Filling gaps in ocean satellite data

Aida Alvera-Azcárate & Alexander Barth

GHER, University of Liège
Belgium

The GHER

Physical oceanography group at the University of Liège (Belgium)

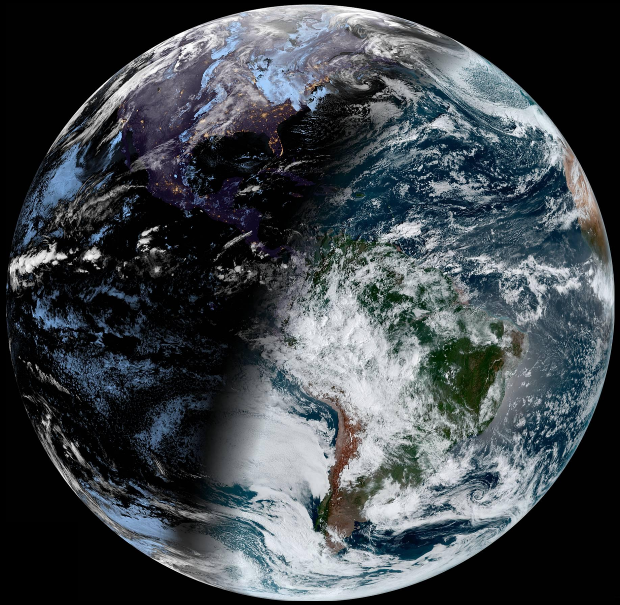
Main research activities

- Ocean modelling
- Data assimilation
- Development & application of data analysis techniques
  - DIVA, DIVAnd
  - DINEOF
  - DINCAE

Master in Oceanography, Erasmus+ Master MER2030

Organizers of the Liège Colloquium in Ocean Dynamics

GOES-East



Meteosat

Himawari-8

Do you see beautiful marble pictures of the Earth?

… I see clouds

The problem

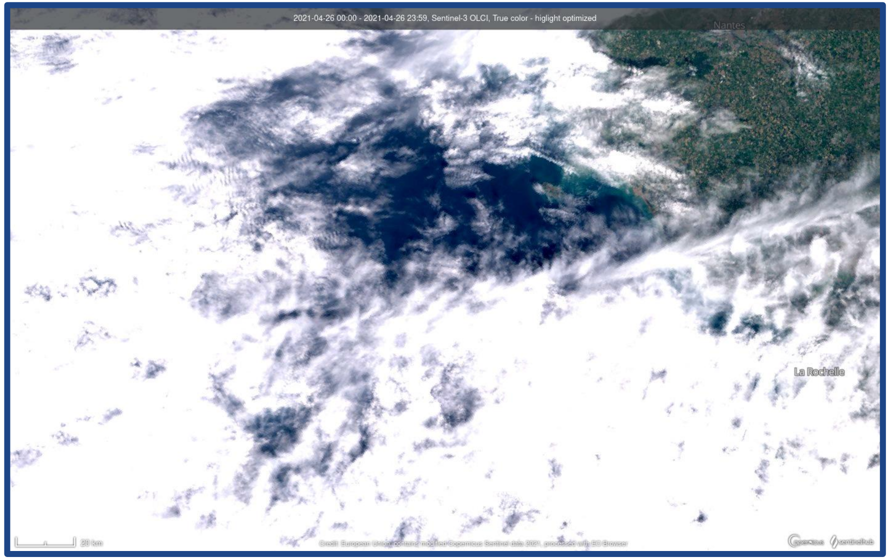
Satellite sensors measuring in the visible and infrared wavelengths can’t “see” through clouds, dust, haze...

As a result, satellite data for variables like sea surface temperature, chlorophyll concentration, suspended sediments, etc, are heavily affected by missing data

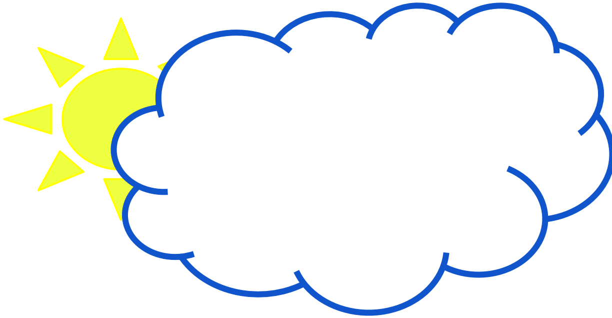
- Latitudinal and seasonal variability in the % of missing data

What you asked for...

... what you get



Interpolating missing data in satellite datasets



- Clouds have been **always** a problem

- Luckily they move around: spatio-temporal analyses can help

- Several approaches have been used to remove or minimise the effect of clouds, e.g. :

    - Compositing (loss of spatial/temporal resolution)
    - Interpolation techniques (e.g. Optimal Interpolation or Objective Analysis)
        Gridded field = First guess + weighted sum of observations

- Typically previous knowledge of the characteristics of the interpolated variable are needed → **subjectivity**

- Beckers & Rixen (2003) develop a method to **estimate missing information from the EOF basis calculated from the data**

    - EOFs provide a series of main modes of variability, classified by importance
    - Uses an SVD method to calculate the EOFs (provides best truncated EOF matrix)
    - For a data matrix X → $X = USV^T$

EOFs should **not** be calculated with missing data

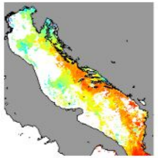
- SVD assumes data matrix X is perfectly and completely known
- If covariance matrix ($C = X^T X$) is only calculated on available data:
    - C no longer semipositive defined
    - Eigenvalues can be negative: classification of EOFs by their importance no longer possible

In short: we're calculating EOFs (that shouldn't be used when missing data) to find the values of the missing data

How does that work??

DINEOF

DINEOF (Data Interpolating Empirical Orthogonal Functions)



1st: Demeaned matrix: missing data flagged and set to zero

Some data are set aside for cross-validation

2nd: EOF decomposition with N=1 EOF

Calculate missing values:

$$\boxed{X_{i,j} = \sum_{p=1}^{k} \rho_p (u_p)_i (v_p^T)_j}$$

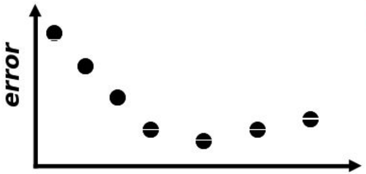
Improved guess for missing values

Convergence: {*best value for missing data with 1 EOF*; *cross validation: error*}

EOF decomposition with N=2 EOFs

Calculate missing values

Improved guess for missing values



Then we repeat with N= 3 EOFs

and so on...

DINEOF (Data Interpolating Empirical Orthogonal Functions)

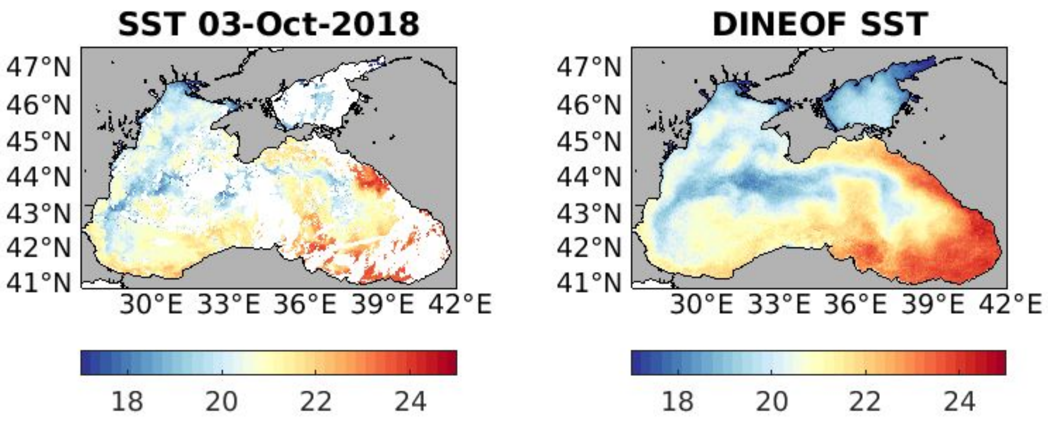
- Technique to **fill in missing data** in geophysical data sets, based on a EOF decomposition

- Missing data? They get initialised to the mean value (and anomalies calculated)

  First guess has low accuracy
  Incremental & iterative calculation of EOF modes

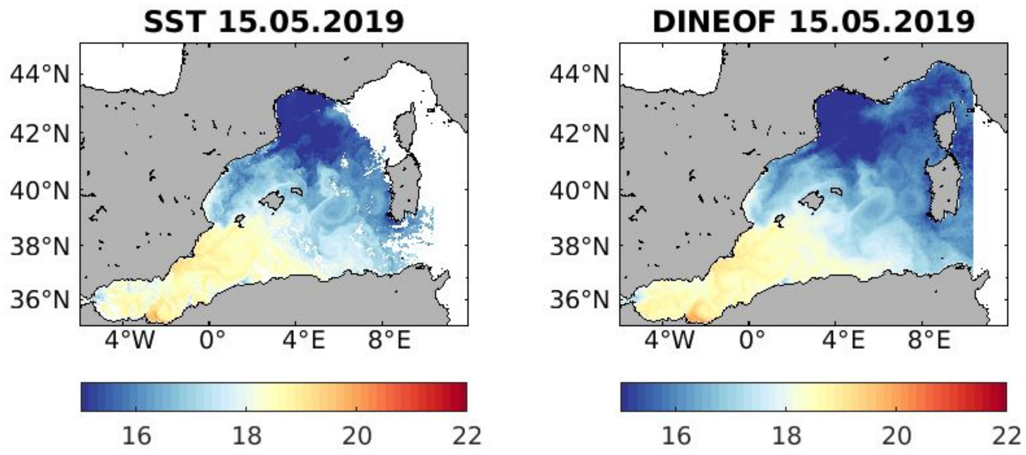
- **Truncated EOF basis** to calculate missing data
  - EOFs extract main patterns of variability
  - Reduced noise
  - Downside: reduced variability as well

- Optimal number of EOFs?
  - Reconstruction error by cross-validation:
  2-3% of valid data set aside
  Comparison at each converged EOF



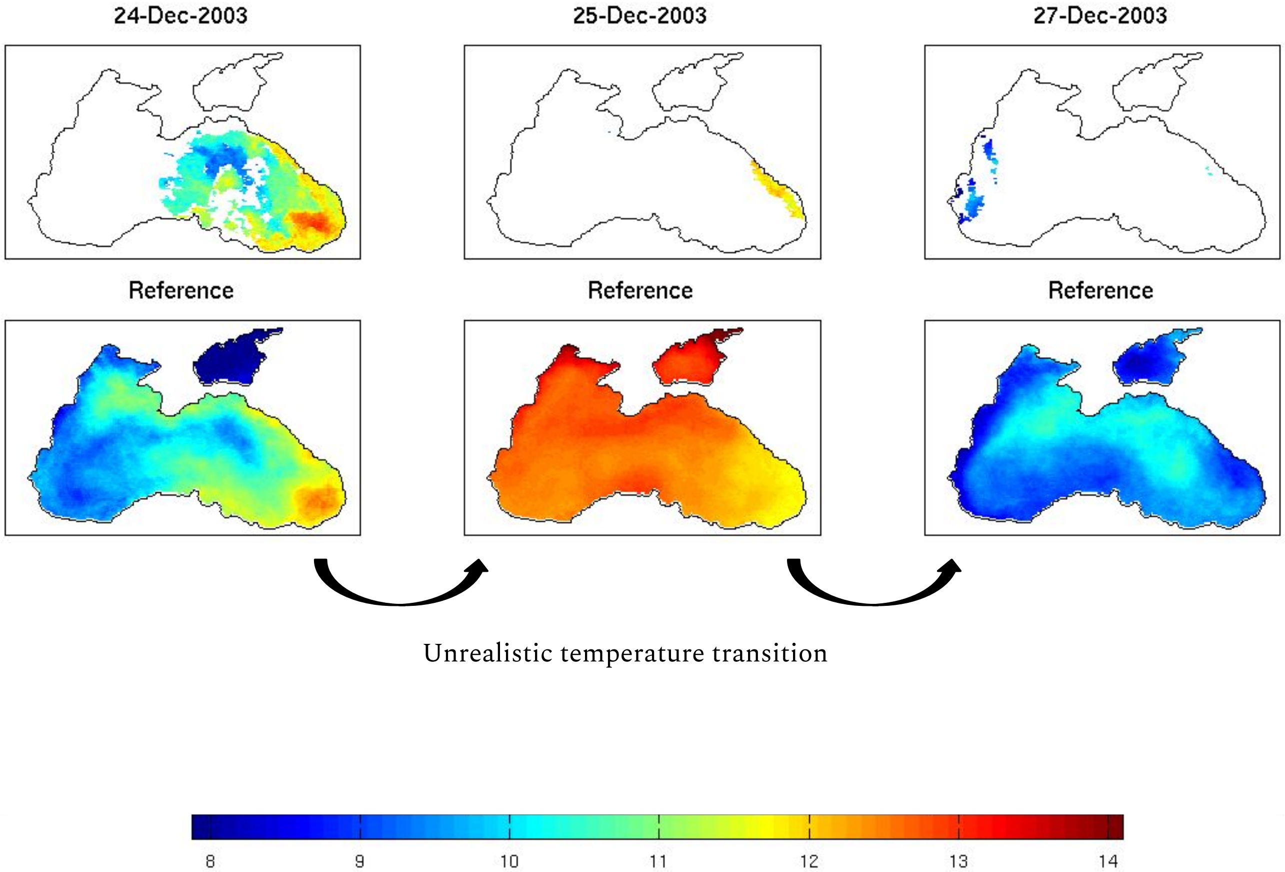
- Uses EOF basis to infer missing data:
    - **non-parametric, data-based**
    - No need of a priori information (correlation length, covariance function...)

- The spatio-temporal coherence present in the data is used to calculate missing values.
    → Three-dimensional data are used. Correlated information in space and time is used to infer missing data values.



Enhancement of temporal coherence in DINEOF reconstructions

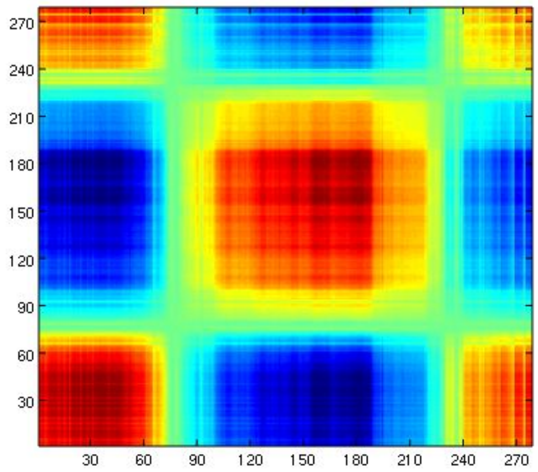
When too few data are present: temporal EOFs poorly constrained: unrealistic discontinuities



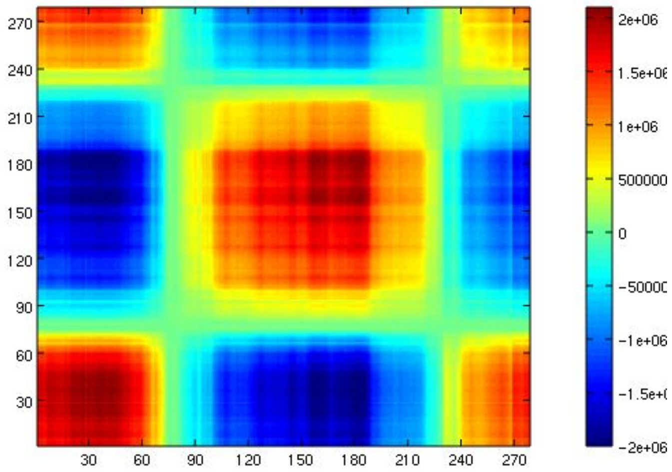
Unrealistic temperature transition

Unrealistic transitions are reflected in the covariance matrix ($\mathbf{C} = \mathbf{X}^T\mathbf{X}$)

→ filter to the temporal covariance matrix to reduce this



→ $\mathbf{C}' = \mathbf{F}^T\mathbf{C}\mathbf{F}$ →



- F is a Laplacian filter
- Filter on **C** instead of **X**: **C** is much smaller and less sensitive to missing data
- Filter applied iteratively: more iterations, further reach of the filter

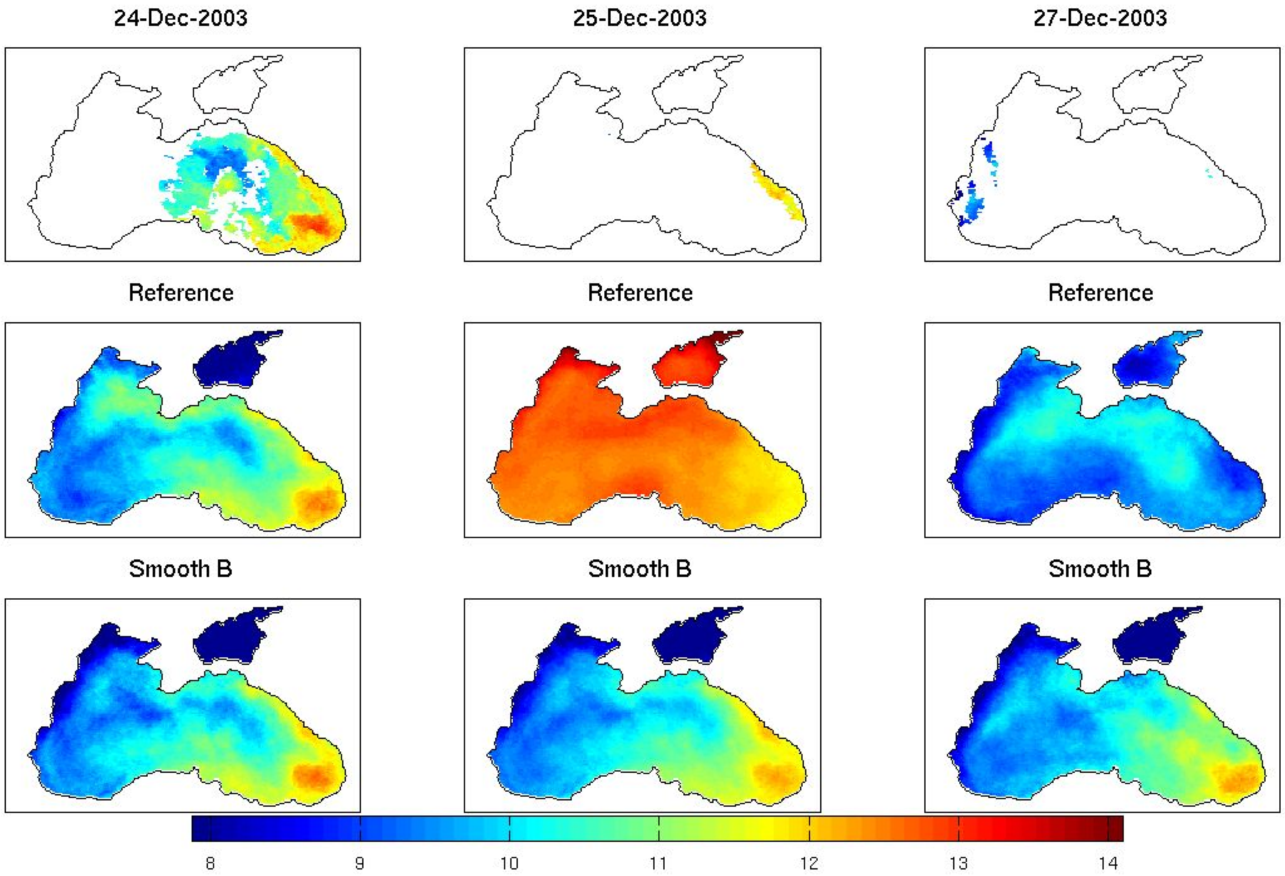
Alvera-Azcárate et al, 2009

Unrealistic transitions are removed efficiently using this filter

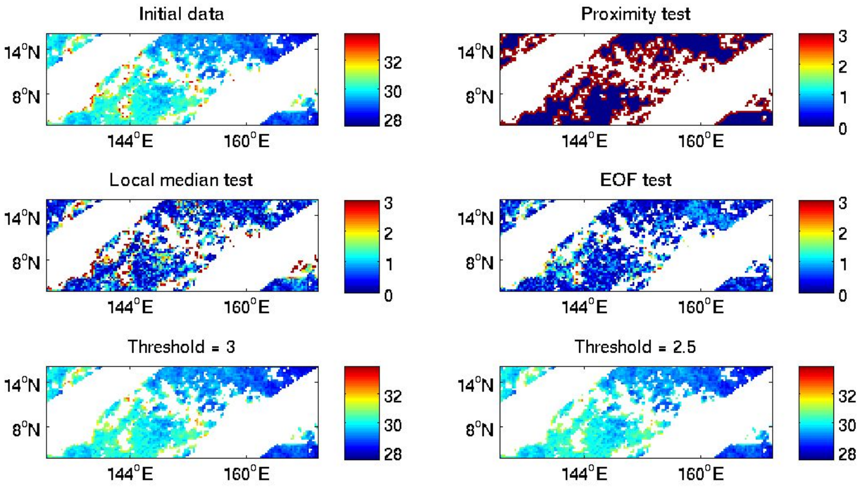
(in this case, the length of the filter was 1.1 days)

no filter

filter



Other developments



Outlier detection

Based on EOF basis + median test + proximity tests

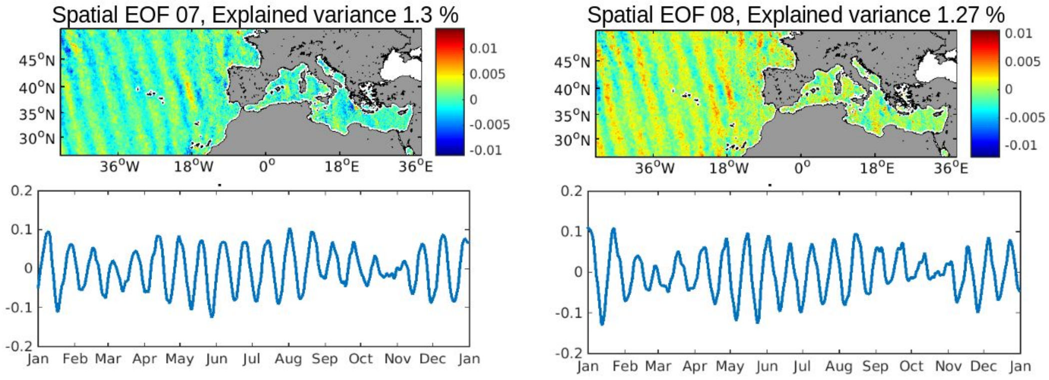
Allows for threshold decision on outliers

Removal of non-physical signals

If consistent biases present, EOFs can detect those (e.g. seasonal biases)

Removal of those EOFs improves quality of data

SMOS L2 data, biases at swath edges picked by repeat cycle



Shadow detection

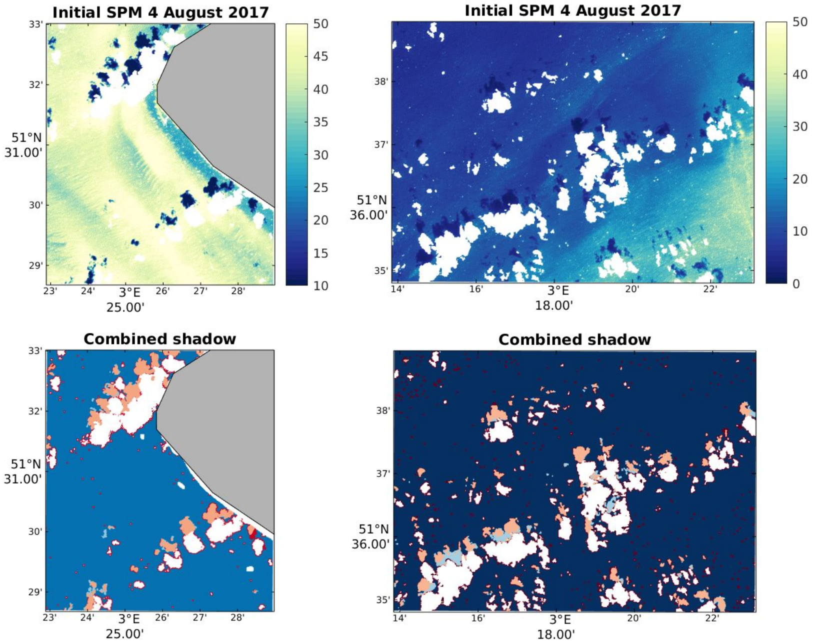
High resolution satellite data (e.g. Sentinel-2 with 10 m resolution) resolve cloud shadows

Difficult to remove because pixels have a “correct” spectral information

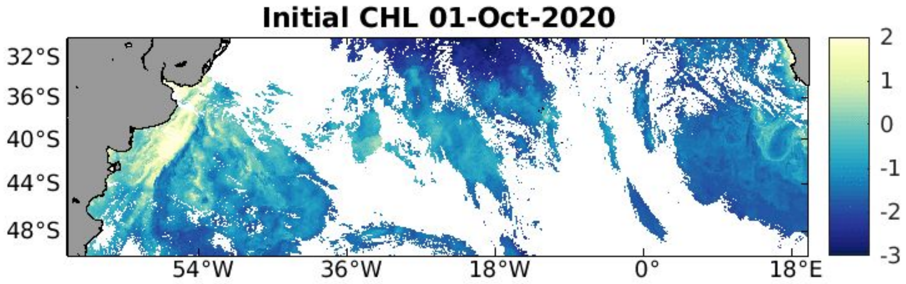
EOF basis can be used to detect and remove cloud shadows

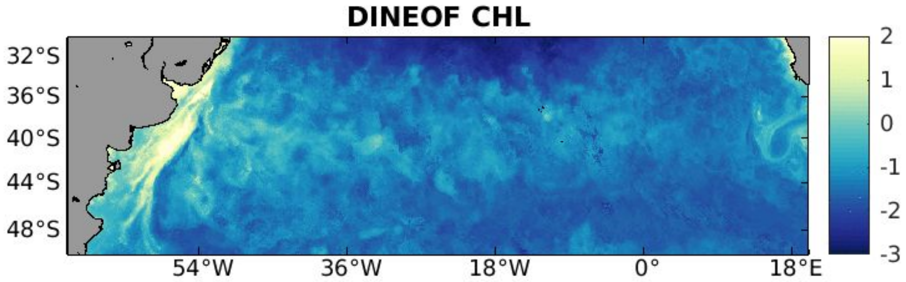
Additional tests:

- Low values penalised
- Departure from median
- Ray tracing



Alvera-Azcárate et al, 2021





In short...

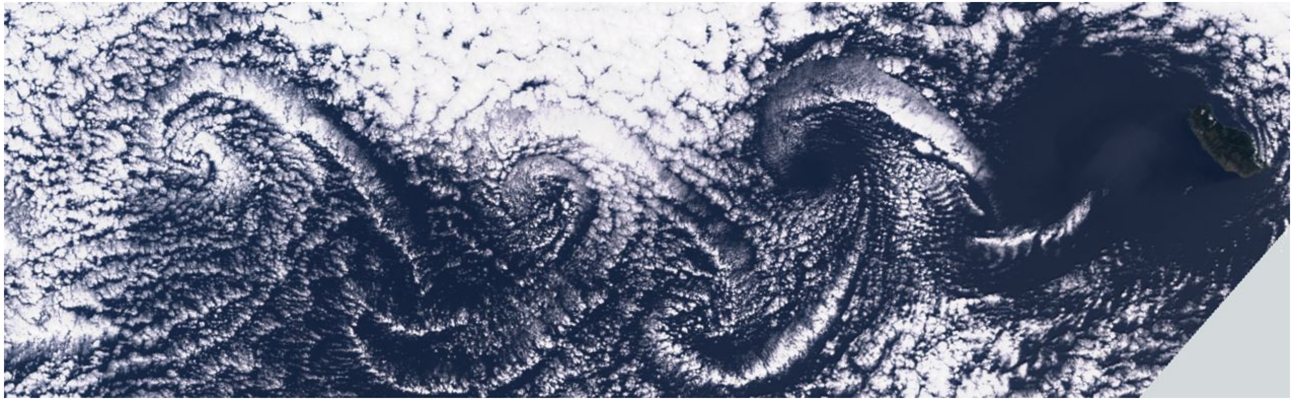
DINEOF is a reliable method for filling missing data.
It's been used, developed & improved for many years.
Several applications for data quality improvement have been developed from DINEOF

Data-Interpolating Convolutional Auto-Encoder
(DINCAE)

Objectives

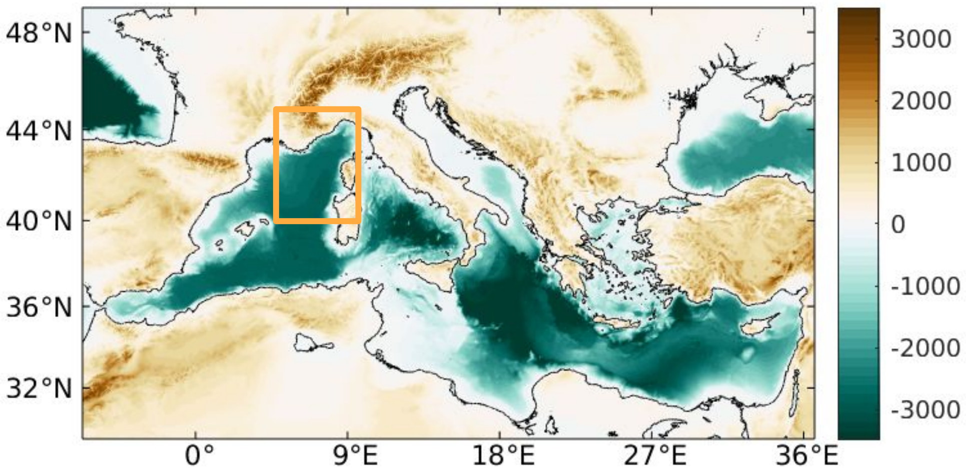
- To derive a methodology to reconstruct missing information in satellite data
    - Based on **neural networks**
    - Making use of ~four decades of sea surface temperature measurements
    - Able to **retain small scale variability**

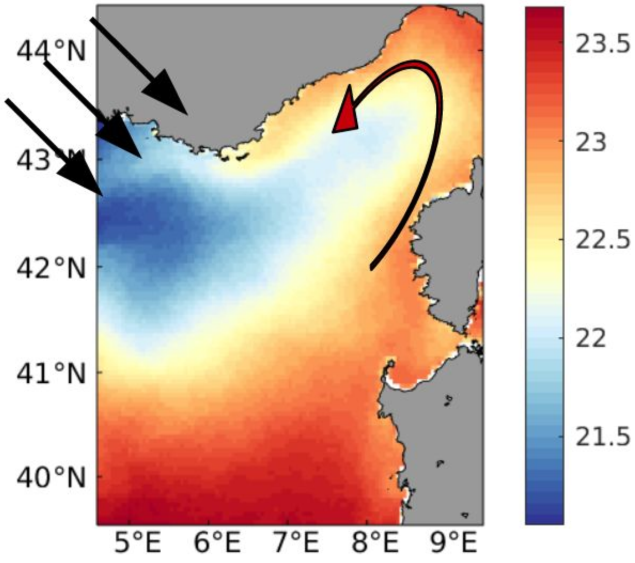
- To assess the benefit of using neural networks in comparison with other state-of-the-art methodologies
    - DINEOF (Data Interpolating Empirical Orthogonal Functions)



Data used

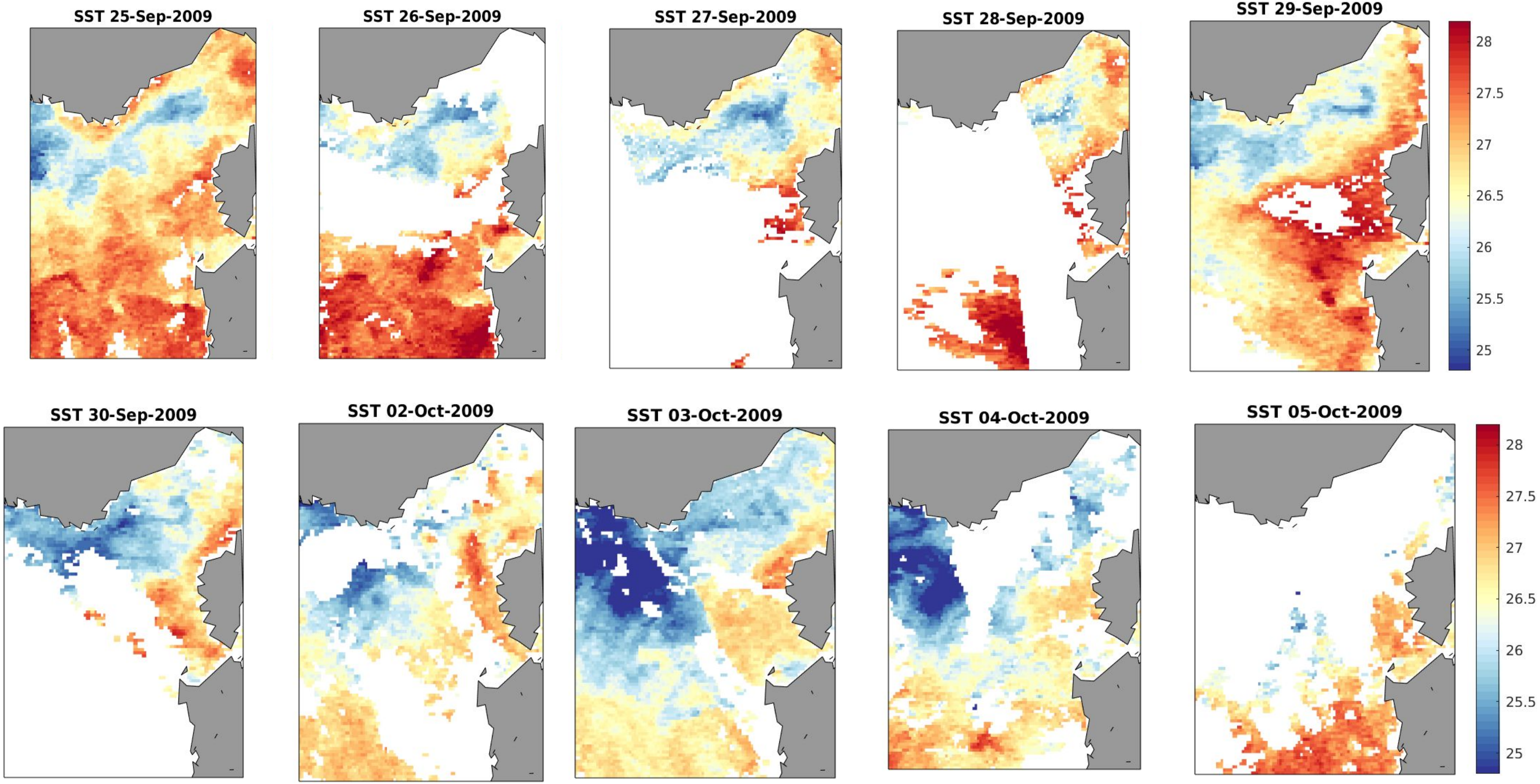
- Daily Advanced Very High Resolution Radiometer (AVHRR) Sea Surface Temperature (SST) data
- **4 km spatial resolution**
- **Liguro-Provençal basin** (western Mediterranean Sea)
- 1 April 1985 to 31 December 2009 (**25 years**) -> **longest homogenous time serie**
- **47 % of missing data**





Average SST (°C)

Challenge: **training on gappy data (lots of gaps!)**



The Bayes' rule or how to handle information of different accuracy

For **Gaussian-distributed errors**:

- prior: $\mathcal{N}(x^f, \sigma^f)$
- observations: $\mathcal{N}(y^o, \sigma^o)$
- posterior: $\mathcal{N}(x^a, \sigma^a)$

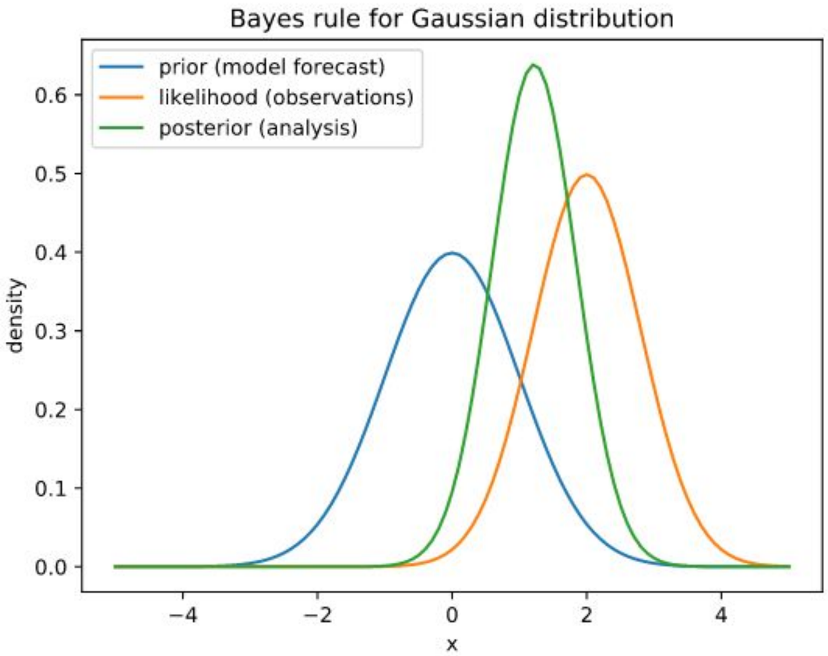
Bayes' rule:

$$p(x|y^o) = \frac{p(x)p(y^o|x)}{p(y^o)}$$

- Mean and variance of posterior given by:

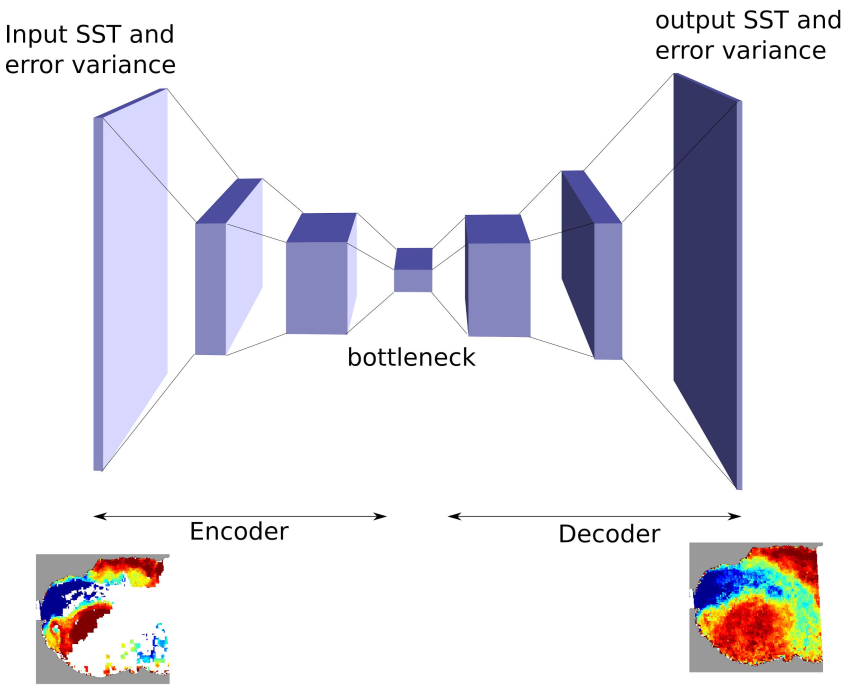
$$\sigma^{a^{-2}} x^a = \sigma^{f^{-2}} x^f + \sigma^{o^{-2}} y^o$$
$$\sigma^{a^{-2}} = \sigma^{f^{-2}} + \sigma^{o^{-2}}$$

- **Inverse of the variance are simply added linearly**



Methodology

DINCAE: Data-Interpolating Convolutional Auto-Encoder



Auto-Encoder: used to efficiently compress/decompress data, by extracting main patterns of variability

- Similarity to EOFs (= auto-encoder with 1 encoding/decoding layer and no activation function)

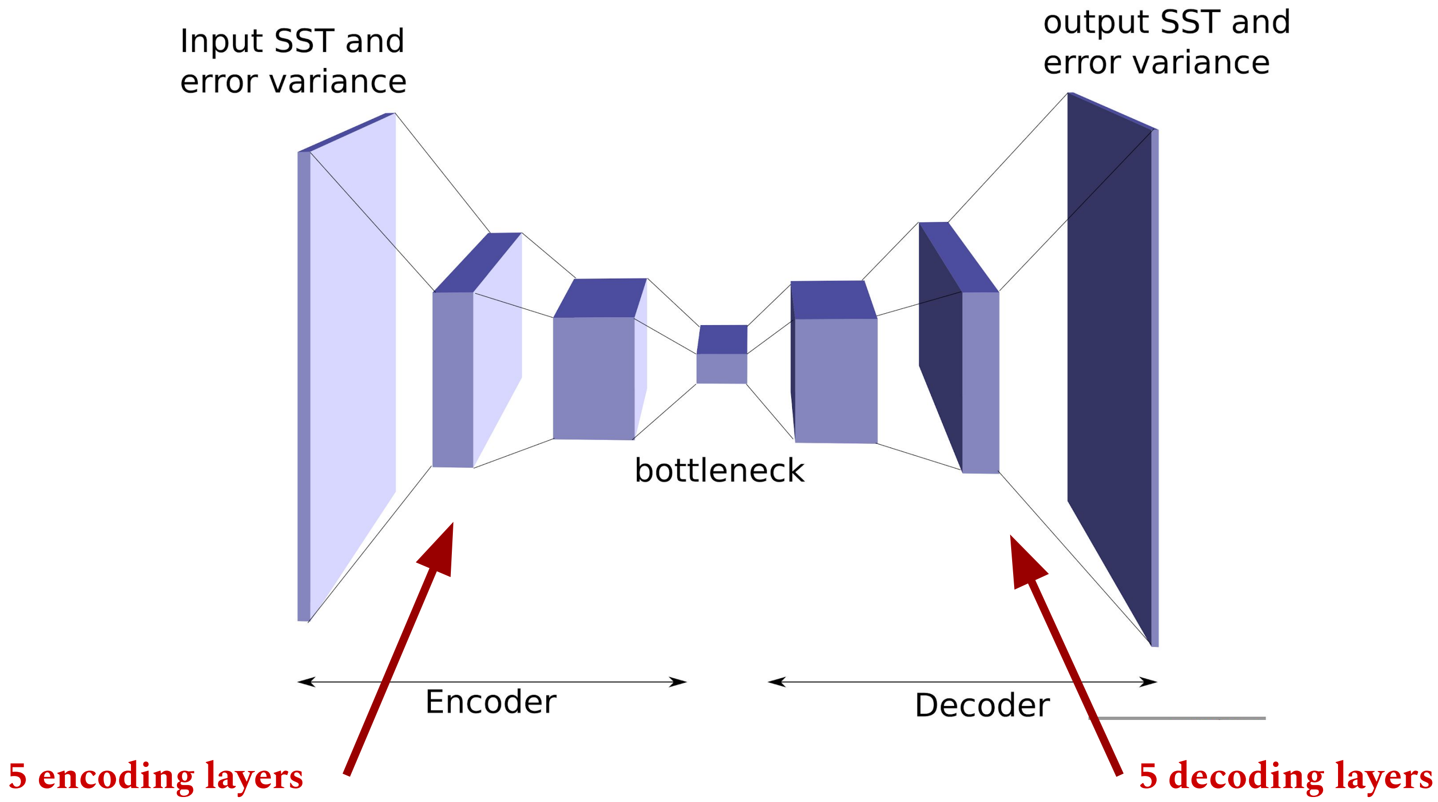
Convolutional: works on subsets of data, i.e. trains on local features

Missing data handled as data with different initial errors

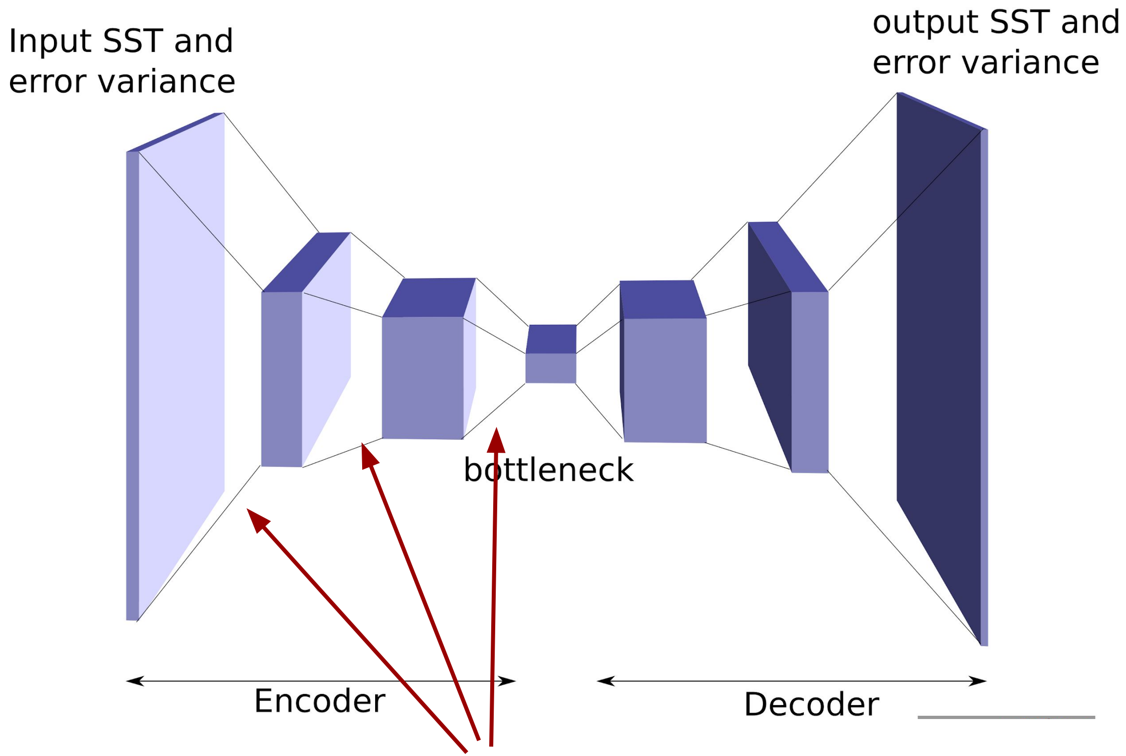
- If **missing, error variance** ($\boldsymbol{\sigma^2}$) **tends to infinity**

Input data:

- SST/σ^2 (previous day, current day, following day)
- 1/σ^2 (previous day, current day, following day)
- Longitude
- Latitude
- Time (cosine and sine of the year-day/365.25)

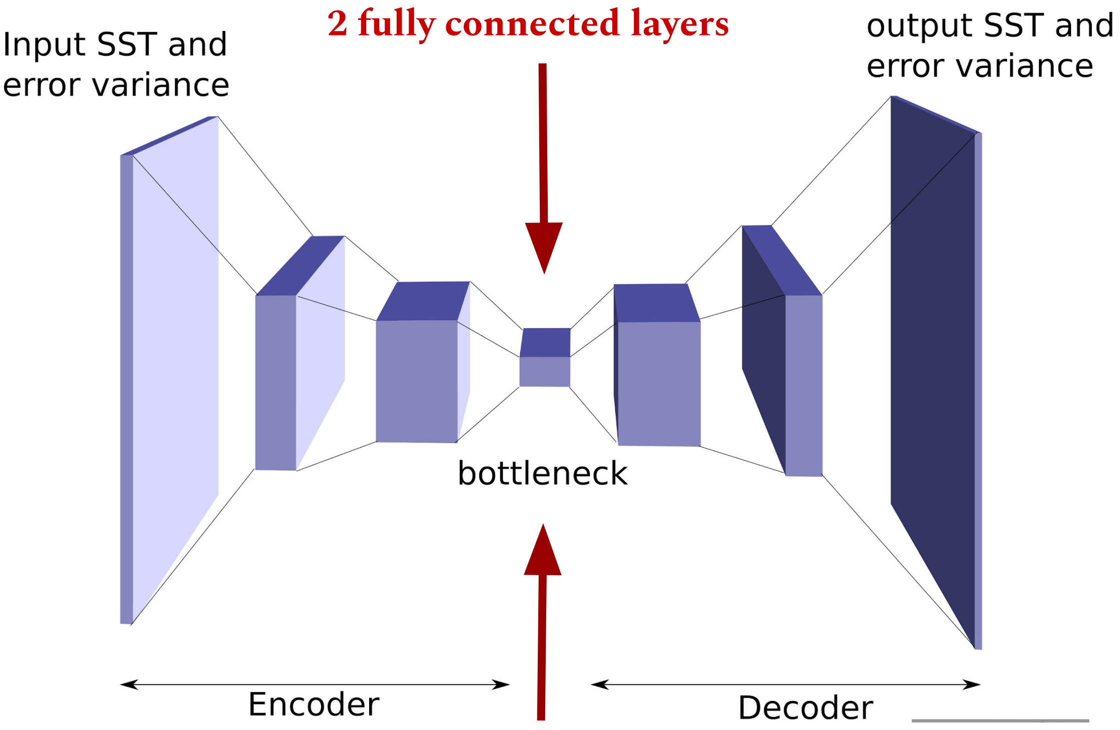


3x3 filters applied at each layer



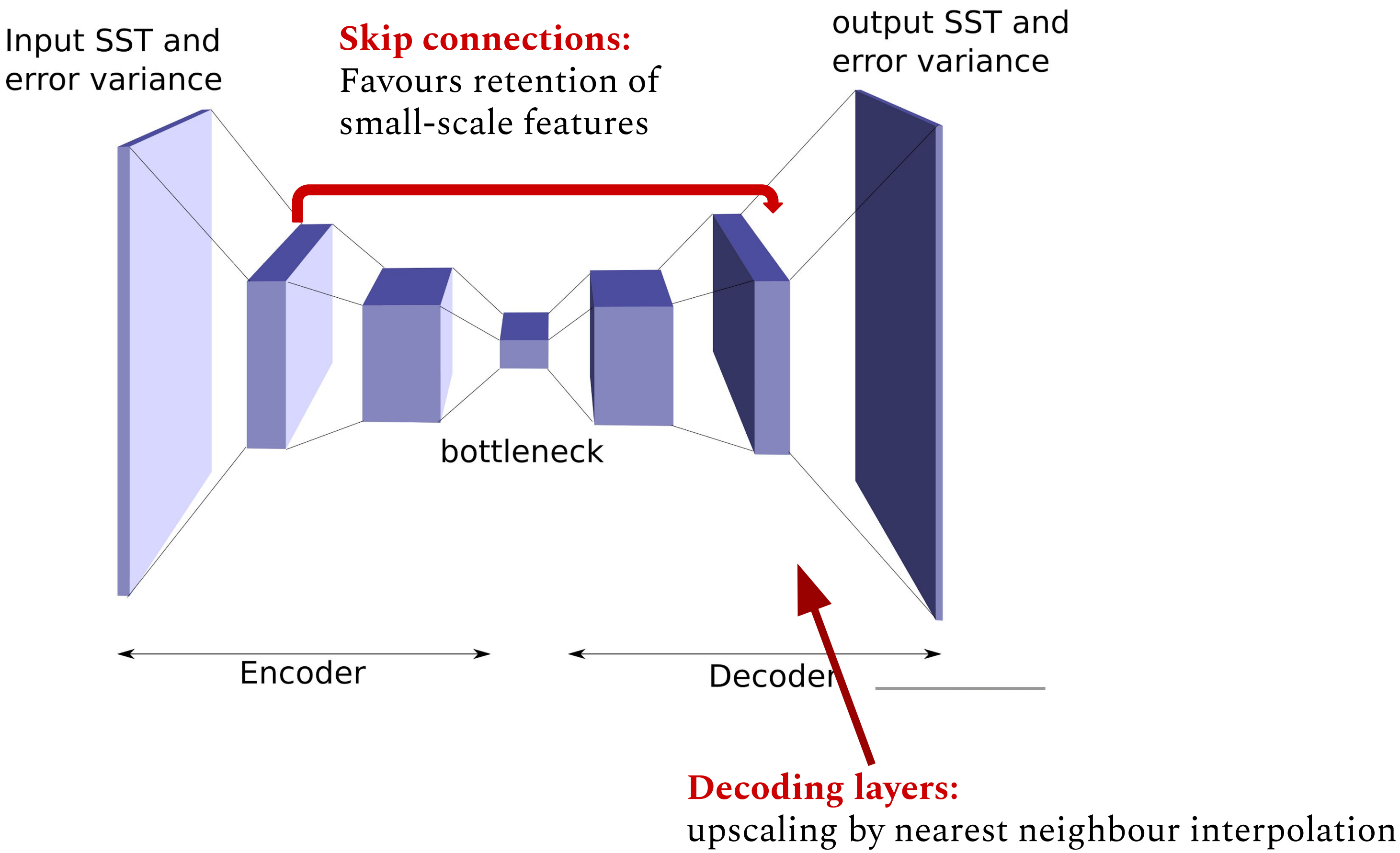
Average pooling layers

Reduce size by retaining the average
value on 2x2 boxes



+ 2 drop-out layers

Take out 30% of neurons (pixels) to avoid overfitting


Input SST and error variance
Skip connections:
Favours retention of small-scale features
output SST and error variance
bottleneck
Encoder
Decoder
Decoding layers:
upscaling by nearest neighbour interpolation

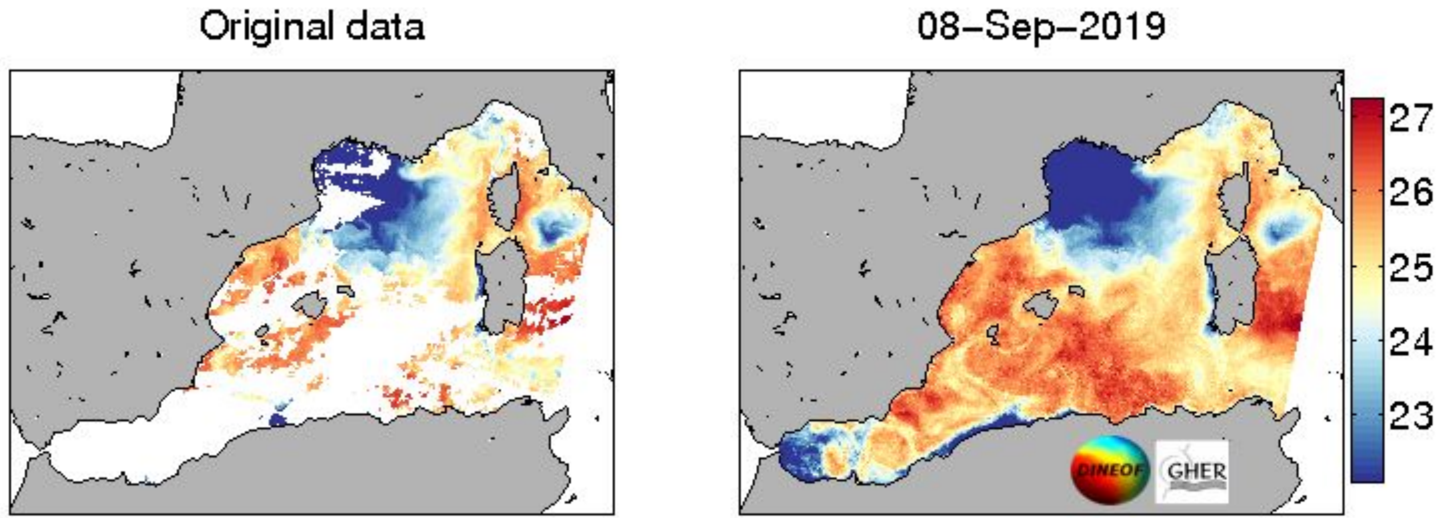
Baseline method to be improved

DINEOF (Data Interpolating Empirical Orthogonal Functions)

A reconstruction method based on the EOF basis from the dataset

~15 years of development & improvements

http://www.dineof.net/DINEOF/



Training

- Partitioned into so-called **mini-batches** of 50 images
- The entire dataset is used **multiple times (epochs)**
- For every input image, **more data points were masked** (in addition to the cross-validation) by using a **randomly chosen cloud mask during training** (data set augmentation).
- The output of the neural network (for every single grid point i,j) is a **Gaussian probability distribution** function characterized by a mean $\hat{y}_{ij}$ and a standard deviation $\hat{\sigma}_{ij}$.

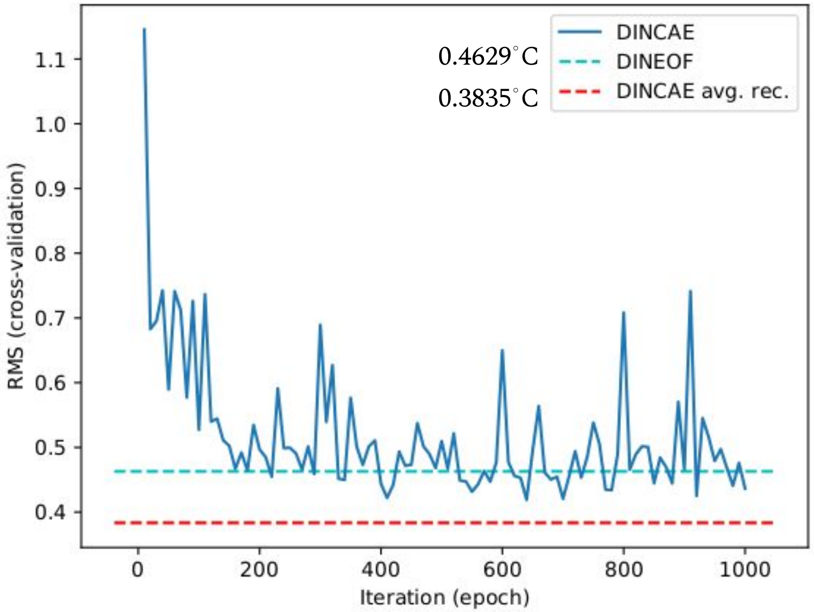
$$J(\hat{y}_{ij}, \hat{\sigma}_{ij}) = \frac{1}{2N} \sum_{ij} \left[\left(\frac{y_{ij} - \hat{y}_{ij}}{\hat{\sigma}_{ij}} \right)^2 + \log(\hat{\sigma}_{ij}^2) + 2\log(\sqrt{2\pi}) \right]$$

- The first term: mean square error, but scaled by the estimated error standard deviation.
- The second term: penalizes any over-estimation of the error standard deviation.

Results

Cross-validation: data removed from the last **50 images of the times series** (with cloud mask from first 50 images)

Averaging epochs 200 to 100 improved DINCAE results



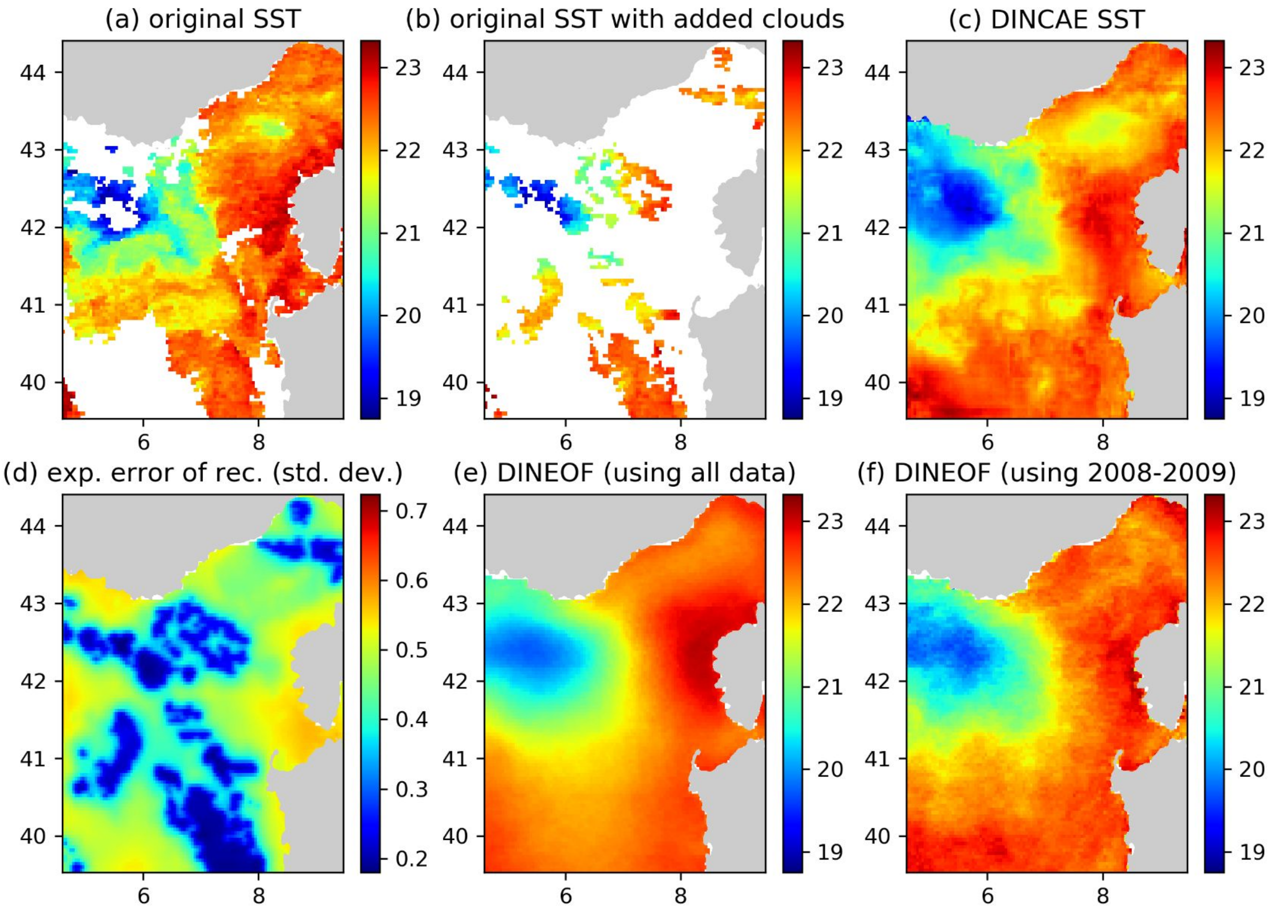
Reconstruction results -full time series- compared to WOD in situ data (under clouds)

RMS (DINEOF) 1.1676°C

RMS (DINCAE) 1.1362°C

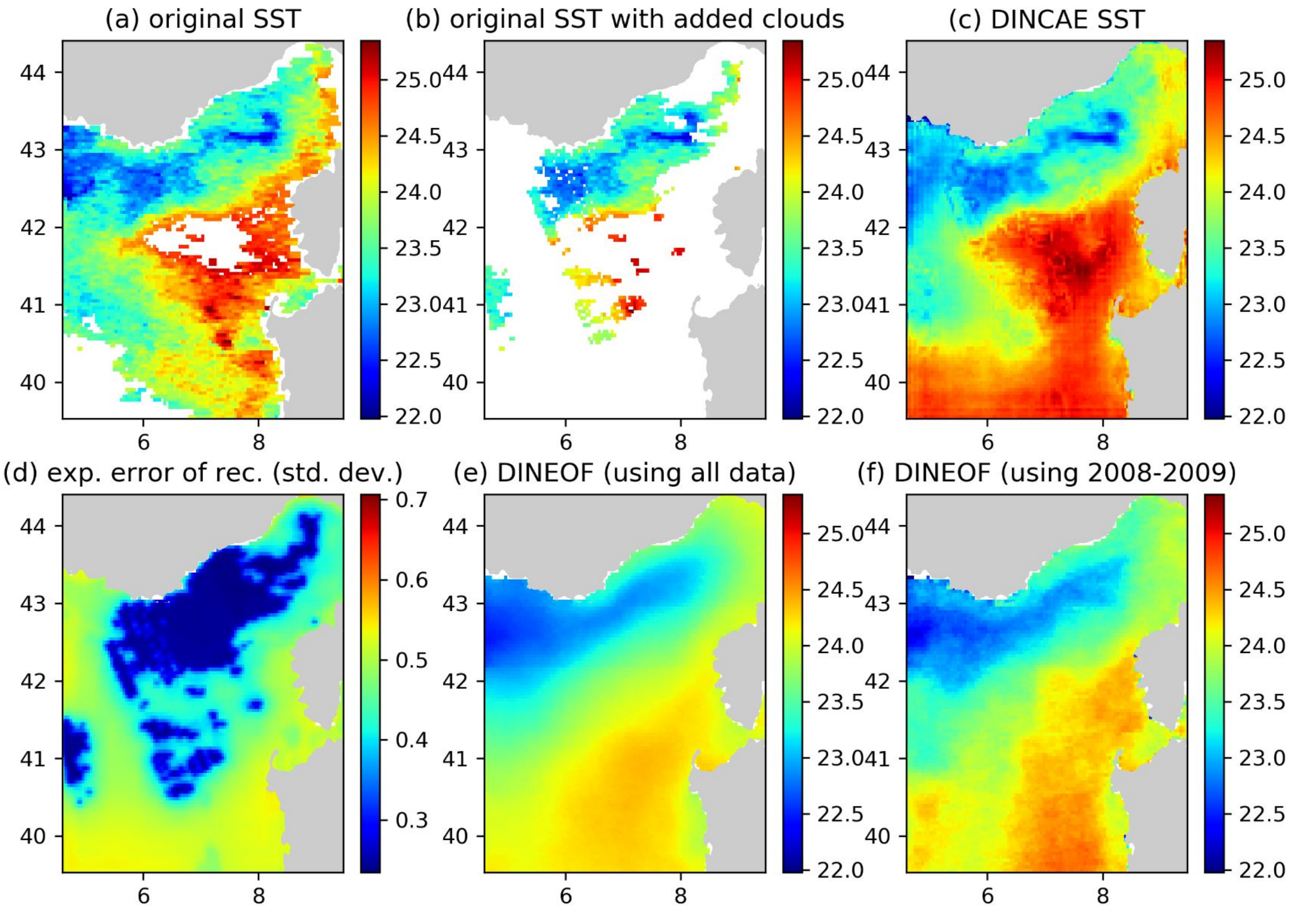
Results

Reconstruction examples



Results

Reconstruction examples



If you want to know more...

- Manuscript in GMD: https://doi.org/ghf3cd

Model description paper

27 Mar 2020

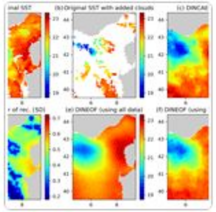
DINCAE 1.0: a convolutional neural network with error estimates to reconstruct sea surface temperature satellite observations

Alexander Barth[1], Aida Alvera-Azcárate[1], Matjaz Licer[2], and Jean-Marie Beckers[1]

[1]GeoHydrodynamics and Environment Research (GHER), University of Liège, Liège, Belgium

[2]National Institute of Biology, Marine Biology Station, Piran, Slovenia

Correspondence: Alexander Barth (a.barth@uliege.be)



Download

- Article (3738 KB)

- Python Code available at:

https://github.com/gher-ulg/DINCAE (currently rewritten in Julia)

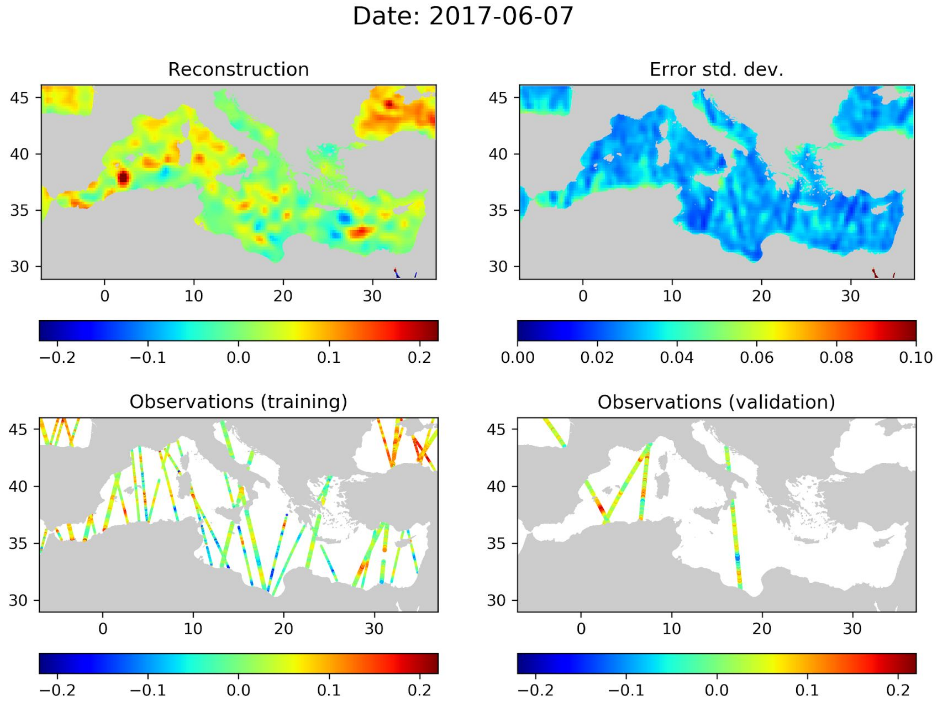
Unstructured data

Altimetry data from 1993-01-01 to 2019-05-13 from CMEMS

Multiple satellites missions

- 70% training data (determine weight of the networks)
- 20% developpement data (determine structure of the network,...)
- 10% test data (independent validation)

Structure of the network determined by Bayesian optimization



Validation

Reasonable **good match** with the validation data

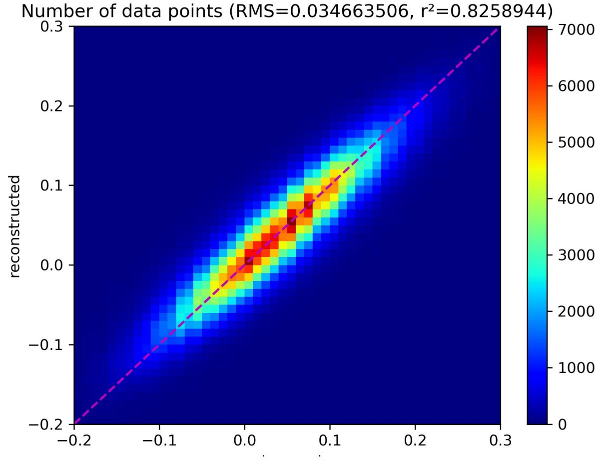
Reliable expected reconstruction errors are notoriously hard to obtain from methods like optimal interpolation

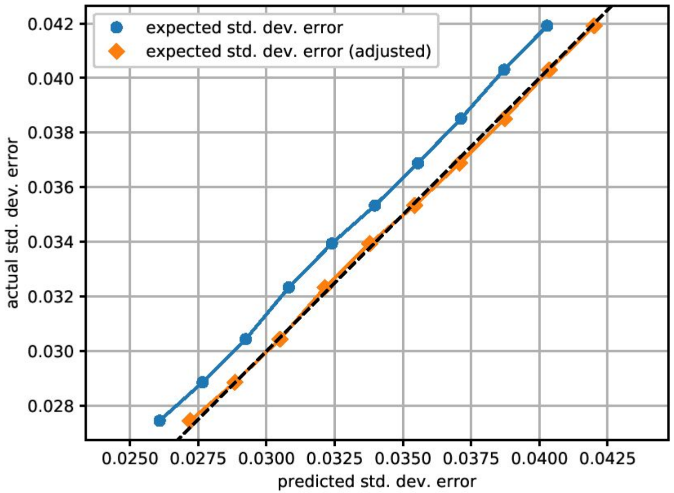
DINCAE also provide as expected error of the reconstruction (per pixel)

The validation data has been **grouped into bins** using the expected error

For every bin the **standard deviation of the actual error** has been computed

The predicted error underestimates the actual error only by 4%



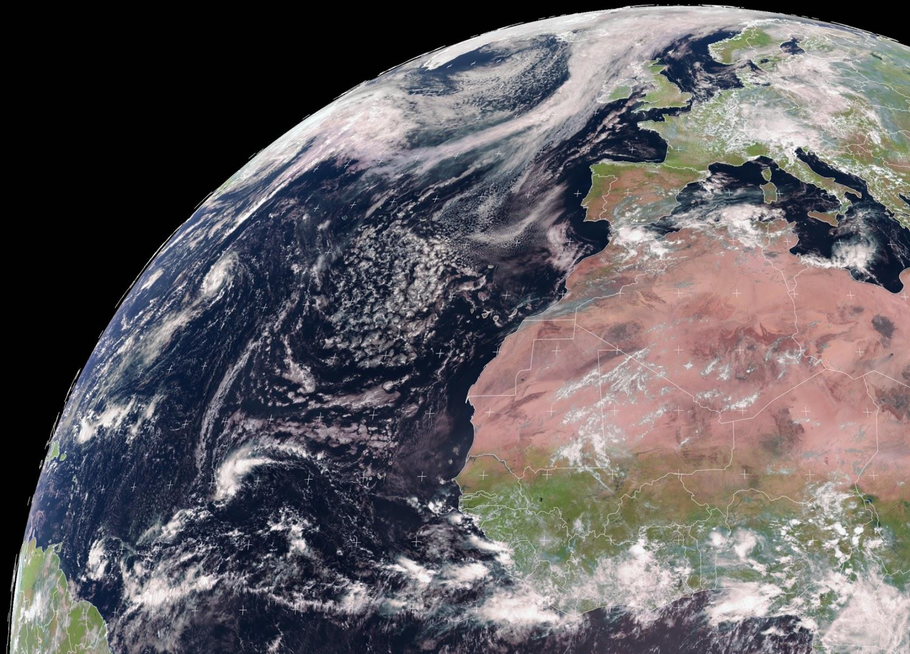


Conclusions

DINEOF:
- A reliable method for filling missing data.
- It’s been used, developed & improved for many years.
- Several applications for data quality improvement have been developed from DINEOF
  Outlier detection, temporal filter, shadow detection…

DINCAE:
- A convolutional Autoencoder approach to reconstruct missing data
- Missing data handled by including expected error variance in the input data
- Estimation of missing data + estimation of error of the reconstruction obtained

Both methods aim to compress the data into a low dimensional subspace and they reconstruct a full field from this compressed representation.



number	type	output size	parameters
1	input	112 x 112 x 10	
2	conv. 2d	112 x 112 x 16	n. filters = 16, kernel size = (3,3)
3	pooling 2d	56 x 56 x 16	pool size = (2,2), strides = (2,2)
4	conv. 2d	56 x 56 x 24	n. filters = 24, kernel size = (3,3)
5	pooling 2d	28 x 28 x 24	pool size = (2,2), strides = (2,2)
7	conv. 2d	28 x 28 x 36	n. filters = 36, kernel size = (3,3)
8	pooling 2d	14 x 14 x 36	pool size = (2,2), strides = (2,2)
9	conv. 2d	14 x 14 x 54	n. filters = 54, kernel size = (3,3)
10	pooling 2d	7 x 7 x 54	pool size = (2,2), strides = (2,2)
11	fully connected layer	529	
12	drop-out layer	529	drop-out rate for training = 0.3
13	fully connected layer	2646	
14	drop-out layer	2646	drop-out rate for training = 0.3
15	nearest neighbor interpolation	14 x 14 x 54	
16	concatenate output of 15 and 8	14 x 14 x 90	
17	conv. 2d	14 x 14 x 36	n. filters = 36, kernel size = (3,3)
18	nearest neighbor interpolation	28 x 28 x 36	
19	concatenate output of 18 and 5	28 x 28 x 60	
20	conv. 2d	28 x 28 x 24	n. filters = 24, kernel size = (3,3)
21	nearest neighbor interpolation	56 x 56 x 24	
22	concatenate output of 21 and 3	56 x 56 x 40	
23	conv. 2d	56 x 56 x 16	n. filters = 16, kernel size = (3,3)
24	nearest neighbor interpolation	112 x 112 x 16	
25	concatenate output of 24 and 1	112 x 112 x 26	
26	conv. 2d	112 x 112 x 2	n. filters = 2, kernel size = (3,3)