navisp Element 1
Final Presentation

EL1-035 BIS "Machine Learning Techniques to model GNSS"

15th December 2022, 10:00 – 11:30 CEST

Time	Topic	Speaker
10:00-10:05	Welcome	S. Binda, ESA NAVISP Element 1 Manager
10:05-10:10	Project Introduction	H. Sobreira, Technical Officer, ESA
10:10-11:00	Project Implementation and Results	Rosario Messineo, Technical Leader, ALTEC Chiara Leuzzi, Data Science Specialist, ALTEC Fabio Dovis, Research Scientist POLITO, Andrea Nardin, Research Scientist POLITO.
11:00-11:30	Question and Answers	Moderator: H. Sobreira

Welcome

S. Binda

Project Introduction

H. Sobreira

Project Implementation and Results

- Project Implementation & GMLD Overview [5 min]
  Rosario Messineo
- Presentation and Key Points of the Selected Applications [10 min]
  Fabio Dovis
- Technical Aspects of the Applications & ML Models Functional Validation Overview [10 min]
  Chiara Leuzzi
- GNSS Performance and Overall Application Evaluation [15 min]
  Andrea Nardin
- GMLD Outcome Evaluation and Future Recommendations [5 min]
  Rosario Messineo

Project Implementation & GMLD Overview

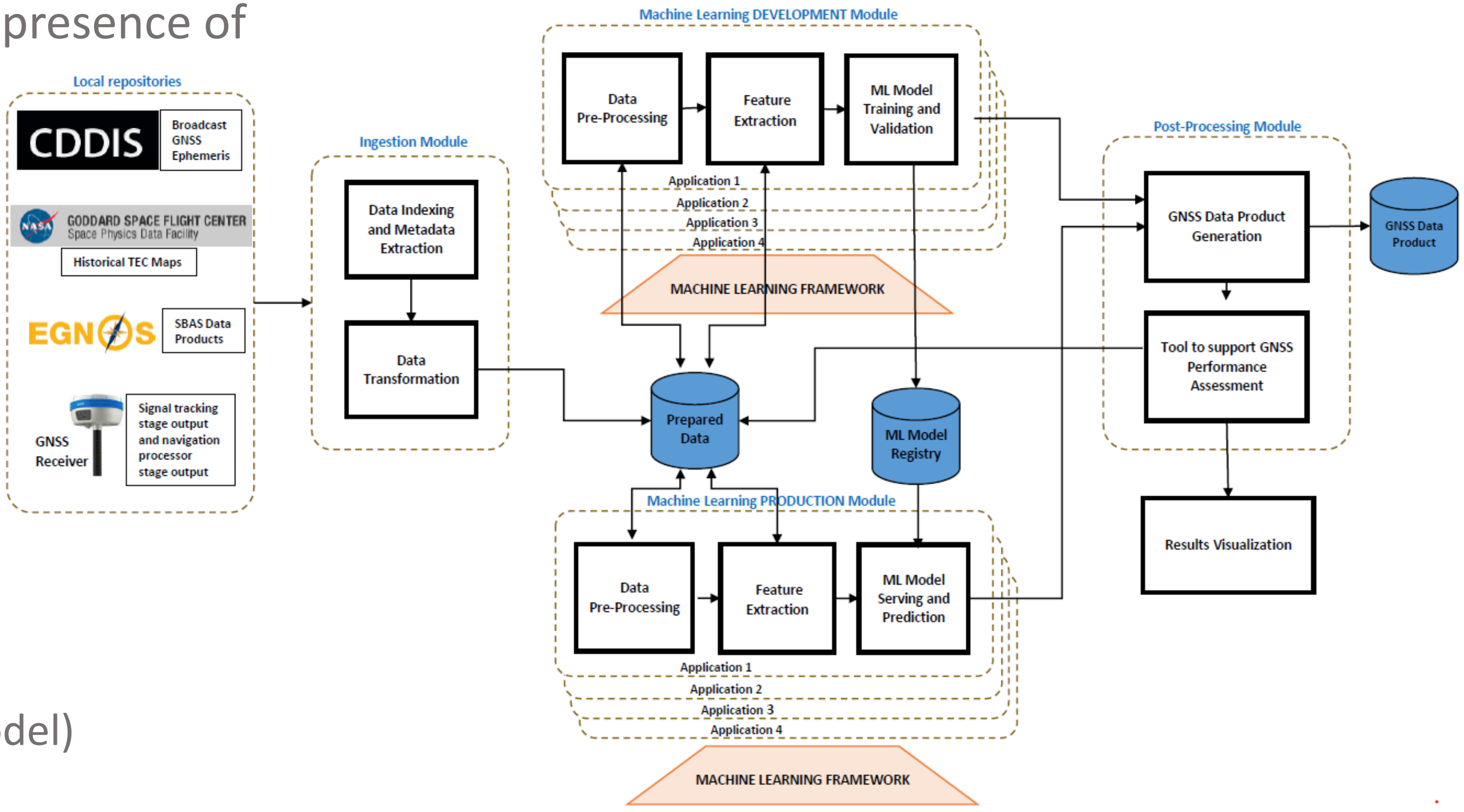
Rosario Messineo

Project Objectives

- The main objective of the activity is to assess the usability of machine learning (ML) techniques to improve availability and quality of GNSS navigation data.
- This can be achieved trying to reduce the impact of the different error sources, which could be mitigated by predicting some additional parameters, when missing, or by not considering data affected by the errors, when their presence is detected.
- The results of the activity provide:
  - demonstration of usability of ML to improve data availability and/or quality in some segments of a GNSS system;
  - implementation of selected ML techniques into tools to simulate GNSS system segments capabilities and behaviour;
  - implementation of a framework to conduct additional investigation on the usage of ML techniques in other use cases of the GNSS domain.

GMLD Overview

- The GNSS Machine Learning Demonstrator (GMLD) aims at **investigating and demonstrating the use of ML in various areas of GNSS domain**.
- It is composed by **4 applications that implement ML models** able to predict missing information or the presence of possible error sources
  - The applications have been identified in order to cover **a variety of diverse error sources**, as well as different data structures.
- The GMLD provides capabilities to implement the applications as composition of the following modules:
  - Data indexing and transformation
  - ML model (including pre-processing, feature extraction and machine learning model)
  - Data output transformation



Project Team

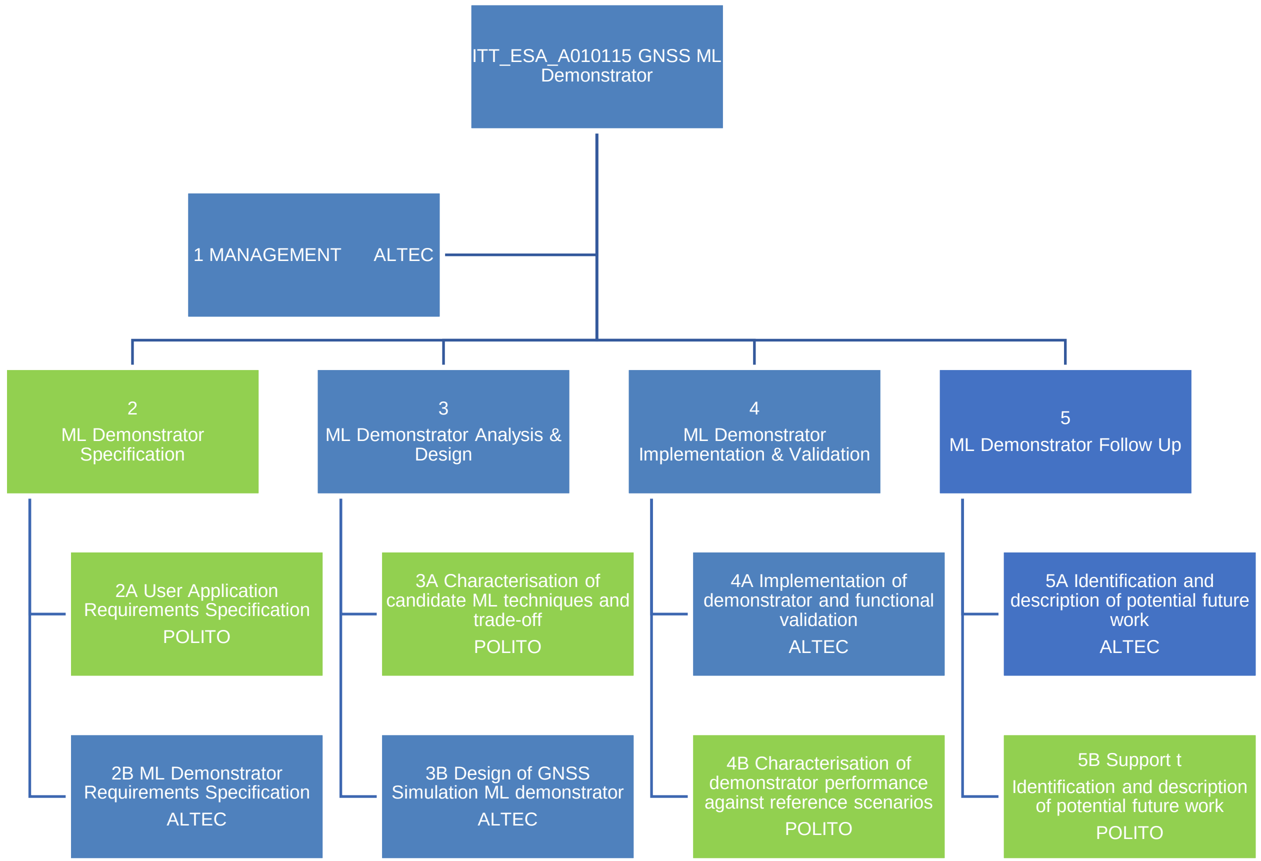
- The team was composed of the following companies:
  - Aerospace Technology Engineering Company (ALTEC), Torino Italy - PRIME
  - Politecnico di Torino (POLITO), Torino Italy – SUBCO-1

- As Prime, ALTEC coordinated the industrial team, relaying on its data processing capabilities maturated during the last ten years for complex space missions, like in particular the ESA Gaia astronomy satellite, and during several R&D projects. ALTEC covered the following roles:
  - Software Engineer to specify functional requirement of GNSS ML Demonstrator;
  - Data Scientist to design and validate ML models of proposed applications;
  - Software Engineer to implement and validate the integrated GNSS ML Demonstrator;

- The team at Politecnico di Torino developed in-house concepts for the processing of GNSS signals via ML techniques, and basic software tool for the assessment of the performance. Their previous researches were the initial knowledge used to select applications and address their implementation and validation within the project. Politecnico di Torino covered the following roles:
  - GNSS domain experts to define and assess performance of application as well as contribute to address future activities;
  - Machine and deep learning algorithms expert to address ML class model candidate selection.

Project Implementation

- Project Schedule
  - Project activities officially started with the kick-off meeting held on 01/12/2020.
  - Key Point Meeting 1 was held on 07/06/2021.
  - MTR was held on 28/10/2021.
  - Key Point Meeting 2 was held on 16/05/2022.
  - FR was held on 07/11/2022.
- Project Phases
  - Definition Phase focused on the application definition consolidation, SW system specification and trade-off execution for candidate ML models class
  - Design Phase aimed at providing SW system and ML application detailed design and identifying the deployment schema
  - Implementation Phase was dedicated to complete the ML application development lifecycle, implement the SW system and perform the final integration of the applications into the SW framework. Moreover, the GMLD final validation campaign was prepared.
  - Acceptance Phase allowed aimed at generating the final version of the applications and execute the final application validation and GNSS performance assessment.

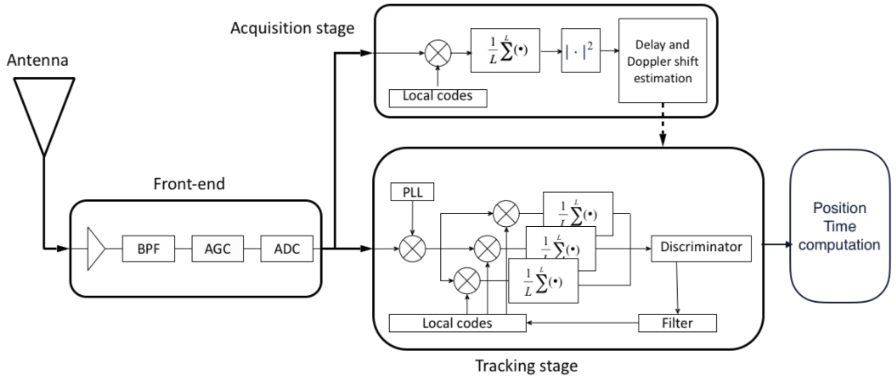


Presentation and Key Points of the Selected Applications

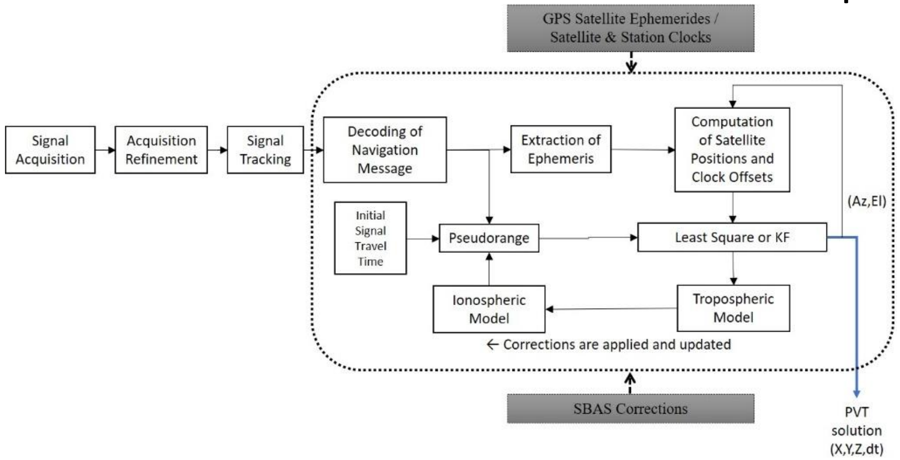
Fabio Dovis

GNSS Applications

- GNSS ML Demonstrator is intended as a set of GNSS ML applications.
- GNSS ML Demonstrator shall include both system prediction models to predict parameters and corrections that could contribute to improve GNSS **navigation algorithm behavior** and additional models needed to improve the **quality of data used by navigation algorithms**.



Generic GNSS Receiver Architecture



GNSS Receiver Detailed Position Computation Diagram

- **Four use-cases** have been chosen in order to cover a variety of diverse **applications**, as well of different data structures.

Identified GNSS Applications

01 **Improving Orbit Prediction by means of Machine Learning Approaches**

02 **Prediction of daily maps of the ionosphere**

03 **Estimation of the SBAS correction parameters in the missed messages**

04 **Disturbances classifications – Outlier Detection**

App 1 - Improving Orbit Prediction by means of ML approaches

Objective:

use of ML approaches to **improve satellite orbit prediction accuracy**.

- In case of <u>missed updated ephemeris</u>, the aims of this application are
  - forecasting the ephemeris parameters relying on the historical data by means of ML algorithms
  - improving the computed satellite positions in a stand-alone receiver as a complement to the approximations generated by classical theory

- Selected algorithm:

  Long short-term memory *Recurrent Neural Network*

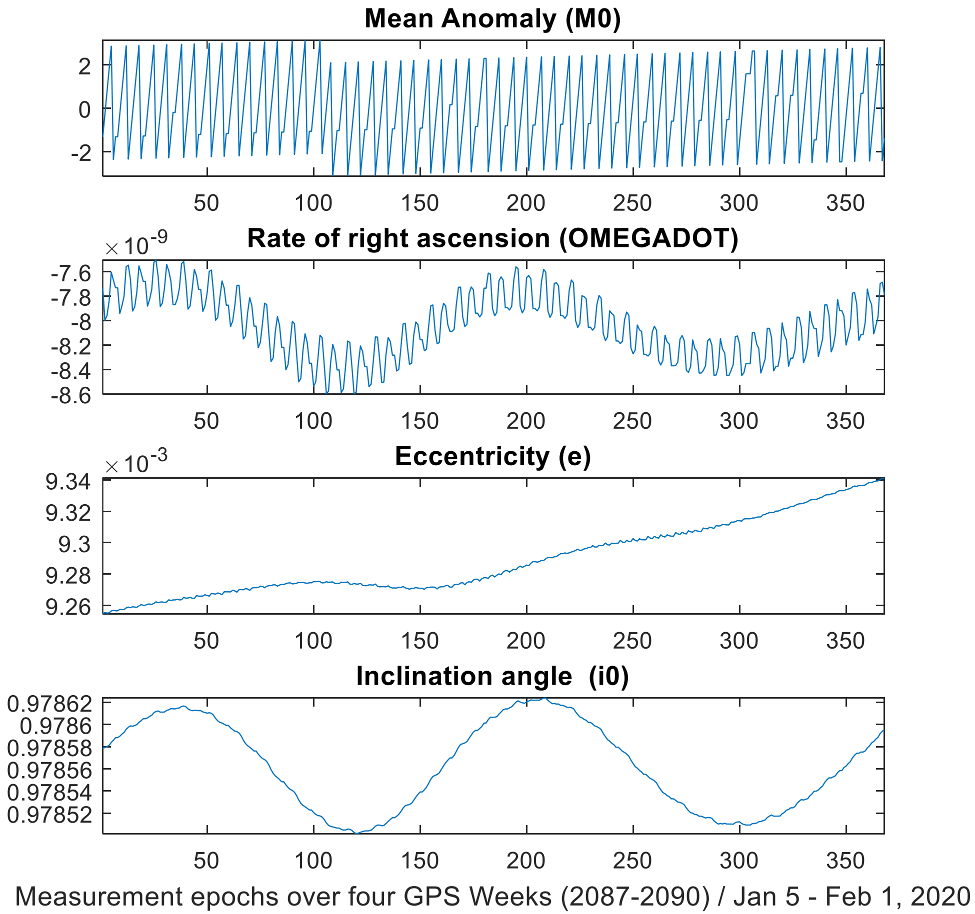


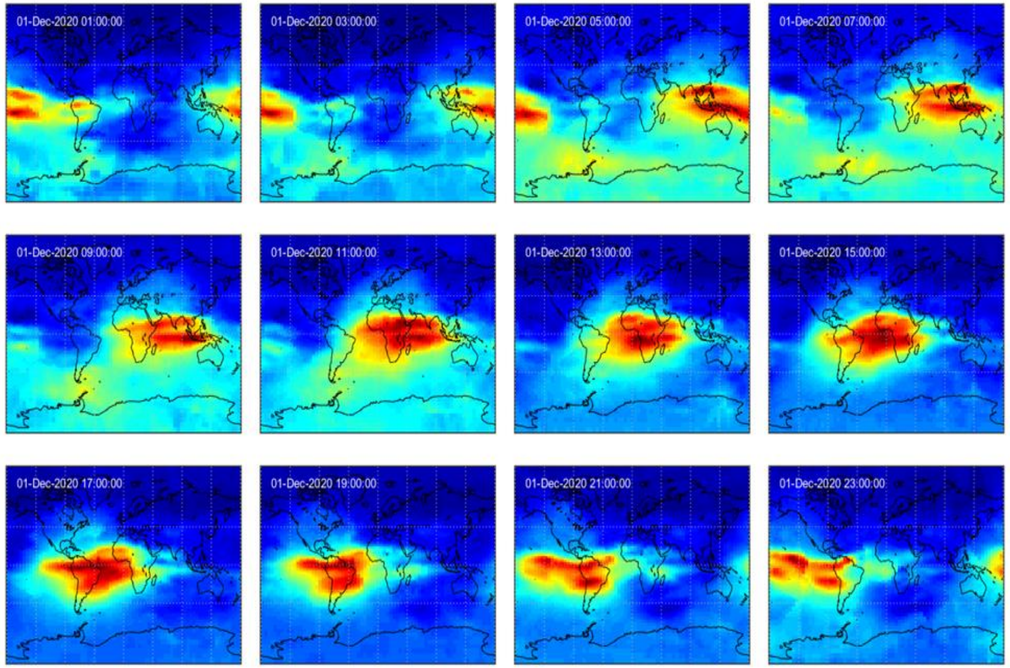
Fig. Evolution of some ephemeris parameters over 4 weeks.

App 2 - Prediction of daily maps of the ionosphere

Objective:

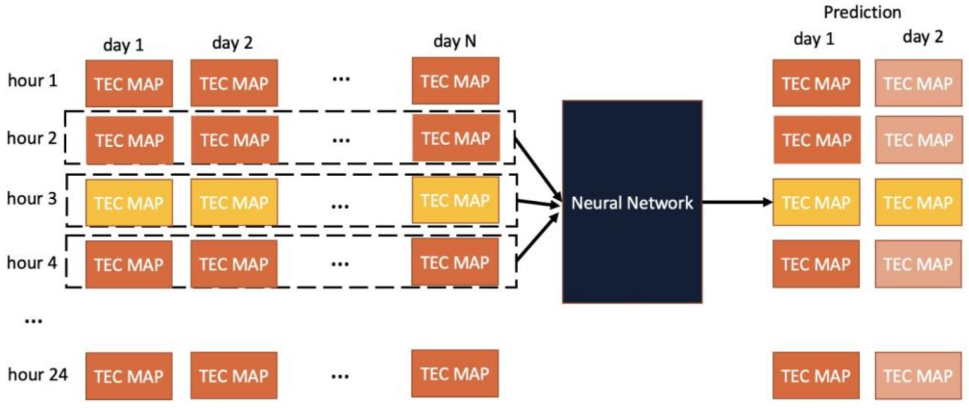
predict **global TEC map** sequences given the previous states of the ionosphere

- When looking at the temporal evolution of TEC maps, they represent a time series of **image frames**, and the current status maps can be inferred from the ionosphere previous states.
- The **performance assessment** has been performed using **historical time series of TEC maps** to assess the quality of the prediction with respect to measured data maps.



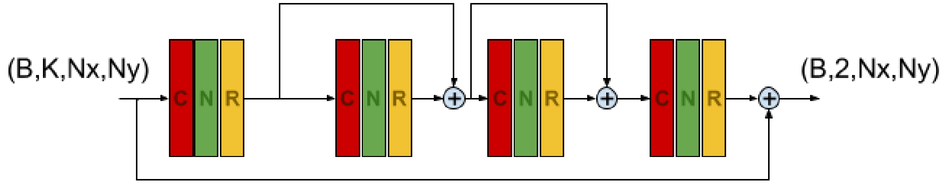
App 2 - Prediction of daily maps of the ionosphere

- The first model concatenates TEC maps from the same hour-of-the-day as time T from the last K days, resulting in an input image with K channels.
- Due to Earth's rotation, the highest temporal correlation between TEC maps can be found between maps that are 24 hours apart.
- The model considers K maps each 24 hours apart to capture long-term trends in the evolution of TEC



- The second model concatenates all the available past TEC maps from T to T-K hours, creating an input image with K channels.
- Differently from the first method, even maps at hours-of-the-day that are different from the target one are exploited. The reason for this choice is that it allows better exploitation of short-term spatial and temporal correlation patterns thanks to the smooth evolution of TEC over temporally subsequent maps.

Better for short term prediction



App 3 - Estimation of the SBAS correction parameters in the missed messages

esa

Objective:

to predict the parameters in the missing SBAS messages by utilizing ML algorithms to fill the gaps in the information flow.

- Time to time, the user could **miss** some SBAS messages broadcasted. The use of the old data yields **an increased error**.
- The main target users for this application are **non-Safety-of-Life (SoL)** users of wide area differential corrections (e.g. **precision farming**.)
- Fast and long-term corrections and their impact on the **pseudorange correction** are of particular interest since they are specific of the SBAS messages content.

Message Type (MT)		Content
Satellite Information Messages	1	PRN mask assignments
	2-5	Fast corrections
	6	Integrity information
	7	Fast correction degradation factor
	9	Geo navigation message (X, Y, Z, time, etc.)
	17	Geo satellite almanacs
	24	Mixed fast corrections / long term satellite corrections
	25	Long term satellite error corrections
	28	Clock ephemeris covariance matrix message
Ionospheric Information Messages	18	Ionospheric grid point masks
	26	Ionospheric delay corrections

App. 4 - Disturbances classifications – Outlier Detection

Objective:

Detect the existence of **outliers** due to scintillation and multipath events at the early stages, by processing the post correlation outputs and observables/measurements of a GNSS receiver.

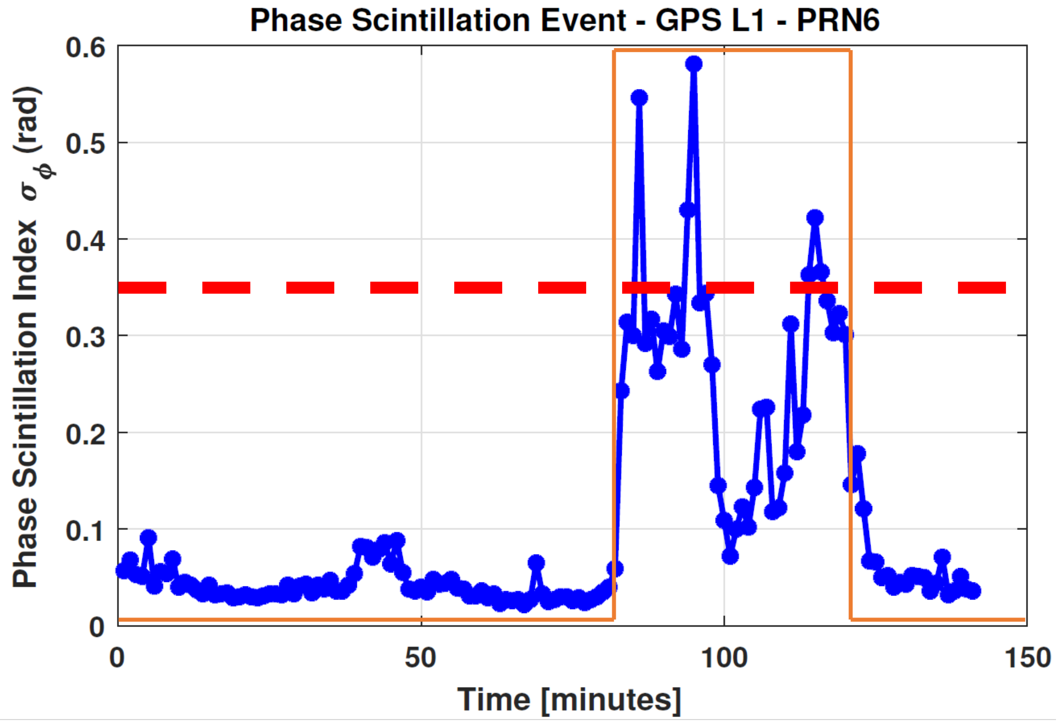
APP 4a – Detection of scintillation events

APP 4b – Detection of multipath events

- Input for the algorithm are raw data (output of tracking and correlation stages obtained by processing datasets of raw samples using Polito SDR receiver
- Labelling of the data is based on “manual annotation” of the data, validated by experience (e.g. repeatability of multipath event on GPS signals for a static user) and other sources of information who detected scintillation at a given location/time.

App 4a – scintillations detection

- Use of Machine Learning algorithms for automatic detection of scintillation events
  - More effective than classical detection based on thresholding of amplitude and phase indexes of $S4$ and σ_ϕ
  - Resembles the human recognition of the event
  - Reduction of the probability of false alarm by automatic clustering of other disturbances (multipath)
  - Early detection



App 4b – multipath detection

- Multipath detection has been based on clustering techniques in the space of metrics obtained from raw PRNs obseravtions

- The clusters are labelled as multipath/non multipath based on the average C/N0

Technical Aspects of the Applications & ML Models Functional Validation Overview

Chiara Leuzzi

Application 1 - Improving Orbit Prediction by means of Machine Learning Approaches

ML Models and Functional Validation

App1 – ML Models

- Dataset is composed by GPS data
  - Data rate: 7200 seconds
  - 2019 and 2020 years
- Regression task: time series of past values are used to predict future values (+2H, +4H, +6H, +8H)
  - 24 hours of past observations
- Univariate ML models: a model for each ephemeris parameter
  - LSTM based models
  - Similar architectures with different number of layes or hyperparameters values
  - Some parameters benefit from the data normalization
  - Learning strategies:
    - Test different splitting strategy (stratified, randomic, temporal split)
    - Test different loss function → MSE Improves learning process
    - Early Stopping with difference Patience values (w.r.t. MAE)
    - Save best model while training (w.r.t. MAE)

App1 – Functional Validation for the Final Models

- Final models:
  - Model architectures have 1 or 2 LSTM layers (Vanilla model or 2-stacked LSTM model) → More deep models showed overfit behaviours.
  - The number of the neurons inside each LSTM layers can vary among 30 and 70 units.
  - Only in few cases, the application of the Rectified Linear Unit as activation function improve values of functional validation metrics.
  - As expected, normalization implementation performs better when the absolute values of the parameter to be predicted is small.
- Results:
  - Test sets confirm validation results
  - Errors increase when more steps forward are predicted

[dataset_metric_step]	M0 [semi-circles]	e	delta_n [semi-circles/s]	sqrtA [√m]	OMEGA [semi-circles]	i0 [semi-circles]
train_mae_02H	2.91E-02	5.67E-07	4.85E-12	1.45E-03	6.08E-03	1.85E-06
train_mae_04H	2.93E-02	5.31E-07	5.16E-12	1.53E-03	1.25E-02	2.35E-06
train_mae_06H	3.77E-02	5.73E-07	6.49E-12	1.66E-03	1.16E-02	2.57E-06
train_mae_08H	3.75E-02	7.31E-07	6.88E-12	1.48E-03	2.11E-02	3.85E-06
val_mae_02H	1.92E-02	5.53E-07	5.43E-12	1.50E-03	4.22E-03	1.84E-06
val_mae_04H	4.16E-02	5.23E-07	5.55E-12	1.54E-03	1.26E-02	2.41E-06
val_mae_06H	2.98E-02	5.65E-07	7.05E-12	1.70E-03	1.54E-02	2.67E-06
val_mae_08H	4.53E-02	7.05E-07	7.37E-12	1.47E-03	2.71E-02	3.90E-06
test_mae_02H	1.77E-02	5.72E-07	5.29E-12	1.40E-03	4.49E-03	1.86E-06
test_mae_04H	2.24E-02	5.50E-07	6.06E-12	1.59E-03	9.81E-03	2.40E-06
test_mae_06H	2.97E-02	5.64E-07	6.55E-12	1.57E-03	1.61E-02	2.59E-06
test_mae_08H	3.04E-02	7.03E-07	7.10E-12	1.48E-03	2.47E-02	3.81E-06

Functional validation results for six parameters

Application 2 - Prediction of daily maps of the ionosphere

ML Model and Functional Validation

App 2 - Dataset and ML Model

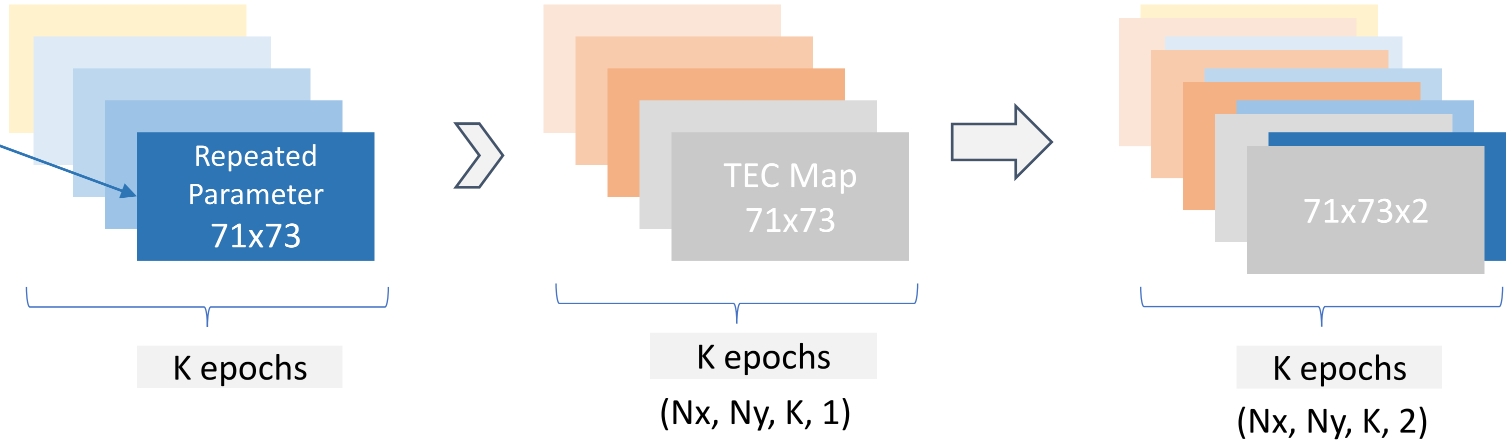
- Source data:
  - TEC (Total Electron Content) maps, data rate of 2 hours,
    - 2001-2004 (maximum of the solar cycle 23)
    - 2009-2010 (low solar activity)
  - Additional parameters were tested:
    - Sunspot number
    - Kp index
- Application Design:
  - ResNet based model with partial and global skip connection
  - 3D convolution (Nx, Ny, K), K number of previous time step in the time series
  - Number of features/channel depending on input dataset
    - (Nx, Ny, K, 1) when using only TEC maps
    - (Nx, Ny, K, 2+) when using TEC maps and one or more additional parameter

App 2 - Dataset and ML Model

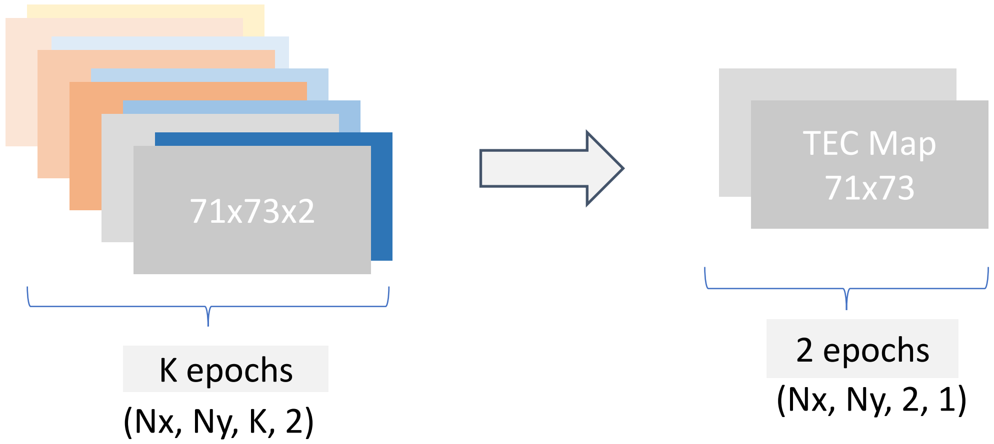
Tests with TEC + additional parameters

➢ Input structure

YEAR	DOY	Hour	Kp_index	R_SunspotNumber
2001	1	1	0	133
2001	1	3	3	133
2001	1	5	3	133
2001	1	7	10	133
2001	1	9	10	133



➢ Model



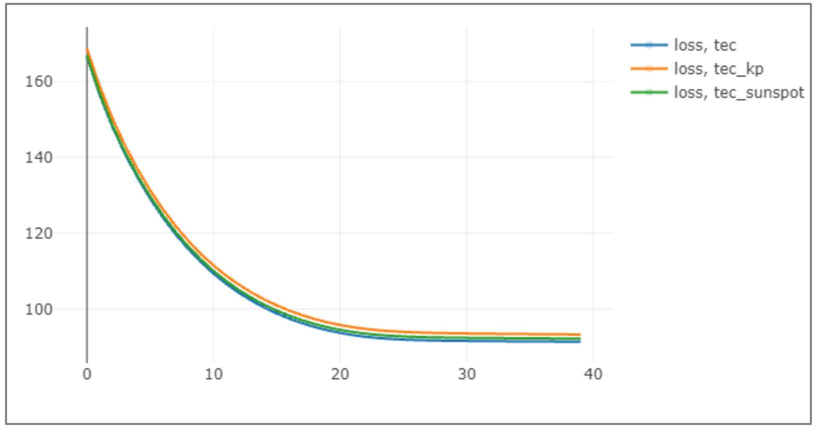
➢ Easy to Scale the Input Size, when considering more parameters (m)

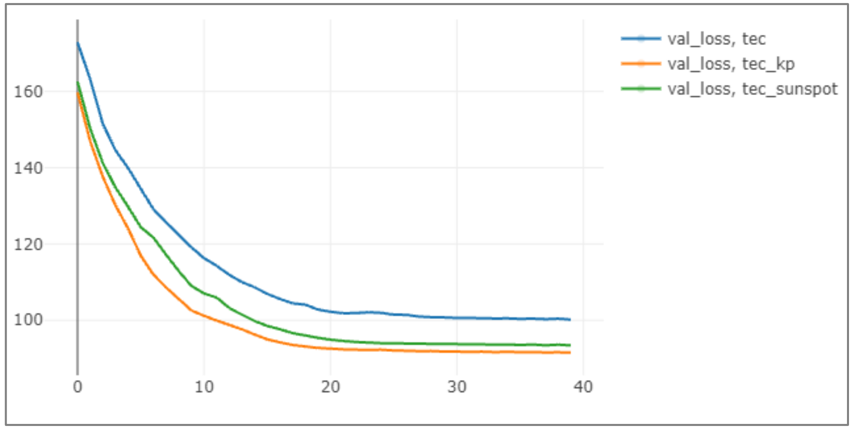
K epochs (Nx, Ny, K, 1+m) → 2 epochs (Nx, Ny, 2, 1)

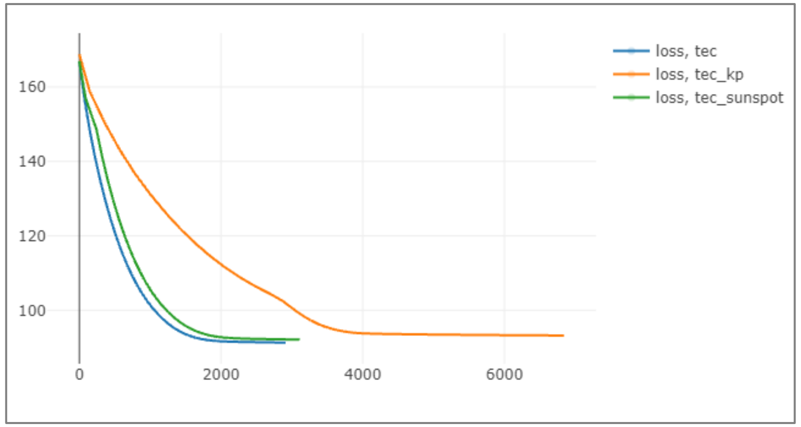
App2 – Model Tuning and Results

Comparison between TEC, TEC and sunspot, TEC and Kp index

- Same model configuration, apart the different number of channels

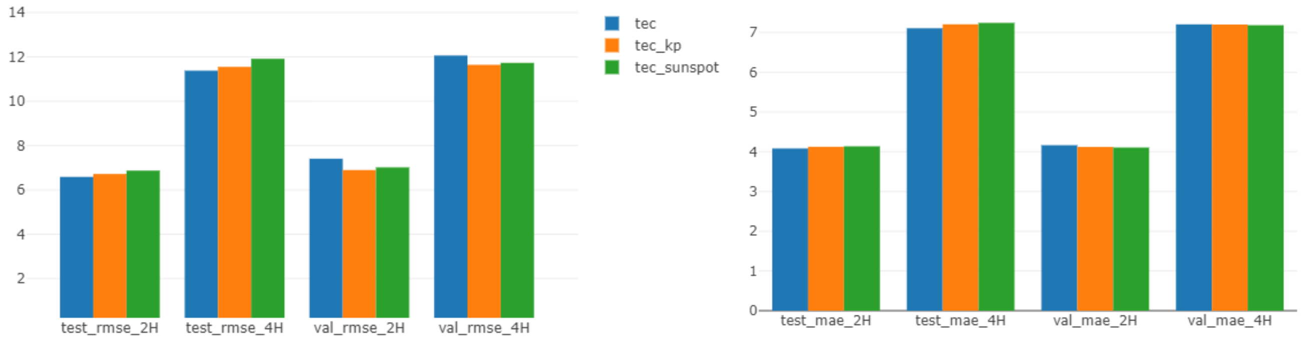






- Results:
  - Loss function (MSE) has similar behavior
  - Validation loss function is better when using also Kp
  - Use of Kp is more time consuming

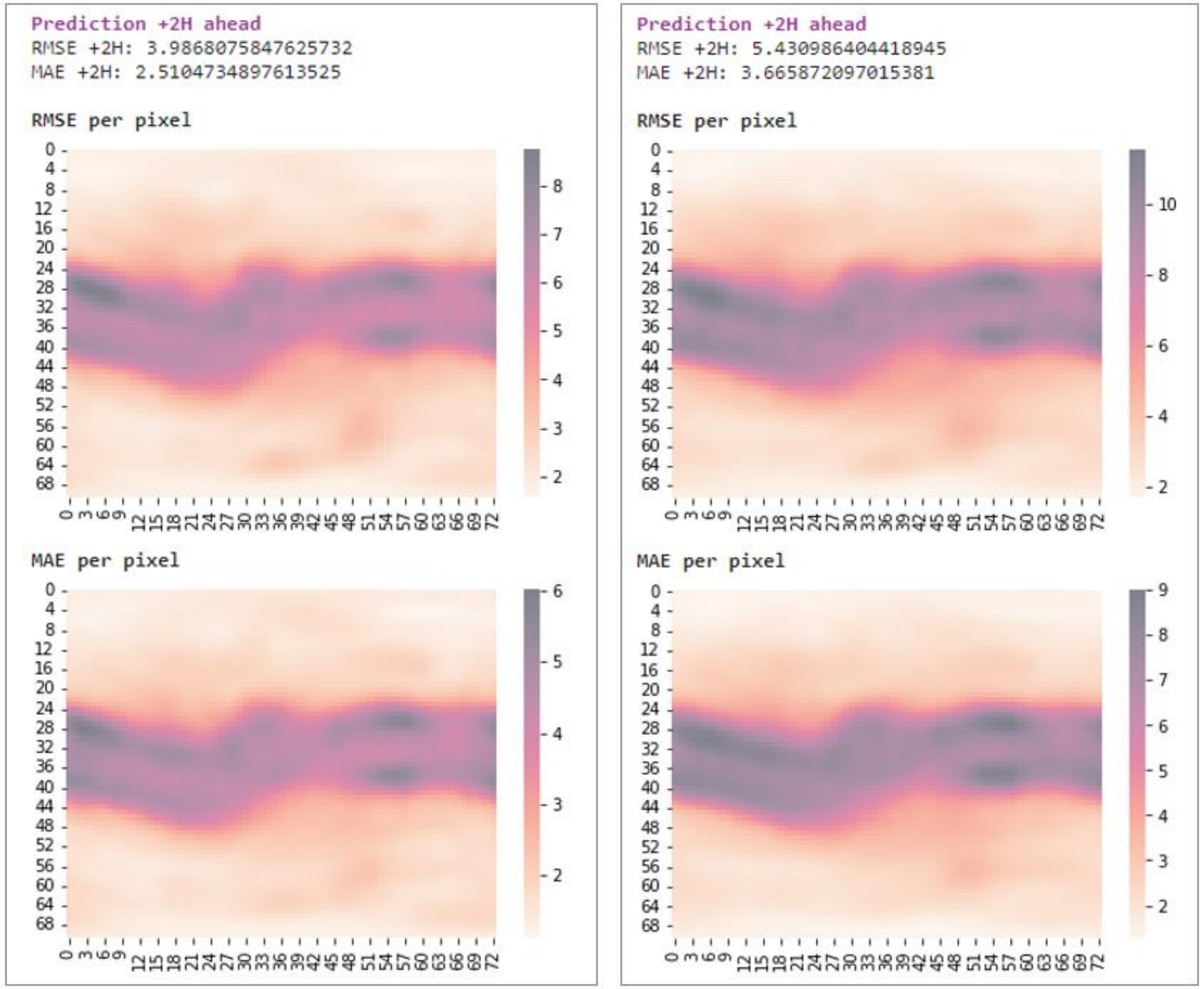
  - Metrics on test dataset:
    - Metrics values are inverted when evaluated on test dataset
    - test dataset preserves the order of magnitude of errors
    - → test also here the early stop strategy



App2 – Benchmark Methods Comparison

- Benchmark Method 1: **Last Available Map**, which consists in using the map at time T as prediction of next maps, that are map at time T+2H and map at T+4H

- Benchmark Method 2: **Previous Day Map**, which consists in using the map at 24H before as prediction of next maps, that are
  - map at time T-22H as map at T+2H and
  - map at time T-20H as map at T+4H



Errors comparison between DL (left) and Benchmark Method 1 (right).

Application 3 - Estimation of the SBAS correction parameters in the missed messages

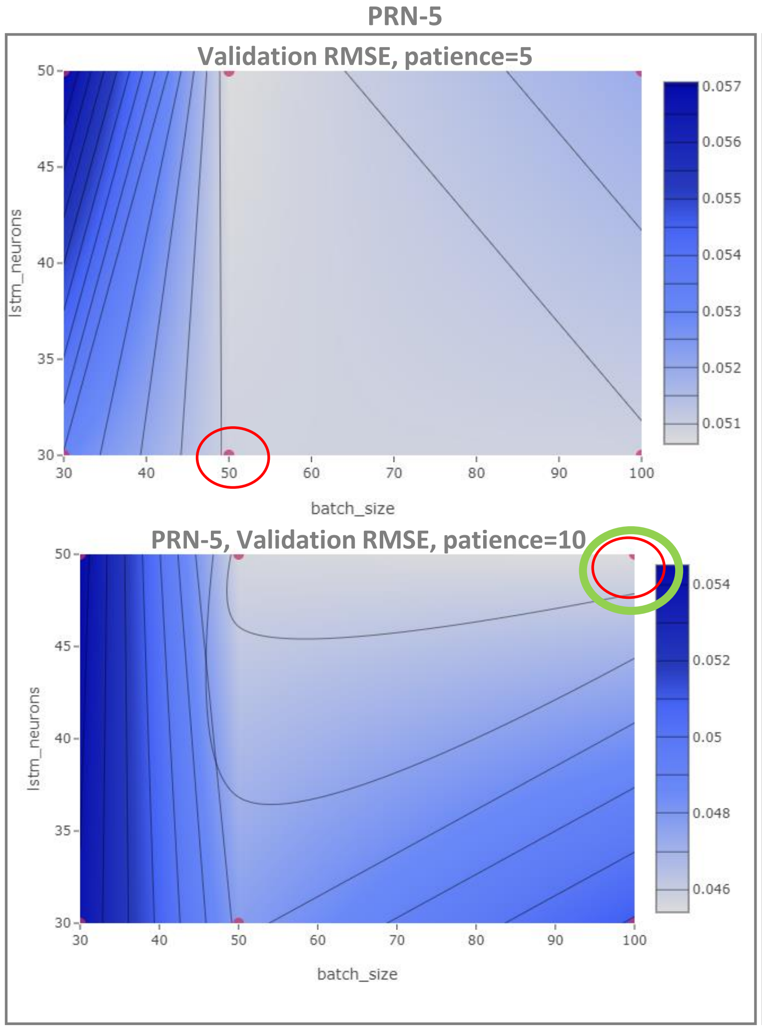
ML Models and Functional Validation

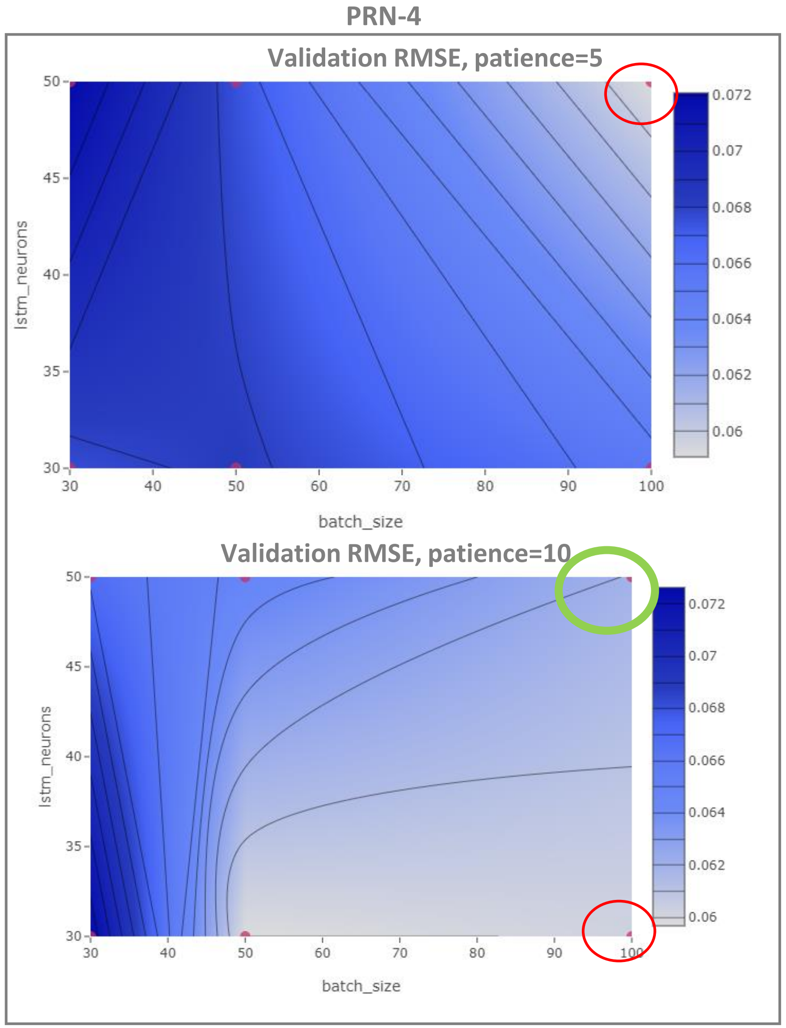
Dataset and ML Models

- Source data:
  - EGNOS .ems files, 2020
  - MT 2-5 (for fast corrections) and MT 25 (for slow corrections) used as data input
- Data rate is quite uniform:
  - Fast corrections: 4 seconds
  - Slow corrections: 30 seconds (i.e., between 28s - 32s)
- Application Design:
  - One uni-variate ML model for all PRNs (fast corrections) and one uni-variate model for each correction vector (slow corrections, in particular DX, DY, DZ and DAF0)
  - LSMT based models
  - Post processing to join results, having all the values grouped by messages type
- Same regularization strategies used for App 1 were adopted.

App3 – FC Model Tuning and Results

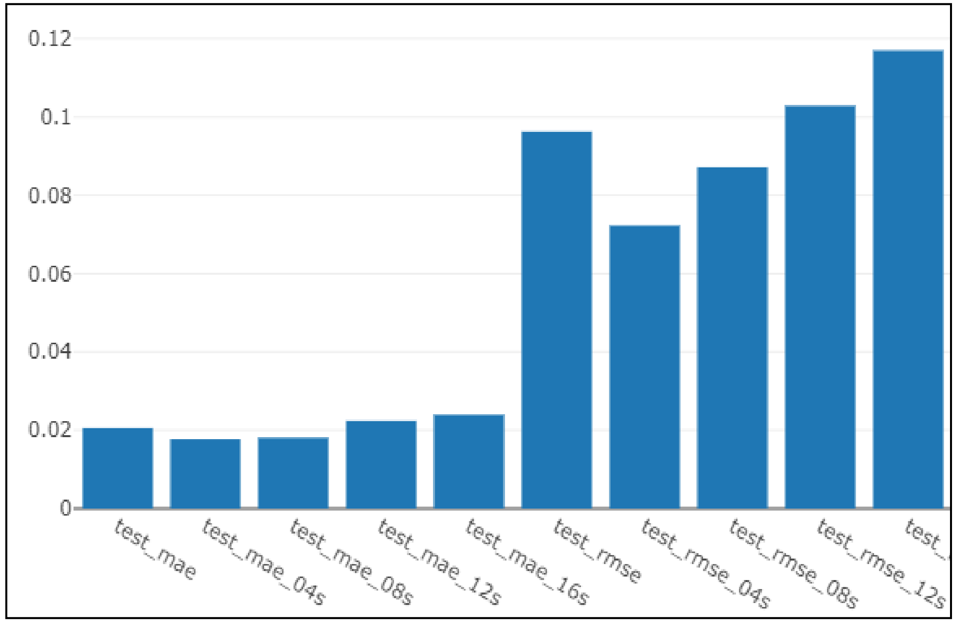
Comparison between best models obtained for PRN4 and PRN5, MT 2





→ Best model for PRN-5 is not the same for PRN-4, but it would be a good model also for PRN-4

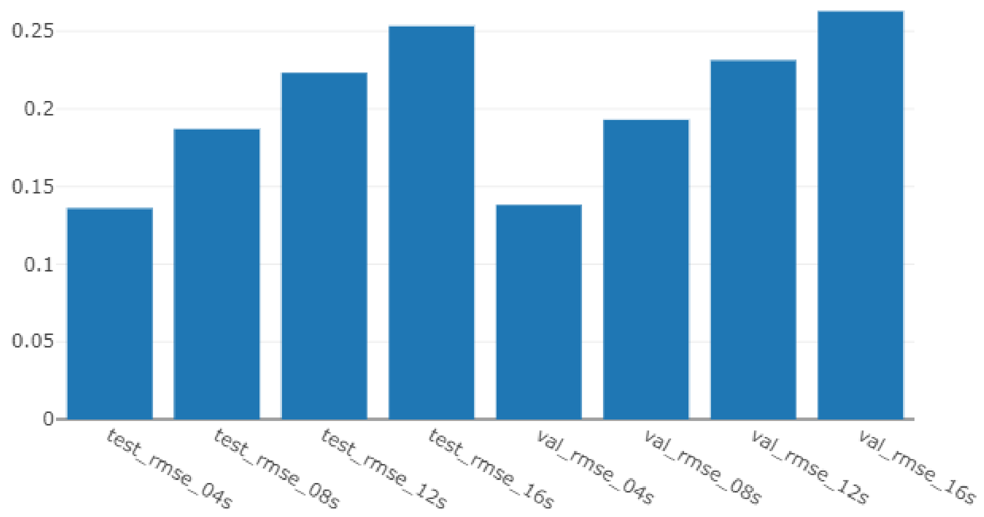
→ Use a unique 1-variate model for all PRNs



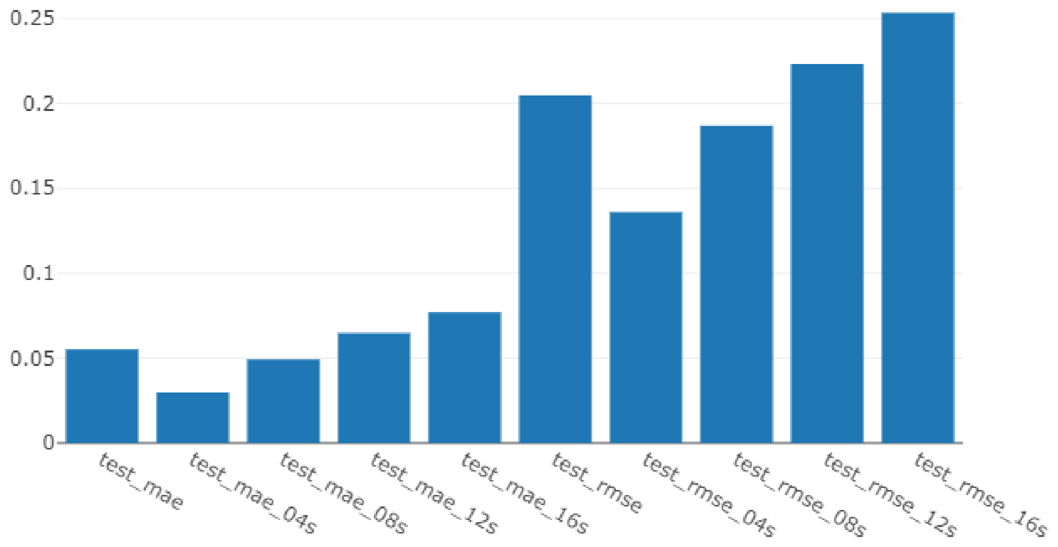
Metrics (RMSE and MAE) evaluation on test dataset, FC NN

App 3 – SC Functional Validation

- Validation performance were confirmed by the evaluation of the metric on the test dataset. Only DZ correction showed a difference slightly bigger.
- The errors get bigger when more seconds forward are predicted.



RMSE evaluation on test and validation datasets, SC, DX parameter.



Metrics (RMSE and MAE) evaluation on test dataset, SC, DX parameter

- The errors are not constant among the different samples, in fact the RMSE is about 4 times bigger than the MAE.

Application 4 - Disturbances classifications and outlier detections

ML Models and Functional Validation

App 4a – Scintillations Detection

- Dataset preparation:
  - CSV files containing the SDR receiver tracking stage outputs + labels (1 if scintillation, 0 otherwise)
  - The raw GNSS signal measurements (the 50 Hz I and Q correlators output) are used to calculate the features: peculiar parameters are computed over a fixed period of 60 seconds and then data are partitioned into 3 minutes blocks through a moving time window: $\{\langle I\rangle, \langle Q\rangle, \langle SI\rangle, \langle I^2\rangle, \langle Q^2\rangle, \langle SI^2\rangle\}$
  - Each sequence of (multi) input past observations is mapped to the output label

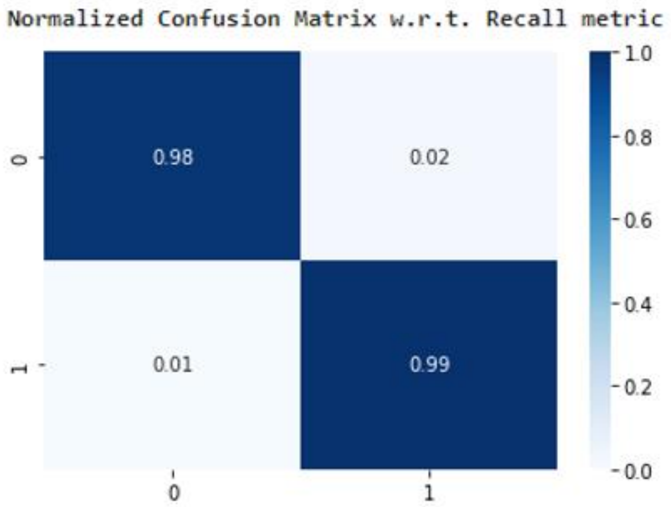
- Decision Tree model
  - Different splitting strategies (stratified, temporal and randomic) → Chosen the stratified splitting strategy, since it seems to be more effective when trying to balance dataset partitions.
  - Grid searching among interesting parameters
  - for each configuration of the grid, a cross validation strategy was adopted
  - the best configuration of the model is then fitted again by cross-validating the training process

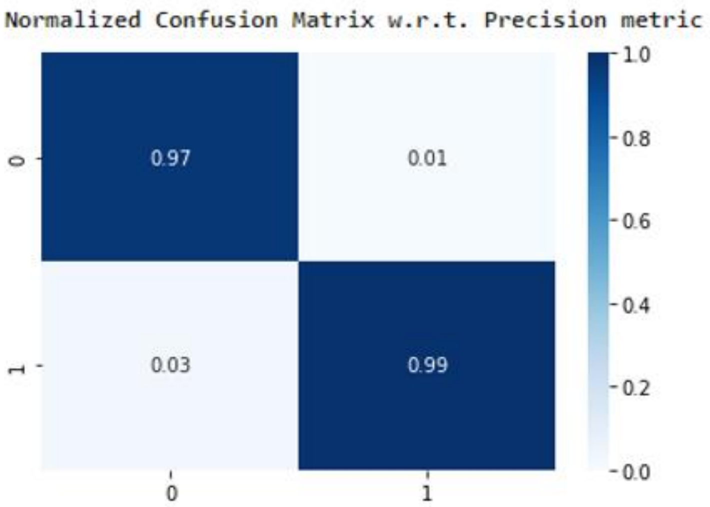
App 4a – Validation Results

- Functional Validation of the final model

Classification Report

	precision	recall	f1-score	support
0	0.970	0.977	0.974	132
1	0.994	0.992	0.993	499
accuracy			0.989	631
macro avg	0.982	0.985	0.983	631
weighted avg	0.989	0.989	0.989	631

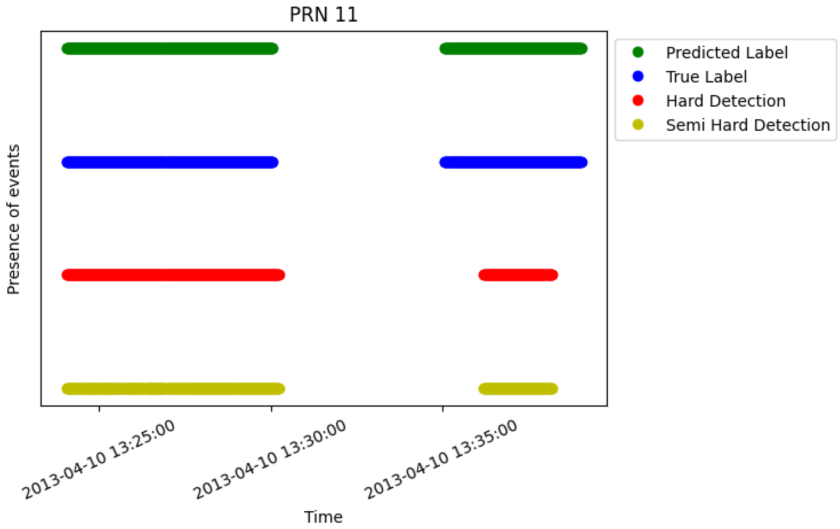


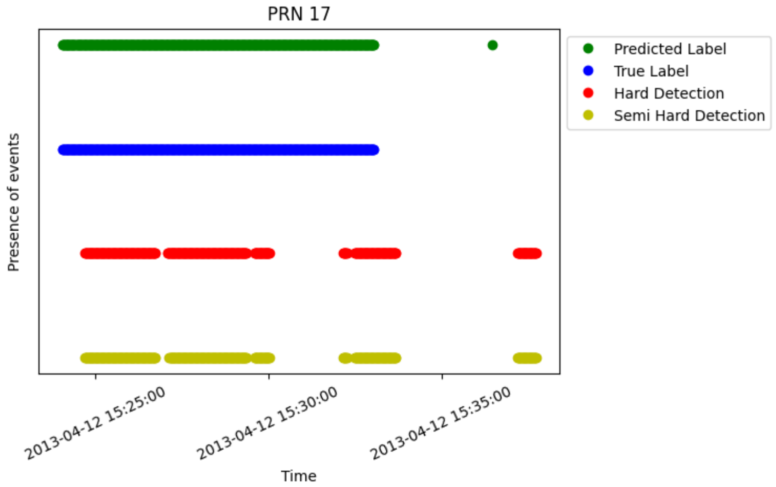


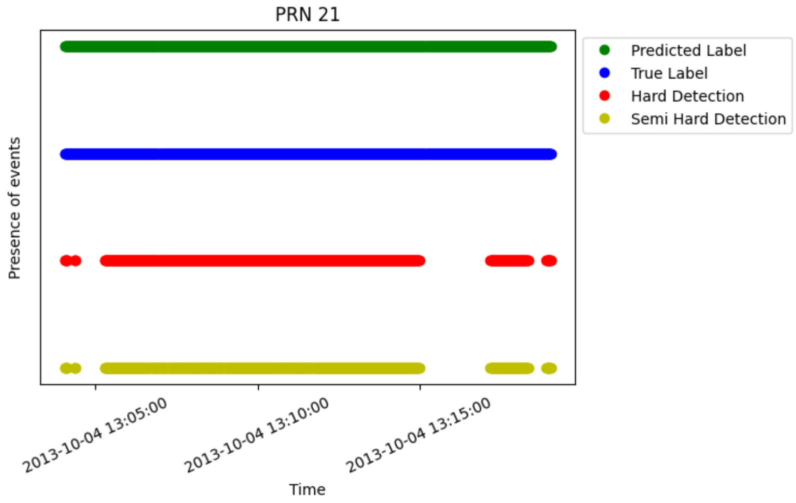
- In addition, a post-processing task has been implemented to return events identification and duration:
  - Labels returned by the ML model are also post-processed, in order to evaluate if little gaps between the events can be filled and to have events duration and frequency more reliable

App 4a – Benchmark Methods Comparison

- **Hard Detection:** $S_4 > Th_{S_4}$
- (Simplified) **Semi-hard Detection:** $(S4[n] > Th_{S4}) \wedge \left(C/N_0\,[n] > Th_{C/N_0}\right)$



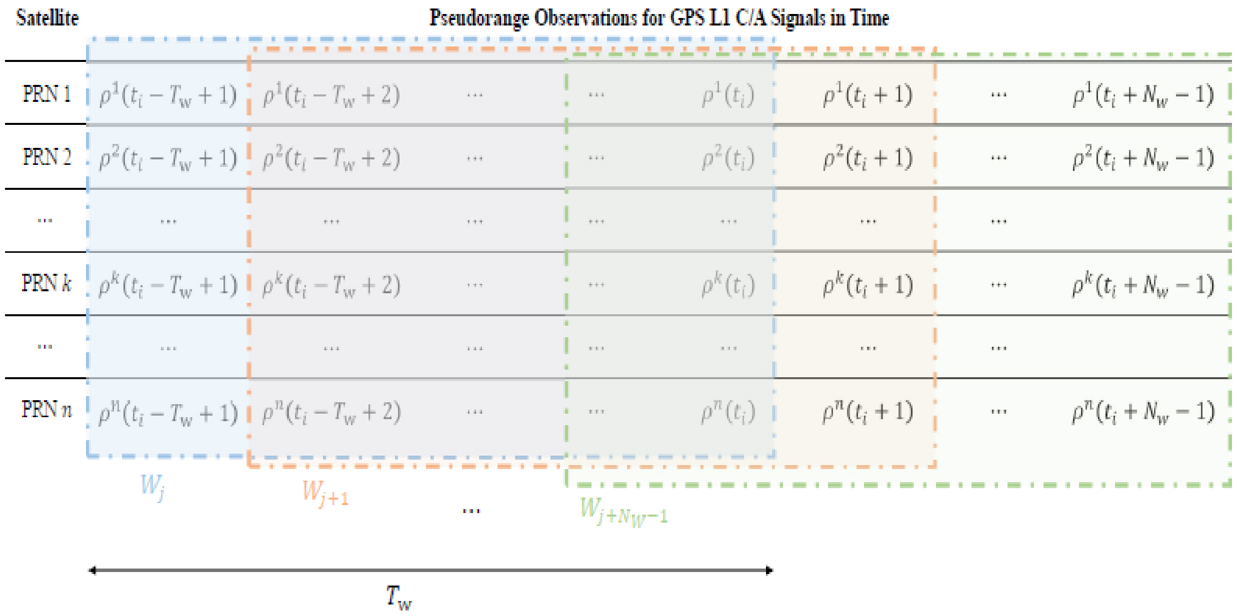




- The final model outperforms both the implemented benchmark methods. The colored dots are presents only in correspondence of class equals to 1 (presence of the scintillation event)

App 4b – Multipath Detection

- Dataset preparation:
  - CSV files containing the pseudorange measurements, the carrier phase and C/N_0 measurements + labels (1 if multipath, 0 otherwise)
  - **Time series detrending**: The measurements of each visible satellite during the considered window duration are passed through a designed feed-forward finite-duration impulse response FIR filter
  - **Features preparation**: After collecting the results of the filter of a number of consecutive sliding windows, standard deviations of the measurements are computed and organized in the matrix



$$X = \begin{bmatrix} \sigma_\phi^1 & \sigma_\rho^1 & \sigma_{C/N_0}^1 \\ \cdots & \cdots & \cdots \\ \sigma_\phi^k & \sigma_\rho^k & \sigma_{C/N_0}^k \\ \cdots & \cdots & \cdots \\ \sigma_\phi^n & \sigma_\rho^n & \sigma_{C/N_0}^n \end{bmatrix}$$

App 4b – Model Design

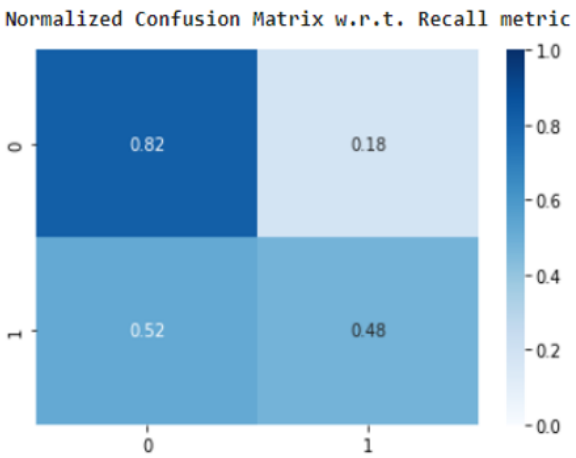
- K-means clustering model
  - Unsupervised models tested for the application, due to the small size of the dataset
  - k = number of clusters = 2 → "multipath" (class 1) and "no multipath" (class 0).
  - At each step the data input X are passed to the model, which find the 2 clusters of PRNs → the labels assigned by the model are simply representative of the two identified clusters
- Post-processing algorithm based on: when the multipath is present, the value of the parameter SNR is often smaller than the nominal condition
  → once the 2 clusters are identified, the average value of SNR of the PRNs inside each cluster is calculated and the mean SRNs are compared among themselves and w.r.t. to:
  - *the minimum difference between mean SNR values* (averaged inside each cluster) to consider the 2 clusters separated
  - *the maximum nominal SNR value*: when a unique post-processed cluster is considered, if both the mean SNR values of original clusters are lesser than this threshold, it is assumed that all the PRNs are affected by multipath.

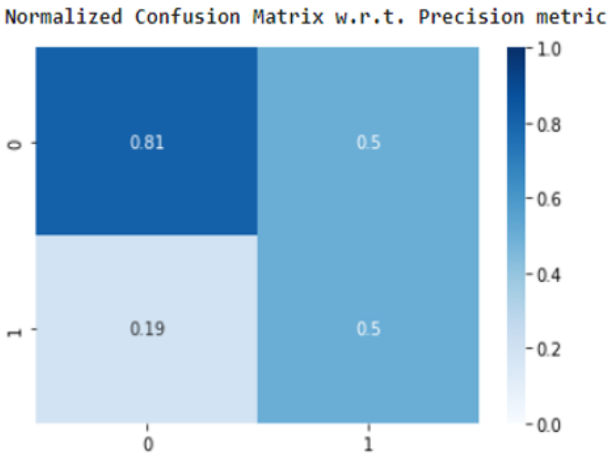
App 4b – Functional Validation Results

- Functional Validation Results

Classification Report

	precision	recall	f1-score	support
0	0.814	0.821	0.817	3417
1	0.496	0.484	0.490	1245
accuracy			0.731	4662
macro avg	0.655	0.652	0.653	4662
weighted avg	0.729	0.731	0.730	4662





- The clustering results are quite bad: each cluster often contains both multipath and no-multipath affected PRNS
- values of FP and FN are quite high: only the half of PRNs actually characterized by multipath are detected, and only the half of the PRNs returned as affected by multipath are actually affected by it.

GNSS Performance and Overall Application Evaluation

Andrea Nardin

Application 1 - Improving Orbit Prediction by means of Machine Learning Approaches

Validation of the end user application

App 1: End-user performance assessment

- The performance assessment is done comparing the computed **satellite positions** obtained through the classical algorithm by using **both the estimated and real ephemeris** (ground truth)

- To provide a preliminary **assessment of the impact on the accuracy of the estimated position**, a simplistic case of static user implementing a Least Mean Square (LMS) PVT solution is considered (i.e. impact of the ephemeris error)

App 1: case studies

1. ***Ground truth***. Ephemerides at the current time are **available**. Position of the satellites computed at the ephemeris reference time
2. ***NN prediction***. Ephemerides at the current time are **not available**. Ephemerides **predicted through ML** 2, 4, 6,8 hours forward. Satellite positions computed at the predicted ephemeris current time.
3. ***Kepl. Propagation.*** Ephemerides at the current time are **not available**. Sat. positions at the current time computed from old ephemerides, applying the **corrective equations** described in ICDs.
4. ***Linear prediction***. Ephemerides at the current time are **not available**. They have been **linearly extrapolated** from old ephemeris at time t_k and t_{k-1}, according to

$$\theta_{t_k+\Delta} = \theta_{t_k} + \Delta \frac{\theta_{t_k} - \theta_{t_{k-1}}}{t_k - t_{k-1}}$$

where θ is an ephemeris element and Δ is the forward prediction of 2,4,6, or 8 hours. Computation of satellite positions at current time is obtained based on the extrapolated ephemeris $t_k + \Delta$. This is a basic benchmark with minimal requirements in terms of available data and processing burden

App 1: Satellite position estimation

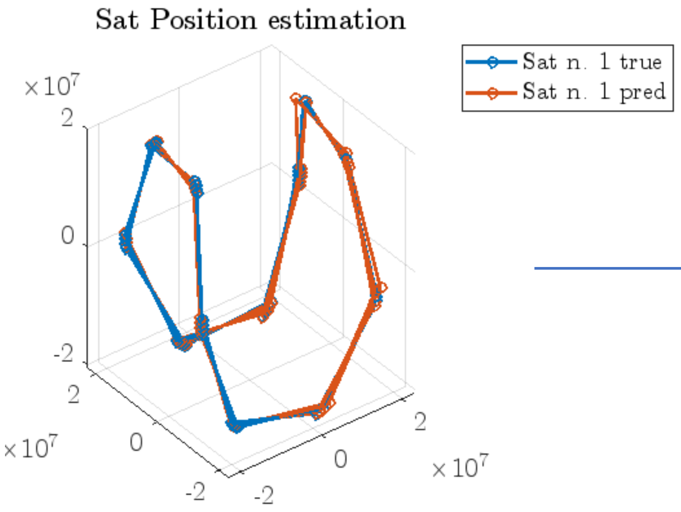
Settings

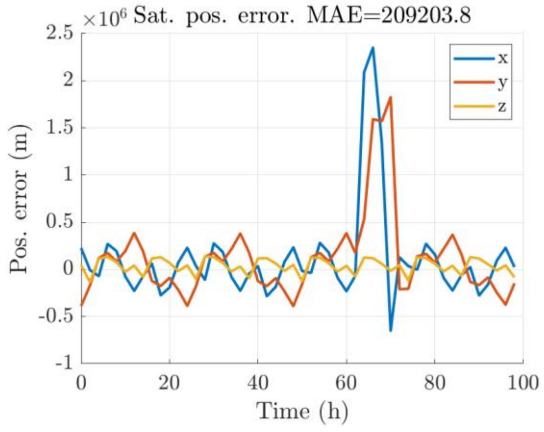
Parameter	Value
Pseudorange measurement noise	0
User motion	static
Observation time	100 h
PVT computation	LMS
Forward prediction	2h – 8h
No. of satellites	4-10

Results (Sat positions)

Forward prediction	NN prediction MAE (m)	Kepl. propagation MAE (m)	Linear prediction MAE (m)
2 hours	5.6e4	3.7e3	7.2e3
4 hours	10.4e4	3.4e3	10.9e3
6 hours	13.6e4	3.8e3	15.0e3
8 hours	20.9e4	3.8e3	19.6e3

8h forward NN prediction

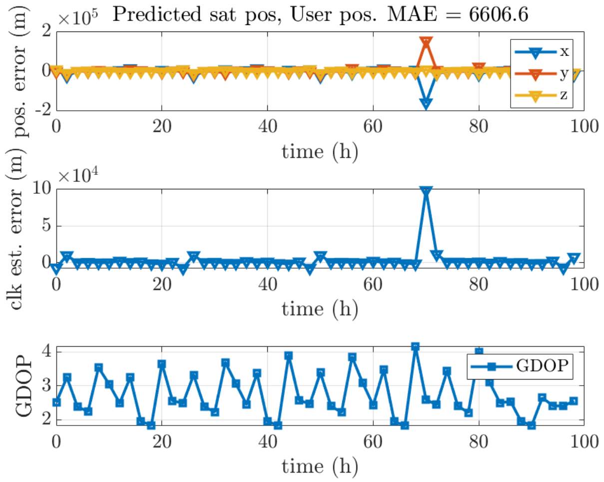


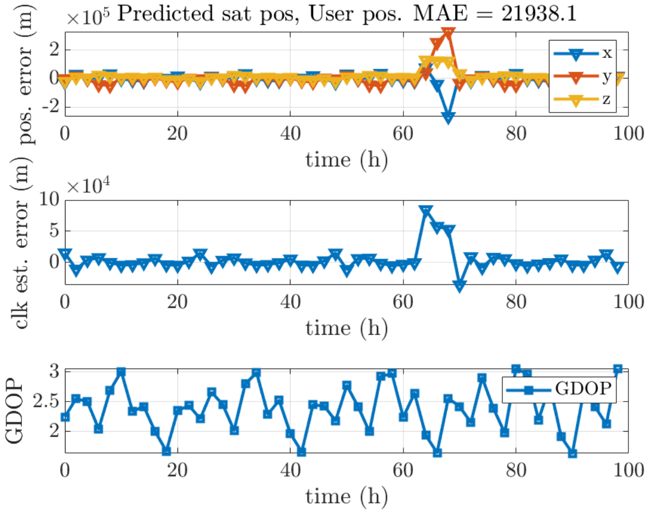


App 1: User position estimation

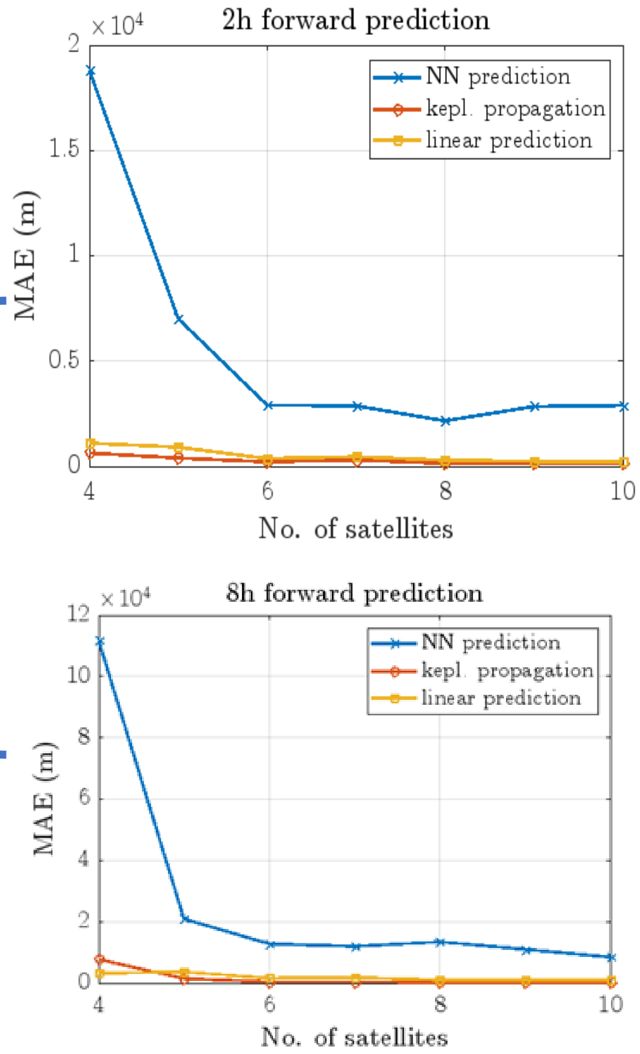
Results (User positions)

Forward prediction	Ground truth MAE (m)	NN prediction MAE (m)	Kepl. propagation MAE (m)	Linear prediction MAE (m)
2 hours	1.4e-9	6.6e4	3.2e2	7.0e2
4 hours	1.7e-9	11.7e4	7.3e2	14.3e2
6 hours	3.1e-9	15.2e4	3.5e2	23.3e2
8 hours	1.7e-9	21.9e4	7.7e2	32.6e2





Variable no. of satellites



App 1: Parameter Impact on user position

- Results are not satisfactory
- By assessing the **impact of each predicted parameter** on the obtained satellite and user error, the Keplerian parameter M_0 (mean anomaly) has been identified as main source of error (by far)
- Its prediction through the NN model is not effective
- The error on the predicted M_0 is the reason of the large error experienced

- Example: Working with predicted Keplerian elements via NN (excluding M_0) provides on average a user position error:
  - MAE = 29.5 m
  - average RMS =34.3 m

Parameters	**Value**
Pseudorange measurmenet noise std	**8 m**
User motion	static
Observation time	100 h
PVT computation	LMS
Forward prediction	**8h**
No. of satellites	**4**

Application 2 - Prediction of daily maps of the ionosphere

Validation of the end user application

App 2: Prediction of daily maps of the ionosphere

- Assessment of the performance has been performed:
  - comparison of predicted vs. actual TEC maps, for different time windows of forecast ahead;
  - impact on the **slant TEC assessment within the grid** in order to use the classical correction formula:

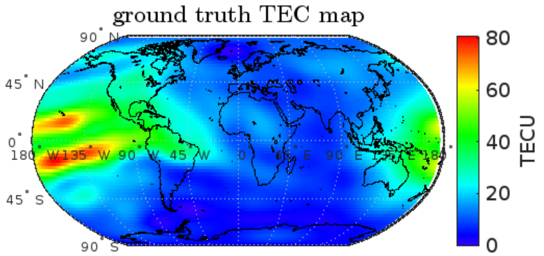
$$I_{\rho_j} = \frac{40.3\ sTEC}{f^2}$$

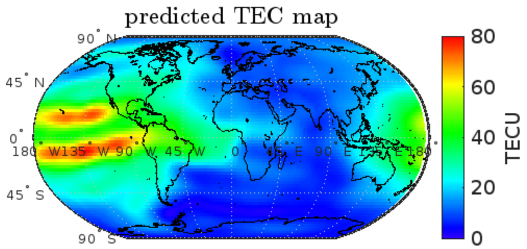
App 2: Average vertical ionospheric bias error

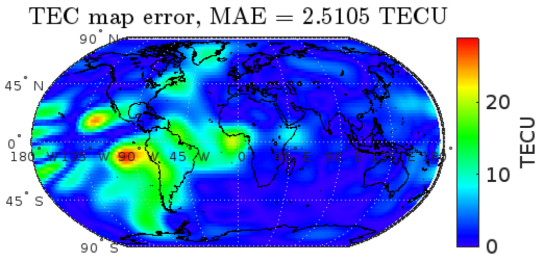
- Analysis of iononospheric bias error affecting pseudorange
- TEC map estimation from IGS product (ground truth) vs. TEC map prediction

Parameter	**Value**
Pseudorange measurmenet noise	0
User motion	static
TEC map observations	4379
PVT computation	LMS
Forward prediction	2h – 4h
TEC grid	71x73

Example (single TEC snapshot)



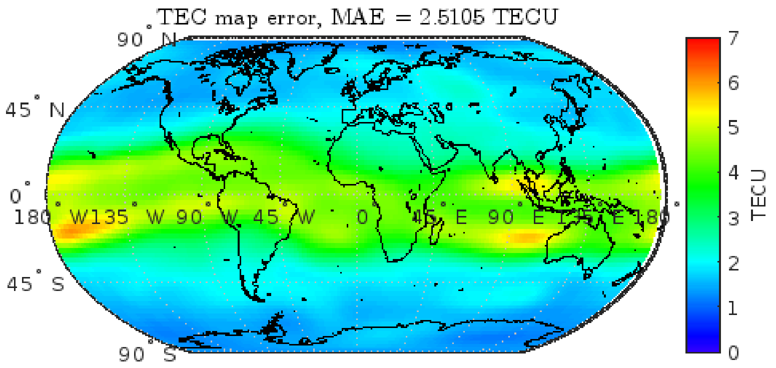


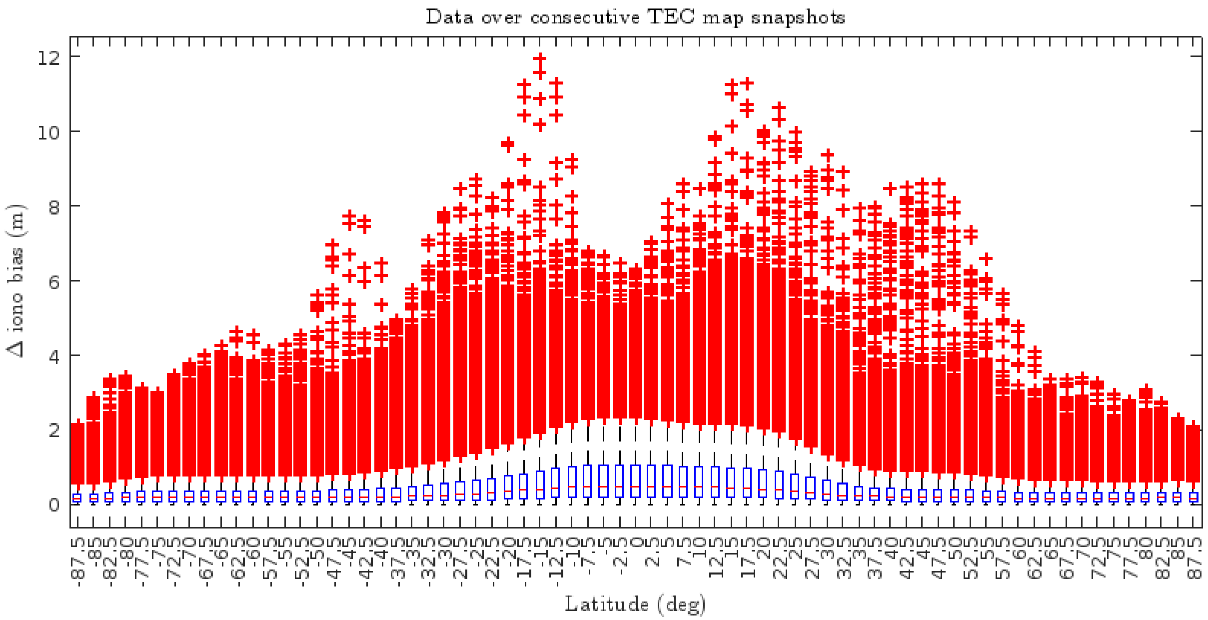


1 snapshot of TEC map error. The **average** ionospheric bias error will be obtained by averaging consecutive snapshots

App 2: Average vertical ionospheric bias error

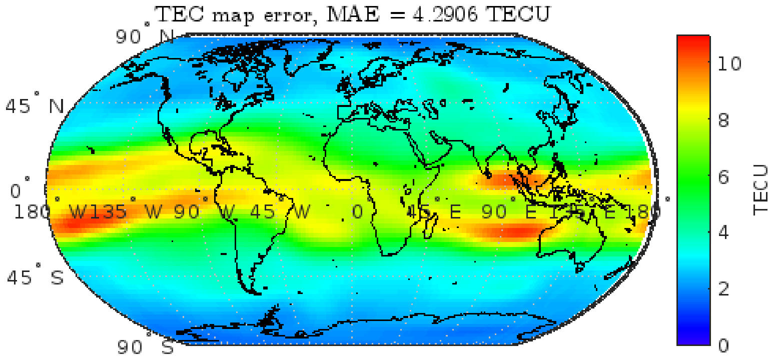
- TEC prediction **average over consecutive snapshots**
  - 2h forward prediction
  - Average vertical ionospheric bias error wrt ground truth = **40.8 cm** (2.5 TECU)

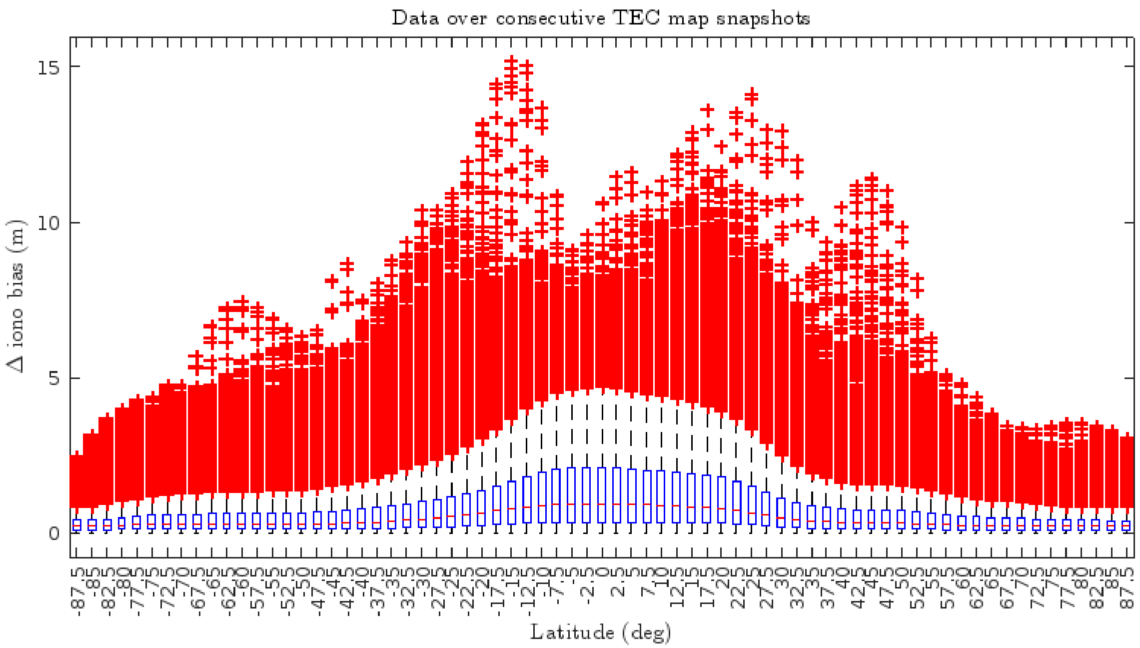




App 2: Average vertical ionospheric bias error

- TEC prediction **average over consecutive snapshots**
  - 4h forward prediction
  - Average vertical ionospheric bias error wrt ground truth = **69.7 cm** (4.3 TECU)





App 2: Case studies based on user position

- Analysis complemented evaluating the impact on the **slant TEC** using classical correction formula:

$$I_{\rho_j} = \frac{40.3\ sTEC}{f^2}$$

- Three different case studies at mild elevation:

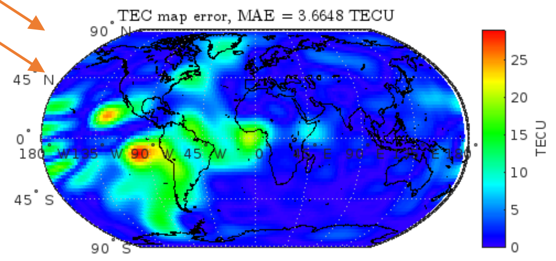
1. ***Best case***. user located in correspondence of **the minimum vertical TEC prediction error**;
2. ***Worst case***. user located where the vertical TEC prediction error is **maximum**;
3. ***Mode***. user located in correspondence to the **mode** value of the vertical TEC prediction error.

case 3 is representative of an "average" performance at medium elevation. Since the analysis addresses specific locations, the most representative location is the one corresponding to the most common vTEC error, i.e. the mode.

App 2: Case studies based on user position

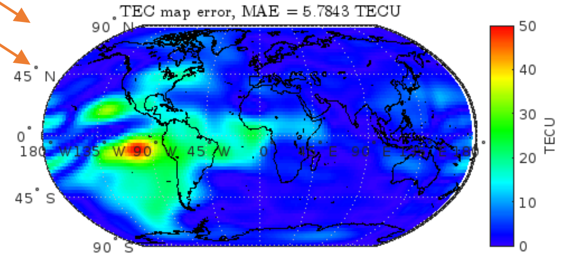
Case study	User location (lat, lon)	Selected satellite elevation (deg)	Slant ionospheric bias error (2h)
Best vTEC	84°, -129°	46°	**6.5 cm**
Worst vTEC	9.4°, -99°	46°	**584.7 cm**
Mode vTEC	82°, -129°	46°	**9.7 cm**

2h prediction



Case study	User location (lat, lon)	Selected satellite elevation (deg)	Slant ionospheric bias error (4h)
Best vTEC	-48°, -164°	42°	**41.4 cm**
Worst vTEC	7°, -104°	38°	**1060 cm**
Mode vTEC	82°, -129°	44°	**113.9 cm**

4h prediction



Application 3 - Estimation of the SBAS correction parameters in the missed messages

Validation of the end user application

App 3: End-user performance assessment

- Since the goal of the application is the prediction of missed messages from the stream of the SBAS messages, a clear advantage of a good regression is the **availability of corrections for an extended amount of time**.
- The **benchmark** is the **absence of any SBAS correction**, so the estimation of the capability of filling up missed messages, for different outage time windows,
- Furthermore evaluation of the accuracy of correction with respect to the ground-truth is assessed.

- **Note**: no SoL applications are targeted since the reliability of these predictions should be assessed. However, these predicted values could be used for other applications exploiting EGNOS corrections (e.g. precision farming)

from the *Functional requirements specification* deliverable

App 3: case studies

Synthetic pseudoranges, corrected with SBAS fast correction, have been used to compute the user position. The user location computed with pseudoranges corrected with SBAS corrections is used as a **ground truth**. The PVT computation with such exact measurements allows to define:

1. ***Best case*** scenario. SBAS messages are **received** and pseudoranges are profitably corrected;
2. ***Worst case*** scenario. SBAS corrections are **missing**, PVT computed using non-corrected pseudoranges
3. ***Benchmark*** scenario. SBAS corrections are **missing**, corrections estimated through **a linear extrapolation** formula

$$PR_{corrected}(t) = PR_{measured}(t) + PRC(t_{of}) + RRC(t_{of}) \cdot (t - t_{of}) \qquad \text{with} \qquad RRC(t_{of}) = \frac{PRC_{current} - PRC_{previous}}{\Delta t}$$

4. ***ML*** scenario. SBAS corrections are **missing**, correction terms **predicted through a ML** approach.

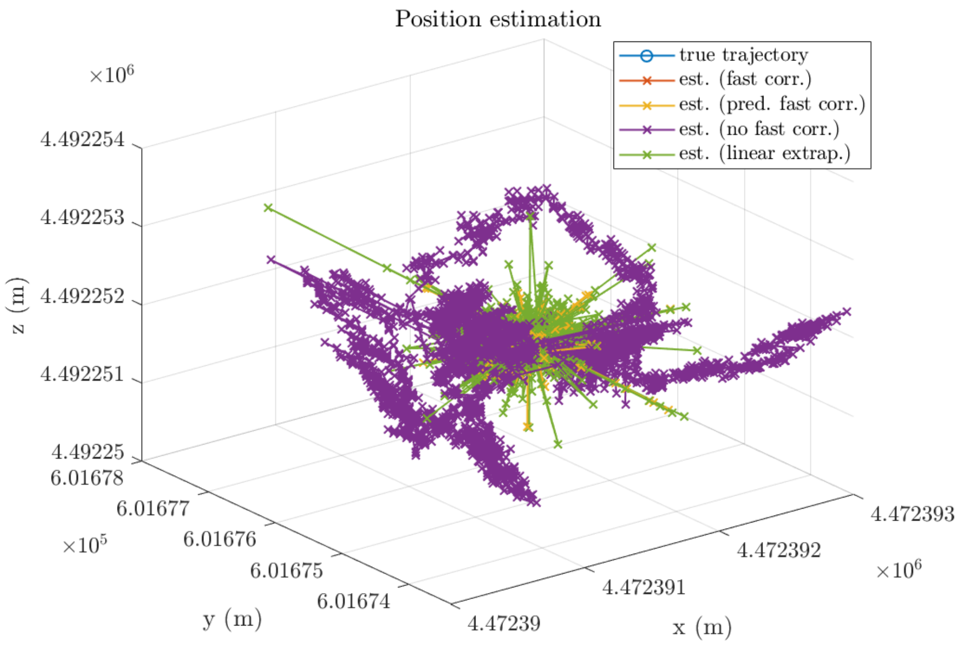
The SBAS corrections in (*iii*) and (*iv*) have been computed as a forward prediction of 4, 8,12 and 16 s.

Settings

Parameter	**Value**	**Notes**
Pseudorange measurmenet noise	0	
User motion	static	
Observation time	20 h	4s step
PVT computation	LMS	
Forward prediction	4s – 16s	
No. of satellites	6	Corrections applied to all

App 3: User position error (4s prediction)

4s prediction example

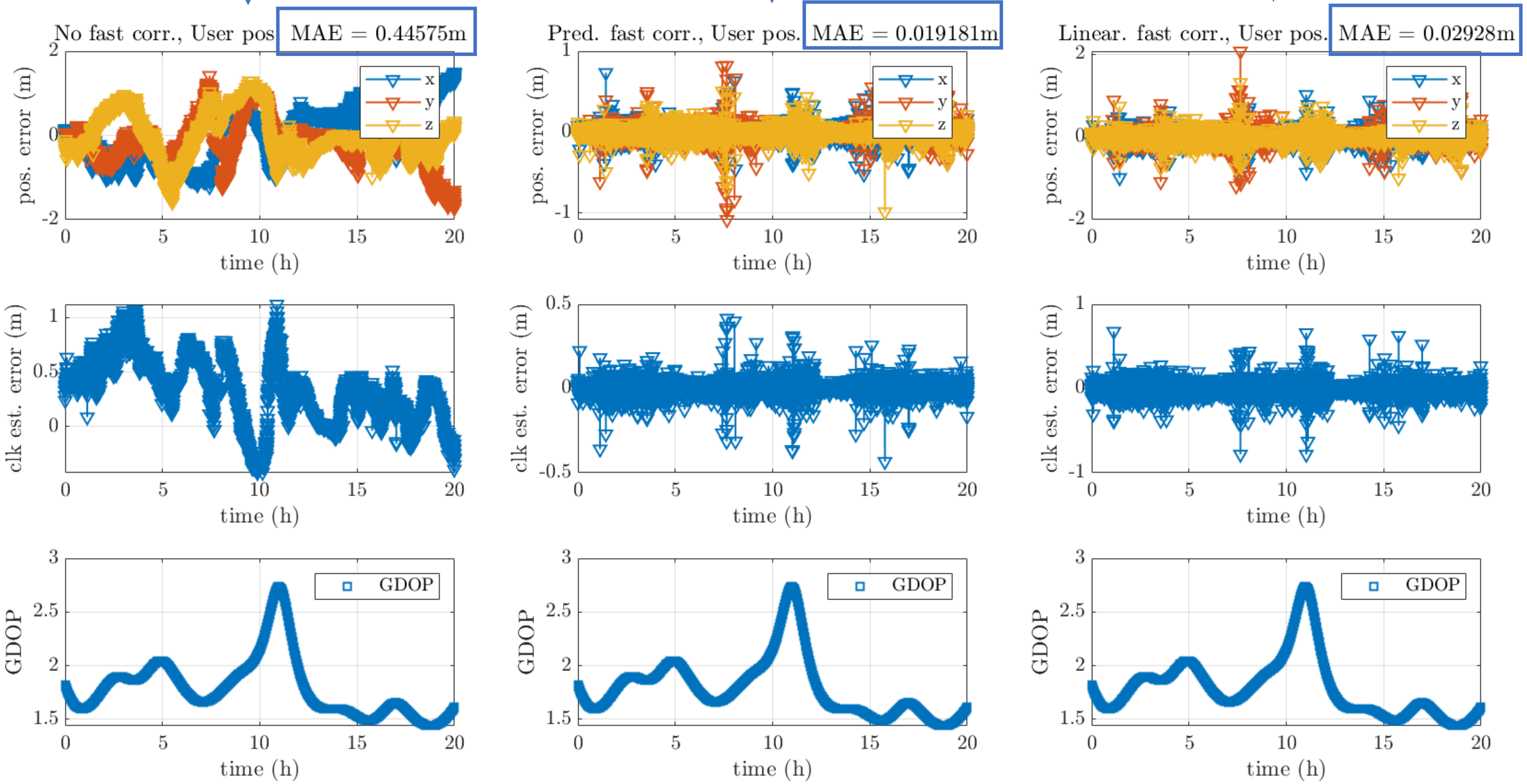


Forward prediction	Best case: Fast corrections MAE (m)	Worst case: No fast correctionbs MAE (m)	Benchmark: Linear prediction MAE (m)	Investigated method: ML prediction MAE (m)
4 seconds	1.7e-9	0.45	0.029	**0.019**
8 seconds	1.8e-9	0.45	0.048	**0.021**
12 seconds	1.8e-9	0.45	0.066	**0.025**
16 seconds	1.7e-9	0.45	0.084	**0.028**

App 3: User position error (4s prediction)

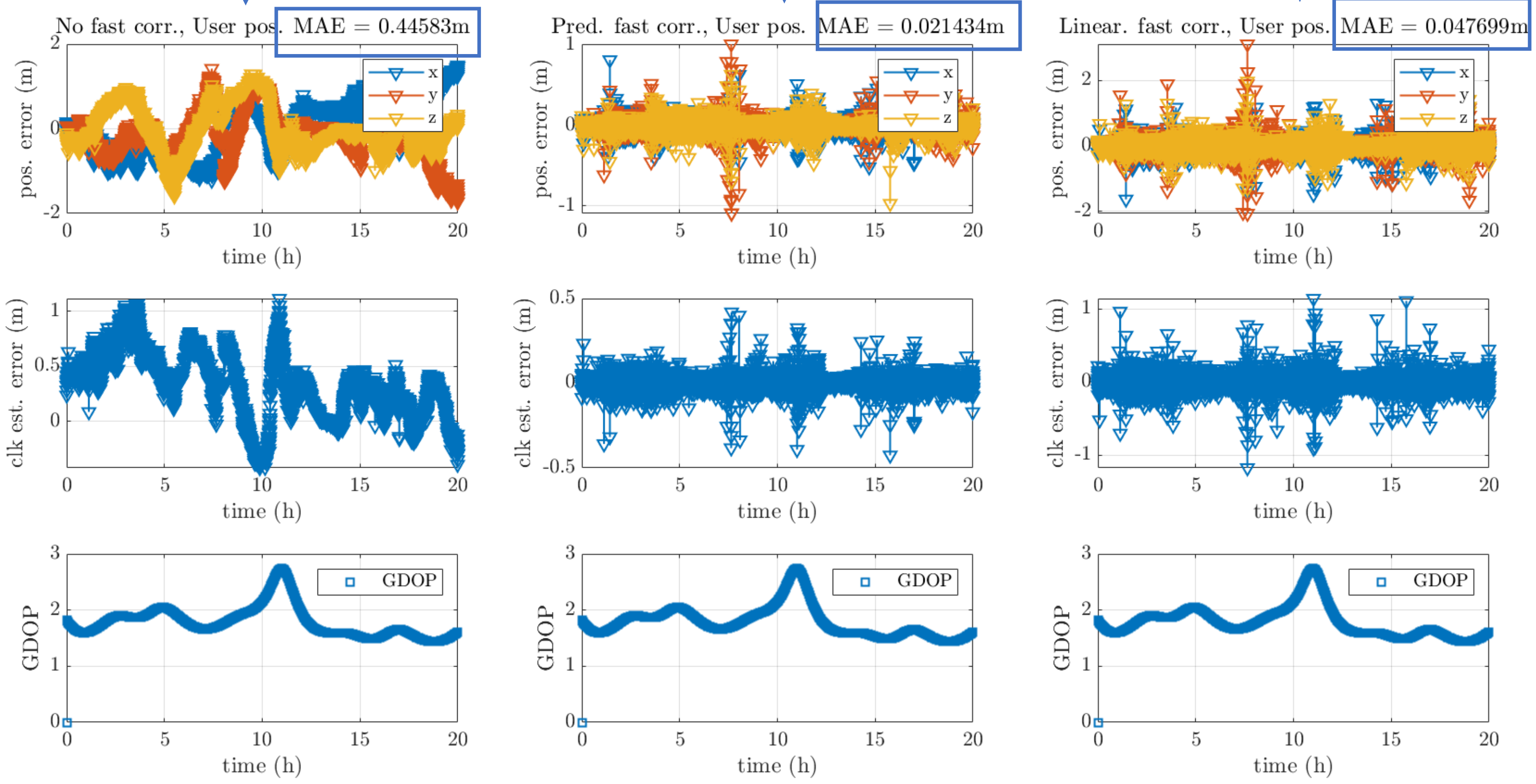
esa

No corrections | **ML prediction** | **Linear prediction**



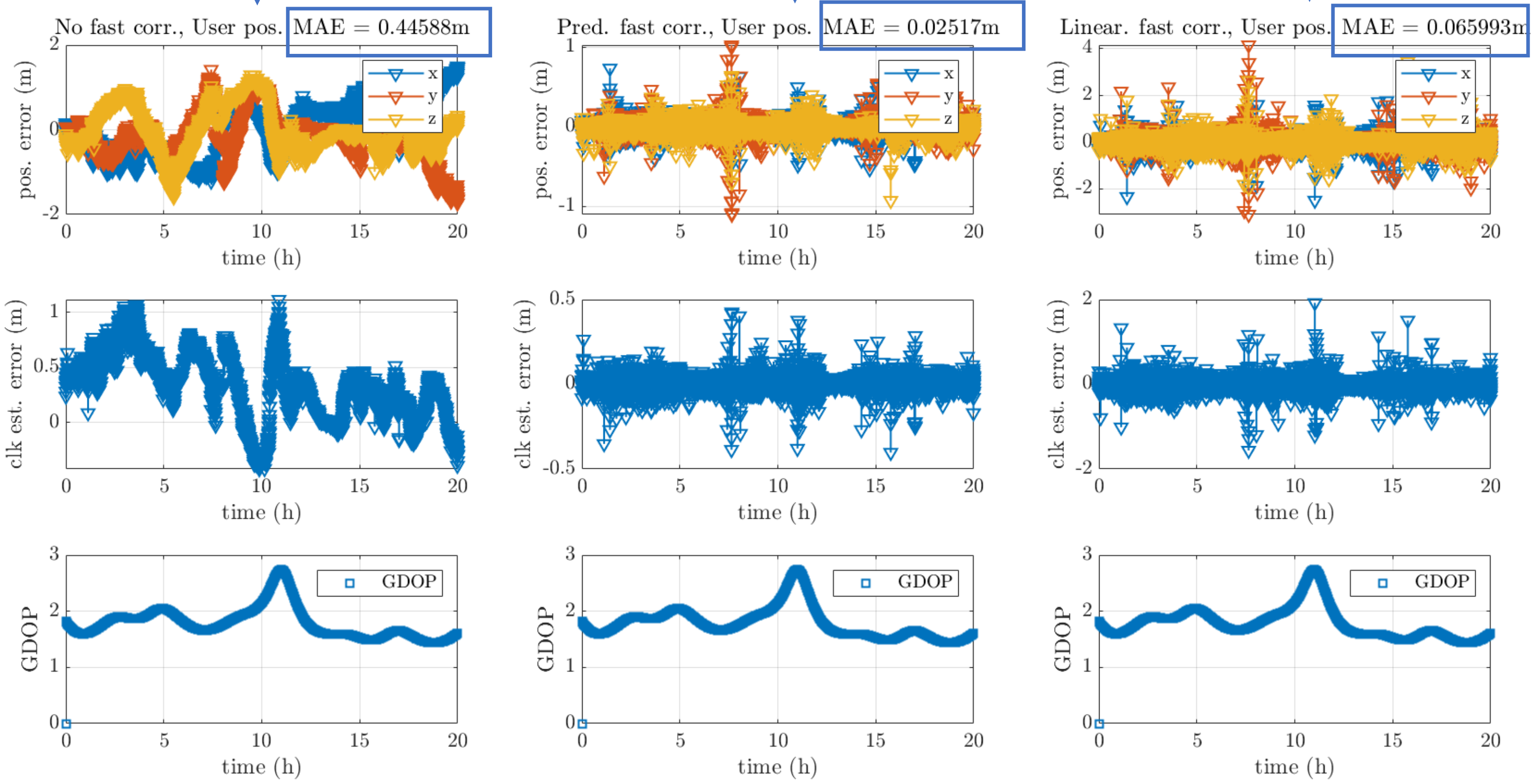
App 3: User position error (8s prediction)

No corrections | **ML prediction** | **Linear prediction**



App 3: User position error (12s prediction)

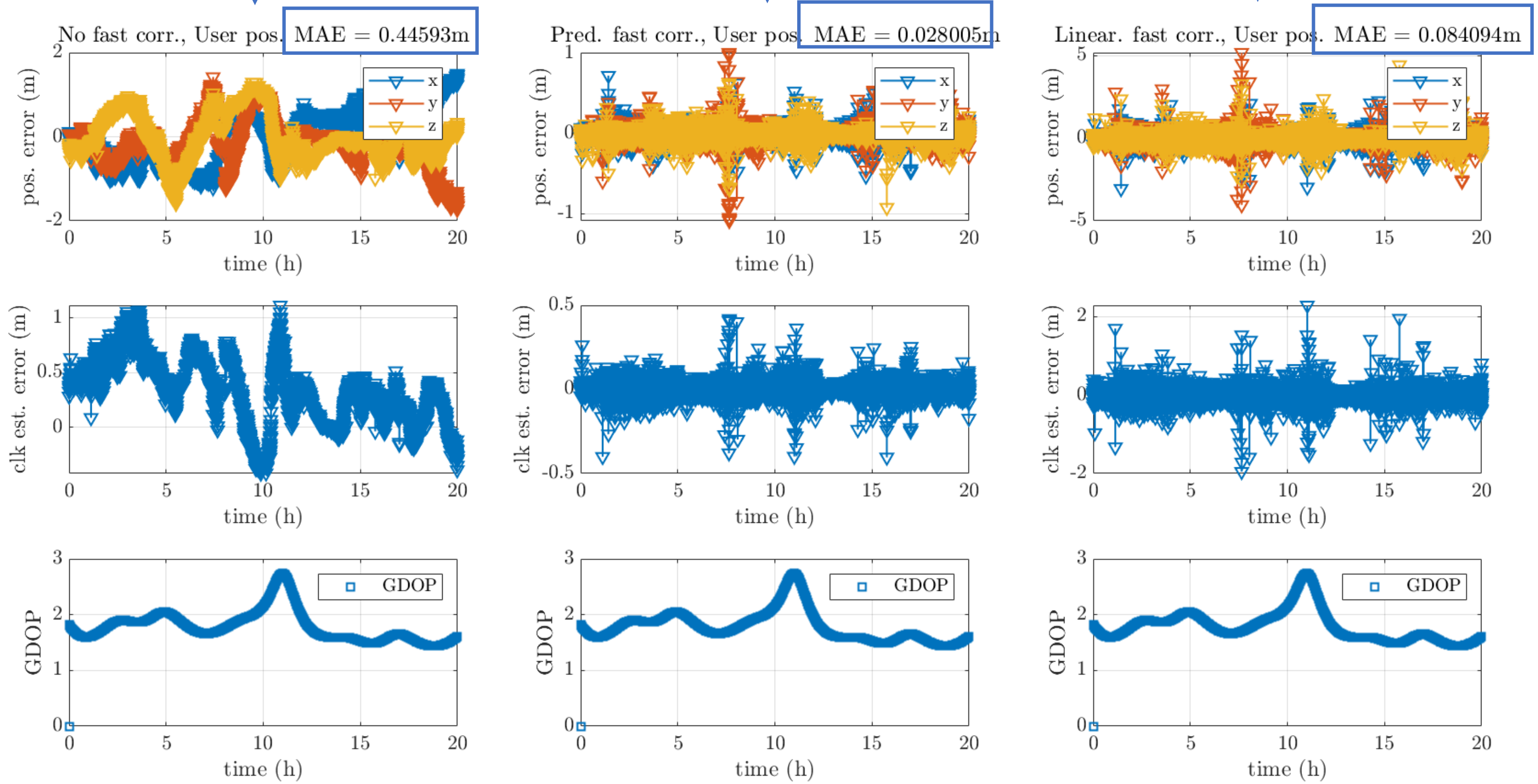
No corrections | **ML prediction** | **Linear prediction**



App 3: User position error (16s prediction)

esa

No corrections | ML prediction | Linear prediction



Application 4 - Disturbances classifications and outlier detections

Validation of the end user application

App 4: End-user performance assessment

- Since the scope of these applications is the detection of the outliers due to scintillation and multipath no end-user validation was foreseen.

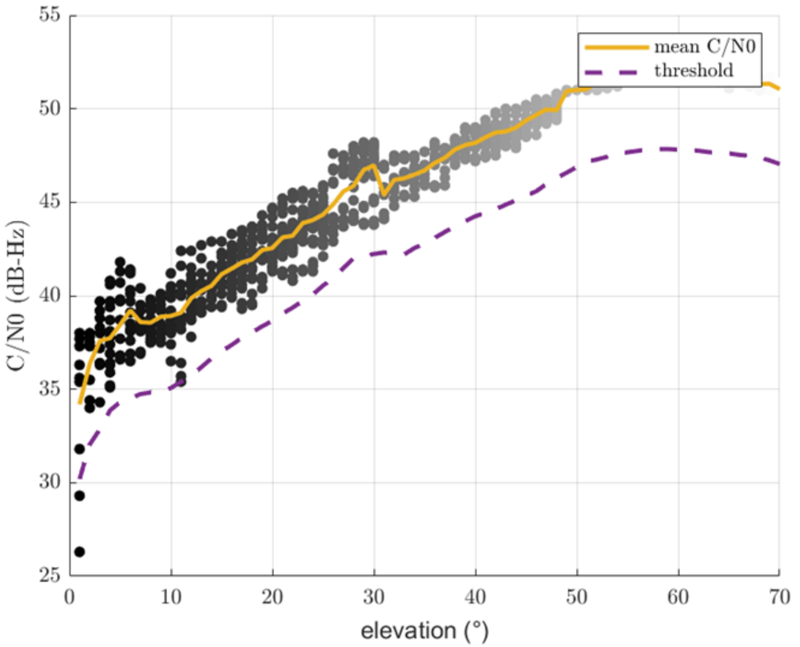
- In terms of detection performance the functional assessment characterizes detection rate and missed detection

- An end-user performance assessment has been setup for multipath detection (App. 4b)

App 4b: Multipath detection

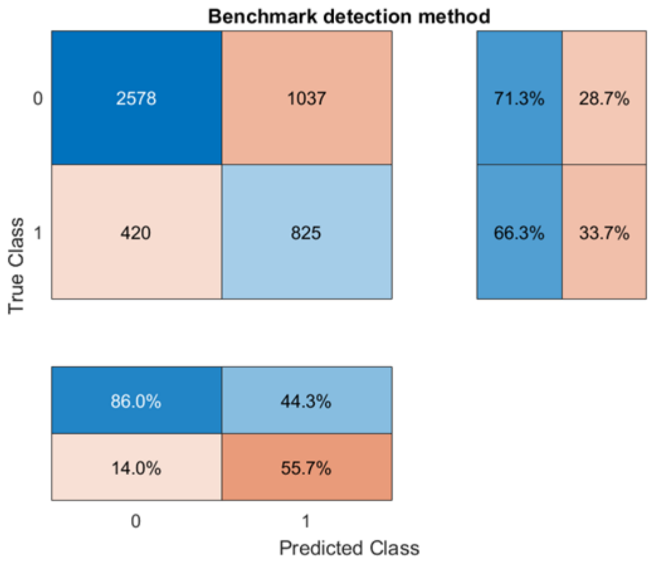
Case studies:

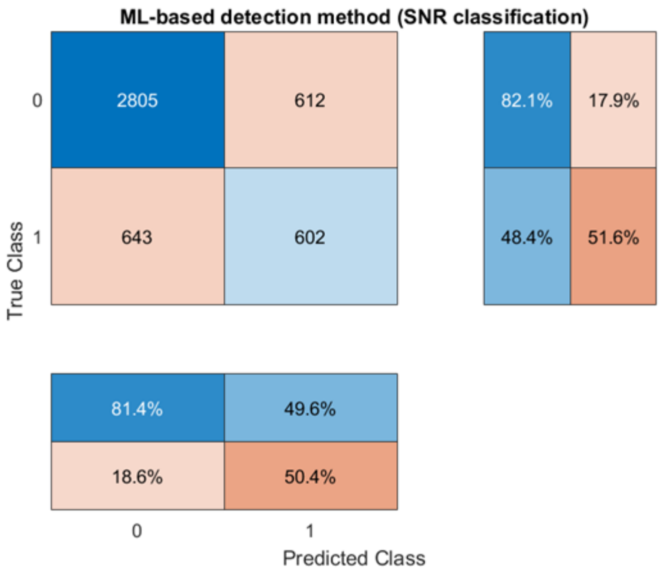
1. ***Ground truth***. Labeled multipath dataset
2. ***Benchmark***. Elevation-dependent thresholding over C/N0 values. The threshold is based on a training dataset
3. ***ML-based multipath detection***. The proposed clustering approach



App 4: Multipath detection

- Results:



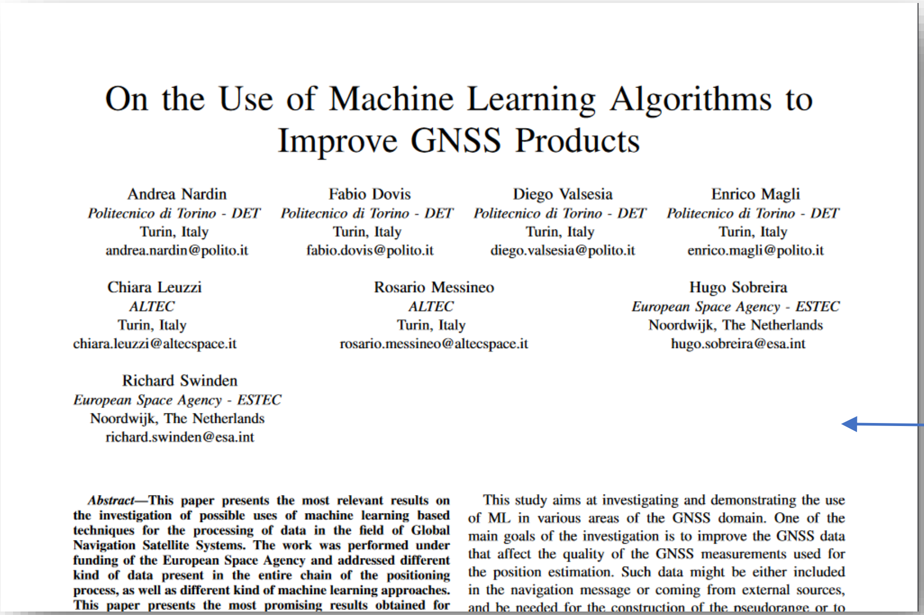


Dissemination: Conference Paper

Dissemination

Final Manuscripts due February 3, 2023

- Extended abstract accepted at ***IEEE/ION PLANS 2023***
  - **Session B5: Receiver Design, Signal Processing, and Antenna Technology**
    GNSS antennas, receivers, and processing methods for improving accuracy, reliability, or robustness. Methods including tracking loops, direct positioning, optimum and suboptimal multi-antenna systems, beamforming, polarization, and direction-of-arrival methods. Baseband signal processing, and software-defined implementations. Machine learning and data-driven methods for receiver design.
    Chair: Dr. Thomas Pany, *University of Bundeswehr*



On the Use of Machine Learning Algorithms to Improve GNSS Products

Andrea Nardin, *Politecnico di Torino - DET*, Turin, Italy, andrea.nardin@polito.it
Fabio Dovis, *Politecnico di Torino - DET*, Turin, Italy, fabio.dovis@polito.it
Diego Valsesia, *Politecnico di Torino - DET*, Turin, Italy, diego.valsesia@polito.it
Enrico Magli, *Politecnico di Torino - DET*, Turin, Italy, enrico.magli@polito.it
Chiara Leuzzi, *ALTEC*, Turin, Italy, chiara.leuzzi@altecspace.it
Rosario Messineo, *ALTEC*, Turin, Italy, rosario.messineo@altecspace.it
Hugo Sobreira, *European Space Agency - ESTEC*, Noordwijk, The Netherlands, hugo.sobreira@esa.int
Richard Swinden, *European Space Agency - ESTEC*, Noordwijk, The Netherlands, richard.swinden@esa.int

***Abstract*—This paper presents the most relevant results on the investigation of possible uses of machine learning based techniques for the processing of data in the field of Global Navigation Satellite Systems. The work was performed under funding of the European Space Agency and addressed different kind of data present in the entire chain of the positioning process, as well as different kind of machine learning approaches. This paper presents the most promising results obtained for**

This study aims at investigating and demonstrating the use of ML in various areas of the GNSS domain. One of the main goals of the investigation is to improve the GNSS data that affect the quality of the GNSS measurements used for the position estimation. Such data might be either included in the navigation message or coming from external sources, and be needed for the construction of the pseudorange or to

In addition to a general introduction of the project, the **focus** is on the most promising results, obtained for **application 2** and **application 3**

GMLD Outcome Evaluation and Future Recommendations

Rosario Messineo

GMLD Achievement

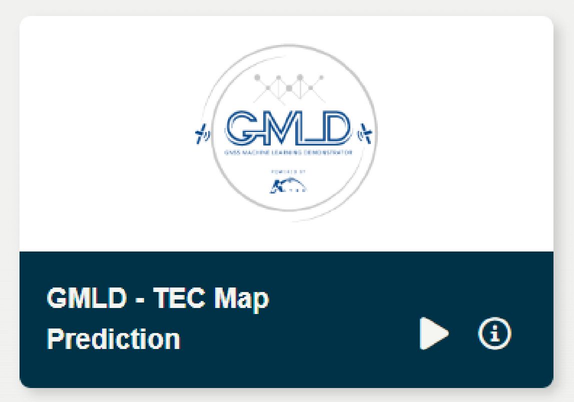
- The main achievement reached at the end of the project is that all proposed GNSS applications identified to investigate machine learning in the GNSS domain have been defined, implemented, validated and the obtained result have been analysed in depth to recognise pro and cons w.r.t. standard approaches.

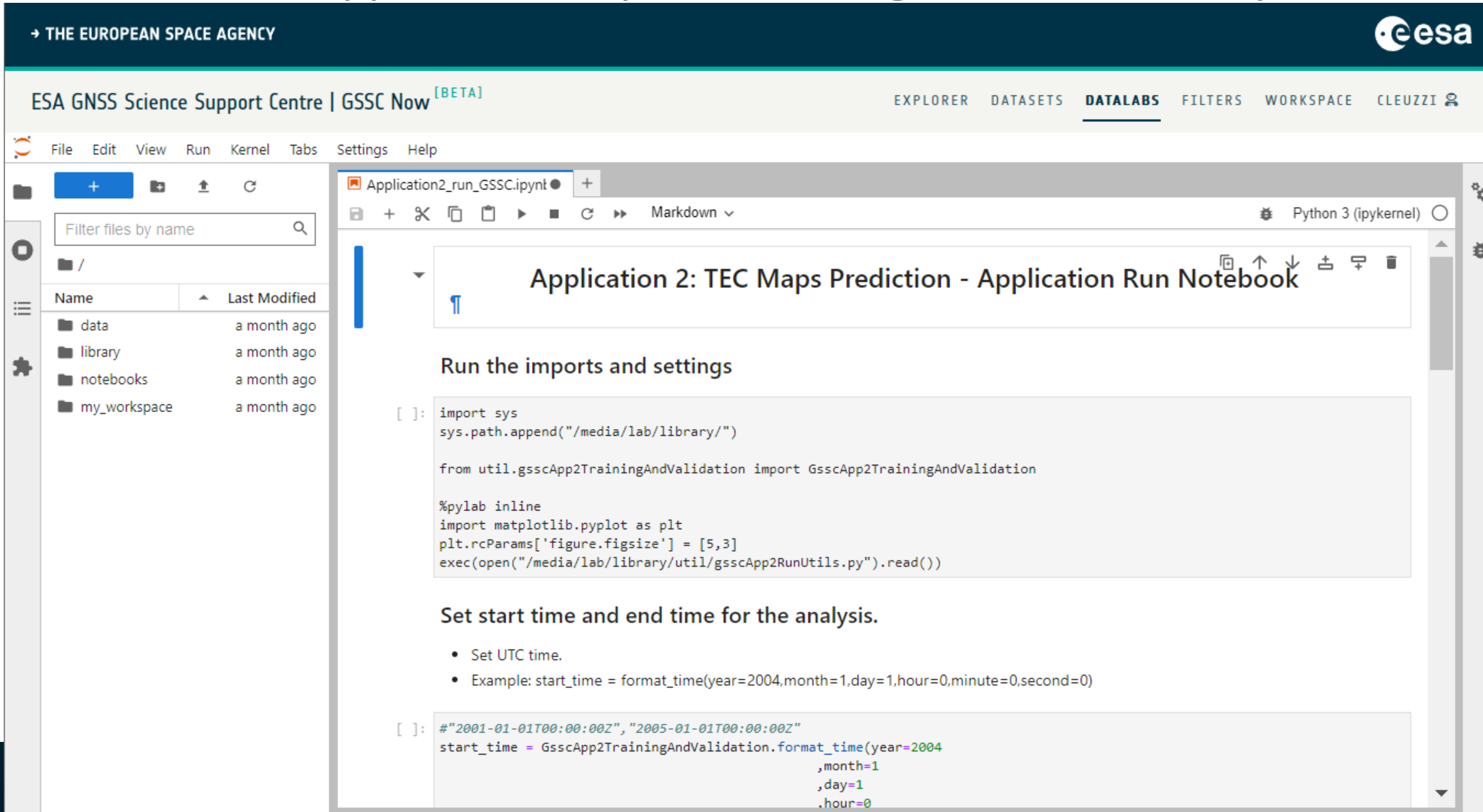
- Applications have been integrated in the GMLD software demonstrator allowing GNSS experts to run them in a simple and effective way. Moreover, the GMLD software has been designed and implemented providing the general data management and ML capabilities as part of the framework therefore it can be easily reused to execute further investigation and implement new applications.

- The GMLD software has reached the TRL 3 target and there are two running instances: one @ALTEC deployed as standalone system and one instance @GSSC deployed as DataLab containers.

GMLD Deployed into GSSC

- **Each GMLD application corresponds to a GSSC DataLab**, which is a container encapsulating different tools
  - In our case, it is Jupyter Hub that exposes the applications in the form of notebooks.
- Each container provides a set of libraries needed by the processing, both custom and off-the-shelf modules.
- The DataLab contains source data made available for the application, any useful configuration/accessory file and one or more notebooks, which processing output is saved into the user persistent workspace.
- For the time being Application 2 has been integrated, while the integration of the other apps is on going.





Applications Achievement

- Investigation has provided several evidences that the usage of ML techniques in several contexts can contribute to improve the availability and quality of GNSS navigation data reducing the impact of the different error sources.
  - **App 2** (TEC map prediction) and **App 3** (SBAS message prediction) have good results therefore they could be used to improve GNSS navigation performance predicting respectively TEC maps and SBAS messages when they are missing in the GNSS receiver.
  - Also the results obtained in the **App 4a** defined to detect scintillation events are positive as the implemented approaches based on ML shows advantages w.r.t. thresholding methods.
  - On the contrary, in several cases applying ML techniques do not provide any advantages as the results are not better that classical approaches. For example, in case of **App 1**, most of the ephemeris parameters is predicted with reasonable errors, while the error provided by the prediction of the mean anomaly value is too big and worse when compared to simple linear regression based on the observed time series of the parameters.
  - **App 4b**, aimed to solve the problem of the small availability of labelled real data by testing unsupervised models (e.g. K-means Clustering), did not provide satisfactory results since the values of both precision and recall is about 0.5 in case of multipath, so only the half of times multipath is detected and still only the half of times returned multipath is reliable.

GMLD Applications Follow-Up

- **Application 1**: Better investigation for estimating the M0 parameter through and approach that embeds some a priori information about the parameter: M0 can be propagated through corrective equations, starting from the last available ephemeris. Then residual between the propagated M0 and the predicted one can be used as a feature for the ML model.
- **Application 2**: A hybrid approach in which the predicted TEC map is fine-tuned by the prediction of some geophysics parameters would be worth to be investigated, with the specific goal of having a prediction suitable also to the use in those areas of the world where the errors are bigger.
- **Application 3**: The results obtained suggest that could be interesting to use ML based prediction for other kind of corrections, even obtained by sources different from the EGNOS messages.
- **Application 4a**: Use the ML which gives better results w.r.t. benchmark methods avoiding false alarms and activate countermeasure based on the attempt of minimizing the effects of the scintillation presence as long as the event is not ended (explore literature methods such as the adaptation of the GNSS receiver parameters)
- **Application 4b**: a dedicated data collection campaigns in a controlled environment where It should be possible to reasonably control the multipath phenomena to some extent and blend in additional complex propagation effects according to the experimental needs. This methodology will be not only suitable to effectively exploit the ML potential, but it could also allow to design and validate more complex algorithms.

GMLD Follow-Up

- The GMLD SW framework is already reusable to easily implement new applications reusing data management and machine learning capabilities. However, it could be extended to as follows:
  - Data Management evolution adding further GNSS data products
  - Machine Learning evolution improving dataset tracking and integrating further ML/DL framework to have better performance
  - Data Visualization evolution with more capabilities to plot GNSS data and result of the ML training, validation and test process
  - Advanced ML Model deployment to create ML model(s) suitable to run inside the GNSS receivers.

- However, it is worth to highlight that machine learning investigation in the GNSS domain is very time-consuming tasks as the effort needed to tune the model to meet the expected result is often unpredictable and the validation process up to the GNSS algorithmic level is complex to setup as it requires additional tools and data.

Question and Answers

Moderator: H. Sobreira

navisp