WCCI 2010 IEEE World Congress on Computational Intelligence
July, 18-23, 2010 - CCIB, Barcelona, Spain IJCNN

Effect of Different Detrending Approaches on Computational Intelligence Models of Time Series

Federico Montesino Pouzols, *Member, IEEE* and Amaury Lendasse, *Member, IEEE*

***Abstract*— This paper analyzes the impact of different detrending approaches on the performance of a variety of computational intelligence (CI) models. Three approaches are compared: Linear, nonlinear detrending (based on empirical mode decomposition) and first-differencing. Five representative CI methods are evaluated: Dynamic evolving neural-fuzzy inference system (DENFIS), Gaussian process (GP), multilayer perceptron (MLP), optimally-pruned extreme learning machine (OP-ELM) and Support Vector Machines (SVM). Four major conclusions are drawn from experiments performed on six time series benchmarks: 1) qualitatively, the effect of detrending is remarkably uniform for all the CI methods considered, 2) extraction of the overall trend does not improve performance in general 3) the EMD-based method provides better performance than linear detrending (while the difference is negligible in most cases, it is noticeable in some cases), and 4) first-differencing, while effective in some cases, can be counterproductive for series showing common patterns.**

I. Introduction

Traditional statistical methods [1], [2] common in time series applications assume stationarity of the dataset to be modeled. This is likewise the case of conventional regression methods based on neural networks and computational intelligence, which are commonly used to address time series modeling and prediction problems.

However, the phenomena underlying time series often exhibit complex nonlinear behavior. Many classes of dynamical behavior have been described, including regular predictable and unpredictable behavior, transient and intermittent chaos, narrow-band and broad-band chaos, pseudo-randomness and superposition of several basic patterns [3]. In real world applications, most time series exhibit nonstationarity, whether dynamical of statistical. As a consequence, it is generally accepted that removing trends (detrending) and seasonal components (deseasonalizing) can help improve the performance of (stationarity assuming) modeling methods.

Indeed, finding and modeling trends is one of the major tasks in the analysis and prediction of time series. Trends also play a central role in time series data mining. Thus, there has been a great deal of work on trend identification and detrending in a variety of disciplines from diverse areas of science and engineering. Hence, finding trends can be motivated by at least two main reasons: turn a nonstationary time series into stationary, and characterzing its behavior by separating components such as trend, cycles, fluctuations and noise.

Federico Montesino Pouzols and Amaury Lendasse are with the Department of Information and Computer Science, Aalto University School of Science and Technology, P.O. Box 15400, FI-00076 Aalto, Espoo, Finland (phone: +358-9-470-25237; email: federico.pouzols@tkk.fi, amaury.lendasse@tkk.fi). FMP is supported by a Marie Curie Intra-European Fellowship for Career Development (grant agreement PIEF-GA-2009-237450) within the European Community´s Seventh Framework Programme (FP7/2007–2013). We acknowledge CSC-IT Center for Science Ltd, administered by the Ministry of Education of Finland, for its support and high-performance computing services.

However, there is no general consensus on how trends should be modeled [4]. Furthermore, the concept of trend, despite the widespread use of the term, lacks concrete and formal definitions [5]. It has not been until recently that a first definition of overall trend that can account for nonlinear, nonstationary data has been formally established [5].

The problem of trend extraction from time series still poses many fundamental questions. It is generally accepted that methods for analysis and prediction of time series can be affected by preprocessing steps that operate on trends. However there is also a lack of consensus on how these steps should be applied.

In this context, an analysis of the influence of detrending techniques on computational intelligence (CI) models would shed some light on the design of sound CI-based methodologies for time series.

Despite the importance detrending can have, there is little experimental or sufficiently general theoretical results in the literature. There are some results in the literature concerning prediction with neural networks for series with trend [4]. However, these are generally application specific and/or based on limited data or restricted to a particular detrending approach. Results available are disperse, usually showing ad-hoc analyses, often contradictory, and are at best based on synthetic data for simple dynamics or limited real-world data [4].

In this paper we analyze the impact of different detrending approaches on the performance of a variety of CI models. The paper is organized as follows. In section II detrending is put in the general context of preprocessing steps for time series. Section III provides a precise definition of trend and the methods for trend extraction used here. Sections IV and V present and further discuss experimental results. Finally, conclusions are summarized.

II. Preprocessing and Detrending

In the most simple approach, the trend of a time series is identified by fitting a deterministic component, usually linear. The trend is then subtracted in order to guarantee stationarity of the resulting time series. More generally, detrending is the process by which a trend is removed from a time series. The residue after this operation can be called variability or fluctuation. One common constraint on this process is that the detrended series must be a zero-mean process for the time span considered in the detrending.

During the last few decades, stochastic approaches have become popular specially in econometrics. Also different approaches to detrending have become popular in diverse fields, such as model-based approaches, nonparametric filtering (such as linear filtering and singular spectrum analysis), and wavelets [6]. Different approaches have been proposed for so-called trend-stationary series and difference-stationary series [4].

In general, detrending can be considered one of the possibly many preprocessing steps that are performed on a time series before building a CI model. These steps can include, among others, imputation of missing values, treatment of outliers and uncertain values, denoising, filtering, nonlinear transformations for improving smoothness, variable selection, and computation of dynamical and statistical characteristics, such as residual variance and time lags. However, the separation of preprocessing stages is not clear in many cases. For instance it is possible to perform tasks related to both model selection and outliers jointly [7].

Also, the difference between filtering, denoising and detrending may not be obvious in some cases. Furthermore, one same method can be used to perform several of these tasks. For instance, single spectrum analysis can be used to perform denoising, extraction of trends and periodic components, prediction and change-point detection [6].

In this paper certain detrending approaches are applied as the only preprocessing step (besides normalization of values). In this approach, assuming a simple additive superposition of trend and variability, a time series can be decomposed as:

$$y(t) = T(t) + r(t),$$

where $r(t)$ is a residual, variability or fluctuation around the trend, $T(t)$. The fluctuation component can have both stochastic and deterministic components.

If only detrending is done, a model $\hat{y}(t)$ for $y(t)$ becomes $\hat{y}(t) = T(t) + m_{CI}(t)$, where the first component is the overall trend and the second component is the model of the fluctuations around the overall trend.

We note though that it is very common in some fields to split a time series into at least three components: trend, seasonal component and irregular component. For simplicity, here we restrict our analysis to the approach above. The analysis of the effect of deseasonalization methods is left for future work.

III. Trend extraction: Methods

Detecting, identifying trends and removing them from time series, i.e., detrending, is key for data analysis in virtually all fields. It is also a necessary preprocessing step for statistical techniques such as computing correlations, as well as spectral analysis techniques.

Time series are addressed from different viewpoints in fields such as statistics [2], [1], and nonlinear dynamics [3]. Some comparative studies have been performed for neural networks models [8], international time series prediction competitions [9], [10], [11], [10] have been organized, and global methodologies have been proposed [12]. However, there is no general consensus on how trends should be modeled [4] or if trends should be modeled separatedly in general. Furthermore, the concept of trend, despite the widespread use of the term, lacks concrete and formal definitions [5]. As a consequence, there is a large variety of approaches and concrete methods for trend extraction. In this context, it is highly difficult to make a decision on which method is best. As Wu et al. point out, in the econometrics field, what is called trend by some economists is called cycle by others [13].

For instance, according to Alexandrov et al. [6], it can be considered "a smooth additive component that contains information about global change." Several key aspects are included in this definition: smoothness, additivity, and overall information.

In many applications, besides the overall trend, other cyclical behaviors can be separated from residual fluctuations, i.e., so-called multidecadal trends in geophysics and environmental sciences, or seasonal components in econometrics.

From the time-frequency viewpoint the trend is often defined as the residue after higher frequency components above a certain threshold have been removed. This is for instance the case of methods based on wavelets, curvelets or singular spectrum analysis (SSA) [6], and nonparametric filtering methods, such as the well-known Hodrick-Prescott filter or more recent filters for extracting piecewise linear trends based on l_1 minimization [14].

On the other hand, regression model based approaches, using for instance ARIMA models, assume an a priori model structure for the data, failing thus to be truly adaptive. Similarly, moving mean approaches are based on an a priori given time scale to compute the averages.

Thus, in general the definition of overall trend is not clear when using these methods, as the process depends on the selection of parameters and a priori structures. For instance, if using wavelet analysis, the choice of the type of wavelet can have a determining impact on the outcome of the detrending process. In this paper, we concentrate on paremeter-free, automatic methods of general applicability that identify a clearly defined overall trend.

A. Linear detrending

The most common trend identification approach consists in fitting a straight line to the time series. Then, the corresponding detrending process by which the straight line trend, $T(t) = a_0 + b_0 t$, is subtracted from the time series yields a zero mean residue.

This linear detrending approach is a particular case of polynomial detrending: Polynomial detrending of order 1. The implementation used in the next sections of this paper is based on on a least squares fitting performed by a QR decomposition of the time domain. In order to improve the numerical performance of the fitting, it is applied on the following transformation of the time domain:

$$\hat{t} = \frac{t - \mu_1}{\mu_2},$$

where μ_1 and μ_2 are the average and standard deviation of t, respectively. Detrending with higher order polynomials usually suffers from overfitting issues. Moreover, it is not difficult to show that detrending by polynomial fitting can have negative effects on spectral analysis techniques [15].

B. EMD-based detrending

The aforementioned methods are based on a time-independent regression formula applied on the dataset. Except for physical process with well-known dynamics this implies fitting an a priori model to data that in most cases is nonstationary.

This limitation can be overcome by using the empirical mode decomposition (EMD), introduced by Huang et al. [16], [13] Wu, Huang et al. define the trend as an intrinsically determined monotonic function or a function, within a certain temporal span, where there can be at most one extremum [5]. The method to extract the trend must be thus adaptive, and the EMD is proposed in order to fully account for the nonlinearities and nonstationarities of the data. In this proposal the overall trend is computed as the residual of the EMD decomposition process.

EMD is performed through an iterative process known as sifting. What follows is an algorithm that performs sifting:

1) Identify all the local extrema of $y(t)$.
2) Build an upper and a lower envelope, $y_{up}(t)$ and $y_{low}(t)$, respectively, by connecting all the local extrema (maxima and minima, respectively) using a cubic spline.
3) Compute an envelope mean $m(t)$ point by point.
4) Extract the details signal, $d(t) = y(t) - m(t)$.
5) Check the properties of the details signal.
6) Repeat steps 1-5 until the residual satisfies a certain stopping condition.

At the end of the sifting process, the original time series $y(t)$ is decomposed as follows:

$$y(t) = \sum_{j=1}^{p} c_j(t) + r_p(t),$$

where $r_p(t)$ is the final residue, i.e., the overall trend of $y(t)$, $c_j(t)$ are the intrinsic mode functions (IMFs) identified and p is the total number of IMFs.

The frequency bands contained in each IMF are different and change over time as the time series $y(t)$ changes. The basis of expansion of the EMD method is therefore adaptive and locally determined. Thus, it offers a more physically meaningful representation as compared to methods based on a priori bases.

Then, by applying the Hilbert transform on each IMF, $y(t)$ can be expressed in the time-frequency domains as follows:

$$Y(t, \omega) = RP \left(\sum_{j=1}^{n} a_j(t) e^{i \int_0^t \omega_j(\tau) d\tau} \right),$$

where RP denotes the real part of the transforms, $a_j(t)$ is an expansion coefficient that depends on time, as opposed to the coefficients of the Fourier transform, and $\omega_j(\tau)$ is the instantaneous frequency at time τ. Thus, both the frequency and amplitude of each compoent are functions of time. This contrasts with the Fourier representation, which would have the following form:

$$Y(t) = RP \sum_{j=1}^{\infty} a_j e^{i w_j t},$$

where both w_j and a_j are constants. In this sense, the IMFs can be seen as generalized Fourier expansions with two advantages: more concise expansion and the capability to accommodate nonlinear, nonstationary behavior directly from the dataset. This distribution in the time-frequency domains, $Y(t, \omega)$ is known as the Hilbert spectrum.

EMD has a strong parallelism to a mathematical technique widely used in the field of fluid dynamics: The Reynolds decomposition. Stated loosely, EMD performs a decomposition that separates the average and fluctuating components of a signal (or the steady components and perturbations, where the time average of the perturbations is zero).

In practice, some problems such as mixing of different mode into a same IMF and wrong separation of modes into two or more IMFs can arise. Several approaches have been proposed to overcome these limitations. Indeed, several aspects can be fine-tuned in the EMD process in order to obtain better results under certain conditions. However, no theoretical conditions have been established and in practice it is difficult to define a variant of the EMD that can avoid every limitation in a sufficiently general manner. Wu et al. have recently proposed ensemble EMD (EEMD) [13] as a noise-assisted method with improved robustness against the aforementioned problems. In order to further enhance the robustness and accuracy of the EMD decomposition and final trend, in this paper we apply the EEMD with extended postprocessing recently proposed in [13].

IV. Experiments

In the experiments that follow, the focus is on automatic modeling (both for detrending and then modeling of fluctuations), that can capture the underlying behavior in a robust manner with as few assumptions as possible, thus taking full advantage of CI methods. We pursue to find out whether certain detrending approaches can be beneficial in general or under particular conditions. It is shown that clear patterns can be found by carefully analyzing a number of benchmarks.

Three approaches to detrending are evaluated: linear, EEMD-based detrending and first-differencing. First-differencing has been extensively used in the econometrics field, where difference-stationarity has been found to play a key role in macroeconomic aggregates [4]. Thus it is often compared against detrending approaches. We note however that while we include the first-differencing approach in our analysis, it is not a method for extracting the overall trend. In fact, after the first differences of a time series are computed, cycles or seasonal components are removed as well as the overall trend.

A. CI Methods

The following CI methods have been used for offline modeling of time series: DENFIS, GP, MLP, OP-ELM and SVM. For DENFIS the offline training mode of the DENFIS toolbox [17] was used. The GP models are build using the default settings of the GPML toolbox [18]. In particular, a covariance function as the sum of a squared exponential (SE) contribution and an independent noise contribution is used. OP-ELM models are build using the OP-ELM toolbox [19], [20]. Default settings are selected, in particular all possible kernels, linear, sigmoid, and Gaussian are used for a maximum number of 100 hidden nodes.

A ten-fold crossvalidation strategy was used for model selection in the case of the MLP and SVM models. For SVM models, a grid-search strategy is applied to explore the three parameters in a logarithmic scale for values in the range $[-2, 10]$. The $\varepsilon-$Support Vector Regression scheme was used with radial basis function kernels. The libSVM version 2.9 [21] library was used to implement this method. The MLP method is implemented using the standard neural network toolbox included in the Matlab environment. The networks are optimized using the Levenberg-Marquardt method. In order to mitigate the effect of local minima, ten repetitions are done for each fold and each particular size of the network (between 1 and 20 hidden nodes), and the model yielding the lowest validation error is selected.

B. Datasets

For the sake of simplicity we restrict this study to univariate time series. The datasets were chosen with the following considerations in mind: a) there should be at least a few hunderd test values so that test errors are statistically significant, and b) datasets should be representative of real world applications, where the underlying dynamics is often unknown and modeling errors are usually higher than for most synthetic series employed in the literature [4].

The main characteristics of the datasets used are shown in table I. These particular datasets were selected in order to find a compromise between the following objectives: a) easing comparison with the related literature, b) selecting datsets for a broad range of characteristics (variables, size, dynamical behavior, etc.). In particular, some datasets represent clearly nonstationary processes.

The datasets can be obtained from either the Time Series Data Library [22] or the datasets repository of the Time Series Prediction and Chemoinformatics Group [23]. As an example, figure 1 shows the tree rings time series.

The series of tree rings contains yearly measures of tree rings widths in dimensionless units. This series was measured in Argentina for the 441–1974 period and corresponds to the arge030 dataset of the Time Series Data Library [22]. In this case, $y(t+1)$ (width for the next year) has to be predicted using the 10 previous values $y(t-9), \ldots, y(t)$, except $y(t-4)$ and $y(t-6)$. Refer to [24], [25] for further descriptions and all the necessary details to reproduce results.

TABLE I

DATASETS: NUMBER OF INPUTS, TRAINING OBSERVATIONS AND TEST OBSERVATIONS

Dataset	# Inputs	Training length	Test length
ENSO	3	465	400
Darwin SLP	5	904	467
Internet2	4	708	730
Sunspots	7	2085	1000
Santa Fe Laser	3	988	9093
Tree Rings	8	1013	511

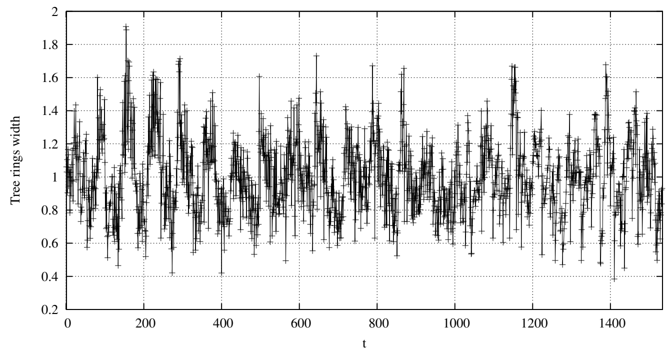

Fig. 1

TREE RING TIME SERIES.

The Santa Fe Laser dataset of the Santa Fe time series competition [26], [11]. represents the intensity of a far-infrared-laser in a chaotic state, measured in a physics laboratory experiment. The series is a cross-cut through periodic to chaotic pulsations of the laser, and can be closely modeled analytically [11]. This series is a remarkable example of noise-free complicated behavior in a clean, stationary, low-dimensional physical system for which the underlying dynamics is well understood. In this case, the next value, $x(t+1)$) has to be modeled based on 3 inputs: $x(t), x(t-1), x(t-2)$ and $x(t-12)$. This subset of inputs has been previously identified as optimal for a maximum regressor size of 12 [24].

We also analyzed the series of monthly averaged Sunspot numbers covering from January 1749 through December 2007, as provided by the National Geographical Data Center from the US National Oceanic and Atmospheric Administration[1]. Given the yearly periodicity of the series, a maximum regressor size of 12 was defined. $y(t+1)$ (next month value) has to be predicted using $y(t), y(t-1), y(t-2), y(t-3), y(t-4), y(t-8)$ and $y(t-10)$.

[1]The series is available online from http://www.ngdc.noaa.gov/stp/SOLAR/. The International Sunspot Number is produced by the Solar Influence Data Analysis Center (SIDC) at the Royal Observatory of Belgium [27].

The Internet2 time series represents the total amount of aggregated incoming traffic in the routers of the Abilene network, the Internet2 backbone, during several years. The series consists of 1458 daily averages from the 4th of January of 2003 through the 31st of December of 2006. The data are available from the Abilene Observatory [28]. The traffic load for the next day ($y(t+1)$) has to be predicted using $y(t), y(t-3), y(t-6), y(t-7), y(t-9), y(t-11), y(t-12)$ and $y(t-13)$.

$y(t+7)$ (next week) has to be predicted using $y(t), y(t-2), y(t-4)$, and $y(t-11)$.

The Darwin SLP time series consists of monthly values of the Darwin Sea sea level Pressure for the years 1882–1998. The SLP for the next month, $y(t+1)$ has to be predicted using five known values from the past, $y(t-11)$, $y(t-6)$, $y(t-3)$, $y(t-2)$, and $y(t-1)$.

Finally, the ENSO series is the data set from the ESTSP 2007 time series prediction competition [10]. This data set consists of 875 samples of weekly sea surface temperatures associated to the El Niño-Southern Oscillation phenomenon. $y(t+1)$ has to be predicted using $y(t), y(t-2)$, and $y(t-7)$ as inputs.

First, detrending is performed using the techniques discussed in previous sections. Table II reports the percentage of energy of the original signal extracted as trend.

TABLE II

DETRENDING RESULTS IN TERMS OF ENERGY.

Dataset	Method	% of energy in trend
ENSO	Linear	98.89
	EEMD	99.10
	1st diff.	99.90
Darwin SLP	Linear	93.51
	EEMD	91.39
	1st diff.	99.89
Internet2	Linear	93.18
	EEMD	93.78
	1st diff.	99.93
Santa Fe Laser	Linear	61.79
	EEMD	79.94
	1st diff.	99.98
Sunspots	Linear	59.49
	EEMD	66.59
	1st diff.	99.99
Tree rings	Linear	94.80
	EEMD	93.93
	1st diff.	99.95

Figure 2 shows the ENSO series together with its linear and EMD-based trends. The first difference of the ENSO series is shown in figure 3. It is clear that the yearly seasonality is removed.

C. Modeling and prediction results

Predictive models for detrended time series are generated here following a conventional offline modeling approach. First, a training and a test set are defined. Then, the model is built for the training set and evaluated on the test set. This is so as apposed to online approaches [17] where the problem of trend computation raises additional issues.

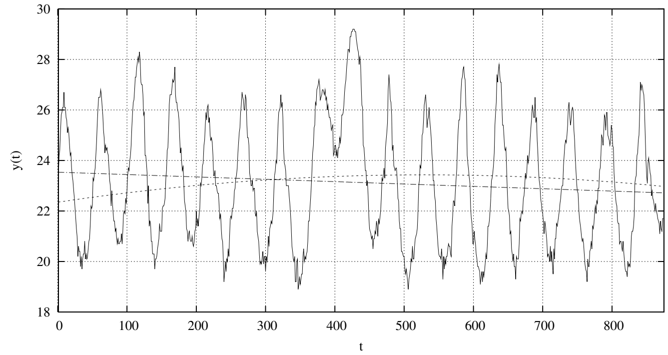

Fig. 2

ENSO TIME SERIES (CONTINUOUS), LINEAR TREND (DASHED-DOTTED LINE) AND EMD TREND (DASHED).

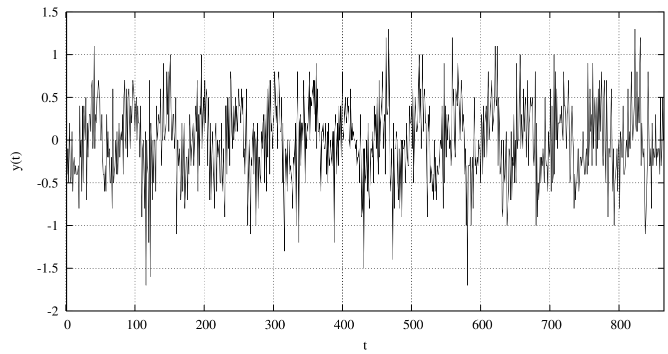

Fig. 3

FIRST DIFFERENCE OF THE ENSO TIME SERIES.

We concentrate on methods for removing the overall trend for two reasons: a) it is a first step that we consider should be analyzed separately, b) extrapolation for long-term prediction seems more feasible. In order to compare with previous results [4], the first-differencing method is also considered. Obviously, the first difference method raises issues when the objective is long-term prediction of sequences of values.

Training and test errors as well as training time are reported in tables III through VIII. The average and standard deviation of the training and test errors are shown normalized against the standard deviation of the respective time series. The time column in these tables shows the processor time consumed for the learning process on the same environment[2].

In order to compare the performance of different detrending approaches, the errors are measured on the final output of the global model, i.e., on the extracted trend plus fluctuations modeled with CI methods. In what follows, *linear* denotes

[2]A multiprocessor system running Matlab Version 7.9.0.529 (R2009b) 64-bit (GNU/Linux operating system, glnxa64 architecture), where each process was allocated a core of a Quad-Core AMD Opteron(tm) Processor 8360 SE.

the linear detrending method, *EEMD-based* denotes the nonlinear, nonstationary detrending method based on EEMD described in the previous section, and *first-difference* denotes the first-difference filtering method. We focus on test errors.

TABLE III

RESULTS FOR ENSO USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	1.41e-1±8.91e-2	1.80e-1±1.13e-1	1.07
	Linear	1.42e-1±8.70e-2	**1.77e-1±1.09e-1**	8.90e-1
	EEMD	1.42e-1±8.75e-2	1.78e-1±1.10e-1	7.00e-1
	1st diff.	1.55e-1±9.52e-2	1.95e-1±1.20e-1	1.19
GP	-	1.41e-1±8.80e-2	1.74e-1±1.09e-1	3.28e+1
	Linear	1.42e-1±8.79e-2	1.73e-1±1.07e-1	2.75e+1
	EEMD	1.43e-1±8.84e-2	**1.72e-1±1.08e-1**	4.04e+1
	1st diff.	1.59e-1±9.70e-2	1.90e-1±1.18e-1	3.54e+1
MLP	-	1.41e-1±8.90e-2	**1.89e-1±1.17e-1**	7.38e+2
	Linear	1.41e-1±8.61e-2	2.12e-1±1.31e-1	7.01e+2
	EEMD	1.66e-1±1.04e-1	1.96e-1±1.22e-1	9.70e+2
	1st diff.	1.63e-1±9.96e-2	1.97e-1±1.22e-1	6.68e+2
OP-ELM	-	1.39e-1±8.70e-2	**1.73e-1±1.09e-1**	6.80e-1
	Linear	1.42e-1±8.83e-2	1.83e-1±1.15e-1	6.00e-1
	EEMD	1.34e-1±8.25e-2	1.86e-1±1.13e-1	9.80e-1
	1st diff.	1.58e-1±9.73e-2	1.94e-1±1.22e-1	7.00e-1
SVM	-	1.53e-1±8.74e-2	1.98e-1±1.29e-1	3.83e+1
	Linear	1.54e-1±8.76e-2	1.96e-1±1.28e-1	3.98e+1
	EEMD	1.54e-1±8.70e-2	1.95e-1±1.28e-1	3.89e+1
	1st diff.	1.58e-1±9.76e-2	**1.90e-1±1.18e-1**	2.39e+2

Table III reports results for the ENSO time series. *linear* provides a slight improvement for DENFIS, GP and SVM. However, results are noticeably worse (errors 5-10% higher) for MLP and OP-ELM. The advantages of *EEMD-based* are consistent but negligible. *First-difference* leads to worse results with a slight exception for SVM.

Results for the Darwin SLP time series are shown in table IV. Results for *linear* are slightly worse for all the methods. *EEMD-based* provides better results than *linear* (except for OP-ELM), but still provides very slightly improvements or worse results than the case without detrending. However, *First-difference* achieves test errors at least 25% lower for all the methods.

In the case of the Internet2 time series (results shown in table V), *linear* improves noticeably the test error for all methods except OP-ELM. *EEMD-based* yields again similar or better results than *linear*, although GP is an exception. *First-difference* provides the worst results in general, with an exception for SVM models.

Table VI reports results for the Santa Fe laser time series. Test errors are in general higher for *linear* detrending than no detrending, with an exception for MLP. *EEMD-based* provides in general significantly better results than *linear* but still worse test errors than no detrending. Test errors for *First-difference* are clearly worse with no exceptions.

Let us now consider the results for the Sunspots time series, shown in table VII. Test errors are in general slightly worse for *linear* detrending than no detrending, being clearly worse for MLP and OP-ELM. *EEMD-based* is again equivalent to or better than *linear* but still worse than no detrending (with a slight exception for OP-ELM). *First-difference* yields clearly better results in all cases.

Finally, table VIII shows the results obtained for the tree rings series. In this case *linear* detrending improves the results slightly for all the methods except DENFIS. *EEMD-based* also provides equivalent or better results than *linear* and no detrending (with an exception for SVM). *First-difference* provides clearly worse results, except for SVM.

TABLE IV

RESULTS FOR DARWIN SLP USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	3.88e-1±2.34e-1	4.37e-1±2.69e-1	2.11
	Linear	3.85e-1±2.34e-1	4.40e-1±2.71e-1	2.17
	EEMD	3.88e-1±2.36e-1	4.34e-1±2.68e-1	2.42
	1st diff.	3.02e-1±1.94e-1	**3.16e-1±2.00e-1**	3.05
GP	-	3.82e-1±2.34e-1	4.31e-1±2.66e-1	1.61e+2
	Linear	3.82e-1±2.34e-1	4.32e-1±2.66e-1	1.71e+2
	EEMD	3.84e-1±2.35e-1	4.29e-1±2.66e-1	2.75e+2
	1st diff.	3.00e-1±1.93e-1	**3.07e-1±1.99e-1**	2.05e+2
MLP	-	3.87e-1±2.37e-1	4.41e-1±2.77e-1	8.66e+2
	Linear	3.92e-1±2.37e-1	4.47e-1±2.75e-1	8.78e+2
	EEMD	3.94e-1±2.41e-1	4.45e-1±2.74e-1	1.32e+3
	1st diff.	3.14e-1±2.05e-1	**3.18e-1±2.07e-1**	1.02e+3
OP-ELM	-	3.87e-1±2.39e-1	4.42e-1±2.73e-1	1.55
	Linear	3.95e-1±2.46e-1	4.47e-1±2.79e-1	1.41
	EEMD	3.76e-1±2.31e-1	4.55e-1±2.90e-1	2.79
	1st diff.	3.04e-1±1.94e-1	**3.15e-1±2.02e-1**	2.27
SVM	-	3.84e-1±2.31e-1	4.31e-1±2.65e-1	4.47e+2
	Linear	3.84e-1±2.30e-1	4.32e-1±2.65e-1	4.49e+2
	EEMD	3.86e-1±2.32e-1	4.30e-1±2.65e-1	4.66e+2
	1st diff.	2.95e-1±1.89e-1	**3.09e-1±2.01e-1**	5.92e+2

TABLE V

RESULTS FOR INTERNET2 USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	3.60e-1±2.85e-1	5.47e-1±4.22e-1	1.84
	Linear	3.50e-1±2.74e-1	5.22e-1±4.01e-1	2.27
	EEMD	3.51e-1±2.73e-1	**5.11e-1±3.89e-1**	2.26
	1st diff.	4.30e-1±3.53e-1	5.55e-1±4.32e-1	1.92
GP	-	1.47e-1±1.12e-1	6.19e-1±4.57e-1	2.04e+2
	Linear	1.67e-1±1.24e-1	**5.33e-1±3.86e-1**	3.63e+2
	EEMD	1.57e-1±1.16e-1	5.41e-1±3.85e-1	2.84e+2
	1st diff.	7.93e-2±5.81e-2	6.24e-1±4.76e-1	2.23e+2
MLP	-	4.79e-1±3.84e-1	5.75e-1±4.29e-1	4.21e+2
	Linear	4.74e-1±3.80e-1	**5.50e-1±3.97e-1**	6.40e+2
	EEMD	4.88e-1±3.80e-1	5.54e-1±3.92e-1	5.41e+2
	1st diff.	5.16e-1±4.18e-1	6.32e-1±4.71e-1	4.55e+2
OP-ELM	-	4.62e-1±3.75e-1	5.51e-1±4.01e-1	1.74
	Linear	4.35e-1±3.40e-1	5.72e-1±4.00e-1	4.94
	EEMD	4.53e-1±3.59e-1	**5.42e-1±3.92e-1**	3.17
	1st diff.	4.39e-1±3.49e-1	5.94e-1±4.59e-1	1.56
SVM	-	3.10e-1±2.42e-1	6.10e-1±4.41e-1	1.11e+2
	Linear	3.01e-1±2.37e-1	**5.11e-1±3.88e-1**	1.28e+2
	EEMD	2.99e-1±2.35e-1	**5.11e-1±3.89e-1**	1.26e+2
	1st diff.	4.31e-1±3.73e-1	5.54e-1±4.58e-1	9.44e+1

V. DISCUSSION

Broadly, it can be concluded that detrending does not generally improve the performance of the CI methods studied,

TABLE VI

RESULTS FOR SANTA FE LASER USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	2.34e-1±1.91e-1	**2.35e-1±1.93e-1**	1.84
	Linear	2.33e-1±1.90e-1	2.44e-1±1.94e-1	2.11
	EEMD	2.34e-1±1.91e-1	2.37e-1±1.92e-1	1.99
	1st diff.	3.82e-1±3.18e-1	4.02e-1±3.34e-1	1.73
GP	-	5.02e-2±4.14e-2	**1.17e-1±1.08e-1**	1.64e+2
	Linear	5.29e-2±4.35e-2	1.39e-1±1.21e-1	1.69e+2
	EEMD	5.03e-2±4.15e-2	1.26e-1±1.13e-1	1.88e+2
	1st diff.	1.41e-1±1.20e-1	1.92e-1±1.66e-1	2.75e+2
MLP	-	2.56e-1±2.49e-1	2.30e-1±2.21e-1	2.01e+3
	Linear	8.40e-2±6.87e-2	1.83e-1±1.45e-1	2.14e+3
	EEMD	1.28e-1±1.17e-1	**1.42e-1±1.23e-1**	2.38e+3
	1st diff.	2.23e-1±1.88e-1	2.56e-1±2.15e-1	1.96e+3
OP-ELM	-	1.09e-1±9.08e-2	**1.31e-1±1.11e-1**	1.92
	Linear	1.12e-1±9.03e-2	2.17e-1±1.64e-1	1.55
	EEMD	1.29e-1±1.06e-1	1.67e-1±1.38e-1	2.35
	1st diff.	2.05e-1±1.49e-1	2.40e-1±1.83e-1	1.60
SVM	-	1.59e-1±1.06e-1	**1.65e-1±1.14e-1**	1.66e+2
	Linear	1.59e-1±1.07e-1	1.79e-1±1.21e-1	1.61e+2
	EEMD	1.59e-1±1.06e-1	1.70e-1±1.16e-1	1.66e+2
	1st diff.	1.92e-1±1.24e-1	2.31e-1±1.70e-1	1.84e+2

TABLE VII

RESULTS FOR SUNSPOTS USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	3.82e-1±2.69e-1	**3.40e-1±2.28e-1**	6.69
	Linear	3.82e-1±2.70e-1	3.47e-1±2.37e-1	6.82
	EEMD	3.82e-1±2.69e-1	3.41e-1±2.29e-1	6.47
	1st diff.	3.78e-1±2.68e-1	3.55e-1±2.41e-1	1.80e+1
GP	-	3.81e-1±2.67e-1	**3.36e-1±2.27e-1**	2.43e+3
	Linear	3.80e-1±2.66e-1	**3.36e-1±2.27e-1**	3.45e+3
	EEMD	3.78e-1±2.65e-1	3.37e-1±2.27e-1	2.64e+3
	1st diff.	2.04e-1±1.37e-1	3.71e-1±2.51e-1	2.12e+3
MLP	-	3.93e-1±2.75e-1	**3.53e-1±2.41e-1**	2.72e+3
	Linear	3.91e-1±2.77e-1	3.77e-1±2.64e-1	3.01e+3
	EEMD	3.90e-1±2.78e-1	3.83e-1±2.70e-1	2.89e+3
	1st diff.	3.91e-1±2.76e-1	**3.53e-1±2.38e-1**	2.16e+3
OP-ELM	-	3.90e-1±2.76e-1	3.40e-1±2.33e-1	1.21e+1
	Linear	3.85e-1±2.72e-1	3.66e-1±2.52e-1	1.29e+1
	EEMD	3.89e-1±2.75e-1	**3.36e-1±2.26e-1**	1.14e+1
	1st diff.	3.90e-1±2.76e-1	3.43e-1±2.32e-1	1.15e+1
SVM	-	3.60e-1±2.46e-1	3.55e-1±2.33e-1	2.69e+3
	Linear	3.57e-1±2.47e-1	**3.49e-1±2.33e-1**	2.61e+3
	EEMD	3.59e-1±2.46e-1	3.56e-1±2.34e-1	2.31e+3
	1st diff.	3.82e-1±2.73e-1	3.52e-1±2.40e-1	2.74e+3

TABLE VIII

RESULTS FOR TREE RINGS USING DIFFERENT DETRENDING APPROACHES AND CI MODELS.

CI method	Detrending	Training RMSE	Test RMSE	Time (s)
DENFIS	-	7.03e-1±4.35e-1	8.30e-1±5.78e-1	6.20
	Linear	6.98e-1±4.30e-1	8.89e-1±6.40e-1	6.85
	EEMD	6.97e-1±4.29e-1	**8.28e-1±5.74e-1**	7.23
	1st diff.	7.57e-1±4.67e-1	8.42e-1±5.92e-1	5.56
GP	-	7.67e-1±4.80e-1	7.69e-1±5.38e-1	2.98e+2
	Linear	7.66e-1±4.78e-1	**7.68e-1±5.38e-1**	3.59e+2
	EEMD	7.67e-1±4.79e-1	7.69e-1±5.38e-1	4.06e+2
	1st diff.	7.84e-1±4.82e-1	8.01e-1±5.58e-1	4.26e+2
MLP	-	7.65e-1±4.75e-1	8.09e-1±5.47e-1	8.37e+2
	Linear	7.77e-1±4.89e-1	7.92e-1±5.26e-1	8.61e+2
	EEMD	7.77e-1±4.87e-1	**7.88e-1±5.45e-1**	1.15e+3
	1st diff.	7.85e-1±4.81e-1	8.26e-1±5.70e-1	1.32e+3
OP-ELM	-	7.54e-1±4.65e-1	7.92e-1±5.39e-1	2.05
	Linear	7.64e-1±4.75e-1	**7.70e-1±5.26e-1**	2.09
	EEMD	7.75e-1±4.85e-1	**7.70e-1±5.34e-1**	2.07
	1st diff.	7.84e-1±4.84e-1	8.07e-1±5.61e-1	2.07
SVM	-	6.41e-1±3.81e-1	8.31e-1±5.61e-1	2.90e+2
	Linear	6.43e-1±3.80e-1	8.25e-1±5.54e-1	2.83e+2
	EEMD	6.41e-1±3.80e-1	8.30e-1±5.60e-1	3.02e+2
	1st diff.	7.21e-1±4.44e-1	**7.92e-1±5.52e-1**	2.67e+2

neither in accuracy nor computational cost. It should also be noted that the effect of detrending methods is not necessarily of the same sign for training and test errors. Thus, it is difficult to estimate a priori whether detrending is desirable for a particular application case. This is arguably a reason to use detrending methods with care.

Regarding the linear and EEMD-based methods for extracting the overall trend, it can be concluded that:

- Extraction of the overall trend does not improve performance in general. In 4 out of the 6 benchmarks, it is just slightly advantageous for some models, ineffective or harmful.
- EEMD based detrending yields better results than linear detrending. While the difference is negligible in most cases, it is consistent and noticeable in some cases.

Considering first-differencing, it can be concluded that:

- It can significantly improve the accuracy of models in some cases. This may be in particular a consequence of its ability to extract cyclic components.
- It can also yield clearly worse results than those obtained with no detrending or the other two methods.

Regarding computational time, the impact of linear detrending, whether positive or negative, depends on the method and dataset and is not significant. On the other hand, EEMD-based detrending produces a significant increase of the training time, which implies that its advantages come at a certain computational cost. First-differencing does not in general speed up the learning process. On the contrary, in many cases it also leads to a significant slow down.

In principle, modeling methods can benefit from effective detrending methods. A good detrending method should improve the results of a certain modeling method. In turn, finding which detrending methods perform better from the perspective of models built on detrended data can shed some light on the performance of the detrending process itself. In other words, the impact on performance of models a detrending approach has can be seen as a goodness criteria for the detrending approach itself.

We should mention that the use of the EMD-based detrending approach for offline training is conceptually contradictory. Indeed, the advantages of EMD derive from the fact that the theory behind has no stationarity assumptions. This implies that using an offline training approach, i.e., assuming the dataset to be stationary, misses the fundamental aim o of the EMD. For EMD-based detrending, evolving methods [29] should be expected to take full advantage of the nonlinear, nonstationary decomposition performed by EMD.

Finally, the results reported here apply to overall trend extraction methods used as the only preprocessing step. The impact of such methods when combined with methods for implementing other complementary tasks such as deseasonalization may substantially differ.

VI. Conclusions

We have presented an experimental analysis of the effect of different detrending approaches on CI models of time series. While being far from exhaustive, different modeling methods coming from diverse areas of CI have been considered. Also, disparate detrending approaches popular in various fields were included in this study.

Several detrending approaches have been compared in terms of the performance of predictive models build on detrended datasets. Modeling of detrended series was compared for three detrending approaches: linear, EEMD-based (or nonlinear, nonstationary) and first-differencing. The following CI methods were compared: DENFIS, GP, MLP, OP-ELM and SVM.

Although the results presented here provide no general answer to the many open issues, the following major conclusions can be drawn:

- Qualitatively, the effect of detrending is remarkably uniform for all the CI methods considered.
- Extraction of the overall trend does not improve performance in general. In 4 out of the 6 benchmarks, it is just slightly advantageous, ineffective or harmful.
- A nonlinear, nonstationary detrending method such as the EEMD-based method used here provides better performance than linear detrending. While the difference is negligible in most cases, it is noticeable in some cases.
- First-differencing, while clearly effective as detrending and deseasonalizing method in some cases, can be counterproductive for times series that show common patterns.

The bottom line of these results is that detrending should be used with extreme care. Further work is required in order to draw more general and quantitative conclusions. It is also worth to explore the performance of detrending and deseasonalizing methods when combined.

References

[1] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting & Control*. Prentice Hall; 3rd edition, 1994, ISBN: 0130607746.

[2] C. Chatfield, *The Analysis of Time Series. An Introduction*. CRC Press, Jul. 2003, Sixth edition, ISBN: 1-58488-317-0.

[3] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[4] M. Qi and G. P. Zhang, "Trend time–series modeling and forecasting with neural networks," *IEEE Trans.. Neural Netw.*, vol. 19, no. 5, pp. 808–816, May 2008.

[5] Z. Wu, N. E. Huang, S. R. Long, and C.-K. Peng, "On the trend, detrending, and variability of nonlinear and nonstationary time series," *Proc. Nat. Acad. Sci.*, vol. 104, no. 38, pp. 14 889–14 894, Sep. 2007.

[6] T. Alexandrov, S. Bianconcini, E. B. Dagum, P. Maass, and T. McElroy, "A review of some modern approaches to the problem of trend extraction," U.S. Census Bureau, Statistical Research Division, Research Report Series, Statistics #2008-3, Mar. 2008.

[7] C. Chen and L.-M. Liu, "Joint estimation of model parameters and outlier effects in time series," *Journal of the American Statistical Association*, vol. 88, no. 421, pp. 284–297, Mar. 1993.

[8] J. Faraday and C. Chatfield, "Time Series Forecasting with Neural Networks: A Comparative Study Using the Airline Data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 47, no. 2, pp. 231–250, 1998.

[9] "NN3 Artificial Neural Network & Computational Intelligence Forecasting Competition," Jun. 2008. [Online]. Available: http://www.neural-forecasting-competition.com

[10] "ESTSP: European Symposium on Time Series Prediction," Feb. 2010. [Online]. Available: http://www.estsp.org

[11] A. Weigend and N. Gershenfeld, *Times Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley Publishing Company, 1994.

[12] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for Long-Term Prediction of Time Series," *Neurocomputing*, vol. 70, no. 16–18, pp. 2861–2869, Oct. 2007.

[13] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, vol. 1, no. 1, pp. 1–41, Jan. 2009.

[14] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "l_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, May 2009.

[15] M.-T. Lo, V. Novak, C.-K. Peng, Y. Liu, and K. Hu, "Nonlinear phase interaction between nonstationary signals: A comparison study of methods based on Hilbert-Huang and Fourier transforms," *Phys. Rev. E*, vol. 79, no. 6, p. 061924, Jun. 2009.

[16] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Snin, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. R. Soc. A*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[17] N. K. Kasabov and Q. Song, "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans.. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.

[18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[19] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Trans.. Neural Netw.*, vol. 21, no. 1, pp. 158–162, Jan. 2010.

[20] Y. Miche, A. Sorjamaa, and A. Lendasse, "OP-ELM: Theory, Experiments and a Toolbox," in *Proc. Int. Conf. Artif. Neural Netw.*, ser. Lect. Notes Comput. Sci, vol. 5163, Prague, Czech Republic, Sep. 2008, pp. 145–154.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," Nov. 2009, version 2.9. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[22] R. J. Hyndman, "Time series data library," Feb. 2010. [Online]. Available: http://www.robjhyndman.com/TSDL

[23] "Time Series Datasets. Time Series Prediction and Chemoinformatics Group," Feb. 2010. [Online]. Available: http://www.cis.hut.fi/projects/tsp

[24] F. Montesino Pouzols, A. Lendasse, and A. Barriga, "Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation," *Fuzzy Sets Syst.*, vol. 161, no. 4, pp. 471–497, Feb. 2010.

[25] ——, "Fuzzy Inference Based Autoregressors for Time Series Prediction Using Nonparametric Residual Variance Estimation," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Hong Kong, China, Jun. 2008, pp. 613–618.

[26] "The Santa Fe Time Series Competition Data. Data Set A: Laser generated data," Mar. 2010. [Online]. Available: http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

[27] R. A. M. Van der Linden and the SIDC Team, "Online Catalogue of the Sunspot Index," RWC Belgium, World Data Center for the Sunspot Index, Royal Observatory of Belgium, years 1748-2007, http://sidc.oma.be/html/sunspot.html, Jan. 2008.

[28] "The Internet2 Observatory," Jul. 2008. [Online]. Available: http://www.internet2.edu/observatory/

[29] N. Kasabov, *Evolving Connectionist Systems: The Knowledge Engineering Approach*, 2nd ed. Springer, Jul. 2007.