Predicting Sea Surface Temperatures with Coupled Reservoir Computers

Benjamin Walleshauser[1,3] and Erik Bollt[2,3]

[1]Department of Physics and Department of Mechanical Engineering, Clarkson University
[2]Department of Electrical and Computer Engineering, Clarkson University
[3]Clarkson Center for Complex Systems Science

Correspondence: Benjamin Walleshauser (wallesbt@clarkson.edu)

Abstract. Sea surface temperature (SST) is a key factor in understanding the greater climate of the Earth and is an important variable when making weather predictions. Methods of machine learning have become ever more present and important in data-driven science and engineering including in important areas for Earth Science. We propose here an efficient framework that allows us to make global SST forecasts by use of a coupled reservoir computer method that we have specialized to this domain allowing for template regions that accommodate irregular coastlines. Reservoir computing is an especially good method for forecasting spatiotemporally complex dynamical systems, as it is a machine learning method that despite many randomly selected weights, is nonetheless highly accurate and easy to train. Our approach provides the benefit of a simple and computationally efficient model that is able to predict sea surface temperatures across the entire Earth's oceans. The results are demonstrated to generally follow the actual dynamics of the system over a forecasting period of several weeks.

1 Introduction

As most of Earth's surface is covered by water, global sea surface temperatures (SST) are an important parameter in understanding the greater climate of the Earth. Sea surface temperature is an important variable in the study of marine ecology (Gomez et al., 2020; Novi et al., 2021), weather prediction (Dado and Takahashi, 2017), and to help project future climate scenarios (Pastor, 2021). Yet the task of predicting changes in the SST is quite difficult, due to large variations in heat flux, radiation, and diurnal wind near the surface of the sea (Patil et al., 2016).

Given the importance of sea surface temperature to the fuller Earth weather and climate system, there is significant interest in forecasting this spatiotemporally complex process. Methods for predicting changes in the SST can be divided into two different categories: numerical methods and data-driven methods. Numerical methods are based on the underlying knowledge of the governing physics behind the system, and simulation thereof. These are widely used to predict SST over a large area (Stockdale et al., 2006; Krishnamurti et al., 2006). However, data-driven methods encompass statistical and machine learning approaches, and are widely used to predict the SST, often with little to no knowledge regarding relevant physical principles behind the dynamics of the system, hence reducing the complexity of the model. Several statistical methods that have been used include: Markov models (Xue and Leetmaa, 2000; Johnson et al., 2000), linear regression (Kug et al., 2004), and empir-

ical canonical correlation analysis (Collins et al., 2004). Meanwhile machine learning methods have included: support vector machines (SVMs) (Lins et al., 2013), long short term memory neural networks (LSTMs) (Zhang et al., 2017; Kim et al., 2020; Xiao et al., 2019a), memory graph convolutional networks (MGCNs) (Zhang et al., 2021), etc. In this paper, we utilize coupled reservoir computers (RCs), therefore taking advantage of the reduced complexity of data-driven methods, while still being able to predict temperatures globally due to the minimal training required by each RC. We have adapted the RC concept for spatiotemporal processes to allow for coupled local templates that accommodate the peculiarities associated with varying coastlines.

Reservoir computers have been shown to be excellent at predicting complex dynamical systems (Ghosh et al., 2021; Pandey and Schumacher, 2020) regardless of the relative simplicity of the approach. They have even been shown to be proficient in the prediction of spatiotemporally complex systems, such as the Kuramoto-Sivashinsky PDE (Vlachas et al., 2020) and cell segmentation (Hadaeghi et al., 2021). In our case, the use of coupled of reservoir computers is introduced due to the large number of points on the map, making it computationally challenging to utilize a single large reservoir computer. The reservoirs are coupled together by making the reservoirs functions of points outside their forecast domain, effectively creating overlap.

2   Background

Reservoir computing is a type of recurrent neural net where the only layer trained is the output layer, which is done with a simple linear method. Compared to traditional recurrent neural networks, reservoir computers utilize randomly generated input and middle weights, which in effect reduces training time significantly. The reservoir computer is stated as follows (Jaeger and Haas, 2004),

$$\boldsymbol{r}_{i+1} = q(\mathbf{A}\boldsymbol{r}_i + \mathbf{W}_{in}\boldsymbol{X}_i + \boldsymbol{b}) \tag{1}$$

$$\boldsymbol{Y}_{i+1} = \mathbf{W}_{out}\boldsymbol{r}_{i+1} \tag{2}$$

The inputs $\boldsymbol{X}_i$ (of total length dx) is the raw data which describes a system, in our case these would be the temperatures at points in the sea. These are fed into the reservoir via the input matrix $\mathbf{W}_{in}$, which has weights which are determined via sampling a uniform distribution $U(-\sigma, \sigma)$. The reservoir dimension N describes how many nodes there are to be within the reservoir. Therefore in order to transform the inputs into the space of the reservoir, the dimensions of $\mathbf{W}_{in}$ will be $N \times dx$.

The reservoir state $\boldsymbol{r}$, which evolves according to Eq. 1, carries information about the current and previous states of the system. During training, the reservoir states are horizontally concatenated as time evolves to form the matrix $\mathbf{R} = \left[\boldsymbol{r}_1 | \boldsymbol{r}_2 | \ldots | \boldsymbol{r}_{t_{train}}\right]$, where t_{train} is the number of data points being used to train the model, and so correspondingly the computational complexity associated with the matrix operations stated in Eq. 2. The dimensions of $\mathbf{R}$ at the end of the training phase will then be $N \times t_{train}$.

The reservoir matrix $\mathbf{A}$ which contains the middle weights, is a sparse matrix with a set density d, with nonzeros values that are sampled from a uniform distribution $U(-\beta, \beta)$. The matrix $\mathbf{A}$ will have dimensions of $N \times N$ corresponding to the N reservoir nodes. The spectral radius ρ of $\mathbf{A}$ is an important metaparameter in the formation of the RC (Jiang and Lai, 2019), and can be adjusted by scaling $\mathbf{A}$ (which essentially just involves changing β). The activation function $q(\mathbf{s})$ is usually picked to be a nonlinear function such as the sigmoid function or the hyperbolic tangent function. Often, a bias term b is included when one desires to shift the activation function a set amount. After the training dataset is cycled through and the matrix $\mathbf{R}$ is completed, the output matrix $\mathbf{W}_{out}$ is then found via a ridge regression:

$$\mathbf{W}_{out} = \mathbf{Y}\mathbf{R}^{\mathbf{T}}(\mathbf{R}\mathbf{R}^{\mathbf{T}} + \lambda\mathbf{I})^{-1} \tag{3}$$

Which utilizes a regularization parameter λ to prevent overfitting. As the output matrix is transforming reservoir states to a desired output $\boldsymbol{Y}_{i+1}$ (of length px), the dimension of $\mathbf{W}_{out}$ will be $px \times N$. The trained model can then be used to forecast autonomously by inserting the newly predicted values $\boldsymbol{Y}_{i+1}$ back into the reservoir on the next iteration as $\boldsymbol{X}_i$.

3   Coupled Reservoir Computers

As we would like to predict the SST across the entire globe, it is computationally challenging to use a single reservoir computer to forecast for the entire map due to the number of points on the map. From the dataset, there will be a total of $n * m$ points on the map. If we were to use a single reservoir computer, the size of the input data $\boldsymbol{X}_i$ would be around $0.71 * n * m$ (as 71% of the Earth's surface is water) for each time step. It will be seen in the *Data* section that the values of n and m that are ultimately to train our model are 120 and 240 respectively, hence the total number of points that are not on land across the map is about 20448. Hence, even if the reservoir dimension N is $15 : 1$ with respect to the size of the input data (which is relatively small, for example: the value of N commonly used for the 3D Lorenz system is 1000 (Bollt, 2021)), the size of the resulting matrix $\mathbf{A}$ would be 306720×306720, which assuming each element requires 8 bytes, would require 750GB to store.

Therefore in this application, it is better to follow the methodology laid out by Pathak et al. (2018) to model the evolution of the SST with the use of smaller reservoir computers which each cover a small domain of the map. This approach is advantageous as the input and middle weights for a reservoir computer are chosen randomly, hence allowing us to reuse the same input matrices[1] $\mathbf{W}_{in}$ and the same reservoir matrix $\mathbf{A}$ between the individual RCs. Now, each individual RC learns the local behavior of the points within its forecasting domain (defined as a *pack*) while still being connected to a the global system via coupling with its neighbors.

A pack is essentially a collection of contiguous indices on the greater map that will be assigned to a given RC. In other words, a pack is an RC model of the dynamics, for a local region of the globe, that is designed to accommodate any local peculiarities of the land-water interface and also it couples into other neighboring packs. This reservoir computer will then be solely responsible for predicting the temperatures within its pack as time evolves. For simplicity's sake in this study, points

---

[1]Due to varying values of dx due to points on land, we cannot share a single input matrix $\mathbf{W}_{in}$ between all RCs.

within a pack are grouped in the shape of rectangles, where the number of rows of points within the pack is defined as
n_{pack} and the number of columns subsequently m_{pack}. Therefore, each RC will be responsible for forecasting a maximum of
$n_{pack} * m_{pack}$ spatial points on the map. As there are $n * m$ total points on the map, the number of RCs needed will then be P_f
$= (n * m)/(n_{pack} * m_{pack})$. An illustration of the various packs on a sample map are provided in Fig. 1.
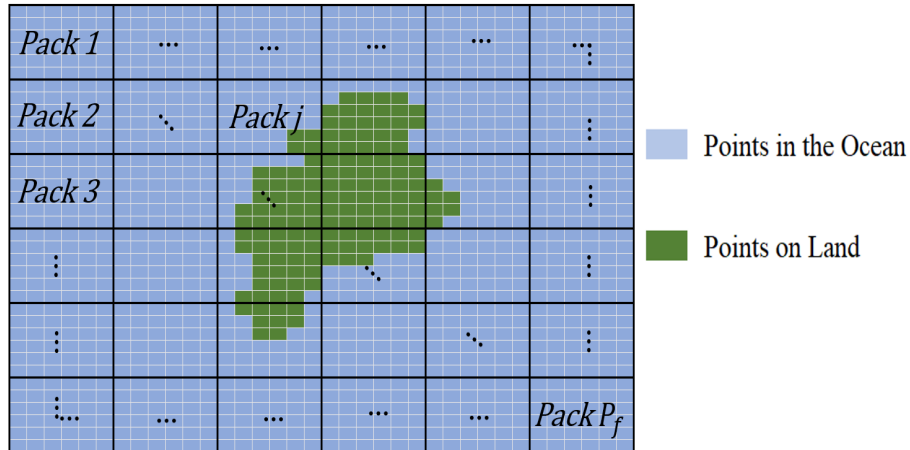
Figure 1. The various packs on a sample map. The points within a pack that are blue are the eligible points in the ocean that the pack's designated reservoir computer will attempt to forecast. Points in green represent land, which are not eligible points and whose indices are ignored in the formation of the pack.

No points containing land are assigned to a RC, therefore some packs will have more or less points than others, due to these points on land. It should also be noted that no point on the map can be in more than one pack, as packs do not overlap. The sea surface temperatures of the points within the jth pack at the ith day are stacked in the vector $\boldsymbol{X}_i^{j_{pack}}$ which is of length px.

$$\boldsymbol{X}_i^{j_{pack}} = \begin{bmatrix} X_i^{j_{pack},1} \\ X_i^{j_{pack},2} \\ \vdots \\ X_i^{j_{pack},px} \end{bmatrix} \tag{4}$$

For the various reservoirs to interact with one another, coupling is introduced by including the SST of the neighbors surrounding a pack. The neighbors of a pack are the non-land points that are either directly touching or on the corner of a point within the pack. As many points within a pack share similar neighbors, only unique neighbors are kept and their sea surface temperatures at the ith day are compiled into the vector $\boldsymbol{X}_i^{j_{neighbor}}$. The neighbors for a given pack on the sample map are illustrated below in Fig. 2.
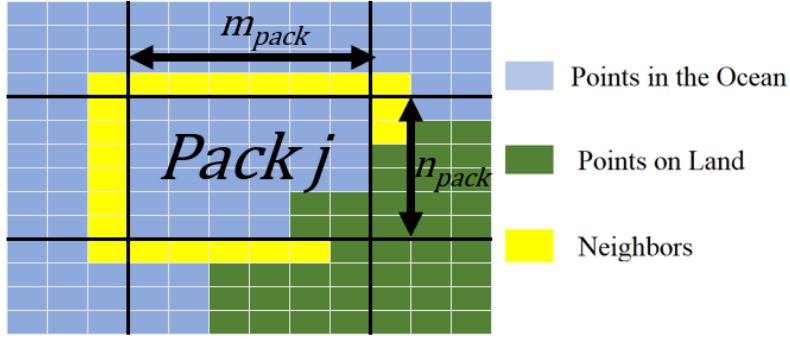
Figure 2. Illustration of the jth pack with its valid neighbors denoted by a yellow highlight. In this example it is evident that px is equal to 32 (number of pack points in the ocean) and dx is equal to 54 (px plus the number of neighbors). It is also evident that n_{pack} and m_{pack} are both equal to 6 in this case.

The vectors $\boldsymbol{X}_i^{j_{pack}}$ and $\boldsymbol{X}_i^{j_{neighbor}}$ are combined to form the vector $\boldsymbol{X}_i^j$ (of length dx), which contains all of the sea surface temperatures for the pack and its neighbors at the ith day, and is ultimately the input for the jth reservoir computer.

$$\boldsymbol{X}_i^j = \begin{bmatrix} \boldsymbol{X}_i^{j_{pack}} \\ \boldsymbol{X}_i^{j_{neighbors}} \end{bmatrix} \tag{5}$$

100   As we would like the model to predict sea surface temperatures within the pack at the next day $\boldsymbol{Y}_{i+1}^{j_{pack}}$, the reservoir computer for the jth pack can now be written as:

$$\boldsymbol{r}_{i+1}^j = q(\mathbf{A}\boldsymbol{r}_i^j + \mathbf{W}_{in}^{dx}\boldsymbol{X}_i^j + \boldsymbol{b}) \tag{6}$$

$$\boldsymbol{Y}_{i+1}^{j_{pack}} = \mathbf{W}_{out}^j \boldsymbol{r}_{i+1}^j \tag{7}$$

Each reservoir computer is trained over the entire training dataset from $i = 1 : t_{train}$ days. Each pack contains a distinct
105   output matrix $\mathbf{W}_{out}^j$, such that the reservoir states are matched with the values of the SST within the pack at the next day. In order to save computer memory, it is advisable to create an array of input matrices with values of dx from 1 to $(n_{pack} + 2) * (m_{pack} + 2)$ and then assign these to RCs with a similar value of dx, denoted by $\mathbf{W}_{in}^{dx}$. One middle weight matrix $\mathbf{A}$ is shared between all the RCs, as the reservoir dimension N is set to be fixed between all of the reservoirs. The architecture of a single reservoir computer is described in Fig. 3.
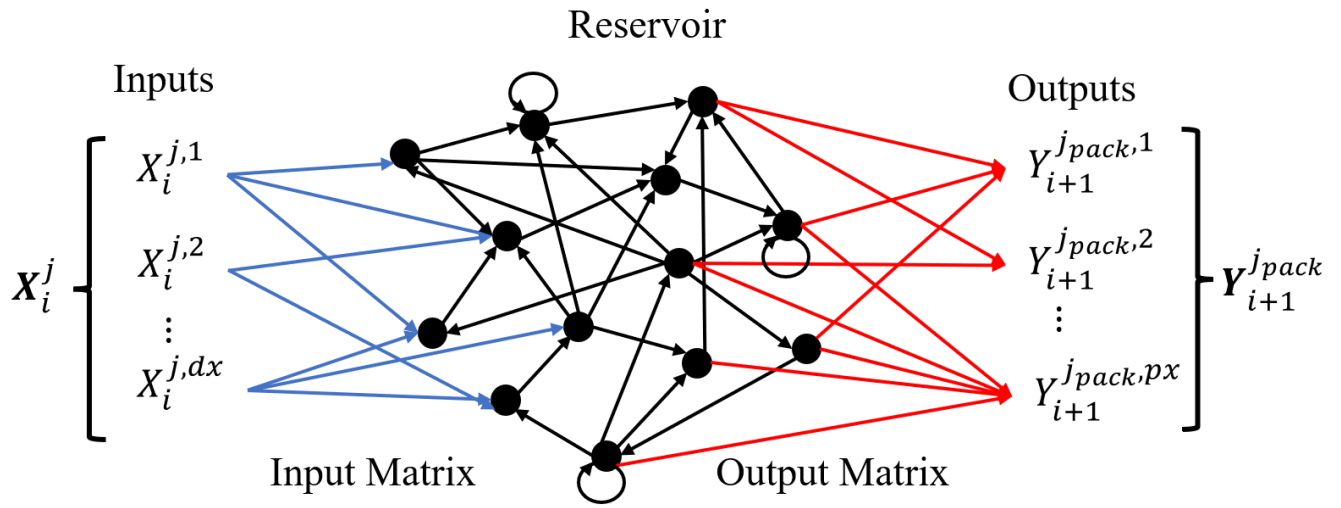
Figure 3. Illustration of the architecture of the jth reservoir computer.

4 Data

The dataset used to train and validate the model is titled "GHRSST Level 4 MUR 0.25deg Global Foundation Sea Surface Temperature Analysis (v4.2)" which contains sea surface temperature data in degrees Kelvin on a global $0.25°$ grid from 2002 to 2021 in one day increments. This version is based on nighttime GHRSST L2P skin and sub skin SST observations from several instruments, and is publicly available online via PODAAC (MUR25-JPL-L4-GLOB-v04.2). The data was downloaded with the use of OPeNDAP on 10/10/2021.

The years 2003 to 2020 of the dataset were selected to form the training and validation dataset. The data is given in an equirectangular format, which is used throughout the modelling process for simplification purposes even though this consequentially leads to a more refined mesh near the poles. The time series data was split into a training and validation dataset, consisting of 6,533 days and 42 days respectively. We choose not to normalize the data, as the data is univariate and the reservoir state is effectively scale free, with scale being re-introduced with the trained output matrix.

In order to reduce the number of spatial points within the dataset, the data was discretized such that the sea surface temperature was now on a global $1.5°$ degree grid. This was performed by grouping original data points in a 6×6 matrix and then taking the average over the group. If a grouping contained a point on land, this value was ignored in the computation of the average of the group. Therefore, the dataset for a given day went from a 720×1440 to a 120×240. Hence, $n = 120$ and $m = 240$.

5 Forecasting

To subsequently forecast the global SST with the trained model, two different prediction types are performed. To begin testing the short-term accuracy of the model, daily predictions are performed over the course of 6 weeks. The actual values of the SST are fed as inputs into the reservoir during this time period and the predicted values for the next day are read out. This type of forecast has a real world application in the form of filling in SST datasets when there is cloud cover or data corruption (Case, Jonathan L. et al., 2008), as the model can be used to estimate data for the missing days. Then, to test the long-term accuracy of the model, the model is allowed to run autonomously over the same 6 weeks. Now, the model is still predicting SST each day but it only has access to it's own previous prediction. This form of the forecast would be more applicable to weather prediction, as it could be coupled with an atmospheric model to help predict near future weather patterns.

For both prediction types, the reservoir states are all cleared to zero prior to forecasting, and then ran over t_{warmup} days prior to the validation time frame, hence providing an initial condition for the model to begin from. Via cross validation, several metaparameters (σ, N, and t_{warmup}) were optimized and the values that were found to perform the best are described in Table 1. It was noticed that the results were not significantly increased for a value of t_{warmup} greater than 35 days. Results did consistently improve with an increasing reservoir dimension N, leading us to choose the value $N = 1000$. One of the more sensitive metaparameters was the value of σ, which was found to provide optimal results when having a magnitude of the order 10^{-4}. In the spirit of simplicity, we choose not to rigorously optimize the remaining metaparameters, and rather choose them based off of heuristics.

Table 1. Metaparameters Used.

Metaparameter	Value
σ	$3e-4$
ρ	1.0
b	0
$q(s)$	$tanh(s)$
λ	0.02
d	0.05
N	1000
n_{pack}	4
m_{pack}	4
t_{train}	$6,533$ days
$t_{validate}$	42 days
t_{warmup}	35 days

The model was implemented on the MATLAB R2019b platform, from a personal laptop, and the training time was slightly over 40 minutes. It is likely, given the embarrassingly parallelizable nature of this task, that an implementation that leverages
145 GPU style computation could speed this stage up considerably. To observe the effect of the random input and middle weights on the model, 15 different models are created all with the same metaparameters as described in Table 1. Even though the model predicts SST across the entire Earth, several time series at points chosen arbitrarily are included in each section to locally validate the forecast, the coordinates of which can be found below in Table 2. In this local time series analysis, the forecasts from the 15 different models are averaged and one standard deviation in the predicted values are represented above and below
150 the average value with a shaded outline.

Table 2. Coordinates of Chosen Locations.

Location	Key	Latitude ($^\circ$ N)	Longitude ($^\circ$ E)
Cook Strait	a	-41.25	174.50
Gulf of Mexico near Key West, FL	b	24.75	-81.75
Coast of Gabon	c	0.75	8.25
Southern Ocean near Heard Island	d	-55.00	73.50
East Coast of Japan	e	35.25	141.75
Mozambique Channel	f	-18.75	41.25
Pacific Ocean near Tuvalu	g	-8.25	179.25
Coast of Ecuador	h	-3.75	-81.75
Bass Strait	i	-39.75	146.25
Laccadive Sea	j	-6.25	77.00

To determine the quality of the forecast over the entire ocean, the mean absolute error (MAE), the root mean square error (RMSE), and the maximum error in the forecast for a given day across the entire map are found. To find the MAE in the forecast at a given day, a weighted average is performed on the error e_i across the map, where $e_{p,i}$ denotes the absolute error at point p on the map at the ith day. We perform this weighting due to the mesh being more refined near the poles compared to points near the equator, hence the area attributable to a given point Ω_p isn't necessarily the same for other points. It should also be noted that we treat areas near the poles consisting of time-dependent sea ice simply as points within the ocean. The actual area encompassed by a given point was found by simply using MATLAB's built in function areaquad(). The MAE in the forecast across the map at the ith day is then given by Equation 8, where k is the number of points on the map that lie in the ocean ($k \approx 0.71 * n * m$).

$$MAE_i = \frac{1}{\sum_{p=1}^{k} \Omega_p} \sum_{p=1}^{k} e_{p,i} \Omega_p \tag{8}$$

Similarly, the RMSE on the ith day is then given by Equation 9:

$$RMSE_i = \sqrt{\frac{1}{\sum_{p=1}^{k} \Omega_p} \sum_{p=1}^{k} e_{p,i}^2 \Omega_p} \tag{9}$$

Finally, the maximum error is simply the largest error in the forecast across all points on the ith day. These error values are found for each of the 15 models every day in the forecasting period, and then subsequently average values and standard deviations between models are found to observe the effect of randomness.

5.1 Daily Forecasts

Daily forecasting operates by continually inserting the actual SST of the previous day $\boldsymbol{X}_i$ into the reservoirs and then reading out what the model predicts will be the SST at the next day $\boldsymbol{Y}_{i+1}$, subsequently repeating this procedure over the course of the validation time frame. The time series for the forecasted SST at the 10 different points are provided below in Fig. 4 and is also matched with the actual SST each day in Fig. 5.
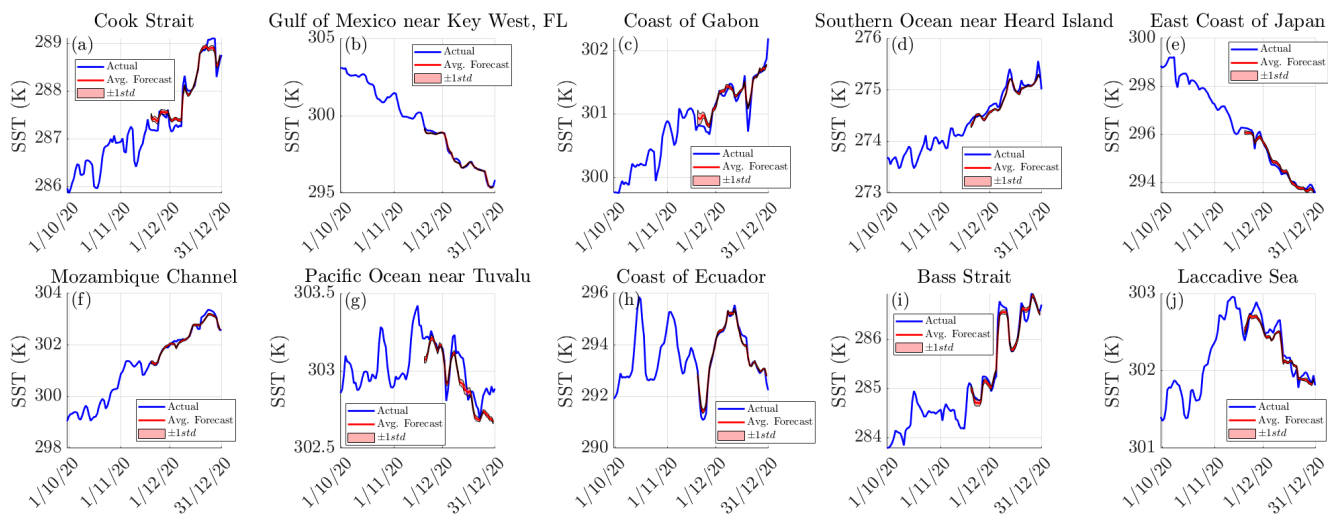
Figure 4. One day forecasts of the sea surface temperature at various points in the ocean. The average forecasted SST and the true SST is represented by the red and the blue line respectively. About the average there is a shaded outline representing ±1 standard deviation in the forecast between models, though it is not entirely evident due to all models ultimately predicting similar values.
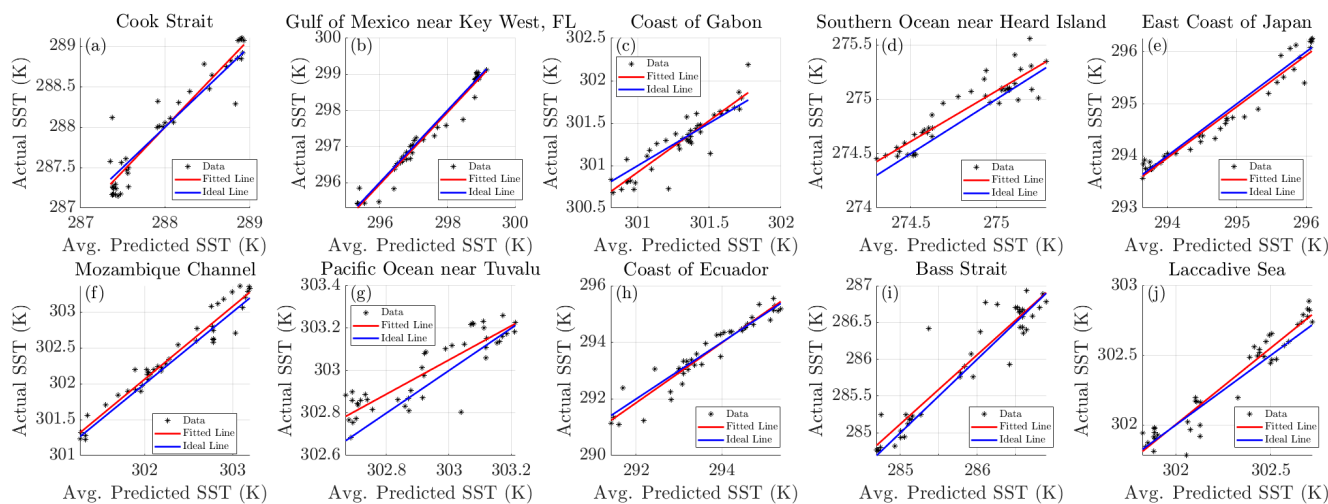
Figure 5. The true SST compared to the average predicted SST, for the model predicting one day at a time. The red line indicates the resulting linear fit, while the blue line is the ideal place where the forecasted SSTs would match the true SST.

Table 3. Regression Statistics for the Daily Forecasts. Note that ideally, the slope m would be equal to one, the intercept b would be equal to zero, and the correlation coefficient r would be one. Standard errors are provided for both m and b.

Location	m	b	r
a	1.09 ± 0.06	-27.62 ± 16.1	0.95
b	1.00 ± 0.03	0.36 ± 8.81	0.98
c	1.21 ± 0.09	-63.96 ± 25.9	0.91
d	0.93 ± 0.06	19.94 ± 17.8	0.89
e	0.99 ± 0.03	3.92 ± 10.3	0.98
f	1.01 ± 0.04	-3.86 ± 11.5	0.97
g	0.80 ± 0.07	59.17 ± 21.2	0.88
h	1.06 ± 0.05	-17.51 ± 14.4	0.95
i	0.94 ± 0.05	16.90 ± 15.0	0.94
j	1.10 ± 0.04	-29.89 ± 14.6	0.96

Via Fig. 4 it is apparent that the forecasted SST closely follows the actual SST over the validation time frame for almost all locations. The model is seen to pick up on the changes in SST at locations where there is not a clear trend and the change in SST is seemingly chaotic, such as locations *h*, *i*, and *j*, though there are some inaccuracies with the forecast at location *g*. There is also very little deviation between models, indicating that the effect of randomness on the model is practically negligent with regard to forecasting daily SST. The correlation coefficients for the chosen sites are all 0.88 or greater, and the value of m is close to one (and with minimal standard error) for most locations, therefore indicating a strong relationship between the model's forecast and the actual values. To quantify how the model performs over the entire globe the MAE, RMSE, and the maximum error for the daily forecasts are described below in Fig. 6.

175
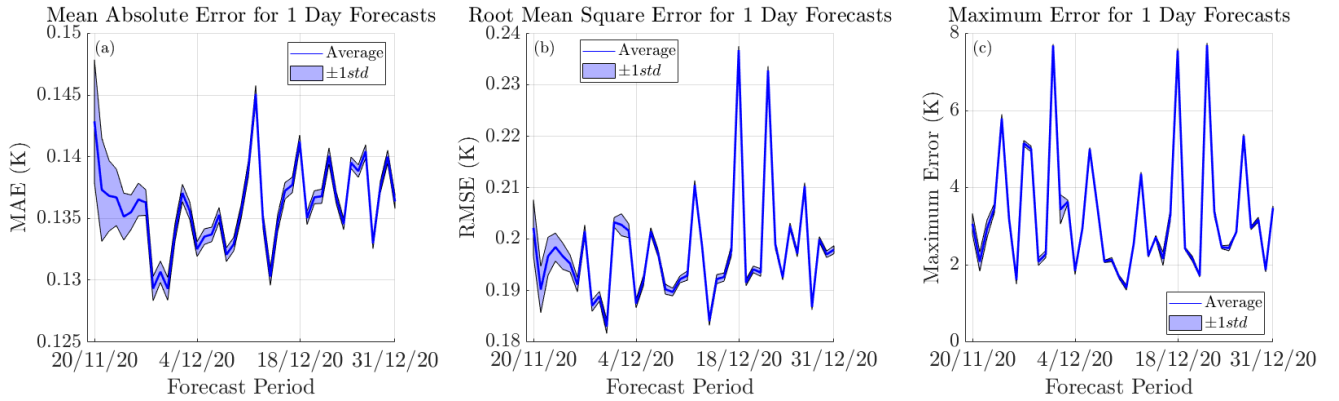
Figure 6. The evolution of the error for the 1 day forecasts. The mean absolute error in the forecast over the entire ocean is depicted in (a), the root mean square in (b), and the maximum error in (c). The model average is represented via a dark blue line and one standard deviation is represented by a shaded outline above and below the average.

From Fig. 6 we see that for each daily prediction the average MAE between models typically fluctuates between $0.13 - 0.15K$, the RMSE between $0.18 - 0.24K$, and the maximum error between $1 - 8K$. By calculating the mean of the model averages over the course of the 6 week forecasting period we find that for a 1 day prediction horizon, the model has as an average MAE of $0.136K$, an average RMSE of $0.197K$, and an average maximum error of $3.308K$. For context, for a 1 day prediction horizon Xiao et al. (Xiao et al., 2019b) reported a RMSE of $0.35K$ for their forecast over the East China Sea with the use of a deep learning model constructed from ConvLSTM's, and Shi et. al (Shi et al., 2022) reported a RMSE of $0.241K$ for their cyclic evolutionary network model forecasting over the South China Sea. It should also be noted that our error values typically don't decrease over the forecasting period, indicating that the chosen warm-up time of 35 days is sufficient, as there would be a decline in the error over time if the reservoir was gradually benefiting from more provided information.

5.2   6 Week Forecast

Meanwhile for the 6 week forecast, the forecasted sea surface temperatures $\boldsymbol{Y}_{i+1}$ are inputted back into the reservoir on the next day, therefore taking the place of the actual SST $\boldsymbol{X}_{i+1}$. This effectively allows the model to run autonomously over the validation time time frame for a total of 42 days. The 10 time series for the forecasted SST are provided below in Fig. 7.
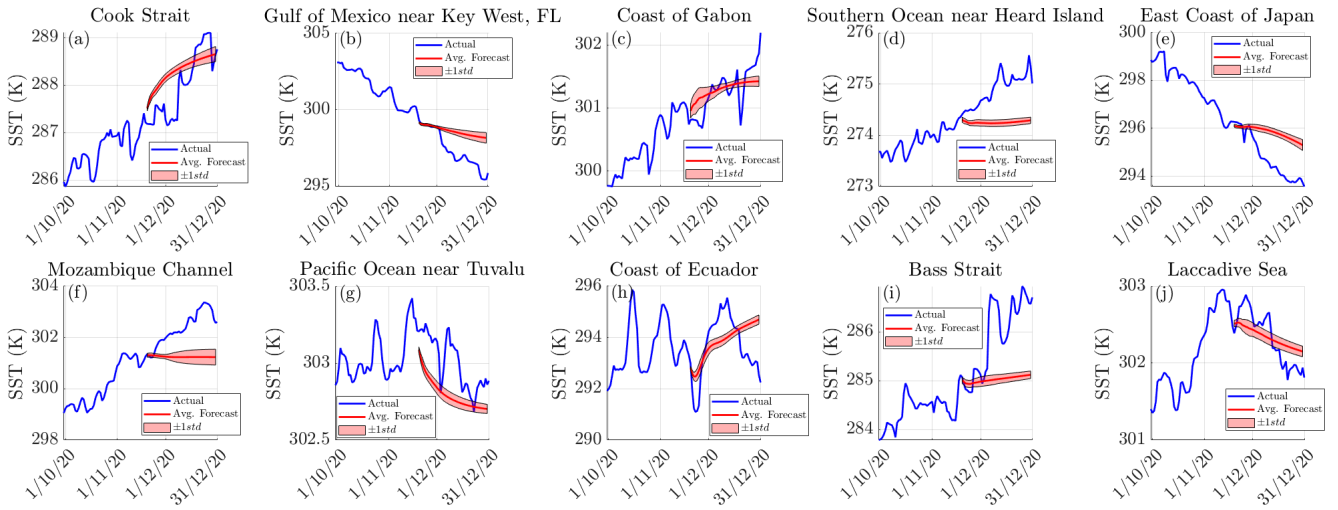
Figure 7. Forecasted sea surface temperatures at various points in the ocean with the model running autonomously. The average forecasted and the actual SST is represented by the red and the blue line respectively, with a shaded outline surrounding the average forecasted SST representing ±1 standard deviation in the forecast between the models.

From Fig. 7 it is apparent that the model typically predicts the general change in the SST for locations *a*, *b*, *c*, *g*, *h*, and *j*. We find it especially impressive that the model is able to predict the cooling of the SST found at *g* and *j* and the warming at *h*. Meanwhile, the results for locations *d*, *e*, and *f* are very poor given that the change in SST is fairly linear in the time leading up
to the forecasting period and during it. The results at location *i* are also poor, as the model is unable to anticipate the rise in SST that began around the commencement of the forecasting period. These forecasts also depict how the intrinsic randomness of the model does play a slight role in the model output as time evolves, but it is an interesting feature that the standard deviation between models appears to reach a limit after several days of forecasting as seen in Fig. 7 *(d)*, *(e)*, *(g)*, *(h)*, *(i)* and *(j)*, and the standard deviation between models even decreases in Fig. 7 *(c)*. To quantify how the model performs globally, we now refer
to Fig. 8 which depicts the error in the SST forecast (which is an average between the 15 models) as well as the MAE, RMSE, and maximum error in the forecast described by Fig. 9.
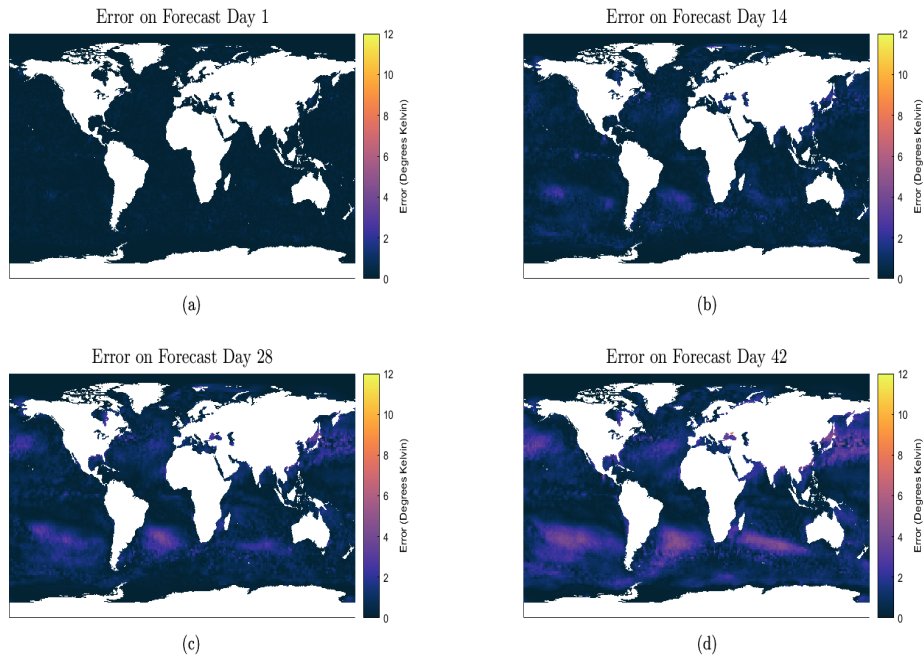
Figure 8. The evolution of the error across the globe for the 6 week forecast. The error at each location is simply the difference between the average forecasted value and the actual value.
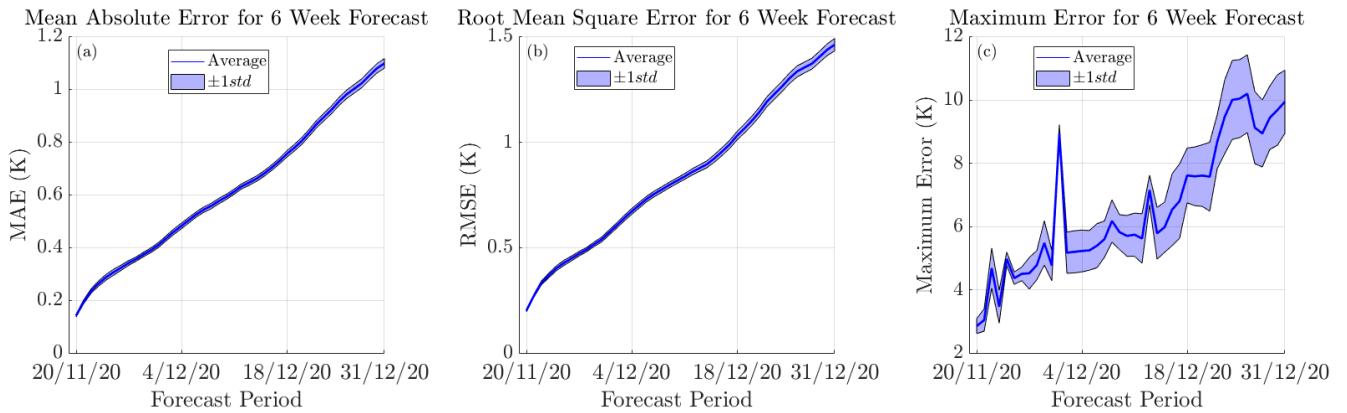
Figure 9. The evolution of the error for the 6 week forecast. The average model error statistics are plotted each day as well as ±1 standard deviation. The mean absolute error in the forecast over the entire ocean is depicted in (a), the root mean square error in (b), and the maximum error in (c).

Via Fig. 8 it is observed that the model generally performs the best near the equator, with some slight difficulty predicting the general warming of the ocean in the Southern Hemisphere and cooling in the Northern Hemisphere during this time frame.

The average MAE and RMSE rise to $0.32K$ and $0.45K$ respectively on the 7th day of forecasting, subsequently increasing to $0.73K$ and $0.99K$ respectively by the end of 4 weeks. For context, for a prediction horizon of 1 week Xiao et. al (Xiao et al., 2019b) reports a RMSE of $0.85K$ and Shi et. al (Shi et al., 2022) reports a RMSE of $0.687K$ for their respective models mentioned previously. Meanwhile for a prediction horizon of 4 weeks, Yang et al. (Yang et al., 2018) reports a RMSE of $0.726K$ for their CFCC-LSTM model forecasting over the Bohai Sea and a RMSE of $1.070K$ for the China Ocean. Similar to the daily forecasts, it is also apparent that there is little deviation in the RMSE and MAE across the 15 models, once again indicating that the models typically perform similarly regardless of the random weights which they are constructed from.

6 Conclusions

With the use of coupled reservoir computers, and specifically a collection of patches that represent local regions and designed to accommodate coastal-land interface variations, we were able to model for excellent forecasting the spatiotemporally complex dynamics of the global sea surface temperature over several weeks. The relative simplicity of the network architecture and the minimal training time is striking relative to other machine learning concepts. Even though our model is intended to describe the dynamics of the entire ocean, it is still able to predict SST at specific locations. In the future, it is of interest to explore the use of Next-Generation Reservoir Computers (NG-RC) in the task of predicting SST, as NG-RCs provide the added benefit of less metaparameters to tune compared to a traditional RC (Jaeger and Haas, 2004; Bollt, 2021; Gauthier et al., 2021). It is also of interest to input other variables into the reservoir besides the SST, such as the surrounding air temperature (Jahanbakht et al., 2021) to observe if the results can be further improved.

Code availability. The code is available online at https://github.com/BenWalleshauser/Predicting-SST-w-.-Coupled-RCs.

Author contributions. The idea was originally conceived of by E.M.B. and B.W. was responsible for the creation of the model. B.W. and E.M.B. both contributed to the analyses and the creation of the paper.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. E.M.B. has received funding from the Army Research Office (ARO), the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation (NSF) and National Institutes of Health (NIH) CRNS program, and the Office of Naval Research (ONR) during the period of this work.

References

Bollt, E.: On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD, Chaos: An Interdisciplinary Journal of Nonlinear Science, 31, 013 108, 2021.

Case, Jonathan L., Santos, Pablo, Lazarus, Steven M., Splitt, Michael E., Haines, Stephanie L., Dembek, Scott R., and Lapenta, William M.: A Multi-Season Study of the Effects of MODIS Sea-Surface Temperatures on Operational WRF Forecasts at NWS Miami, FL, New Orleans, LA, 2008.

Collins, D. C., Reason, C. J. C., and Tangang, F.: Predictability of Indian Ocean sea surface temperature using canonical correlation analysis, Climate Dynamics, 22, 481–497, https://doi.org/10.1007/s00382-004-0390-4, 2004.

Dado, J. M. B. and Takahashi, H. G.: Potential impact of sea surface temperature on rainfall over the western Philippines, Progress in Earth and Planetary Science, 4, 23, https://doi.org/10.1186/s40645-017-0137-6, 2017.

Gauthier, D. J., Bollt, E., Griffith, A., and Barbosa, W. A. S.: Next generation reservoir computing, Nature Communications, 12, 5564, https://doi.org/10.1038/s41467-021-25801-2, 2021.

Ghosh, S., Senapati, A., Mishra, A., Chattopadhyay, J., Dana, S., Hens, C., and Ghosh, D.: Reservoir computing on epidemic spreading: A case study on COVID-19 cases, Physical Review E, 104, https://doi.org/10.1103/PhysRevE.104.014308, 2021.

Gomez, A. M., McDonald, K. C., Shein, K., DeVries, S., Armstrong, R. A., Hernandez, W. J., and Carlo, M.: Comparison of Satellite-Based Sea Surface Temperature to In Situ Observations Surrounding Coral Reefs in La Parguera, Puerto Rico, Journal of Marine Science and Engineering, 8, https://doi.org/10.3390/jmse8060453, 2020.

Hadaeghi, F., Diercks, B.-P., Schetelig, D., Damicelli, F., Wolf, I. M. A., and Werner, R.: Spatio-temporal feature learning with reservoir computing for T-cell segmentation in live-cell CA-2+ fluorescence microscopy, Scientific Reports, 11, 8233, https://doi.org/10.1038/s41598-021-87607-y, 2021.

Jaeger, H. and Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication, Science, 304, 78–80, https://doi.org/10.1126/science.1091277, 2004.

Jahanbakht, M., Xiang, W., and Azghadi, M. R.: Sea Surface Temperature Forecasting With Ensemble of Stacked Deep Neural Networks, IEEE Geoscience and Remote Sensing Letters, pp. 1–5, https://doi.org/10.1109/LGRS.2021.3098425, 2021.

Jiang, J. and Lai, Y.-C.: Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius, Physical Review Research, 1, https://doi.org/10.1103/PhysRevResearch.1.033056, 2019.

Johnson, S. D., Battisti, D. S., and Sarachik, E. S.: Empirically Derived Markov Models and Prediction of Tropical Pacific Sea Surface Temperature Anomalies, Journal of Climate, 13, 3 – 17, https://doi.org/10.1175/1520-0442(2000)013<0003:EDMMAP>2.0.CO;2, place: Boston MA, USA Publisher: American Meteorological Society, 2000.

Kim, M., Yang, H., and Kim, J.: Sea Surface Temperature and High Water Temperature Occurrence Prediction Using a Long Short-Term Memory Model, Remote Sensing, 12, https://doi.org/10.3390/rs12213654, 2020.

Krishnamurti, T. N., Chakraborty, A., Krishnamurti, R., Dewar, W. K., and Clayson, C. A.: Seasonal Prediction of Sea Surface Temperature Anomalies Using a Suite of 13 Coupled Atmosphere–Ocean Models, Journal of Climate, 19, 6069 – 6088, https://doi.org/10.1175/JCLI3938.1, place: Boston MA, USA Publisher: American Meteorological Society, 2006.

Kug, J.-S., Kang, I.-S., Lee, J.-Y., and Jhun, J.-G.: A statistical approach to Indian Ocean sea surface temperature prediction using a dynamical ENSO prediction, Geophysical Research Letters, 31, https://doi.org/https://doi.org/10.1029/2003GL019209, _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2003GL019209, 2004.

265   Lins, I. D., Araujo, M., Moura, M. d. C., Silva, M. A., and Droguett, E. L.: Prediction of sea surface temperature in the tropical Atlantic by support vector machines, Computational Statistics & Data Analysis, 61, 187–198, https://doi.org/https://doi.org/10.1016/j.csda.2012.12.003, 2013.

MUR25-JPL-L4-GLOB-v04.2: GHRSST Level 4 MUR 0.25deg Global Foundation Sea Surface Temperature Analysis (v4.2), jPL MUR MEaSUREs Project. 2019. GHRSST Level 4 MUR 0.25 deg Global Foundation Sea Surface Temperature Analysis. Ver. 4.2. PO.DAAC,

270   CA, USA. Dataset accessed [2021-10-10] at https://doi.org/10.5067/GHM25-4FJ42.

Novi, L., Bracco, A., and Falasca, F.: Uncovering marine connectivity through sea surface temperature, Scientific Reports, 11, https://doi.org/10.1038/s41598-021-87711-z, 2021.

Pandey, S. and Schumacher, J.: Reservoir computing model of two-dimensional turbulent convection, Physical Review Fluids, 5, https://doi.org/10.1103/PhysRevFluids.5.113506, 2020.

275   Pastor, F.: Sea Surface Temperature: From Observation to Applications, Journal of Marine Science and Engineering, 9, 1284, https://doi.org/10.3390/jmse9111284, 2021.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E.: Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, Physical Review Letters, 120, https://doi.org/10.1103/PhysRevLett.120.024102, 2018.

Patil, K., Deo, M. C., and Ravichandran, M.: Prediction of Sea Surface Temperature by Combining Numerical and Neural Techniques,

280   Journal of Atmospheric and Oceanic Technology, 33, 1715 – 1726, https://doi.org/10.1175/JTECH-D-15-0213.1, place: Boston MA, USA Publisher: American Meteorological Society, 2016.

Shi, J., Yu, J., Yang, J., Xu, L., and Xu, H.: Time Series Surface Temperature Prediction Based on Cyclic Evolutionary Network Model for Complex Sea Area, Future Internet, 14, 96, https://doi.org/10.3390/fi14030096, place: Basel Publisher: MDPI AG, 2022.

Stockdale, T. N., Balmaseda, M. A., and Vidard, A.: Tropical Atlantic SST Prediction with Coupled Ocean–Atmosphere GCMs, Journal

285   of Climate, 19, 6047 – 6061, https://doi.org/10.1175/JCLI3947.1, place: Boston MA, USA Publisher: American Meteorological Society, 2006.

Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P.: Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics, Neural Networks, 126, 191–217, https://doi.org/https://doi.org/10.1016/j.neunet.2020.02.016, 2020.

290   Xiao, C., Chen, N., Hu, C., Wang, K., Gong, J., and Chen, Z.: Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach, Remote Sensing of Environment, 233, 111 358, https://doi.org/https://doi.org/10.1016/j.rse.2019.111358, 2019a.

Xiao, C., Chen, N., Hu, C., Wang, K., Xu, Z., Cai, Y., Xu, L., Chen, Z., and Gong, J.: A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data, Environmental Modelling & Software, 120, 104 502,

295   https://doi.org/https://doi.org/10.1016/j.envsoft.2019.104502, 2019b.

Xue, Y. and Leetmaa, A.: Forecasts of tropical Pacific SST and sea level using a Markov model, Geophysical Research Letters, 27, 2701–2704, https://doi.org/https://doi.org/10.1029/1999GL011107, 2000.

Yang, Y., Dong, J., Sun, X., Lima, E., Mu, Q., and Wang, X.: A CFCC-LSTM Model for Sea Surface Temperature Prediction, IEEE Geoscience and Remote Sensing Letters, 15, 207–211, https://doi.org/10.1109/LGRS.2017.2780843, 2018.

300   Zhang, Q., Wang, H., Dong, J., Zhong, G., and Sun, X.: Prediction of Sea Surface Temperature Using Long Short-Term Memory, IEEE Geoscience and Remote Sensing Letters, 14, 1745–1749, 2017.

Zhang, X., Li, Y., Frery, A., and Ren, P.: Sea Surface Temperature Prediction With Memory Graph Convolutional Networks, https://doi.org/10.25455/wgtn.15111642.v1, 2021.