Geosci. Model Dev. Discuss., doi:10.5194/gmd-2015-273, 2016
Manuscript under review for journal Geosci. Model Dev.
Published: 11 February 2016

Geoscientific Model Development
Discussions
Open Access
EGU

ClimateLearn: A machine-learning approach for climate prediction using network measures

Qing Yi Feng[1], Ruggero Vasile[2,3], Marc Segond[4], Avi Gozolchiani[5], Yang Wang[5], Markus Abel[3], Shilomo Havlin[5], Armin Bunde[6], and Henk A. Dijkstra[1]

[1]Institute for Marine and Atmospheric research Utrecht, Utrecht University, The Netherlands
[2]UP Transfer, Potsdam, Germany
[3]Ambrosys, Potsdam, Germany
[4]European Centre for Soft Computing, Mieres, Spain
[5]Bar-Ilan University, Isreal
[6]University of Giessen, Germany

Correspondence to: Q. Y. Feng (Q.Feng@uu.nl) and R. Vasile (ruggero.vasile@ambrosys.de)

Abstract. We present the toolbox `ClimateLearn` to tackle problems in climate prediction using machine learning techniques and climate network analysis. The package allows basic operations of data mining, i.e. reading, merging, and cleaning data, and running machine learning algorithms such as multilayer artificial neural networks and symbolic regression with genetic programming. Because spatial-temporal information on climate variability can be efficiently represented by complex network measures, such data are considered here as input to the machine-learning algorithms. As an example, the toolbox is applied to the prediction of the occurrence and the development of El Niño in the equatorial Pacific, first concentrating on the occurrence of El Niño events one year ahead and second on the evolution of sea surface temperature anomalies with a lead time of three months.

1 Introduction

Machine learning is a branch of computer science concerned with automated recognition of (spatio-temporal) patterns from data (Mitchell, 1997). It has been increasingly employed in the study of "big data" with the aim to investigate data syntactically and semantically. In essence, this means an automated search for a best model, given a certain task and corresponding data. A large number of algorithms have been designed for different tasks in which the approach is borrowed from bio-inspired investigations on artificial intelligence (in older times a synonym of machine learning). Given a task, a human learns what to do and -hopefully- optimizes the working schedule according to the given side conditions. This is how machines learn from data: a task is formulated and then a learning process starts, which consists in building statistical models (in terms of probability distributions) or functional models. Eventually, optimality criteria and discriminant functions are used to evaluate the performance of such a model given new data.

The algorithms are divided roughly into three different categories: supervised learning, unsupervised learning and reinforcement learning (Bishop, 2006). Supervised learning comprises techniques that predict the value of a target variable y given an input variable x, where x and y might be vectors. A training set of many (x,y) pairs is used to supervise the learning process and to build a model, which is subsequently used to find the target values corresponding to new data points (x_{new},y_{new}). In unsupervised learning the dataset is not labelled, i.e. there is no target variable y, and the aim is to find patterns in the data such that target variables are identified, e.g. using clustering methods. Finally, in reinforcement learning, a certain goal is pursued in a dynamic environment without knowing explicitly whether the approach converges to the goal or not, and the learning process is driven by the feedbacks from the environment.

Machine learning has shown to be very efficient in prediction, for example in solar energy prediction for solar power plants (Sharma et al., 2011). This forecasting task can be reduced to learning how the solar plant reacts to the environmental conditions, and forecasting the future response of the plant using reliable weather data. As such, the methodology can in principle also be directly applied to climate prediction problems (Slingo and Palmer, 2011), such as the prediction of El Niño events (Chen et al., 2004) and of interannual variations of the path of the Kuroshio Current in the North Pacific Ocean (Qiu and Chen, 2005). In particular the occurrence of an El Niño event has large impacts on the weather around the Pacific (Reilly, 2009). It is therefore crucial to develop precise and reliable predictions of such events with considerable lead time and, if so, provide information on how the events could develop in time.

Since the 1990s, both dynamical models and statistical models have been used to predict El Niño events (Latif and Barnett, 1994; Fedorov et al., 2003; Chen et al., 2004; Yeh et al., 2009). Although about 20 models currently provide El Niño forecasts routinely, all reliable forecasts are generally limited to a 6 months ahead horizon. The reason is the so-called *Spring Predictability Barrier*: during spring errors are greatly amplified due to the coupled feedbacks in the equatorial ocean-atmosphere system (Goddard et al., 2001; Duan and Wei, 2013). Moreover, the prediction skill for the development of El Niño events is still disappointing for the current models as can be seen by following the 2015 El Niño development at *http://www.cpc.ncep.noaa.gov/products/ analysis_monitoring/enso_advisory/ensodisc.html*.

Recently, approaches from complex network theory have been applied to problems in climate dynamics and shown that spatial-temporal information on climate variability can be efficiently represented by network measures (Tsonis and Roebber, 2004; Steinhaeuser et al., 2011; Tantet and Dijkstra, 2014; Fountalis et al., 2015). The central two elements of this approach are Climate Network (CN) reconstruction and subsequent network analysis (Tsonis and Swanson, 2006; Yamasaki et al., 2008; Donges et al., 2009). A notion of connectedness (defining a 'link' in the network) between time series at different locations (the 'nodes' in the network) can be obtained by considering their Pearson correlation. Software packages, such *pyunicorn* (Donges et al., 2015) and *Par@graph* (Ihshaish et al., 2015), are now available for efficient climate network reconstruction and analysis.

Complex-network based indicators of El Niño occurrences have been developed using climate networks for example reconstructed from atmosphere surface temperature observations (Yamasaki et al., 2008; Gozolchiani et al., 2011). These studies have shown that links based on the spatial correlations of the temperature anomalies tend to weaken significantly during El Niño events. A large-scale cooperative mode, linking the El Niño basin and the rest of the Pacific climate system builds up one calendar year before the warming event (Ludescher et al., 2013). Based on such findings on the temporal evolution of the CN, Ludescher et al. (2014) developed a forecasting scheme for El Niño events. They suggest that a threshold on the average link weight in the reconstructed CN can reliably forecast an El Niño event one year ahead.

When machine-learning techniques are applied to the prediction of climate variability using data from CNs, one typical task is to infer or 'learn' the dynamics of the climate system from past states and predict its future states. In this paper, we present a machine-learning approach for climate forecasting using the measures of CNs. The originality and advantage of this approach is that the temporal information is already contained in the measures of the CNs, so the machine-learning techniques will take those into account when making predictions of the future states of the system. This is a big advantage that is not that common in most of the applications where machine learning is used for prediction. In section 2, we start with an explanation on how the data for the machine-learning approach is obtained from complex network analysis. The machine-learning methodology itself is described in section 3 and subsequently applied in section 4 to the prediction of El Niño events. A summary and discussion are given in section 5.

2 Climate Networks

Climate scientists have been long interested in studying the statistical correlations between observables for gaining a good understanding of the large-scale development of the climate system. By investigating the correlation structures of global or regional fields, such as surface air temperature and geopotential height, much insight is gained into the patterns of climate variability. For example, through such analyses, the Southern Oscillation was discovered by Sir Gilbert Walker and also its relation with the equatorial Tropical sea surface variability, i.e. El Niño, was clarified (Katz, 2002).

Suppose that a certain climate system observable indicated by O below, such as sea surface temperature (SST) or surface atmospheric temperature (SAT), is available at fixed measurement stations, certain predefined regions, or at grid cells (e.g. from observations, proxy reconstructions, reanalysis, or model simulations). The corresponding data can be represented by an $n \times N$ matrix F, ordered in such a way that each column vector $\mathbf{O}_i = (O_i(t_1),\cdots,O_i(t_n))^T$ at each grid point i $(i = 1,\ldots,N)$ contains a time series of length n.

As mentioned above, one way to define the links in the climate network is to use the Pearson Correlation, defining a PCCN, or the Mutual Information, defining a MICN (Feng and Dijkstra, 2014). To reconstruct a PCCN, first the linear Pearson correlation coefficient between the time series at two grid points i and j is determined. The elements R^P_{ij} of the correlation matrix R^P are given by

$$R^P_{ij} = \frac{\sum_{k=1}^{n} O_i(t_k) O_j(t_k)}{\sqrt{(\sum_{k=1}^{n} O_i^2(t_k))(\sum_{k=1}^{n} O_j^2(t_k)))}}. \tag{1}$$

To reconstruct a MICN , the correlation between the time series of two grid points i and j is determined by the nonlinear mutual information coefficient, giving

$$R^M_{ij} = \sum_{x \in \mathbf{O_i}} \sum_{y \in \mathbf{O_j}} p(x,y)\; log(\frac{p(x,y)}{p(x)p(y)}), \tag{2}$$

where $p(x,y)$ is the joint probability density function of events x and y and $p(x)$ and $p(y)$ are the marginal probability density functions.

We consider that two nodes i and j have an unweighted link, if the absolute value of their correlation coefficient R_{ij}^X (either $P = X$ or $X = M$) is larger than a certain threshold value τ. All links are then represented by an $N \times N$ adjacency matrix A, which can be determined from the correlation matrix R^X according to

$$A_{ij} = \mathcal{H}(|R_{ij}^X| - \tau), \tag{3}$$

where $\mathcal{H}$ is the Heaviside function. The threshold τ is in most cases based on statistical significance (say above the 95% level) of the correlations between the time series (Donges et al., 2015).

Another way to define a link between nodes i and j was presented in Gozolchiani et al. (2011) and also used in Ludescher et al. (2014). First, the cross-correlation function $C_{ij}(\Delta t)$ between the time series at locations i and j is calculated, where Δt is a positive time lag and $C_{ij}(\Delta t) = C_{ji}(-\Delta t)$. Next, the time lag Δt^* at which $C_{ij}(\Delta t)$ is maximal (or minimal) is determined. Finally, weights ($W_{ij}^{\pm}$) for positive and negative links are defined as:

$$W_{ij}^+ = \frac{\mathrm{MAX}(C_{ij}) - \mathrm{MEAN}(C_{ij})}{\mathrm{STD}(C_{ij})}, \tag{4}$$

and

$$W_{ij}^- = \frac{\mathrm{MIN}(C_{ij}) - \mathrm{MEAN}(C_{ij})}{\mathrm{STD}(C_{ij})}, \tag{5}$$

where MAX and MIN are the maximum and minimum values and MEAN, STD are the mean and standard deviation, respectively. In this way, a weighted and directed link between nodes i and j is obtained (Gozolchiani et al., 2011; Wang et al., 2013).

There are many other ways to reconstruct climate CNs and an overview is given in Donges et al. (2015). By reconstructing CNs, the correlations in time series of observables at different locations is represented with a graph, defined by its adjacency matrix A. Subsequently, many topological properties of this graph are analysed, such as the degree d_i of each node i, given by

$$d_i = \sum_{j=1}^{N} A_{ij} \tag{6}$$

which is the total number of links that a node possesses. Next step is to use the properties of such a CN as the input of machine-learning techniques. Besides the statistical properties of the CN, such as those of the correlation matrices, also the topological properties of the graph can be used.

3 Machine learning approach

In `ClimateLearn`, supervised learning approaches are implemented for the prediction of climate variability. Specifically we focus on multilayer artificial neural networks (ANN) and symbolic regression with genetic programming (GP), both explained in this subsection. The approaches follow the typical outline of machine learning: the algorithms are trained or ‘learn’ a certain behavior from the data This results in a model that is evaluated using test data, which are different from the ones used for training.

3.1 Artificial neural networks

Artificial neural networks are a class of statistical learning models inspired by the physiology of biological neural networks. They consists of a network of computing units, the neurons, which process input information transforming it in an output signal whose form depends on the network internal state. Their importance has increased due to recent availability of software capable to efficiently train the network and allow to use these methods for a large variety of problems, from speech and image recognition up to the forecasting of time series and high-dimensional clustering. Many neural network topologies have been proposed in the literature, as specific problems require specific topologies to be solved efficiently. Here, we concentrate on a specific configuration known as *multilayer perceptron* or multilayer neural network. In Fig. 1a the typical structure of a multilayer perceptron is shown: the inputs enter the network and are processed by one or more hidden layers and exit at an output layer. Therefore the computation can be seen as a mapping operation from a n-dimensional input vector to a m-dimensional output vector. In a multilayer perceptron, information travels from the input to the output layer because the neuron connections are chosen to be unidirectional. When all neurons of one layer are connected to all the following we speak about a fully connected multilayer perceptron.

Each neuron performs a specific kind of computation and Fig. 1b shows the functionality of the Rosenblatt *perceptron*. First, a weighted sum of the input variables and the bias term b is built, with the result being then processed by an activation function $f(t)$. Once the single neuron operation is specified, one can easily calculate the network outputs given an input vector by evaluating the output of each layer by forward input propagation. The result is a function of the network configuration, i.e. its topology and the value of the connection weights. It will be the job of the training phase to *learn* the weights in order to induce the desired computation; training and learning are used here as synonymous.

Since the advent of neural networks, the training phase has been considered a computationally demanding problem mainly because of the absence of efficient algorithms relative to the available computing power. This has been overcome by the back-propagation algorithm (Bishop, 2006), nowadays widely applied in training multilayer perceptrons. Given a supervised training set $\{\boldsymbol{x}_i,\boldsymbol{t}_i : i=1...N\}$ with $\boldsymbol{x}_i$ input variables and $\boldsymbol{t}_i$ target variables, we denote by $\boldsymbol{y}_i$ the correspondent output computed by the network when $\boldsymbol{x}_i$ is fed forward. In general we have $\boldsymbol{t}_i \neq \boldsymbol{y}_i$. A global error on the training set can be then defined as a quadratic function of the form

$$E(\boldsymbol{w}) = \frac{1}{2N}\sum_i ||\boldsymbol{t}_i - \boldsymbol{y}_i||^2 \tag{7}$$

and can be seen as a function of the network weights $\boldsymbol{w}$. Other error definitions are possible, for example by choosing a different norm. The idea behind back propagation is to minimize this error by updating the weights using the gradient descend method (with k as iteration index), i.e.

$$w_{ij}^{(k)} \rightarrow w_{ij}^{(k)} - \alpha \frac{\partial E(\boldsymbol{w})}{\partial w_{ij}^{(k)}} \tag{8}$$

The calculation of the partial derivatives is thus crucial for the algorithm. It is done by using directly the dependence of the error function on the training set instances. When all the instances have been used, one ‘epoch’ of training is completed. Usually many epochs of training are needed in order for the error function to converge to a local, or global minimum, resulting in longer training periods.

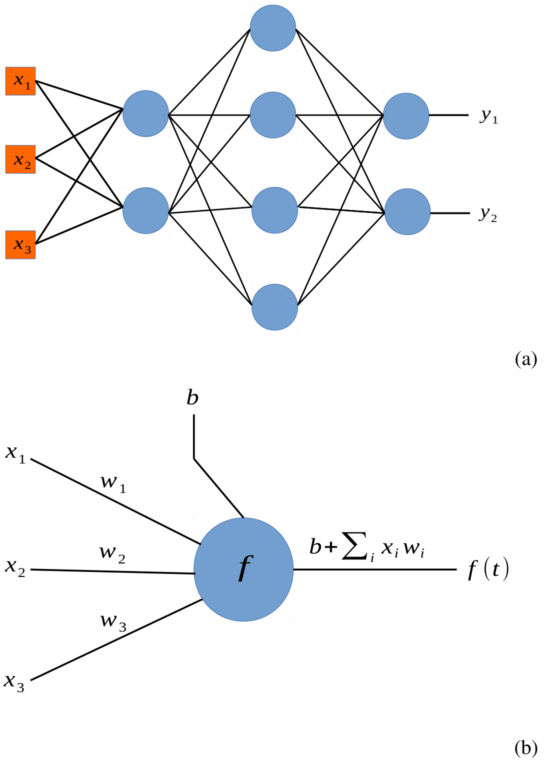

Fig. 1: *(a) Schematics of a fully connected multilayer perceptron with three input variables x_1,x_2 and x_3 and two output variables y_1 and y_2, with two hidden layers. (b) The Rosenblatt perceptron, with three inputs and a bias unit. The weighted input sum is added to the bias term and then enters as argument of the activation function f which generates the neuron output.*

Once training is completed, the model is tested by checking the error on a test set. One main concern in this procedure is to avoid overfitting of data, i.e. a model that adapts too much on the training data and may not generalize well when new data are used. In order to minimize the risk of overfitting one can employ cross-validation methods, which consist in providing several bi-partitions of the training set, a training partition and a validation partition. The network is then trained on the training partition and tested on the validation partition. Once this process has been performed on several cross-validation partitions and the statistics of the training and validation errors are examined, the quality of the model can be established. If no signs of overfitting are detected, the training is considered successful and the network can be employed for generalization.

3.2 Genetic programming

Genetic programming and genetic algorithms are a class of evolutionary based algorithms whose principles are based on Charles Darwin's Theory of Evolution (Darwin, 1959). In this masterpiece, Darwin explains that, given a population of individuals living within an environment, only a subset of them are properly fitted and therefore have higher chances of survival and reproduction. New generations may inherit these favourable genetic characteristics and they will end up being dominant inside the population. Variations in the individual characteristics can be classified in three categories. In the first category are variations that are damageable for the individual. To the second belong the beneficial ones and in the third category, the variations have no effective influence. Natural selection consists in the preservation of beneficial characteristics and their transmittance to the next generation, since those fitted individuals live longer and most of the time are better able to beat the competition for reproduction.

Given a problem P to be solved, we imagine to have an ensemble of solutions S to this problem. According to its efficiency in solving P, each solution can be considered as an individual for which the degree of adaptation to its environment can be measured in terms of a *fitness value*. Genetic Programming (GP) is a particular case where the evolved individuals are computer programs (Koza, 1993). The aim is to appropriately evolve computer programs by creating new generations, evaluating their fitness value and finally selecting the best program that solves the problem at hand. Here, we restrict such programs to functions $f(x_1,\ldots,x_n)$ of a given number of variables $x_i, i = 1,\ldots n$ and we aim at finding a given function that approximates the solution to our problem accurately enough. Therefore our application is nothing more than a symbolic regression achieved through genetic programming algorithms. The fitness values can be represented mathematically by a real valued functional $F[f(x_1,\ldots,x_n)]$, mapping the space of possible solutions onto the real axis. In GP, the programs are typically represented as trees, where each tree represents an expression of a potential solution to a problem (cf. examples in Fig. 2).

To implement the variations in genetic programming, two operators are commonly used: mutation and crossover. Their behaviour is very similar to the biological mutation and crossover concepts. In a mutation step, a random node in the tree that represents the individual is selected and the corresponding subtree is replaced randomly by another one (Fig. 2a). Mutation is very important to keep diversity inside the population, and diversity helps the algorithm to explore all the search space and preventing encounters of local maxima of the fitness functional. Crossover is based on the exchange of characteristics among two individuals. In GP, this is implemented by selecting randomly a node in two trees that represent two individuals and exchange the two subtrees attached to two those nodes (Fig. 2b). In this way, the two individuals inherit characteristics of both *parent* trees.

In a so-called symbolic regression scheme, an initial random population of individuals is generated randomly using elementary functions taken from a *function set* (e.g. $\sin x, \exp\, x, \max[x_1,x_2]$) and combined using a certain algebraical or functional operations (e.g. $+,-,*$) in an *operation set*, whose choice depends on the problem at hand. The initial population is then evaluated according to a fitness function. The genetic operations mutation and crossover are then applied on the population in order to create a second generation which is as well evaluated. This process continues until a prescribed termination condition, for example when the maximum number of generations has been reached, or an individual of high enough fitness is found (e.g. when the absolute value of the functional F is smaller than a certain tolerance).

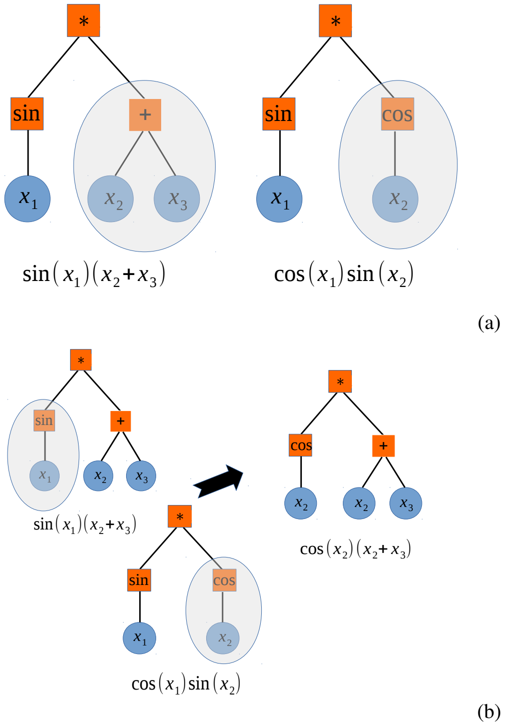

Fig. 2: *(a) Example of mutation operation: a branch of the tree on the left is changed (mutated) by substitution with another compatible branch determined randomly. (b) Example of crossover: two individuals are selected from the population and a new individual is created by mixing the highlighted branches.*

4 Application: El Niño variability

As an example of the application of `ClimateLearn`, we consider the forecasting of El Niño events using two different approaches. First, we focus on the forecasting of the occurrence of events, i.e. the presence (or not) of an El Niño in a given interval of time regardless of the intensity of the phenomenon. This problem can be considered as a classification problem, where a set of discrete classes is the output of the model (section 4.1). The second approach is the forecasting of the time evolution of a scalar characteristic of El Niño, where we aim at the prediction of a real-valued time series by regression. The results in this case will give information on both the presence and intensity of the event (section 4.2). In section 4.3, we provide specific results for the occurrence and development of the El Niño conditions in the year 2014.

4.1 Results: Occurrence of El Niño events

Just as in Ludescher et al. (2014), the data consist of atmospheric surface temperature anomalies over the May 1949 - March 2014 from the NCEP Reanalysis project (Kalnay et al., 1996). From this dataset, a directed, weighted network was reconstructed (Gozolchiani et al., 2011; Ludescher et al., 2014) using the methodology presented in section 2. Several measures $x_i, i = 1,\ldots,N$ of this network are used as the attributes in the machine-learning approach and for each quantity a time series $(x_i^1,\ldots,x_i^T)$ is available. We use a time interval of 10 days (which gives $T = 2365$) and choose eight measures (with time included, $N = 9$). These eight measures are the maximum correlation $\mathrm{MAX}(C_{ij})$, the minimum correlation $\mathrm{MIN}(C_{ij})$, the maximum delay $\mathrm{MAX}(\Delta t^*)$, the minimum delay $\mathrm{MIN}(\Delta t^*)$, the maximum link weight $\mathrm{MAX}(W_{ij})$, the minimum link weight $\mathrm{MIN}(W_{ij})$, the standard deviation of the correlation $\mathrm{STD}(C_{ij})$, and the mean correlation $\mathrm{MEAN}(C_{ij})$ (see section 2).

The target variable is discrete valued and distinguishes the presence or absence of an event. Operationally (http://www.cpc.ncep.noaa.gov/), an El Niño event is said to occur when the sea surface temperature anomaly over the region 120°W-170°W × 5°S-5°N, the so-called NINO3.4 index (the SST anomaly averaged over the region [120°W-170°W]×[5°S-5°N]), is above the threshold of $+0.5$°C for at least 5 consecutive months. Hence, we put $y = 0$ (no event) when it belongs to a interval of time where El Niño is not present, and $y = 1$ when it is present. Here we do not want to smooth the data and hence we flag an El Niño event when NINO3.4 values are continuously above the threshold of $+0.4$°C for five months. Regarding the build of the training and test sets, the condition $t_1^{test} > t_T^{train}$ has to be satisfied. This means that the instances in the test set happen after the one in the training set, since we are only interested in a chronological prediction for the El Niño event.

The method we choose for the supervised learning is an artificial neural network (ANN) with a 3×3 layer structure (3 neurons per layer). The training set is from May 1949 to June 2001 (80% of T), the test set is from June 2001 to March 2014 (20% of T). Similar to Ludescher et al. (2014), the prediction lead time τ is 12 months. Fig. 3a shows the classification results on the test set, where 1 stands for the occurrence of an El Niño event and 0 means absence. The result is then filtered by eliminating the isolated and transient events, and by batching the adjacent events together. Fig. 3b then shows that our forecasting scheme gives accurate alarms 12 months ahead for the El Niño events in 2002, 2006 and 2009, and no alarm in 2004. Compared with the results in Ludescher et al. (2014), the machine-learning toolbox enables us to give a better prediction for the occurrence of El Niño events when using more measures of the same CN.

One advantage of using supervised learning for prediction is that the predictor model is constructed automatically from the training set without subjective decisions like the choice of thresholds. However, because the available data for prediction as well as the amount of instances is limited, (for example, only a few El Niño events occurred between May 1949 and March 2014), the accuracy of the prediction will mostly depend on the length of the training set. Consequently we need to choose proper proportions of the available data as the training/test set to avoid 'under training'. To demonstrate that the current proportion for the test set (20% of T) gives the best performance, we conduct a Receiver Operating Characteristic (ROC)- type analysis by varying the proportion from 16% to 30% of T as the test set. With a proportion between 16% and 20%, the averaged hit rate $D = 0.90$ and the averaged false-alarm rate $\alpha = 0.10$. For 21% to 25%, we find $D = 0.71$ and $\alpha = 0.29$, and for 26% to 30%, $D = 0.21$ and $\alpha = 0.79$. Thus, to have a higher hit rate and a lower false-alarm rate, the best proportion for

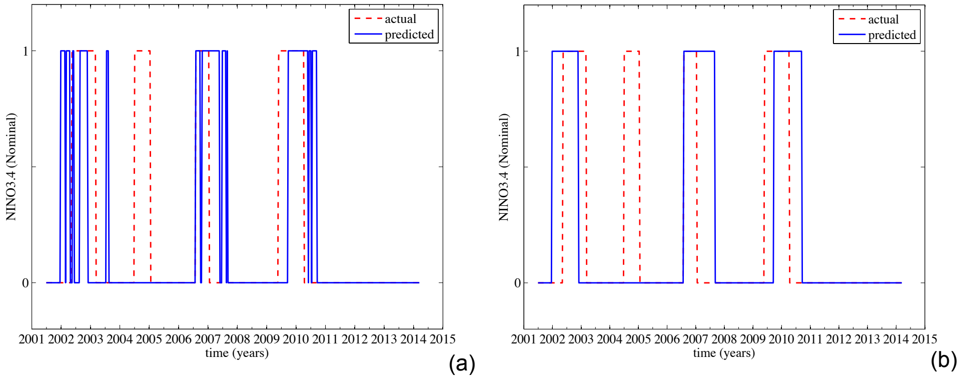

Fig. 3: *Prediction results on the test set from June 2001 to March 2014 (a) without filtering and (b) with filtering, using an artificial neural network (ANN) with a* 3×3 *layer structure (3 neurons per layer) for a 12 months lead time prediction for the occurrence of El Niño events. The red dashed lines are the actual nominal quantity of the NINO3.4 index (*1 *stands for the occurrence of an El Niño event where NINO3.4 values are continuously above the threshold of* $+0.4°C$ *for five months, and* 0 *for the absence of such an event), and the blue solid lines indicate the predicted ones.*

the test set is $\leq 20\%$. Of course, we should also maximize the length of the test set to incorporate more El Niño events for testing, and this motivated our choice of 20% (Fig. 3).

4.2 Results: NINO3.4 index development

Predictions for the development of the NINO3.4 index are more difficult than those for the occurrence of El Niño events. For example, consider the results of the CFS version 2 (CFSv2) model developed by the Environmental Modeling Center at National Centers for Environmental Prediction (NCEP). This is a fully coupled model representing the interaction between the Earth's atmosphere, oceans, land and sea ice (Saha et al., 2014). In August 2014 this model predicted that the NINO3.4 index would go over $+1.0°$C in October 2014 but the actual value in October 2014 was just around $+0.5°$C. Hence even for short term predictions (up to few months) a good skill of the NINO3.4 index development is still hard to achieve by this model.

Short-term development of the NINO3.4 index is strongly related to the stability of the Pacific background state and the occurrence of westerly wind bursts (WWBs) near the dateline. In Feng and Dijkstra (2015), PCCNs were reconstructed using sea surface temperature data from the HadISST dataset Rayner (2003) using the methodology presented in section 2. As a measure of the coherence in the PCCN, they determine the number of links of each node, i.e. the degree of the node. As a measure of the stability of the Pacific climate, they use the skewness S_d of the degree distribution of the PCCN. In addition to S_d, also the time series of the second principal component (PC2) of the wind stress residual (the signal due to SST variability is filtered out) is used as a measure the WWB strength (Feng and Dijkstra, 2015).

Next, we use the machine-learning toolbox to investigate the importance of S_d and PC2 for the NINO3.4 index development by supervised learning regression. The attributes are therefore the background stability index $x_1 = S_d$, the westerly wind burst measure x_2 = PC2 and the time $x_3 = t$ from November 1961 to October 2014 with a time interval of one month (i.e., $T = 636$ and $N = 3$). Given the data set we again have to choose a training and a test set. In the case of regression we can randomly choose a given percentage of the instances to belong to a training set and the rest to a test set. Since we do not possess a large amount of data, it is however important that these two dataset are as homogeneous as possible in order to avoid overfitting issues.

The training set chosen is from November 1961 to April 2004 (80% of T) and the test set is from May 2004 to October 2014 (20% of T). The quality of the predicted results in the test set is measured by the normalized root mean squared error (NRMSE) defined as

$$NRMSE(y_A, y_P) = \frac{1}{\max(y_A) - \min(y_A)} \sqrt{\frac{\sum\limits_{t_1^{test} \le t_k \le t_n^{test}} (y_A^k - y_P^k)^2}{n}}, \quad (9)$$

where y_A^k is the actual time series of NINO3.4 index, the predicted is indicated by y_P^k,n and n is the number of points in the test set.

We first employ an ANN with a 2×1 layer structure (2 neurons in the first layer and 1 neuron in the second one) to do the regression. Since we do not know the optimal prediction time τ which would give a reasonable prediction $y(t+\tau)$ at time $t+\tau$, we vary τ from 1 month up to 12 months. Fig. 4 shows the regression results on the test set for the 2-4 months lead time NINO3.4 forecasts. The best prediction, with the smallest value of NRMSE=0.18, is given at $\tau = 3$ months (Fig. 4b).

To test the robustness of the regression result for the three months lead time NINO3.4 index forecast (Fig. 4b), we perform a series of cross-validations by keeping specific percentage splits between training set and test set (70-30, 75-25, 80-20, and 85-15), but randomly choosing 200 initial times t_1^{test} of the test set from November 1961 to October 2014 for each percentage split. From Fig. 5, one can see that the peak values of the NRMSE remain near 0.17, independent of the choices of the percentage splits and t_1^{test}. Therefore the regression result in Fig. 4b is considered robust.

Due to the irregular behavior of the PC2 representing the WWBs (cf. Figure 3 in Feng and Dijkstra (2015)), the predicted NINO3.4 indices in Fig. 4 show more fluctuations than the actual one. When a 3-month running mean is applied to the predicted NINO3.4 index (three months lead time, Fig. 4b) as well as the actual one, the forecast has a better skill (NRMSE=0.14) as shown in Fig. 6a. To further demonstrate that the result in Fig. 6a is robust and independent of the choices of the ANN layer structures and the methods for the supervised learning, we perform the same regression task with an ensemble of 49 ANNs with different binary layer structures and up to 7 neurons per layer and an ensemble of 50 GP runs. The averaged result of the best 10 ANNs (with the smallest NRMSE values) is shown in Fig. 6b with NRMSE=0.15. The averaged result of the best 10 GP runs (having the smallest regression error) is shown in Fig. 6c with NRMSE=0.17, which are both similar to the one obtained by the ANN with a 2 × 1 layer structure in Fig. 6a.

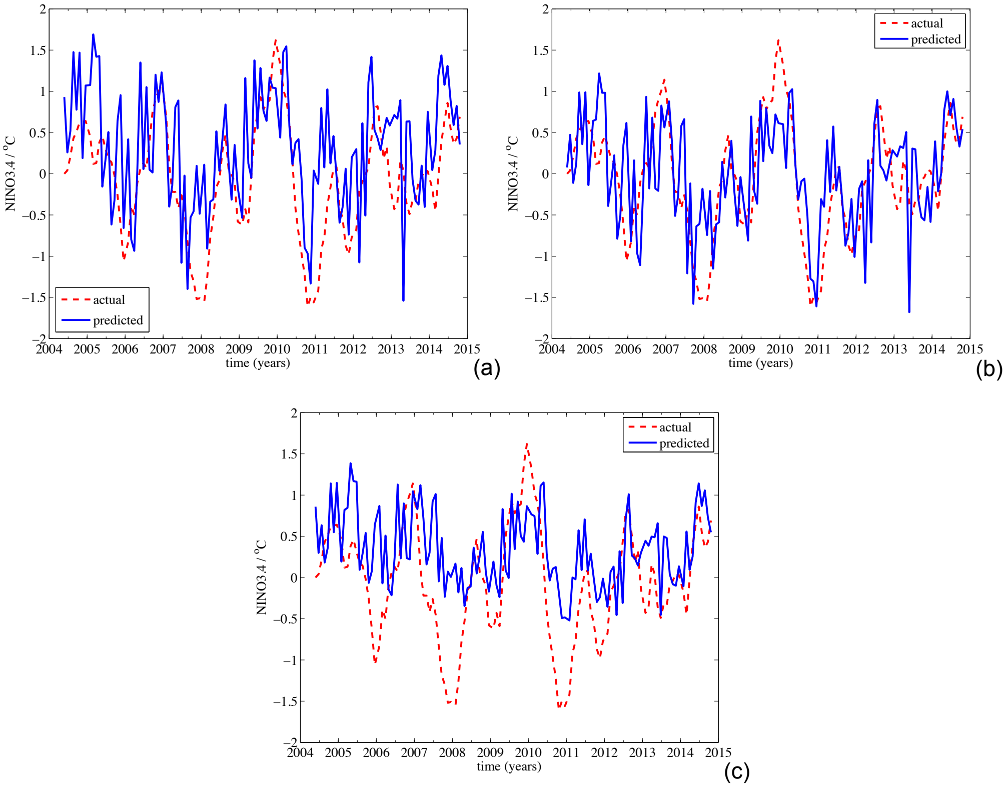

Fig. 4: *Regression results on the test set from May 2004 to October 2014 using an ANN with a* 2×1 *layer structure (2 neurons in the first layer and 1 neuron in the second one) for the prediction of the NINO3.4 index with a lead time of (a) 2 months (NRMSE=*0.23*), (b) 3 months (NRMSE=*0.18*), and (c) 4 months (NRMSE=*0.22*). The red dashed lines are the actual values of NINO3.4 index, and the blue solid lines indicate the predicted ones.*

4.3 Results: El Niño development in 2014

In the previous sections we have seen that by using the measures of the CNs from Ludescher et al. (2014) and Feng and Dijkstra (2015), the machine-learning toolbox `ClimateLearn` can give robust predictions on the occurrence of El Niño events one year ahead and the development of NINO3.4 index with a lead time of three months, respectively. We now apply these techniques to the occurrence and development of the situation in 2014.

First, we consider the occurrence of an El Niño event up to March 2015, by using the same data used in section 4.1 till March 2014. The prediction results on El Niño occurrence in 2014 are shown in Fig. 7a, by employing an ensemble of 36 ANNs with different binary layer structures and up to 6 neurons per layer. Like the event in 2012 in Fig. 3b, our forecast scheme tends to ignore the ENSO-neutral favored events or the weak El Niño events. Hence, no El Niño event between January 2014 to March 2015 is predicted one year ahead (Fig. 7a).

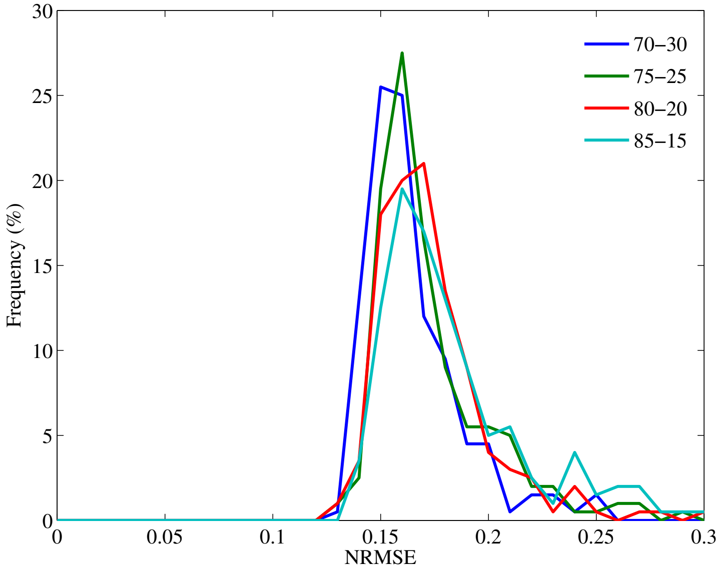

Fig. 5: *Cross-validation results of NINO3.4 index forecast on the test set by keeping certain percentage splits between training set and test set (70-30, 75-25, 80-20, and 85-15), but randomly choosing 200 initial times of the test set t_i^{test} from November 1961 to October 2014 for each percentage split. The blue dashed curve is the NRMSE distribution of 70-30 split (70% of T as the training sets and 30% of T as test sets), the green solid line for a 75-25 split, the red solid curve for a 80-20 split and the cyan solid curve for a 85-15 split.*

Second, we consider the development of the NINO3.4 index from January 2014 till January 2015 using the same data used in section 4.2 till October 2014. The accuracy of the predicted NINO3.4 index over 2014 with a lead time of three months (Fig. 7b) is quite good (NRMSE=0.19) for example compared with the one given by CFSv2 model over that period (NRMSE=0.34).

5 Summary and discussion

In this paper, we have presented the machine-learning toolbox `ClimateLearn` for climate prediction problems, based on climate data obtained from complex network reconstruction and analysis. Besides handing multivariate data from these networks and other sources, another advantage of using this machine-learning toolbox for climate variability prediction is that the development of predictor models is dynamic and data-driven (Bishop, 2006). Using machine-learning techniques with the measures from reconstructed Climate Networks (CNs), we have provided novel prediction schemes for the occurrence of an El Niño event (with a lead time of one year) and for the development of the NINO3.4 index (with a lead time of three months).

By using measures of a directed and weighted CN (Ludescher et al., 2014) and supervised learning classifi-

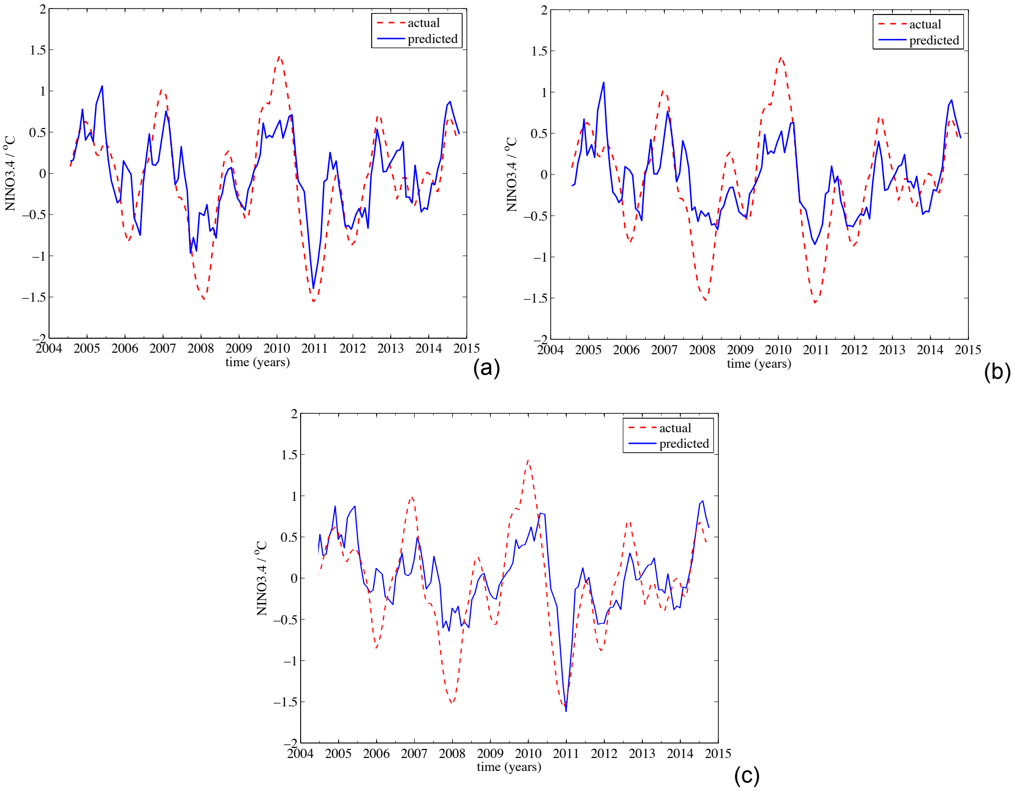

Fig. 6: *Results for a 3-month running mean regression on the test set from May 2004 to October 2014 using (a) an ANN with a* 2×1 *layer structure (2 neurons in the first layer and 1 neuron in the second one, NRMSE=*0.14*), (b) an ensemble of 49 ANNs with different binary layer structures and up to 7 neurons per layer (only the ensemble mean of the best 10 is showed, NRMSE=*0.15*) and (c) an ensemble of genetic programmings (only the ensemble mean of the best 10 is showed, NRMSE=*0.17*) for the three months ahead prediction for the development of the NINO3.4 index. The red dashed curves are the actual values of NINO3.4 index, and the blue solid curves indicate the predicted ones.*

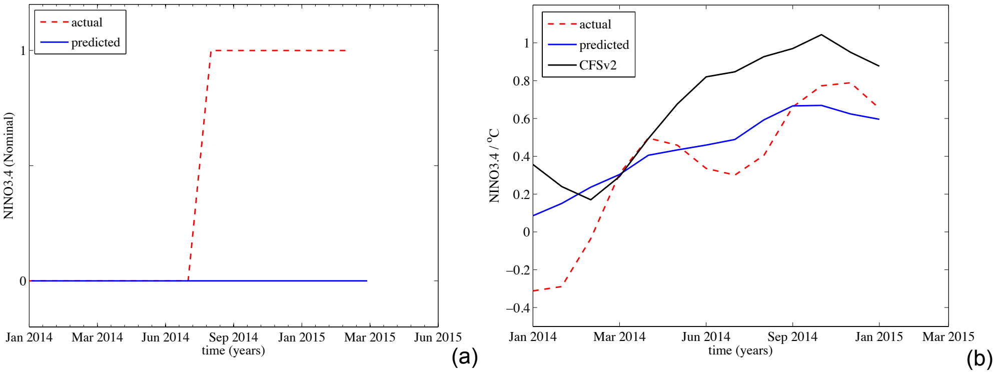

Fig. 7: *Prediction results on ENSO variability in 2014 using an ensemble of 36 ANNs with different binary layer structures and up to 6 neurons per layer. (a) The occurrence of the El Niño event given one year ahead, and (b) the development of NINO3.4 index with a three months a lead time (only the ensemble mean is shown, NRMSE*=0.19*). The red dashed lines are the actual nominal quantity/actual values of NINO3.4 index, the blue solid lines indicate the predicted ones, and the black solid line indicates the predicted one by CFSv2 model (only the ensemble mean is shown, estimated from http://www.cpc.ncep.noaa.gov/products/people/wwang/cfsv2_fcst_history/).*

cation, we developed a forecast scheme in predicting the occurrence of an El Niño event one year ahead. This scheme apparently does not seem to suffer from the 'spring predictability barrier' (Goddard et al., 2001). This is probably due to the fact that the network measures adequately capture the changes of spatial patterns one calendar year before the warming event (Ludescher et al., 2013). Apparently, the prediction schemes can well represent the nonlinear relationships among the attributes and give an objective prediction. For example, in the forecast scheme proposed by Ludescher et al. (2014), the prediction may be sensitive to the choice of the decision threshold θ. Moreover, the false alarms and the misses (the El Niño events in 2006 and 2009 are not detected) show the limitations of their scheme. These deficiencies may be caused by the fact that this forecast scheme is based only on one single measure of the CN. The supervised learning method in our forecast scheme does not have these problems.

In addition, by using measures of an undirected and unweighted CN (Feng and Dijkstra, 2015) that monitor the stability of the Pacific climate state and a measure of the atmospheric wind-stress noise in combination with supervised learning regression, we provided reasonable forecasts of the development of the NINO3.4 index three months ahead. A lead time of three months is of course too short to make this forecast scheme outcompete existing ones. However, comparing these forecast results with those from much more sophisticated models like the CFSv2 model indeed confirm that the quantities S_d and PC2 are important factors in the development of El Niño events.

The software package `ClimateLearn` is written in python 2.7, and it makes full use of the open source packages Weka (available at http:/www.cs.waikato.ac.nz/ml/weka/) and ECJ (available at https://cs.gmu.edu/~eclab/projects/ecj/). The package `ClimateLearn` allows basic operations of data mining, i.e. reading, merging, and cleaning data, and running machine learning algorithms. Building on the success of complex network approaches to investigate aspects of climate variability, `ClimateLearn` provides an innovative and convenient way to predict the occurrence and development of El Niño events. It can also be directly applied to the prediction of other climate variability phenomena.

Code availability

`ClimateLearn` is available through github at https://github.com/Ambrosys/climatelearn. The package is still in a raw version and we plan however to refine it by a full python implementation using other open source third-party libraries (e.g. Deap and Pybrain) in the near future.

Acknowledgements. The authors would like to acknowledge the support of the LINC project (no. 289447) funded by EC's Marie-Curie ITN program (FP7-PEOPLE-2011-ITN). The computations on the Cartesius machine were funded by the Exact Sciences division of the Netherlands Organization of Scientific Research under grant SH-286.

References

Bishop, C. M.: Pattern recognition and machine learning, Springer, New York, 2006.

Chen, D., Cane, M. A., Kaplan, A., Zebiak, S. E., and Huang, D.: Predictability of El Niño over the past 148 years, Nature, 428, 733–736, 2004.

Darwin, C.: On the origin of species by means of natural selection, New York :D. Appleton and Co., 1959.

Donges, J. F., Zou, Y., Marwan, N., and Kurths, J.: Complex networks in climate dynamics, The European Physical Journal Special Topics, 174, 157–179, 2009.

Donges, J. F., Heitzig, J., Beronov, B., Wiedermann, M., Runge, J., Feng, Q. Y., Tupikina, L., Stolbova, V., Donner, R. V., Marwan, N., Dijkstra, H. A., and Kurths, J.: Unified functional network and nonlinear time series analysis for complex systems science: The pyunicorn packageUnified functional network and nonlinear time series analysis for complex systems science: The pyunicorn package, Chaos, pp. 1–26, 2015.

Duan, W. and Wei, C.: The spring predictability barrier for ENSO predictions and its possible mechanism: results from a fully coupled model, International Journal of Climatology, 33, 1280–1292, 2013.

Fedorov, A., Harper, S., Philander, S., Winter, B., and Wittenberg, A.: How predictable is El Niño?, Bulletin of the American Meteorological Society, 84, 911–919, 2003.

Feng, Q. Y. and Dijkstra, H.: Are North Atlantic multidecadal SST anomalies westward propagating?, Geophysical Research Letters, 2014.

Feng, Q. Y. and Dijkstra, H.: Climate Network Based Stability Index for El Niño Variability, arXiv:1503.05449, 2015.

Fountalis, I., Bracco, A., and Dovrolis, C.: ENSO in CMIP5 simulations: network connectivity from the recent past to the twenty-third century, Climate Dynamics, pp. 1–28, 2015.

Goddard, L., Mason, S. J., Zebiak, S. E., Ropelewski, C. F., Basher, R., and Cane, M. A.: Current approaches to seasonal to interannual climate predictions, International Journal of Climatology, 21, 1111–1152, 2001.

Gozolchiani, A., Havlin, S., and Yamasaki, K.: Emergence of El Niño as an Autonomous Component in the Climate Network, Physical Review Letters, 107, 148 501, 2011.

Ihshaish, H., Tantet, A., Dijkzeul, J. C. M., and Dijkstra, H. A.: Par@Graph: a parallel toolbox for the construction and analysis of large complex climate networks, Geoscientific Model Development, 8, 3321–3331, 2015.

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., et al.: The NCEP/NCAR 40-year reanalysis project, Bulletin of the American meteorological Society, 77, 437–471, 1996.

Katz, R. W.: Sir Gilbert Walker and a connection between El Nino and statistics, Statistical Science, pp. 97–112, 2002.

Koza, J. R.: Genetic programming - on the programming of computers by means of natural selection, Complex adaptive systems, MIT Press, 1993.

Latif, M. and Barnett, T. P.: Causes of decadal climate variability over the North Pacific and North America, Science, 266, 634–637, 1994.

Ludescher, J., Gozolchiani, A., Bogachev, M. I., Bunde, A., Havlin, S., and Schellnhuber, H. J.: Improved El Niño forecasting by cooperativity detection, Proceedings of the National Academy of Sciences, 110, 11 742–11 745, 2013.

Ludescher, J., Gozolchiani, A., Bogachev, M. I., Bunde, A., Havlin, S., and Schellnhuber, H. J.: Very early warning of next El Niño, Proceedings of the National Academy of Sciences, 111, 2064–2066, 2014.

Mitchell, T.: Machine Learning, McGraw-Hill, New York, 1997.

Qiu, B. and Chen, S.: Variability of the Kuroshio Extension Jet, Recirculation Gyre and Mesocale Eddies on decadal time scales, Journal Of Physical Oceanography, 35, 2090–2103, 2005.

Rayner, N. a.: Global analyses of sea surface temperature, sea ice, and night marine air temperature since the late nineteenth century, Journal of Geophysical Research, 108, 4407, doi:10.1029/2002JD002670, http://www.agu.org/pubs/crossref/2003/2002JD002670.shtmlhttp://doi.wiley.com/10.1029/2002JD002670, 2003.

Reilly, B.: Disaster and Human History: Case Studies in Nature, Society and Catastrophe, McFarland, Jefferson, 2009.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H.-y., Iredell, M., et al.: The NCEP climate forecast system version 2, Journal of Climate, 27, 2185–2208, 2014.

Sharma, N., Sharma, P., Irwin, D., and Shenoy, P.: Predicting solar generation from weather forecasts using machine learning, in: Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on, pp. 528–533, IEEE, 2011.

Slingo, J. and Palmer, T.: Uncertainty in weather and climate prediction, Philosophical Transactions Of The Royal Society A-Mathematical Physical And Engineering Sciences, 369, 4751–4767, 2011.

Steinhaeuser, K., Ganguly, A. R., and Chawla, N. V.: Multivariate and multiscale dependence in the global climate system revealed through complex networks, Climate Dynamics, 39, 889–895, 2011.

Tantet, A. and Dijkstra, H. A.: An interaction network perspective on the relation between patterns of sea surface temperature variability and global mean surface temperature, Earth System Dynamics, 5, 1–14, 2014.

Tsonis, A. a. and Roebber, P.: The architecture of the climate network, Physica A: Statistical Mechanics and its Applications, 333, 497–504, 2004.

Tsonis, A. A. and Swanson, K. L.: What do networks have to do with climate?, Bulletin Of The American Meteorological Society, 87, 585–595, 2006.

Wang, Y., Gozolchiani, A., Ashkenazy, Y., Berezin, Y., Guez, O., and Havlin, S.: Dominant imprint of rossby waves in the climate network, Physical review letters, 111, 138 501, 2013.

Yamasaki, K., Gazit, O., and Havlin, S.: Pattern of climate network blinking links follows El Niño events, EPL (Europhysics), 83, 28 005, 2008.

Yeh, S.-W., Kug, J.-S., Dewitte, B., Kwon, M.-H., Kirtman, B. P., and Jin, F.-F.: El Niño in a changing climate, Nature, 461, 511–514, 2009.