Author's response

Reviewer 1, 3rd December 2019

We would like to thank the reviewer for carefully reading the manuscript and providing constructive criticism. We hope that our reply below answers the questions and comments by the reviewer in an adequate manner. Our response is in bold-face.

General Comments: It is an interesting paper overall. The author uses Auto Encoder to reconstruct the missing data commonly found in optical satellite remote sensing caused by instrument failure or cloud cover. The author uses an interesting way to handle missing data in training image. As compared to the widely used DINEOF method, the author showed that DINCAE can, on some degree, produce better results measured in RMSE metrics, as well as spatial distributions of SST. From the technique side, Auto Encoder is a commonly used machine learning method in semantic segmentation and object detection. The author uses this method to solve, particularly SST, reconstruction obtained from satellite remote sensing. It is an interesting and meaningful problem to tackle. But as for developing a new methodology, I have some concerns.

1. The applicability of this method to other satellite measurements. Variables such as SST, have low frequency variability both in space and time (If I am right). This nature suggest that it is relatively easier for CNN to estimate the spatial correlation (e.g. for an image in which there are multiple people, it is harder to do segmentation than for an image with only lawn and sky). This also gives ground that average pooling turns out to achieve better results than max pooling, as stated in paper. For variables, especially those on land, such as plant reflectance, usually have high frequency variability both in space and time due to heterogeneous growth stage, background, and so on. These feature creates additional challenges, which I think, cannot be handled with the method configuration stated in paper. It will be interesting to see how it does (This may not directly related to the topic of this paper). Additionally, the method is tested at one site,which hardly persuasive to show its applicability over the globe. Will a model trained at one site be able to use at another site, or it is needed to develop a new model to a new site, which usually needs a lot of work to prepare data, training model, parameter tuning and so on? If so, from model deployment side, what the advantage of using it?

   We think that it is quite normal to first apply a technique to a limited area before addressing the problem at global scale and we choose a parameter which is quite important for oceanography. The initial papers of DINEOF (Beckers and Rixen, 2003; Alvera-Azcárate et al., 2005) focused also on sea surface temperature reconstruction and in subsequent papers it was shown that it can also be applied to other ocean parameters with (some adaptations) such as Chlorophyll concentration, sea surface salinity and sea surface currents. We are envisioning a similar path for the DINCAE technique. As our background is in physical oceanography, we can only speculate whether this technique can be applied to land-based data. The underlying motivation of DINEOF (and to some extent of DINCAE) is the fact that a large fraction (90% and more) of the variability of the ocean (as obtained from remote sensing data) can be explained by a reduced number modes (often 10 to 50 modes). So even when the satellite scene is partially obscured the missing data can be recovered using the data present in the satellite scene because the number of inherent degrees of freedom is relatively low. The reviewer mentions the case of plant reflectance which seems indeed to be a case where the number of degrees of freedom is apparently much higher and where it can be indeed difficult to use the same method. But we would say that this is a difficult case for any reconstruction method because the data actually measured by the satellite has less information on the data obscured by e.g. clouds. For such cases, it is particularly important to associate a reconstructed scene with a reliable error estimate. At least for sea surface temperature we were able to demonstrate that this could be done.

   In the same way that the EOFs obtained from DINEOF are specific and only optimal to the studied zone, we see that the network of DINCAE is, until the contrary is proven, only specific to a given zone. The aim of this paper is

to use the typical use-case of DINEOF (limited zone, ocean parameter) and to see if DINCAE can provide a better reconstruction than DINEOF. Many other reconstruction techniques used in oceanography, such as optimal interpolation or variational analysis, require some set of parameters to be tuned to a specific site. For DINCAE it is rather the structure of the network which can be optimized for a given site but there is arguably a greater chance that these depend less of the studied site than for example parameters like the correlation length in optimal interpolation. We actually have good results using the present network structure on the Adriatic Sea and we have been contacted by a researcher using the same network architecture on the South China Sea and West Philippine Sea providing a convincing reconstruction.

For this paper we worked on a regional scale, because we believe that this matches the typical approach of oceanographic studies which focus on a specific zone of interest and improve the understanding of the processes in this area (instead of trying to understand a process directly on a global scale).

2. Temporal feature of reconstructed variables EOF method is essentially PCA analysis. DINEOF method does take into consideration of both temporal and spatial correlation of variables, to my understanding. Though DINCAE, as described in the paper, also uses the spatial and temporal correlation of variables, it only uses correlation presented in 3 days (the day, the day before and the day after). In other words, spatial information is what it uses mainly for reconstructing. Do you have persuasive arguments that 3 days correlation in time are enough to capture temporal dependency? However, longer time dependency, e.g. seasonality, may also be important on estimating missing values. In this case, network configuration both capture spatial and temporal structure of variables (e.g. LSTM + CNN) could be more general and powerful.

The cloud cover varies normally quite rapidly from one day to the next as it does so on the time scale of a couple of hours as revealed by geostationary satellites like SEVIRI. For polar orbiting satellite we typically have a sea-surface image every day. So the one day before and one day after, is justified by the fact that we will have a reasonable chance that a pixel covered at a given day is not covered by the day before or the day after. Providing more than just 3 days could improve the performance as it would increase the available information, but it could also increase the risk of overfitting. As a response to the reviewer, we also tried with 5 time instances but it degraded the results. The following has been added to the manuscript:

For every time instance we use the data from 3 time instances in the reconstruction: the current day, as well the data from the previous and next day. As a variant of the previous reconstruction experiment we increase the number of time instance from 3 to 5 centered at the current time instance. However, the cross-validation error for this experiment is 0.433 ^{o}C and the results are not improved. Increasing the number of input features can aggravate the potential for overfitting as the number of parameters in the neural network is increased. A combination of convolutional neural network with recurrent neural networks (like Long short-term memory, LSTM) might be a better way to include the time dependencies.

The idea using an LSTM is indeed an interesting idea. But we rather think this should be addressed in a follow-up study as we were able to show progress using the present structure of the neural network.

The present technique uses also the day-of-the-year as input of the neural network so the information about the season is available to the neural network. The day-of-the-year is transformed by a cosinus and sinus specifically to facilitate the representation of the seasonal cycle (e.g. the 1st January is as close to the 2nd January as the 31st December).

Technique Comments

Page 1 line 2: 'A method to reconstruct missing data in satellite data using a neural network is presented' The first sentence is not as precise as it should be. As the first impression that this paper is going to introduce a neural network based method

to reconstruct/interpolate gappy satellite images caused by cloud coverage, instrument failures (e.g. LandSat 7) and so on. However the following paper mostly discussed an AutoEncoder method to reconstruct SST and tested only on SST.

While we think that the method is generic, we have only tested it on SST and thus we changed the abstract accordingly as suggested by the reviewer to make it clear that so far we only demonstrated its use for SST. The abstract now starts with:

A method to reconstruct missing data in sea surface temperature data using a neural network is presented.

This matches in fact the scope as set by the title of the manuscript which also mentioned specifically sea surface temperature.

Page 2 line 31: 'effectively reducing....' What is the meaning of putting this sentence here.

We think that the dimensionality reduction is a central aspect for the reconstruction of missing data. This aspect is actually shared with DINEOF. To make this clear we have expanded this paragraph.

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. Projecting the input data on a low-dimensional subspace is also the central idea of DINEOF, where it is achieved by an EOF decomposition.

Page 4. Figure 1 caption 'The arrow represent...' There is no arrow on figure

Unfortunately, we included an earlier version of the figure in the manuscript (without the arrows). The correct figure is the one below and the manuscript is updated.

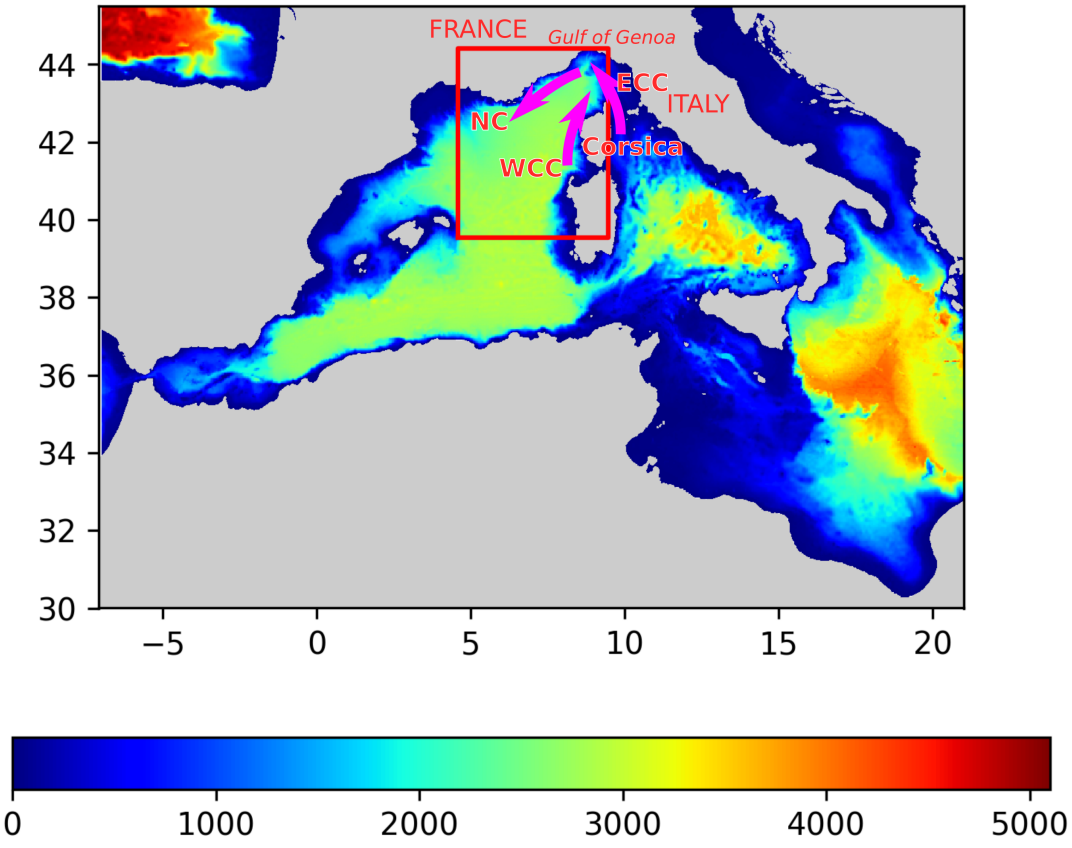

Figure 1. The red rectangle delimits the studied region and the color represents the bathymetry in meters. The arrows represent the main currents: the Western Corsican Current (WCC), the Eastern Corsican Current (ECC) and the Northern Corsican Current (NC)

Page 4 line 6: 'so that for a given date also the satellite' Delete 'also'

Ok, done, thanks.

Page 5 line 12, '...in the following' Delete 'in the following'

Ok, done, thanks.

Page 5 line 19 'assimilation of data' Change to 'data assimilation'

Ok, done, thanks.

Page 7 line 20 'skip connection' Does the resolution of SST data have effect on how you use skip connections? How large scale is called large scale for resolution of 4KM by 4KM, how about SST with resolution 1KM by 1KM. From another point of view,this operation again consolidate to use the spatial information for reconstruction, while temporal information somehow is ignored.

For us, large-scale refers to scale of variability which affects the SST over the entire domain: for example the overall position of the main current and the heating and cooling related to the seasonal cycle. Short scale refers to mesoscale circulation features (visible also in SST) related to meanders and eddies which typically have a length-scale of 50 km.

The initial idea is that these large scales should go through the bottle-neck of the convolutional autoencoder while the small scales are handled by the skip connections (experiment labeled "DINCAE (2 skip connections)" in table 1). However, it turned out that it is beneficial to have these skip connections at all levels of the convolutional neural network (experiment labeled "DINCAE (all skip connections)") so that a distinction between scale with and without skip connections (and large versus small scale) is no longer necessary.

Page 8 line 5 The two parameters here seemly have profound effect on reconstruction result, how does these two parameter chosen?

It is not clear to us, why the reviewer thinks that these parameters have a profound effect on the reconstruction. The range of allowed values are very far from restrictive. We added the following to clarify this point.

The effective range of the error standard deviation is thus from $\exp(-\gamma/2) = 0.0067\,^\circ\mathrm{C}$ **to** $\delta^{-\frac{1}{2}} = 31.6\,^\circ C$ **which is a relatively wide range as the error is expected to be O(0.1) to O(1)** ^{o}**C . The bounds are only effective during the very first epochs of the neural network where the weights are still close to random values.**

Page 9 line 15 'the output of the neural network is a Gaussian probability distribution' The author assume the output is a Gaussian distribution, 'is a Gaussian distribution' means the author know it is Gaussian.

We agree and changed "is a Gaussian probability distribution" by "is assumed to be a Gaussian probability distribution".

Page 10 line 18-21 'As mentioned before, ...neural network' Not quite understand the training procedure here. 'a random subset of data is marked as missing'? Since the missing data is marked randomly for each epoch, it is possible that at epoch = k, some part of data is marked as missing, while at epoch = k+1, the same part of data of the same image is marked as available. If this is the case, it essentially means the model was told what it should predict randomly? This is somewhat contradictory with Page 9 line 10.

We agree that this part is confusing and more information is added to the manuscript. First, we want to explain how a traditional auto-encoder works:

- **Some data are marked for validation and never used during training**
- **The network is given some data as input and produce an output which should be as close as possible to the input. So all training data are given at all epochs to the network**
- **The network is validated using the validation data set aside.**

So the traditional auto-enoder optimises how well the provided input data can be recovered after dimensionality reduction.

In our approach, there are two steps where data are intentionally hidden to the network:

1. **The validation data that were set aside and never used during the training (page 3, line 19 of the original manuscript), similar to the traditional auto-encoder.**

2. **Some additional data in every minibatch were set aside to compute the reconstruction error and its gradient (unlike the traditional auto-encoder). This additional subset is chosen at random.**

This is done because the main purpose of the network is to assess the ability of the network to reconstruct the missing data using the available data. In fact, we are not withholding less data than the traditional auto-encoder. The downside of the approach is that the cost function fluctuates more because it is computed only over a relatively smaller set of data. But for us this is acceptable (and controlled by taking the average of the output of the network at several epochs) because the cost function reflects more closely the objective: reconstruct missing data from the available data (instead of reproducing the input data as it is the case of the traditional auto-encoder).

The traditional auto-encoder approach trained using only clear images was not considered because only 13 images of out 5266 have a cloud coverage of less than 5%. So the ability the handle missing data was for us a requirement from the start.

Concerning the specific question "Since the missing data is marked randomly for each epoch, it is possible that at epoch = k, some part of data is marked as missing, while at epoch = k+1, the same part of data of the same image is marked as available. . . .". The reviewer is right in its interpretation. But this is always the case in supervised learning. The gradients are computed using observations (or true labels,...) of the training dataset and observations are used multiple times (once in every epoch). But of course, the validation dataset is never used during training and used only at the last step to assess the accuracy of the network.

Page 10 line 21-22 'we average ...intermediate result' Why do not average multiple runs?

We agree that averaging of multiple runs would be preferable but it would increase tremendously the computation time, by e.g. a factor of 30 if one would average over 30 runs for example. We added the following to the manuscript:

Alternatively one would average the output of an ensemble of neural networks initialized with different weights (and possibly using different structures) but this would significantly increase the necessary computing resources of the technique (Krizhevsky et al., 2012). But this ensemble averaging approach could be beneficial to improve the representation of the expected error and the accuracy of the reconstruction.

Page 11 Figure 2 caption 'red dash line ...' How come the average DINCAE reconstruction is smaller than RMSE at any given epoch? Also, the error curve indicates that the model has no sign of convergence. I bet if you continue training the model for another 1000 epochs, the cross validation error curves will not converge. This also indicates that there might be something wrong in the training procedure. Can you plot your lossfunction here as well?

It is quite common that the RMS error relative to a cross-validation data of a neural network does not converge. This is actually the basis of strategies like early stopping (*e.g.* Prechelt, 2012). The RMSE of the average DINCAE reconstruction is smaller than the RMSE at any given epoch because computing the RMSE is a non-linear operation. The DINCAE reconstruction at a given epoch included some variability which is not (or insufficiently) constrained by the observations. This explains also why the CV RMSE fluctuates. By taking the mean of the reconstruction at different epoch these fluctuations are averaged out and a better reconstruction is obtained. An alternative technique would be the use of an ensemble of neural networks (Krizhevsky et al., 2012) as noted also by the reviewer in his/her other comment.

The figure 2 shows the loss function for every minibatch. High fluctuations are quite apparent from this figure. But it is expected that the loss function using any optimization method based on mini-batch fluctuates (unless the learning explicitly is forced to zero, which is not the case here) because the loss function is evaluated using a different mini-batch at every iteration. Consequently the gradient of the cost function includes also some stochastic variability. Even if the dataset is small and the gradient could be computed over the entire dataset at once, using mini-batches is still advised because these fluctuations allow the cost function to get out of a local minima (Ge et al., 2015; Masters and Luschi,

2018). While the mini-batch selection effectively computes the gradient over a temporal subset, the additional data marked as missing within a minibatch is a spatial subset which enhances these fluctuations but allows us to define the cost function more closely to our objective (i.e. inferring the missing data from observations, as explained above). (The previous paragraph has also been added to the manuscript).

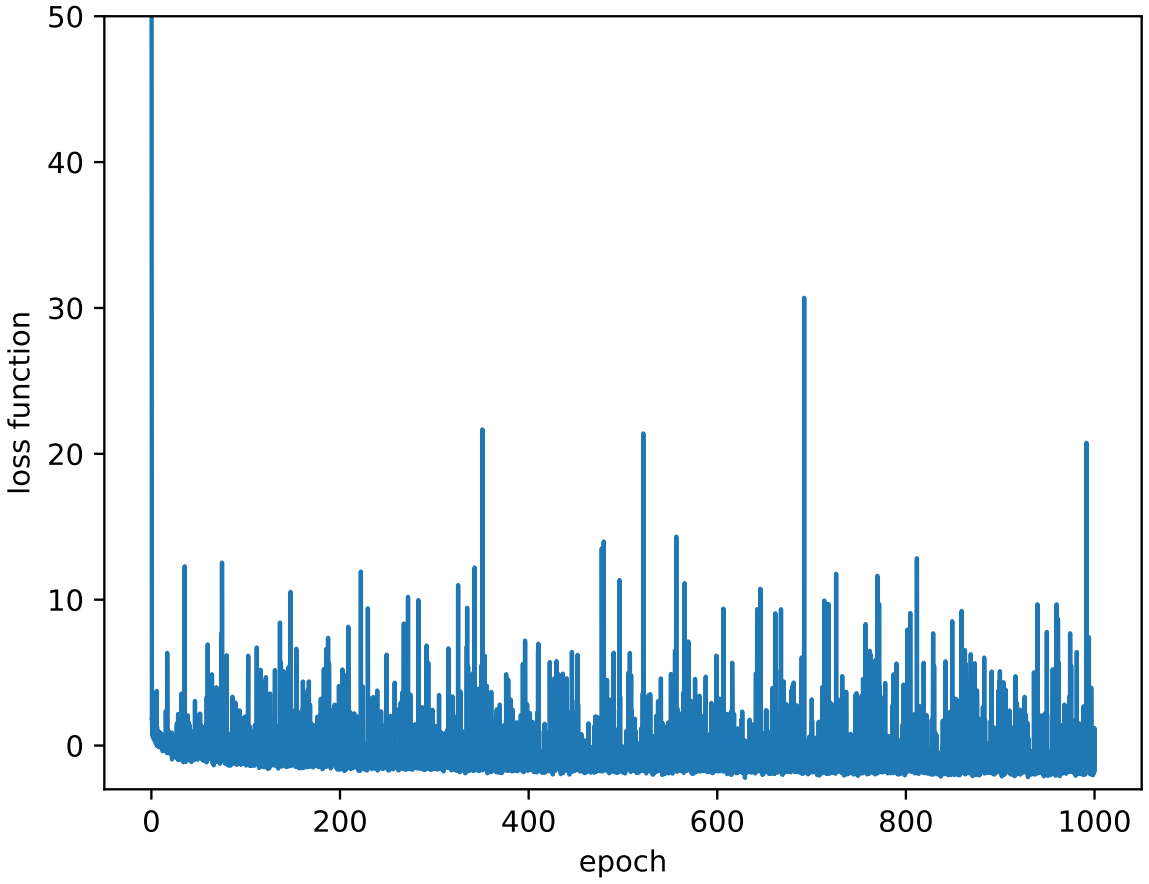

Figure 2. The loss function computed internally for every minibatch during the optimization.

Page 14 line 16 'also tried ...' The max pooling operation tries to extract distinct signals from neighbors, while average pooling operation tries to extract common signals from neighbors. For SST, which has low frequency variation in space, it makes sense average pooling should do better than max pooling.

Thanks for this remark. We added this interpretation to the manuscript. In fact, in the current research literature max pooling has completely replaced average pool prosed in the pioneering work from LeCun et al. (1998) for CNN and image recognition. It was a surprise to see that the seemingly outdated average pooling worked better than the max pooling for our case. But we agree with the interpretation of the reviewer which has been included in the revised manuscript. Another way to look at this is the fact that for a dynamical system in the linear regime, different flow features (solution to the underlying primitive equations) coexist and contribute in an additive way to the total flow.

Page 17 line 14 '...reconstruction is it thus...' Change 'is it' to 'it is

Sorry for the typo, and thank you for reading the manuscript so carefully to the end!

Reviewer 1, 13th December 2019

We thank the review again for this quick response and very insightful comments.

"This is indeed interesting to see" **Thank you for your encouragement.**

The channels of input data increased from 8 to 10. But the filter size, the number of output feature maps, and layer size, number of layers stay the same. Thus,the parameters of the network should stay the same. Right?

In addition to the SST (divided by the expected error variance), we are also providing the inverse of the expected error variance. The number of input channels changed from 4+3*2 = 10 to 4+5*2 = 14. The filter size of the first convolution network stayed at 16 filters. While previously the 3x3 convolution was realized with 3 x 3 x 10 x 16 (width x height x input channels x output channels), in the version with more time instances the convolution matrix had the dimensions 3 x 3 x 14 x 16 (so a 40% increase of parameters in the first convolutional layer). We added this to the revised manuscript. In the first submitted version of the manuscript we indeed wrote that the total size of the array is 8 x 112 x 112 x 5266. This should be 10 x 112 x 112 x 5266 and this is corrected in the revised version. We apologize for this confusion.

3. This is helpful. Previously, the author introduced two variables with no explaination of what and why. Previously, from the formula only, it seems like the value of these two variables will affect strongly the computation. e.g. delta = 100 vs.delta = 0.01

We agree that this part was unclear in the first submission and thank the reviewer for highlighting that the parameters were not properly discussed.

4. I am not very sure I fully understand it. But I will leave it to other reviewers!

Maybe it is clearer with an (admittedly) extreme example: if there is some part of the domain where there are no training data and this domain is dynamically completely disconnected from the rest, then its value is (per construction) completely unconstrained, except for an a priori information of reasonable values. So e.g. 10^{o}C is probably as good a guess as 14^{o}C. The neural network tends to oscillate between these two values because there are no constraints from the data. Assume further that we have a validation data point of 13^{o}C in this area, then the average RMS would be (abs(10-13) + abs(14-13))/2 = 2 ^{o}C but the RMS of the average is abs(12-13) = 1 ^{o}C. (for a single value the RMS is directly related to the absolute value which we use here to simplify the notation, but the same results are true for a series of numbers).

5. I would guess the fundamental reason why the RMSE and Loss fluctuate so much is that the random mark missing data in every mini batch. Because in every epoch, the spatial correlation of missing and available data is disrupted due to random marking, hence what the network has learned in previous epoch is disrupted as well, which eventually is reflected in RMSE and Loss. The fluctuations may not have so much to do with mini-batch optimization. Perhaps one way to check is to use same random mark missing data for every 20 epochs, and average at every 20 epochs. Just my opinion

We conducted this experiment and the reviewer is right that it had indeed a quite significant/dramatic effect on the convergence of the cost function (see the attached figure). Unfortunately, the average reconstructed SST or the reconstruction from the last epoch are not better than the best experiments that we had already in the manuscript.

The experiment "missing data change every epoch" is the experiment "DINCAE (all skip connections and average pooling)" from the manuscript. Despite that the results are not better in this case, the reviewer's idea is promising and

	last reconstruction	average reconstruction
missing data change every epoch	0.4968	0.3834
missing data change every 10 epochs	0.4423	0.4146
missing data change every 20 epochs	0.4387	0.3984

Table 1. CV error for different experiments keeping the mask of the missing data constant for several epochs

we include it as an option in our code. Application to other cases will tell if this proposed option (keeping the same data marked as missing for every 20 mini-batch) should rather be preferred.

Another way to interpret the marking of some data as missing would be to view it as a drop-out layer as the value of zero does indeed represent an "infinitely" large error. Change the mask of missing data at every epoch seems to help the generalization.

We also verified that the current version of our code reproduces exactly the same results as the code when the article was submitted if the same random seeds are used to exclude the possibility that any other change to the code has an impact here.

Reviewer 2, 8th February 2020

We would like to thank the reviewer for carefully reading the manuscript. Her/His complete review is copied below while our answers are inserted below every comment and written in boldface.

This study presents a novel approach of reconstructing sea surface temperatures from cloudy satellite data by making good use of modern deep learning techniques. While I believe that the study has been carefully designed and executed, I have severe problems with the paper in terms of its presentation and accuracy in writing. This study could become a high-impact publication if it were better structured and methods and outcomes were described clearer. I advise the authors to apply major revisions and seek the help of a native speaker to avoid erroneous or ambiguous statements. Below, please find my detailed comments.

All coauthors and I, tried our best to improve the manuscript to avoid erroneous or ambiguous statements. We hope that the manuscript is now clear.

Abstract: the sentence "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." is not very clear. Perhaps rephrase as "Contrary to standard image reconstruction with neural networks, this application requires a method to handle missing data (or data with variable accuracy)."

OK, we changed this sentence to:

> **Contrary to standard image reconstruction with neural networks, this application requires a method to handle missing data (or data with variable accuracy) already in the training phase.**

L7: suggest to remove "essentially"

Ok, done.

L9: what is "relatively long"? Provide a number, please.

We agree, and we changed the sentence to:

> **The approach, called DINCAE (Data-Interpolating Convolutional Auto-Encoder) is applied to a 25-year time-series of Advanced Very High Resolution Radiometer (AVHRR) sea surface temperature data**

L11: "a method to reconstruct missing data": suggest to rephrase "a previously pub- lished method", "the current standard method", "the state-of-the-art DINEOF method", or similar.

Thank you for the suggestion. In the revised manuscript we refer to the methods as "state-of-the-art".

L16: what is meant by "the ocean current signal"? A signal always refers to a measurement or sensing process. Here you want to refer to a physical process in the ocean.

We used the term signal indeed quite broadly in the original manuscript and replaced it in the revised manuscript with a more precise term. In the revised manuscript, this was changed to "the ocean velocity variability depends thus partially on ocean temperature".

L17: replace "like" with "e.g."

OK, done.

L20: replace "sensor" with "measurement". This sentence refers to the measurement principle, not the technical instrument, which performs the measurement.

The submitted manuscript reads:

> **However, as for any sensor working in the infrared or visible bands, clouds often obscure large parts of the field-of-view.**

In the revised manuscript we changed sensor by "measuring technique".

L22: "but often small scale information is filtered out because of the transient and stochastic nature of these structures." The transient and stochastic nature produces variability and is clearly not the reason why small scale information is filtered out. This is rather a result of the averaging procedures which are applied in practically all known techniques to interpolate. As this is a critical sentence for setting the stage of this study, please rethink the phrasing and provide a more precise description of the issue, which you are trying to solve.

We agree. The following has been added to the manuscript:

> **A truncated EOF decomposition will focus primarily in spatial structures with a "strong" signature (or more formally defined with a significant L2 norm compared to the total variance). Small scale structures can be included in a truncated EOF decomposition as long as their related variance is large enough to be present in the retained EOF modes. But small scale structures tend to be transient (short-lived) and therefore are often not retained in the dominant EOF modes. It should be noted that there is no explicit spatial filtering scale in DINEOF removing small scales (unlike other methods like optimal interpolation, kriging, spline interpolation). But in practice a similar smoothing effect is noticed because of the EOF truncation.**

We removed the terms "transient" and "stochastic" in the revised manuscript and we clarified that there is no explicit filtering in DINEOF by using a predefined spatial length-scale.

L23: DINEOF falls from the sky here. For non-experts in the field of sea surface temperature reconstructions, it is completely unclear what this is. Also, as DINEOF appears here for the first time, it is a must that you provide a reference. The reference comes two sentences further down, which is too late. A brief description of the method would be appropriate here.

The original manuscript was:

> **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).**

The reference was now put in the first sentence. The following information was added when describing DINEOF. More information has been added later in DINEOF sections, because it would be too technical for the introduction.

> **DINEOF (Data Interpolating Empirical Orthogonal Functions, Beckers and Rixen, 2003; Alvera-Azcárate et al., 2005), is an iterative method to reconstruct missing observations reducing noise in satellite datasets using empirical orthogonal functions (EOF). A truncated EOF decomposition using the leading EOFs is performed and the initially missing data are reconstructed using this EOF decomposition. The EOF decomposition and reconstruction is repeated until convergence.**

page 2: L9: "to detect the presence of non-linear, stochastic features" - I disagree that neural networks "detect" these features. Rather they are able to "learn" such features and thus potentially produce more detailed reconstructions of them.

OK, this is changed the revised manuscript:

> **Neural networks are therefore specially well positioned to learn nonlinear, stochastic features measured at the sea surface by satellite sensors, and their use might prove efficient in retaining these structures when analysing satellite data, for example for reconstructing missing data.**

L11: This statement is too general. I strongly suggest to first explain briefly what types of neural networks exist and how they can/could be used for the problem you want to solve (including references to the most important deep learning papers). Then you can make the argument that these networks (and in particular CNN derivates) are generally trained with complete data, whereas in your application you need to find a method which can train with scenes containing missing data, because there are no complete satellite scenes available (or only very few).

We added the following information to the manuscript to give a general overview of the types of neural networks. Later in the manuscript we will focus on some examples in oceanography.

> **Neural networks can be composed of a wide variety of building blocks, such as fully connected layers (Rosenblatt, 1958; Widrow and Hoff, 1962) recurrent networks (e.g. Long Short-term Memory (Hochreiter and Schmidhuber, 1997), Gated recurrent unit (Cho et al., 2014)), convolutional layers (LeCun et al., 1998; Krizhevsky et al., 2012). Recurrent networks work typically with a one dimensional list of inputs of a variable length (such as a text sentence). Fully connected layers and convolutional layers require to have a full dataset without missing data, at least for the training phase. For a review on neural networks the reader is referred to Schmidhuber (2015) and references therein.**

L14: this paragraph contains some of the literature review I am asking for in my previous comment. However, here it is on the one hand too specific (only ocean data applications), but on the other hand too superficial as it doesn't become clear why you need to develop a new approach and cannot simply apply for example the method of Krasnopolsky et al.

In the revised manuscript we first referenced the general idea in artificial neural networks and then give some examples in oceanography. The field is too wide to give a comprehensive overview of all applications in geoscience and we limit therefore the applications to oceanography. The updated manuscript goes into more details on the limitations of the method of Krasnopolsky et al. (2016):

> **The neural network by Krasnopolsky et al. (2016) uses as input satellite sea surface elevation, sea surface salinity, sea surface temperature and *in situ* Argo salinity and temperature vertical profiles with some auxiliary information (like longitude, latitude and time) to estimate the Chlorophyll-a concentration. The network does not use measured Chlorophyll-a concentration at a given location as input during inference (the reconstruction phase), nor the information from nearby grid points to infer Chlorophyll-a concentration. The network is exposed to the chlorophyll measurements only during the training phase.**

L33: I storngly suggest to re-organize the paper so that it follows the classical structure and describes the method before the data, and in particular before another reconstruction (DINEOF) is reported.

For us it was important to describe the data first because the ubiquity of clouds and the strong seasonal cycle in the data sets are an important constraint for the method. Therefore we believe that for most readers it will be easier to understand the method once its input (i.e. the satellite data) is presented. In fact, the type of data motivates the choices that were made during the design of the method. Also, we believe it helps the reader if we give concrete size of the arrays and matrices involved when the method is described. However, these matrix sizes depend on the data. Since the method section depends heavily on the data section, and the data section does not depend on the method section, we choose to present the data first. We actually tried to reverse the order as suggested by the reviewer but we ended up with too many forward references which would harm the readability.

We however presented the DINEOF method after the DINCAE method as suggested in the revised manuscript.

Page 3: L4: Delete the first sentence. You don't need a motivation within your "Data" section. Such an argument belongs in the introduction, if you wish to explain, for example, why you designed your study based on this dataset and not another one. In section 2 you should only describe the dataset, without any "discussion".

We deleted the first sentence.

L19: The cross-validation deserves more explanation, because you are publishing in a journal which is primarily read by non-experts in the field of machine learning. It is important to note that (contrary to standard image analysis) subsequent scenes from the AVHRR data are not independent. Therefore you cannot use a random sample to construct your test dataset (or "validation dataset", whichever terminology you prefer). I am wondering if 50 scenes are indeed sufficient to thoroughly test the generalization of the network. Not being an ocean scientist, I can only assume that typical transport time scales in your study region are on the order of a week(?). This would imply that the first 7 scenes of your "independent" test data are still "polluted" and thus not fully independent. Have you tried retaining a larger test sample?

In the revised manuscript we also added a reference to a classical textbook for the cross-validation method outside of the realm of machine learning.

> **To assess the accuracy of the reconstruction method, cross-validation is used (*e.g.* Wilks, 1995). For cross-validation a subset of the data is withheld from the analysis and the final reconstruction is compared to the withheld dataset to access its accuracy. Since clouds have a spatial extent, we wanted to withhold data with a similar spatial structure. In the last 50 images we removed data according to the cloud mask of the first 50 images of the SST time series. The last 50 images represent the data from 2009-09-25 to 2009-12-27 (since some scenes with too few data have been dropped as mentioned before). 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 (i.e. individual pixels) have been withheld this way.**

We did not try to have a temporal gap between the training data and test data or a larger test sample. It is true that there is some correlation between the training data and test data (the last few scenes used for learning might correlate with the first few scenes of the validation set). But we also performed a validation with in situ data in the manuscript. Both validation methods lead to the same outcome. We realize by reading the other comments from the reviewer that the reference to the table with in situ validation was not quite clear in the original manuscript.

For the "best" DINCAE experiment, we also recomputed the cross-validation RMS error using only the last 43 scences and the RMS error is with 0.3754 ^{o}C very similar (and even slightly lower) that the RMS error using the last 50 images 0.3834^{o}C. If there would be a significant "pollution" effect, then one would expect that the RMS error with 43 scence to be larger than with the 50 scences. But this was not observed. Given the large pool of training data (5216 scenes) the effect of a handful potentially correlated scenes does not have any significant effect.

L21: How can you retain > 100,000 measurements from 50 scenes? Are these in- dividual pixels, or did you actually apply cross-validation with random sampling, thus ignoring the argument I made above?

Yes, this is the count of individual pixels of the 50 images used for cross-validation as measured by the AVHRR sensor. We clarified this in the revised manuscript.

Page 4: Figure 1: I cannot see any arrows, which are referenced in the figure caption.

We are sorry about this problem. It has been corrected in the revised manuscript.

L3: Again, some explanation of DINEOF is warranted in this paper. It should be clear what this method does, without having to access the referenced papers. For details you can refer to them, but not for the fundamental "explanation".

We expand this section and included more information about how DINEOF reconstructed the missing data:

> **A truncated EOF decomposition using the leading N EOFs is performed and the initially missing data are reconstructed by combining the retained EOF modes and their corresponding amplitudes. The EOF decomposition and reconstruction of missing data is repeated until convergence. The optimal number of EOFs N is determined by cross-validation.**

Page 5: Table 1: instead of "fewer layers" or "more layers", the number of layers should be given.

We agree, and this has been added to the table ("fewer layers": 3 convolutional layers and "more layers": 5 convolutional layers)

L5: I don't think this is an appropriate citation here. It is the principle of EOFs to detect relations between variables and construct an orthogonal set of linear functions to model these relations. Due to the cutoff after N EOFs, there is always smoothing applied. A proper citation here would be some standard statistics book.

We added a reference to Wilks (1995).

L7: I am not at all surprised by this result: if you extend the timeseries, you will be more likely to sample patterns, which have not been observed before and which don't fit well to the already "learned" EOFs. Hence, there is less structure that can be described by the EOFs and more noise

Original manuscript reads:

> **As only 13 modes are retrained by DINEOF for the reconstruction, 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.**

Here we did not extend the time series but just used a subsample of the time series. For the full time series (1985-2009, 25 years), 13 modes have been retained as optimal. For a subset (2008-2009, 2 years), more EOFs (19 modes) have been retained. The number of time instances is an upper bound for the number of EOFs with non-zero singular values. For a shorter time series, this upper bound is thus lower, yet more EOFs have been retained with the shorter time series. This result was unexpected for us.

L14: I disagree that deep neural networks are "extensively" used in Earth sciences. This field is developing rapdily, but the applications are so far far from "extensive".

We agree and the sentence was revised:

Convolutional and other deep neural networks are extensively used in computer vision and find an increasing number of applications in Earth sciences [. . .]

Page 6: L6: what does "different errors" mean? Different to what?

The error can be different from one pixel to another. We changed this in the revised manuscript as "error varying in space and/or time" to be more clear.

Also: as mentioned before, for an article in this journal there needs to be a brief description of CAEs in the method section, which should contain enough information that an uninitiated reader (e.g. an ocean modeller) understands why the approach might actually work. Clearly, before the discussion taking place here, the reader must know how the network is constructed, which activation functions are used, which optimizer is used, whether regularization techniques are applied, etc. And the basis for the network built here is probably coming from some deep learning paper, which then needs to be cited. Such a description follows on page 7. Please restructure.

This is the structure of the original manuscript:
On page 6:

- **We explain how missing data is handled in data assimilation which motivates the present work**
- **Handling of missing data in the input is done in analogy here**
- **Describe the input of the neural network**

On page 7-8:

- **General structure of the network**
- **Skip connections**
- **activation function**

On page 9:

- **Cost function**

On page 10:

- **Optimizer**
- **regularization techniques**

To us, it seems logical and didactical to make a description of the method step-by-step: first the input of the network, then how the input is transformed by the network (general structure, activation function, skip connections), how to assess the accuracy of the output (cost function), how to optimize the accuracy and finally how to prevent overfitting (regularization techniques). In the revised manuscript we make it clear that the description of the convolutional autoencoder will come in the following.

In the revised manuscript, we put the reference to Hinton and Salakhutdinov (2006); Ronneberger et al. (2015) more prominently as the neural network structure proposed in those papers is quite similar to one used here. We also updated the overview of the paper in the introduction to make the links between sections clearer.

Page 7: L7: the references refer to convolutional layers only. However, you are applying an autoencoder approach, so the appropriate references should be made. For example: G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504, 2006.

Note that we cite Hinton and Salakhutdinov (2006) on page 2 in the introduction of the original manuscript:

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.

To put the reference of Hinton and Salakhutdinov (2006) more prominently in the manuscript we changed on page 7, line 7 of the revised manuscript:

The main building blocks of the neural network (Table 2) are convolutional layers (LeCun et al., 1998; Krizhevsky et al., 2012).

was changed to:

The overall structure of the neural network (Table 2) is a convolutional autoencoder (Hinton and Salakhutdinov, 2006; Ronneberger et al., 2015). Its main building blocks are convolutional layers (LeCun et al., 1998; Krizhevsky et al., 2012).

L9: "different number of filters" - not filter sizes. Your filter size is always 3x3 as you state-of-the-art just below.

Thank you for pointing this out. We corrected this and changed it throughout the manuscript.

L16: composed of

Thank you, we corrected this (and we found a similar error which is corrected too).

Page 8: Please also write down the loss function.

The cost function is "Equation 9" (page 9) from the original manuscript:

[...] The cost function has finally the following form:

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

We do not make a distinction between "cost function" and "loss function" and use it as synonyms as in Goodfellow et al. (2016):

The function we want to minimize or maximize is called the objective function, or criterion. When we are minimizing it, we may also call it the cost function, loss function, or error function. (page 80, of Goodfellow et al. (2016))

The loss function per individual scalar sample is the term in brackets of equations 9. This information has been added to the manuscript.

Page 9: L2: So here it appears that indeed your "independent" validation data are not independent.

This is line 2, page 9 for the original manuscript:

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.

It is unclear to us why the reviewer thinks that the data is not independent. The data marked for cross-validation is not used during the training. It is just a coincidence that the mini-batch size is equal to the number of images used for cross-validation. These numbers are not related.

L9: I don't understand the random masking: is one random mask applied to each image and then the same image used in each epoch? Or do you apply different random masks to the same image as a means to augment your data and increase generalization?

The later is the case, the paragraph has been revised to make this clear:

For every input image, more data points were masked (in addition to the cross-validation) by using a randomly chosen cloud 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 us to assess the capability of the network to recover missing data under clouds. Without the additional clouds, the neural network would simply learn to reproduce the SST values that are already received as input. At every epoch a different mask is applied to a given image to mitigate overfitting and aid generalization.

L24: Remove the statement about other variables, because you focus exclusively on SST. This can go in the conclusions section, but is confusing here.

OK, done.

Page 10: L10: I am confused here as to how the reconstruction and the training works. Normally, you first train your network and then you reconstruct (in particular on unseen data). Then you cannot take any average between epochs 200 and 1000.

The revised paragraph now reads:

The neural network is updated using the gradient for every mini-batch during training and after every 10 epochs the current state of the neural network is used to infer the missing data over the whole time series, and in particular reconstructing the missing data is the cross-validation dataset. But importantly, the network is not updated using the cross-validation data.

So effectively, we temporarily suspend the training after every 10 epochs and reconstruct the missing data but then continue the training.

Page 13: L5: "[..] 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." This sentence deserves further explanation. What is the generally accepted accuracy of the error estimate?

In the revised manuscript we avoided the term "acceptable" and added the following information:

An interpolation technique which is commonly used in operational context, is optimal interpolation. This technique is able to provide an expected error variance of the interpolated fields based on a series of assumptions, in particular that the errors are Gaussian distributed with a known covariance and zero mean. Given these assumptions, the error variance of the optimal interpolation algorithm is only found to be weakly related to the observed RMSE in a study of Pisano et al. (2016) using satellite sea-surface temperature in the Mediterranean Sea. In this context, the fact that DINCAE underestimates the actual error only by 15% on average can be seen as an improvement.

Page 14: L4: again: not "filter sizes" but "number of filters"

Thank you, this is corrected. We corrected all 5 occurrences of this issue.

Page 15: Figure 5: the figure titles are misleading. Apparently you are always showing results for one specific day. This day should then be mentioned in the caption and nt as title on the first panel. The way the panels are labelled now suggests that you compare apples with oranges (which I don't believe you do). Also here and in Figure 6 it is not quite clear to me if the DINEOF reconstruction also had to deal with the "added clouds" or not. Higher up, when you mention the addition of random clouds it would be good to see a typical fraction of image size which is obstructed by these random clouds. From figures 5 and 6, this obstruction seems to be quite large.

We removed the dates from the first panel. It was already mentioned in the caption and the reviewer is right that the date is common to all panels of figure 5.

Initially, the averaged cloud coverage of the dataset is 46% (over all 25 years). The cloud coverage for the 50 last scenes is increased to 77% when the cross-validation points are excluded. It is true that a significant part of the scene is obscured (after marking the data for cross-validation), but in the Mediteranan Sea the cloud coverage is relatively low compared to the globally average cloud coverage which is 75% (Wylie et al., 2005). Removing some data for cross-validation makes the cloud coverage thus more similar to the global average.

Page 17: L4: where can the reader see the comparison between in-situ obs and the reconstructions? This paragraph remains qualitative and doesn't contribute anything meaningful.

The RMS errors can be seen in table 3, but the reference in the original manuscript was unfortunately not clear:

> **As expected, biases play now a more important role when comparing in situ observations with reconstructed satellite data (3).**

"(3)" has been changed to "(Table 3)". We apologize for this issue which could easily cause a reader to overlook this table.

L15: indeed - if the deep learning method was applied correctly (which is somewhat difficult to judge from this paper due to all the issues described in this review), then this is a very nice and important result, which shows the superiority of deep learning ap- proaches with their ability to learn non-linear functions compared to standard statistical methods. This key result could probably be brought out even clearer.

Thank you for your encouragement! We hope that the revised manuscript based on the comments of all reviewers is now clearer.

Page 18: L2: why is this method "practical"? I assume it is, but this is only because of my background knowledge. This point needs to be made explicit somewhere in the paper. You list computation times for training, but you don't say how long it takes to reconstruct a scene once the network has been trained.

Reconstructing the data of all 25 years takes only 8 seconds on the GeForce GTX 1080 for a trained network, but training the network can take several hours as mentioned in the manuscript. The manuscript has been updated with the reconstruction time for a trained network.

In the revised manuscript we removed the term "practical" because it was not possible for us to give it a precise meaning.

Page 19: After rewriting other parts of the manuscript I suggest to re-read the final part of the conclusions to see if the paper ends with the highest impact message.

We agree that the ending of the conclusion was quite dull in the original manuscript. We revised the conclusions accordingly:

> **The tests conducted in this paper show that DINCAE is able to provide a good reconstruction of missing data in satellite SST observations and retaining more variability than the DINEOF method. In addition, the expected error variance of the reconstruction is estimated avoiding several assumptions (difficult to justify in practice) of other methods like optimal interpolation.**

References

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.

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.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, CoRR, abs/1406.1078, http://arxiv.org/abs/1406.1078, 2014.

Ge, R., Huang, F., Jin, C., and Yuan, Y.: Escaping From Saddle Points - Online Stochastic Gradient for Tensor Decomposition, CoRR, abs/1503.02101, http://arxiv.org/abs/1503.02101, 2015.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, MIT Press, http://www.deeplearningbook.org, 2016.

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.

Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory, Neural computation, 9, 1735–80, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

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, 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, 86, 2278–2324, 1998.

Masters, D. and Luschi, C.: Revisiting Small Batch Training for Deep Neural Networks, CoRR, abs/1804.07612, http://arxiv.org/abs/1804.07612, 2018.

Pisano, A., Nardelli, B. B., Tronconi, C., and Santoleri, R.: The new Mediterranean optimally interpolated pathfinder AVHRR SST Dataset (1982-2012), Remote Sensing of Environment, 176, 107 – 116, https://doi.org/10.1016/j.rse.2016.01.019, 2016.

Prechelt, L.: Early Stopping — But When?, pp. 53–67, Springer Berlin Heidelberg, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-35289-8_5, 2012.

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. 234–241, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review, 65, 386–408, https://doi.org/10.1037/h0042519, 1958.

Schmidhuber, J.: Deep learning in neural networks: An overview, Neural Networks, 61, 85 – 117, https://doi.org/10.1016/j.neunet.2014.09.003, 2015.

Widrow, B. and Hoff, M. E.: Associative Storage and Retrieval of Digital Information in Networks of Adaptive "Neurons", pp. 160–160, Springer US, Boston, MA, https://doi.org/10.1007/978-1-4684-1716-6_25, 1962.

Wilks, D. S.: Statistical Methods in the Atmospheric Sciences, Academic Press, 1995.

Wylie, D., Jackson, D. L., Menzel, W. P., and Bates, J. J.: Trends in Global Cloud Cover in Two Decades of HIRS Observations, Journal of Climate, 18, 3021–3031, https://doi.org/10.1175/JCLI3461.1, 2005.

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~~ sea surface temperature 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 cloud-free pixels in satellite images. ~~However, it is unclear how~~ Contrary to standard image reconstruction with neural networks, this application requires a method 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 the likelihood of the observed value. The approach, called DINCAE (Data-Interpolating Convolutional Auto-Encoder) is applied to a ~~relatively long~~ 25-year time-series of Advanced Very High Resolution Radiometer (AVHRR) sea surface temperature data and compared to DINEOF (Data Interpolating Empirical Orthogonal Functions), a commonly used 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

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~~ the ocean velocity variability depends partially on ocean temperature. The amount of heat stored in the ocean is also critical for weather predictions at various scales (~~like~~ e.g. hurricane path prediction in the short range, as well as for seasonal and climate predictions).

The ocean sea surface temperature (SST) ~~is~~ has been routinely measured since the beginning of the 1980s. However, as for any ~~sensor~~ measuring technique 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 fil-

tered outbecause of the transient and stochastic nature of these structures. DINEOF (Data Interpolating Empirical Orthogonal Functions),provides an accurate way of retrieving missing data and .

DINEOF (Data Interpolating Empirical Orthogonal Functions, Beckers and Rixen, 2003; Alvera-Azcárate et al., 2005), is an iterative method to reconstruct missing observations 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)empirical orthogonal functions (EOF). A truncated EOF decomposition using the leading EOFs is performed and the initially missing data are reconstructed using this EOF decomposition. The EOF decomposition and reconstruction is repeated until convergence. 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 scale and transient features can also be removed from the analysis, resulting in a smooth reconstructiondecomposition will focus primarily in spatial structures with a "strong" signature (or more formally defined with a significant L2 norm compared to the total variance). Small scale structures can be included in a truncated EOF decomposition as long as their related variance is large enough to be present in the retained EOF modes. But small scale structures tend to be transient (short-lived) and therefore are often not retained in the dominant EOF modes. It should be noted that there is no explicit spatial filtering scale in DINEOF removing small scales (unlike other methods like optimal interpolation, kriging, spline interpolation). But in practice a similar smoothing effect is noticed because of the EOF truncation (which is necessary in the presence of clouds).

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 learn nonlinear, stochastic features measured at the sea surface by satellite sensors, and their use might prove efficient in retaining these structures when analysing satellite data, for example for reconstructing missing data. However,

Neural networks can be composed of a wide variety of building blocks, such as fully connected layers (Rosenblatt, 1958; Widrow and Ho recurrent networks (e.g. Long Short-term Memory (Hochreiter and Schmidhuber, 1997), Gated recurrent units (Cho et al., 2014)) convolutional layers (LeCun et al., 1998; Krizhevsky et al., 2012). Recurrent networks work typically with a one dimensional list of inputs of a variable length (such as a text sentence). Fully connected layers and convolutional layers require to have a full dataset without missing data, at least for the training phase. For a review on neural networks the reader is referred to Schmidhuber (2015) and references therein. As 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-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)~~ 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 (i.e. emulating a sensor failure). ~~Jo et al (2018)~~ The neural network by Krasnopolsky et al. (2016) uses as input satellite sea surface elevation, sea surface salinity, sea surface temperature and *in situ* Argo salinity and temperature vertical profiles with some auxiliary information (like longitude, latitude and time) to estimate the Chlorophyll-a concentration. The network does not use measured Chlorophyll-a concentration at a given location as input during inference (the reconstruction phase), nor the information from nearby grid points to infer Chlorophyll-a concentration. The network is exposed to the chlorophyll-a measurements only during the training phase. 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)~~ Renosh et al. (2017) 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)~~ 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)~~ Jouini et al. (2013) reconstructed missing data in ~~chlorophyll maps~~ chlorophyll-a data using 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 to in the following as DINCAE (data-interpolating convolutional auto-encoder). 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. Projecting the input data on a low-dimensional subspace is also the central idea of DINEOF, where it is achieved by an EOF decomposition.

In section 2, the SST dataset used in this study is presented. This dataset is ~~first reconstructed with DINEOF (section 4). The~~ the input of the neural network described in section 3. This section includes the general structure of the network~~and its cost-function able to provide an error estimate of the reconstruction are presented in section 3~~, the activation functions, skip connections, the cost function and its optimization. The SST dataset is also reconstructed with DINEOF (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 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 Current (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 composed ~~by~~ of 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 of 5266 time instances.

To assess the accuracy of the reconstruction method, cross-validation is used ~~as follows: in~~ (*e.g.* Wilks, 1995). For cross-validation a subset of the data is withheld from the analysis and the final reconstruction is compared to the withheld dataset to access its accuracy. Since clouds have a spatial extent, we wanted to withhold data with a similar spatial structure. In the last 50 images we removed data according to the cloud mask of the first 50 images of the SST time series. The last 50 images represent the data from 2009-09-25 to 2009-12-27 (since some scenes with too few data have been dropped as mentioned before). 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 (i.e. individual pixels) have been withheld this way.

~~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 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 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~~ Initially, the average cloud coverage of the dataset is 46% (over all 25 years). The cloud coverage for 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 2). As only 13 modes are retrained by DINEOF for the reconstruction, 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~~ last scenes is increased to 77% when the cross-validation ~~data) is lower (0.3375^{o}C) than when the 25 years dataset is used. However, the RMS error~~ points are excluded. A significant part of the scene is obscured after marking the data for cross-validation, but in the Mediteranan Sea the cloud coverage is relatively low compared to the globally average cloud coverage which is 75% (Wylie et al., 2005). Removing some data for 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~~makes the cloud coverage thus more similar to the global average.

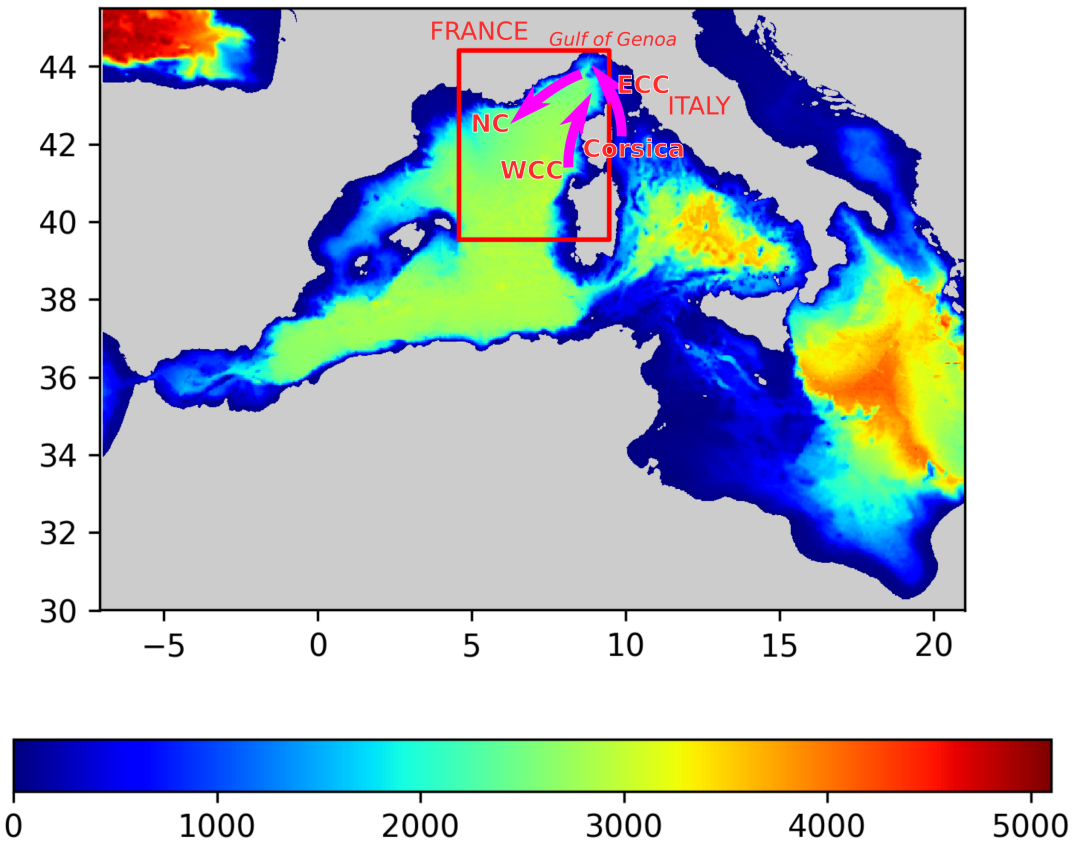

Figure 1. ~~Comparison with~~ The red rectangle delimits the ~~independent cross-validation data~~ studied region and the ~~dependent data used for training (~~color represents the bathymetry in ~~°C~~meters. The arrows represent the main currents: the Western Corsican Current (WCC), the Eastern Corsican Current (ECC) and the Northern Current (NC)

3 Neural network with missing data as input

Convolutional and other deep neural networks are extensively used in computer vision and find an increasing number of applications in Earth sciences (Rasp et al., 2018; 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~~ data assimilation in numerical ocean models. The standard optimal interpolation equations (*e.g.* Bretherton et al., 1976; Buongiorno Nardelli, 2012) can be written as follows:

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

$$\mathbf{P}^{a\,-1} = \mathbf{P}^{f\,-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \quad (2)$$

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~~errors varying in space and/or time). The main input datasets of the CAE are i) the SST divided by its error variance (corresponding to $\mathbf{R}^{-1}\mathbf{y}^o$) and ii) the inverse of the error variance (corresponding to the diagonal elements of $\mathbf{R}^{-1}$, assuming spatially uncorrelated errors). 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 (the scaled anomaly is 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~~ thus represented by an array of the size ~~8~~ 10 x 112 x 112 x ~~5266.~~ 5266 (number of inputs, number of grid points in the zonal direction, number of grid points in the meridional direction, number of time instances). 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. Using the previous and next day as inputs and the information on the season (last two inputs) will allow for a temporal coherency of the results. It should be noted that DINEOF does not use the day of the year of each satellite image, but it uses a temporal filter which increases the temporal coherence of the reconstruction (Alvera-Azcárate et al., 2009). The 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~~ overall structure of the neural network (Table ~~1)~~ 2) is a convolutional autoencoder (CAE; Hinton and Salakhutd . Its main building blocks 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~~filters. Beside the input and output ~~layer, the filter sizes~~ layers, the number of filters are 16, 24, 36 and 54 (the ~~filter sizes increase~~ number of filters increases 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 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~~ of convolutional layers and interpolation layers (to the 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 1. The motivation of this choice is that ~~large-scale~~ large scale information of the SST would be captured by the neurons in the bottle-neck, but ~~small-scale~~ small scale 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 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 1, is an array $T^{(26)}_{ijk}$ 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 the second slice is the temperature anomaly divided by the error variance. The reconstructed temperature anomaly $\hat{y}_{ij}$ and the corresponding error variance $\hat{\sigma}^2_{ij}$ (for every single grid point i, j) are computed as:

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

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

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. The effective range of the error standard deviation is thus from $\exp(-\gamma/2) = 0.0067\,°\mathrm{C}$ to $\delta^{-\frac{1}{2}} = 31.6\,°C$ which is a relatively wide range as the error is expected to be O(0.1) to O(1) ^{o}C. The bounds are only effective during the very first epochs of the neural network where the weights are still close to random values.

3.1 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~~ 10 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 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~~ also introduces some stochasticity which prevents the minimization algorithm ~~to be~~ from being 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 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 us to assess the capability of the network to recover missing data under clouds. Without the additional clouds, the neural network would simply learn to reproduce the SST values that are already received as

Table 1. 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)

input. At every epoch a different mask is applied to a given image to mitigate overfitting and aid generalization.

The aim of DINCAE is to provide a good SST reconstruction but also an assessment of the accuracy of the reconstruction. The output of the neural network ~~(for every single grid point i, j) is~~ is assumed to be a Gaussian probability distribution function (pdf) 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:

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

where N is the number of measurements in y_{ij} (excluding ~~thus~~ therefore land points and cross-validation points). Including the number measurements N is important as ~~the number~~ it 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 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~~ loss function per individual scalar sample is the term in brackets of the previous equation. The first term 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 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 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^{o}C.

It is useful to compare the proposed approach to the traditional autoencoder to highlight the different choices that have been adopted. The essential steps to implement and validate an auto-encoder are the following:

- Some data are marked for validation and never used during training
- The network is given some data as input and produces an output which should be as close as possible to the input. All training data are given thus at all epochs to the network

– The network is validated using the validation data set aside.

In essence, the traditional auto-enoder optimises how well the provided input data can be recovered after dimensionality reduction. In the present approach, there are two steps where data are intentionally hidden to the network:

1. The validation data that were set aside and never used during the training, similar to the traditional auto-encoder.
2. Some additional data in every minibatch were set aside to compute the reconstruction error and its gradient (unlike the traditional auto-encoder). This additional subset is chosen at random.

This is done because the main purpose of the network is to assess the ability of the network to reconstruct the missing data using the available data. The proposed method is not withholding less data than the traditional auto-encoder. The downside of the approach is that the cost function fluctuates more because it is computed only over a relatively smaller set of data. But for us this is acceptable (and controlled by taking the average of the output of the network at several epochs, as explained later) because the cost function reflects more closely the objective: reconstructing missing data from the available data (instead of reproducing the input data as it is the case of the traditional auto-encoder).

The traditional auto-encoder approach trained using only clear images was not considered because only 13 images of out 5266 have a cloud coverage of less than 5%. So the ability to handle missing data was a requirement for us from the start.

4 DINEOF reconstruction

The results of the DINCAE method are compared to the reconstruction obtained by the DINEOF method (Alvera-Azcárate et al., 2005) which uses an EOF-basis to infer the missing data. As a first step, the spatial and temporal mean is removed from the data, and the missing data are set to zero. The leading EOF modes are then computed and the missing data are reconstructed using these EOFs (Alvera-Azcárate et al., 2005). A temporal low-pass filter with a cut-off period of 1.08 days is applied to improve the 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 the satellite data from the previous and next days are used in the reconstruction. The optimal 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 2). As only 13 modes are retrained by DINEOF for the reconstruction, some small scale structures are smoothed-out, which is a well known property of a truncated EOF decomposition (Wilks, 1995) . This smoothing effect results in an RMS (root mean square) 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 with DINEOF (only from the last two years, *i.e.* 2008 to 2009), 19 EOFs modes are retained, leading to a reconstruction with

Table 2. Comparison with the independent cross-validation data and the dependent data used for training (in ^{o}C). CRMS is the centered root mean square error.

	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 5 conv. layers)	0.4603	0.4557	-0.0648	0.1396	0.1364	-0.0295
DINCAE (all skip connections and 3 conv. 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

richer structures. Therefore, the RMS error compared to all the initially present SST provided to DINEOF (but excluding the cross-validation 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. DINEOF used 15 hours of a single core on an Intel Core i7-3930K CPU to reconstruct the 25 years.

5 Results

The figure 2 shows the cost function for every minibatch. Large fluctuations are quite apparent from this figure. But it is expected that the cost function will fluctuate using any optimization method based on mini-batch (unless the learning rate explicitly is decreased to zero, which is not the case here) because the cost function is evaluated using a different mini-batch at every iteration. Consequently, the gradient of the cost function also includes some stochastic variability. Even if the dataset is small and the gradient could be computed over the entire dataset at once, using mini-batches is still advised because these fluctuations allow the cost function to get out of a local minima (Ge et al., 2015; Masters and Luschi, 2018). While the mini-batch selection effectively computes the gradient over a temporal subset, the additional data marked as missing within a minibatch is a spatial subset which enhances these fluctuations but allows us to define the cost function more closely to our objective (i.e. inferring the missing data from observations, as explained above).

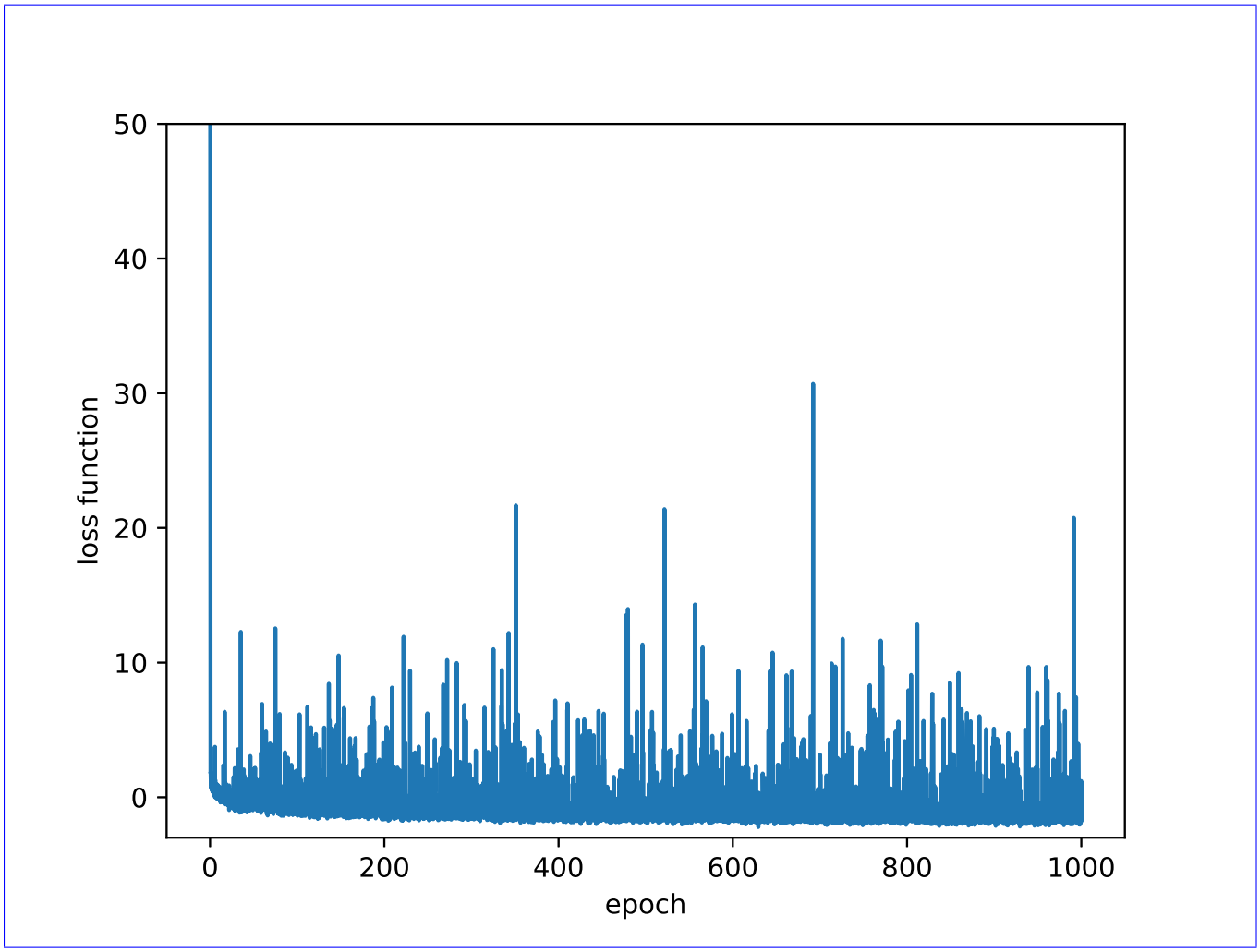

Figure 2. The cost function computed internally for every minibatch during the optimization.

The neural network is updated using the gradient for every mini-batch during training and after every 10 epochs the current state of the neural network is used to infer the missing data over the whole time series, and in particular reconstructing the missing data is the cross-validation dataset. But importantly, the network is not updated using the cross-validation data.

Figure 3 shows the RMS error relative to the cross-validation dataset computed every ten epochs (during this reconstruction phase drop-out is disabled) using DINCAE. 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 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~~ could result in overesti-
5 mating the expected error of the reconstruction. Alternatively one would average the output of an ensemble of neural networks initialized with different weights (and possibly using different structures) but this would significantly increase the necessary computing resources of the technique (Krizhevsky et al., 2012). But this ensemble averaging approach could be beneficial to improve the representation of the expected error and the accuracy of the reconstruction.

10 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.

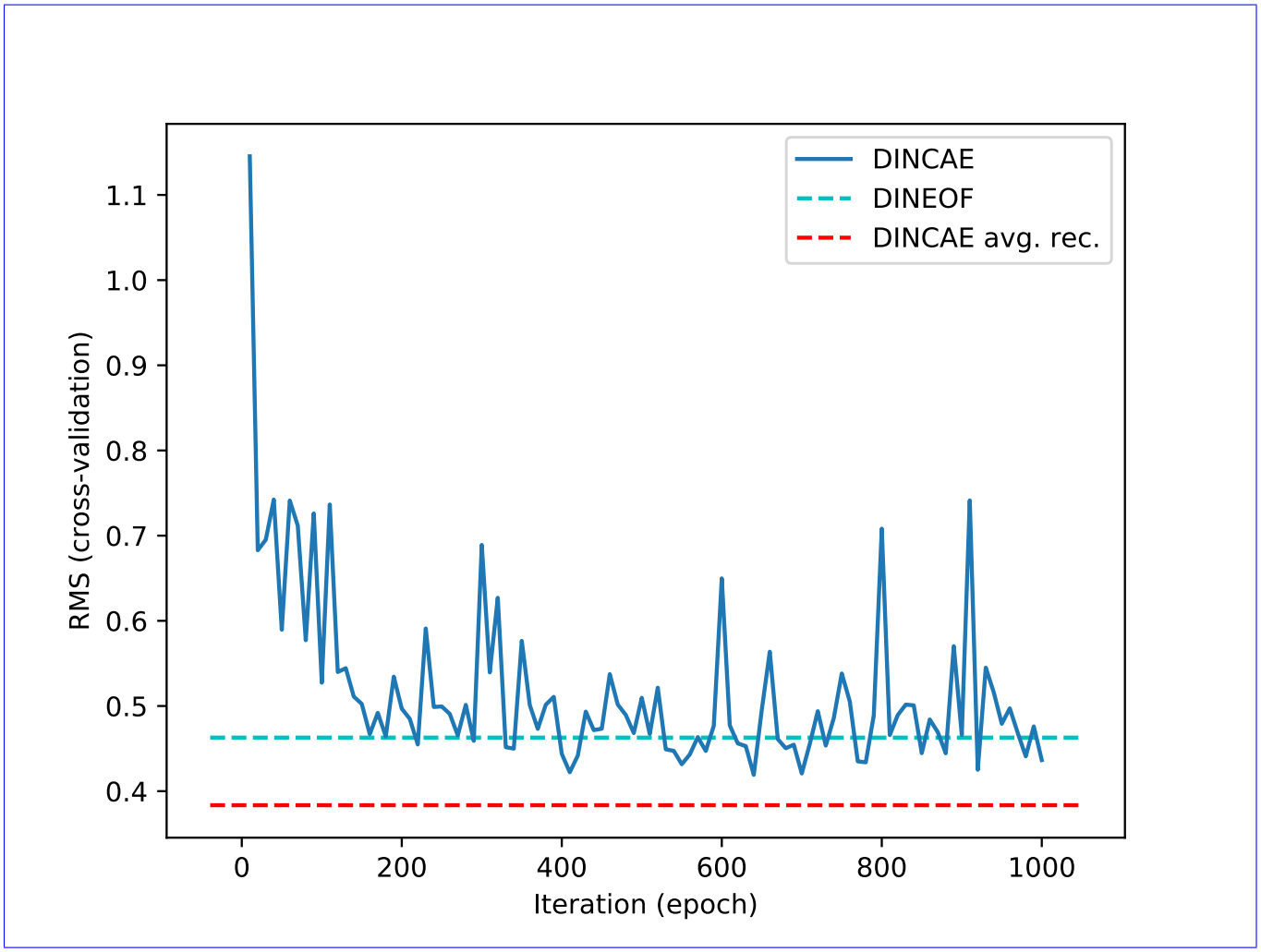

Figure 3. RMS difference with cross-validation dataset as a function of iteration. The solid blue line ~~represent~~ represents the DINCAE reconstruction at different steps of the iterative minimization algorithm. The dashed cyan line is the DINEOF reconstruction and the dashed red line is the average DINCAE reconstruction between epoch 200 and 1000.

Training this network for 1000 epochs takes ~~32~~ 4.5 hours on a GeForce GTX 1080 and Intel Core i7-7700 with the neural network library tensorflow (Abadi et al., 2015). For a trained network, reconstructing all 25-year takes only 8 seconds. All computations are done in single precision.

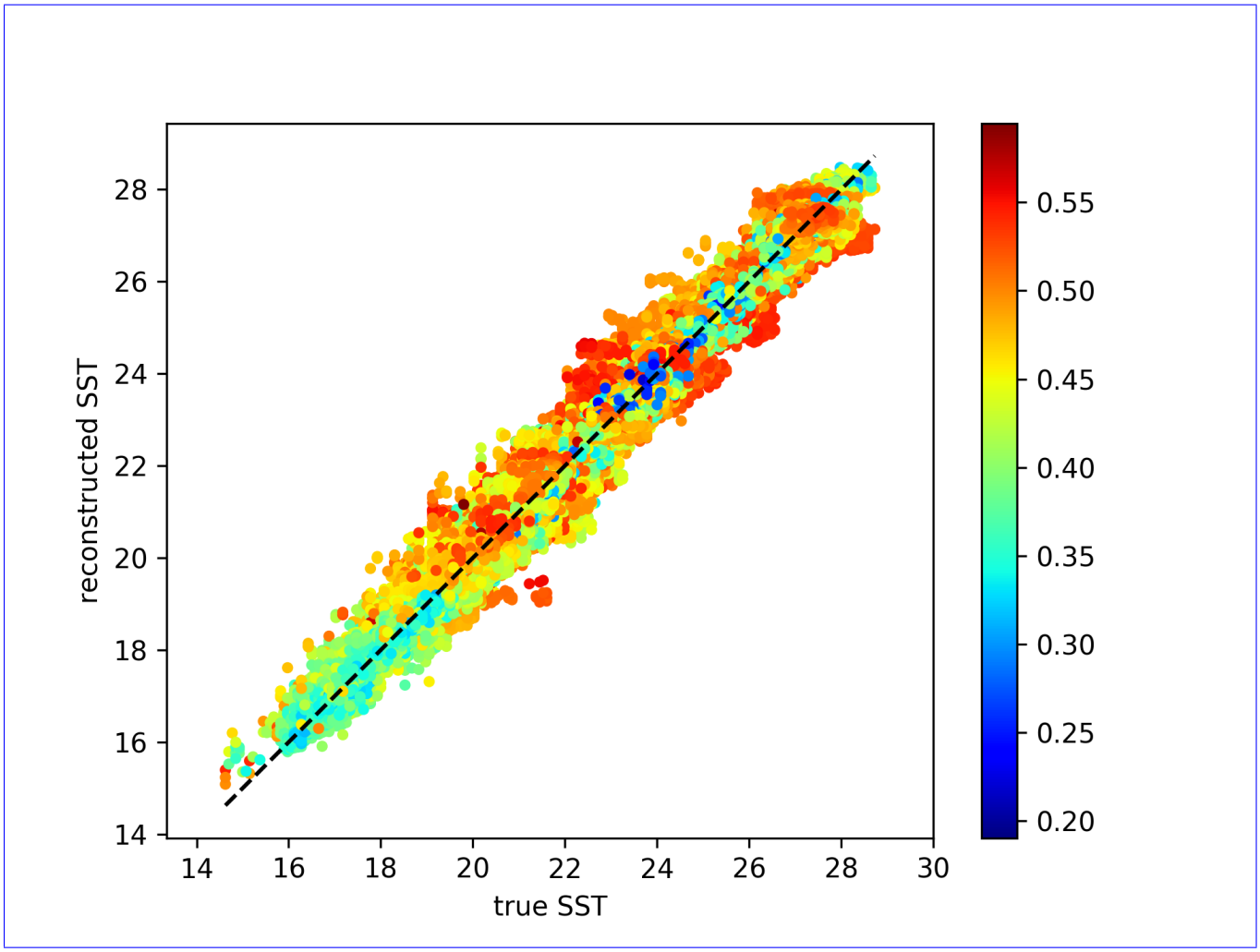

Figure 4. ~~RMS difference with~~ The original SST versus the reconstructed SST for the cross-validation dataset~~as a function of iteration~~. The color represents the estimated expected error standard deviation.

Figure 4 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.

5

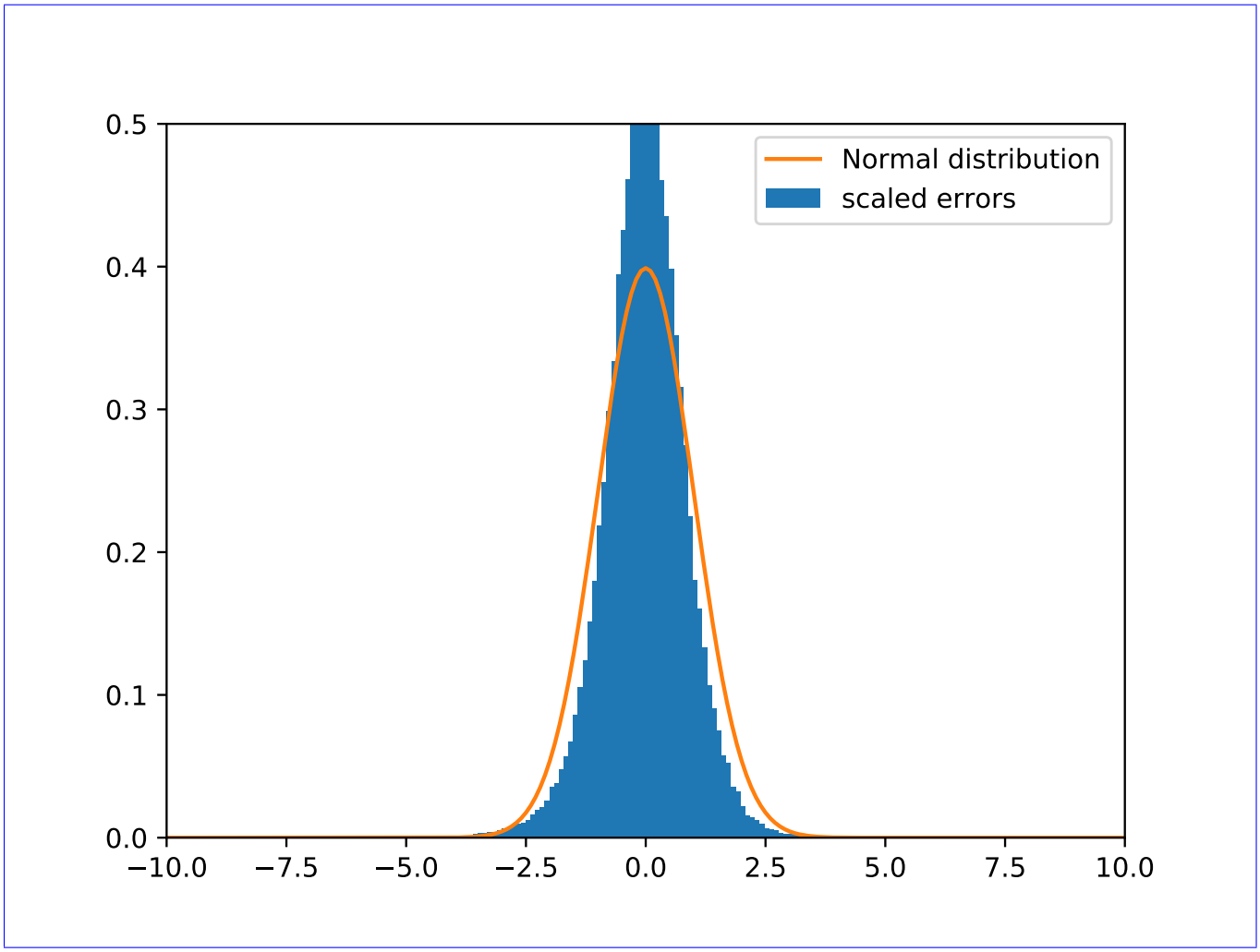

Figure 5. Scaled ~~error~~ errors are computed as the difference between the reconstructed SST and the actual measured SST (withheld during cross-validation) divided by the expected standard deviation error.

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 5. 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), so that generally speaking DINCAE is overestimating the expected error by 15 %.

An interpolation technique which is commonly used in operational context, is optimal interpolation. This technique is able to provide an expected error variance of the interpolated fields based on a series of assumptions, in particular that the errors are Gaussian distributed with a known covariance and zero mean. Given these assumptions, the error variance of the optimal

interpolation algorithm is only found to be weakly related to the observed RMSE in a study of Pisano et al. (2016) using satellite sea surface temperature in the Mediterranean Sea. The averaged results of DINCAE ~~underestimate~~ overestimate 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~~which in this context can be seen as an improvement.

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 ^{o}C (no skip connections), to 0.4222 ^{o}C with 2 skips connections (layer 22 and 25 of Table 1), and further to 0.3900 ^{o}C with all skip connections between the encoder and decoder layer of the same size. At the same time, the ~~degree of smoothing of~~ RMS error relative to the used data (i.e. data not reserved for cross-validation) measuring the degree of smoothing is reduced from 0.2957 ^{o}C (no skip connections) to 0.1383 ^{o}C with all skip connections.

Increasing the ~~filter sizes~~ number of filters of the convolutional layers from 16, 24, 36, 54 to 16, 32, 48, 64 (with the input convolution layer fixed by ~~8~~ 10 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 (5 or 3 convolutional layers) did not improve the results. However, it is 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~~ commonly used 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 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. This better performance of average pooling can be related to the fact that SST images do generally not have as abrupt gradients as typical images used for classification. Another way to look at this is the fact that for a dynamical system in the linear regime, different flow features (solutions to the underlying primitive equations) coexist and contribute in an additive way to the total flow.

For every time instance we use the data from 3 time instances in the reconstruction: the current day, as well the data from the previous and next day. As a variant of the previous reconstruction experiment we increase the number of time instances from 3 to 5 centered at the current time instance. However, the cross-validation error for this experiment is 0.433 ^{o}C and the results are not improved. Increasing the number of input features can aggravate the potential for overfitting as the number of parameters in the neural network is increased. Here the number of parameters is increased by 40% in the first convolutional

layer. A combination of convolutional neural network with recurrent neural networks (like Long Short-Term Memory, LSTM) might be a better way to include the time dependencies.

In all cases the biases are relatively small and the present discussion is essentially also valid when considering the ~~centred~~ centered RMS (i.e. the RMS difference when the bias is removed). In the following, we only use DINCAE with all skip connections and 4 convolutional layers with ~~filter sizes~~ a number of filters of 16, 24, 36, 54 and average pooling for future comparison.

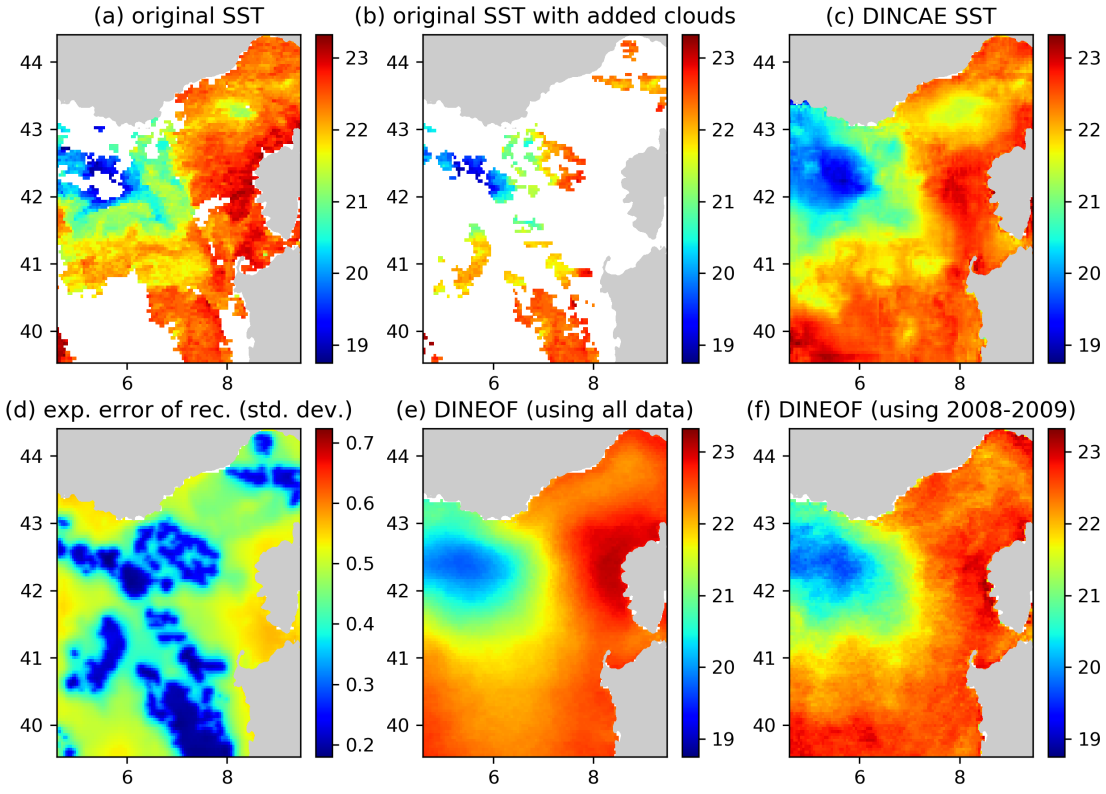

Figure 6. ~~Example reconstruction~~ Panel (a) the original AVHRR SST, (b) AVHRR SST with additional clouds for cross-validation, (c) the DINCAE ~~and~~ reconstruction, (d) the expected error variance of the DINCAE reconstruction, (e) the DINEOF reconstruction using all data, (f) the DINEOF reconstruction using only the data from 2008-2009. All panels are in degrees Celsius and valid for the date 2009-10-13.

Figure 6 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^{o}C isotherm agrees better with the SST observations in the DINCAE reconstruction than ~~in~~ with the DINEOF results.

Table 3. Comparison with the World Ocean Database for SST grid points covered by clouds. The RMS, CRMS and bias are in degree Celsius.

	RMS	CRMS	bias
DINEOF	1.1676	1.1102	-0.3616
DINCAE	1.1362	1.0879	-0.3278

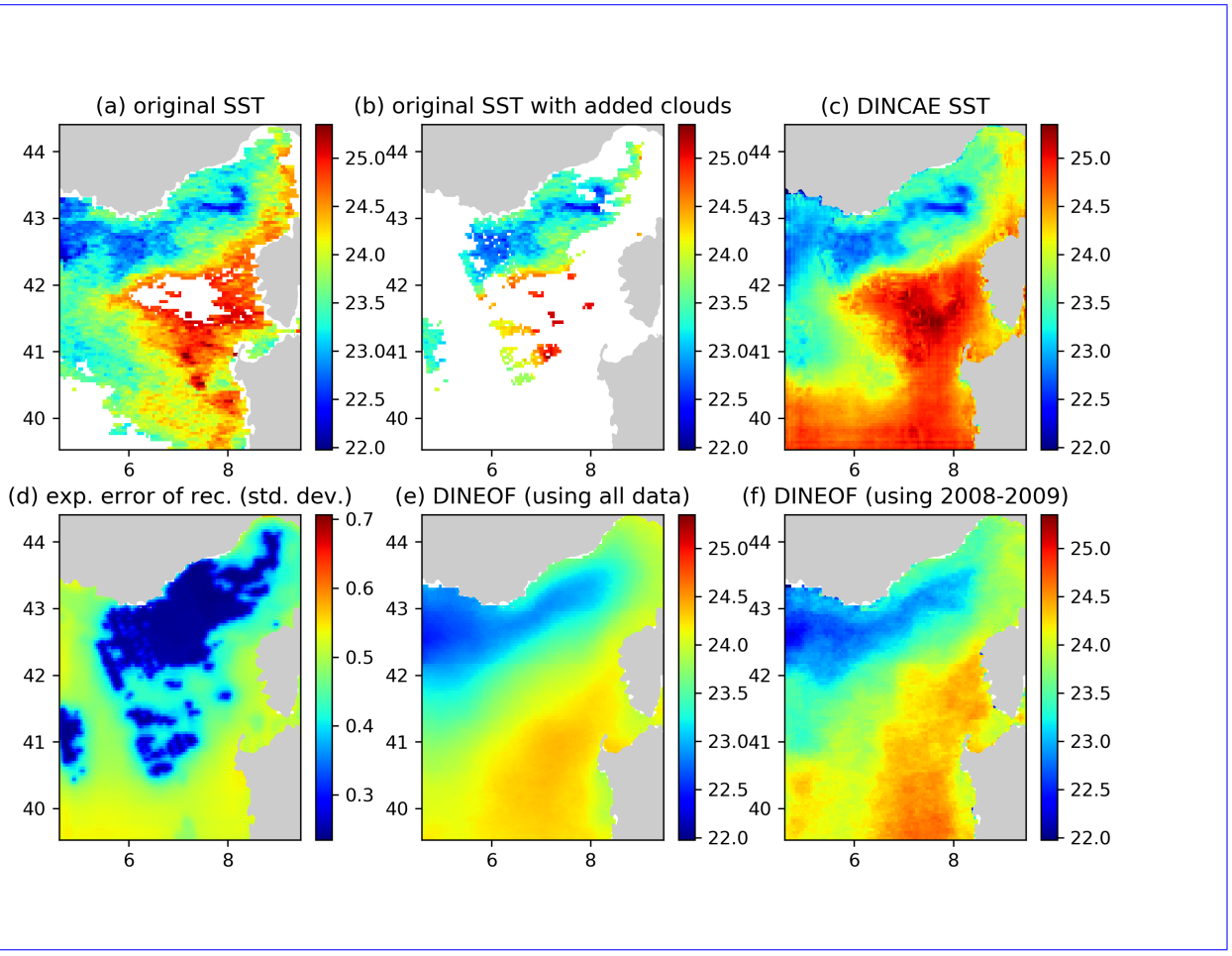

Figure 7. ~~Example reconstruction~~ Panel (a) the original AVHRR SST, (b) AVHRR SST with ~~some artefacts~~ additional clouds for cross-validation, (c) the DINCAE reconstruction, (d) the expected error variance of the DINCAE reconstruction, (e) the DINEOF reconstruction using all data, (f) the DINEOF reconstruction using only the data from 2008-2009. All panels are in degrees Celsius and valid for the date 2009-09-29.

In some cases the DINCAE reconstruction ~~introduces also~~ also introduces some artefacts as some zonal and meridional gradients near the open boundaries (Figure 7). ~~In~~ This is probably due to the fact that 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.

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~~observations. Only in situ observations corresponding to a cloudy SST pixel are used in the following. In total, there are 774 surface in situ observations. 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 now when comparing in situ observations with reconstructed satellite data (Table 3). DINCAE represented a small improvement relative to the DINEOF reconstruction confirming the results from the cross-validation comparison.

In Figure 8 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. 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~~ it is thus remarkable 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.

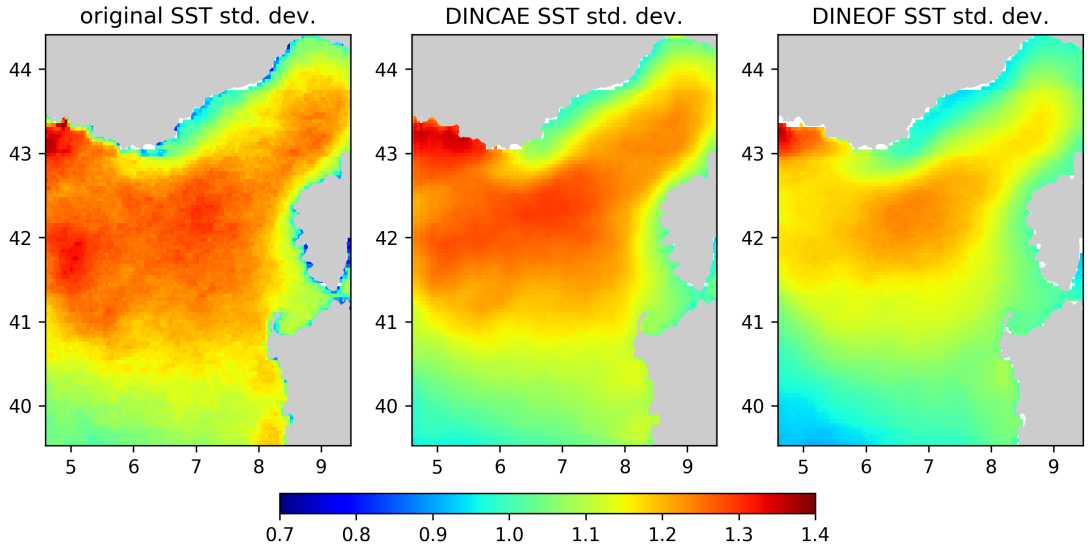

Figure 8. Standard deviation computed around the seasonal average in degrees Celsius.

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 reconstruction 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~~ chlorophyll-a concentration or suspended sediment concentration.

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~~ of 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~~0.46^{o}C to ~~0.3835~~0.38^{o}C.

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 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 variability of the DINCAE reconstruction matched the variability of the original data relatively well.

The tests conducted in this paper show that DINCAE is able to provide a good reconstruction of missing data in satellite SST observations and retaining more variability than the DINEOF method. In addition, the expected error variance of the reconstruction is estimated avoiding several assumptions (difficult to justify in practice) of other methods like optimal interpolation.

Code availability. The source code is released as open source under the terms of the GNU General Public Licence v3 (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.

Competing interests. The authors have no competing interests

Acknowledgements. We thank the anonymous reviewer #1, #2 and Zhaohui Han for reading carefully the manuscript, providing constructive remarks and interesting interpretations of the results.

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 III 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 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~~ was 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., Goodfellow, 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., 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.

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/5/475/2009/, 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 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.

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

Chapman, C. and Charantonis, A. A.: Reconstruction of Subsurface Velocities From Satellite Observations Using Iterative Self-Organizing Maps, IEEE Geoscience and Remote Sensing Letters, 14, 617–620, https://doi.org/10.1109/LGRS.2017.2665603, 2017.

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.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, CoRR, abs/1406.1078, http://arxiv.org/abs/1406.1078, 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 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.

Ge, R., Huang, F., Jin, C., and Yuan, Y.: Escaping From Saddle Points - Online Stochastic Gradient for Tensor Decomposition, CoRR, abs/1503.02101, http://arxiv.org/abs/1503.02101, 2015.

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.

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.

Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory, Neural computation, 9, 1735–80, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Jo, Y.-H., Kim, D.-W., and Kim, H.: Chlorophyll concentration derived from microwave remote sensing measurements using artificial neural network algorithm, Journal of Marine Science and Technology, 26, 102–110, https://doi.org/10.6119/JMST.2018.02_(1).0004, 2018.

Jouini, M., Lévy, M., Crépon, M., and Thiria, S.: Reconstruction of satellite chlorophyll images under heavy cloud coverage using a neural classification method, Remote Sensing of Environment, 131, 232 – 246, https://doi.org/10.1016/j.rse.2012.11.025, 2013.

Kilpatrick, K. A., Podestá, G. P., and Evans, R.: Overview of the NOAA/NASA advanced very high resolution radiometer Pathfinder 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, 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, 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.

Masters, D. and Luschi, C.: Revisiting Small Batch Training for Deep Neural Networks, CoRR, abs/1804.07612, http://arxiv.org/abs/1804.07612, 2018.

Patil, K. and Deo, M. C.: Prediction of daily sea surface temperature using efficient neural networks, Ocean Dynamics, 67, 357–368, https://doi.org/10.1007/s10236-017-1032-9, 2017.

Pisano, A., Nardelli, B. B., Tronconi, C., and Santoleri, R.: The new Mediterranean optimally interpolated pathfinder AVHRR SST Dataset (1982-2012), Remote Sensing of Environment, 176, 107 – 116, https://doi.org/10.1016/j.rse.2016.01.019, 2016.

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.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.

Renosh, P. R., Jourdin, F., Charantonis, A. A., Yala, K., Rivier, A., Badran, F., Thiria, S., Guillou, N., Leckler, F., Gohin, F., and Garlan, T.: Construction of Multi-Year Time-Series Profiles of Suspended Particulate Inorganic Matter Concentrations Using Machine Learning Approach, Remote Sensing, 9, https://doi.org/10.3390/rs9121320, 2017.

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. 234–241, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-319-24574-4_28, 2015.

Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review, 65, 386–408, https://doi.org/10.1037/h0042519, 1958.

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.

Schmidhuber, J.: Deep learning in neural networks: An overview, Neural Networks, 61, 85 – 117, https://doi.org/10.1016/j.neunet.2014.09.003, 2015.

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

Widrow, B. and Hoff, M. E.: Associative Storage and Retrieval of Digital Information in Networks of Adaptive "Neurons", pp. 160–160, Springer US, Boston, MA, https://doi.org/10.1007/978-1-4684-1716-6_25, 1962.

Wilks, D. S.: Statistical Methods in the Atmospheric Sciences, Academic Press, 1995.

Wylie, D., Jackson, D. L., Menzel, W. P., and Bates, J. J.: Trends in Global Cloud Cover in Two Decades of HIRS Observations, Journal of Climate, 18, 3021–3031, https://doi.org/10.1175/JCLI3461.1, 2005.

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.