Jarkko Tikka and Jaakko Hollmén. 2008. Sequential input selection algorithm for long-term prediction of time series. Neurocomputing, volume 71, numbers 13-15, pages 2604-2615.

Neurocomputing 71 (2008) 2604–2615

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Sequential input selection algorithm for long-term prediction of time series

Jarkko Tikka *, Jaakko Hollmén

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

ARTICLE INFO

Available online 29 April 2008

Keywords:
Input selection
Time series prediction
Parsimonious modeling
Multilayer-perceptron networks
Sensitivity analysis

ABSTRACT

In time series prediction, making accurate predictions is often the primary goal. At the same time, interpretability of the models would be desirable. For the latter goal, we have devised a sequential input selection algorithm (SISAL) to choose a parsimonious, or sparse, set of input variables. Our proposed algorithm is a sequential backward selection type algorithm based on a cross-validation resampling procedure. Our strategy is to use a filter approach in the prediction: first we select a sparse set of inputs using linear models and then the selected inputs are used in the nonlinear prediction conducted with multilayer-perceptron networks. Furthermore, we perform a sensitivity analysis by quantifying the importance of the individual input variables in the nonlinear models using a method based on partial derivatives. Experiments are done with the Santa Fe laser data set that exhibits very nonlinear behavior and a data set in a problem of electricity load prediction. The results in the prediction problems of varying difficulty highlight the range of applicability of our proposed algorithm. In summary, our SISAL yields accurate and parsimonious prediction models giving insight to the original problem.

1. Introduction

Time series analysis [3,11,24] is an important problem in natural and engineering sciences, both from the viewpoint of prediction and for understanding the behavior of the system under study. There are numerous applications of time series analysis scattered in the literature of econometrics, system identification, chemistry, statistics, pattern recognition, and neural networks [25]. Interesting applications of time series prediction can also be found, for instance, in climatology [5], ecology [19], electricity production [17], and economics [10]. Most of the research is concentrated on finding an accurate predictor of future values based on the available past data. However, it would be very appealing to be able to predict behavior of the time series accurately, and at the same time to provide insight into the system itself. Our target is to estimate time series prediction models that are both accurate and interpretable. By interpretability we mean that the models contain only a relatively small subset of input variables for the prediction. In our applications, approximately only one-third of the available input variables were included in the final prediction models. This puts emphasis on what is important in the prediction of system behavior. These kinds of parsimonious, or sparse, time series models are the focus of this work. Inputs of the sparse models are selected from a large set of autoregressive input variables for a given past horizon. This approach tries to circumvent the problems of a high-dimensional input space, i.e. the curse of dimensionality [23].

Approaches to input selection are reviewed in [7,9]. Two main approaches used in the input selection are the wrapper approach [16] and the filter approach [1]. In the wrapper approach, inputs are selected according to their suitability in the prediction using the final model itself. In the filter approach, an input selection procedure is independent from the final model, which is built after a fixed set of inputs is determined. While the wrapper approach is expected to yield better input variables for the particular model class, the computational costs can be high or prohibitive in the presence of a large number of inputs. The filter approach makes the compromise of using a simple model in the selection phase, which saves a lot of computational burden, but it possibly introduces some inaccuracies. Our contribution builds on the filter approach to input selection.

In this article, we present our sequential input selection algorithm (SISAL) for a long-term time series prediction problem building on our previous work [21,22]. Long-term prediction is difficult and time-consuming, since it extends the horizon of prediction further into the future, adding significant uncertainty in the prediction task. Technically, our algorithm works in the spirit of the backward selection algorithm [12], in which input variables are progressively removed from the autoregressive prediction model. In SISAL, the removal of inputs is based on a median and an empirically estimated width of parameter distributions sampled with a cross-validation resampling procedure [6]. These statistics reflect the importance of an input in the

* Corresponding author. Tel.: +358 9 451 3543.
E-mail address: tikka@mail.cis.hut.fi (J. Tikka).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.11.037

prediction task. In the second phase, the nonlinear prediction model is constructed using a fixed subset of inputs. In this work, we used the multilayer-perceptron (MLP) networks [2], although any kind of nonlinear predictors could be used (as for instance in [20]). We also investigate experimentally applicability of the filter approach to problems with different degree of nonlinearities. This is done comparing the result of the proposed filter approach to the traditional forward selection (FS) algorithm, which belongs to the class of wrapper input selection methods. The results give impressions of behavior in general situations, but exact rules cannot be given.

The rest of the article is organized as follows: Section 2 introduces relevant background in time series prediction. Section 3 reviews our SISAL in the context of linear time series prediction models. Section 4 focuses on nonlinear prediction models, i.e. MLP networks, in which the selected input variables are finally used. Also, the FS algorithm is described followed by the presentation of sensitivity analysis of the MLP networks based on partial derivatives (PADs). Section 5 presents the empirical experiments in an electricity load prediction problem and in a problem of predicting the Santa Fe laser data set. Summary and conclusions are presented in Section 6.

2. Long-term prediction of time series

In a time series prediction problem, future values of time series are predicted using the previous values [3,11,24]. The previous and future values of time series are referred to inputs and outputs of the prediction model, respectively. One-step-ahead prediction is needed in general and it is called short-term prediction. If multistep-ahead predictions are needed, it is known as long-term prediction.

Unlike short-term prediction, long-term prediction faces typically growing amount of uncertainties arising from various sources. For instance, an accumulation of errors and lack of information make the prediction more difficult. In the case of long-term prediction, there are several strategies to build prediction models [25]. In Sections 2.1 and 2.2, the common strategies of recursive and direct prediction are described briefly.

2.1. Recursive prediction strategy

In the case of multistep-ahead prediction, the recursive strategy uses the predicted values as known data to predict the next ones. First, a one-step-ahead prediction is done

$$\hat{y}_t = f_1(y_{t-1}, y_{t-2}, \ldots, y_{t-l}),$$

where $y_{t-i},\ i = 1, \ldots, l$ are the inputs. It is also possible to use exogenous variables as inputs, but they are not considered here in order to simplify the notation. After that, the same model is used to make a two-step-ahead prediction

$$\hat{y}_{t+1} = f_1(\hat{y}_t, y_{t-1}, y_{t-2}, \ldots, y_{t-l+1}),$$

where the predicted value of $\hat{y}_t$ is used instead of the true value, which is unknown. Then, the k-step-ahead predictions $y_{t+k-1}, k \geqslant 3$ are obtained iteratively. In the prediction of kth step, $l-k$ observed values and k predicted values are used as the inputs in the case of $k < l$. When $k \geqslant l$, all the inputs are predicted values. The use of the predicted values as inputs may deteriorate the accuracy of the prediction.

2.2. Direct prediction strategy

In the direct strategy, the model

$$\hat{y}_{t+k-1} = f_k(y_{t-1}, y_{t-2}, \ldots, y_{t-l})$$

is used for k-step-ahead prediction. The predicted values are not used as inputs at all in this approach, thus the errors in the predicted values do not get accumulated in subsequent predictions. When all the values from y_t to y_{t+k-1} need to be predicted, k different models must be constructed. This increases the computational complexity, but more accurate results are achieved, as it is shown in [15,21]. We only apply the direct strategy in this work.

3. Sequential input selection algorithm (SISAL)

Let us assume that there are N measurements available from a time series $y_t,\ t = 1, \ldots, N$. Future values of time series y_t are predicted using the previous values $y_{t-i},\ i = 1, \ldots, l$. If the dependency between the output y_t and the inputs y_{t-i} is assumed to be linear it can be written as

$$y_t = \sum_{i=1}^{l} \beta_i y_{t-i} + \varepsilon_t, \tag{1}$$

which is a linear autoregressive process of order l, or AR(l). The errors ε_t are assumed to be independently normally distributed with zero mean and common finite variance $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$. In addition, all the variables are assumed to have zero mean and unit variance, thus the constant term is dropped from the model (1). The ordinary least squares (OLS) estimates of the regression coefficients $\hat{\beta}_i$ are obtained by minimizing the mean squared error

$$\mathrm{MSE} = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2, \tag{2}$$

where $\hat{y}_t$ is the estimated output.

The usual goal is to define the order l in the model (1) and use all the inputs $y_{t-i},\ i = 1, \ldots, l$ in the prediction of y_t. This kind of solution is not typically satisfactory [12]. Firstly, the generalization ability of the model may be improved by shrinking some coefficients toward zero or setting them exactly to zero [4]. Secondly, if the number of inputs l is large, interpretation of the model might be difficult. The understanding of the underlying process can be improved by selecting the subset of inputs which have the strongest effect in the prediction. Many approaches to input selection are presented in [12,18].

We have proposed and analyzed an algorithm to efficiently select a subset of inputs that have the strongest explanatory power in the autoregressive (AR) process in our previous publications [21,22]. The algorithm is based on the resampling procedures, such as bootstrapping or cross-validation [6]. The advantage of resampling procedures is that the input selection can be carried out using measured data without restrictive assumptions.

First, the maximum number of inputs l have to be set, which defines the order of the AR process. It is selected to be relatively large to ensure that all the important inputs in the prediction are included in the final model. The algorithm continues by estimating the AR process using all the input variables $y_{t-i},\ i = 1, \ldots, l$ and calculating the OLS estimates of the parameters $\hat{\beta}_i$. The sampling distributions of the OLS estimates $\hat{\beta}_i$ and the standard deviation s_{tr} of the training MSEs are estimated using M times k-fold cross-validation. We have Mk different training and validation sets. Mk estimates of the each regression coefficients $\hat{\beta}_i^j,\ j = 1, \ldots, Mk$ formulate the sampling distribution of the corresponding parameter β_i. In addition, we have Mk estimates

for both training and validation MSE. The validation error is the MSE for the validation set, i.e. for the data that are not used in the evaluation of the regression coefficients $\hat{\beta}_i$.

The median m_{β_i} is calculated from Mk estimates of $\hat{\beta}_i^j$. The median is used as the location parameter for the distribution, since it is a reasonable estimate for skewed distributions and distributions with outliers and it coincides with the mean in the case of symmetric distributions. The width of the distributions can be estimated using the standard deviation σ_{β_i} of replications $\hat{\beta}_i^j$ or a deviation based on the median

$$d_{\beta_i} = \sqrt{\frac{1}{Mk-1}\sum_{j=1}^{Mk}(m_{\beta_i} - \hat{\beta}_i^j)^2}, \tag{3}$$

where $\hat{\beta}_i^j$ is the jth replication in the cross-validation repetitions. Another alternative to estimate the width is to evaluate the difference

$$\Delta_{\beta_i} = \hat{\beta}_i^{\text{high}} - \hat{\beta}_i^{\text{low}}, \tag{4}$$

where $\hat{\beta}_i^{\text{high}}$ and $\hat{\beta}_i^{\text{low}}$ are the $Mk(1-q)$th and Mkqth values in the ordered list of the Mk estimates of $\hat{\beta}_i$, respectively [6]. The constant q is predefined and an appropriate value is $q = 0.165$. With this choice of q, the difference Δ_{β_i} is twice as large as the standard deviation in the case of the normal distribution. The standard deviation σ_{β_i} and deviation based on the median d_{β_i} are reasonable measures for symmetric distributions, whereas the difference Δ_{β_i} describes well the width of both asymmetric and symmetric distributions.

The next step is to delete the least significant input variable from the model. The ratio $|m_{\beta_i}|/\Delta_{\beta_i}$ is used as a measure of significance of the corresponding input variable. The ratio can be considered as a signal-to-noise ratio. Firstly, if the median m_{β_i} is close to zero then the corresponding input is not significant in the prediction. Secondly, if the difference Δ_{β_i} is large the effect of the corresponding input in the prediction is unclear. The sampling distributions of parameter β for different combinations of the median m_β and the width Δ_β are exemplified in Fig. 1. The input corresponding to the distribution in the upper left corner would be the most non-informative and the input corresponding to the distribution in the lower right corner would be the most informative. On the other hand, the ratio $|m_{\beta_i}|/\Delta_{\beta_i}$ can be seen as a Z-score test statistic to test the hypothesis that a value of a parameter is $\hat{\beta}_i = 0$ [12]. The statistic is estimated using cross-validation replications, so no assumptions are made about the forms of the probability distributions of the parameters. This applies especially when the median m_{β_i} and the difference Δ_i are used. Thus, the input y_{t-i} with the smallest ratio $|m_{\beta_i}|/\Delta_{\beta_i}$ is dropped from the set of inputs. After that, the cross-validation procedure and pruning are repeated using the remaining inputs, as long as there are variables left in the set of input variables.

The previous procedure removes inputs sequentially from the set of possible inputs. In the end, we have l different models. Our purpose is to select a model which is as sparse as possible in the sense that it includes only a few input variables. However, the sparse model should still have comparable prediction accuracy. An obvious choice is to select the set of inputs $\mathscr{L}_{\text{v}}$, which produces the minimum validation error $E_{\text{v}}^{\min}$. Another option is to include our uncertainty of the prediction accuracy in the training phase into the selection of the set of input variables. This is done by selecting the inputs based on the minimum validation error $E_{\text{v}}^{\min}$ and the corresponding standard deviation of the training error $s_{\text{tr}}^{\min}$. The final set of inputs $\mathscr{L}_{\text{f}}$ corresponds to the least complex model whose validation error is under the threshold $E_{\text{v}}^{\min} + s_{\text{tr}}^{\min}$, which means that $\mathscr{L}_{\text{f}} \subseteq \mathscr{L}_{\text{v}}$. The set of inputs $\mathscr{L}_{\text{f}}$ makes a compromise between sparsity and prediction accuracy. The whole SISAL is described step by step in Algorithm 1.

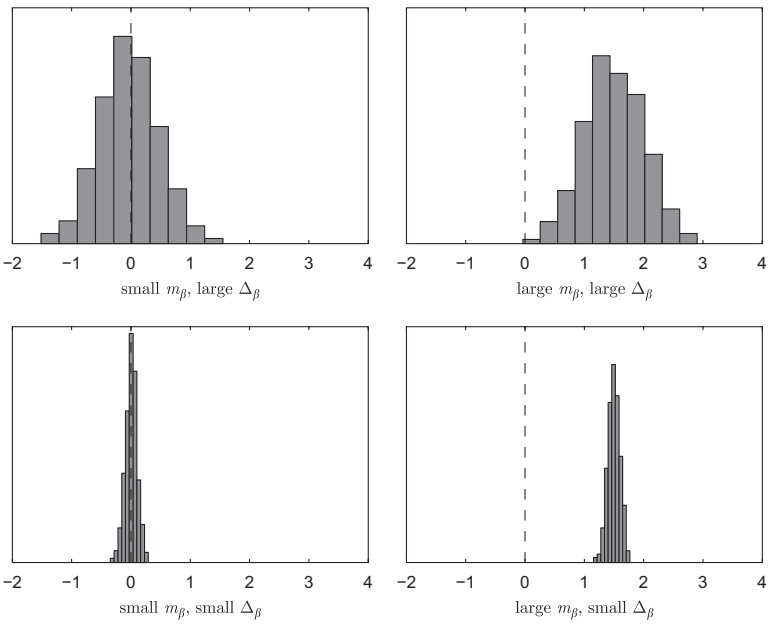

Fig. 1. The sampling distributions of β for different combinations of the median m_β and the width Δ_β. The vertical dashed line marks the point $\beta = 0$.

Algorithm 1. Sequential input selection algorithm (SISAL)
1. Let $\mathscr{L}$ be the set of indices of the inputs. In the beginning, $\mathscr{L}$ includes all the indices i, y_{t-i}, $i=1,\ldots,l$.
2. For $i\in\mathscr{L}$ estimate Mk replications $\hat{\beta}_i^j$, $j=1,\ldots,Mk$ of the parameters $\hat{\beta}_i$ using k-fold cross-validation repeated M times. For $i\notin\mathscr{L}$ set $\hat{\beta}_i=0$.
3. Compute the mean $E_{\mathrm{tr}}^{\mathscr{L}}$ and the standard deviation $s_{\mathrm{tr}}^{\mathscr{L}}$ of the cross-validation replications of the training MSEs.
4. Compute the mean $E_{\mathrm{v}}^{\mathscr{L}}$ of the cross-validation replications of the validation MSEs.
5. Evaluate the median m_{β_i} and the width of the sampling distribution Δ_{β_i} for all the inputs y_{t-i}, $i\in\mathscr{L}$ from the cross-validation replications $\hat{\beta}_i^j$.
6. Evaluate the ratio $|m_{\beta_i}|/\Delta_{\beta_i}$, $\forall i\in\mathscr{L}$ and delete the input with the smallest ratio from the set $\mathscr{L}$.
7. If $\mathscr{L}\neq\emptyset$ go to step 2, otherwise go to step 8.
8. Select the set of inputs $\mathscr{L}_{\mathrm{v}}$ based on the minimum validation error $E_{\mathrm{v}}^{\min}$or select the inputs $\mathscr{L}_{\mathrm{f}}$ corresponding to the least complex model whose validation error is under the threshold $E_{\mathrm{v}}^{\min}+s_{\mathrm{tr}}^{\min}$.

The idea of the proposed algorithm is similar to the traditional backward stepwise selection, which sequentially deletes inputs based on the F-statistics [12]. The computational complexity of the backward stepwise selection algorithm is quadratic $\mathcal{O}(l^2)$ with respect to the number of available inputs, whereas the computational complexity of the proposed algorithm is linear $\mathcal{O}(l)$. It is also clearly faster than an exhaustive search of all possible variable configurations whose complexity is exponential $\mathcal{O}(2^l)$. Therefore, the SISAL (Algorithm 1) is computationally feasible in the case of a large number of inputs. Another advantage of the described algorithm is the ranking of inputs according to their explanatory power. The pruning starts from the least significant inputs and the resulting model includes only a few of the most significant ones. This may give useful information for interpretation of the underlying process which has generated the data. In addition, the application of SISAL to a long-term prediction or any regression problem is straightforwarded.

4. Nonlinear modeling using MLP networks

Although the linear models are easy to interpret and fast to calculate, they are unfortunately not accurate enough in some problems. The dependencies between the variables are better described using a nonlinear model. However, many nonlinear models are black-box models and thus the interpretation of the underlying process that generated the data is nearly impossible. Our proposal is to use the selected set of inputs $\mathscr{L}_{\mathrm{f}}$ in the nonlinear prediction model as well. The goals of this approach are to avoid the curse of dimensionality, over-parameterization, and over-fitting in the nonlinear modeling phase. In addition, interpretability of the nonlinear model is improved since only a subset of inputs is included in the final model, i.e the significant dependencies between the variables are highlighted.

MLP networks [2] with one hidden layer are used in the nonlinear modeling phase

$$\hat{y}_t=\alpha_0+\sum_{j=1}^{p}\alpha_j\tanh\left(\sum_{i\in\mathscr{L}_{\mathrm{f}}}w_{ji}y_{t-i}+w_{j0}\right),\tag{5}$$

where p is the number of neurons in the hidden layer, $\hat{y}_t$ is the estimate of the output y_t and α_0, α_j, and w_{ji} are the weights of the network. The weights are optimized by minimizing the MSE, see Eq. (2). It is known that only one hidden layer is required to approximate any continuous function if the number of connection weights is sufficient [14].

The number of connection weights is controlled by the number of neurons in the hidden layer. Another option is to set the number of neurons to be large enough and to apply weight decay (WD) to reduce the effective number of connection weights [2]. In the case of WD, the cost function is

$$E=\frac{1}{N}\left(\sum_{t=1}^{N}(y_t-\hat{y}_t)^2+\lambda\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta}\right),\tag{6}$$

where λ is the regularization parameter and $\boldsymbol{\theta}$ is the vector containing all the weights of the network w_{ji} and α_j. In WD, the values of the weights are shrunk toward zero, but they are not set exactly to zero. So, it is very likely that WD does not perform input selection. In the context of the linear regression, WD is known as ridge regression [13].

4.1. FS algorithm

A well known but computationally unattractive algorithm to select input variables is the FS algorithm [12]. It is used as a baseline method to compare the prediction accuracy of the proposed two-phase modeling strategy. In the FS algorithm, the idea is to start from the empty set of input variables and add sequentially inputs to the model. In this work, the FS algorithm is started by finding the single input variable which gives the minimum validation error. After that, an input is found, which minimize the validation error with the already added input variable. This procedure is continued as long as all the inputs are added to the model. If l is the number of available inputs, $(l+1)l/2$ MLP networks need to be trained. In addition, in each case the number of neurons in the hidden layer have to validated, which increases significantly the computational burden. Obviously, the final subset of inputs has the smallest validation error of all the subsets. The FS algorithm belongs to the class of wrapper input selection methods with previous formulation.

4.2. Sensitivity analysis

It may not be enough that the most relevant inputs are found. In many applications, it is important to evaluate how the inputs contribute in the prediction of the output.

The contribution of each input to the output can be evaluated using PADs of the MLP network [8]. The PADs method gives two results. Firstly, a set of graphs of the PADs versus each corresponding input can be plotted. The graphs profile the effect of small changes in each input on the output variable. Secondly, the sensitivity of the output variable for the data set with respect to each input can be evaluated. This gives a ranking of the inputs according to their relative importance in the prediction of output. Several methods that describe the relative importances of the inputs in the MLP networks are compared in [8], and the PAD method is found to give stable results.

The PAD of the MLP network (5) with respect to the input y_{t-i} is

$$d_{t,i}=\frac{\partial\hat{y}_t}{\partial y_{t-i}}=\sum_{j=1}^{p}\alpha_j(1-l_j^2)w_{ji},\quad i\in\mathscr{L}_{\mathrm{f}},\tag{7}$$

where $l_j=\tanh(\sum_{i\in\mathscr{L}_{\mathrm{f}}}w_{ji}y_{t-i}+w_{j0})$. The graphs are plotted using the values $(y_{t-i},d_{t,i})$, $i=1,\ldots,N$.

The sensitivity of the MLP output for the data set with respect to an input is calculated as

$$\mathrm{SSD}_i=\sum_{t=1}^{N}d_{t,i}^2,\quad \mathrm{SSD}_i\leftarrow\frac{\mathrm{SSD}_i}{\sum_i\mathrm{SSD}_i},\quad i\in\mathscr{L}_{\mathrm{f}}.\tag{8}$$

Sum of squared derivatives (SSD) value is obtained for each input. SSD_i is the sum over all the observations. In the end, the SSD_i values are normalized such that $\sum_{i\in\mathscr{L}_{\mathrm{f}}}\mathrm{SSD}_i=1$. The input having the highest SSD_i value has the most influence on the output. The ranking of inputs based on SSD values can be compared to the

ranking given by SISAL. The comparison tells how well the input selection based on the linear model performs in the nonlinear problem.

5. Experiments

The two-phase modeling strategy described in the previous sections is applied to time series prediction. Firstly, the input selection is performed using SISAL (see Algorithm 1). Secondly, MLP networks are trained using the selected set of inputs.

The first data set is the Poland electricity load time series [15]. It contains daily measurements from the electricity load in Poland in the 1990s. The data set includes $N_{\text{tr}} = 1400$ observations in the training set and $N_{\text{test}} = 201$ observations in the test set. The training and test sets are not consecutive. The objective is to predict the electricity load one-day-ahead (y_t), two-day-ahead (y_{t+1}), and seven-day-ahead (y_{t+6}). The maximum number of inputs is set to be $l = 15$, i.e. the available inputs are y_{t-i} ($\mathscr{L} = \{i\}$), $i = 1, \ldots, 15$ in each of the three prediction cases.

The second data set is the Santa Fe laser time series described in [25]. The training set size is $N_{\text{tr}} = 1000$, which is the same as in the Santa Fe time series prediction competition [25]. The size of the test set is $N_{\text{test}} = 9093$. The large test set should ensure reliable estimates for the prediction accuracies. In this case the initial set of inputs is y_{t-i} ($\mathscr{L} = \{i\}$), $i = 1, \ldots, 20$. The results are shown for the one-step-ahead (y_t), 10-step-ahead (y_{t+9}), and 20-step-ahead (y_{t+19}) predictions.

We use the direct prediction approach in the long-term prediction, i.e. we have to construct own model for each case. All the variables are scaled to have zero mean and unit variance before the analysis. Both training data sets are illustrated in Fig. 2.

5.1. Phase I: input selection

In SISAL, 10-fold cross-validation repeated $M = 100$ times is used. This choice produces $Mk = 1000$ estimates for the coefficients β_i, which is considered to be large enough for reliably estimating the sampling distributions of the parameters in the linear model.

Fig. 3 illustrates the input selection procedure. In all the prediction cases, it is notable that the validation error does not start to increase significantly until near the end. Almost all the inputs are pruned from the model by that point. It indicates that most of the inputs are irrelevant, at least in the linear model. In the case of the electricity load time series, the minimum validation errors are achieved using 11, 7, and 5 inputs in one-, two-, and seven-day-ahead prediction, respectively. In the case of the Santa Fe time series 13, 11, and 10 inputs produce the minimum validation error in the prediction of y_t, y_{t+9}, and y_{t+19}, respectively. More parsimonious sets of inputs are obtained if thresholding is used. The least complex models whose validation errors are inside the threshold (see Algorithm 1) include five (y_t), five (y_{t+1}), and two (y_{t+6}) inputs (electricity load), and eight (y_t), seven (y_{t+9}), and seven (y_{t+19}) inputs (Santa Fe). It would also be possible to use the standard deviation of the validation error instead of the standard deviation of the training error as the threshold. This choice would result in even more parsimonious sets of inputs, but probably the prediction accuracy would decrease.

In Fig. 4, the inputs selected by SISAL for all the k-step ahead prediction models are visualized. The smaller the white number in the selected inputs (in black and gray rectangles), the more important the corresponding input is in the prediction. That is, the number 1 indicates that the input was the last to be pruned from the linear model. For instance, the first row of the upper figure shows the selected inputs for the one-day-ahead prediction in the electricity load time series. The minimum validation error model includes the following inputs: $\mathscr{L}_{\text{v}} = \{y_{t-7}, y_{t-1}, y_{t-8}, y_{t-14}, y_{t-15}, y_{t-5}, y_{t-12}, y_{t-2}, y_{t-10}, y_{t-6}, y_{t-13}\}$ (black and gray rectangles). The least complex model whose validation error is under the threshold $E_{\text{v}}^{\min} + s_{\text{tr}}^{\min}$ includes only the following inputs: $\mathscr{L}_{\text{f}} = \{y_{t-7}, y_{t-1}, y_{t-8}, y_{t-14}, y_{t-15}\}$. The inputs are listed in decreasing order of importance in $\mathscr{L}_{\text{v}}$ and $\mathscr{L}_{\text{f}}$. The model with the inputs $\mathscr{L}_{\text{f}}$ can be nicely interpreted, since the inputs correspond to values from 7, 1, 8, 14, and 15 days before the predicted value. It is plausible that the two most important inputs are the values of one week and one day before the predicted value. In the case of Santa

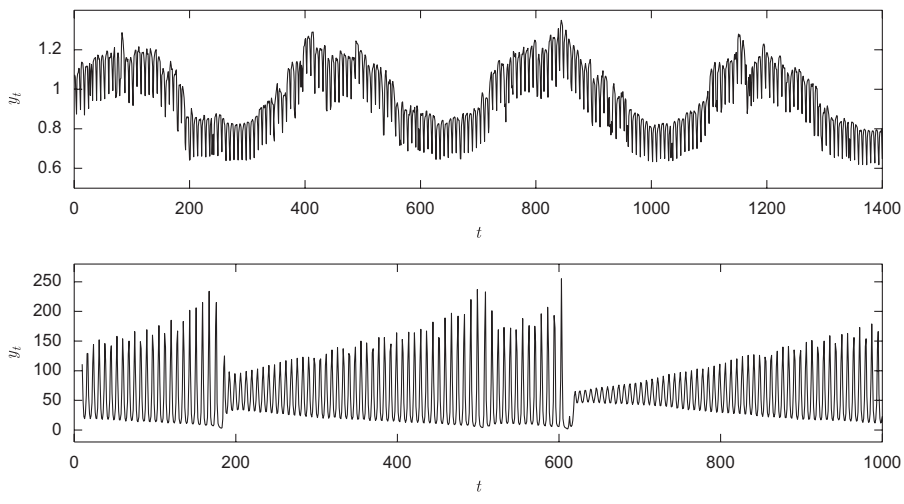

Fig. 2. The training data sets of the Poland electricity load (above) and the Santa Fe laser (below) time series.

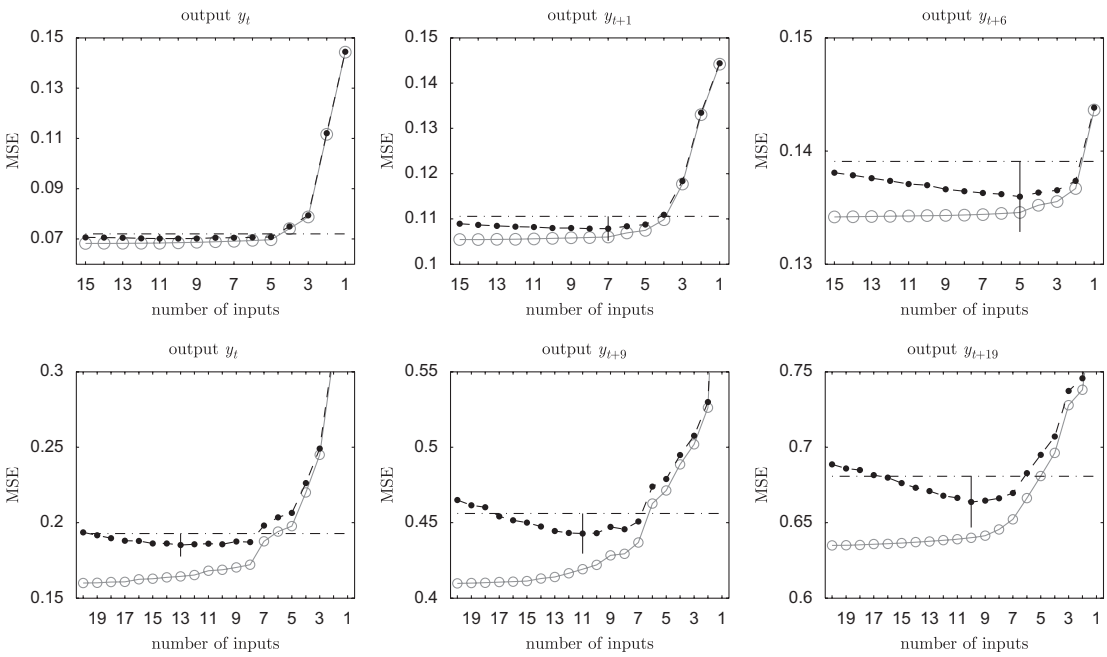

Fig. 3. Illustration of input selection procedure for the electricity (first row) and the Santa Fe (second row) time series. Training error (gray line with circles) and validation error (black line with dots) as a function of the number of inputs in the linear model. The vertical line marks the minimum validation error and the horizontal dash-dotted line represents the threshold, which is used in the selection of the final model.

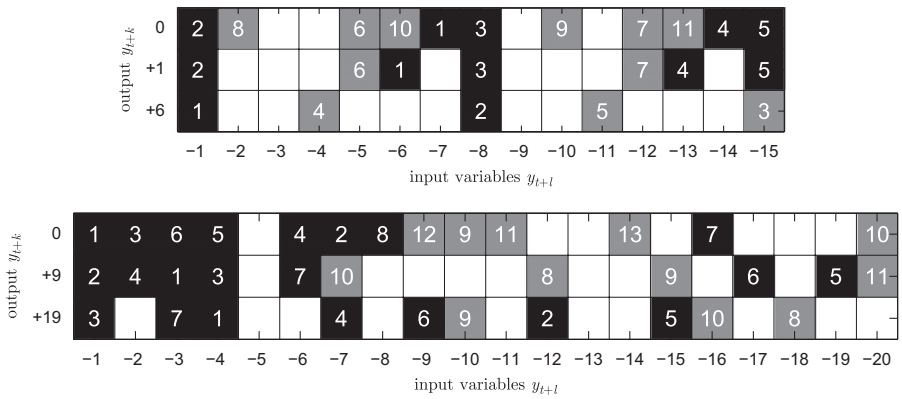

Fig. 4. The selected inputs for the electricity (above) and the Santa Fe (below) time series using SISAL. The outputs are in the vertical axis and the possible inputs in the horizontal axis. The selected inputs are denoted by black and gray rectangles on each row and the white numbers indicate the ranking of the inputs according to the importance in the prediction. The inputs marked by the gray rectangles are left out after thresholding ($E_{\mathrm{v}}^{\min} + s_{\mathrm{tr}}^{\min}$).

Fe laser time series it is hard to give a physical interpretation of the inputs. However, in all three prediction models the most important inputs (black rectangles) are at the beginning of the input horizon.

5.2. Phase II: nonlinear modeling

Based on the results of the input selection, the final models are trained and their prediction performance is evaluated using the test sets. The following prediction models are built for each k-step-ahead prediction case

(1) Linear model using the inputs $\mathscr{L}_{\mathrm{f}}$.
(2) MLP network with the inputs $\mathscr{L}_{\mathrm{f}}$ ($\mathscr{L}_{\mathrm{f}}$-MLP), number of neurons varied, no WD.
(3) MLP network with the inputs $\mathscr{L}_{\mathrm{v}}$ ($\mathscr{L}_{\mathrm{v}}$-MLP), number of neurons varied, no WD.
(4) MLP network with all the available inputs ($\mathscr{L}$-MLP), number of neurons varied, no WD.

(5) MLP network with all the available inputs ($\mathscr{L}$-MLP with WD), 20 neurons in the hidden layer, complexity controlled by WD.
(6) MLP network with inputs selected by the FS algorithm ($\mathscr{L}_{\mathrm{fs}}$-MLP), number of neurons varied, no WD.

The sets of inputs $\mathscr{L}_{\mathrm{f}}$ and $\mathscr{L}_{\mathrm{v}}$ were selected using SISAL (see Algorithm 1), thus the cases (2) and (3) corresponds to the two-phase modeling strategy. The other cases are used as baseline models to compare the prediction accuracies.

The number of neurons in the $\mathscr{L}_{\mathrm{f}}$-MLP, $\mathscr{L}_{\mathrm{v}}$-MLP, $\mathscr{L}$-MLP, and $\mathscr{L}_{\mathrm{fs}}$-MLP, networks were varied from 1 to 15. $\mathscr{L}$-MLP with WD was evaluated using 30 values of the regularization parameter λ, which were equally spaced on a logarithmic scale in the range $\lambda \in [10^{-4}, 10^{3}]$. The optimal number of neurons in $\mathscr{L}_{\mathrm{f}}$-, $\mathscr{L}_{\mathrm{v}}$-MLP, and $\mathscr{L}$-MLP and the optimal value for regularization parameter λ in $\mathscr{L}$-MLP with WD were selected using 10-fold cross-validation repeated five times to increase the reliability of the results. To decrease the computational burden in the FS algorithm, 2-fold cross-validation repeated five times was used to select the number of neurons. All the networks were trained using the Levenberg–Marquardt optimization algorithm [2]. Ten different initializations were used in the training of the networks in order to avoid local minima. The network having the smallest MSE was used in the analysis.

The training errors were monotonically decreasing as a function of increasing complexity in all the cases. In Fig. 5, the validation errors are shown as a function of λ for $\mathscr{L}$-MLP in the case of the electricity data (left panel) and the Santa Fe laser data (right panel). It is notable that the validation error curves are flatter in the prediction of y_t than in the long-term prediction cases. Thus, a sparser grid for λ could be used in the one-step-ahead prediction cases, especially with the Santa Fe time series.

In Fig. 6, the inputs selected by FS algorithm for all the k-step ahead prediction models are presented. The white numbers indicate the order, in which the inputs are selected. The smaller the white number the more important the corresponding input is in terms of the prediction. In the case of the electricity data, the FS algorithm selects nearly the same sets of inputs as SISAL selects into the sets $\mathscr{L}_{\mathrm{f}}$, see Figs. 4 and 6. Also, the same inputs are the most relevant ones according to SISAL and the FS algorithm in the prediction of the Santa Fe time series. Nearly the same sets of inputs were selected based on both the linear and the nonlinear selection method.

The prediction accuracy of the final models was evaluated using the test set, which is not used at all in the input selection

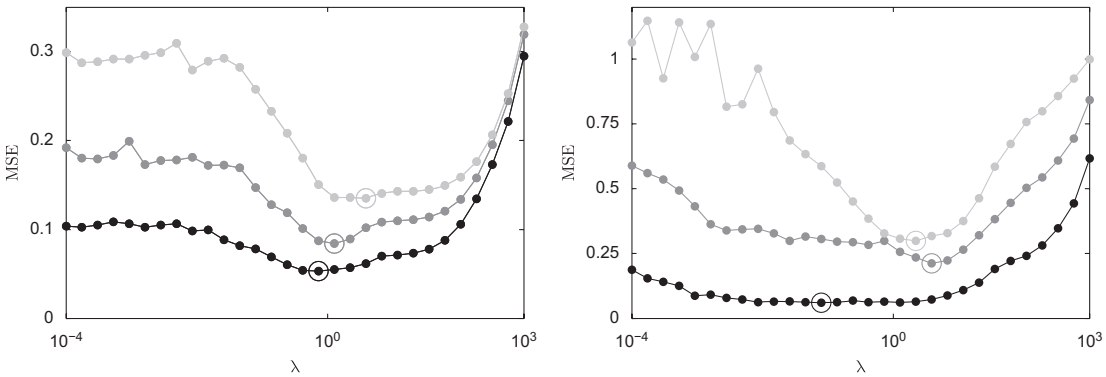

Fig. 5. Validation errors as a function of the regularization parameter λ in the prediction of y_t (black), y_{t+2} (dark gray), and y_{t+6} (light gray) of the electricity load (left panel) and in case of y_t (black), y_{t+9} (dark gray), and y_{t+19} (light gray) of the Santa Fe time series (right panel).

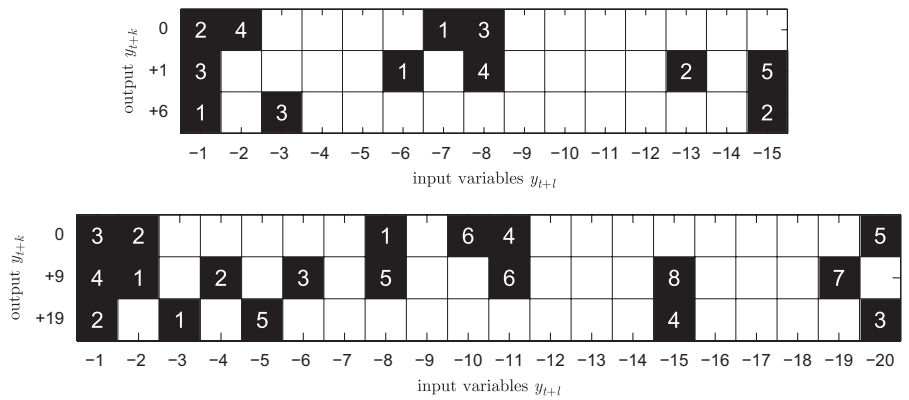

Fig. 6. The selected inputs for the electricity (above) and the Santa Fe (below) time series using the FS algorithm and MLP networks. The selected inputs are denoted by black rectangles on each row and the white numbers indicate the order, in which the inputs are added to the models.

Table 1
MSEs and standard deviations of MSEs (presented in parentheses) for the test set of the electricity data calculated using the bootstrap resampling procedure

Prediction model	One-day-ahead	Two-day-ahead	Seven-day-ahead
Sparse Linear	0.055(0.012) $n=5$	0.086(0.018) $n=5$	0.116(0.023) $n=2$
$\mathscr{L}_f$-MLP No WD	0.038(0.010) $n=5, p=6$	0.072(0.017) $n=5, p=5$	0.116(0.022) $n=2, p=7$
$\mathscr{L}_v$-MLP No WD	0.047(0.013) $n=11, p=4$	0.079(0.017) $n=7, p=4$	0.113(0.023) $n=5, p=3$
$\mathscr{L}$-MLP No WD	0.040(0.010) $n=15, p=4$	0.104(0.021) $n=15, p=4$	0.139(0.031) $n=15, p=3$
$\mathscr{L}$-MLP With WD	0.038(0.010) $\lambda=0.73, p=20$	0.079(0.016) $\lambda=1.27, p=20$	0.114(0.023) $\lambda=3.86, p=20$
$\mathscr{L}_{fs}$-MLP No WD	0.037(0.009) $n=4, p=6$	0.072(0.017) $n=5, p=5$	0.138(0.028) $n=3, p=3$

n is the number of inputs, p is the number of neurons, and λ is the regularization parameter.

Table 2
MSEs and standard deviations of MSEs (presented in parentheses) for the test set of the Santa Fe data calculated using the bootstrap resampling procedure

Prediction model	One-step-ahead	10-step-ahead	20-step-ahead
Sparse Linear	0.191(0.008) $n=8$	0.482(0.014) $n=7$	0.696(0.018) $n=7$
$\mathscr{L}_f$-MLP No WD	0.013(0.001) $n=8, p=2$	0.136(0.008) $n=7, p=4$	0.298(0.009) $n=7, p=5$
$\mathscr{L}_v$-MLP No WD	0.007(0.001) $n=13, p=2$	0.119(0.008) $n=11, p=2$	0.221(0.010) $n=10, p=4$
$\mathscr{L}$-MLP No WD	0.006(0.001) $n=20, p=2$	0.189(0.012) $n=20, p=2$	0.382(0.031) $n=20, p=2$
$\mathscr{L}$-MLP With WD	0.012(0.001) $\lambda=0.08, p=20$	0.100(0.006) $\lambda=3.86, p=20$	0.150(0.008) $\lambda=2.12, p=20$
$\mathscr{L}_{fs}$-MLP No WD	0.006(0.001) $n=6, p=4$	0.137(0.009) $n=8, p=3$	0.253(0.011) $n=5, p=5$

n is the number of inputs, p is the number of neurons, and λ is the regularization parameter.

phase and in the training of the final nonlinear predictors. One thousand bootstrap replications were drawn with replacement from the test set and the MSE was calculated for each replicated data set. The means and the standard deviations of replicated MSEs for each model and k-step-ahead prediction case were evaluated.

In Table 1, the results are shown for the Poland electricity time series. The sparse linear model with the inputs $\mathscr{L}_f$ is as equally accurate as the linear model with the inputs $\mathscr{L}_v$, or with all the inputs $\mathscr{L}$. This indicates that the selected inputs $\mathscr{L}_f$ are most informative, at least in the linear model. In the cases of one- and two-day-ahead prediction, $\mathscr{L}_f$-MLP and $\mathscr{L}_{fs}$-MLP trained without WD are the most accurate models. They decreases the MSE by 30% and 14% compared to the sparse linear model in the prediction of y_t and y_{t+1}, respectively. For seven-day-ahead prediction, $\mathscr{L}_v$-MLP without WD has the lowest prediction error, although the errors of the $\mathscr{L}_f$-MLP and the $\mathscr{L}$-MLP with WD are nearly the same. Thus, the sets of inputs $\mathscr{L}_f$ and $\mathscr{L}_v$ selected by SISAL include the most informative input variables for the nonlinear prediction models.

The prediction accuracies of all the models for the Santa Fe laser time series are collected in Table 2. Again the linear model with the inputs $\mathscr{L}_f$ is as equally accurate as the linear models with the inputs $\mathscr{L}_v$ and $\mathscr{L}$. However, it should be noted that the linear models clearly perform worse in this case than with the electricity load time series. The nonlinear prediction models reduce the MSEs significantly compared to the linear models. Therefore, the problems can be considered to be more nonlinear than the prediction of the electricity time series. In the prediction of y_t and y_{t+9} the sparse MLP networks ($\mathscr{L}_f$-MLP, $\mathscr{L}_v$-MLP, and $\mathscr{L}_{fs}$-MLP) were approximately as accurate as the MLP network with all the inputs ($\mathscr{L}$-MLP with WD). Only in the most nonlinear case, i.e. in the prediction of y_{t+19}, an increase in the number of inputs decreases the prediction error considerably. However, the selected inputs based on SISAL and the FS algorithm performs equally well in each case. This indicates that SISAL has succeeded to select the inputs for highly nonlinear problems as well.

The relative importances of the inputs (see Eq. (8)) in the prediction of the electricity load and the Santa Fe data are shown in Figs. 7 and 8. The SSD values are shown for the models based on SISAL ($\mathscr{L}_f$-MLP and $\mathscr{L}_v$-MLP) and for the most accurate model ($\mathscr{L}$-MLP with WD). Colors of the bars are the same as the colors of the rectangles in Fig. 4. This makes the comparison between the rankings of inputs based on SISAL and SSD values easier. The inputs are equally relevant according to SISAL and the SSD values if the black bars are the highest, the gray bars are the second highest, and the white bars should be almost invisible. The SSD values are averages over 1000 bootstrap replications of the test set.

In general, the inputs $\mathscr{L}_f$ are also the most relevant according to the SSD values in each prediction task of the electricity load (Fig. 7). In other words, the black bars are the highest ones. For instance, in the case of one-day-ahead prediction and $\mathscr{L}_f$-MLP, the inputs are ranked by SSD values in the order of decreasing importance as follows: $y_{t-1}, y_{t-7}, y_{t-15}, y_{t-8}$, and y_{t-14}. The ranking is nearly the same as with the linear models, see Fig. 4. In $\mathscr{L}$-MLP with WD, the five most important inputs in the order of decreasing importance are y_{t-1}, y_{t-7}, y_{t-2}, y_{t-8}, y_{t-15}. Four of them are the same as obtained with the linear models. Also, in the cases of two-day-ahead and seven-day-ahead prediction, the relative importances of inputs in the MLP networks are nearly the same as with the linear model.

Fig. 8 illustrates the SSD values in the prediction of Santa Fe laser data. Again in this case, the inputs $\mathscr{L}_f$ are the most relevant ones, except in the prediction of y_{t+19} by the $\mathscr{L}$-MLP with WD. In the prediction of y_t and y_{t+9}, the most informative input variables are at the beginning of the input horizon. It is noteworthy that all the MLP networks use effectively only the inputs y_{t-1} and y_{t-2} in the one-step-ahead prediction case. Comparing the importances of the inputs in $\mathscr{L}_f$-MLP and in $\mathscr{L}_v$-MLP, it can be seen that inputs pruned after the thresholding do not contribute significantly in the nonlinear models. Only in the prediction of y_{t+19} using the network $\mathscr{L}$-MLP with WD do the inputs pruned by SISAL (white bars) have clear contribution in the prediction. Nevertheless, the two most informative inputs (y_{t-3} and y_{t-4}) are also found by SISAL.

The influence of the inputs y_{t-1} and y_{t-8} in the prediction of y_t by $\mathscr{L}_f$-MLP are shown in Fig. 9. The shown result is for the test set. The values of $\partial y_t/\partial y_{t-1}$ are positive, which means that y_t tends to increase when y_{t-1} increases. Although the relative importance of

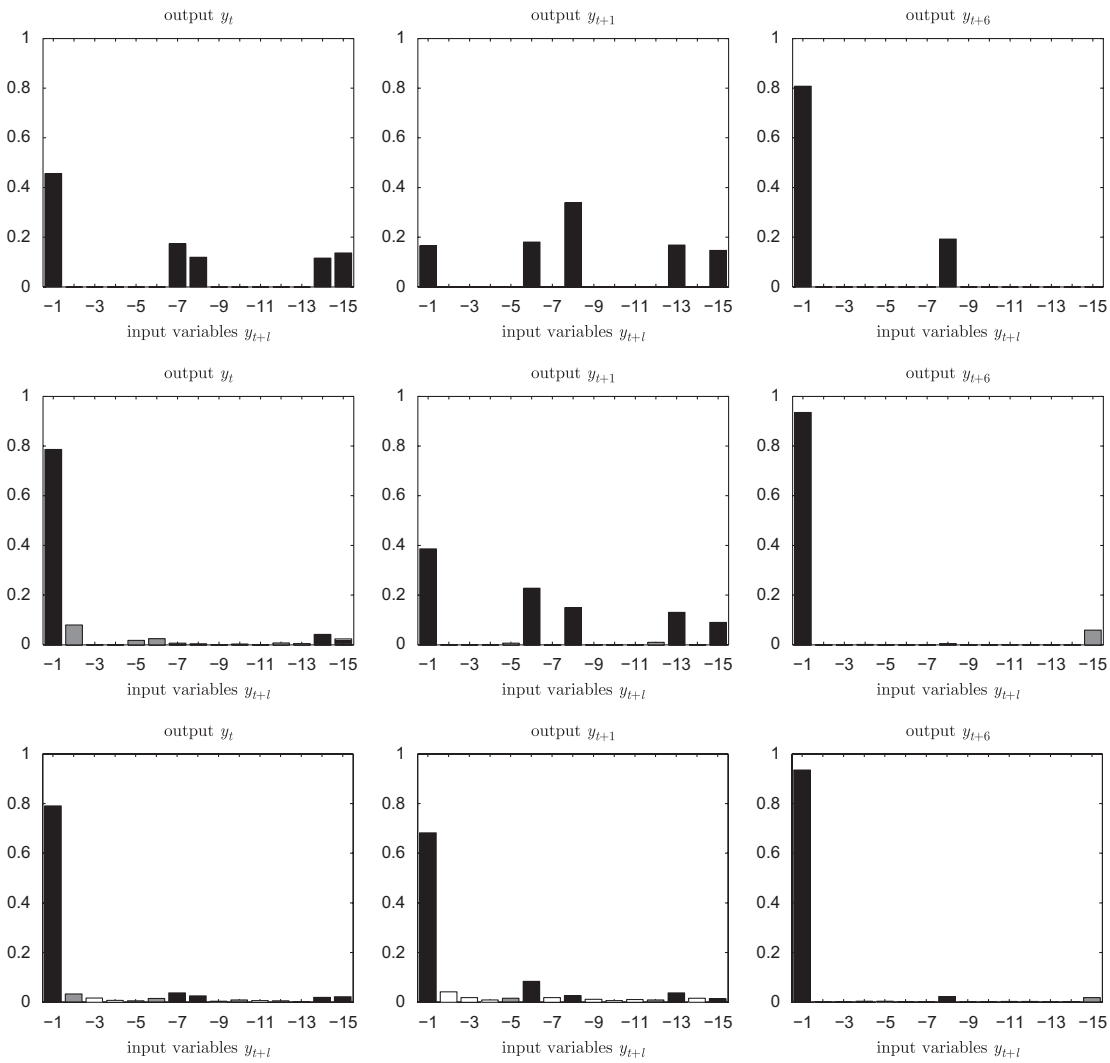

Fig. 7. Relative importances (normalized SSD values) of the input variables in $\mathscr{L}_f$-MLP (first row), $\mathscr{L}_v$-MLP (second row), and $\mathscr{L}$-MLP with WD (third row) in the prediction of the Poland electricity load time series.

y_{t-8} is notably smaller than y_{t-1}, the PADs $\partial y_t/\partial y_{t-8}$ are still clearly non-zero and negative. Thus, y_{t-8} also has a contribution to the prediction. When y_{t-8} increases the output y_t tends to decrease. The effect of the inputs can be considered to be nearly constant, which indicates that the dependency between the output y_t and the inputs y_{t-1} and y_{t-8} is nearly linear.

The PAD plots of the inputs y_{t-1} and y_{t-3} in the $\mathscr{L}_f$-MLP network in the prediction of 10-step-ahead in the Santa Fe data are presented in Fig. 10. The dependency between the output y_{t+9} and both the inputs y_{t-1} and y_{t-3} is certainly nonlinear. These inputs are the most and the third most important according to SSD values. They are also the first (y_{t-3}) and the second (y_{t-1}) in the ranking obtained by SISAL, which indicates, that perhaps unexpectedly, nonlinearly dependent inputs can be identified using the linear model. Based on these results and prediction accuracies it seems to be possible to find the most relevant inputs using SISAL in the nonlinear problem. It was shown in the experiments that the inputs found by SISAL are also meaningful in the nonlinear models and the inputs rejected by the SISAL do not significantly improve the prediction accuracy.

6. Summary and conclusions

A sequential input selection algorithm (SISAL) for a long-term time series prediction problem was presented. The prediction strategy applies a filter approach. Firstly, linear models are used in the time series prediction and a parsimonious set of input variables is selected using SISAL in the style of backward selection based on a cross-validation resampling procedure. Input variables

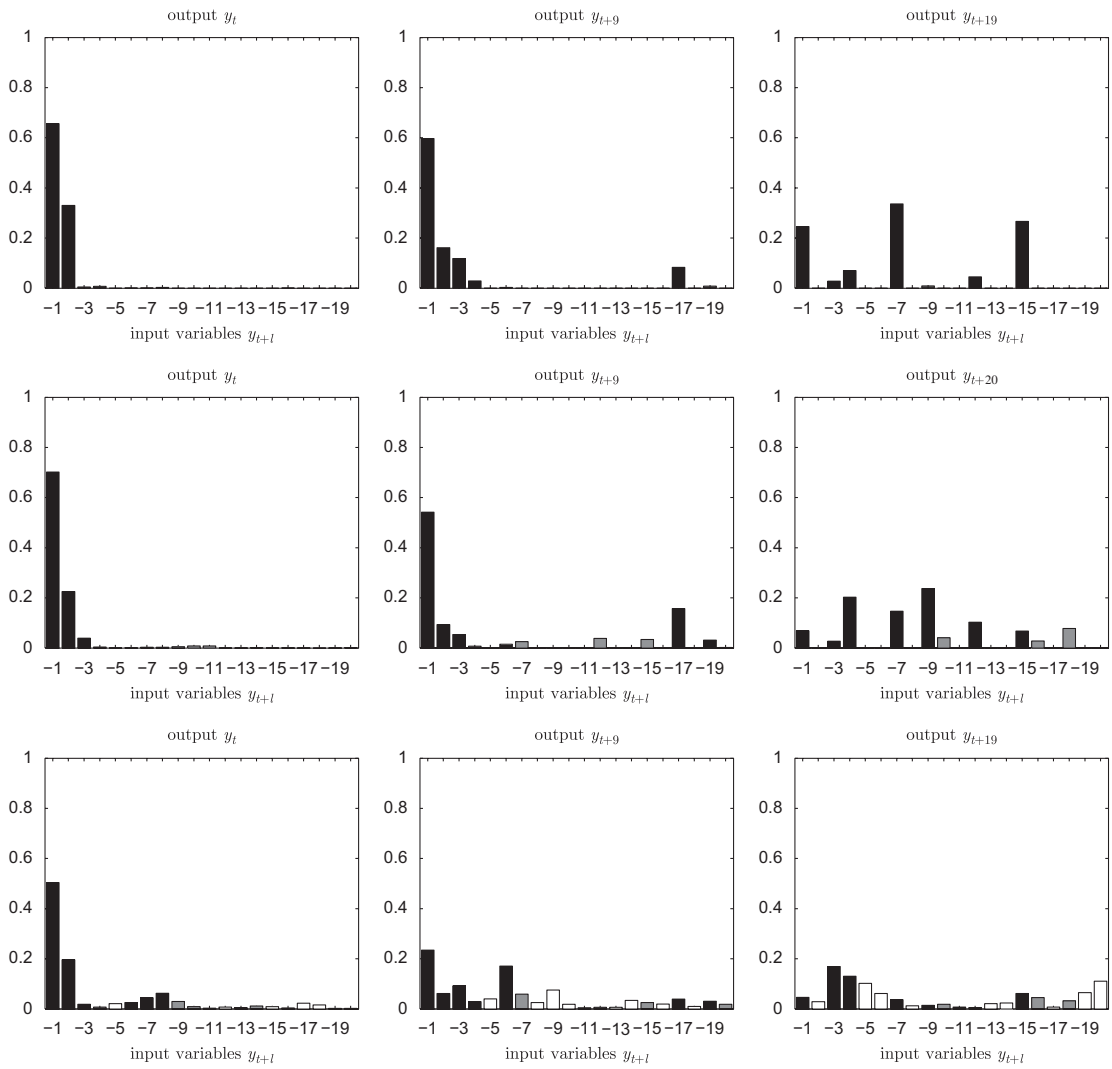

Fig. 8. Relative importances (normalized SSD values) of the input variables in $\mathscr{L}_1$-MLP (first row), $\mathscr{L}_v$-MLP (second row), and $\mathscr{L}$-MLP with WD (third row) networks in the prediction of the Santa Fe laser time series.

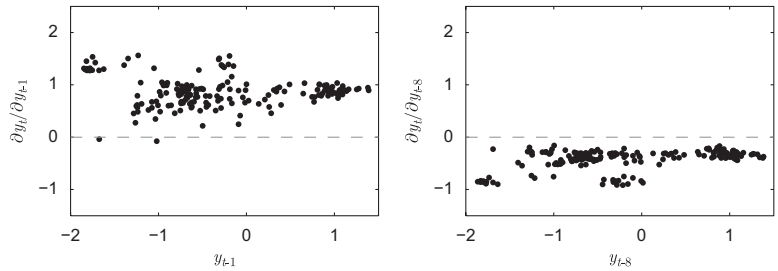

Fig. 9. The profiles of the inputs y_{t-1} (left) and y_{t-8} (right) in the $\mathscr{L}_1$-MLP network in one-day-ahead prediction of the electricity time series.

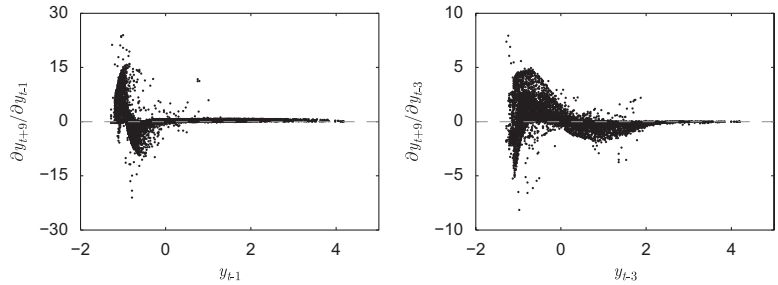

Fig. 10. The profiles of the inputs y_{t-1} (left) and y_{t-3} (right) in the $\mathscr{L}_1$-MLP network in 10-step-ahead prediction of the Santa Fe laser time series.

are successively dropped from the model one at a time according to a measure, which is based on the empirical distribution of model coefficients. The ranking measure is similar to the signal-to-noise-ratio. Inputs whose regression coefficients are markedly different from zero are retained. Secondly, the set of inputs selected by SISAL is used as inputs in a training of nonlinear predictor, such as a multilayer-perceptron network. SISAL produces sets of inputs that are much sparser than selecting a sparse set of inputs based on the minimum validation error. For nonlinear prediction, this reduces the number of weights in the network, which allows faster training of the network and makes it less prone to over-fitting. Sparsity of inputs also makes the nonlinear models more interpretable.

Experiments in the electricity load prediction and the prediction of the Santa Fe laser data set demonstrated that the two-phase strategy using input selection in a linear prediction model and subsequent nonlinear modeling using MLP yields accurate prediction. Based on the experiments, it can be concluded that as long as the linear model performs at least adequately, i.e. has a normalized mean squared error less than 0.5, the proposed two-phase modeling strategy gives competitive results. However, it is hard to define exact rules for the degree of nonlinearity, that the input selection based on the linear models works in general. Nonetheless, this could be studied by extensive simulation studies, since the correct models and phenomena would be completely known and correctness of the results easily justified. On the other hand, the findings should be tested using different well-known real world data sets. In the experiments it was also found that the importance of the inputs in the prediction obtained using the linear models reflected very well the importance of the inputs in the nonlinear models. When the prediction problem is highly nonlinear, such as in the case of the Santa Fe data with a long prediction horizon, training an appropriately regularized MLP network with all the inputs and a large number of neurons yields more accurate results than the proposed prediction strategy. Nevertheless, SISAL also found the most relevant input variables in this case.

Acknowledgments

This work has been supported by the Academy of Finland, Grants 207469 (SYSBIO) and 116853 (Analysis of dependencies of environmental time series data). Jarkko Tikka is also funded by the Graduate School of the Department of Computer Science and Engineering at the Helsinki University of Technology. The authors thank Dr. Amaury Lendasse for fruitful collaboration in the previous work and Paul Grouchy for comments on the manuscript.

References

[1] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, J. Mach. Learn. Res. 3 (2003) 1229–1243.
[2] C. Bishop, Neural Networks in Pattern Recognition, Oxford University Press, Oxford, 1996.
[3] C. Chatfield, Time Series Forecasting, Chapman & Hall, CRC, London, Boca Raton, FL, 2002.
[4] J. Copas, Regression, prediction and shrinkage, J. R. Stat. Soc. (Ser. B) 45 (3) (1983) 311–354.
[5] C. Dal Cin, L. Moens, P. Dierickx, G. Bastin, Y. Zech, An integrated approach for real-time flood-map forecasting on the Belgian Meuse river, Natural Hazards 36 (1–2) (2005) 237–256.
[6] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, CRC, London, Boca Raton, FL, 1993.
[7] K. Fukunaga, Introduction to Statistical Pattern Recognition. Computer Science and Scientific Computing, second ed., Academic Press, New York, 1990.
[8] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecol. Modelling 160 (3) (2003) 249–264.
[9] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[10] J.D. Hamilton, Analysis of time series subject to changes in regime, J. Econometrics 45 (1990) 39–70.
[11] J.D. Hamilton, Time Series Analysis, Princeton University Press, Princeton, NJ, 1994.
[12] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning—Data Mining, Inference and Prediction. Springer Series in Statistics, Springer, Berlin, 2001.
[13] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[14] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359–366.
[15] Y. Ji, J. Hao, N. Reyhani, A. Lendasse, Direct and recursive prediction of time series using mutual information selection, in: J. Cabestany, A. Prieto, F. Sandoval (Eds.), Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN 2005), Lecture Notes in Computer Science, vol. 3512, Springer, Berlin, 2005, pp. 1010–1017.
[16] R. Kohavi, G. John, Wrappers for feature selection, Artif. Intell. 97 (1997) 273–324.
[17] A. Lendasse, J. Lee, V. Wertz, M. Verleysen, Forecasting electricity consumption using nonlinear projection and self-organizing maps, Neurocomputing 48 (2002) 299–311.
[18] L. Ljung, System Identification—Theory for the User, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[19] M. Sulkava, J. Tikka, J. Hollmén, Sparse regression for analyzing the development of foliar nutrient concentrations in coniferous trees, Ecol. Modelling 191 (1) (2006) 118–130.
[20] J. Tikka, J. Hollmén, Long-term prediction of time series using a parsimonious set of inputs and LS-SVM, in: A. Lendasse (Ed.), Proceedings of the First Symposium on Time Series Prediction (ESTSP 2007), 2007, pp. 87–96.
[21] J. Tikka, J. Hollmén, A. Lendasse, Input selection for long-term prediction of time series, in: J. Cabestany, A. Prieto, F. Sandoval (Eds.), Proceedings of the Eighth International Work-Conference on Artificial Neural Networks (IWANN 2005), Lecture Notes in Computer Science, vol. 3512, Springer, Berlin, 2005, pp. 1002–1009.
[22] J. Tikka, A. Lendasse, J. Hollmén, Analysis of fast input selection: application in time series prediction, in: Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN 2006), Lecture Notes in Computer Science, vol. 4132, Springer, Berlin, 2006, pp. 161–170.
[23] M. Verleysen, D. François, The curse of dimensionality in data mining and time series prediction, in: J. Cabestany, A. Prieto, F. Sandoval (Eds.),

Proceedings of the Eighth International Work-Conference on Artificial Neural Networks (IWANN 2005), Lecture Notes in Computer Science, vol. 3512, Springer, Berlin, 2005, pp. 758–770.
[24] W.W.S. Wei, Time Series Analysis—Univariate and Multivariate Methods, second ed., Pearson Education, 2006.
[25] A.S. Weigend, N.A. Gershenfeld (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, MA, 1994.

Jarkko Tikka received the degree of M.Sc. (Tech.) at the Department of Automation and Systems Technology at the Helsinki University of Technology in 2004. He is pursuing the Ph.D. degree from Helsinki University of Technology with a dissertation on construction of comprehensible dependency structures from multivariate data. Since 2004, he has been working as a Research Scientist at the Department of Information and Computer Science (formerly Laboratory of Computer and Information Science), Helsinki University of Technology. His research interests include sparse models in multivariate data analysis, machine learning, and data exploration.

Jaakko Hollmén received the degrees of M.Sc. (Tech.) in 1996, Lic.Sc. (Tech.) in 1999, and D.Sc. (Tech.) in 2000, all at the Department of Computer Science and Engineering at the Helsinki University of Technology. He has worked at the Department of Information and Computer Science (formerly Laboratory of Computer and Information Science) at the Helsinki University of Technology (TKK) since 2000. Before joining TKK, he worked at the Siemens Corporate Technology in Munich, Germany. Currently, he is a Chief Research Scientist at TKK. His research interests include theory and practice of data analysis and data mining, especially their applications in bioinformatics and environmental informatics. Jaakko Hollmén is a Senior Member of IEEE.