MLCommons Science Research Working Group Benchmarks

Version 1, October 21, 2021

1 Benchmarks

1.1 CloudMask (Segmentation)

1.1.1 Description

Estimation of sea surface temperature (SST) from space-borne sensors, such as satellites, is crucial for a number of applications in environmental sciences. One of the aspects that underpins the derivation of SST is cloud screening, which is a step that marks each and every pixel of thousands of satellite images as containing cloud or clear sky, historically performed using either thresholding or Bayesian methods.

1.1.2 Benchmark objectives

1) The objective of CloudMask benchmark is to try to improve accuracy compared to the one produced by thresholding or Bayesian methods.
2) Demonstrate the scalability of distributed training by running the application on many GPUs.

1.1.3 Data description

This benchmark focuses on using a machine learning-based model for masking clouds, in the Sentinel-3 satellite, which carries the Sea and Land Surface Temperature Radiometer (SLSTR) instrument. More specifically, the benchmark operates on multispectral image data. The baseline implementation is a variation of the U-Net deep neural network. The benchmark includes two datasets of *cloud_slstr_ds1* and *cloud_slstr_ds2*, with sizes of 180GB and 4.9TB, respectively. Each dataset is made up of two parts: reflectance and brightness temperature. The reflectance is captured across six channels with the resolution of 2400 x 3000 pixels, and the brightness temperature is captured across three channels with the resolution of 1200 x 1500 pixels.

1.1.4 Data location

The data is located at the STFC object store, by running the aws command:

aws s3 --no-sign-request --endpoint-url https://s3.echo.stfc.ac.uk sync s3://sciml-datasets/en/cloud_slstr_ds1 ./

aws s3 --no-sign-request --endpoint-url https://s3.echo.stfc.ac.uk sync s3://sciml-datasets/ins/cloud_slstr_ds1 ./

1.1.5 Reference implementation

The code can be downloaded from the GitHub repository:

https://github.com/stfc-sciml/sciml-bench/tree/master/sciml_bench/benchmarks/slstr_cloud

Detailed documentation can be found at: https://github.com/stfc-sciml/sciml-bench

1.1.6 Implementation results

The CloudMask benchmark was implemented on Pearl (STFC) and Summit (ORNL) platforms. The runtime results are presented in Figure 1 and Figure 2.

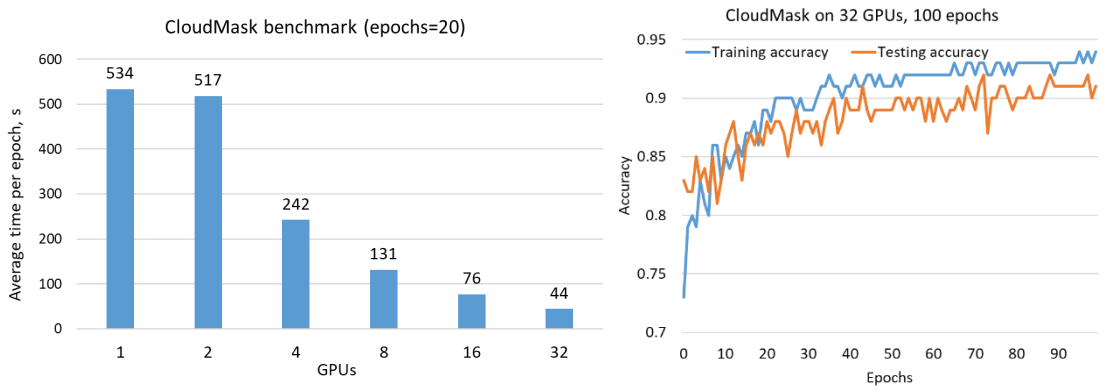


Figure 1 - Scalability and accuracy on Pearl (DGX-2)

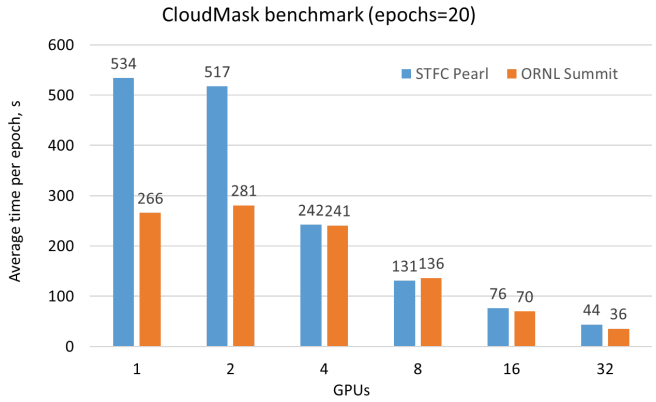


Figure 2 - Pearl vs Summit

1.1.7 Summary table

CloudMask (Segmentation)	**Current status**
Description	✓
Objectives	✓
Data description	✓
Data location	✓
Reference implementation	✓

Implementation results	✓

1.1.8 CloudMask (Segmentation) Appendix – info for the website

1. **Scientific objective(s):**
    - Objective: Compare the accuracy produced by the Neural Network with the accuracy of a Bayesian method.
    - Formula: Weighted Binary Cross Entropy of validation data
    - Score: 0.9 for convergence
2. **Data:**
    - Download: The data from the STFC server can be accessed by running the aws command:

      aws s3 --no-sign-request --endpoint-url https://s3.echo.stfc.ac.uk sync s3://sciml-datasets/en/ cloud_slstr_ds1 ./

    - Input Size: 180GB
    - Training samples: 15488
    - Validation samples: 3840
3. **Baseline implementation**
    - Model: U-Net
    - Reference Code: https://github.com/stfc-sciml/sciml-bench/tree/master/sciml_bench/benchmarks/slstr_cloud
    - Run Instructions: https://github.com/stfc-sciml/sciml-bench/blob/master/README.md
    - Time-to-solution: 180GB dataset runs 59 min on DGX-2 with 32 V100 GPUs
4. **Hardware systems explored and performance:**
    - Summit: 180 GB dataset runs 48 min on 32 GPUs
    - RAL: 180 GB dataset runs 59 min on DGX-2 with 32 V100 GPUs
5. **Example improvements:**
    - Investigate the use of unsupervised segmentation

1.2 STEMDL (Classification)

1.2.1 Description

State of the art Scanning Transmission Electron Microscopes (STEM) produce focused electron beams with atomic dimensions and allow to capture diffraction patterns arising from the interaction of incident electrons with nanoscale material volumes. Backing out the local atomic structure of said materials requires compute- and time-intensive analyses of these diffraction patterns (known as convergent beam electron diffraction, CBED). Traditional analyses of CBED requires iterative numerical solutions of partial differential equations and comparison with experimental data to refine the starting material configuration. This process is repeated anew for every newly acquired experimental CBED pattern and/or probed material.

1.2.2 Benchmark objectives

The objectives are based on the list of challenges described at: 2020: Challenge 2 « SMC Data Challange 2021 (ornl.gov)

1) Perform exploratory data analysis on both CBED patterns and materials properties to summarize data characteristics.
2) Develop an ML algorithm for space group classification of CBED data.
3) Implement proper ML techniques to overcome data/label imbalance and show how it affects the performance of the ML algorithm in (1)
4) Implement an ML algorithm for multi-task prediction of a space group in addition to other material structural properties and show how it affects the performance of the ML algorithm in (1).

1.2.3 Data description

A data sample from this data set is given by a 3-d array formed by stacking various CBED patterns simulated from the same material at different distinct material projections (i.e. crystallographic orientations). Each CBED pattern is a 2-d array with float 32-bit image intensities. Associated with each data sample in the data set is a host of material attributes or properties which are, in principle, retrievable via analysis of this CBED stack. Of note are (1) 200 crystal space groups out of 230 unique mathematical discrete space groups and (2) local electron density which governs material's property. A more detailed description of data can be found at: 10.13139/OLCF/1510313 (ornl.gov)

1.2.4 Data location

Data URL: https://doi.ccs.ornl.gov/ui/doi/70

1.2.5 Reference implementation

This benchmark consists of two tasks: classification for crystal space groups and reconstruction for local electron density, the baseline implementation of which are provided in [4] and [5]. The reference implementation can be found at:

a) https://github.com/at-aaims/stemdl-benchmark

b) https://code.ornl.gov/jqyin/stemdl-benchmark

1.2.6 Implementation results

In this benchmark, we used newly developed multi-GPU and multi-node electron scattering simulation codes [1] on the Summit supercomputer to generate CBED patterns from over 60,000 materials (solid-state materials), representing nearly every known crystal structure. A scaled-down version of this data [2] is used for one of the data challenges [3] at SMC 2020 conference, and the overarching goals are to: (1) explore the suitability of machine learning algorithms in the advanced analysis of CBED and (2) produce a machine learning algorithm capable of overcoming intrinsic difficulties posed by scientific datasets.

From Junqi's email: The code is running on GPUs, and we did a throughput scaling up to 3000 GPUs.

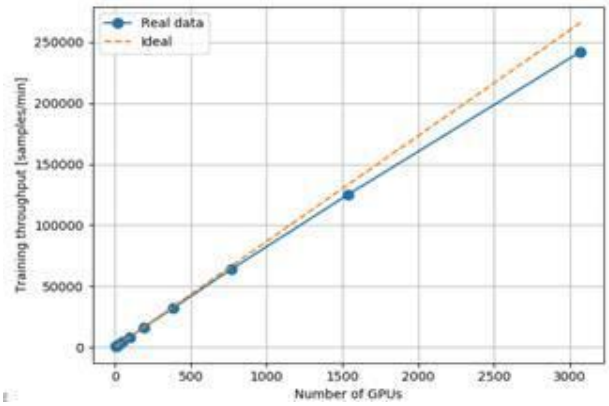


Figure 1 - Scalability study of STEMDL running on Summit at ORNL

It is measured in trained samples/min. As is for many DL applications though, the scaling of time-to-solution is another matter b/c large-batch training generally hurts accuracy.

1.2.7 Summary table

STEMDL	Current status
Description	✓
Objectives	✓
Data description	✓
Data location	✓
Reference implementation	✓
Implementation results	✓

1.2.8 References

[1] N Laanait, J Yin, "NAMSA", 10.11578/dc.20201001.90, 2019

[2] N Laanait, A Borisevich, J Yin, "Database of Convergent Beam Electron Diffraction Patterns for Machine Learning of the Structural Properties of Materials", 10.13139/OLCF/1510313, 2019

[3] N Laanait, J Yin, A Borisevich, "Towards a Universal Classifier for Crystallographic Space Groups", SMC data challenges, 2020

[4] Jin Pan, "Probability Flow for Classifying Crystallographic Space Groups", SMC 2020

[5] N Laanait, J Romero, J Yin, M Young, S Treichler, V Starchenko, A Borisevich, A Sergeev, M Matheson, “Exascale deep learning for scientific inverse problems”, arxiv:1909.11150, 2019.

1.3 CANDLE-UNO

1.3.1 Description

CANDLE (Exascale Deep Learning and Simulation Enabled Precision Medicine for Cancer) project aims to implement deep learning architectures that are relevant to problems in cancer. These architectures address problems at three biological scales: cellular (Pilot1 P1), molecular (Pilot P2) and population (Pilot3).

Pilot1 (P1) benchmarks are formed out of problems and data at the cellular level. The high level goal of the problem behind the P1 benchmarks is to predict drug response based on molecular features of tumor cells and drug descriptors. Pilot2 (P2) benchmarks are formed out of problems and data at the molecular level. The high level goal of the problem behind the P2 benchmarks is molecular dynamic simulations of proteins involved in cancer, specifically the RAS protein. Pilot3 (P3) benchmarks are formed out of problems and data at the population level. The high level goal of the problem behind the P3 benchmarks is to predict cancer recurrence in patients based on patient related data.

1.3.2 Benchmark objectives

Uno application from Pilot1 (P1): The goal of Uno is to predict tumor response to single and paired drugs, based on molecular features of tumor cells across multiple data sources. Combined dose response data contains sources: [‘CCLE’ ‘CTRP’ ‘gCSI’ ‘GDSC’ ‘NCI60’ ‘SCL’ ‘SCLC’ ‘ALMANAC.FG’ ‘ALMANAC.FF’ ‘ALMANAC.1A’].

1.3.3 Data description

Combined dose response data contains sources: ['CCLE' 'CTRP' 'gCSI' 'GDSC' 'NCI60' 'SCL' 'SCLC' 'ALMANAC.FG' 'ALMANAC.FF' 'ALMANAC.1A']

Summary of combined dose response by source:

Source	Growth	Sample	Drug1	Drug2	MedianDose
ALMANAC.1A	208605	60	102	102	7.000000
ALMANAC.FF	2062098	60	92	71	6.698970
ALMANAC.FG	1415772	60	100	29	6.522879
CCLE	93251	504	24	0	6.602060
CTRP	6171005	887	544	0	6.585027
GDSC	1894212	1075	249	0	6.505150
NCI60	18862308	59	52671	0	6.000000
SCL	301336	65	445	0	6.908485
SCLC	389510	70	526	0	6.908485
gCSI	58094	409	16	0	7.430334

Combined raw dose response data has 3070 unique samples and 53520 unique drugs

1.3.4 Data location

The datasets are publicly available at:

http://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot1/combo/

These are directly downloaded with the available scripts. To ease, a static dataset is prebuilt and available at:

http://ftp.mcs.anl.gov/pub/candle/public/benchmarks/Pilot1/uno/top_21_auc_1fold.uno.h5

1.3.5 Reference implementation

Uno implements a deep learning architecture with 21M parameters in TensorFlow framework in Python. The code is publicly available on GitHub. The script in this repository downloads all required datasets. The primary metric to evaluate this applications is throughput (samples per second).

More details on running Uno can be found here:

https://github.com/ECP-CANDLE/Benchmarks/tree/master/Pilot1/Uno

1.3.6 Implementation results

The primary metric use to evaluate Uno is 'throughput (samples/second)'. On ThetaGPU with Nvidia A100 GPUs, the throughput numbers with varying batch sizes on a single GPU are presented in the following figure.

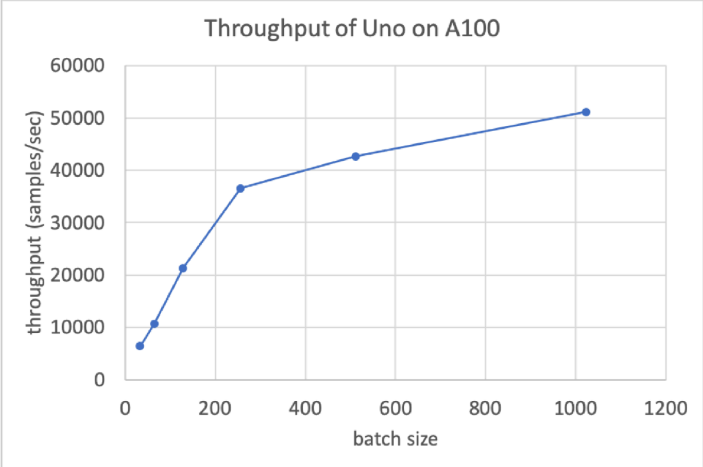


Figure 1 - Throughput of CANDLE-UNO benchmark

1.3.7 Summary table

CANDLE-UNO	Current status
Description	✓
Objectives	✓
Data description	✓
Data location	✓
Reference implementation	✓
Implementation results	✓

1.4 TEvolOp Earthquake Forecasting

1.4.1 Description

Time series are seen in many scientific problems and many of them are geospatial – functions of space and time and this benchmark illustrates this type. Some time series have a clear spatial structure that for example strongly relates nearby space points. The problem chosen is termed a spatial bag where there is spatial variation but it is not clearly linked to the geometric distance between spatial regions. In contrast, traffic-related time series have a strong spatial structure. We intend benchmarks that cover a broad range of problem types.

1.4.2 Benchmark objectives

The scientific objective is to improve the quality of Earthquake forecasting in a region of Southern California. This is quantified in a few numbers -- the Nash Sutcliffe Efficiency NSE and in the visualization of predicted versus observed earthquake activity. NSE is a measure of a difference between the variation of the data over time with the discrepancy between observation and prediction so that predicting the observation to be the mean gives a default NSE of 0. The presentation https://docs.google.com/presentation/d/1nTM-poaFzrT_KBB1J7BlZdOMEIMTu-s48mcBA5DeP30/edit?usp=sharing goes into more detail

1.4.3 Data description

The earthquake data comes from USGS and we have chosen a 4 degrees of Latitude (32 to 36 N) and 6 degrees of Longitude (-120 to -114) region covering Southern California. The data runs from 1950 to the present day and is presented as events: magnitude, ground location, depth, and time. We have presented the data in time and space bins. The time interval is daily but in our reference models, we accumulate this into fortnightly data. Southern California is divided into a 40 by 60 grid of 0.1 by 0.1-degree “pixels” which corresponds roughly to squares with an 11 km side, The dataset also includes an assignment of pixels to known faults and a list of the largest earthquakes in that region from 1950 until today. We have chosen various samplings of the dataset to provide both input and predicted values. These include time ranges from a fortnight up to 4 years. Further, we calculate summed magnitudes and depths and counts of significant quakes (magnitude > 3.29). Other easily available quantities are powers of quake energy (using Energy ~ $10^{1.5m}$ where m is magnitude). Quantities are “Energy averaged” when there are multiple events in a single space-time bin except for simple event counts.

1.4.4 Data location

Data URL: https://earthquake.usgs.gov/earthquakes/search/
The data is processed from open USGS data on earthquakes from 1950 to the present day. It can be found at https://drive.google.com/drive/folders/1wz7K2R4gc78fXLNZMHcaSVfQvIpIhNPi?usp=sharing. This data is binned in space and time whereas USGS records individual events at discrete space-time points

1.4.5 Reference implementation

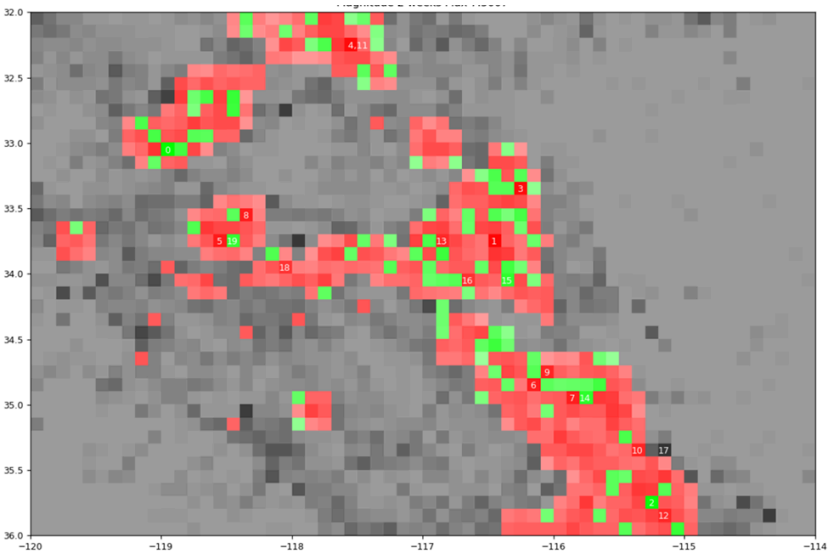
3 distinct deep learning reference implementations will be provided with initial results to be found at the presentation given in a). The implementations are

1) **LSTM** Pure recurrent Neural Network -- two-layer LSTM
2) **TFT** A version of the Google Temporal Fusion Transformer TFT that uses two distinct LSTM's -- one each for Encoder and Decoder with a temporal (attention-based) transformer
3) **Hybrid Transformer** A hybrid model that was built at the University of Virginia with a space-time transformer for the Decoder and a two-layer LSTM for the encoder.
Each predicts NSE and visualizations illustrated

Details of the current reference models can be found at:

a) https://docs.google.com/presentation/d/1ykYnX0uvxPE-M-c-Tau8irU3IqYuvj8Ws8iUqd5RCxQ/edit?usp=sharing
b) https://www.researchgate.net/publication/346012611_DRAFT_Deep_Learning_for_Spatial_Time_Series

1.4.6 Implementation results



A variety of results are available See link in section 1.4.2. A set of predictions just considered the 500 most active pixels in the 2400 in the full dataset. Of these 400 were used for training and 100 for validation. The 2400 0.1 by 0.1 degree subregions analyzed (pixels) are illustrated in figure with 400 red as training and green 100 as validation pixels and the pixel intensity corresponding to earthquake activity.

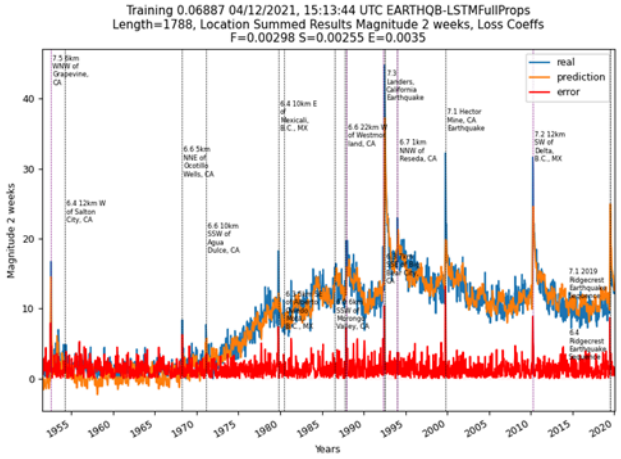
$$NSE = 1 - \frac{\sum_{t=1}^{T} \left(Q_m^t - Q_o^t\right)^2}{\sum_{t=1}^{T} \left(Q_o^t - \overline{Q}_o\right)^2}$$

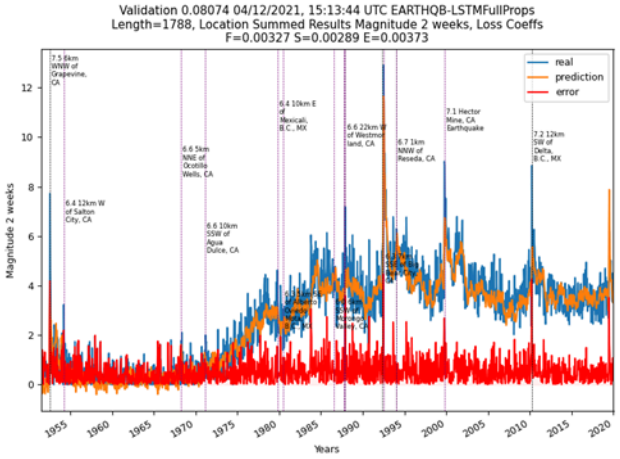
The Nash Sutcliffe Efficiency NSE is given on the left where quantity Q_m comes from model and Q_0 is observed. We use the normalized NSE, NNSE = 1/(2-NSE) runs between 0 (bad) and 1 (perfection) and results are for activities (Q) summed over all 400/100 locations are shown in the table. Note all time interval bins use "log-energy" where events in bin are converted to energy; energies are summed and converted

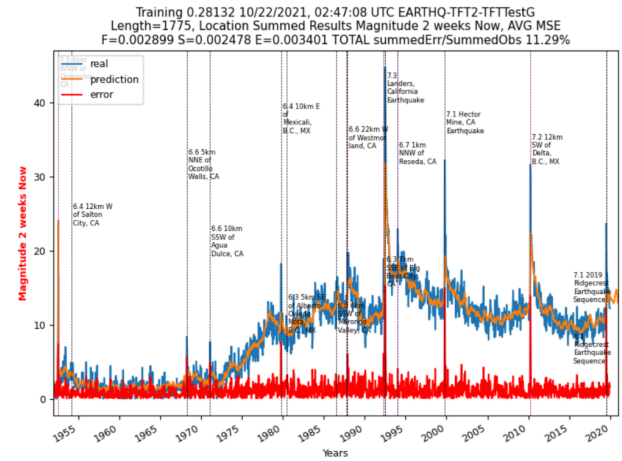
back to a magnitude. The time period listed corresponds to the forward prediction of that length.

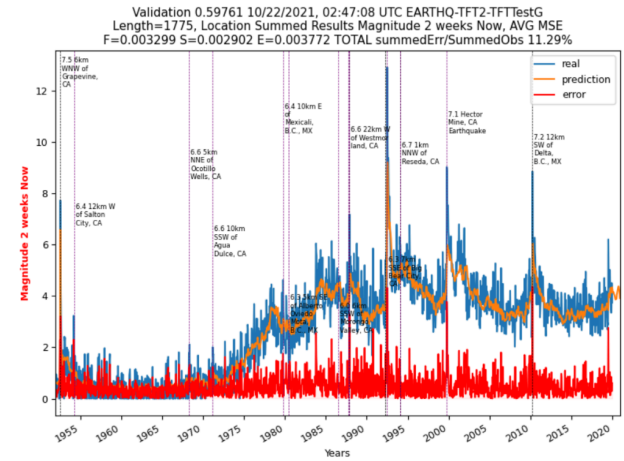
	Normalized Nash Sutcliffe Efficiency NNSE					
Time Period	**LSTM Train**	**TFT Train**	**Transformer Train**	**LSTM Val**	**TFT Val**	**Transformer Val**
2 weeks	0.902	0.931	0.893	0.869	0.885	0.856
4 weeks	0.896		0.916	0.866		0.883
2 months	0.887		0.913	0.865		0.881
3 months	0.925	0.976	0.919	0.893	0.922	0.881
6 months	0.95	0.972	0.954	0.9	0.882	0.896
Year	0.923	0.976	0.955	0.865	0.853	0.876
2 years	0.928		0.937	0.855		0.83
4 years	0.937		0.921	0.817		0.77

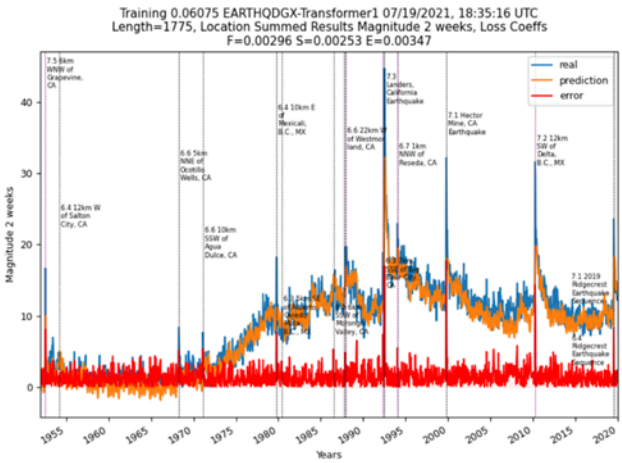
Time dependent results for 2 weeks (training on left, validation on the right) are given below for 3 methods in the order: LSTM, TFT, Hybrid Transformer.

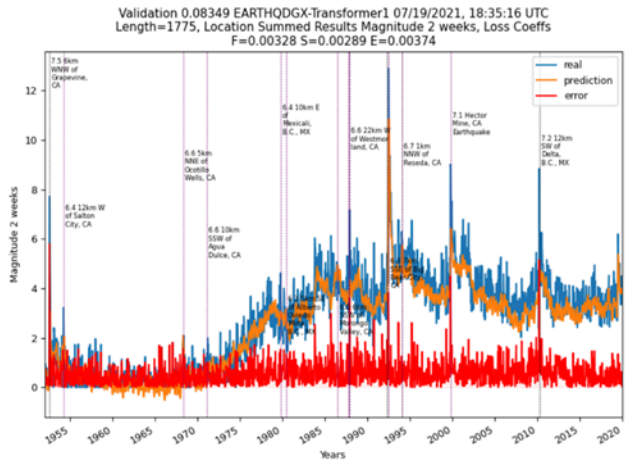




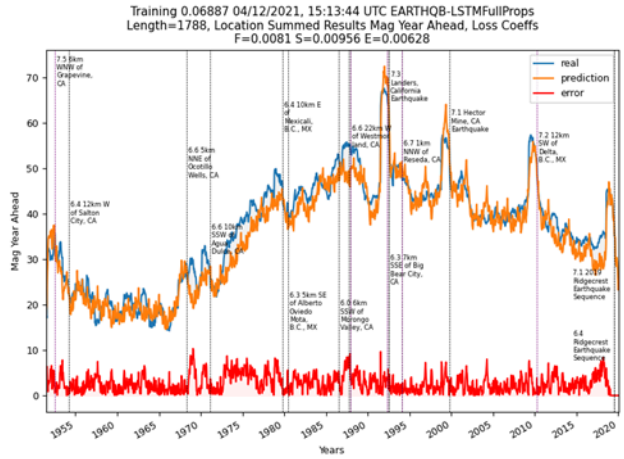


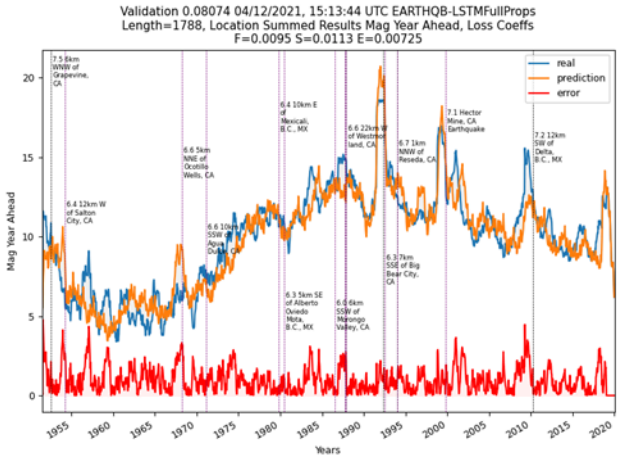


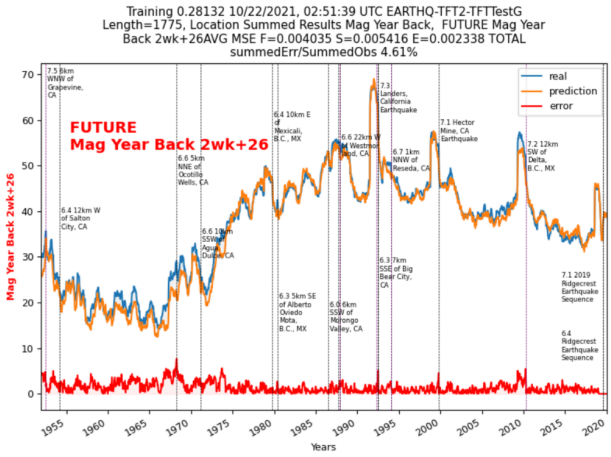


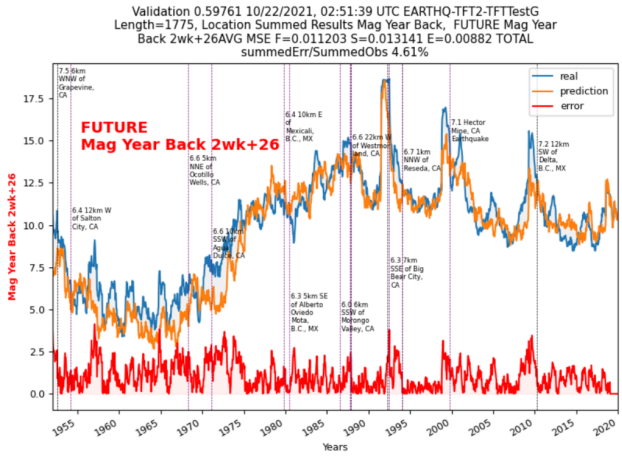


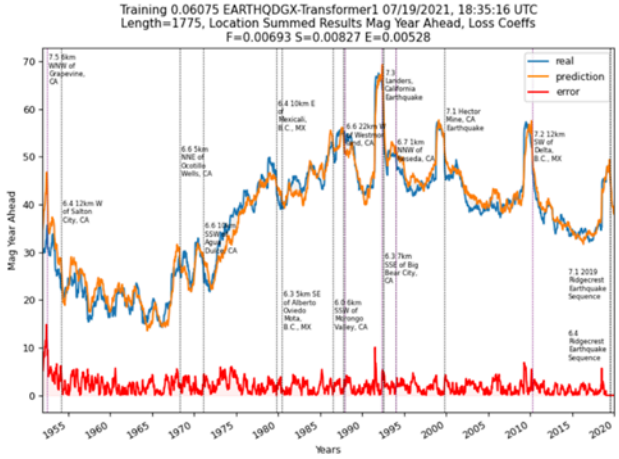
And for predicting one year ahead

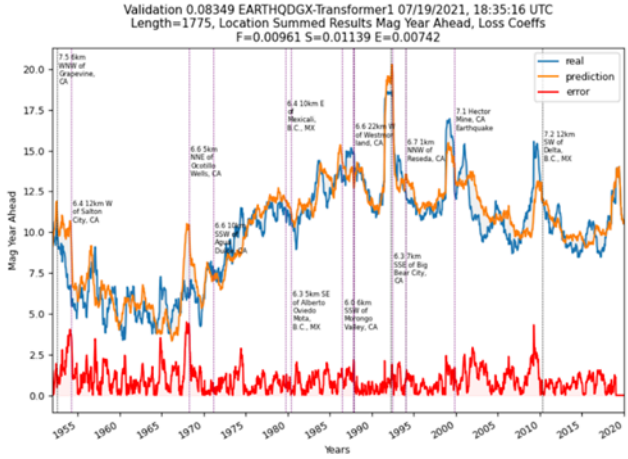












1.4.7 Summary table

TEvolOp Earthquake Forecasting	Current status
Description	✓
Objectives	✓
Data description	✓

Data location	✓
Reference implementation	✓
Implementation results	✓