Time Series Prediction as a Problem of Missing Values

Antti Sorjamaa and Amaury Lendasse [*]

Helsinki University of Technology - Laboratory of Computer and Information Science
P.O. Box 5400, 02015 HUT - Finland

Abstract. In this paper, time series prediction is considered as a problem of missing values. A new method for the determination of the missing time series values is presented. The new method is based on two projection methods: a nonlinear one (Self-Organized Maps) and a linear one (Empirical Orthogonal Functions). The presented global methodology combines the advantages of both methods to get accurate candidates for prediction values. The methods are applied to a time series competition dataset.

1 Introduction

The presence of missing values in the underlying time series is a recurrent problem when dealing with databases. Number of methods have been developed to solve the problem and fill the missing values. The methods can be classified into two distinct categories: deterministic methods and stochastic methods.

Self-Organizing Maps [1] (SOM) aim to ideally group homogeneous individuals, highlighting a neighborhood structure between classes in a chosen lattice. The SOM algorithm is based on unsupervised learning principle where the training is entirely stochastic, data-driven. No information about the input data is required. Recent approaches propose to take advantage of the homogeneity of the underlying classes for data completion purposes [2]. Furthermore, the SOM algorithm allows projection of high-dimensional data to a low-dimensional grid. Through this projection and focusing on its property of topology preservation, SOM allows nonlinear interpolation for missing values.

Empirical Orthogonal Function (EOF) [3] models are deterministic enabling linear projection to high-dimensional space. They have also been used to develop models for finding missing data [4]. Moreover, EOF models allow continuous interpolation of missing values, but are sensitive to the initialization.

This paper describes a new method, which combines the advantages of both the SOM and the EOF. The nonlinearity property of the SOM is used as a denoising tool and then continuity property of the EOF method is used to recover missing data efficiently.

The SOM is presented in the Section 3, the EOF in Section 4 and the global methodology SOM+EOF in Section 5. Section 6 presents the experimental results using a new competition dataset.

[*] Part the work of A. Sorjamaa and A. Lendasse is supported by the project of New Information Processing Principles, 44886, of the Academy of Finland. The work of A. Lendasse is supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

2 Time Series Prediction

2.1 Data with Missing Values

In time series prediction problem, the samples are generated by sliding a fixed window over the time series and taking each window full of values as a sample. The size of the window and thus the length of the samples is T. All samples are collected to a *regressor matrix*

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_j \end{bmatrix}, \; j = 1, 2, ..., N, \tag{1}$$

where N is the number of samples and each $\mathbf{x}_j$ is a T-dimensional sample vector.

When predicting the future of the time series, the missing values are added to the end of the known values of the time series. Then, logically the regressor matrix is missing some values in the lower right corner. The shape and the size of the area of the missing values depend on the used method and the horizon of prediction.

2.2 Prediction Strategy

There are three prediction strategies for the long-term prediction of time series that are mainly used. The first and the least calculation intensive is the *Recursive* prediction strategy, where the model selected in the learning phase for the first time step is used repeatedly, or recursively, as far as necessary. The predicted values are used as known values and the prediction is done always only one step at a time.

The next alternative is to use different model to predict each time step. This *Direct* prediction strategy needs different model for each time step and is therefore many times more calculation intensive. In many cases the Direct is still appealing choice, because of the increased accuracy compared to the Recursive strategy. Where the Recursive strategy suffers from accumulation of prediction errors, the Direct does not.

Third alternative is to use a mix of the two, called *DirRec* prediction strategy [5]. With this prediction strategy different model is trained for each time step and all predicted values are used as known values in the process. It means that the regressor is increased by one in every time step when the previous prediction is included in the learning data. This increases the calculation time in the learning process but in many cases, the accuracy is also better.

In this case, when the time series prediction is considered as a missing value problem, the whole set of values to be predicted is estimated at once. Strictly speaking the strategy used here is none of the above, but instead *all-at-once* strategy.

3 Self-Organizing Map

The SOM algorithm is based on an unsupervised learning principle, where training is entirely data-driven and no information about the input data is required [1]. Here we use a 2-dimensional network, compound in c units (or code vectors) shaped as a square *lattice*. Each unit of a network has as many weights as the length T of the learning data samples, $\mathbf{x}_n$, $n = 1, 2, ..., N$. All units of a network can be collected to a weight matrix $\mathbf{m}\,(t) = [\mathbf{m}_1\,(t)\,, \mathbf{m}_2\,(t)\,, ..., \mathbf{m}_c\,(t)]$ where $\mathbf{m}_i\,(t)$ is the T-dimensional weight vector of the unit i at time t and t represents the steps of the learning process. Each unit is connected to its neighboring units through neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$, which defines the shape and the size of the neighborhood at time t. Neighborhood can be constant through the entire learning process or it can change in the course of learning.

Learning starts by initializing the network node weights randomly. Then, for randomly selected sample $\mathbf{x}_{t+1}$, we calculate a Best Matching Unit (BMU), which is the neuron whose weights are closest to the sample. BMU calculation is defined as

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \left\{ \| \mathbf{x_{t+1}} - \mathbf{m}_i\,(t) \| \right\}, \tag{2}$$

where $I = [1, 2, ..., c]$ is the set of network node indices, BMU denotes the index of the best matching node and $\|.\|$ is standard Euclidean norm.

If the randomly selected sample includes missing values, the BMU cannot be solved outright. Instead, an adapted SOM algorithm, proposed by Cottrell and Letrémy [6], is used. The randomly drawn sample $\mathbf{x_{t+1}}$ having missing value(s) is split into two subsets $\mathbf{x}_{t+1}^T = NM_{\mathbf{x_{t+1}}} \cup M_{\mathbf{x_{t+1}}}$, where $NM_{\mathbf{x_{t+1}}}$ is the subset where the values of $\mathbf{x_{t+1}}$ are not missing and $M_{\mathbf{x_{t+1}}}$ is the subset where the values of $\mathbf{x_{t+1}}$ are missing. We define a norm on the subset $NM_{\mathbf{x_{t+1}}}$ as

$$\| \mathbf{x_{t+1}} - \mathbf{m}_i\,(t) \|_{NM_{\mathbf{x_{t+1}}}} = \sum_{k \in NM_{\mathbf{x_{t+1}}}} \left(\mathbf{x}_{t+1,k} - \mathbf{m}_{i,k}(t) \right)^2, \tag{3}$$

where $\mathbf{x}_{t+1,k}$ for $k = [1, ..., T]$ denotes the k^{th} value of the chosen vector and $\mathbf{m}_{i,k}(t)$ for $k = [1, ..., T]$ and for $i = [1, ..., c]$ is the k^{th} value of the i^{th} code vector.

Then the BMU is calculated with

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \left\{ \| \mathbf{x_{t+1}} - \mathbf{m}_i\,(t) \|_{NM_{\mathbf{x_{t+1}}}} \right\}. \tag{4}$$

When the BMU is found the network weights are updated as

$$\mathbf{m}_i\,(t+1) = \mathbf{m}_i\,(t) - \varepsilon(t)\lambda \left(\mathbf{m}_{BMU(\mathbf{x}_{t+1})}, \mathbf{m}_i, t \right) \left[\mathbf{m}_i\,(t) - \mathbf{x_{t+1}} \right], \forall i \in I, \tag{5}$$

where $\varepsilon(t)$ is the adaptation gain parameter, which is $]0, 1[$-valued, decreasing gradually with time. The number of neurons taken into account during the weight update depends on the neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$. The number of neurons, which need the weight update, usually decreases with time.

After the weight update the next sample is randomly drawn from the data matrix and the procedure started again by finding the BMU of the sample. The recursive learning procedure is stopped when the SOM algorithm has converged.

Once the SOM algorithm has converged, we obtain some clusters containing our data. Cottrell and Letrémy proposed to fill the missing values of the dataset by the coordinates of the code vectors of each BMU as natural first candidates for missing value completion:

$$\pi_{(M_{\mathbf{x}})}\left(\mathbf{x}\right) = \pi_{(M_{\mathbf{x}})}\left(\mathbf{m}_{BMU(\mathbf{x})}\right), \tag{6}$$

where $\pi_{(M_{\mathbf{x}})}(.)$ replaces the missing values $M_{\mathbf{x}}$ of sample $\mathbf{x}$ with the corresponding values of the BMU of the sample. The replacement is done for every data sample and then the SOM has finished filling the missing values in the data.

The procedure is summarized in Table 1. There is a toolbox available for performing the SOM algorithm in [7].

Table 1: Summary of the SOM algorithm for finding the missing values.

1. SOM node weights are initialized randomly
2. SOM learning process begins
   - (a) Input $\mathbf{x}$ is drawn from the learning data set $\mathbf{X}$
     - i. If $\mathbf{x}$ does not contain missing values, BMU is found according to Equation 2
     - ii. If $\mathbf{x}$ contains missing values, BMU is found according to Equation 4
   - (b) Neuron weights are updated according to Equation 6
3. Once the learning process is done, for each observation containing missing values, the weights of the BMU of the observation are substituted for missing values

4 Empirical Orthogonal Functions

This section presents Empirical Orthogonal Functions (EOF) [3]. In this paper, EOF are used as a denoising tool and for finding the missing values at the same time [4].

The EOF are calculated using standard and well-known Singular Value Decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^* = \sum_{k=1}^{K} \rho_k \mathbf{u}_k \mathbf{v}_k, \tag{7}$$

where $\mathbf{X}$ is 2-dimensional data matrix, $\mathbf{U}$ and $\mathbf{V}$ are collections of singular vectors $\mathbf{u}$ and $\mathbf{v}$ in each dimension respectively, $\mathbf{D}$ is a diagonal matrix with the singular values ρ in its diagonal and K is the smaller dimension of $\mathbf{X}$ (or the number of nonzero singular values if $\mathbf{X}$ is not full rank). The singular values and the respective vectors are sorted to decreasing order.

When EOF are used to denoise the data, not all singular values and vectors are used to reconstruct the data matrix. Instead, it is assumed that the vectors corresponding to larger singular values contain more data with respect to the noise than the ones corresponding to smaller values [3]. Therefore, it is logical to select q largest singular values and the corresponding vectors and reconstruct the denoised data matrix using only them.

In the case where $q < K$, the reconstructed data matrix is obviously not the same than the original one. The larger q is selected, the more original data, which also includes more noise, is preserved. The optimal q is selected using validation methods, for example [8].

EOF (or SVD) cannot be directly used with databases including missing values. The missing values must be replaced by some initial values in order to use the EOF. This replacement can be for example the mean value of the whole data matrix $\mathbf{X}$ or the mean in one direction, row wise or column wise. The latter approach is more logical when the data matrix has some temporal or spatial structure in its columns or rows.

After the initial value replacement the EOF process begins by performing the SVD and the selected q singular values and vectors are used to build the reconstruction. In order not to lose **any** information, only the missing values of $\mathbf{X}$ are replaced with the values from the reconstruction. After the replacement, the new data matrix is again broken down to singular values and vectors with the SVD and reconstructed again. The procedure is repeated until convergence criterion is fulfilled.

The procedure is summarized in Table 2.

5 Global Methodology

The two methodologies presented in the previous two sections are combined and the global methodology is presented. The SOM algorithm for missing values is first ran through performing a nonlinear projection for finding the missing values. Then, the result of the SOM estimation is used as initialization for the EOF method. The global methodology is summarized in Table 1

For the SOM we must select the optimal grid size c and for the EOF the optimal number of singular values and vectors q to be used. This is done using validation, using the same validation set for all combinations of the parameters c and q. Finally, the combination of SOM and EOF that gives the smallest validation error is used to perform the final filling of the data.

Table 2: Summary of the EOF method for finding missing values.

1. Initial values are substituted into missing values of the original data matrix $\mathbf{X}$
2. For each q from 1 to K
   (a) SVD algorithm calculates q singular values and eigenvectors
   (b) A number of values and vectors are used to make the reconstruction
   (c) The missing values from the original data are filled with the values from the reconstruction
   (d) If the convergence criterion is fulfilled, the validation error is calculated and saved and the next q value is taken under inspection. If not, then we continue from step a) with the same q value
3. The q with the smallest validation error is selected and used to reconstruct the final filling of the missing values in $\mathbf{X}$

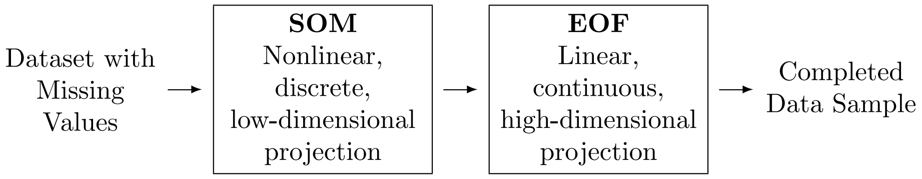
Dataset with Missing Values
SOM
Nonlinear, discrete, low-dimensional projection
EOF
Linear, continuous, high-dimensional projection
Completed Data Sample

Fig. 1: Global methodology summarized.

Even the SOM as well as the EOF are able to fill the missing values alone, the experimental results demonstrate that together the accuracy is better. The fact that these two algorithms suit well together is not surprising. Two approaches can be considered to understand the complementarity of the algorithms.

Firstly, the SOM algorithm allows nonlinear projection. In this sense, even for dataset with complex and nonlinear structure, the SOM code vectors will succeed to capture the nonlinear characteristics of the inputs. However, the projection is done on a low-dimensional grid (in our case two-dimensional) with the possibility of losing the intrinsic information of the data.

The EOF method is based on a linear transformation using the Singular Value Decomposition. Because of the linearity of the EOF approach, it will fail to reflect the nonlinear structures of the dataset, but the projection space can be as high as the dimension of the input data and remain continuous.

There is a toolbox for performing the SOM+EOF in [9].

6 Experimental Results

In this paper, the ESTSP2007 competition dataset is used as an example. It includes a total of 875 values. The dataset is shown in Figure 2.

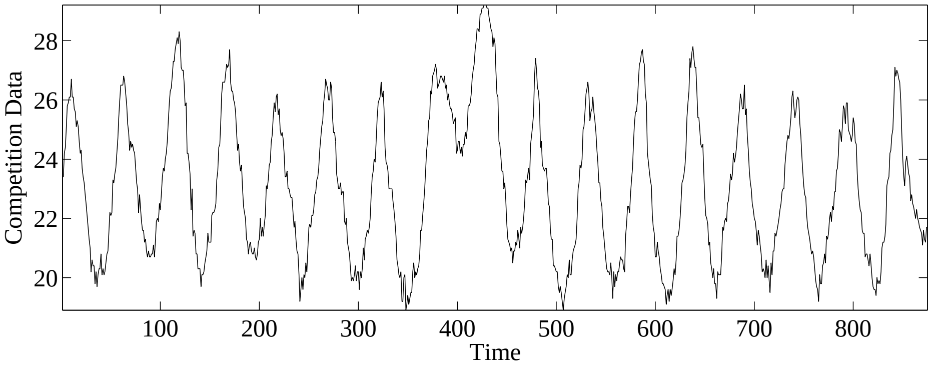

Fig. 2: Competition dataset.

For the SOM algorithm, the dataset is divided into two sets, learning and validation set. The learning set consists of 465 first values and the rest belongs to the validation set. The optimal regressor size is set to 11 after many trial and error experiments.

The optimal SOM size is selected using a simple validation procedure, where the SOM learning is performed using only the learning set and the validation set is used to tune the SOM size for one step ahead prediction. The validation errors are shown in Figure 3.

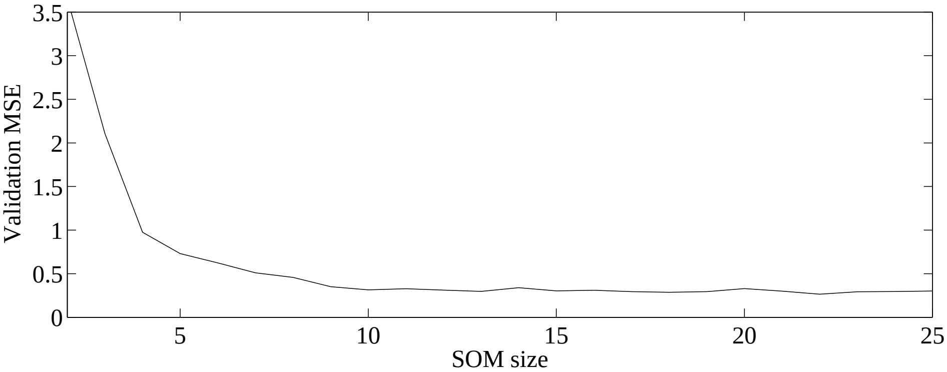

Fig. 3: Validation errors with respect to the SOM grid size.

From Figure 3 the optimal SOM size is selected to 13×13 with validation error of 0,297. There is only very small difference in the validation error with larger SOM sizes.

The only parameter of the EOF method is tuned using the same learning and validation sets than with the SOM to get comparable results. Also the regressor size is kept the same than with the SOM and the optimization is done for one step ahead prediction. The validation errors are shown in Figure 4.

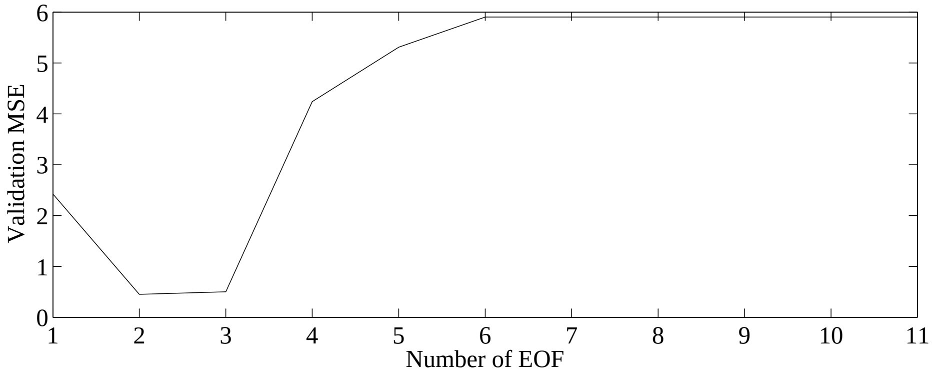

Fig. 4: Validation errors with respect to the number of EOF.

From Figure 4 the optimal number of EOF is selected to 2 with validation error of 0,451. The result suggests relatively strong noise influence in the singular values after the third one, where the validation error is increasing rapidly.

For the SOM+EOF method the two separate methods are combined and the validation is performed for each combination of the SOM sizes and the number of EOF. The validation errors are shown in Figure 5 and 6.

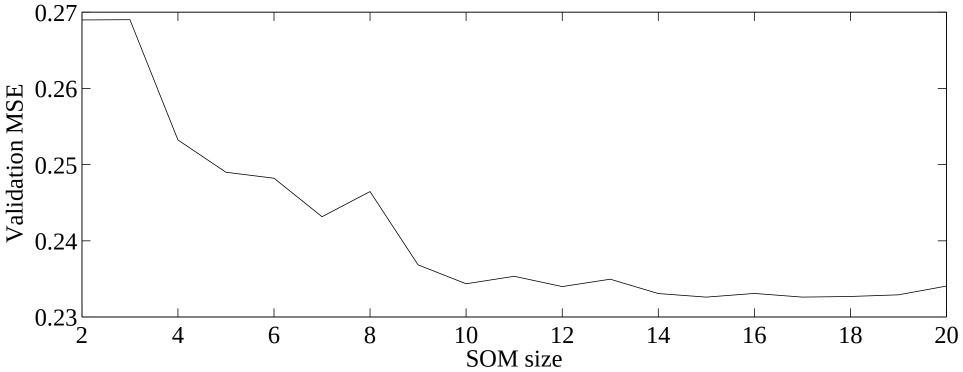

Fig. 5: Minimum validation errors with respect to the SOM size using the SOM+EOF method.

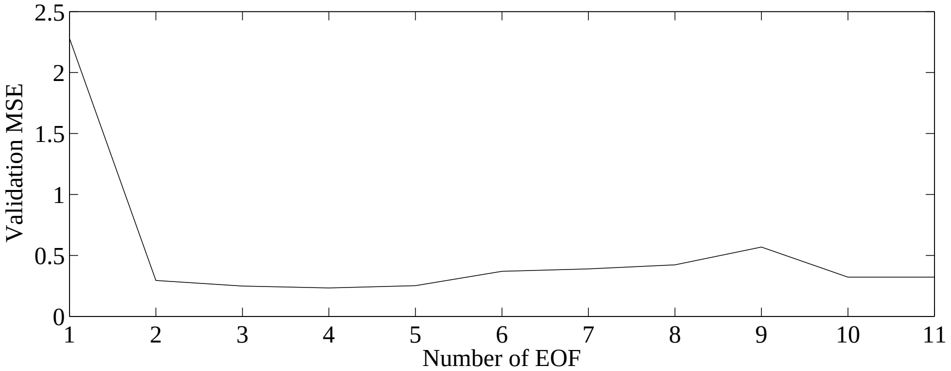

Fig. 6: Validation errors with respect to the number of EOF using SOM size 15×15.

From Figure 5 the optimal SOM is selected to be 15×15 and from Figure 6 the optimal number of EOF to 4 with the validation error of 0,233.

For one step ahead prediction the regressor size is selected to 11, but for the 50 steps ahead the regressor size is increased to 60 in order to fit the missing values to the regressor.

Our experiments with several other datasets have shown that the EOF method uses larger number of EOF when the regressor size is increased. Therefore, the final prediction is done using the number of EOF fixed to 8. The prediction of the 50 timesteps is shown in Figure 7.

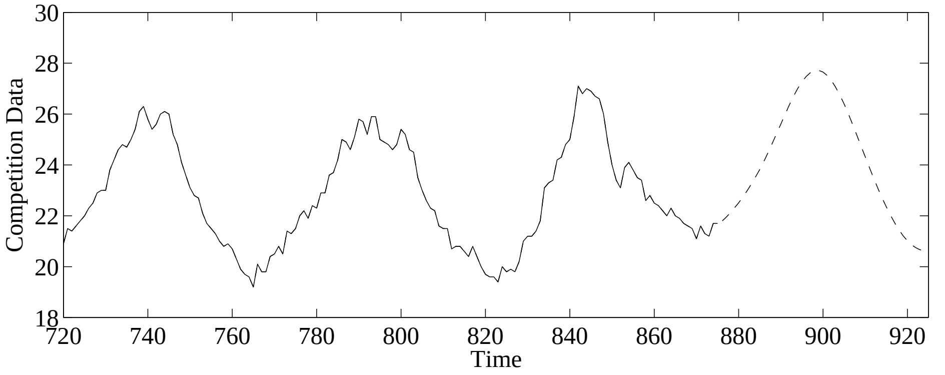

Fig. 7: Prediction of 50 next values of the competition dataset. The real values are presented by the solid line and the dashed one presents the prediction.

From the Figure 7 it seems that that the prediction has removed the noise and is predicting the next peak of the time series quite well.

7 Conclusion

In this paper, we have presented 3 methods for finding missing values in temporal database. The methods are Self-Organizing Maps (SOM), Empirical Orthogonal Function (EOF) and the combination of the two SOM+EOF. The methods are used to find the future values of a time series.

The advantages of the SOM include the ability to perform nonlinear projection of high-dimensional data to lower dimension with interpolation between discrete data points.

For the EOF, the advantages include high-dimensional linear projection of high-dimensional data and the speed and the simplicity of the method.

The SOM+EOF includes the advantages of both individual methods, leading to a new accurate approximation methodology for the missing future values of a time series. The performance obtained show the accuracy of the new methodology.

It is also evident that the EOF is greatly dependent from good initialization in order to produce accurate results. The SOM gives good initialization even the method alone is not so accurate. The two methods complete each other and work well together.

For further work, the modifications and performance upgrades for the global methodology are investigated and applied to other types of datasets from other fields of science, for example climatology and finance.

References

[1] Teuvo Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[2] Shouhong Wang. Application of self-organising maps for data mining with incomplete data sets. *Neural Computing and Applications*, 12(1):42–48, 2003.

[3] R. Preisendorfer. *Principal Component Analysis in Meteorology and Oceanography.* Elsevier, 1988.

[4] J. Boyd, E. Kennelly, and P. Pistek. Estimation of eof expansion coefficients from incomplete data. *Deep Sea Research.*

[5] Antti Sorjamaa and Amaury Lendasse. Time series prediction using dirrec strategy. pages 143–148. European Symposium on Artificial Neural Networks, ESANN 2006, Bruges (Belgium), 26-28 April, 2006.

[6] Marie Cottrell and Patrick Letrémy. Missing values: Processing with the kohonen algorithm. pages 489–496. Applied Stochastic Models and Data Analysis, Brest, France, 17-20 May, 2005.

[7] SOM Toolbox: http://www.cis.hut.fi/projects/somtoolbox/.

[8] Amaury Lendasse, V. Wertz, and Michel Verleysen. Model selection with cross-validations and bootstraps - application to time series prediction with rbfn models. In *LNCS*, number 2714, pages 573–580, Berlin, 2003. ICANN/ICONIP (2003), Springer-Verlag.

[9] SOM+EOF Toolbox: http://www.cis.hut.fi/projects/tsp/?page=Downloads.