Long-term Time Series Prediction using OP-ELM

Alexander Grigorievskiy [a,*], Yoan Miche [a], Anne-Mari Ventelä [b], Eric Séverin [c] and Amaury Lendasse [a,d,e]

[a] *Department of Information and Computer Science, Aalto University School of Science, FI-00076 Aalto, Finland*

[b] *Pyhäjärvi Institute, Sepäntie 7, FI-27500, Kauttua, Finland*

[c] *University of Lille 1, IAE, 104 avenue du peuple Belge, 59043 Lille cedex, France*

[d] *IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain*

[e] *Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia-San Sebastián, Spain*

Abstract

In this paper, an Optimally Pruned Extreme Learning Machine (OP-ELM) is applied to the problem of long-term time series prediction. Three known strategies for the long-term time series prediction i.e. Recursive, Direct and DirRec are considered in combination with OP-ELM and compared with a baseline linear least squares model and Least-Squares Support Vector Machines (LS-SVM). Among these three strategies DirRec is the most time consuming and its usage with nonlinear models like LS-SVM, where several hyperparameters need to be adjusted, leads to relatively heavy computations. It is shown that OP-ELM, being also a nonlinear model, allows reasonable computational time for the DirRec strategy. In all our experiments, except one, OP-ELM with DirRec strategy outperforms linear model with any strategy. In contrast to the proposed algorithm, LS-SVM behaves unstably without variable selection. It is also shown that there are no superior strategy for OP-ELM: any of three can be the best. In addition, prediction accuracy of ensemble of OP-ELM is studied and it is shown that averaging predictions of ensemble can improve the accuracy (Mean Square Error) dramatically.

Key words: Time series prediction, ELM, OP-ELM, LS-SVM, Recursive strategy, Direct strategy, DirRec strategy, Ordinary Least Squares

* Corresponding author
Email address: alexander.grigorevskiy@aalto.fi (Alexander Grigorievskiy).

1 Introduction

Time series prediction (TSP) has already been studied for a long time and has a variety of applications [1]. For instance, it is used for climate forecasting, prediction of economical characteristics, stock market prediction, electricity consumption, sales forecasting and many others.

Since time series prediction arises so frequently in applications, large number of methods has been developed for this task. Relatively recent overview of various methods and future directions is given in [2]. Historically, statistical linear methods dominated in TSP. In particular, ARIMA based modeling became widely adopted after the remarkable book [3]. The complete methodology for model selection, parameter optimization and prediction was introduced there and it is still widely used. ARIMA models time series (or it's differences) as a linear combination of previous values of time series and previous values of noise (often called innovations). However, real time series come from many different sources, and have very different properties. So it is obvious that there is no single best approach for time series modeling. Not surprisingly, other methods which may outperform classical methods, have emerged.

Neural network (NN) methods have attracted significant attention for time series prediction problems [4]. NNs are general nonlinear regression technique which can be applied to time series. In addition, they are able to relax some assumptions made by classical methods, e.g. model linearity and Gaussian distribution of noise. In contrast to ARIMA(p,n,q) model where fine tuning of model hyperparameters - (p,n,q) is required for obtaining good forecasts, neural networks allow to avoid this complication. Therefore, the way the forecasting process is done may be changed. Instead of many hours of work of a statistician (quite often with a domain knowledge) trying to select the right model and adjust hyperparameters, modeler without domain knowledge is able to apply NN and obtain competitive results. There are situations where intensive human involvement or large computational time is not affordable. Neither we nor other authors claim that neural networks are generally better method than classical statistical methods, but definitely they and other computational intelligence methods have shown its viability [4].

In time series prediction one can distinguish one-step-ahead prediction and long-term prediction. As it is clear from these names, in one-step-ahead prediction interest constitutes only estimation of the next single value ahead, while in the long-term prediction estimations of multiple future values are required. Quite often researchers address these problem separately [5], [6] since accumulation of errors and increasing uncertainties [7] make long-term prediction inherently more difficult problem. In this paper we consider long-term time series prediction.

There exist three universal strategies for long-term time series prediction: Recursive strategy, Direct and DirRec strategy. Recently, another strategy [8] was introduced but we do not study it here. A detailed description of prediction strategies is given in Section 2. Strategies differ in how we estimate future values using the past values. There is no definite indication of superiority of one strategy over the others, as has been shown in [6],[7].

Earlier works have shown that variable selection is needed to improve the accuracy of long-term predictions. For instance, it has been shown [7] that DirRec strategy with variable selection and K-Nearest Neighbours (K-NN) model is beneficial in terms of accuracy. As a variable selection method forward-backward algorithm is used. Especially, unimportant variables (features) can deteriorate performance of the models which are very sensitive to those, for example K-NN [9].Variable selection methods can be very time consuming especially if we consider wrapper class of methods [7].Thus, the motivation for our approach is the desire to avoid computationally expensive variable selection.

In this paper, we propose to use OP-ELM model which is more robust to irrelevant or correlated variables due to internal pruning of inessential neurons [10]. Performance of OP-ELM has been shown to be comparable to other popular nonlinear models like Support Vector Machines (SVM), Multi-Layer Perceptron (MLP), Gaussian Processes (GP), etc. [10]. Moreover, for other nonlinear models fine tuning of hyperparameters is necessary, for example, (C, σ) in Least-Squares Support Vector Machine with Gaussian kernel. This is often done through cross-validation on a grid in parameters space. So, for each point on a grid new model must be trained and accuracy needs to be computed on a validation set, The point in parameters space with the highest accuracy is selected as a final value for parameters. Therefore, to select good values of (C and σ) as many LS-SVMs as many points in the grid are, need to be trained. Furthermore, coming back to time series prediction, this grid search is necessary for every consecutive future value prediction (for Direct and DirRec strategies). Thus, this parameters selection procedure dramatically increases computational time and many well known nonlinear machine learning models may become impractical for long-term time series prediction.

OP-ELM model is described in more details in Section 3. Further more, predictions made by ensemble of 100 [11] OP-ELMs are analyzed. Here we show that ensemble averages can significantly improve the prediction accuracy. It is also empirically established that none of the predictive strategies is always superior when ensemble method is used.

Application of various types of ELMs to time series prediction or similar problems has been studied recently as well. Some references are [12], [13], [14]. However, here we address the problem of long-term time series prediction with emphasis on computational time and prediction accuracy. Combination of ELM based model and DirRec prediction strategy has not been investigated before.

In the next section, three strategies used in time-series prediction are explained in detail. In the Section 3, concepts of ELM and OP-ELM are presented. After that, experiment Section 4 follows.

2 Long-term time series prediction

There are three main strategies for long-term time series prediction as mentioned earlier. Here an overview of each one of them is presented.

2.1 Recursive strategy

The Recursive strategy for long-term time series prediction is a simple and intuitive strategy. The goal is to build the model which estimates the next value by using r previous values. Here r, which is called regressor size, is a hyperparameteter of a model, and can be determined via cross-validation or other methods for selection of hyperparameters. Section 4 explains how r is selected for datasets in this article. Thus, on the first step the model computes the following estimation:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \dots y_{t-r+1}) \tag{1}$$

To predict the second value, the first predicted value is introduced into the model:

$$\begin{aligned} &\hat{y}_{t+2} = f(\hat{y}_{t+1}, y_t, \dots y_{t-r+2}) \\ &\vdots \\ &\hat{y}_{t+r+1} = f(\hat{y}_{t+r}, \hat{y}_{t+r-1}, \dots \hat{y}_{t+1}) \end{aligned} \tag{2}$$

The process can be continued until we predict as many values as needed. It is clear that prediction of $t + r + 1$-th value is based only on estimations $\hat{y}_{t+r}, \cdots \hat{y}_{t+1}$, and does not depend on any original values of time series. Since each prediction has some error, errors accumulate with the increase of the prediction horizon.

2.2 Direct strategy

In the Direct strategy, the regressor size r is also a hyperparameter of the model. The goal is to directly predict p steps ahead using regressors $y_t, y_{t-1}, \dots y_{t-r+1}$. Later in this article p is called prediction horizon. Hence, for every next future value training of a separate model is needed, that is:

$$\begin{aligned}
\hat{y}_{t+1} &= f_1(y_t, y_{t-1}, \dots y_{t-r+1}) \\
\hat{y}_{t+2} &= f_2(y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\vdots \\
\hat{y}_{t+p} &= f_p(y_t, y_{t-1}, \dots y_{t-r+1})
\end{aligned} \tag{3}$$

It is seen, that predictions are always based on true values of time series, but the time lag between regressors and prediction value is constantly growing. This often causes a gradual growth of prediction error. In addition, number of training samples decreases for the next predicted value. However, Direct strategy is usually more accurate than Recursive [7].

2.3 DirRec strategy

The DirRec strategy has been introduced in [15] and combines both Recursive and Direct strategies. The number of regressors is not constant anymore. On the first step, DirRec strategy coincides with the Direct strategy, then all predicted values serve as new regressors and the order of the model grows. In mathematical form it is written as:

$$\begin{aligned}
\hat{y}_{t+1} &= f_1(y_t, y_{t-1}, \dots y_{t-r+1}) \\
\hat{y}_{t+2} &= f_2(\hat{y}_{t+1}, y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\vdots \\
\hat{y}_{t+p} &= f_p(\hat{y}_{t+p-1}, \dots, \hat{y}_{t+1}, y_t, y_{t-1}, \dots y_{t-r+1})
\end{aligned} \tag{4}$$

As in Direct strategy for every future prediction the corresponding model needs to be trained. So, the complexity of the training is proportional to the number of values to be predicted p. It has been shown [15] that in general DirRec strategy with variable selection have superiority over two other strategies when the model f is nonlinear.

The goal of this article is to show that this statement holds without variable selection when for the role of a model f OP-ELM is taken. The motivation for this is that OP-ELM intrinsically performs a variable selection in a hidden space.

3 OP-ELM for Time Series Prediction

3.1 Extreme Learning Machine (ELM)

The ELM algorithm was originally proposed by Guang-Bin Huang *et al* in [16] and it makes use of the Single Layer Feedforward Neural Network (SLFN). The main concept behind the ELM lies in the random initialization of the SLFN weights and biases. Therefore, the input weights and biases do not need to be adjusted and it is possible to calculate explicitly the hidden layer output matrix and hence the output weights. The network is obtained with very few steps and very low computational cost.

Consider a set of M distinct samples $(\mathbf{x}_i, \mathbf{y}_i)$ with $\mathbf{x}_i \in \mathbb{R}^{d_1}$ and $\mathbf{y}_i \in \mathbb{R}^{d_2}$; then, a SLFN with N hidden neurons is modeled as the following sum

$$\sum_{i=1}^{N} \boldsymbol{\beta}_i f(\mathbf{w}_i^T \mathbf{x}_j + b_i), \quad 1 \leq j \leq M, \tag{5}$$

with f being the activation function, $\mathbf{w}_i$ the input weights, b_i the biases and $\boldsymbol{\beta}_i$ the output weights.

ELM is constructed in a way that it perfectly approximates the given output data:

$$\sum_{i=1}^{N} \boldsymbol{\beta}_i f(\mathbf{w}_i^T \mathbf{x}_j + b_i) = \mathbf{y}_j, \quad 1 \leq j \leq M, \tag{6}$$

which writes compactly as $\mathbf{HB} = \mathbf{Y}$, with

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1\mathbf{x}_1 + b_1) & \cdots & f(\mathbf{w}_N\mathbf{x}_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1\mathbf{x}_M + b_1) & \cdots & f(\mathbf{w}_N\mathbf{x}_M + b_N) \end{pmatrix}, \tag{7}$$

and $\mathbf{B} = (\boldsymbol{\beta}_1^T \ldots \boldsymbol{\beta}_N^T)^T$ and $\mathbf{Y} = (\mathbf{y}_1^T \ldots \mathbf{y}_M^T)^T$.

The way to calculate the output weights $\mathbf{B}$ from the knowledge of the hidden layer output matrix $\mathbf{H}$ and target values, is proposed with the use of a Moore-Penrose generalized inverse of the matrix $\mathbf{H}$, denoted as $\mathbf{H}^{\dagger}$ [17].

Theoretical proofs and a more thorough presentation of the ELM algorithm are detailed in the original paper [16], [18].

However, the ELM tends to have problems when irrelevant or correlated variables [10]. For this reason, it is proposed in the OP-ELM methodology, to perform a pruning of the irrelevant variables, via pruning of the related neurons of the SLFN built by the ELM.

3.2 Optimally Pruned ELM (OP-ELM)

The Optimally Pruned Extreme Learning Machine (OP-ELM) is made of three main steps summarized in the following algorithm:

Algorithm 1 OP-ELM

```
Given a training set $(\mathbf{x}_i, \mathbf{y}_i), \mathbf{x}_i \in \mathbb{R}^{d_1}, \mathbf{y}_i \in \mathbb{R}^{d_2}$.
1: - Build a regular ELM model with initially large number of neurons
2: - Rank neurons using multiresponse sparse regression (LARS regression if
   output is one dimensional)
3: - Use leave-one-out validation to decide how many neurons to prune.
```

The very first step of the OP-ELM methodology is the actual construction of the SLFN using the original ELM algorithm with a large number of neurons (100 in our experiments). Second and third steps are presented in more details in the next two subsections and are meant for an effective pruning of the possibly unuseful neurons of the SLFN.

In the original OP-ELM algorithm [10] it was suggested to use a combination of three different types of kernels, for robustness and more generality, where the original ELM proposed to use only sigmoid kernels. Three types are linear, sigmoid and Gaussian kernels. Having the linear kernels included in the network helps when the problem is linear or nearly linear. Experiments in this paper are conducted using only linear and sigmoid neurons. Gaussian neurons are not used because preliminary tests showed that their usage does not improve the results.

The sigmoid weights are drawn randomly from a uniform distribution in the interval $[-5, 5]$. This allows neurons to operate in the right regime when input data is normalized with zero mean and unit variance.

3.2.1 Multiresponse Sparse Regression: MRSR

In order to get rid of the useless neurons of the hidden layer, the Multiresponse Sparse Regression, proposed by Timo Similä and Jarkko Tikka in [19], is used.

The main idea of the algorithm is the following: Denote by $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_m]$ the $n \times m$ regressor matrix. MRSR adds each column of the regressor matrix one by one to the model $\hat{\mathbf{Y}}^k = \mathbf{X}\mathbf{W}^k$, where $\hat{\mathbf{Y}}^k = [\hat{\mathbf{y}}_1^k \dots \hat{\mathbf{y}}_p^k]$ is the target approximation of the model. The $\mathbf{W}^k$ weight matrix has k nonzero rows at kth step of the MRSR. With each new step a new nonzero row, and a new column of the regressor matrix is added to the model. More specific details of the MRSR algorithm can be found from the original paper [19].

It can be noted that the MRSR is mainly an extension of the Least Angle Regression (LARS) algorithm [20] for the multi-output case. Because of the ranking provided by the MRSR, it is used to rank the neurons of the model. The target is the actual output $\mathbf{y}_i$, while the "variables" considered by the MRSR are the outputs of the kernels $\mathbf{h}_i = \mathrm{Ker}(\mathbf{x}_i^T)$, the columns of $\mathbf{H}$.

There has been the decent amount of work dedicated to other methods of neuron selection, namely forward selection (FS) [21] and [22]. However, forward selection is known to be unstable procedure in a sense that small variation in the data can cause a large variation in model parameters [23]. This was one of the reasons for developing the original Lasso and LARS algorithms. The comparison research [24] shows that neither LARS (or Lasso) nor FS is generally better than the other method. Therefore, both deserve an attention.

3.2.2 Leave-One-Out (LOO)

Since the MRSR only provides a ranking of the neurons, the decision over the actual best number of neurons for the model is taken using a Leave-One-Out (LOO) validation method.

In general, computing of LOO error can be very time consuming because we need to take apart each sample, train the model ignoring it, and then compute an error on this sample. Thus, training the model as many times as many samples we have is required. Fortunately, for linear systems there exists a closed formula which provides an exact LOO error without retraining the model for each sample. This is called PRESS (PREdiction Sum of Squares) statistic, see [25] and [26] for details of this formula and its implementation.

The PRESS formula, which exactly calculates LOO error is:

$$\varepsilon^{\mathrm{PRESS}} = \|\mathbf{D}\,(\,\mathbf{y} - \mathbf{H}(\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{H}^{T}\mathbf{y}\,)\|_2^2, \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{D}_{ii} = \dfrac{1}{1 - (\mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T})_{ii}}$. Extension of this formula to multiple output case is straightforward.

Calculation of $\varepsilon^{\text{PRESS}}$ takes $O(N^3) + O(MN^2)$ operations. It is M (number of samples) times smaller than the naive implementation of LOO error. The leading contribution to the complexity is the inverse $(H^T H)^{-1}$. This time can be further reduced by sequentially updating the inverse $(\mathbf{H^T H})^{-1}$ from the previous step.

$$(\mathbf{H^T H})^{-1} = B_{11} - \frac{B_{12}B_{21}}{B_{22}} \tag{9}$$

where B_{11} is the submatrix of $(\mathbf{H^T H})^{-1}$ from the previous step, where last column and last row are excluded. B_{22} is the bottom-right element of the $(\mathbf{H^T H})^{-1}$, and B_{12} and B_{21} are the last row and last column respectively, excluding the bottom-right element. The Equation (9) can easily be derived from inverse of block matrix formula. The complexity of this update is only N^2 operations, hence the complexity of overall $\varepsilon^{\text{PRESS}}$ calculation becomes $O(N^2) + O(MN)$.

An alternative strategy to speed up LOO computation is to do pruning in batches of neurons (for instance in five neurons). This strategy has been used in our experiments.

At first the LOO decreases because insignificant neurons tend to overfit the model. After pruning of several batches, LOO increases. At this point pruning is stopped and OP-ELM is considered to be trained. In the end, a SLFN using a mix of linear and sigmoid kernels is obtained, with a highly reduced number of neurons, all within a small computational time. Comparison of running times for OP-ELM and linear model is given in the Subsection 4.2.

4 Experimental results

The method is applied to three different time series: Sea-water temperature [27], Sun Spots [28] and Santa Fe A [29]. The first one is a weekly measurements of sea water temperature during several years, there are 875 measurements in total. The second is one of the oldest time series in history; it provides monthly averages of a number of dark spots on the sun from year 1749 until 2012, there are 3161 measurements in total. Santa Fe A is a dataset recorded from a far-infrared-laser in a chaotic state and it is explicitly divided into training set (1000 points) and test set (9093 points). We would like to emphasize that these time series are taken from completely different domains, so our method is applied to time series with completely different properties and behavior.

Usually, in time series prediction the number of regressors to use is unknown and it has to be estimated. Here, *a priori* information is used to select appropriate regressor sizes. For the Sea-water temperature dataset regressors of sizes 15 and 50 were analyzed [30]. For the Sun Spots dataset number of regressors equals 28 and is estimated by the following procedure. Linear model is trained for various number of regressors and the number with minimal leave-one-out validation error is taken. For the Santa Fe A dataset number of regressors equals 12 [31] and is known to be enough to predict this time series reasonably well.

4.1 Estimation of generalization accuracies of OP-ELMs trained by different strategies

To estimate accuracies of different models, generalization errors need to be calculated. For this, datasets are divided into two parts i.e. training part and test part. Training part is used to train the model, while test part - to compute predictions and compare them with original values. Mean square error (MSE) criteria serves to compare true and predicted values. For Santa Fe dataset separation into training and test sets is done by the providers of this time series. Two other datasets are divided approximately into equal parts one for training and one for test. For the Sea-water temperature data training and test parts are swapped and results are averaged. Note, that leave-one-out validation which is build-in into OP-ELM is done during training phase, so it uses only the training set.

Predictions are calculated for each subsequence of a test set which length equals regressor size. In other words, if a regressor size is r, for each r consecutive values of a test set predictions up to prediction horizon are calculated. For a certain number of steps ahead prediction, Mean Square Errors (MSE) are averaged over all subsequences of size r, and finally obtained MSEs are averaged over all number of steps ahead up to prediction horizon. Therefore, for an experiment with a single OP-ELM (or linear model) one number is obtained - twice averaged MSE which characterizes the prediction accuracy.

Least-Squares Support Vector Machine (LS-SVM) is a competitive technique which has been intensively used for nonlinear modeling [32]. Experiments have been performed by a famous LS-SVM Matlab Toolbox [33]. LS-SVM has an advantage that training converges to solving a linear system in dual space. So, it is especially interesting to compare it against our method. However for a good model, hyperparameters need to be adjusted. There are two hyperparam-

Sea-water temperature time series				
	Linear Model	**LS-SVM**	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 15, prediction horizon = 15			
Recursive	**2.440**	2.654 2.639±0.178	**2.331 ± 0.346**	**2.156**
Direct	2.887	**2.618** 2.612±0.062	2.664 ± 0.123	2.460
DirRec	2.873	2.628 2.637±0.092	2.410 ± 0.118	2.324
	Regressor size = 50, prediction horizon = 50			
Recursive	**2.848**	6.515 *not computed*	3.072 ± 1.181	**2.364**
Direct	3.308	**5.134** *not computed*	3.5260 ± 0.200	3.030
DirRec	3.264	6.611 *not computed*	**2.860 ± 0.133**	2.698
	Regressor size = 15, prediction horizon = 50			
Recursive	3.938	6.752 4.952±1.061	3.732 ± 0.628	3.202
Direct	**3.686**	**3.212** 3.162±0.064	3.480 ± 0.102	3.136
DirRec	3.702	3.986 4.091±0.093	**3.241 ± 0.094**	**3.069**

Table 1

Mean Square Errors(MSE) for Sea-water temperature dataset. Different regressor sizes and prediction horizons are considered. Results of different models are given in column-wise. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented. In small font bagging results for LS-SVM are presented.

eters C - regularization hyperparameter and σ - Gaussian kernel parameter. They have been adjusted by 10 fold cross validation and a grid search as implemented in the toolbox. In addition, bagging [34] with 50 bootstrap samples is applied to LS-SVM and results are listed in the same column in the smaller font. Bagging for linear model has been tried but the performance is very close to the complete data performance as mentioned in the original bagging paper, therefore they are not shown here.

Results of experiments are given in Tables 1,2,3. Because of a randomness involved in the OP-ELM definition, many instances of OP-ELMs need to be studied in order to estimate its performance. For every set of parameters 100 [11] OP-ELMs are build, for each of those MSE described in previous paragraph is computed. Averages and standard deviations of these MSEs are presented in the third columns of the tables. In addition, arithmetic mean

Sun spots time series				
	Linear Model	**LS-SVM**	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 28, prediction horizon = 12			
Recursive	496.811	1661.7 1705.574±23.090	491.047 ± 11.2947	471.874
Direct	**493.389**	**1413.4** 1464.967±29.881	487.127 ± 5.908	**456.372**
DirRec	493.485	1764.5 1804.640±20.696	**482.166 ± 4.494**	467.993
	Regressor size = 28, prediction horizon = 24			
Recursive	785.610	2063.3 2053.930±30.014	748.210 ± 30.327	716.536
Direct	**772.982**	**1527.3** 1570.166±23.591	739.334 ± 8.893	**692.122**
DirRec	773.778	2068.1 2086.731±16.874	**734.116 ± −8.245**	713.702
	Regressor size = 28, prediction horizon = 28			
Recursive	891.363	2206 2170.886±45.850	832.184 ± 39.071	791.281
Direct	**874.878**	**1554** 1595.346±29.829	825.926 ± 11.614	**773.803**
DirRec	876.144	2149.8 2165.569±15.847	**824.160 ± 10.671**	801.817

Table 2

Mean Square Errors(MSE) for Sun spots dataset. Different regressor sizes and prediction horizons are considered. Results of different models are given in column-wise. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented. In small font bagging results for LS-SVM are presented.

between forecasts of 100 OP-ELMs and its MSE are calculated and depicted in the fourth columns. This is called ensemble method [35].

For each time series three sets of parameters (r, p) were investigated. For each set of parameters best MSE of each column is marked in a boldface. There are several findings one can notice in the results tables:

- Average MSE of DirRec strategy (second column) is better than the best MSE among all strategies for linear ordinary least squares model.

  This statement holds for all time series under investigation and all sets of parameters, except for one experiment: Sea-water temperature time series, second set of parameters. In this case linear model provides slightly better MSE: 2.848 vs. 2.860, see Table 1.

Santa Fe time series				
	Linear Model	**LS-SVM**	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 12, prediction horizon = 12			
Recursive	817.498	259.42 285.526±24.602	682.553 ± 138.229	310.729
Direct	**764.451**	**191.92** 212.660±11.364	**396.736 ± 12.111**	284.997
DirRec	764.561	263.54 301.216±18.933	468.217 ± 20.519	**259.616**
	Regressor size = 12, prediction horizon = 24			
Recursive	1207.5	452.76 465.961±27.125	1410.1 ± 491.463	596.997
Direct	**1114.6**	**277.6** 296.090±7.940	**595.015 ± 18.903**	429.574
DirRec	1115.0	402.79 448.428±18.928	706.750 ± 30.060	**403.014**
	Regressor size = 12, prediction horizon = 100			
Recursive	2049.6	1659.2 1491.386±85.0916	$1.2086e+10 \pm 1.2005e+11$	1.2811e+08
Direct	**1896.7**	**952.86** 964.091±9.565	**1494.0 ± 26.024**	**1262.8**
DirRec	1898.0	1347.9 1377.740±14.135	1816.5 ± 54.2133	1289.8

Table 3

Mean Square Errors(MSE) for Santa Fe dataset. Different regressor sizes and prediction horizons are considered. Results of different models are given in column-wise. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented. In small font bagging results for LS-SVM are presented.

Except for the Santa Fe time series, where Direct strategy significantly outperforms other strategies, standard deviation of DirRec strategy is less than standard deviations of other strategies. This indicates that in a single run OP-ELM with DirRec strategy tends to be the most accurate.

- For other strategies there are no such straightforward results as in the previous item.

  For instance, if we again perform comparison with the best linear model: OP-ELM with Recursive strategy can be better than the best linear model (Sea-water time series, parameters set 1) or worse (Sea-water time series, parameters set 3). The same is true for Direct strategy, it is superior to the best linear model (Sun spots time series, parameters set 1) or inferior

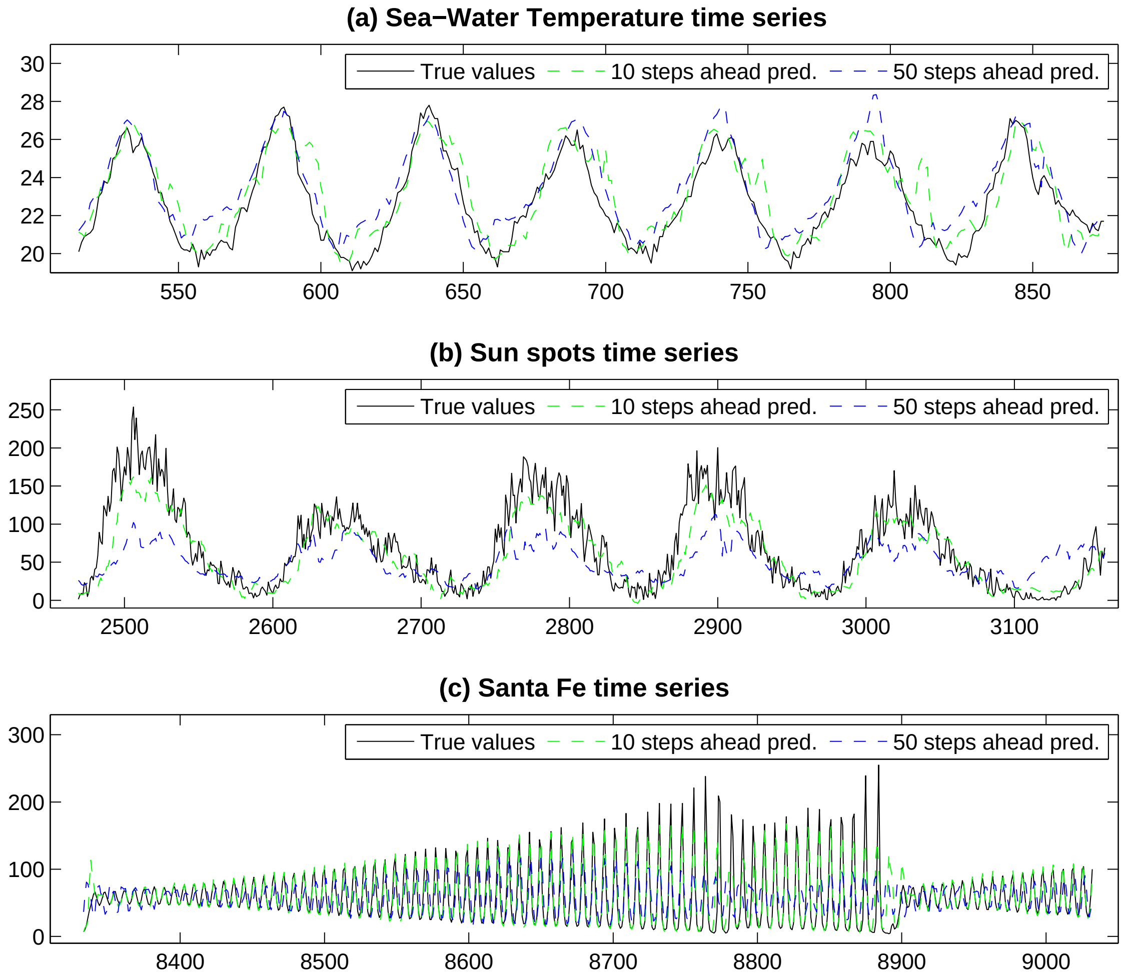

Figure 1. Visualization of predictions from ensemble of OP-ELMs. Ten steps ahead predictions as well as fifty steps ahead predictions are plotted for each time series. Regions for predictions are taken from the end of each time series and consist of aroud 700 points. (a) regressor size - 15, (b) regressor size - 28, (c) regressor size - 12

(Sea-water time series, parameters set 1).

Comparing only OP-ELM with three strategies, it is seen that there exist cases where each one of them is the best. Thus, DirRec is not generally the best strategy, but is it almost always better that the best linear model.

- LS-SVM outperforms OP-ELM only for Santa Fe time series.

This is a highly nonlinear time series which can be seen form Figure 1. For other two time series performance of LS-SVM is significantly worse even than linear model. Therefore, in contrast to OP-ELM, LS-SVM in general is not able to perform well without variable selection procedure and is a bad model for unfavorable time series. In any case, in the next subsection it is shown that computational time for LS-SVM is several times longer than OP-ELM.

- Using an ensemble method can improve the results dramatically.

  For example, for Santa Fe dataset (parameter set 1) MSE of ensemble of OP-ELMs is 259.6164 while for the best linear model it is 764.4508. So, the accuracy is improved by 66%. However, again, any of three strategies can be superior for the ensemble method. Standard bagging ensemble method for LS-SVM has not provided accuracy improvement.

On the Figure 1 predictions of all three time series are presented for various prediction horizons. For instance, each point on a curve for 10 steps ahead predictions, is calculated from regressors which are ten points behind the given point. For Sea-water and Sun Spots time series we see that even 50 steps ahead predictions repeat basic pattern of time series. For Santa Fe dataset 50 steps ahead predictions are quite far away from the original values, however 10 steps ahead predictions match reasonably well.

4.2 Running Times

This subsection is given to provide estimates of how fast our method is in comparison with linear model and LS-SVM. Linear ordinary least squares is one of the fastest and widely used in practice method for regression and/or time series prediction problems. Hence, it is given as a baseline method against which OP-ELM is compared. Characteristics and parameters of time series prediction which influence a running time are: length of time series, regressor size r and prediction horizon p. Length of time series and regressor size determine sizes of matrices which are intrinsically involved in computations. Prediction horizon is the number of future values to be predicted and, therefore, defines number of steps in prediction loop. Table 4 shows running times comparison for one experiment.

One of our most computationally heavy experiment is described in Table 4. It is Sea-water temperature time series with regressor size - 50 and prediction horizon - 50. Accuracy estimation for this experiment is summarized in Table 1, and the length of the training part of this time series equals 320 values.

Running times (seconds)			
	Linear	**OP-ELM**	**LS-SVM**
Recursive	0.03	0.57	4
Direct	0.14	21	172
DirRec	0.28	29	181

Table 4
Running times for Sea-water time series, regressor size - 50, prediction horizon - 50

From this table one can conclude that OP-ELM approximately 100-200 times slower than linear least squares model. Thus, if the standard trade-off between an accuracy and computational cost can afford such increase, nonlinear OP-ELM model can be exploited for time series prediction. LS-SVM is 5-7 times slower than OP-ELM.

5 Conclusions

In this paper, OP-ELM model is applied for long-term time series prediction problem. Three different strategies i.e. Recursive, Direct and DirRec are studied and compared. It is shown that OP-ELM, being a nonlinear model, needs roughly a hundred times more computing time than linear ordinary least squares model. Unlike LS-SVM, OP-ELM is shown to be robust against irrelevant or correlated variables. Hence it can be used without computationally heavy variable selection techniques and, unlike other nonlinear methods, there are no hyperparameters to adjust. This makes OP-ELM appealing to the problems where such increase in computations is affordable.

To analyze accuracy of predictions three time series were taken from completely different domains. For all our experiments except one OP-ELM with DirRec strategy outperforms linear model with the best of three strategies. In the exceptional experiment the difference is very small. Therefore, using OP-ELM with a DirRec strategy as a black box method may be considered preferable than using linear model. For highly nonlinear time series, OP-ELM may not perform very well. Considering only results for OP-ELM, experiments show that there are no superior strategy i.e. any strategy can be the best for a given time series.

Another way to improve accuracy of predictions is to run several OP-ELMs (possibly in parallel) and average their predictions (ensemble method). Which prediction strategy to use in this case is unclear - each one can be the best, however increase in accuracy can be very substantial.

Utilizing Recursive, Direct and DirRec strategies in one ensemble of OP-ELMs seems feasible direction for future work. This ensemble could obtain the global optimum in terms of MSE without the need of multiple trials for each prediction strategy. Different ensemble methods such as weighted ensemble of models and comparison with other methods for long-term time series prediction can be investigated in the future.

References

[1] A. Weigend and N. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, 1993.

[2] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[3] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* (revised ed. 1976) ed., 1970.

[4] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? empirical evidence from the {NN3} competition on time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 635 – 660, 2011.

[5] T. McElroy and M. Wildi, "Multi-step-ahead estimation of time series models," *International Journal of Forecasting*, vol. 29, no. 3, pp. 378 – 394, 2013.

[6] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert Syst. Appl.*, vol. 39, June 2012.

[7] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, pp. 2861–2869, October 2007.

[8] G. Bontempi and S. B. Taieb, "Conditionally dependent strategies for multiple-step-ahead prediction in local learning," *International Journal of Forecasting*, vol. 27, no. 3, pp. 689 – 699, 2011.

[9] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer Series in Statistics, 2001.

[10] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally-pruned extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 21, pp. 158–162, January 2010.

[11] Y. Yu, T.-M. Choi, and C.-L. Hui, "An intelligent fast sales forecasting model for fashion products," *Expert Syst. Appl.*, vol. 38, pp. 7373–7379, June 2011.

[12] M. van Heeswijk, Y. Miche, T. Lindh-Knuutila, P. Hilbers, T. Honkela, E. Oja, and A. Lendasse, "Adaptive ensemble models of extreme learning machines for

time series prediction," in *ICANN 2009, Part II* (C. Alippi, M. M. Polycarpou, C. G. Panayiotou, and G. Ellinas, eds.), vol. 5769 of *LNCS*, (Heidelberg), pp. 305–314, Springer, 2009.

[13] Z. L. Sun, T. M. Choi, K. F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decis. Support Syst.*, vol. 46, pp. 411–419, Dec. 2008.

[14] J. Ruksenaite and P. Vaitkus, "Prediction of composite indicators using combined method of extreme learning machine and locally weighted regression," *Nonlinear Analysis: Modelling and Control*, vol. 17, no. 2, p. 238251, 2012.

[15] A. Sorjamaa and A. Lendasse, "Time series prediction using DirRec strategy," in *ESANN06, European Symposium on Artificial Neural Networks* (M. Verleysen, ed.), (Bruges, Belgium), pp. 143–148, European Symposium on Artificial Neural Networks, April 26-28 2006.

[16] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[17] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications.* John Wiley & Sons Inc, January 1972.

[18] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *Neural Networks, IEEE Transactions on*, vol. 17, pp. 879–892, July 2006.

[19] T. Similä and J. Tikka, "Multiresponse sparse regression with application to multidimensional scaling," in *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005*, vol. 3697/2005, pp. 97–102, 2005.

[20] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," in *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[21] S. Chen, "Local regularization assisted orthogonal least squares regression," *Neurocomput.*, vol. 69, no. 4-6, pp. 559–585, 2006.

[22] D. Du, X. Li, M. Fei, and G. W. Irwin, "A novel locally regularized automatic construction method for rbf neural models," *Neurocomput.*, vol. 98, pp. 4–11, Dec. 2012.

[23] L. Breiman, "Heuristics of instability and stabilization in model selection," *The Annals of Statistics*, vol. 24, no. 6, pp. pp. 2350–2383, 1996.

[24] P. Radchenko and G. M. James, "Improved variable selection with forward-lasso adaptive shrinkage," 2011.

[25] D. M. Allen, "The relationship between variable selection and data augmentation and a method for prediction," *Technometrics*, vol. 16, pp. 125–127, 1974.

[26] R. Myers, *Classical and Modern Regression with Applications, 2nd edition.* Pacific Grove, CA, USA: Duxbury, 1990.

[27] F. Corona and A. Lendasse, "Variable scaling for time series prediction," in *Proceedings of ESTSP 2007, European Symposium on Time Series Prediction, Espoo (Finland)*, pp. 69–76, 2007.

[28] N. Gershenfeld and A. Weigend, "Monthly sunspot numbers." `http://solarscience.msfc.nasa.gov/greenwch.shtml`, 1749-2012.

[29] N. Gershenfeld and A. Weigend, "The santa fe time series competition data." `http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html`, 1994.

[30] A. Lendasse, ed., *ESTSP 2007: Proceedings*, Multiprint Oy / Otamedia, 2007. ISBN: 978-951-22-8601-0.

[31] A. Lendasse, D. Francois, V. Wertz, and M. Verleysen, "Vector quantization: A weighted version for time-series forecasting," *Future Generation Computer Systems*, vol. 21, no. 7, pp. 1056–1067, 2005.

[32] J. Suykens, *Least Squares Support Vector Machines.* World Scientific, 2002.

[33] K. Pelckmans, J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, L. Lukas, B. Hamers, B. D. Moor, and J. Vandewalle, "Ls-svmlab: a matlab/c toolbox for least squares support vector machines," 2002.

[34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[35] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse, "GPU-accelerated and parallelized ELM ensembles for large-scale regression," *Neurocomputing*, vol. 74, pp. 2430–2437, September 2011.