reference, which results in tropospheric corrections that reflect the change in tropospheric conditions between stations separated by approx. 5.2 km. Concerning two rover stations, ZIMM00CHE and ZIM200CHE are separated by a distance of approx. 19 m with the height difference of approx. 0.1 m. For the considered period, the median difference of dZWD for ZIM200CHE with respect to the dZWD derived at ZIMM00CHE is  $-0.7$  mm with the sample standard deviation of the epoch-specific dZWD differences of 1.00 mm.

CamalioGNSS allows VTEC to be derived using the EKF-based CCL approach, currently with the use of DCB information provided externally. As they are generated on an operational basis with a daily resolution, satellite and station biases computed by the Institute of Geodesy and Geophysics (IGG) of the Chinese Academy of Sciences (CAS) in Wuhan are used here (Wang et al., 2016). The GNSS processing strategy for the CCL-based satellite-specific VTEC determination using GPS (G) and Galileo (E) observations is summarized in Table 2. As a validation of the implemented VTEC extraction approach, STEC time series based on the GNSS observations from two selected IGS stations (WTZR00DEU and AIRA00JPN), experiencing different ionospheric conditions, are shown in Fig. 7 in the form of differences between satellite-specific STEC derived with the use of CamalioGNSS and the same quantities calculated based on publicly available GIM. For the sake of consistency in terms of the utilized DCB information, GIM as disseminated by CAS were used here for calculating the STEC differences.

The median offset between the satellite-specific STEC time series for WTZR00DEU and those generated with the use of GIM is 0.92 TECU and 0.87 TECU for the group of GPS and Galileo satellites, respectively. In the case of the ARIA00JPN station, the same quantity is larger for both constellations and amounts to 2.39 TECU and 3.25 TECU for GPS and Galileo satellites, respectively. Note that the results shown in Fig. 7 concern the period of high solar activity.

### 3.2.2. Smartphone Observations

The following subsection is dedicated to results concerning the smartphone observations. Those were collected with *CamalioApp* running on Xiaomi Mi 8, hereafter referred to as XIM8, which supports dual-frequency observations. The resulting CSV files (native format of the developed application) were then converted to RINEX-3 files with our custom RINEX converter. As an additional fea-

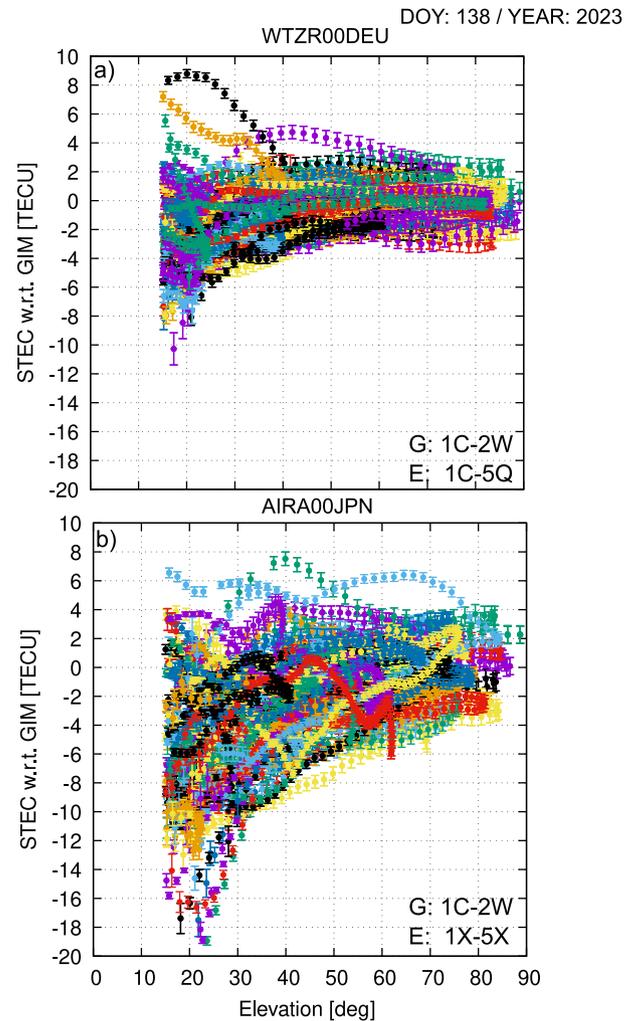


Fig. 7. STEC time series (and their formal errors) using GPS and Galileo observations from a) WTZR00DEU and b) AIRA00JPN for 18.05.2023 (day of year 138) reduced by the STEC generated with the use of GIM by CAS. The differences from a 30-s solution are shown with a 5-min resolution as a function of satellite elevation. Color-coded are satellite-specific differences. For GPS, 1C-2W pair of observations was used for the geometry-free combination, whereas Galileo-related geometry-free combination was created with the use of 1C-5Q and 1X-5X pairs for WTZR00DEU and AIRA00JPN, respectively.

ture, the same converter is also available in *CamalioApp*. The measurement platforms, consisting of a holder and a radome, used during the measurements are shown in Fig. 8, where XIM8 was placed both horizontally and vertically. The installed measurement platforms were located in a close vicinity to the ETH200CHE station, hereafter abbreviated as ETH2, that served in this case as a means of validation of the smartphone-derived tropospheric and ionospheric results. In terms of the observation continuity, the utilized smartphone has its limitations and did not allow continuous recording over periods spanning several days. After some trial runs and screening through available options in the settings, XIM8 was able to continuously record 1-s data over a period of up to 24 h without any human intervention during the performed measurement



Fig. 8. Two platforms used during measurements carried out with Xiaomi Mi 8 and *Camaliot.App*. **a)** location of the smartphone collecting data between 26/04/2022 12:00 UTC and 27/04/2022 12:00 UTC (day of year 116, 117) and between 28/04/2022 07:30 UTC and 29/04/2022 00:00 UTC (day of year 118), **b)** the measurement platform (the radome located in the background) utilized for measurements carried out on 17/05/2023 and 18/05/2023 (day of year 137 and 138). The platform in the foreground utilized during measurements carried out (in 2022) on day of year 116, 117, and 118.

sessions. Prior to collecting the measurements, duty cycling was also disabled through the change of the available settings in the Android OS in that regard. In relation to carrier-phase observations, duty cycling has a negative impact on the GNSS analysis as it results in unwanted cycle slips.

For smartphone observations, baseline processing on short to moderate baselines can be used to retrieve the tropospheric contribution alongside the smartphone coordinates based on single-frequency observations, which in this case are L1 and E1, corresponding to GPS and Galileo frequency bands, respectively. The applied processing strategy is given in Table 1. In order to be compliant with the common observation resolution of geodetic-grade data that is publicly available to the analysis, baseline processing was carried out with smartphone observations that were down-sampled to 30 s. The created baseline consisted of a smartphone and a geodetic-grade GNSS receiver that was utilized as a base station. The base station (Septentrio PolaRx5 with the JAVAD GrAnt G3T antenna), hereafter referred to as PRTW, was located on top of the Prime Tower in the city of Zürich, which is approximately 2.4 km from both ETH2 and XIM8, with the height differ-

ence between PRTW and ETH2/XIM8 of approx. 17 m. For validation purposes, ETH2 was used in conjunction with the same base station to produce reference differential ZWD (dZWD) time series for the same time period. The differential ZWD acquired with the described approach for both of the employed baselines are shown along with their formal errors in Fig. 9 for the period between 26/04/2022 12:00:00 UTC and 27/04/2022 12:00:00 UTC and period between 28/04/2022 00:00:00 UTC and 29/04/2022 00:00:00 UTC. The time series of the co-estimated smartphone position based on those two sessions are given in Appendix A. The average number of satellites that were used in baseline processing involving XIM8 amounts to nine for both periods considered. The RMS differences between the dZWD time series derived using XIM8 and ETH2 are 2.60 mm and 3.05 mm for the first and the second period, respectively.

Similarly to the utilization of IGS stations for ionosphere delay retrieval, KF-based CCL can be utilized to derive STEC time series from dual-frequency smartphone observations by forming an L1/L5 geometry-free linear combination of the observations. The applied processing strategy is given in Table 2. The 2-s observation resolution was chosen in order to reduce the processing time while including sufficient amount of good observations that KF-based CCL requires to properly separate leveling biases from ionosphere delays. The STEC extraction was complemented here also with DCB data from BSX files with the exception that the constellation-specific receiver DCB (RDCB) has to be either derived additionally or set manually. In our case, the RDCBs for GPS and Galileo were set to zero and only the relative change with respect to the reference satellite-specific STEC time series was investigated. STEC time series from ETH2 served in this case as a means to validate the time series obtained with XIM8. The remaining non-zero offsets between the created time series were reduced by calculating the median offset between those two types of STEC time series to facilitate the comparison of differences in the relative change of ionosphere over the course of the considered periods. The calculated and reduced offsets can be mainly associated with the lack of information on the RDCBs for both stations and the antenna phase-centre offsets (PCOs) for the smartphone.

The results from KF-based CCL processing of dual-frequency observations are shown in Fig. 10 in the form of satellite-specific differences between STEC time series obtained with XIM8 and ETH2 and based on the observations recorded in the period between 17/05/2023 18:00:00 UTC and 18/05/2023 18:00:00 UTC. For the first considered period, the relative satellite-specific variations of STEC are characterized by RMS differences below 0.55 TECU, with the exception that for E09 this statistic increases to 1.14 TECU, which can be explained by the last period, in which a significant deviation between STEC time series based on XIM8 and ETH2 is visible for that satellite. For the second period, the smartphone-based STEC time

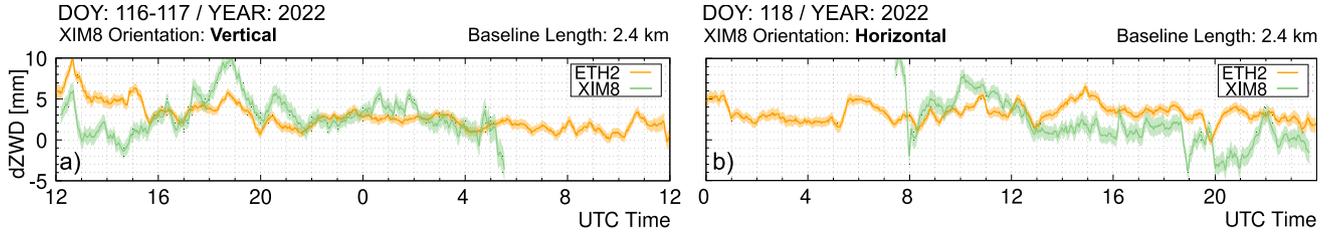


Fig. 9. Time series of differential ZWD with related formal errors (confidence interval of 68%) from baseline processing concerning the ETH200CHE-PRTW and XIM8-PRTW baselines for two different observing sessions **a)** day 116–117 with the smartphone placed vertically during the measurements, **b)** day 118 with the smartphone placed horizontally during the measurements. ETH200CHE abbreviated as ETH2.

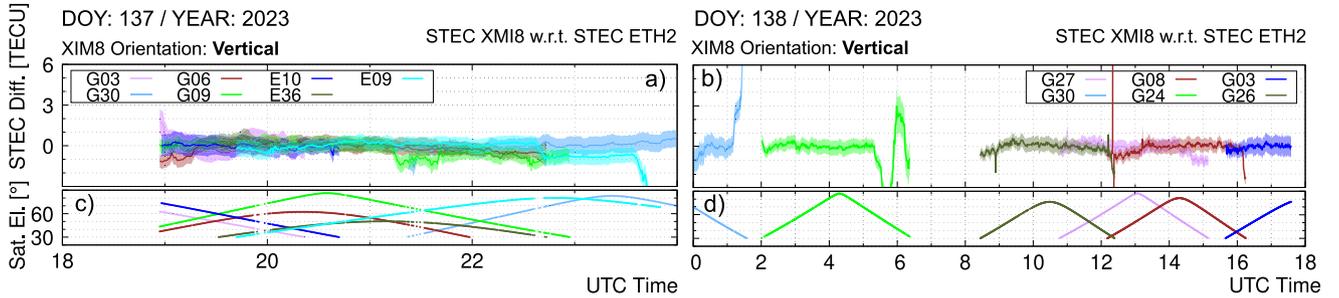


Fig. 10. STEC time series (with corresponding formal errors) generated using a geometry-free combination of observations collected with XIM8 and based on the CCL approach with 2-s obs. resolution. **a), b)** STEC time series of good quality extracted using GPS and Galileo observations from XIM8 and reduced by STEC time series generated using observations from ETH2 and satellite-specific median offset between the two types of time series, **c), d)** corresponding time series of satellite elevation angles during the measurement session. Observation pairs for ETH2: 1C-2W (GPS) and 1X-5X (Galileo). Observation pairs for XIM8: 1C-5Q (GPS and Galileo). The time series shown with the 30-s resolution.

series exhibit less agreement with the time series acquired with ETH2 with an average RMS of 1.11 TECU. In both periods, multiple epochs with noticeable outliers are visible, too.

#### 4. Discussion

The outcomes of the conducted crowdsourcing campaign demonstrated a potential for involving the crowd in contributing to the field of GNSS by collecting measurements on a voluntary basis. Neither the discussion on the quality and characteristics of the collected observations nor the quality of the solve-for parameters from crowdsourced data is discussed in this contribution. This is a topic for a separate study, which also requires further refinement of the GNSS engine for its use in conjunction with such data as they require special treatment, e.g., in terms of the ionosphere handling for single-frequency observations (Deng et al., 2009) or outlier removal. However, one can state with great certainty that only a small fraction of the collected observations might be of satisfactory quality for their use in the aforementioned use cases due to the way in which such observations might have been collected. Nevertheless, the crowdsourced GNSS data and the collection process demonstrated in this activity may provide further insights concerning, for instance, model-specific performance of smartphones in terms of carrier-phase observation quality and integrity or how one could

revise such data collection campaigns in the future in order to increase the number of GNSS observations of sufficient quality for their use in atmospheric research.

In terms of the performance of the GNSS engine forming a part of the proof-of-concept cloud-native software, one can notice, in general, a good agreement between the obtained ZTD time series from the IF-PPP processing mode, as described in Section 3.2.1 for two selected IGS stations, and the external tropospheric products generated with the use of state-of-the-art GNSS analysis software packages. However, in order to improve the quality of the solve-for parameters and their consistency with respect to external products, further work would be required in terms of the PPP module of CamaliotGNSS in order to implement the missing models and identify improper modeling of the individual effects that undifferenced GNSS processing is sensitive to. Introduction of new processing modes such as UC-PPP might be also of benefit for simultaneous retrieval of VTEC and ZTD based on geodetic-grade GNSS observations or low-cost devices. In general, the VTEC extraction method based on UC-PPP might be beneficial for the creation of global or regional ionospheric models as this is a method that is characterized by better precision and consistency compared to the CCL-based VTEC retrieval (Xiang et al., 2019). Of benefit could also be a work towards real-time processing for both real-time troposphere and ionosphere products or exploitation of the PPP-RTK approach. Those are not yet supported in the developed engine.

The single-frequency baseline processing was useful to acquire tropospheric estimates from smartphone observations. In terms of the approach, it is similar to the investigation made by [Stauffer et al. \(2023\)](#), but with few differences. Apart from utilization of a different smartphone model, the processing was made on a much shorter baseline, with much lower temporal resolution, and by utilizing a combined EKF run with a slightly different parameterization. In addition, no subsequent processing of the resulting tropospheric time series was made here. In general, the vertical orientation of the phone (day of year 116–117, 2023) in our case turned out to be more beneficial than the horizontal orientation (day of year 118, 2023) as the derived dZWD time series correspond more closely to the reference time series acquired with the use of the high-end GNSS station.

The derived satellite-specific VTEC time series are generally in good agreement with GIM in terms of the trend. The CCL approach implemented in CamalioTGNSS has not yet been refined in order to properly handle observations collected from smartphones or non-IGS receivers. In those cases, the estimation of receiver DCB (while using external information on the satellite DCBs) is required in order to express the VTEC estimates in an absolute manner. Apart from the standard approaches where the local ionosphere model is used to separate the ionosphere from instrumental effects ([Zhang et al., 2018](#); [Xu et al., 2022](#)), for special cases as those including smartphones, one could derive receiver DCBs using an alternative approach, either with the use of external TEC products or based on the rudimentary assumptions concerning the stability of the receiver DCBs and the spatial behavior of the ionosphere on local scales ([Otsuka et al., 2000](#); [Arikan et al., 2008](#)), assuming that the sufficient number of observations is available for the analysis and taking into account the impact of the time-varying receiver DCB on the STEC retrieval accuracy. In terms of the high temporal resolution of observations that was applied to retrieve STEC time series from local measurements with Xiaomi Mi 8, this was mainly performed in order to diminish the negative impact of spurious observations on the estimates. In relation to the same smartphone model and a similar smartphone orientation, STEC extraction has already been undertaken by [Xu et al. \(2022\)](#) who used the entire 1-s data set in the analysis.

Currently, there are a few technical aspects that limit the accuracy of smartphone-based positioning and the quality of the derived troposphere-related or ionosphere-related parameters. Among others, an important factor to consider in relation to smartphones is a less stable tracking loop compared to high-grade GNSS receivers, resulting in a more frequent loss of lock causing more cycle slips. The orientation of the phone (vertical or horizontal) also tends to play a major role in the availability of carrier-phase observations and the quality of the solve-for parameters. Throughout the course of the activity, it was noticed that placing the smartphone in a vertical position is beneficial

due to the greater availability and integrity of the GNSS observations per epoch. Such an observation tends to be also claimed in other studies related to the processing of smartphone-based GNSS observations ([Yong et al., 2021](#); [Li et al., 2022a](#)). Another factor that needs to be taken into account when coping with smartphone observations is the lower level of multipath suppression of the built-in smartphone antennas compared with external antennas that are used in conjunction with either low-cost or high-end GNSS receivers ([Li and Geng, 2019](#)). In order to address this issue, recognized approaches for multipath mitigation, as applied to static GNSS observations ([Dong et al., 2016](#)), could potentially be transferred to static smartphone measurements in order to improve their usability for scientific investigations. In a more direct approach, the utilization of an external low-cost antenna ([Li et al., 2022c](#)) could be beneficial to further reduce the impact of this error source on the target parameters. In our case, the most pronounced impact on the results was the lack of the ability of Xiaomi Mi 8 to continuously collect data over long periods of time, which we connect to hardware and software constraints. There are, however, other smartphones for which such a problem does not exist and GNSS data collection could span long periods of time ([Stauffer et al., 2023](#)). With advances in research on the utilization of smartphones for precise positioning or atmospheric sounding, dedicated calibration efforts with the aim of deriving reliable antenna PCO and phase-centre variation of smartphone antennas would also be of benefit for improving the accuracy of the estimates ([Darugna et al., 2020](#)).

## 5. Conclusions and Outlook

This contribution provides a general overview concerning the proof-of-concept cloud-native software developed to meet the goals of the CAMALIOT project. In addition, the description of the approach for consistent processing of GNSS observations collected by high-end receivers and smartphones is presented. This is complemented with example processing results concerning both types of GNSS data. In this case, the aim of the GNSS analysis was to derive tropospheric and ionospheric estimates from both single-frequency and dual-frequency GNSS observations.

The CAMALIOT project was an ESA NAVISP Element 1 project (NAVISP-EL1-038.2) that consisted of a set of activities covering acquisition of GNSS observations from the modern generation of smartphones and the development of a cloud-native software dedicated to GNSS processing and ML at scale based on both conventional and crowdsourced GNSS observations complemented with auxiliary data sets and models relevant to the troposphere and ionosphere.

The CAMALIOT software encompasses acquisition, ingestion, and pre-processing of smartphone-based GNSS observations, aggregation of the data into the data lake (object storage), and their conversion to RINEX files.

The integral part of the software are ML-related components with the aim of performing a fusion of GNSS-based tropospheric and ionospheric products with relevant models and indices of various spatio-temporal resolution and latency. As a part of this activity, an Android application was also designed, implemented and made publicly available in order to collect GNSS observations at scale from a wide range of smartphones as operated by the registered application users throughout the course of the two dedicated crowdsourcing campaigns. This resulted in a significant amount of GNSS observations that were collected per day and are available for the subsequent analysis. That part of the activity demonstrated the capacity of the CAMALIOT SW to collect GNSS observations at scale.

In terms of the GNSS analysis, the related processing component includes a dedicated engine (CamalioTGNSS) that has been established in order to process heterogeneous GNSS data in a consistent manner. The engine employs an Extended Kalman Filter with a smoother to make use of both forward and backward filter solutions. Concerning processing of diverse GNSS observations, a two-step approach is proposed, where in the first step ZTD and STEC are retrieved from GNSS observations collected by geodetic-grade static GNSS stations. This is achieved by applying the carrier-to-code leveling and precise point positioning to retrieve ionospheric and tropospheric estimates, respectively. The second step concerns alternative sources of GNSS data, such as smartphones, for which one applies carrier-to-code leveling and a single-frequency double-difference approach to retrieve STEC and differential ZWD, respectively. Concerning the latter, this is performed on short to moderate baselines in order to avoid the negative impact of the ionosphere delays on the target parameters.

Example processing results in relation to ZWD and STEC concerning both types of aforementioned GNSS observations were derived by employing CamalioTGNSS and validated with the use of the reference time series and external products. Smartphone observations were collected with the use of the developed Android application running on Xiaomi Mi 8. The latter was placed both vertically and horizontally during local measurement sessions. In general, the extracted STEC and differential ZWD time series are in a good agreement with respect to the reference data. However, special attention needs to be paid when processing smartphone observations to acquire reliable atmospheric estimates. This includes additional processing steps, custom parameterization of the solve-for parameters, or the orientation of the smartphone itself. Currently, there are also a few technical aspects that limit the accuracy of smartphone-based positioning and quality of the derived troposphere-related or ionosphere-related parameters.

The outcomes of the CAMALIOT project and the developed proof-of-concept cloud-native software will be useful for extending the existing infrastructure of the GNSS Science Support Centre of ESA. Due to the way in which raw observations are stored and submitted to

the centralized server, crowdsourcing of observations from smartphones with the use of the proposed approach could be seamlessly extended to data from motion and environmental sensors for their potential use in various domains. Although the primary focus of the activity was the collection of GNSS observations from smartphones, the proposed architecture could be also utilized to collect various data from other types of receivers/sensors with data transmission capabilities such as low-cost GNSS receivers or devices mounted on ships, trains, drones, or vehicles. The proposed scalable architecture for processing crowdsourced GNSS observations and data fusion with the use of ML could, therefore, be extended to accommodate new IoT data sources for future GNSS science use cases.

### Acknowledgments

The European Space Agency and IGS are acknowledged for making the GNSS products and data available to all interested parties. We thank CAS in Wuhan for making the daily BSX files and GIM publicly available. We are grateful to Yara Rossi for collecting and curating GNSS observations collected at the Prime Tower, a small subset of that data was used in this study. We would like to acknowledge all registered users of *CamalioTApp* for their active participation throughout the course of two CAMALIOT crowdsourcing campaigns. Software Routines from the IAU SOFA Collection were used as a part of CamalioTGNSS. Copyright © International Astronomical Union Standards of Fundamental Astronomy (<http://www.iausofa.org>).

Data, results, and supplementary material concerning the presented study available via the related data repository: <https://www.research-collection.ethz.ch/handle/20.500.11850/630998>. The RINEX-3 files with observations from the IGS stations and utilized in this study are available at <https://cddis.nasa.gov/archive/gnss/data/daily/>. The BSX files with satellite and station biases available via <https://cddis.nasa.gov/archive/gnss/products/bias/>. GIM used in this study are available via <https://cddis.nasa.gov/archive/gnss/products/ionex/>. The ESA precise orbit and clock products available via <http://navigation-office.esa.int/products/gnss-products/>. Due to the quantity and size of the crowdsourced data collected throughout the course of the project, the crowdsourced GNSS observations are not publicly available yet. Once the data migration process has been devised and applied, the data should be publicly available in the future via ESA's GSSC portal (<https://gssc.esa.int/portal/>). For the GNSS community interested in using the crowdsourced observations for scientific research purposes, we advise to contact us directly.

This work has been carried out as part of the project (NAVISP-EL1-038.2) entitled *CAMALIOT: Application of Machine Learning Technology for GNSS IoT Data Fusion*, funded by the ESA NAVISP Element 1 Program.

Appendix A. GNSS Position Time Series

This section contains a summary of the results concerning coordinates derived with the use of CamaliotGNSS that was utilized to analyze GNSS observations collected by selected IGS stations and Xiaomi Mi 8, with the latter employed during two local and controlled measurement sessions.

Time series of the estimated positions in the north (N), east (E), and up (U) components (and their formal errors) at ZIM200CHE and ZIMM00CHE derived from baseline processing (carrier-phase ambiguity parameters treated as float) at L1/E1 are shown in Fig. A1 for a 24-h period from the year 2023. The related parameterization is given in Table 1. In both cases, WAB200CHE was used as a reference station, which led to utilization of two approx. 5.2-km baselines in the analysis. For both stations, the position change with respect to the position derived at the initial epoch is characterized with the sample standard deviation of below 0.36 mm when taking into account all three position components.

Time series of the position estimates (carrier-phase ambiguity parameters treated as float) of Xiaomi Mi 8 and ETH200CHE obtained as a result of the single-frequency baseline processing (see Table 1) are shown in Fig. A2. Smartphone observations were collected during two dedicated local measurement sessions. The base station

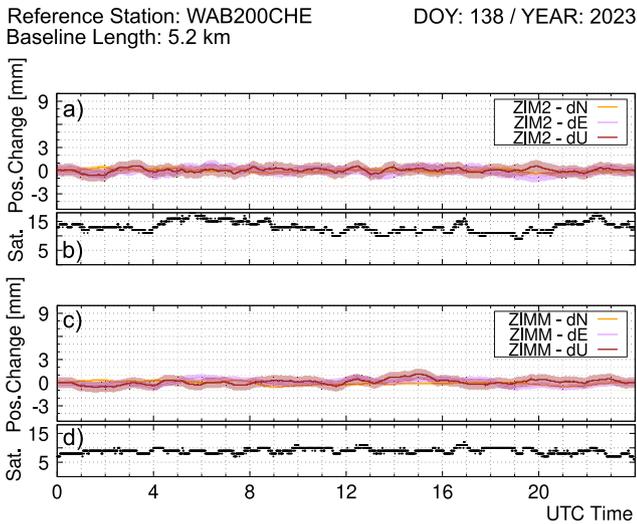


Fig. A1. Baseline processing results (float carrier-phase ambiguity parameters) for the ZIM200CHE-WAB200CHE and ZIMM00CHE-WAB200CHE baselines over the period of 24 h for May 18th, 2023. **a)** time series of position estimates for ZIM200CHE in north (N), east (E), and up (U) components expressed with respect to the estimates at the initial epoch, **b)** the number of GPS and Galileo satellites available and utilized for each epoch for the ZIM200CHE-WAB200CHE baseline, **c)** time series of position estimates for ZIMM00CHE in N, E, and U components expressed with respect to the estimates at the initial epoch, **d)** the number of GPS satellites available and utilized for each epoch for the ZIMM00CHE-WAB200CHE baseline. ZIMM00CHE does not support Galileo observations. ZIM200CHE and ZIMM00CHE abbreviated as ZIM2 and ZIMM, respectively.

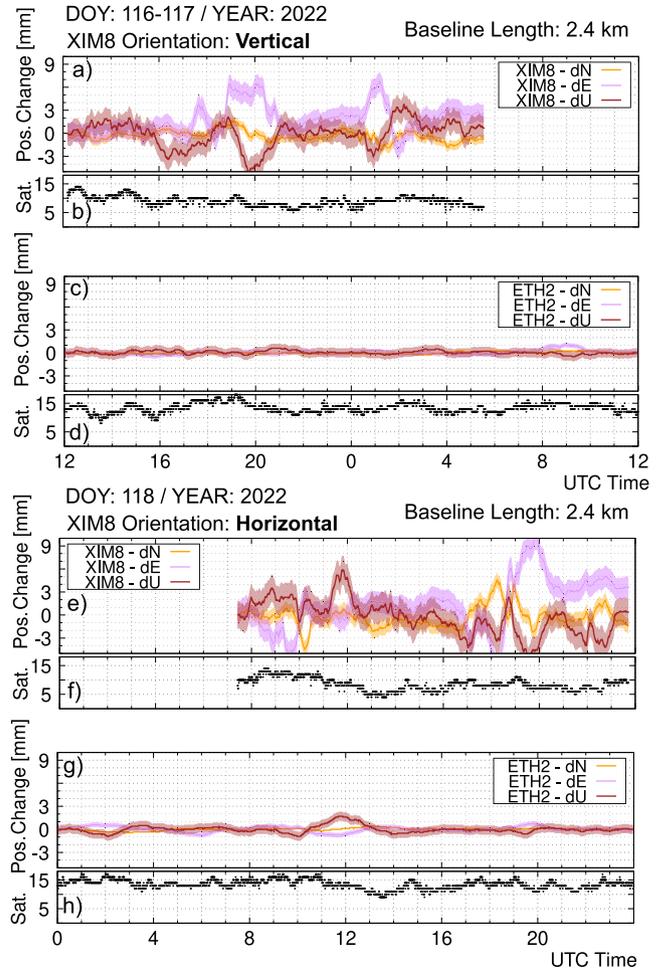


Fig. A2. Baseline processing results (carrier-phase ambiguity parameters treated as float) concerning the XIM8-PRTW and ETH200CHE-PRTW baselines for two different observing sessions (**a-d)** day 116–117, year 2022; **e-h)** day 118, year 2022). **a, e)** time series of position estimates for XIM8 in north (N), east (E), and up (U) components expressed with respect to the estimates at the initial epoch, **b, f)** the number of GPS and Galileo satellites available and utilized for each epoch for the XIM8-PRTW baseline, **c, g)** time series of position estimates for ETH200CHE in N, E, and U components expressed with respect to the estimates at an initial epoch, **d, h)** the number of GPS and Galileo satellites available and utilized for each epoch for the ETH200CHE-PRTW baseline. ETH200CHE abbreviated as ETH2.

(Septentrio PolaRx5 with the JAVAD GrAnt G3T antenna) was located on top of the Prime Tower in the city of Zürich, which was approx. 2.4 km from the location of both ETH200CHE and the employed Xiaomi Mi 8. The latter is abbreviated as XIM8.

For the two considered periods, the change in the estimated position components for XIM8 varies significantly compared to the same quantity derived for ETH200CHE. For the first period, the sample standard deviation of the change in the N, E, and U components with respect to the first epoch amount to 0.68 mm, 1.87 mm and 1.68 mm, respectively. Regarding the second observation period, where the phone was placed horizontally, the position change with respect to the first epoch is characterized

with the sample standard deviation of 1.31 mm, 3.18 mm, and 2.21 mm in the N, E, and U components, respectively. For ETH200CHE, the same metrics for three position components are at large below 0.28 mm and 0.47 mm for the first and the second considered period, respectively.

## References

- Altamimi, Z., Rebischung, P., Collilieux, X., et al., 2023. ITRF2020: an augmented reference frame refining the modeling of nonlinear station motions. *J. Geodesy* 97, 47. <https://doi.org/10.1007/s00190-023-01738-w>.
- Arikan, F., Nayir, H., Sezen, U., et al., 2008. Estimation of single station interfrequency receiver bias using GPS-TEC. *Radio Sci.* 43, RS4004. <https://doi.org/10.1029/2007RS003785>.
- Benevides, P., Catalao, J., Nico, G., 2019. Neural Network Approach to Forecast Hourly Intense Rainfall Using GNSS Precipitable Water Vapor and Meteorological Sensors. *Remote Sensing* 11 (8). <https://doi.org/10.3390/rs11080966>.
- Bennitt, G.V., Jupp, A., 2012. Operational Assimilation of GPS Zenith Total Delay Observations into the Met Office Numerical Weather Prediction Models. *Mon. Weather Rev.* 140 (8), 2706–2719. <https://doi.org/10.1175/MWR-D-11-00156.1>.
- Bertiger, W., Bar-Sever, Y., Dorsey, A., et al., 2020. GipsyX/RTGx, a new tool set for space geodetic operations and research. *Adv. Space Res.* 66 (3), 469–489. <https://doi.org/10.1016/j.asr.2020.04.015>.
- Beutler, G., Villiger, A., Dach, R., et al., 2020. Long polar motion series: Facts and insights. *Adv. Space Res.* 66 (11), 2487–2515. <https://doi.org/10.1016/j.asr.2020.08.033>.
- Bevis, M., Businger, S., Chiswell, S. et al., 1994. GPS Meteorology: Mapping Zenith Wet Delays onto Precipitable Water. *Journal of Applied Meteorology* (1988–2005), 33(3), 379–386. doi: 10.1175/1520-0450(1994)033<0379:GMMZWD>2.0.CO;2.
- Bevis, M., Businger, S., Herring, T.A., et al., 1992. GPS meteorology: Remote sensing of atmospheric water vapor using the global positioning system. *J. Geophys. Res.: Atmos.* 97 (D14), 15787–15801. <https://doi.org/10.1029/92JD01517>.
- Bosser, P., Bock, O., Flamant, C., et al., 2021. Integrated water vapour content retrievals from ship-borne GNSS receivers during EUREC<sup>4</sup>A. *Earth Syst. Sci. Data* 13 (4), 1499–1517. <https://doi.org/10.5194/essd-13-1499-2021>.
- Cesaroni, C., Spogli, L., Aragon-Angel, A., et al., 2020. Neural network based model for global Total Electron Content forecasting. *J. Space Weather Space Clim.* 10, 11. <https://doi.org/10.1051/swsc/2020013>.
- Chen, C., Xiao, G., Chang, G., et al., 2021. Assessment of GPS/Galileo/BDS Precise Point Positioning with Ambiguity Resolution Using Products from Different Analysis Centers. *Remote Sensing* 13 (16). <https://doi.org/10.3390/rs13163266>.
- Ciraolo, L., Azpilicueta, F., Brunini, C., et al., 2007. Calibration errors on experimental slant total electron content (TEC) determined with GPS. *J. Geodesy* 81, 111–120. <https://doi.org/10.1007/s00190-006-0093-1>.
- Crocetti, L., Soja, B., Kłopotek, G. et al., 2022. Machine learning and meteorological data for spatio-temporal prediction of tropospheric parameters. *EGU General Assembly 2022* (pp. EGU22–4531). Vienna, Austria, 23–27 May 2023. doi:10.5194/egusphere-egu23-9260.
- Dach, R., Andritsch, F., Arnold, D. et al., 2015. Bernese GNSS Software Version 5.2. doi:10.7892/boris.72297.
- Darugna, F., Wübbena, J.B., Wübbena, G., et al., 2020. Impact of robot antenna calibration on dual-frequency smartphone-based high-accuracy positioning: a case study using the Huawei Mate20X. *GPS Solutions* 25, 15. <https://doi.org/10.1007/s10291-020-01048-0>.
- Davies, K., Hartmann, G.K., 1997. Studying the ionosphere with the Global Positioning System. *Radio Science* 32 (4), 1695–1703. <https://doi.org/10.1029/97RS00451>.
- Deng, Z., Bender, M., Dick, G., et al., 2009. Retrieving tropospheric delays from GPS networks densified with single frequency receivers. *Geophys. Res. Lett.* 36, L19802. <https://doi.org/10.1029/2009GL040018>.
- Dong, D., Wang, M., Chen, W., et al., 2016. Mitigation of multipath effect in GNSS short baseline positioning by the multipath hemispherical map. *GPS Solutions* 90, 255–262. <https://doi.org/10.1007/s00190-015-0870-9>.
- Fermi, A., Realini, E., Venuti, G., 2019. The impact of relative and absolute GNSS positioning strategies on estimated coordinates and ZWD in the framework of meteorological applications. *Ann. Geophys.* 11, 25–38. <https://doi.org/10.1007/s12518-018-0234-2>.
- Flores, A., Ruffini, G., Rius, A., 2000. 4D tropospheric tomography using GPS slant wet delays. *Ann. Geophys.* 18 (2), 223–234. <https://doi.org/10.1007/s00585-000-0223-7>.
- Glaner, M., Weber, R., 2021. PPP with integer ambiguity resolution for GPS and Galileo using satellite products from different analysis centers. *GPS Solutions* 25, 102. <https://doi.org/10.1007/s10291-021-01140-z>.
- Guerova, G., Jones, J., Douša, J., et al., 2016. Review of the state of the art and future prospects of the ground-based GNSS meteorology in Europe. *Atmos. Meas. Tech.* 9 (11), 5385–5406. <https://doi.org/10.5194/amt-9-5385-2016>.
- Hammond, W.C., Blewitt, G., Kreemer, C., 2016. GPS Imaging of vertical land motion in California and Nevada: Implications for Sierra Nevada uplift. *J. Geophys. Res.: Solid Earth* 121 (10), 7681–7703. <https://doi.org/10.1002/2016JB013458>.
- Hernández-Pajares, M., Juan, J.M., Sanz, J., et al., 2008. The IGS VTEC maps: A reliable source of ionospheric information since 1998. *J. Geodesy* 83, 263–275. <https://doi.org/10.1007/s00190-008-0266-1>.
- Hu, Y., Yao, Y., 2019. A new method for vertical stratification of zenith tropospheric delay. *Adv. Space Res.* 63 (9), 2857–2866. <https://doi.org/10.1016/j.asr.2018.10.035>.
- Kaselim, M., Voulodimos, A., Doulamis, N., et al., 2020. A Causal Long Short-Term Memory Sequence to Sequence Model for TEC Prediction Using GNSS Observations. *Remote Sensing* 12 (9). <https://doi.org/10.3390/rs12091354>.
- Kouba, J., Héroux, P., 2001. Precise Point Positioning Using IGS Orbit and Clock Products. *GPS Solutions* 5, 12–28. <https://doi.org/10.1007/PL00012883>.
- Li, B., Miao, W., Chen, G., et al., 2022a. Ambiguity resolution for smartphone GNSS precise positioning: effect factors and performance. *J. Geodesy* 96, 63. <https://doi.org/10.1007/s00190-022-01652-7>.
- Li, G., Geng, J., 2019. Characteristics of raw multi-GNSS measurement error from Google Android smart devices. *GPS Solutions* 23, 42. <https://doi.org/10.1007/s10291-019-0885-4>.
- Li, X., Huang, J., Li, X., et al., 2022b. Review of PPP–RTK: achievements, challenges, and opportunities. *Satellite Navigation* 3, 28. <https://doi.org/10.1186/s43020-022-00089-9>.
- Li, X., Wang, H., Li, X., et al., 2022c. PPP rapid ambiguity resolution using Android GNSS raw measurements with a low-cost helical antenna. *J. Geodesy* 96, 65. <https://doi.org/10.1007/s00190-022-01661-6>.
- Mao, S., Kłopotek, G., Awadaljeed, M. et al., 2023. Machine learning for global modeling of the ionosphere based on multi-GNSS data. *EGU General Assembly 2023* (pp. EGU23–9260). Vienna, Austria, 24–28 Apr 2023. doi:10.5194/egusphere-egu23-9260.
- Navarro, V., Grieco, R., Soja, B. et al., 2021. Data Fusion and Machine Learning for Innovative GNSS Science Use Cases. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)* (pp. 2656–2669). The Institute of Navigation. doi:10.33012/2021.18115.
- Ning, T., Elgered, G., Willén, U., et al., 2013. Evaluation of the atmospheric water vapor content in a regional climate model using ground-based GPS measurements. *J. Geophys. Res.: Atmos.* 118 (2), 329–339. <https://doi.org/10.1029/2012JD018053>.

- Odolinski, R., Teunissen, P.J.G., 2019. An assessment of smartphone and low-cost multi-GNSS single-frequency RTK positioning for low, medium and high ionospheric disturbance periods. *J. Geodesy* 93, 701–722. <https://doi.org/10.1007/s00190-018-1192-5>.
- Otsuka, Y., Ogawa, T., Saito, A., et al., 2000. A new technique for mapping of total electron content using GPS network in Japan. *Earth, Planets and Space* 54, 63–70. <https://doi.org/10.1186/BF03352422>.
- Paziewski, J., 2020. Recent advances and perspectives for positioning and applications with smartphone GNSS observations. *Meas. Sci. Technol.* 31 (9), 091001. <https://doi.org/10.1088/1361-6501/ab8a7d>.
- Paziewski, J., Fortunato, M., Mazzoni, A., et al., 2021. An analysis of multi-GNSS observations tracked by recent Android smartphones and smartphone-only relative positioning results. *Measurement* 175, 109162. <https://doi.org/10.1016/j.measurement.2021.109162>.
- Poluzzi, L., Tavasci, L., Corsini, F., et al., 2020. Low-cost GNSS sensors for monitoring applications. *Appl. Geomat.* 12, 35–44. <https://doi.org/10.1007/s12518-019-00268-5>.
- See, L., Soja, B., Kłopotek, G., et al., 2023. Collecting volunteered geographic information from the Global Navigation Satellite System (GNSS): experiences from the CAMALIOT project. *International Journal of Digital Earth* 16 (1), 2818–2841. <https://doi.org/10.1080/17538947.2023.2239761>.
- Soja, B., Kłopotek, G., Pan, Y., et al., 2023. Machine Learning-Based Exploitation of Crowdsourced GNSS Data for Atmospheric Studies. In: *IGARSS 2023–2023 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1170–1173. <https://doi.org/10.1109/IGARSS52108.2023.10283441>.
- Stauffer, R., Hohensinn, R., Herrera-Pinzón, I.D., et al., 2023. Estimation of tropospheric parameters with GNSS smartphones in a differential approach. *Meas. Sci. Technol.* 34 (9), 095126. <https://doi.org/10.1088/1361-6501/acd077>.
- Stępiak, K., Paziewski, J., 2022. On the quality of tropospheric estimates from low-cost GNSS receiver data processing. *Measurement* 198, 111350. <https://doi.org/10.1016/j.measurement.2022.111350>.
- Strasser, S., Mayer-Gürr, T., Zehentner, N., 2019. Processing of GNSS constellations and ground station networks using the raw observation approach. *J. Geodesy* 93, 1045–1057. <https://doi.org/10.1007/s00190-018-1223-2>.
- Takahashi, H., Wrasse, C.M., Denardini, C.M., et al., 2016. Ionospheric TEC Weather Map Over South America. *Space Weather* 14 (11), 937–949. <https://doi.org/10.1002/2016SW001474>.
- Takasu, T., & Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In *International Symposium on GPS/GNSS*. International Convention Center Jeju, Korea. URL: [https://gpspp.sakura.ne.jp/paper2005/isgps\\_2009\\_rtklib.pdf](https://gpspp.sakura.ne.jp/paper2005/isgps_2009_rtklib.pdf).
- Teunissen, P., De Jonge, P., Tiberius, C., 1997. Performance of the LAMBDA Method for Fast GPS Ambiguity Resolution. *NAVIGATION* 44 (3), 373–383. <https://doi.org/10.1002/j.2161-4296.1997.tb02355.x>.
- Teunissen, P.J.G., Khodabandeh, A., 2015. Review and principles of PPP-RTK methods. *J. Geodesy* 89, 217–240. <https://doi.org/10.1007/s00190-014-0771-3>.
- Uppala, S.M., Källberg, P.W., Simmons, A.J., et al., 2005. The ERA-40 re-analysis. *Quart. J. Roy. Meteorol. Soc.* 131 (612), 2961–3012. <https://doi.org/10.1256/qj.04.176>.
- Van Malderen, R., Brenot, H., Pottiaux, E., et al., 2014. A multi-site intercomparison of integrated water vapour observations for climate change analysis. *Atmos. Meas. Techniques* 7 (8), 2487–2512. <https://doi.org/10.5194/amt-7-2487-2014>.
- Wang, J., Huang, G., Zhou, P., et al., 2020. Advantages of Uncombined Precise Point Positioning with Fixed Ambiguity Resolution for Slant Total Electron Content (STEC) and Differential Code Bias (DCB) Estimation. *Remote Sensing* 12 (2). <https://doi.org/10.3390/rs12020304>.
- Wang, N., Yuan, Y., Li, Z., et al., 2016. Determination of differential code biases with multi-GNSS observations. *J. Geodesy* 90, 209–228. <https://doi.org/10.1007/s00190-015-0867-4>.
- Webb, S.R., Penna, N.T., Clarke, P.J., et al., 2016. Kinematic GNSS Estimation of Zenith Wet Delay over a Range of Altitudes. *J. Atmos. Ocean. Technol.* 33 (1), 3–15. <https://doi.org/10.1175/JTECH-D-14-00111.1>.
- Wilgan, K., Geiger, A., 2018. High-resolution models of tropospheric delays and refractivity based on GNSS and numerical weather prediction data for alpine regions in Switzerland. *J. Geodesy* 93, 819–836. <https://doi.org/10.1007/s00190-018-1203-6>.
- Xiang, Y., Gao, Y., Shi, J., et al., 2019. Consistency and analysis of ionospheric observables obtained from three precise point positioning models. *J. Geodesy* 93, 1161–1170. <https://doi.org/10.1007/s00190-019-01233-1>.
- Xu, L., Zha, J., Li, M., et al., 2022. Estimation of ionospheric total electron content using GNSS observations derived from a smartphone. *GPS Solutions* 26, 138. <https://doi.org/10.1007/s10291-022-01329-w>.
- Yong, C.Z., Odolinski, R., Zaminpardaz, S., et al., 2021. Instantaneous, Dual-Frequency, Multi-GNSS Precise RTK Positioning Using Google Pixel 4 and Samsung Galaxy S20 Smartphones for Zero and Short Baselines. *Sensors* 21 (24). <https://doi.org/10.3390/s21248318>.
- Yu, C., Penna, N.T., Li, Z., 2017. Generation of real-time mode high-resolution water vapor fields from GPS observations. *J. Geophys. Res.: Atmos.* 122 (3), 2008–2025. <https://doi.org/10.1002/2016JD025753>.
- Zhang, B., Ou, J., Yuan, Y., et al., 2012. Extraction of line-of-sight ionospheric observables from GPS data using precise point positioning. *Science China Earth Sciences* 55, 1919–1928. <https://doi.org/10.1007/s11430-012-4454-8>.
- Zhang, B., Teunissen, P.J.G., Yuan, Y., et al., 2018. Joint estimation of vertical total electron content (VTEC) and satellite differential code biases (SDCBs) using low-cost receivers. *J. Geodesy* 92, 401–413. <https://doi.org/10.1007/s00190-018-1192-5>.
- Zhang, B., Yao, Y., 2021. Precipitable water vapor fusion based on a generalized regression neural network. *J. Geodesy* 95, 36. <https://doi.org/10.1007/s00190-021-01482-z>.
- Zhang, X., Tao, X., Zhu, F., et al., 2020. Quality assessment of GNSS observations from an Android N smartphone and positioning performance analysis using time-differenced filtering approach. *GPS Solutions* 22, 11. <https://doi.org/10.1007/s10291-018-0736-8>.
- Zhao, C., Zhang, B., Li, W., et al., 2019. Simultaneous Retrieval of PWV and VTEC by Low-Cost Multi-GNSS Single-Frequency Receivers. *Earth and Space Science* 6 (9), 1694–1709. <https://doi.org/10.1029/2019EA000650>.
- Zumbege, J.F., Heflin, M.B., Jefferson, D.C., et al., 1997. Precise point positioning for the efficient and robust analysis of GPS data from large networks. *J. Geophys. Res.: Solid Earth* 102 (B3), 5005–5017. <https://doi.org/10.1029/96JB03860>.