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]GHER, University of Liège, Liège, Belgium
[2]National Institute of Biology, Marine Biology Station, Piran, Slovenia

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

Abstract.

A method to reconstruct missing data in satellite data using a neural network is presented. Satellite observations working in the optical and infrared bands are affected by clouds, which obscure part of the ocean underneath. In this paper, a neural network with the structure of a convolutional auto-encoder is developed to reconstruct the missing data based on the available

5   cloud-free pixels in satellite images. However, it is unclear how to handle missing data (or data with variable accuracy) in a neural network when using incomplete satellite data in the training phase. The present work shows a consistent approach which uses essentially the satellite data and its expected error variance as input and provides the reconstructed field along with its expected error variance as output. The neural network is trained by maximizing likelihood of the observed value. The approach, called DINCAE (Data-Interpolating Convolutional Auto-Encoder) is applied to a relatively long time-series of Advanced Very

10 High Resolution Radiometer (AVHRR) sea surface temperature data and compared to DINEOF (Data Interpolating Empirical Orthogonal Functions), a method to reconstruct missing data based on an EOF decomposition. The reconstruction error of both approaches is computed using cross-validation and in situ observations from the World Ocean Database. DINCAE results have lower error, while showing higher variability than the DINEOF reconstruction.

1   Introduction

15 The ocean temperature is an essential variable to study the dynamics of the ocean because density is a function of temperature and therefore part of the ocean current signal depends on ocean temperature. The amount of heat stored in the ocean is also critical for weather predictions at various scales (like hurricane path prediction in the short range, as well as for seasonal and climate predictions).

20 The ocean sea surface temperature (SST) is routinely measured since the beginning of the 1980s. However, as for any sensor working in the infrared or visible bands, clouds often obscure large parts of the field-of-view. Several techniques have been proposed for reconstructing gappy satellite data, but often small scale information is filtered out because of the transient and stochastic nature of these structures. DINEOF (Data Interpolating Empirical Orthogonal Functions), provides an accurate way of retrieving missing data and reducing noise in satellite datasets using a set of optimal EOFs. The optimal number of EOF

is determined by cross-validation. More information on the DINEOF approach is documented in (Beckers and Rixen, 2003; Alvera-Azcárate et al., 2005). DINEOF has been applied to several oceanographic variables, at different spatial resolutions (e.g. Alvera-Azcárate et al. (2005) for SST, Alvera-Azcárate et al. (2007) for ocean colour, Alvera-Azcárate et al. (2016) for Sea Surface Salinity), providing accurate reconstructions. A truncated EOF basis is used to remove noise, although some small

5   scale and transient features can also be removed from the analysis, resulting in a smooth reconstruction.

Neural networks are mathematical models that can efficiently extract nonlinear relationships from a mapping problem (i.e. an input/output relationship that can be determined through a mathematical function). Neural networks are therefore specially well positioned to detect the presence of nonlinear, stochastic features measured at the sea surface by satellite sensors, and

10   their use might prove efficient in retaining these structures when analysing satellite data, for example for reconstructing missing data. However, neural networks are typically applied on a large and complete data set (i.e. no or almost no gaps) as input data, and therefore a solution needs to be found to handle a large number of missing data.

The use of neural networks in the frame of Earth Observation has been increasing recently. Garcia-Gorriz and Garcia-

15   Sanchez (2007), for example, used meteorological variables like wind and air temperature (among others) to infer SST, with the aim of reproducing annual and interannual variability of SST during the pre-satellite era. Patil and Deo (2017) used a wavelet neural network to predict SST at various locations in the Indian Ocean, which allowed to focus on daily variations of SST. Pisoni et al. (2008) resorted to past instances and averaging to overcome gaps in SST, which results in smooth reconstructions. Krasnopolsky et al. (2016) used neural networks to infer ocean colour in the complete absence of these data

20   (i.e. emulating a sensor failure). Jo et al (2018) infers ocean colour from related data (SST and wind among others), taking advantage of the close relation between different ocean variables, but also at a lower spatial resolution. Renosh et al (2018) produced a suspended particulate matter dataset from model and in situ data using Self Organizing Maps, that was compared to satellite data. Chapman and Charantonis (2017) used surface satellite data to infer subsurface ocean currents also using Self Organizing Maps. Also using Self Organizing Maps, Jouini et al (2013) reconstructed missing data in chlorophyll maps using

25   the relation between this variable and ocean currents (proxied by SST and sea surface height).

The objective of this manuscript is to present a neural network in the form of a convolutional auto-encoder which can be trained on gappy satellite observations, in order to reconstruct missing observations and also to provide an error estimate of the reconstruction. This neural network is referred in the following as DINCAE (data-interpolating convolutional auto-encoder).

30   An auto-encoder is a particular type of network which can compress and decompress the information in an input dataset (Hinton and Salakhutdinov, 2006),effectively reducing the dimensionality in the input data.

In section 2, the SST dataset used in this study is presented. This dataset is first reconstructed with DINEOF (section 3). The structure of the network and its cost-function able to provide an error estimate of the reconstruction are presented in section 4.

The results are validated by cross-validation and by a comparison to the World Ocean Database 2018 in section 5. Finally, the conclusions are presented in section 6.

2   Data availability

Having long time series to train neural networks is quite important to achieve good results. For this study we used the longest

5   available time series coming from the Advanced Very High Resolution Radiometer (AVHRR) dataset (Kilpatrick et al., 2001) spanning 25 years, from 1 April 1985 to 31 December 2009. The data are distributed by the Physical Oceanography Distributed Active Archive Center (PODAAC), and have a spatial resolution of 4 km and a temporal resolution of 1 day. The dataset can directly be accessed by following the DOI link in the references (AVHRR Data). In this study, we focus on part of the Provençal basin (4.5625^{o}E, 9.5^{o}E and 39.5^{o}N, 44.4375^{o}N, Figure 1) where the main circulation features are the Western Corsican Cur-

10   rent (WCC) and the Northern Corsican Current (NC) describing a cyclonic circulation pattern. In addition, several mesoscale and submesoscale circulation features are present in this area. With a resolution of 4 km, the SST data measures only mesoscale and basin-wide variability.

For this study, only SST data with quality flags of 4 or higher are retained (Evans et al., 2009). One single image is com-

15   posed by 112 x 112 grid points. If a given pixel has measurements less than 5% of the time, then it is not reconstructed and it is considered as a land point in the following. In total, 27% of grid points correspond to land. Images with at least 20% of valid sea points are retained for the reconstruction which corresponds to a total 5266 time instances.

To assess the accuracy of the reconstruction method, cross-validation is used as follows: in the last 50 images we removed

20   data according to the cloud mask of the first 50 images of the SST time series. These data are not used at all during either the training or the reconstruction phases, and can therefore be considered independent. In total, 106 816 measurements have been withheld this way.
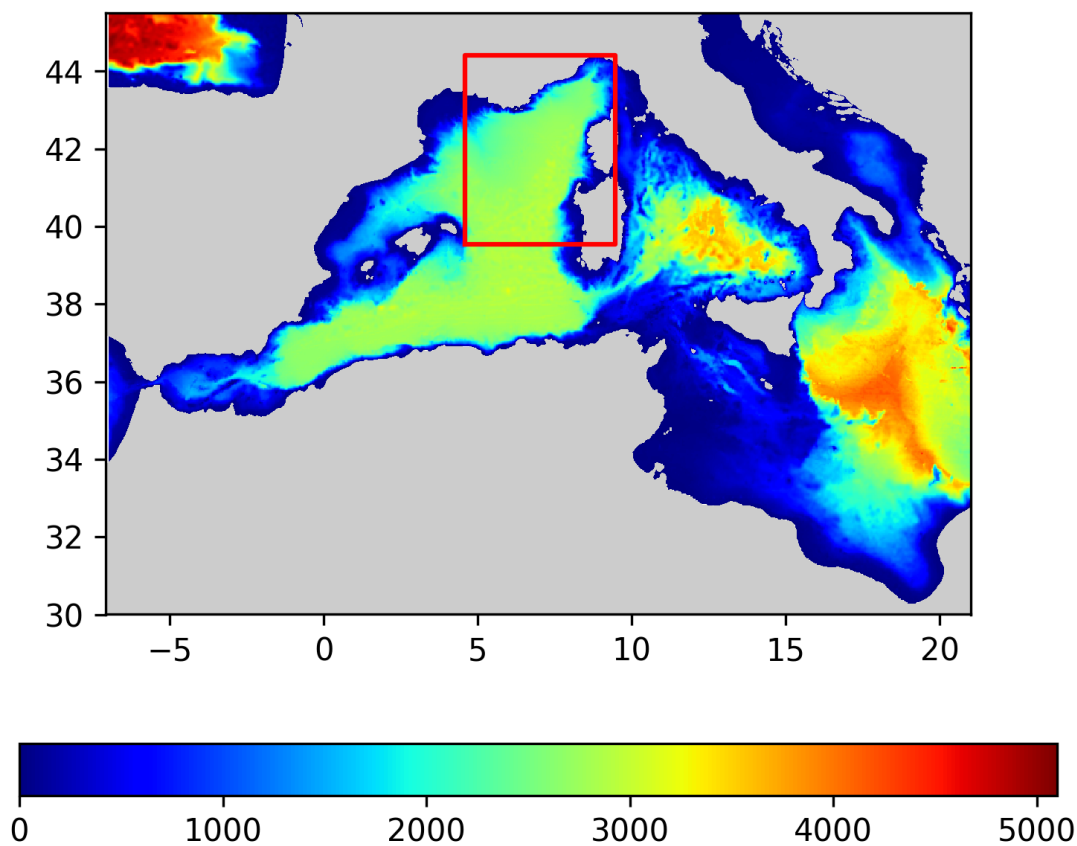
Figure 1. The red rectangle delimits the studied region and the color represents the bathymetry in meters. The arrows represent the main currents: the Western Corsican Current (WCC), the Eastern Corsican Current (ECC) and the Northern Corsican Current (NC)

3   DINEOF reconstruction

DINEOF is applied to the SST time series of 25 years. Before calculating the EOFs, the spatial and temporal mean is removed from the data, and the missing data are set to zero. The missing data are then calculated through an iterative procedure, as explained in Alvera-Azcárate et al. (2005). A temporal low-pass filter with a cut-off period of 1.08 days is applied to improve the

5   temporal coherency of the results, following Alvera-Azcárate et al. (2009). This filter effectively propagates the information in time so that for a given date also the satellite data from the previous and next days are used in the reconstruction. The optimal

Table 1. Comparison with the independent cross-validation data and the dependent data used for training (in ^{o}C)

| | CV data | | | non-CV data | | |
	RMS	CRMS	bias	RMS	CRMS	bias
DINEOF	0.4629	0.4536	-0.0922	0.3864	0.3864	-0.0029
DINEOF (2008-2009)	0.4789	0.4715	-0.0839	0.3376	0.3375	-0.0038
DINCAE (no skip connections)	0.4458	0.4456	0.0147	0.2957	0.2953	0.0153
DINCAE (2 skip connections)	0.4222	0.4217	0.0198	0.1519	0.1504	-0.0210
DINCAE (all skip connections)	0.3900	0.3895	0.0199	0.1383	0.1380	-0.0097
DINCAE (all skip connections - median)	0.3922	0.3918	0.0190	0.1342	0.1342	-0.0012
DINCAE (all skip connections and wider layers)	0.4005	0.4003	0.0147	0.1339	0.1328	0.0175
DINCAE (all skip connections and narrower layers)	0.3928	0.3915	0.0318	0.1379	0.1345	-0.0300
DINCAE (all skip connections and less layers)	0.4603	0.4557	-0.0648	0.1396	0.1364	-0.0295
DINCAE (all skip connections and more layers)	0.3991	0.3990	0.0083	0.1350	0.1346	-0.0101
DINCAE (all skip connections and average pooling)	0.3835	0.3834	0.0102	0.1251	0.1250	-0.0063

number of EOFs retained in this reconstruction is 13 modes, which explain 99.4% of the variability of the initial data.

The classical DINEOF technique reconstructs the cross-validation data points withheld in the last 50 images with an error of 0.4629^{o}C and a slight negative bias of -0.0922 ^{o}C (Table 1). As only 13 modes are retrained by DINEOF for the reconstruction, 5 some small scale structures are smoothed-out, which is a well known property of a truncated EOF decomposition (*e.g.* Alvera-Azcárate et al., 2009). This smoothing effect results in an RMS error of 0.3864^{o}C when comparing the reconstructed dataset to all the initially present SST (*i.e.* used for the reconstruction). A somewhat surprising result is that when using less data (only from the last two years, *i.e.* 2008 to 2009), 19 EOFs modes are retained, leading to a reconstruction with richer structures. Therefore, the RMS error compared to all the initially present SST provided to DINEOF (but excluding the cross-validation 10 data) is lower (0.3375^{o}C) than when the 25 years dataset is used. However, the RMS error compared to the cross-validation data is slightly worse with the 2 years dataset (0.4789^{o}C). As the main validation statistic for this study is the RMS error compared to the cross-validation dataset, we use the DINEOF reconstruction of the full 25 years dataset in the following.

4   Neural network with missing data as input

Convolutional and other deep neural networks are extensively used in computer vision and Earth sciences (Rasp et al., 2018; 15 Bolton and Zanna, 2019; Zhou et al., 2016; Geng et al., 2015) where full datasets are available, at least for training a network. However, when using satellite data the number of images without any clouds is very small and it is difficult to provide enough training data when only clear images are used. Therefore the aim is to derive a reconstruction strategy which can cope with the large amounts of missing data typically found in remote sensing data.

The handling of missing data is done in analogy to the assimilation of data in numerical ocean models. The standard optimal 20 interpolation equations (*e.g.* Bretherton et al., 1976; Buongiorno Nardelli, 2012) can be written as follows:

$$\mathbf{P}^{a\,-1}\mathbf{x}^a \;=\; \mathbf{P}^{f\,-1}\mathbf{x}^f + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}^o \tag{1}$$

$$\mathbf{P}^{a\,-1} \;=\; \mathbf{P}^{f\,-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \tag{2}$$

where $\mathbf{x}^f$ is the model forecast with error covariance $\mathbf{P}^f$, $\mathbf{y}^o$ are the observations with error covariance $\mathbf{R}$ and $\mathbf{H}$ is the observation operator extracting the observed part from the state vector $\mathbf{x}^f$. The analysis $\mathbf{x}^a$ is the combined estimate with the error covariance matrix $\mathbf{P}^a$. We use these equations as an analogy to propose an approach to handle of missing data (or data with different errors) in a convolutional auto-encoder (CAE). The main input datasets of the CAE are i) the SST divided by its error variance and ii) the inverse of the error variance. If a data point is missing, then the corresponding error variance is considered infinitely large and the value at this point would be zero for both input datasets. The main difference is that in optimal interpolation, the observation vector $\mathbf{y}^o$ is multiplied by the inverse of the error covariance (possibly including non-diagonal elements) while in the present case we use only the error variance. The structure of the neural network will be used to spatially propagate the information from the observations.

In a neural network, a neuron with a value of zero does not trigger any activation in the following layers and forcibly putting the value of a random neuron to zero is a common strategy (called drop-out) to avoid overfitting.

The time average has been removed from the SST dataset (computed over all years but excluding the cross-validation dataset). The neural network works thus with anomalies relative to this mean SST. To obtain reasonable results the network uses more input than merely SST divided by its error variance and the inverse of the error variance. The total list of input parameters is consequently the following:

– SST anomalies scaled by the inverse of the error variance (zero if the data is missing)

– Inverse of the error variance (zero if the data is missing)

– Scaled SST anomalies and inverse of error variance of the previous day

– Scaled SST anomalies and inverse of error variance of the next day

– Longitude (scaled linearly between -1 and 1)

– Latitude (scaled linearly between -1 and 1)

– cosinus of the day of the year divided by 365.25

– sinus of the day of the year divided by 365.25

The complete dataset is this represented by an array of the size 8 x 112 x 112 x 5266. The inverse of the error variance is either zero (for missing data) or a constant. The precise value of this constant is not important because it will be multiplied by a weight matrix and this weight matrix will be optimized by training the network. In future studies, it would be interesting to use sensor specific error statistics provided with GHRSST products, *i.e.* spatially and temporally varying error estimate. The
5   final layer of the neural network produces the following output:

– SST scaled by the inverse of the expected error variance

– Logarithm of the inverse of the expected error variance

The main building blocks of the neural network (Table 2) are convolutional layers (LeCun et al., 1998; Krizhevsky et al., 2012). DINCAE uses 5 encoding and 5 decoding layers with a different number of filter sizes. Beside the input and output
10   layer, the filter sizes are 16, 24, 36 and 54 (the filter sizes increase 50% from one encoding convolutional layer to the next). All convolutional layers have a receptive field of 3x3 grid points (Simonyan and Zisserman, 2015). Between the convolutional layers there are max pooling or average pooling layers (Scherer et al., 2010) to progressively reduce the spatial resolution by only retaining either the maximum or average value of a region of 2x2 grid points. After the last encoding convolutional layer, there are two fully connected layers (the so-called bottleneck). The number of neurons in the bottleneck is a fifth of
15   the number of the last pooling layer of the encoder (rounded to the nearest integer). Drop-out is used in the fully connected layers to avoid overfitting. The decoding layers are composed by convolutional layers and interpolation layers (to nearest neighbor) to upsample the results. We also added skip connections between the output of pooling layers and the upsampling layers (Ronneberger et al., 2015). These skip connections correspond to layers 16, 19, 22 and 25 of Table 2. The motivation of this choice is that large-scale information of the SST would be captured by the neurons in the bottle-neck, but small-scale
20   structures unrelated to the overall structure in the SST would be handled by these skip connections. In the absence of the skip connections, the small scale structures would be removed from the dataset.

A rectified linear unit (RELU) activation function is commonly used in neural networks which is defined as:

$$f(x) = \max(x, 0) \tag{3}$$

However, in our case it leads quickly (in 10 epochs) to a zero gradient and thus to no improvements in training. This problem
25   is solved by choosing a leaky RELU (Maas et al., 2013) for the convolutions and the standard RELU for the fully connected layers.

$$f(x) = \max(x, \alpha x) \tag{4}$$

where we use here $\alpha = 0.2$. The output of the network, i.e. the 26th layer of Table 2, is an array $T_{ijk}^{(26)}$ with 112 x 112 x 2 elements. The first slice $k = 1$ is essentially interpreted as the logarithm of the inverse of the expected error variance and

Table 2. List of all steps in DINCAE. The additional dimension of the size of the minibatch is omitted in the output sizes below. Max pooling and average pooling are tested for the pooling layers.

number	type	output size	parameters
1	input	112 x 112 x 8	
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)

the second slice is the temperature anomaly divided by the error variance. The reconstructed temperature anomaly $\hat{y}_{ij}$ and the corresponding error variance $\hat{\sigma}^2_{ij}$ are computed as:

$$\hat{\sigma}^2_{ij} = \frac{1}{\max(\exp(\min(T^{(26)}_{ij1}, \gamma), \delta)} \tag{5}$$

$$\hat{y}_{ij} = T^{(26)}_{ij2} \hat{\sigma}^2_{ij} \tag{6}$$

5 where $\gamma = 10$ and $\delta = 10^{-3}\ °\mathrm{C}^{-2}$. The $\min$ and $\max$ functions in the previous equations are introduced to avoid a division by a value close to zero or a floating point overflow.

4.1 Training of the neural network

The input data set is randomly shuffled (over the time dimension) and partitioned into so-called mini-batches of 50 images, as an array of the size 8 x 112 x 112 x 50. The complete time series is splitted into 105 minibatches with 50 images each and one last minibatch with only 16 images (representing a total of 5266 as mentioned before). The splitting of the dataset into

5 minibatches is necessary because the graphical processing unit (GPU) has only a limited amount of memory. Computing the gradient over randomly chosen subsets introduces also some stochasticity which prevents the minimization algorithm to be trapped in a local minima. An optimization cycle using all 106 minibatches is called an epoch.

For every input image, more data points were masked (in addition to the cross-validation) by using a randomly chosen cloud

10 mask during training. The cloud mask of a training image would thus be the union of the cloud mask of the input dataset and a randomly chosen cloud mask. This allows to assess the capability of the network to recover missing data under clouds. Without the additional clouds, the neural network would simply learn reproduce the SST values that are already received as input.

The aim of DINCAE is to provide a good SST reconstruction but also an assessment of the accuracy of the reconstruction.

15 The output 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 $\hat{\sigma}_{ij}$. Given this pdf one can compute the likelihood $p(y_{ij}|\hat{y}_{ij}, \hat{\sigma}_{ij})$ of the observed values y_{ij}. The weights and biases in the neural network are adjusted to maximize the likelihood of all observations. Maximizing the likelihood is equivalent to minimizing the negative log-likelihood:

$$J(\hat{y}_{ij}, \hat{\sigma}_{ij}) = -\frac{1}{N} \sum_{ij} \log \left(p(y_{ij}|\hat{y}_{ij}, \hat{\sigma}_{ij}) \right) \tag{7}$$

20 where N is the number of measurements in y_{ij} (excluding thus land points and cross-validation points). Including the number measurements N is important as the number can change from one mini-batch to the other. The likelihood of the observations $p(y_{ij}|\hat{y}_{ij}, \hat{\sigma}_{ij})$ is given by a Gaussian distribution:

$$p(y_{ij}|\hat{y}_{ij}, \hat{\sigma}_{ij}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{ij}^2}} \exp \left(-\frac{(y_{ij} - \hat{y}_{ij})^2}{2\hat{\sigma}_{ij}^2} \right) \tag{8}$$

For other remote sensed variables like chlorophyll or sediment concentration a log-normal distribution is probably more

25 appropriate. The cost function has finally the following form:

$$J(\hat{y}_{ij}, \hat{\sigma}_{ij}) = \frac{1}{2N} \sum_{ij} \left[\left(\frac{y_{ij} - \hat{y}_{ij}}{\hat{\sigma}_{ij}} \right)^2 + \log(\hat{\sigma}_{ij}^2) + 2\log(\sqrt{2\pi}) \right] \tag{9}$$

The first term of the cost function is directly related to the mean square error, but scaled by the estimated error standard deviation. The second term penalizes any over-estimation of the error standard deviation. The third term is a constant term which can be neglected in the following as it does not influence the gradient. The sum in the previous equation runs over all grid points where a measurement is available but excluding the measurements withheld for cross-validation as the later are

5   never used during training.

We used the Adam optimizer (Kingma and Ba, 2014) with the standard parameters for the learning rate $\alpha = 0.001$, the exponential decay rate for the first moment estimates $\beta_1 = 0.9$, and for the second-moment estimates $\beta_1 = 0.999$, regularization parameter $\epsilon = 10^{-8}$.

During the development of the neural network, it was clear that it tended to overfit the provided observations, leading

10   to degraded results when comparing the network to cross-validation data. Commonly used strategies were therefore used to avoid overfitting, namely introducing a drop-out layer between the fully connected layers of the network. The drop-out layer randomly sets, with a probability of 0.3, the output of these intermediate layers to zero during the training of the network. We also added some Gaussian-distributed noise to the input of the network with a zero mean and a standard deviation of 0.05^oC.

5   Results

15   Figure 2 shows the RMS error relative to the cross-validation dataset computed every ten epochs (during this reconstruction phase drop-out is disabled). There is an initial sharp decrease of the cross-validation error and after 200 epochs the RMS error has mostly stabilized but still presents some fluctuations. These fluctuations are due to the fact that the gradient computed at every optimization set is computed over a subset of the data and this subset varies at every optimization step. As mentioned before, in every minibatch a random subset (in form of clouds) of data is marked as missing and the gradient is computed over

20   this randomly changing subset which leads to some fluctuations in the gradient and thus in the parameters of the neural network. In order to obtain a better estimate of the reconstruction, we average the output of the neural network between epoch 200 and epoch 1000 (saved at every 10th epoch) which leads to a better reconstruction than every individual intermediate results. The expected error of the reconstruction is similarly averaged. Ideally, one would take the correlation of the error between the different reconstructions into account. Ignoring these error correlations results in overestimating the expected error of the

25   reconstruction.

Instead of using the average, the median reconstruction was also tested, as the median is more robust to outliers. The results were very similar and slightly better with the average instead of the median SST. In the following only the average estimate is used.
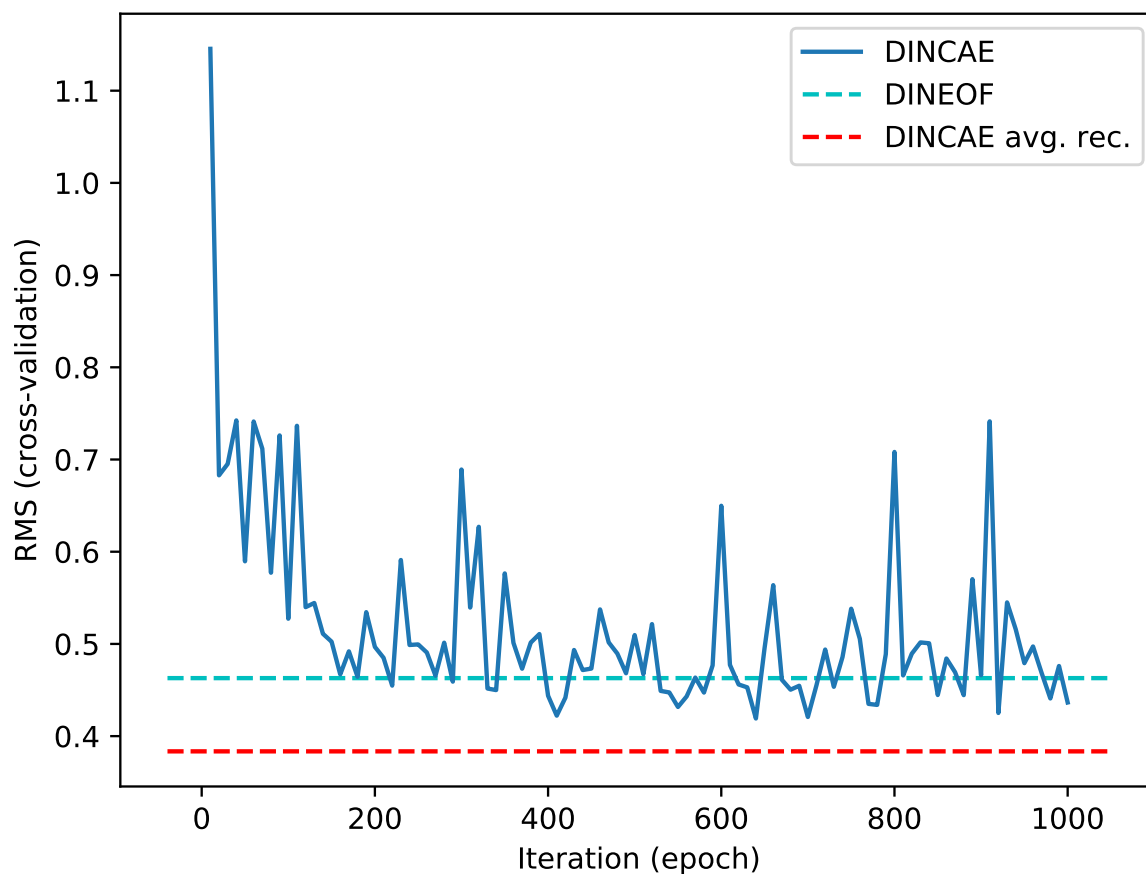
Figure 2. RMS difference with cross-validation dataset as a function of iteration. The solid blue line represent the DINCAE reconstruction at different steps of the iterative minimization algorithm. The dashed cyan line is the DINEOF reconstruction and the dashed red line is the average DINCAE reconstruction between epoch 200 and 1000.

Training this network for 1000 epochs takes 32 hours on a GeForce GTX 1080 and Intel Core i7-7700 with the neural network library tensorflow (Abadi et al., 2015). All computations are done in single precision.
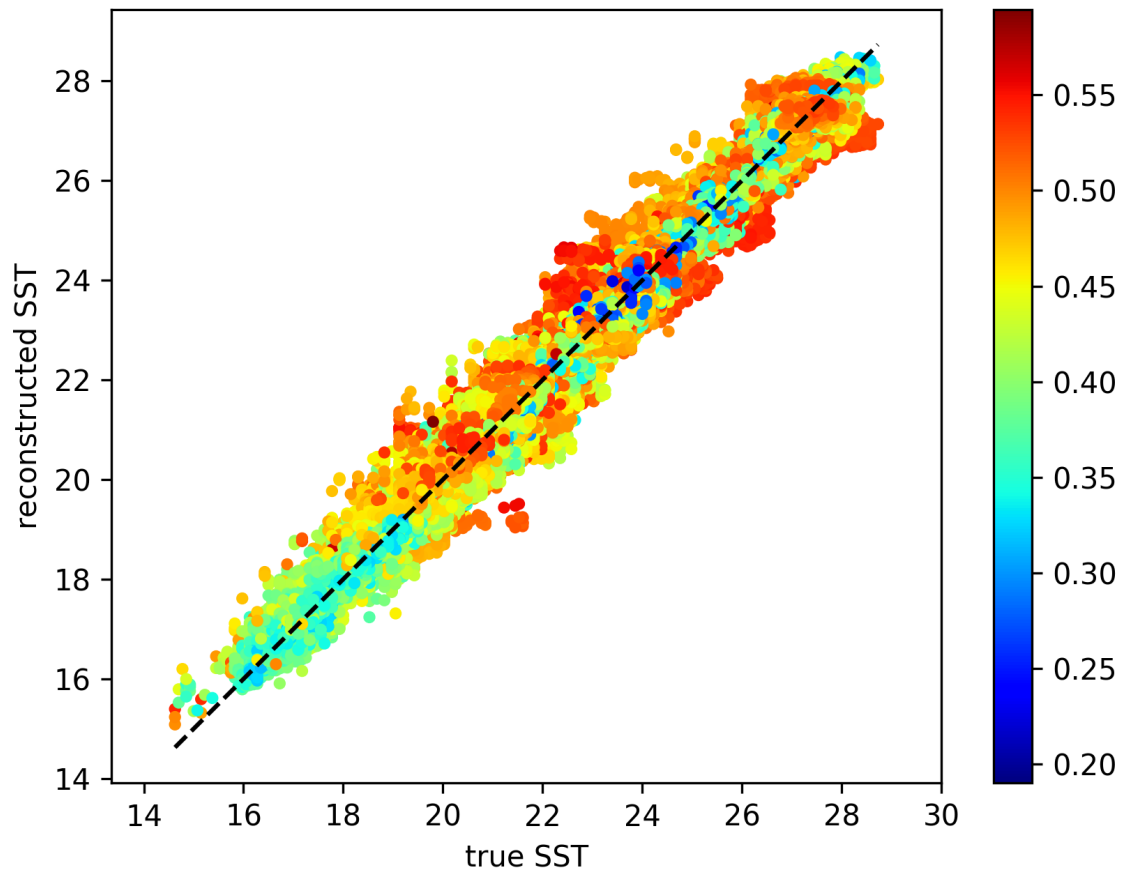
Figure 3. RMS difference with cross-validation dataset as a function of iteration

Figure 3 shows a scatter plot of the true SST (withheld during cross-validation) and the corresponding reconstructed SST. The color represents the estimated expected error standard deviation of the reconstruction. Low error values are expected to be closer to the dashed line. Reconstructed and cross-validation SST tends to cluster relatively well around the ideal dashed line. Typically the lower expected errors are found more often near the dashed line than at the edge of the cluster of points.
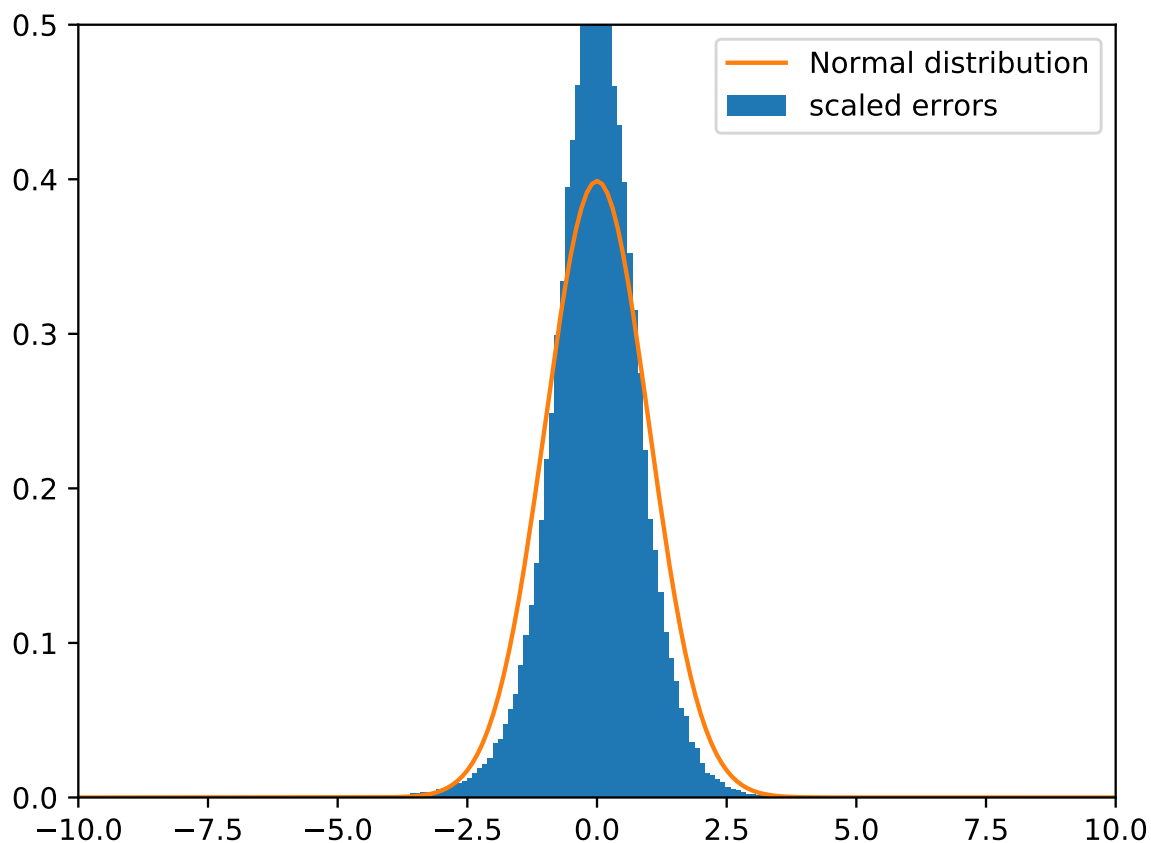
Figure 4. Scaled error are computed as the difference between the reconstructed SST and the actual measured SST (withheld during cross-validation) divided by the expected standard deviation error.

To obtain a clearer idea of the reliability of the expected error we computed the difference between the cross-validation SST and the reconstructed SST divided by the expected error standard deviation. A histogram of the scaled differences is shown in Figure 4. The scaled error follows the theoretical distribution relatively well. When a Gaussian pdf is fitted to the histogram of the scaled error, one obtains a mean of -0.02 and a standard deviation 0.85 (both adimensional). The averaged results of DIN-CAE underestimate the actual error by 15% but one can argue that an underestimation of the expected error of this magnitude should be acceptable for most purposes.

Different variants of the neural network are tested in order to optimize its structure. The number of skip connections has a quite significant impact on the results. The cross-validation RMS error is reduced from 0.4458 ^oC (no skip connections), to 0.4222 ^oC with 2 skips connections (layer 22 and 25 of Table 2), and further to 0.3900 ^oC with all skip connections between the

encoder and decoder layer of the same size. At the same time, the degree of smoothing of the used data (i.e. data not reserved for cross-validation) is reduced from 0.2957 ^{o}C (no skip connections) to 0.1383 ^{o}C with all skip connections.

Increasing the filter sizes of the convolutional layers from 16, 24, 36, 54 to 16, 32, 48, 64 (with the input convolution layer

5   fixed by 8 filters as it has to correspond to the number of inputs) and increasing the number of neurons of the bottleneck accordingly leads to a slight degradation for the present case compared to the cross-validation dataset, which indicates that the neural network starts to overfit if the number of filters is increased. A subsequent test with narrower convolutional layers of size 16, 22, 31 and 44 lead to very similar but slightly worse results with 0.3928 ^{o}C.

The DINCAE neural network with an increasing or decreasing number of layers did not improve the results. However, it is

10 possible that the depth of the neural network is dependent on the available training data set and that for a more extensive data, increasing the number of layers could have a positive effect.

Max pooling layers are used commonly in image classification problems (*e.g.* Simonyan and Zisserman, 2015; Krizhevsky et al., 2012) where the strongest detected feature is passed from one layer to the next. However, the purpose of this network

15 here is rather different as we intend to recover missing data which requires to spread the information spatially. Therefore we also tried the network with average pooling instead of max pooling, which further reduced the reconstruction error to 0.3835 ^{o}C.

In all cases the biases are relatively small and the present discussion is essentially also valid when considering the centred RMS (i.e. the RMS difference when the bias is removed). In the following we only use DINCAE with all skip connections and

20 4 convolutional layers with filter sizes 16, 24, 36, 54 and average pooling for future comparison.
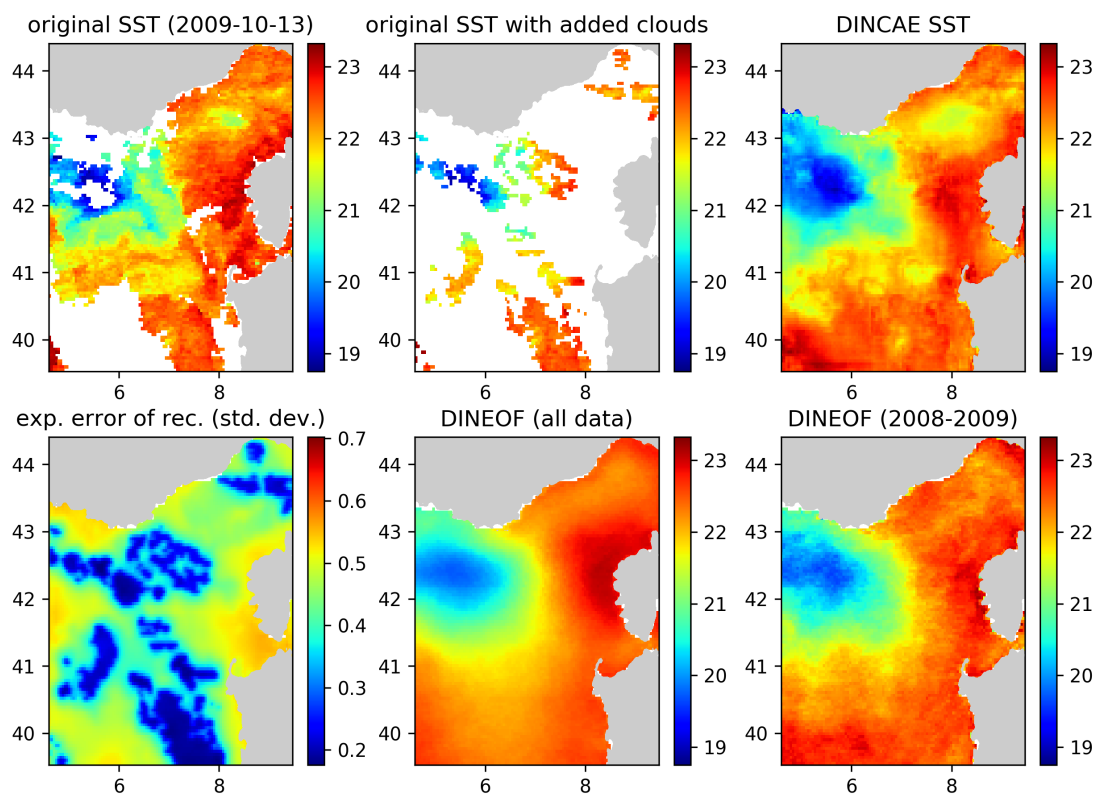
Figure 5. Example reconstruction with DINCAE and DINEOF for the date 2009-10-13

Figure 5 shows the SST reconstructions for 13 October 2009. The overall SST structure is reasonable in all reconstructions. The cold water in the western part of the domain is better defined in the DINCAE reconstruction, and the general position of the 21°C isotherm agrees better with the SST observations in the DINCAE reconstruction than in the DINEOF results.

Table 3. Comparison with the World Ocean Database for SST grid points covered by clouds

	RMS	CRMS	bias
DINEOF	1.1676	1.1102	-0.3616
DINCAE	1.1362	1.0879	-0.3278
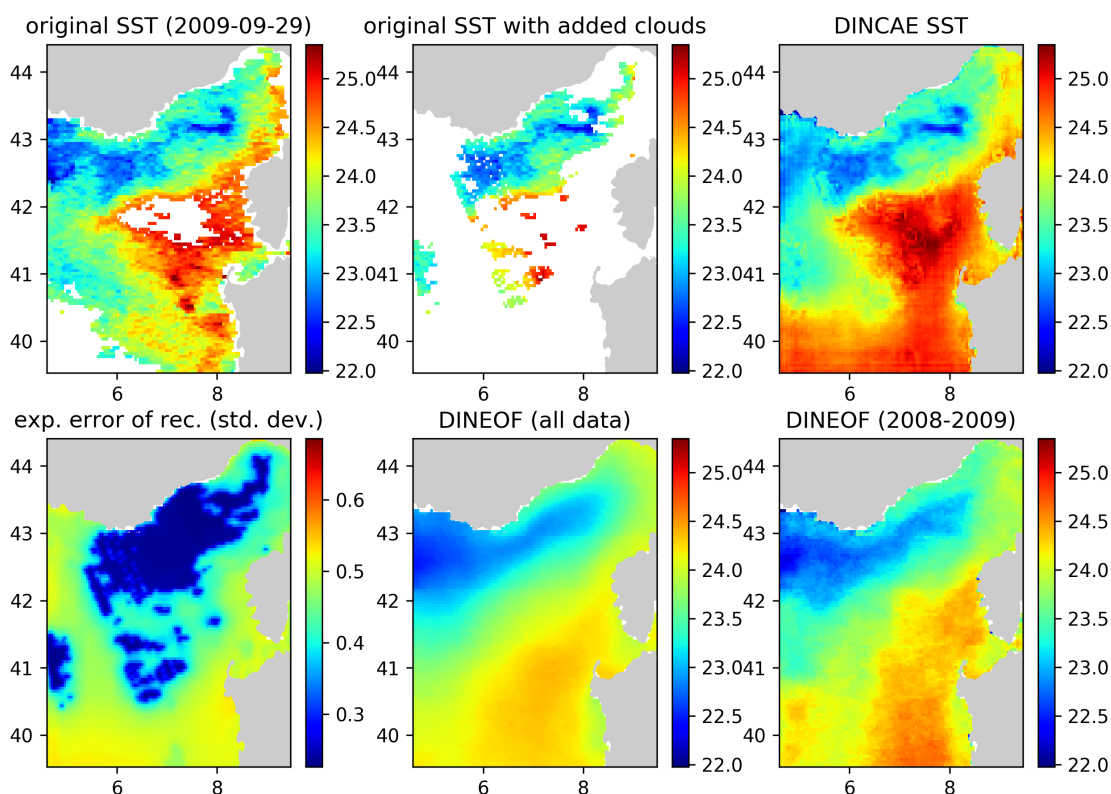
Figure 6. Example reconstruction with some artefacts for the date 2009-09-29

In some cases the DINCAE reconstruction introduces also some artefacts as some zonal and meridional gradients near the open boundaries (Figure 6). In the convolutional layers, zero padding is applied so that the convolution operation does not change the size of the arrays. As this issue is relatively localized at the border it is recommended that one chooses a slightly larger domain than the primary domain of interest for the reconstruction.

5   To further quantify how well the reconstruction methods could recover data under a cloud cover, we use in situ temperature from the World Ocean Database 2018 (Boyer et al., 2018). For every in situ grid point, the SST image with the same time stamp

(ignoring hour, minutes and seconds) is interpolated to the nearest grid cell relative to the location of the in situ observation. Only in situ observations corresponding to a cloudy SST pixel are used in the following. The depth of the in situ observations should be between 0.5 m and 1 m and if there are multiple data points between this depth range, the data point closest to the surface is used. As expected, biases play now a more important role when comparing in situ observations with reconstructed
5   satellite data (3). DINCAE represented a small improvement relative to the DINEOF reconstruction confirming the results from the cross-validation comparison.

    In Figure 7 the variability of the reconstructed SST dataset is assessed. These figures represent the standard deviation relative to a yearly average climatology computed for the original SST, and the reconstructions from DINCAE and DINEOF.
10 For the original SST, the climatological mean SST and the standard deviation were computed only using the available data. The standard deviation derived from DINCAE matches well the standard deviation from the original data, in particular in the interior of the domain, but the standard deviation is too large along the southern coast of France and Corsica. The DINEOF standard deviation matches the original SST standard deviation better in those areas but generally underestimates the SST standard deviation. Given the fact that DINCAE tends to retain more variability in the reconstruction is it thus remarkable
15 that it still features a lower RMS despite the so-called double penalty effect (Gilleland et al., 2009; Ebert et al., 2013), *i.e.* RMS-based error measures tend to be lower for smoother fields with lower variability, but this is not the case here.
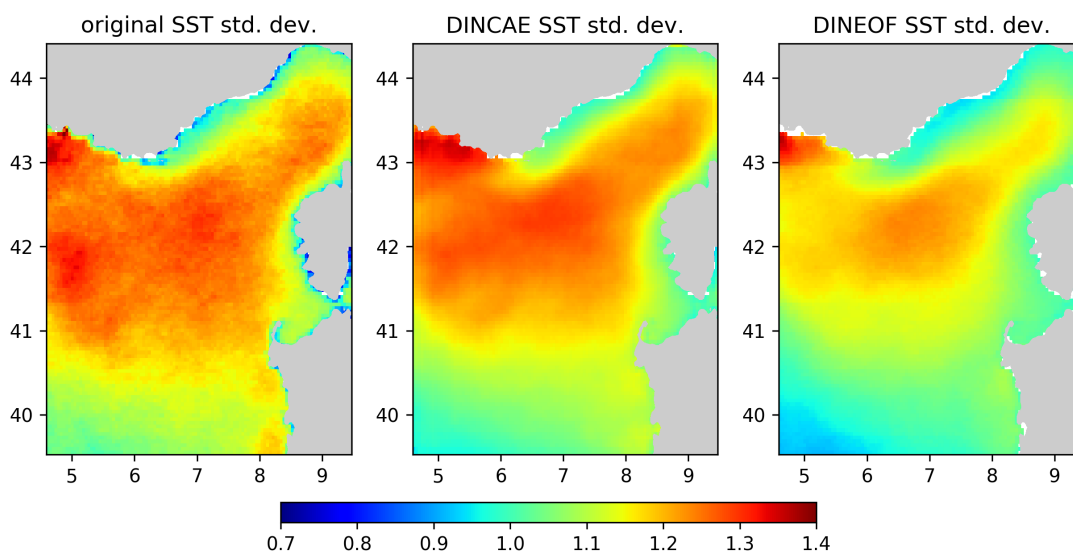
Figure 7. Standard deviation computed around the seasonal average

6   Conclusions

This paper presents a consistent and practical way to handle missing data in satellite images for neural networks. Essentially, the neural network uses the measured data divided by its expected error variance. Missing data are thus treated as data with an infinitely large error variance. The cost function of the neural network is chosen such that the network provides the reconstruc-

5   tion but also the confidence of the reconstruction error (quantified by the expected error variance). An over- or underestimation of the expected error variance are both penalized by maximising the likelihood and assuming Gaussian distributed errors. This approach can be easily generalized to parametric probability distributions, in particular to log-normal distributions for concentrations like remote sensed chlorophyll concentration.

10   The presented reconstruction method DINCAE compared favourably to the widely used DINEOF reconstruction method which is based on a truncated EOF analysis. Formally there are similarities between an auto-encoder (composed by just 2 fully-connected layers) and an EOF projection followed by an EOF reconstruction (Chicco et al., 2014). However, neural networks can represent non-linear relationships which is not possible with an EOF approach. Both methods were compared by

cross-validation and the DINCAE method resulted in RMS error reduction from 0.4629^oC to 0.3835^oC.

The expected error for the reconstruction reflects well the areas covered by the satellite measurements as well as the areas with more intrinsic variability (like meanders of the Northern Current). The expected error predicted by the neural network
5 provides a good indication of the accuracy of the reconstruction.

The accuracy of the reconstructed data under clouds was also assessed by comparing the results to in situ observations of the World Ocean Database 2018. Also compared to this dataset, the RMS error of the DINCAE reconstruction is lower than the corresponding results from DINEOF.

It is quite common that data analysis methods to reconstruct missing data tend to smooth the available observations in order to fill the area of missing observations. Therefore, the temporal variability (relative to the seasonal cycle) of the reconstructed sea surface temperature was computed from the original data and from the reconstructed data using DINCAE and DINEOF. The variability of the reconstructed SST with DINEOF generally underestimated the variability in the original dataset, but the
15 variability of the DINCAE reconstruction matched the variability of the original data relatively well.

Code availability. The source code is released as open source under the terms of the GNU General Public Licence v2 (or, at your option, any later version) and available at the address https://github.com/gher-ulg/DINCAE and https://doi.org/10.5281/zenodo.3251813.

Author contributions. A.B. designed and implemented the neural network. A.A.A made the DINEOF simulations. A.B., A.A.A., M.L. and J.M.B. contributed to the planning and discussions and to the writing of the manuscript.

20 *Competing interests.* The authors have no competing interests

Acknowledgements. The F.R.S.-FNRS (Fonds de la Recherche Scientifique de Belgique) is acknowledged for funding the position of Alexander Barth. This research was partly performed with funding from the Belgian Science Policy Office (BELSPO) STEREO programme in the framework of the MULTI-SYNC project (contract SR/00/359). Matjaz Licer would like to acknowledge COST action ES1402 - "Evaluation of Ocean Syntheses" for funding his contribution to this work. Computational resources have been provided in part by the Consortium des
25 Équipements de Calcul Intensif (CÉCI), funded by the F.R.S.-FNRS under Grant No. 2.5020.11 and by the Walloon Region. The AVHRR v5 dataset were obtained from the NASA EOSDIS Physical Oceanography Distributed Active Archive Center (PO.DAAC) at the Jet Propulsion Laboratory, Pasadena, CA. The National Centers for Environmental Information (NOAA, USA) and the (International Oceanographic Data and Information Exchange (IODE) are thanked for the World Ocean Database 2018.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Good-fellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F.,

5   Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/, software available from tensorflow.org, 2015.

Alvera-Azcárate, A., Barth, A., Rixen, M., and Beckers, J.-M.: Reconstruction of incomplete oceanographic data sets using Empirical Orthogonal Functions. Application to the Adriatic Sea Surface Temperature., Ocean Modelling, 9, 325–346, https://doi.org/10.1016/j.ocemod.2004.08.001, http://hdl.handle.net/2268/4296, 2005.

10   Alvera-Azcárate, A., Barth, A., Beckers, J.-M., and Weisberg, R. H.: Multivariate reconstruction of missing data in sea surface temperature, chlorophyll and wind satellite field, Journal of Geophysical Research, 112, C03 008, https://doi.org/10.1029/2006JC003660, http://hdl.handle.net/2268/9485, 2007.

Alvera-Azcárate, A., Barth, A., Sirjacobs, D., and Beckers, J.-M.: Enhancing temporal correlations in EOF expansions for the reconstruction of missing data using DINEOF, Ocean Science, 5, 475–485, https://doi.org/10.5194/os-5-475-2009, http://www.ocean-sci.net/5/475/

15   2009/, 2009.

Alvera-Azcárate, A., Barth, A., Parard, G., and Beckers, J.-M.: Analysis of SMOS sea surface salinity data using DINEOF, Remote Sensing of Environment, 180, 137 – 145, https://doi.org/10.1016/j.rse.2016.02.044, special Issue: ESA's Soil Moisture and Ocean Salinity Mission - Achievements and Applications, 2016.

AVHRR Data: NODC and Rosenstiel School of Marine and Atmospheric Science, AVHRR Pathfinder Level 3 Monthly Daytime SST Version

20   5. Ver. 5., https://doi.org/10.5067/PATHF-MOD50, pO.DAAC, CA, USA. Dataset accessed 2019-03-15, 2003.

Beckers, J.-M. and Rixen, M.: EOF calculation and data filling from incomplete oceanographic datasets, Journal of Atmospheric and Oceanic Technology, 20, 1839–1856, https://doi.org/10.1175/1520-0426(2003)020<1839:ECADFF>2.0.CO;2, 2003.

Bolton, T. and Zanna, L.: Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization, Journal of Advances in Modeling Earth Systems, 11, 376–399, https://doi.org/10.1029/2018MS001472, 2019.

25   Boyer, T. P., Baranova, O. K., Coleman, C., Garcia, H. E., Grodsky, A., Locarnini, R. A., Mishonov, A. V., Paver, C. R., Reagan, J. R., Seidov, D., Smolyar, I. V., Weathers, K., and Zweng, M. M.: World Ocean Database 2018, Tech. rep., NOAA, NOAA Atlas NESDIS 87, 2018.

Bretherton, F. P., Davis, R. E., and Fandry, C. B.: A technique for objective analysis and design of oceanographic experiment applied to MODE-73., Deep–Sea Research, 23, 559–582, https://doi.org/10.1016/0011-7471(76)90001-2, 1976.

Buongiorno Nardelli, B.: A novel approach for the high-resolution interpolation of in situ sea surface salinity, Journal of Atmospheric and

30   Oceanic Technology, 29, 867–879, https://doi.org/10.1175/JTECH-D-11-00099.1, 2012.

Chicco, D., Sadowski, P., and Baldi, P.: Deep Autoencoder Neural Networks for Gene Ontology Annotation Predictions, in: Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '14, pp. 533–540, ACM, New York, NY, USA, https://doi.org/10.1145/2649387.2649442, 2014.

Ebert, E., Wilson, L., Weigel, A., Mittermaier, M., Nurmi, P., Gill, P., Göber, M., Joslyn, S., Brown, B., Fowler, T., and Watkins, A.: Progress

35   and challenges in forecast verification, Meteorological Applications, 20, 130–139, https://doi.org/10.1002/met.1392, 2013.

Evans, B., Vasquez, J., and Casey, K. S.: 4 km Pathfinder Version 5 User Guide, NOAA, https://www.nodc.noaa.gov/SatelliteData/pathfinder4km/userguide.html, 2009.

Garcia-Gorriz, E. and Garcia-Sanchez, J.: Prediction of sea surface temperatures in the western Mediterranean Sea by neural networks using satellite observations, Geophysical Research Letters, 34, https://doi.org/10.1029/2007GL029888, 2007.

Geng, J., Fan, J., Wang, H., Ma, X., Li, B., and Chen, F.: High-Resolution SAR Image Classification via Deep Convolutional Autoencoders, IEEE Geoscience and Remote Sensing Letters, 12, 2351–2355, https://doi.org/10.1109/LGRS.2015.2478256, 2015.

5 Gilleland, E., Ahijevych, D., Brown, B., Casati, B., and Ebert, E.: Intercomparison of spatial forecast verification methods, Weather and Forecasting, 24, 1416–1430, https://doi.org/10.1175/2009WAF2222269.1, 2009.

Hinton, G. E. and Salakhutdinov, R. R.: Reducing the Dimensionality of Data with Neural Networks, Science, 313, 504–507, https://doi.org/10.1126/science.1127647, https://science.sciencemag.org/content/313/5786/504, 2006.

Kilpatrick, K. A., Podestá, G. P., and Evans, R.: Overview of the NOAA/NASA advanced very high resolution radiometer Pathfinder
10   algorithm for sea surface temperature and associated matchup database, Journal of Geophysical Research: Oceans, 106, 9179–9197, https://doi.org/10.1029/1999JC000065, 2001.

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, CoRR, abs/1412.6980, http://arxiv.org/abs/1412.6980, 2014.

Krasnopolsky, V., Nadiga, S., Mehra, A., Bayler, E., and Behringer, D.: Neural Networks Technique for Filling Gaps in Satellite Measurements: Application to Ocean Color Observations, Computational Intelligence and Neuroscience, 2016,
15   https://doi.org/10.1155/2016/6156513, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems 25, edited by Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., pp. 1097–1105, Curran Associates, Inc., http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, 2012.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE,
20   86, 2278–2324, 1998.

Maas, A. L., Hannun, A. Y., and Ng, A. Y.: Rectifier nonlinearities improve neural network acoustic models, in: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf, 2013.

Pisoni, E., Pastor, F., and Volta, M.: Artificial Neural Networks to reconstruct incomplete satellite data: application to the Mediterranean Sea Surface Temperature, Nonlinear Processes in Geophysics, 15, 61–70, https://doi.org/10.5194/npg-15-61-2008, https://www.
25   nonlin-processes-geophys.net/15/61/2008/, 2008.

Rasp, S., Pritchard, M. S., and Gentine, P.: Deep learning to represent subgrid processes in climate models, Proceedings of the National Academy of Sciences, 115, 9684–9689, https://doi.org/10.1073/pnas.1810286115, https://www.pnas.org/content/115/39/9684, 2018.

Ronneberger, O., Fischer, P., and Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, edited by Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., pp.
30   234–241, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Scherer, D., Müller, A., and Behnke, S.: Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, in: Artificial Neural Networks – ICANN 2010, edited by Diamantaras, K., Duch, W., and Iliadis, L. S., pp. 92–101, Springer Berlin Heidelberg, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-15825-4_10, 2010.

Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, in: International Conference on
35   Learning Representations, https://arxiv.org/abs/1409.1556, 2015.

Zhou, Y., Wang, H., Xu, F., and Jin, Y.: Polarimetric SAR Image Classification Using Deep Convolutional Neural Networks, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, 13, 1935 – 1939, https://doi.org/10.1109/LGRS.2016.2618840, 2016.