Monitoring human body motions during earthquakes

GMLD: A TOOL TO INVESTIGATE AND DEMONSTRATE THE USE OF ML IN VARIOUS AREAS OF GNSS DOMAIN

C. Leuzzi[1], A. Nardin[2], A. Madaro[1], R. Messineo[1], F. Dovis[2], D. Valsesia[2], E. Magli[2], H. Sobreira[3], R. Swinden[3]

[1]ALTEC S.p.A., Turin, Italy
chiara.leuzzi@altecspace.it, antonio.madaro@altecspace.it, rosario.messineo@altecspace.it
[2]Politecnico di Torino, DET, Turin, Italy
andrea.nardin@polito.it, fabio.dovis@polito.it, diego.valsesia@polito.it, enrico.magli@polito.it
[3]European Space Agency, ESTEC, Noordwijk, The Netherlands
hugo.sobreira@esa.int, richard.swinden@esa.int

ABSTRACT

This paper presents relevant results achieved during the NAVISP-EL1-035.02 project funded by the European Space Agency, which aimed to investigate the possible uses of Machine Learning (ML) based techniques for the processing of data in the field of Global Navigation Satellite Systems (GNSSs). For this purpose, we explored different kind of data present in the entire chain of the positioning process and different kind of ML approaches. In particular, this paper presents the system architecture and technologies adopted for developing the GNSS ML Demonstrator (GMLD), as well as the approaches and the results obtained for one of the most promising GNSS implemented applications, which is the prediction of daily maps of the ionosphere. Results show how, based on the historical data and the time correlation of the values, ML methods outperformed benchmark methods for the majority of the applications approached, improving the positioning performance at GNSS user level. Since the GMLD has been designed and implemented providing the general data management and ML capabilities as part of the framework, it can be easily reused to execute further investigation and implement new applications.

Index Terms— Machine-learning, MLOps, GNSS, Demonstrator, Ionosphere, Positioning, GSSC

1. INTRODUCTION

The GNSS ML Demonstrator (GMLD) is a modular software tool, developed to investigate possible applications in the Global Navigation Satellite System (GNSS) domain that could benefit from ML capabilities, thus improving some elements of the entire GNSS process to provide better positioning, navigation, and timing (PNT) services. Since, GNSS-based PNT has become fundamental for a wide range of applications, comes out the need to access correct GNSS values. Therefore, being Machine Learning (ML) methods powerful tools demonstrating their value when dealing with large amount of data, the exploitation of GNSSs historical data by means of the implementation of AI based models has come to assume particular importance.

ML-based techniques have already been investigated for GNSSs and also proved their effectiveness on the receiver side [1], [2]. Instead, this study aimed at investigating and demonstrating the use of ML in the broader area of the GNSS domain. In particular, our major goal was 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 correct the latter from predictable error contributions. When such pieces of information are either missing - due to communication or demodulation issues - or outdated, the impact on the estimation of the position, velocity and time (PVT) could be disruptive. The investigated solutions aim at mitigating the impact of these issues by predicting some parameters, when missing, or implementing proper countermeasures at receiver level when the presence of errors in the data or in the measurements is detected/predicted (e.g. excluding some pseudoranges from the PVT computation or adapting the receiver tracking loops parameters). To this purpose, we investigated a set of case-studies of applications that implement ML models able to predict missing information or the presence of possible error sources, and also designed and implemented a modular software tool, the GMLD, which provides the general data management and ML capabilities as part of the framework itself.

2. THE GNSS ML DEMONSTRATOR

The GMLD consists of the software modules implementing four selected applications plus other shared modules that are used across applications. The shared modules provide capabilities to read and write GNSS data and manage ML model definition, training, validation and run. Fig. 1 represents the logical software architecture showing the data sources to be interfaced by the system, the identified software components, the main data flows intra the GMLD software system, the data stores providing storage services and the two different application pipeline configurations. They are the *development* and *production* configurations needed to build the application ML models and to run them inside the applications, respectively.

The input of the GMLD are files of GNSS data products stored in several repositories (MinIO as object storage system and InfluxDB as timeseries database) accessed to retrieve historical datasets. Data are stored in raw format and then pre-processed to feed the ML application pipelines, which consist of the following three stages. The *Feature Extraction* is dedicated to the extraction of information which is provided as input to the ML based models. Of course, the feature selection is strictly dependent by the application. The *Machine Learning Module* implements training and validation, in the development pipeline, and prediction, in the production pipeline. It is built using algorithm implementations available in open source ML frameworks and libraries (e.g. TensorFlow, Keras, etc.), whose architectures were properly instantiated, configured and tuned to reach the expected application performance. The GMLD integrates a *ML Registry*, based on an instance of MLFlow, in order to record all the experiments of the training and validation activities and to catalog the trained models.

doi:10.2760/46796

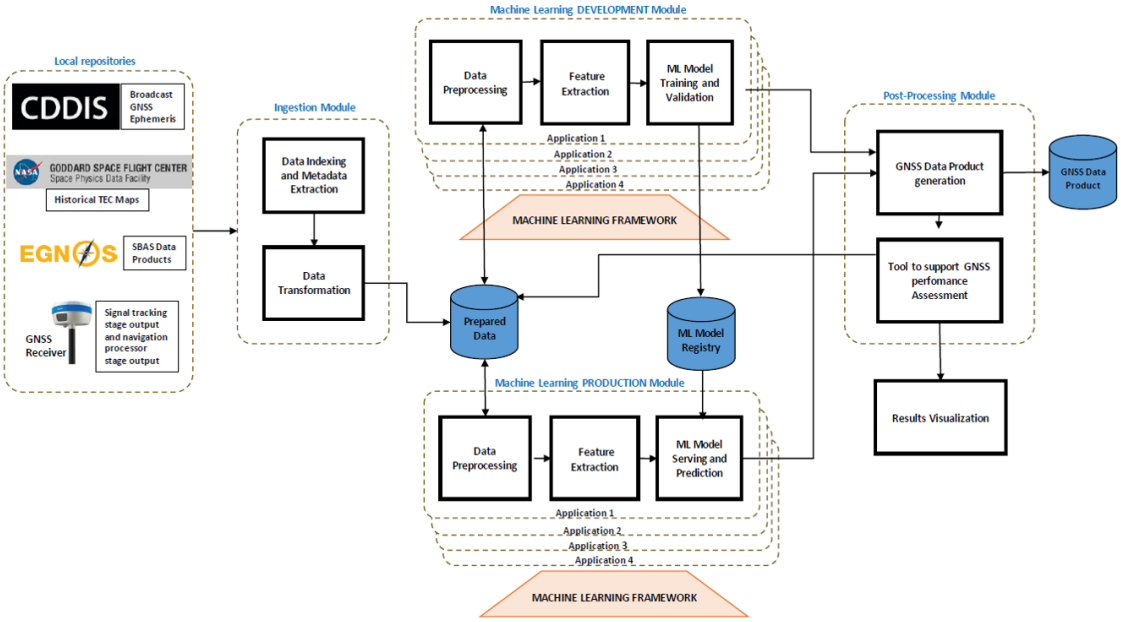

Fig. 1. High level schema of the GMLD.

The *Flow Engine* orchestrates the applications: it allows defining, coordinating, and monitoring both the application pipelines and the other data management flows. It is based on Apache Airflow, so the pipelines are represented by Airflow Direct Acyclic Graphs. The *Post-processing Module* consists of a data product generator and other tools needed to support the GNSS performance assessment. It is in charge of reading the output returned by the ML models and generating the data products according to the GNSS data products format. This step makes the output data ready to be explored and analysed by means external tools that are common in the GNSS domain. Finally, the GMLD has a front-end based on the notebook technologies (Jupyter Notebook). It is used for data exploration, ML models development from preprocessing to model validation, and production application execution. The above mentioned technologies allow to scale the amount of data to be managed and processed by the GMLD, e.g. by increasing the number of data store nodes, the number of processing workers or the computation power granted by the GPUs cluster.

In addition to the standalone software, the applications have been deployed into the GNSS Science Support Centre (GSSC) platform [3]. For this purpose, each application has been refactored in order to be independent from the GMLD software components and integrated as GSSC DataLab tool. In particular, a docker container for each application has been prepared and delivered to the GSSC team that was in charge of the integration.

3. SELECTED GNSS APPLICATIONS

All the identified applications accomplish the purpose of the GMLD, which is the investigation of the use of ML techniques into the GNSS domain to reduce the impacts of the different error sources or the data lack. They are completely developed and live inside the Demonstrator, starting from different type of data source, and explore different factors impacting the navigation message performance. Also, they were selected to cover several classes of ML techniques. They are developed as integration of both generic data modules and specific application modules and their integration is based on data flow control realized through DAG pipeline definition and implementation.

In the following, the four selected applications:

- improving orbit prediction by means of the prediction of ephemeris parameters based on historical data;
- prediction of daily maps of the ionosphere, that is the global Total Electron Content (TEC) map sequences given the previous states of the ionosphere;
- estimation of the SBAS (Satellite-based Augmentation System) correction parameters in the missed messages;
- disturbances and outlier detection (e.g. early scintillation and satellite signals suffering from the multipath effect).

Each application could be seen as a portion or a simplistic case of an extended application that may affect or involve the whole GNSS processing chain. As example, the first application could be extended to the development of an algorithm embedded in a GNSS receiver that can be able to estimate satellites positions when the broadcasted ephemeris are not available, based on the observations and processing the received broadcasted ephemeris.

In the rest of the paper we report the description and results of one of those applications that turned out to be the most promising in terms of performance assessment. Details about other implemented applications can be found in [4], more focused on the applications and their scientific results.

4. TEC MAPS PREDICTION

The *Total Electron Content* in ionosphere corresponds to the total number of electrons integrated along the path from each GNSS satellite to the receiver. It is one of several indicators of ionospheric variability that impacts GNSS signals traveling through this layer of the atmosphere and interacting with the free electrons. The objective of the application is to 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 map can be inferred from the ionosphere previous states. Temporal prediction

doi:10.2760/46796

of TEC maps is analogous to the well-studied problem of future frame prediction in video. This topic has recently received great attention as it is a challenging problem that substantially benefits from the powerful representation learning capabilities of Convolutional Neural Networks (CNN). TEC maps have a simpler content and smoother evolution than arbitrary video sequences, so it is possible to leverage the knowledge and architectures recently developed for video frame prediction ([5], [6]) to solve this task. In the approach proposed, TEC map is treated as a whole and the goal is to predict the entire map, exploiting the spatial and temporal correlation among the different points of the map grid.

4.1. Model Implementation

The *ResNet* backbone architecture was chosen for the application, as it is used in a wide variety of regression problems concerning images/videos: they are known to perform better than architectures without skip connections, as they enable each layer to learn just what is not already present in the input, rather than learning a model of the whole data [7]. Different solution were tested in order to exploit the temporal correlation; in this paper the final configuration of the model is reported. It concatenates the past TEC maps from T to T-K hours, therefore even maps at hours-of-the-day that are different from the target one are exploited. This choice allows better exploitation of short-term spatial and temporal correlation patterns thanks to the smooth evolution of TEC over temporally subsequent maps. The ResNet based model, shown in Fig. 2 is composed of alternating a 3D convolutional layer (C), batch normalization (N), and ReLU (R) non-linearity. Two residual blocks of C, N, R are used to build a deeper model. Convolution over the time dimension allows to track the smooth temporal evolution of the features extracted by spatial convolution. The output is an image with 2 channels, which are summed with the

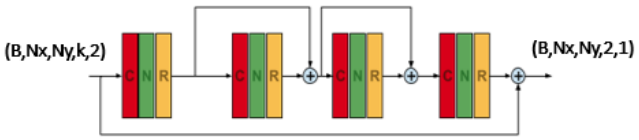

Fig. 2. Neural network architecture, channel last convention.

image representing the TEC map at T-22 and T-20 hours (H), respectively, to generate the outputs that are the predictions for time T+2H and T+4H. The scope of this global skip connection is the exploitation of the strong correlation between the map of 24 hours before and the future maps. The filters in the convolutional layers ensure that a sufficiently large receptive field is available to extract meaningful spatial and temporal features. The model was trained by minimizing the mean squared error (MSE) between the predictions and the ground truth (GT) TEC maps at T+2H and T+4H. The model has a lot less trainable parameters than what is typical for video sequences, so more sophisticated models in the video frame prediction literature cannot be reliably trained or would not provide significant advantages due to overfitting.

In addition, we explored the possibility to use also different geomagnetic indices, providing additional information on the ionosphere status. In particular, Sunspot number and Kp-index time series were added to the TEC map time series, so that the NN can exploit the correlation of the indices over time. Several tests were conducted in order to estimate the best subset of features among all these parameters. The final model manages the presence of these additional parameters, i.e. the sunspot number in our final case. Input data are prepared by creating additional layers of repeated values to be stacked with the TEC maps, according to the scheme in Fig. 3. Nx = 71 and Ny = 73 are the dimensions of the TEC map, K is the number of previous TEC maps to be used as input for the model. At each time step, a map containing the repeated values of the series is created and added as an additional layer. Combined data have thus input size (Nx, Ny, K, m) where m = 2 is the number of features of the ML model in the channels-last convention, i.e. TEC and sunspot number. Following this approach, it is easy to scale the input size when more features are considered.

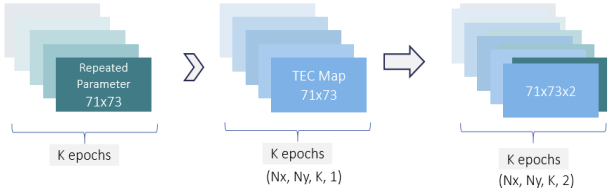

Fig. 3. Input data preparation

4.2. Results

Dataset selection for the experiment was realized by considering the maximum period of the Solar Cycles 23 (November 2001) and 4-year datasets (i.e., 2001-2004) have been created by processing downloaded data. Datasets were divided into training (70%), validation (20%), and test (10%) sets.

The numbers of the past (K) and future observations are customizable, but the tests executed focused on 24 and 36 hours of past data. The majority of tests focused on the prediction of T+2H and T+4H, since we concluded to focus on short term predictions, which are more meaningful. The different model architectures, together with the hyperparameters values, have been compared by considering the values of the functional metrics, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). The metrics were computed on the whole validation and test datasets but also on each step forward of the prediction, in order to have a look on the mean model performance, but also to see how different the models behave when more hours in advance are predicted.

In Fig. 4 it is shown a comparison among three models with same configuration, apart the different data input: only TEC map, TEC and Kp-index, TEC and Sunspot number. In the first two graphs, we see that the Loss function (MSE) has always a good trend, both the training and validation cases. In the latter one, green and orange lines show that the errors on validation set are lesser when an additional parameter is considered, and the Kp-index seems to be the more meaningful. At the same time, on the last graph, we have the time to train on the abscissa: the green line reaches a convergence in less than the half time of the orange line (based on a GPU usage). Fig. 5 reports the comparison among the errors obtained for each step ahead of the prediction and for dataset (validation and test). It can be observed that the error at T+4H is double with respect to T+2H and that the validation results are confirmed by the metrics calculation on test dataset.

The analysis of the overall TEC prediction performance has been carried out with respect to the ground truth TEC maps in the test set. The estimated maps provided by the ML model have been used to compute the average vertical ionospheric bias error (AVIBE) affecting the pseudorange. This error has been computed for each latitude and longitude pair, on a grid of 71x73 points, averaging over 4279 subsequent observations of the TEC map. The resulting AVIBE was drawn onto an Eckert map projection. Fig. 6 shows the AVIBE values for a T+2H prediction of the TEC map. The whole error map can be aggregated into a mean absolute error

doi:10.2760/46796

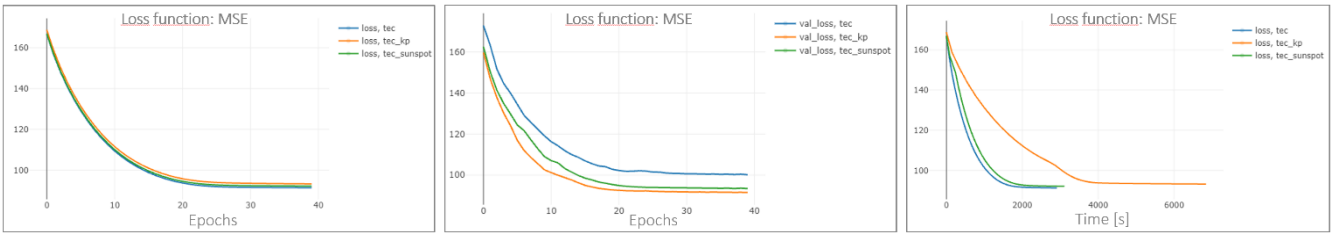

Fig. 4. Training and Validation Loss functions.

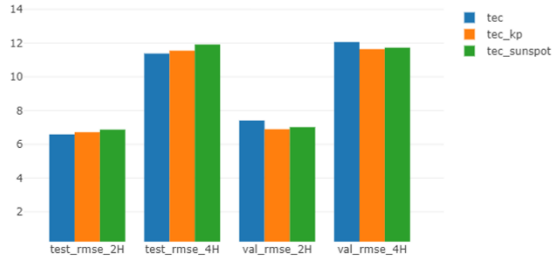

Fig. 5. RMSE values computed on validation and test datasets.

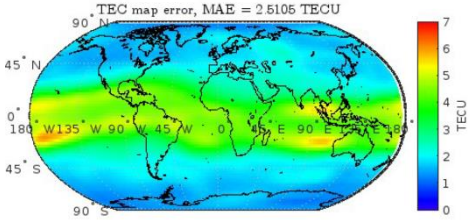

Fig. 6. Average vertical ionospheric bias error, T+2H, performed over consecutive snapshots.

of 2.5105 TECU, equals to a 40.8cm of vertical ionospheric bias.

Moreover, GMLD implements a second level of performance assessments for each application. In this case it is based on the comparison with two benchmark methods. The first one consists in using the last available map, which is the map at time T, as prediction of the future ones. The second method consists in using the map at 24H before, e.g. the map at time T-22H as prediction of map at T+2H. The errors evaluated when the GT data are predicted with ML models are compared with the ones evaluated when a benchmark method is used. Errors are shown through the front-end, both per map and per map points, to see which regions have worse predictions.

In order to explore further the benefit from the usage of the predicted data, in addition to these comparison outputted by the GMLD itself, the assessment of the performance has been extended evaluating the impact on the slant TEC computation within the grid. We considered different user positions on Earth, in order to fulfill the analysis of diverse use cases. The method provided satisfactory results compared to the input data complexity. Other methods in literature aimed at predicting the different variables used to build TEC maps, but the difficulty in finding historical time series of needed data makes the ML based approach here presented worthy of interest. More details about results are reported in [4].

5. CONCLUSIONS

The GMLD, presented in this paper through its architecture and one of its applications, turned out to accomplish the main goal of enabling the data scientists and domain experts to investigate possible applications in the GNSS domain that could benefit from ML capabilities. Thanks to its modularity, to the data management capability, to the integration of useful tools to manage ML dev-ops, and to the pipeline execution and monitoring engine, it enables the users to train deep models, validate them through customized data analysis and benchmark methods comparison, and finally to release full GNSS applications. Moving to these latter, the investigation has provided evidences that the usage of the ML techniques in several GNSS contexts can contribute to improve the data products reducing the impact of the different error sources or data lack. Results presented in the previous chapter about the TEC map prediction show the capability to forecast at different time steps ahead, thus making this method applicable in a potential GNSS service, or in a receiver unable to get updated information neither from a SBAS satellite nor from a third party service.

ACKNOWLEDGEMENT

This work was developed in the framework of the activity “Machine-Learning to model GNSS systems” funded by the European Space Agency. The project, NAVISP-EL1-035.02, was carried out in the NAVISP Element 1, which is dedicated to technology innovation of the European industry in the PNT sector.

REFERENCES

[1] P. Borhani-Darian et al, “Deep neural network approach to GNSS signal acquisition,” in 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), 2020, pp. 1214 1223.

[2] A. Nardin, T. Imbiriba and P. Closas, "Jamming Source Localization Using Augmented Physics-Based Model," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095731

[3] V. Navarro et al., “ESA GNSS science support centre a world-wide reference GNSS environment for scientific communities,” in Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), September 2020, pp. 1174–1199.

[4] A. Nardin et al., "On the Use of Machine Learning Algorithms to Improve GNSS Products," 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 2023, pp. 216-227, doi: 10.1109/PLANS53410.2023.10139920.

[5] H. Choi and I. V. Baji´c, “Deep frame prediction for video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 7, pp. 1843–1855, 2020.

[6] M. Oliu, J. Selva, and S. Escalera, “Folded recurrent neural networks for future video prediction,” in Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIV. Berlin, Heidelberg: Springer-Verlag, 2018, p. 745–761.

[7] ——, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.