Aalto University
School of Science
Degree Programme in Machine Learning and Data Mining

Alexander Grigorievskiy

Missing Values Estimation: The Pyhäjärvi Case.

In application to long-term time series prediction

Master's Thesis
Espoo, March 1, 2013

Supervisor:	Professor Olli Simula, Aalto University
Instructors:	Amaury Lendasse Docent, Aalto University
	Anne-Mari Ventelä Docent, Pyhäjärvi Institute

Aalto University
School of Science
Degree Programme in Machine Learning and Data Mining

ABSTRACT OF
MASTER'S THESIS

Author:	Alexander Grigorievskiy		
Title:			
Missing Values Estimation: The Pyhäjärvi Case. In application to long-term time series prediction			
Date:	March 1, 2013	**Pages:**	79
Major:	Computer and Information Science	**Code:**	T-115
Supervisor:	Professor Olli Simula		
Instructors:	Amaury Lendasse Docent Anne-Mari Ventelä Docent		

Environmental modeling and prediction have been within the scope of human interests since ancient times. Contemporary agriculture and food production despite of all technological advances depend largely on favorable ecological conditions. However, climate change and consequences of human activity may deteriorate biological systems we used to utilize and enjoy. One example is lake Pyhäjärvi. It is a large lake on the south-west of Finland which plays an important role in local agriculture and fishing industry. The lake suffers from eutrophication. It is a process of abundant growth of lake plants and death of animals due to the lack of oxygen. The cause is redundant load of nutrients, especially phosphorus, into the lake from nearby agricultural fields. Due to support of local people and businesses, Pyhäjärvi Institute which develops measures to preserve lake's ecology has been established. This thesis is written in collaboration with researchers from Pyhäjärvi Institute and it is devoted to modeling of phosphorus concentration in the springs of Pyhäjärvi. Phosphorus modeling and prediction help to plan preservation measures and better understand ecology of the lake.

The thesis consists of two parts. In the first part, time series prediction problem is addressed. It is natural to model phosphorus concentration as a time series. However, the problem is studied generally and results can be applied to time series from any domain. It is shown that combination of Optimally-Pruned Extreme Learning Machine and DirRec prediction strategy outperforms widely used in practice linear model. Ensemble methods can further improve the accuracy, sometimes significantly.

In the second part, practical work with Pyhäjärvi dataset is conducted. It is impossible to directly apply methods of time series prediction, because the data contains many missing values. Therefore, in the beginning it is required to fill them. Several methods to estimate missing values of phosphorus are studied in this part. Regression approach, missing values approach and their combination are evaluated. The best model combinations as well as best variables are selected and imputation is done for three locations.

Keywords: Pyhäjärvi, long-term, time series, prediction, missing values, imputation, regression, LS-SVM, SVM, Direct, DirRec, Recursive,OP-ELM, ELM, EOF, SOM, SVT

Language: English

Acknowledgements

Studying during more than two years in Finland has been exiting and important experience for me. I became familiar with interesting scientific discipline, met a great teachers and passionate students from all over the world. Even more important is the fact that living and studying abroad helped me to rediscover myself. Now I look differently on many things and my personal traits have changed. I feel that I became more mature as a professional and as a human.

I would like to thank professor Olli Simula for opportunity to do the research. His friendly attitude and useful comments significantly helped in writing this thesis. My instructor Amaury Lendasse has played an important role in my studies. His interesting lectures, discussions and supervision have taught me a lot. This thesis, conference trip, and many other important thing would not be possible without his active participation. I'm very grateful to him for being my mentor.

I would like to thank Pyhäjärvi Institute for providing funding for the project. In particular, I am thankful to Anne-Mari Ventelä, Marjo Tarvainen and Teija Kirkkala for opportunity to work on practical environmental modeling. Special thanks to all members of EIML group. Help and useful discussions of Emil Eirola and Yoan Miche are very much appreciated. Francesco, Mark, Dusan, Qi, Anton, Tatiana, Tommi, Ajay, Luiza thank you all for valuable comments and positive atmosphere during this time.

I am deeply grateful to my family Vera, Valery and Olga for supporting and believing in me. My good friends Nickolay, Oleg, Leha, Kostik, David, Cho, Dima, Sergey and others provided interesting social life and encouragement. Thanks to all you guys.

Finally, I would like to thank my dear wife Sasha for waiting so long, and because she exists.

Espoo, March 1, 2013 Alexander Grigorievskiy

Abbreviations and Acronyms

TSP	Time Series Prediction
SLFN	Single Layer Feedforward Neural Network
ELM	Extreme Learning Machine
OP-ELM	Optimally-Pruned Extreme Learning Machine
LOO	Leave-One-Out
LARS	Least Angle Regression
MRSR	Multi-Response Sparse Regression
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SVR	Support Vector Regression
EOF	Empirical Orthogonal Functions
SVT	Singular Value Thresholding

Contents

Chapter 1

Introduction

Climate forecasting and environmental prediction have always captured thoughts of humanity. This is not surprising because quite often in the past survival of population depended on favorable climate and weather. In the present nothing has changed since the ancient times. Although new technologies have been developed and many processes have been automated, we still depend largely on nature and favorable climate. Moreover, new challenges started to appear. These are pollution, global warming and abundant use of natural resources. Thus, modeling and prediction of environmental behavior are very important for us as well as for future generations.

Nowadays climate is being intensively studied by scientific community and governmental organizations. For example, world's largest supercomputers are often engaged into environmental calculations. Types of environmental models vary a lot: starting from mass transfer models originated from geophysics and climatology to purely statistical models. Every model can be the best for some specific task. Statistical models have an advantage that they are universal and can be applied to almost any problem. Since the diversity of environmental systems is huge it would be too costly to develop a physical model in every case, so statistical modeling is often a natural choice. Moreover, accuracy and interpretability may be comparable.

This thesis is devoted to modeling of phosphorus concentration in the springs of lake Pyhäjärvi. It has been done in collaboration with researchers from Pyhäjärvi Institute. The institute has been organized by municipality and local businesses to preserve the ecology of the lake in the changing climate conditions. Pyhäjärvi is the largest lake on the south-west of Finland. It plays a central role in the local agricultural and fishing industries. Notably, the annual harvest of fish per hectare is the biggest among all lakes in Finland. However, in the beginning of 1990-s Pyhäjärvi faced a serious threat called eutrophication. It is a process when lake plants grow redundantly and animals die because of lack of oxygen. The cause of that is excessive nutrient load into the lake - mostly runoffs from the neighboring agricultural fields. The main nutrient is phosphorus. Therefore, modeling of phosphorus concentration is crucial for making intelligent decisions about lake conservation and for better understanding of the lake's ecology.

Forecasting of phosphorus concentration is natural to model as a *time series prediction* problem. The future values are predicted from patterns observed in the past. In this thesis, we are interested in a long-term time series prediction because overview of trends and changes is what is important in continuous lake management. Novel approach to long-term time series prediction is proposed and described in the Part 1 of the thesis. It exploits DirRec prediction strategy in combination with Optimally-Pruned Extreme Learning Machine which allows robust and fast time series forecasting. This part of the thesis may be viewed as an independent result and can be applied to other domains where time series prediction is required. One scientific paper [1] has been written and accepted on the basis of Part 1 results.

In ecological and environmental fields data samples are often measured manually. More and more automatic sensors are introduced but this process is not very fast. Since manual samples are costly and human factor is involved, data is often sparse and nonuniform. For example, phosphorus concentration in one ditch (spring) is measured once a month or even more rarely. Therefore, direct application of time series prediction techniques becomes impossible due to the lack of regular and uniform data. Missing values approach is needed to deal with such irregular data. The idea is to fill values for days (or weeks) when they were not measured, and then use obtained uniform data for time series prediction. Several methods and their combinations are studied in the Part 2 of this thesis. They are practically applied for imputation of missing values of phosphorus concentration for three locations. Data preprocessing, variable selection, non uniformity of the data - all these issues have been solved for Pyhäjärvi dataset. Results of the second part of the thesis are going to be published in the near future.

Thus, the scope of this thesis is twofold. At first, long-term time series prediction is studied in idealized settings. Then, practical work with real environmental dataset is elaborated. We do not claim that methods used in the second part are state-of-the-art, because the global comparison has not been performed. However, they allowed us to tackle efficiently the practical problem and satisfy time constraints of the master thesis.

1.1 Publications

[1] Alexander Grigorievskiy, Yoan Miche, Eric Severin, Anne-Mari Ventelä and Amaury Lendasse. Long-Term Time Series Prediction using OP-ELM. Accepted to *Cognitive Computation*.

Part I

Long-Term Time Series Prediction

Chapter 2

Introduction to long-term time series prediction

2.1 Time series prediction

Time series prediction 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 forecasting and many others.

Depending on the application, two main approaches to time series prediction usually appear: 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 long-term prediction estimations of multiple values are required. In this thesis, the problem of long-term time series prediction is addressed. By definition, it is a harder problem than one-step-ahead prediction because of the accumulation of errors and increasing uncertainties [2].

Earlier works for long-term time series prediction have shown that variable selection methods are needed to reduce the influence of irrelevant or correlated variables and to constrain the growth of the prediction error [2]. In part this situation is caused by the use of models like K-NN [3] which are sensitive to irrelevant variables. In this thesis, we propose to use OP-ELM model which is more robust to irrelevant or correlated variables due to internal pruning of inessential neurons [4]. OP-ELM model is described in more details in Section 3.2.

There exist three strategies for long-term time series prediction: Recursive strategy, Direct and DirRec. A detailed description of these strategies is given in Section 2.2. This thesis shows that using DirRec strategy with

OP-ELM model for long-term time series prediction in most cases produces better results than a linear model which is the most commonly used in contemporary time-series prediction [5]. For a particular time series any of the three strategies can provide the best results, however only DirRec strategy in vast majority of cases is better than linear model.

In addition, predictions made by ensemble of 100 [6] OP-ELMs are analyzed. Ensemble methods have been a topic of active scientific research in machine learning in general [7] and in ELM domain in particular [8] [9]. The thesis shows 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 [6], [10], [11], [12]. 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 3.2, concepts of ELM and OP-ELM are presented. Methodology Section 4.2 summarizes the building blocks of our approach. After that, experimental Section 4.3 follows.

In the long-term time series prediction the aim is to predict multiple values ahead. There exists a fundamental problem in this type of prediction. The state of underlying processes behind time series may change for times for which predictions are made. This means that behavior of a time series might become completely different from the observed patterns. Of course, this depends on a particular time series and on prediction horizon but is valid for all interesting time series, for which predictions are required [1]. Even for stationary time series, excluding several simple examples, for which statistical properties do not change over time, autocorrelation function decreases when lags grow [13]. This also implies that after some horizon, prediction becomes infeasible because future values are not related in any way with given values.

There exist another source of inaccuracy for long-term prediction. The strategy which is used for prediction utilize already predicted values to forecast further into the future. Three strategies for a time series predictions are described below in this section. Thus, inputs for a model are already an approximation and contain some errors. Errors, therefore, propagate through an algorithm and may amplify, which leads to inaccurate prediction. Additionally, size of the training data decreases for Direct and DirRec strategies.

Aforementioned problems are typical for long-term prediction while one-step-ahead prediction lacks those. However, the regular issues concerning model selection and parameter optimization are present for both.

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

2.2 Strategies for long-term time series prediction

2.2.1 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 hyper-parameteter of a model, and can be determined via cross-validation or other methods for selection of hyper-parameters. Section 4.2 explains how r is selected for datasets in this article. Thus, on the first step the model computes the following estimation:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \cdots y_{t-r+1}) \tag{2.1}$$

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

$$\begin{aligned} &\hat{y}_{t+2} = f(\hat{y}_{t+1}, y_t, \cdots y_{t-r+2}) \\ &\vdots \\ &\hat{y}_{t+r+1} = f(\hat{y}_{t+r}, \hat{y}_{t+r-1}, \cdots \hat{y}_{t+1}) \end{aligned} \tag{2.2}$$

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.

2.2.2 Direct strategy

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

$$\begin{aligned}
&\hat{y}_{t+1} = f_1(y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\hat{y}_{t+2} = f_2(y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\vdots \\
&\hat{y}_{t+p} = f_p(y_t, y_{t-1}, \dots y_{t-r+1})
\end{aligned} \tag{2.3}$$

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 generally more accurate than Recursive [2].

2.2.3 DirRec strategy

The DirRec strategy has been introduced in [14] 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:

$$\begin{aligned}
&\hat{y}_{t+1} = f_1(y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\hat{y}_{t+2} = f_2(\hat{y}_{t+1}, y_t, y_{t-1}, \dots y_{t-r+1}) \\
&\vdots \\
&\hat{y}_{t+p} = f_p(\hat{y}_{t+p-1}, \dots, \hat{y}_{t+1}, y_t, y_{t-1}, \dots y_{t-r+1})
\end{aligned} \tag{2.4}$$

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 p to be predicted. It has been shown by [14] that in general DirRec strategy with variable selection have superiority over two other strategies when the model f is nonlinear.

The goal of this thesis 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.

Chapter 3

Optimally-Pruned Extreme Learning Machine

3.1 Extreme Learning Machine (ELM)

The ELM algorithm was originally proposed by Guang-Bin Huang *et al* in [15] 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. It has been proven in [15] that ELM possesses an interpolation property, which means that having M distinct samples an ELM with no more that M hidden nodes can approximate these samples with arbitrary low error. Therefore, under conditions of the Theorem 1 in [15] the input weights and biases do not need to be adjusted and it is possible to calculate implicitly 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 $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{M}$ 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

$$\sum_{i=1}^{N} \boldsymbol{\beta}_i f(\mathbf{w}_i^T \mathbf{x}_j + b_i), \quad 1 \leq j \leq M, \tag{3.1}$$

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

In the case where the SLFN perfectly approximates the data, the errors between the estimated outputs $\hat{\mathbf{y}}_i$ and the actual outputs $\mathbf{y}_i$ are zero and the relation is

$$\sum_{i=1}^{N} \boldsymbol{\beta}_i f(\mathbf{w}_i^T \mathbf{x}_j + b_i) = \mathbf{y}_j, \quad 1 \leq j \leq M, \tag{3.2}$$

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

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1\mathbf{x}_1 + b_1) & \cdots & f(\mathbf{w}_N\mathbf{x}_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1\mathbf{x}_M + b_1) & \cdots & f(\mathbf{w}_N\mathbf{x}_M + b_N) \end{pmatrix}, \tag{3.3}$$

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

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 Moore-Penrose generalized inverse of the matrix $\mathbf{H}$, denoted as $\mathbf{H}^\dagger$ [16]. Overall, the ELM algorithm is summarized as:

Algorithm 1 ELM

```
Given a training set $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^M$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$, an activation function $f : \mathbb{R} \mapsto \mathbb{R}$ and the number of hidden nodes $N$.
- Randomly assign input weights $\mathbf{w}_i$ and biases $b_i$, $i \in \ 1 \leq j \leq N$;
- Calculate the hidden layer output matrix $\mathbf{H}$;
- Calculate output weights matrix $\mathbf{B} = \mathbf{H}^\dagger \mathbf{Y}$.
```

The proposed solution to the equation $\mathbf{HB} = \mathbf{Y}$ in the ELM algorithm, as $\mathbf{B} = \mathbf{H}^\dagger \mathbf{Y}$ has three main properties making it an appealing solution:

1. It is one of the least squares solutions of the mentioned equation, hence the minimum training error can be reached with this solution.

2. It is the solution with the smallest norm among the least squares solutions.

3. The smallest norm solution among the least squares solutions is unique and it is $\mathbf{B} = \mathbf{H}^\dagger \mathbf{Y}$.

Theoretical proofs and a more thorough presentation of the ELM algorithm are detailed in the original paper [15]. In Huang *et al.*'s later work it has been proved that the ELM is able to perform universal function approximation [17]. Universal approximation property means that any continuous function on a compact set X can be approximated in the sense of standard $\mathcal{L}^2(X)$ distance by ELM with arbitrary low error, provided enough hidden layer neurons are taken. Hence, from the theoretical point of view ELM is equal to other popular types of Single Layer Feedforward Neural Networks (SLFN).

However, the ELM tends to have problems when irrelevant or correlated variables [4]. For this reason, it is proposed in the OP-ELM methodology, to perform a some sort of variable selection, via pruning of the related neurons of the SLFN built by the ELM.

3.2 Optimally-Pruned ELM (OP-ELM)

The OP-ELM is made of three main steps summarized in the following algorithm:

Algorithm 2 OP-ELM

```
Given a training set $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{M}$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$.
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: Multiresponse Sparse Regression algorithm [18] enables to obtain a ranking of the neurons according to their usefulness, while the actual pruning is performed using the results of the Leave-One-Out validation.

In the original OP-ELM algorithm [4] 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.

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 irrelevant neurons in the hidden layer, the Multiresponse Sparse Regression (MRSR), proposed by Timo Similä and Jarkko Tikka in [18], is utilized. It is mainly an extension to the Least Angle Regression (LARS) algorithm introduced in [19]. LARS assumes that the output is one dimensional. MRSR extends it for the multi-output case and coincide with LARS if the output is one dimensional.

The main idea of these algorithms is that variables enter the regression model one by one. Hence, if this process is stopped at some iteration, the obtained solution is sparse. Spareness means that only a subset of regressors participate in the model and contribution of the rest regressors is neglected. This agrees with the rule of parsimony which states that the simpler model is preferred if it describers data equally well. The criteria by which the next entering variable is selected is that is has maximum absolute correlation (or minimum angle) with the current residual. More specific details are given in the original papers.

If MRSR is not stopped, ranking of variables is obtained. Algorithm has been developed for linear systems and ELM belongs to that class. There is a linear dependency between hidden layer and outputs, so results are fully applicable. Neurons serve as variables and,hence, MRSR provides an exact ranking of neurons.

3.2.2 Leave-One-Out (LOO)

Since 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, number of times training the model is required equals the number of samples. Fortunately, for linear systems there exists a closed formula which provides an exact analytic LOO error without retraining the model for each sample. This is called PRESS (PREdiction Sum of Squares) statistic, see [20], [21] for details of this formula and its implementations.

For ELM training we have $\mathbf{H}\boldsymbol{\beta} = \mathbf{y}$ assuming that the output is one dimensional (notations are changed form $\mathbf{B}$ and $\mathbf{Y}$ to $\boldsymbol{\beta}$ and $\mathbf{y}$ because output is one dimensional and now $\boldsymbol{\beta}$ and $\mathbf{y}$ are vectors). The size of the matrix $\mathbf{H}$ is $(M \times N)$, where M - number of samples and N - number of neurons in the hidden layer. There are several methods to solve $\boldsymbol{\beta}$ in the least squares sense [22]. One of them is to multiply the both sides of the system by $\mathbf{H}^T$ and then solve $\mathbf{H^TH}\boldsymbol{\beta} = \mathbf{H^Ty}$. Matrix multiplication costs $O(MN^2)$, while solving the latter system takes $O(N^3)$ operations. Therefore, the naive implementation of Leave-One-Out cross-validation would require increase in complexity by the factor M: $O(MN^3) + O(M^2N^2)$.

The PRESS formula, which exactly calculates LOO error is:

$$\varepsilon^{\mathrm{PRESS}} = \|\mathbf{D}\,(\,\mathbf{y} - (\mathbf{H^TH})^{-1}\mathbf{Hy}\,)\|_2^2, \tag{3.4}$$

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

Calculation of $\varepsilon^{\mathrm{PRESS}}$ takes $O(N^3) + O(MN^2)$ operations to compute $(\mathbf{H^TH})^{-1}\mathbf{H^T}$ and the final results can be computed by $O(MN)$ operations in addition. Therefore, computational complexity of PRESS statistic is $O(N^3) + O(MN^2)$ and it is M (number of samples) times smaller than the naive implementation of LOO error. Because of the small computational complexity this formula is utilized in OP-ELM to select optimal number of neurons.

After ranking the neurons by MRSR the least important are pruned and LOO error is computed on the rest neurons. In order to further speed up computations, pruning is done in batches of five neurons. 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 4.3.2.

Chapter 4

OP-ELM for long-term time series prediction

Three model training strategies described in Section 2.2 are applied for a long-term time series prediction. Usage of OP-ELM model is analyzed for all three strategies and superiority of DirRec strategy is shown with some small remarks. OP-ELM is also compared with linear ordinary least squares model which is the most commonly used model for a long-term time series prediction [5]. The complete algorithm is summarized further in the Section 4.2.

4.1 Motivation

Earlier, it has been shown [14] that DirRec strategy with variable selection and K-NN model is beneficial in terms of accuracy. As a variable selection method forward-backward algorithm was used [2]. Variable selection methods can be very time consuming especially if we consider wrapper class of methods [2]. Thus, the motivation for our approach is the desire to avoid computationally expensive variable selection. It can be achieved by utilizing OP-ELM which intrinsically prunes irrelevant neurons, and therefore is not so sensitive to non-optimal input variables.

Accuracy of OP-ELM is compared to the ordinary least squares model which serves as a baseline. It is one of the simplest models for time series prediction, however it is fast, easily interpretable and, therefore, frequently used in practice [5].

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. [4]. Moreover, for other nonlinear models fine tuning of hyper-parameters is necessary, for example,

(C, ϵ) in Support Vector Regression (SVR). 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 ϵ) as many SVRs as there are points in the grid, 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 parameter selection procedure dramatically increases computational time and many well known nonlinear machine learning models may become impractical for long-term time series prediction.

4.2 Algorithm

It is worth mentioning that our approach is developed only for a stationary or semi-stationary time series. Therefore, the preliminary step which is required is stationarizing time series. Stationarity means that statistical properties of a time series do not change over time [13]. There are methods to detect non-stationarity e.g. [23], or it can be known or assumed *a priori*. Methods exist to deal directly with non-stationary time series [24], or time series can be transformed into a stationary one. Good overview of these transformation methods and stationarity is given in [25] and references there in. For example, linear trend can be removed from a time series, or instead of an original time series first difference ($y_t - y_{t-1}, \forall t \in [2, \cdots, n]$) can be studied. After stationarizing has been done, or under assumption that initial time series is (semi-)stationary, further steps can be undertaken.

The second step is converting time series prediction problem into a regression problem. To do it, one needs to choose a regressor size r which plays role of (initial) dimensionality of input data. Regressor size is the number of previous values of time series which are used to predict the future value. It can be chosen using preliminary knowledge or some computationally easy algorithms one of which is described in experimental part of this paper. Early mentioned prediction strategies: Recursive, Direct and DirRec immediately influence the way how regression problem is constructed. Exact matrices for Recursive strategy and time series $\{y_1, y_2, \cdots, y_n\}$ under assumptions that regressor size equals r, are presented below:

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_r & 1 \\ y_2 & y_3 & \cdots & y_{r+1} & 1 \\ \vdots & & & & \\ y_{n-r} & y_{n-r+1} & \cdots & y_{n-1} & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} y_{r+1} \\ y_{r+2} \\ \vdots \\ y_n \end{bmatrix} \tag{4.1}$$

The column of ones at the end of the first matrix corresponds to the constant term in the linear model: $\hat{y}_{n+1} = [y_n, y_{n-1}, \cdots, y_{n-r+1}]^T \times \boldsymbol{\beta} + c$.

As described in section 2.2 for the Recursive strategy, the model needs to be trained only once, and after that the same model is used to predict time series for $1, 2, 3, \cdots, p$ values ahead. In contrast, for Direct and DirRec strategies, first a model is trained to predict one value ahead. The regression problem in this case is exactly the same as for Recursive strategy. Having predicted the first value, training a different model is needed to predict the second value. Regressors and right part in (4.1) changes in accordance with the rules in Section 2.2. This process repeats until all the values up to prediction horizon p are predicted.

Algorithm 3 Complete method for long-term time series prediction

```
Stationarize (detrend) time series
Select an appropriate regressor size $r$, and required prediction horizon $p$
for $s = 1$ to $p$ do
    Using selected strategy and chosen regressor size convert time series
    prediction problem into regression problem (For Recursive strategy
    model training is needed only once)
    Train regression model
    Predict the $s$-th value
end for
```

As an approach to regression problem two models are used and compared. The first one is a linear ordinary least squares which is one of the simplest regression model, however it is fast, easily interpretable and, therefore, frequently used in practice. The second one is OP-ELM, which, as was mentioned before, is nonlinear, computationally efficient and robust against irrelevant input variables.

In addition, modern hardware often allows parallel computations almost without any increase in running times. Hence, several OP-ELM can be trained in parallel. These ensemble methods are known to improve accuracy [9] [8] [26]. In this thesis, averaging of predictions of 100 OP-ELM is conducted and investigated.

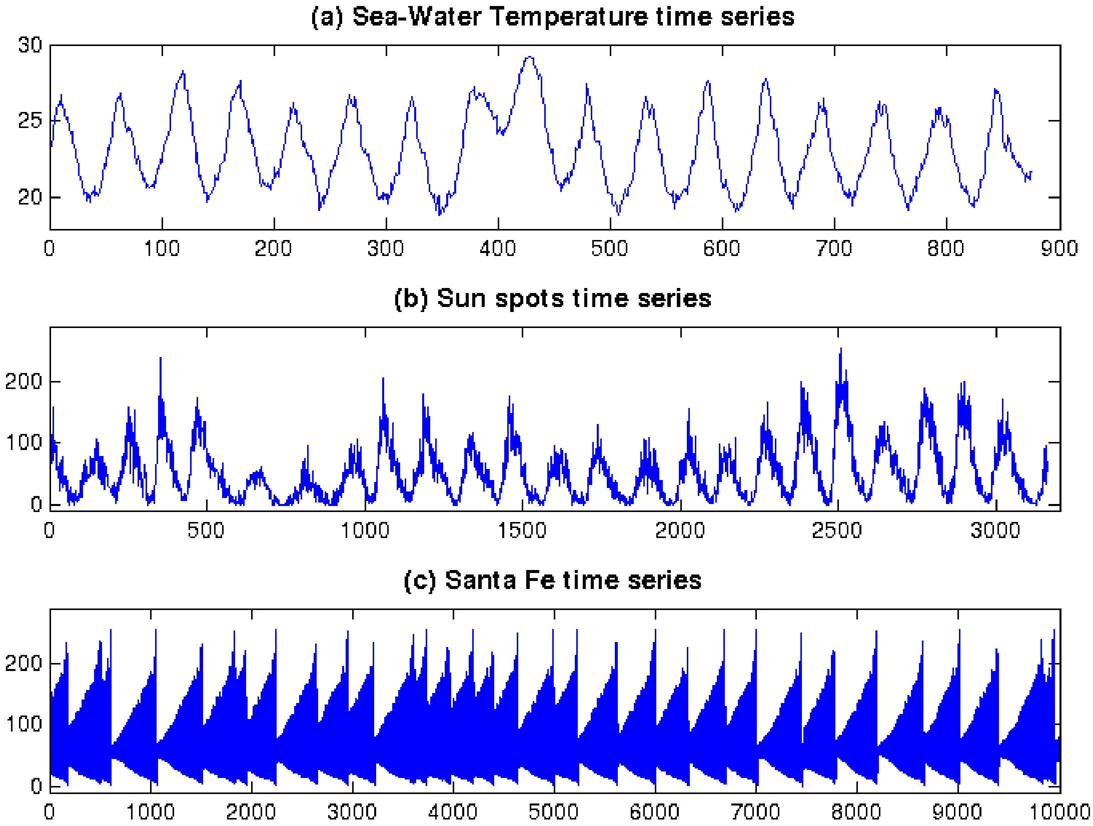


Figure 4.1: Visualization of investigated time series

4.3 Experimental results

The method is applied to three different time series: Sea-water temperature [27], Sun Spots [28] and Santa Fe A [29]. First one contains 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.

Sea-water temperature time series			
	Linear Model	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 15, prediction horizon = 15		
Recursive	**2.4397**	**2.3306 ± 0.3458**	**2.1557**
Direct	2.8868	2.6638 ± 0.1229	2.4604
DirRec	2.8729	2.4096 ± 0.1184	2.3239
	Regressor size = 50, prediction horizon = 50		
Recursive	**2.8476**	3.0721 ± 1.1807	**2.3644**
Direct	3.3081	3.5260 ± 0.1997	3.0299
DirRec	3.2637	**2.8599 ± 0.1332**	2.6984
	Regressor size = 15, prediction horizon = 50		
Recursive	3.9383	3.7315 ± 0.6280	3.2017
Direct	**3.6858**	3.4801 ± 0.1015	3.1361
DirRec	3.7018	**3.2409 ± 0.0943**	**3.0692**

Table 4.1: **Mean Square Errors(MSE) for Sea-water temperature dataset. Different regressor sizes and prediction horizons are considered. In the first column results for linear model are given; second column - mean and standard deviation of MSEs of 100 independent OP-ELMs; third column - MSE of averaged predictions of OP-ELM ensemble. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented.**

Usually, in time series prediction the number of regressors to use is unknown and it has to be estimated, see Section 4.2. Here, *a priori* information is used to select appropriate regressor sizes. For the Sea-water temperature dataset regressors of sizes 15 and 50 are 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 the 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.

Sun spots time series			
	Linear Model	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 28, prediction horizon = 12		
Recursive	496.8105	491.0473 ± 11.2947	471.8737
Direct	**493.3891**	487.1273 ± 5.9080	**456.3721**
DirRec	493.4853	**482.1657 ± 4.4942**	467.9926
	Regressor size = 28, prediction horizon = 24		
Recursive	785.6096	748.2101 ± 30.3272	716.5362
Direct	**772.9818**	739.3338 ± 8.8929	**692.1222**
DirRec	773.7776	**734.1157 ± −8.2450**	713.7017
	Regressor size = 28, prediction horizon = 28		
Recursive	891.3626	832.1842 ± 39.0713	791.2807
Direct	**874.8780**	825.9258 ± 11.6144	**773.8031**
DirRec	876.1438	**824.1604 ± 10.6708**	801.8169

Table 4.2: **Mean Square Errors(MSE) for Sun spots dataset. Different regressor sizes and prediction horizons are considered. In the first column results for linear model are given; second column - mean and standard deviation of MSEs of 100 independent OP-ELMs; third column - MSE of averaged predictions of OP-ELM ensemble. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented.**

4.3.1 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 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.

Santa Fe time series			
	Linear Model	**Mean and std of 100 independent OP-ELMs (ensemble)**	**Ensemble of OP-ELMs (Average)**
	Regressor size = 12, prediction horizon = 12		
Recursive	817.4984	682.5525 ± 138.2294	310.7285
Direct	**764.4508**	$\mathbf{396.7355 \pm 12.1111}$	284.9972
DirRec	764.5608	468.2166 ± 20.5188	**259.6164**
	Regressor size = 12, prediction horizon = 24		
Recursive	1207.5	1410.1 ± 491.4630	596.9967
Direct	**1114.6**	$\mathbf{595.0148 \pm 18.9030}$	429.5736
DirRec	1115.0	706.7502 ± 30.0598	**403.0144**
	Regressor size = 12, prediction horizon = 100		
Recursive	2049.6	$1.2086e + 10 \pm 1.2005e + 11$	1.2811e+08
Direct	**1896.7**	$\mathbf{1494.0 \pm 26.0241}$	**1262.8**
DirRec	1898.0	1816.5 ± 54.2133	1289.8

Table 4.3: **Mean Square Errors(MSE) for Santa Fe dataset. Different regressor sizes and prediction horizons are considered. In the first column results for linear model are given; second column - mean and standard deviation of MSEs of 100 independent OP-ELMs; third column - MSE of averaged predictions of OP-ELM ensemble. In bold font best MSEs for each column (and each regressor size and prediction horizon) are presented.**

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 numbers 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.

Results of experiments are given in Tables 4.1,4.2,4.3. Because of 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 [6] 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 second columns of the tables. In addition, arithmetic mean between forecasts of 100 OP-ELMs and its MSE are calculated and depicted in the third columns. This is called ensemble method [8]. Errors of the linear models are given in the first column.

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.

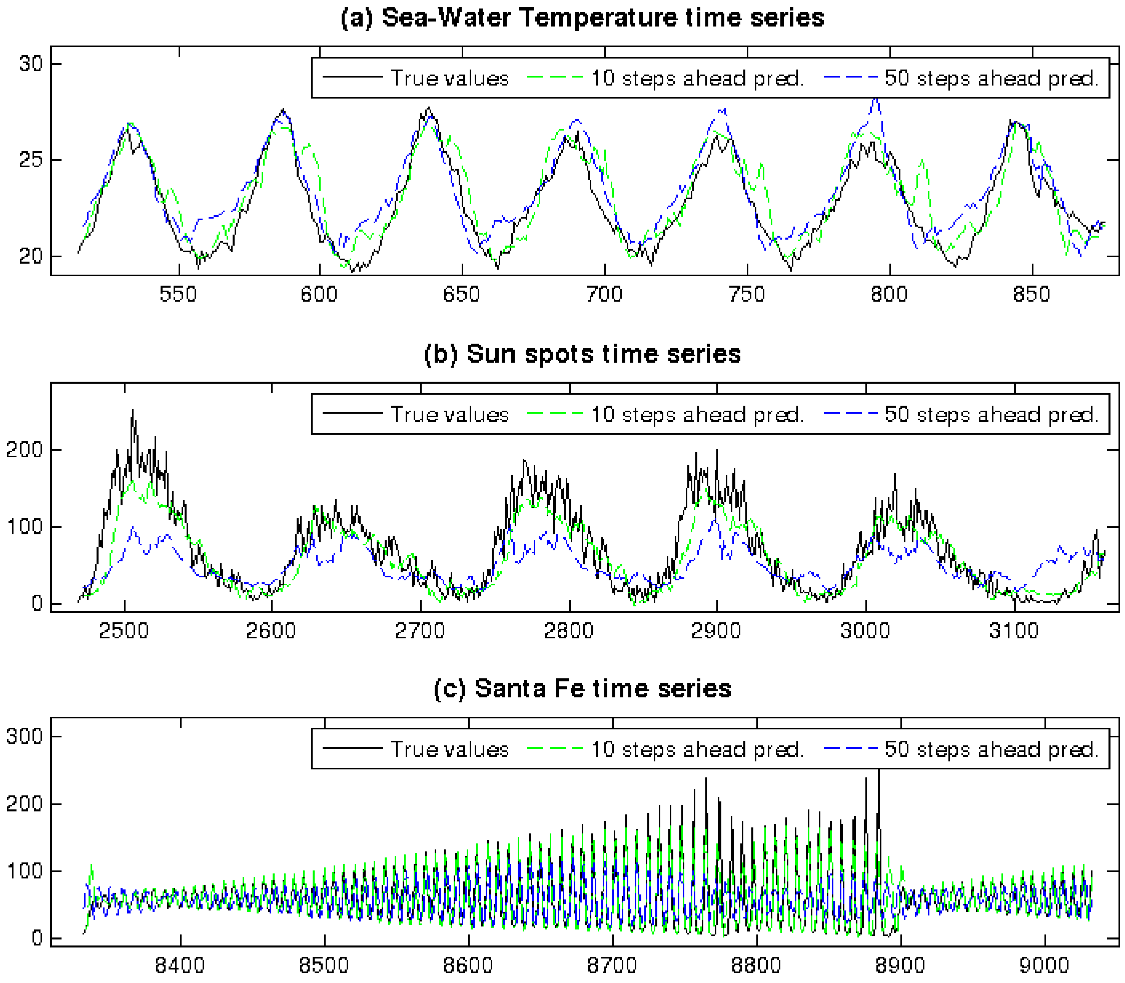


Figure 4.2: Visualization of predictions from ensemble of OP-ELMs. Ten steps ahead predictions as well as fifty steps ahead predictions are plotted for each time series. Regions for predictions are taken from the end of each time series and consist of aroud 700 points. (a) regressor size - 15, (b) regressor size - 28, (c) regressor size - 12

There are several findings one can notice in the result 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: 2.8476 vs. 2.8599, see Table 4.1.

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

  Comparing only OP-ELM and 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.

- 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.

Running times (seconds)		
	OP-ELM	**Linear**
Recursive	1	0.04
Direct	33	0.25
DirRec	47	0.47

Table 4.4: Running times for Sea-water time series, regressor size - 50, prediction horizon - 50

On the Figure 4.2 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.

4.3.2 Running Times

This subsection is given to provide estimates of how fast our method is in comparison with linear model. Linear ordinary least squares is one of the fastest and widely used in practice 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 4.4 shows running times comparison for one experiment.

One of our most computationally heavy experiment is described in Table 4.4. It is Sea-water temperature time series with regressor size - 50 and prediction horizon - 50. Accuracy estimation for this experiment is summarized in Table 4.1, and the length of the training part of this time series equals 320 values.

From this table one can conclude that approximately a factor of 100 is added to the computational cost of 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.

4.4 Conclusions

In this thesis, 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 a hundred times more computing time than linear ordinary least squares model. OP-ELM is known 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 hyper-parameters 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. Considering only results of 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 will be investigated in the future.

Part II

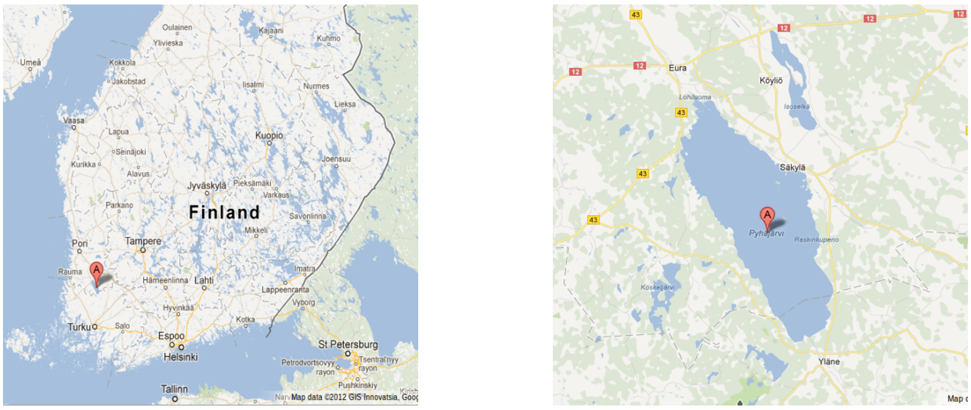
Missing Values Estimation of Pyhäjärvi data

Chapter 5

Datasets construction

5.1 Pyhäjärvi

Pyhäjärvi lake (one of the several lakes with the same name in Finland) is located on the south-west of Finland, in the municipality Säkylä. It is the largest lake in the south-west Finland and it is a center of agricultural activity. It is also famous due to large yields of fish. While average annual fish yield in Finland is 10 kilograms per hectare, Pyhäjärvi lake provides 60. So, there are fishing companies as well as independent fishermen operating in the area. Lake's geographic location is shown on the Figure 5.1 and key characteristics are summarized in the Table 5.1.



(a) Location of the lake

(b) Shape of the lake

Figure 5.1: Google maps images of location and shape of Pyhäjärvi

However, there is a threat which may deteriorate well-being of Pyhäjärvi. The lake suffers from the process called eutrophication. Eutrophication

Parameter	Value
Area	$155\,km^2$
Perimeter	$111\,km$
Max. depth	$26\,m$
Mean depth	$5.5\,m$
Catchment area	$431\,km^2$

Table 5.1: Pyhäjärvi characteristics

means an excess of nutrients coming mostly from the land. Nutrients cause a dense growth of plant life and death of animal life from the lack of oxygen. Due to the high interest of local people to preserve the lake's quality, several measures have been implemented to overcome eutrophication. Municipality and local industry established Pyhäjärvi Institute which monitor, analyze and develop lake's environment. Under the institutes's supervision new water protection practices such as buffer-zones, sedimentation ponds and wet-lands have been introduced. New filtering ditches and sand filters have been installed to prevent excessive nutrients to penetrate into the lake [32]. These efforts have not been spent in vain and for now the eutrophication process is stopped.

The main nutrient which impacts the lake is phosphorus. It is used intensively as a fertilizer by local agricultural industry which operates in the lake's catchment area. The important aim is to reduce amount of phosphorus in the lake. This is done by introducing filtering systems mentioned in the previous paragraph, and also by fishing of commercially unprofitable fish. This fish catch is sponsored by lake conservation project and it helps to remove twenty five percents of annual phosphorus load. Additional challenges arise because of the changing climate. During extremely warm winters of 2007, 2008 ice cover period was very short and it increased phosphorus penetration into the lake.

Modeling of phosphorus concentration is becoming very important. It helps to make intelligent decisions about lake preservation activities and better understand the lake's ecosystem. In this thesis phosphorus concentration in several ditches (springs) is modeled. Physical measurements of phosphorus concentration has been done but they are too sparse to estimate the real dynamics. Therefore, filling missing values is essential for further analysis of underlying processes. In addition, relation with other variables like temperature, precipitation and previous values of phosphorus needs to be investigated. The name of phosphorus variable in the dataset is "Total P" and it's estimation is the topic of the second part of this thesis.

5.2 Data preprocessing

Initial data has been received in several Excel files. The first file contains several variables which are: "Suspended solids (fine)", "Suspended solids (rough)", "Total P", "Total N", "Ammonium N", "Dissolved P", "Nitrite-nitrate N". These variables are given for different dates and different locations. Locations are denoted as S1, S2 ... S13 - thirteen locations in total, and each of them represents one catchment of the lake. In this thesis, analysis of three locations S10, S11 and S12 is conducted and methodology for processing the rest locations is built. The exact meaning of variables is not relevant for understanding the material, except the main variable for which analysis is hold - "Total P" is an abbreviation of total phosphorus. There are other three files, one of them contains daily values of flow for each catchment in $[m^3/sec]$. Another one contains daily average air temperature in the region of the lake. The last file contains precipitation level in $[mm/(5\ days)]$, so the values are given for every 5 days. Precipitation variable is called "Rains" from now on for better illustration and conciseness. However, in general, precipitation includes rains as well as snow and hail.

By consulting with domain specialists from Pyhäjärvi Institute, variable "Suspended solids (rough)" is discarded from the very beginning because it does not provide any additional information. The usefulness of other variables is unclear at this stage. The time period for which at least some values are present is from 01.01.1980 to 10.11.2011, so 11637 days in total. However, important variables like "Total P","Total N",*etc.* are given only for 228 days. Exact number of present values is given in Table 5.2.

The main goal of the work is to estimate total phosphorus concentration ("Total P") on dates when it is unknown. Since almost all other variables contain missing values, using classical statistical methods becomes impossible here. Bayesian regression methods are another potential way to tackle this type of problems but in this thesis they are not investigated due to the time and space constraints of the thesis work. Thus, missing values approach is given a priority here. But which variables to include in the matrix for missing values imputation? On the one hand, including "useful" variables on which phosphorus concentration depends is necessary. On the other hand including variables which are sparse, increases sparsity of resulting matrix. Sparsity refers to the amount of missing values in a matrix. If sparsity is high, and since number of free parameters in the model is fixed, less certain estimations are obtained. Therefore trade-off between including more variables and

		Number of present values		
No.	Variable name	Location S10	Location S11	Location S12
1	"Flow"	11637 (All)	11637 (All)	11637 (All)
1	"Rains"	11637 (All)	11637 (All)	11637 (All)
2	"Ammonium N"	201	205	202
3	"Dissolved P"	187	191	188
4	"Nitride-Nitrate N"	183	185	182
5	"Suspended solids (fine)"	224	228	226
6	"Total P"	225	228	226
7	"Total N"	224	228	226

Table 5.2: Numbers of present values for variables at different locations

reducing number of sparse variables appears.

The decision to construct missing values matrix for each location (S1,S2,...S13) separately follows from aforementioned trade-off. Otherwise, the joint matrix would be too sparse. However, even considering each location separately, the sparsity of the matrix for S11 location in Table 5.2 is 73%. It means that 73% of values are missing. After several trials with this matrix it is concluded that sensible estimations are impossible to make with such level of sparsity.

As it is seen from Table 5.2, besides fully complete variables, the variable "Total P" is the most populated. The disturbing fact, which is revealed by more detailed analysis of dates when missing values appear, is that for each location other variables like "Ammonium N", "Dissolved P"*etc.* are given on exactly the same dates as "Total P". Hence, it is not possible to use these variables in a regression setting where "Total P" is used as a variable to predict, while other variables serve as regressors. Using these variables in the missing values approach seems dubious as well, because there are no cases when some variable, say "Ammonium N", is given and "Total P" is absent.

5.2.1 Correlation analysis of different locations

From the same analysis of dates of missing values, it is established that "Total P" in one location is often given on different dates than "Total P" in some other location. To be precise, not all the dates are different, for substantial part dates are still the same, but also non-intersecting dates present significantly. Therefore, if dependency between phosphorus concentrations in two

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
S1	1.00	0.16	0.24	0.14	0.16	0.11	0.38	0.04	0.33	0.09	0.14	0.09	0.08
S2	0.16	1.00	0.45	0.37	0.26	0.14	0.11	0.13	0.23	0.26	0.25	0.21	0.15
S3	0.24	0.45	1.00	0.59	0.64	0.56	0.59	0.56	0.80	0.68	0.51	0.34	0.27
S4	0.14	0.37	0.59	1.00	0.53	0.46	0.18	0.31	0.38	0.48	0.46	0.25	0.23
S5	0.16	0.26	0.64	0.53	1.00	0.37	0.23	0.32	0.37	0.41	0.38	0.22	0.20
S6	0.11	0.14	0.56	0.46	0.37	1.00	0.35	0.73	0.33	0.70	0.47	0.37	0.29
S7	0.38	0.11	0.59	0.18	0.23	0.35	1.00	0.33	0.76	0.41	0.27	0.21	0.12
S8	0.04	0.13	0.56	0.31	0.32	0.73	0.33	1.00	0.27	0.69	0.29	0.36	0.21
S9	0.33	0.23	0.80	0.38	0.37	0.33	0.76	0.27	1.00	0.47	0.49	0.23	0.13
S10	0.09	0.26	0.68	0.48	0.41	0.70	0.41	0.69	0.47	1.00	**0.67**	0.53	0.36
S11	0.14	0.25	0.51	0.46	0.38	0.47	0.27	0.29	0.49	0.67	1.00	0.54	0.36
S12	0.09	0.21	0.34	0.25	0.22	0.37	0.21	0.36	0.23	0.53	**0.54**	1.00	0.73
S13	0.08	0.15	0.27	0.23	0.20	0.29	0.12	0.21	0.13	0.36	0.36	0.73	1.00

Table 5.3: Pairwise covariance matrix of the variable "Total P" for different locations

locations is estimated, one can predict it at some location using the given value from another location. Moreover, after discussion with researchers from Pyhäjärvi Institute it becomes obvious that two relatively close locations must have very correlated phosphorus concentrations. Hence, to estimate values of phosphorus in one location matrix corresponding to this location is inserted phosphorus from other locations. To determine locations with similar behavior of phosphorus concentration, correlations analysis is performed. Pairwise covariance matrix [33] is presented in Table 5.3.

In the Table 5.3 the most correlated locations with S11 regarding the variable "Total P" are marked in boldface. They are S10 and S12. It is known that in a multivariate problem Pearson's correlation coefficient, which is present in the covariance matrix in Table 5.3, is not always giving correct measure of dependency between two variables because they can be related through other variables. To measure correlation between two variables excluding influence of other variables *partial correlation coefficient* in used. It's definition is the following:

If there are p variables $X^1, X^2, \cdots, X^p$, which are normalized to have zero mean and unit variance, and two sets of coefficients $\mathbf{w_1^\star}, \mathbf{w_2^\star}$ are determined by:

$$\begin{aligned} \mathbf{w_1^\star} &= \underset{w_{13}, w_{14}, \cdots, w_{1p}}{\arg\min} \left\{ \sum_i (X_i^1 - w_{13}X_i^3 - w_{14}X_i^4 - \cdots - w_{1p}X_i^p)^2 \right\} \\ \mathbf{w_2^\star} &= \underset{w_{23}, w_{24}, \cdots, w_{2p}}{\arg\min} \left\{ \sum_i (X_i^2 - w_{23}X_i^3 - w_{24}X_i^4 - \cdots - w_{2p}X_i^p)^2 \right\} \end{aligned} \tag{5.1}$$

then residuals are defined as:

$$\begin{aligned} r_i^1 &= X_i^1 - w_{13}^{\star} X_i^3 - w_{14}^{\star} X_i^4 - \cdots - w_{1p}^{\star} X_i^p \\ r_i^2 &= X_i^2 - w_{23}^{\star} X_i^3 - w_{24}^{\star} X_i^4 - \cdots - w_{2p}^{\star} X_i^p \end{aligned} \tag{5.2}$$

and the *partial correlation coefficient* between variables X_1 and X_2 is Pearson's correlation coefficient between their residuals:

$$\rho_{12,(12)} = \frac{\sum (r_i^1 - E[r^1])(r_i^2 - E[r^2])}{\sigma[r^1]\sigma[r^1]} \tag{5.3}$$

In the notations above $E[r^1]$ denotes mean value of the residual vector r^1 and σr^1 - standard deviation of this vector. So, at first linear regression is performed on each X^1 and X^2 while all other variables $X^3 \cdots X^p$ are being regressors. Then differences between true values of X^1 or X^2 and corresponding regression model are calculated, these are called residuals. Finally, regular correlation coefficient is calculated between two residual vectors. Thus, dependency between X^1 and X^2 is analyzed while influence of other variables is removed. Of course this is true only in linear sense. To perform correlation analysis which is based on partial correlation coefficient, all pairwise coefficients need to be computed. Solving those regression problems for each pair seems cumbersome, but luckily there exist a neat formula for computing partial correlations [34]. If p_{ij} are elements of the inverse of correlation matrix (or covariance matrix), then partial correlation coefficient between $i-$th and $j-$th variables can be computed by:

$$\rho_{12,(12)} = -\frac{p_{ij}}{\sqrt{p_{ii}}\sqrt{p_{jj}}} \tag{5.4}$$

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
S1	1.00	-0.13	0.07	-0.04	-0.08	-0.04	-0.23	0.01	-0.06	0.10	-0.03	-0.00	-0.04
S2	-0.13	1.00	-0.47	-0.12	0.21	0.10	-0.01	0.19	0.31	0.06	-0.09	-0.13	0.12
S3	0.07	-0.47	1.00	-0.17	-0.54	-0.04	0.16	-0.39	-0.76	-0.26	0.26	0.12	-0.21
S4	-0.04	-0.12	-0.17	1.00	-0.17	-0.20	0.16	0.12	0.00	-0.05	-0.10	0.05	-0.03
S5	-0.08	0.21	-0.54	-0.17	1.00	0.01	-0.06	0.14	0.35	0.13	-0.20	-0.01	0.06
S6	-0.04	0.10	-0.04	-0.20	0.01	1.00	-0.16	-0.44	0.13	-0.15	-0.19	0.13	-0.12
S7	-0.23	-0.01	0.16	0.16	-0.06	-0.16	1.00	-0.07	-0.59	-0.17	**0.29**	-0.07	-0.02
S8	0.01	0.19	-0.39	0.12	0.14	-0.44	-0.07	1.00	0.29	-0.29	0.21	-0.21	0.21
S9	-0.06	0.31	-0.76	0.00	0.35	0.13	-0.59	0.29	1.00	0.19	**-0.41**	-0.03	0.17
S10	0.10	0.06	-0.26	-0.05	0.13	-0.15	-0.17	-0.29	0.19	1.00	**-0.47**	-0.14	0.05
S11	-0.03	-0.09	0.26	-0.10	-0.20	-0.19	0.29	0.21	-0.41	-0.47	1.00	-0.27	0.04
S12	-0.00	-0.13	0.12	0.05	-0.01	0.13	-0.07	-0.21	-0.03	-0.14	**-0.27**	1.00	-0.68
S13	-0.04	0.12	-0.21	-0.03	0.06	-0.12	-0.02	0.21	0.17	0.05	0.04	-0.68	1.00

Table 5.4: Matrix of partial correlation coefficients of "Total P" between different locations

Matrix of partial correlation coefficients between "Total P" in different locations is given in Table 5.4. Again, in the boldface locations with the highest absolute correlation with location S11 are emphasized. As we see, there are two other locations: S9 and S7, which have relatively high partial correlation coefficient with S11. However, partial correlations are computed via inverse of covariance matrix, which is taken as pairwise covariance matrix. Pairwise covariance matrix is only an estimate of true covariance matrix under the presence of missing values. Therefore, partial correlations are estimates as well, and it is unwise to completely rely on those. Taking all this into account, decision is made to keep locations S10, S11 and S13 as a group and in missing values matrix for each of them include the other two. The complete list of variables in missing values matrix for location S11 is presented further in the Table 5.6.

5.2.2 Correlation analysis of integrated flow

There is a hypothesis that phosphorus concentration depends not on the flow at the same day, but on the accumulated flow during several previous days (or weeks). It is easy to imagine, that the more total amount of water arrives to some point the more it brings phosphorus, or *vice versa* the more water flows out the less phosphorus is left. Another conjecture is that "Total P" may depend on accumulated flow but with a time delay. For example, total amount of water during five days may influence phosphorus concentration, but these five days were three days ago. In other words, these five days finished three days ago, so there is a time delay between water inflow and its' influence to phosphorus concentration. This delay seems natural since phosphorus penetrates into springs and lake from agricultural fields because of the rains, and phosphorus needs time to reach a spring through soil layers.

Aforementioned effects are analyzed in the following way. Flow variable is integrated backwards with integration horizons varying from 1 to 99 days. Integration backwards means just summing up previous values up to prediction horizon. For instance, for horizon 1: $int\_flow(i) = flow(i) + flow(i-1)$ is performed for all valid values i and so forth, where i is the current day. In addition, time delays from 1 to 99 are being analyzed. Time delay k applied to flow integrated over 2 days is written mathematically as $int\_flow(i) = flow(i-k) + flow(i-1-k)$. For each integration horizon 99 time delays are applied, so in total 99×99 new variables are constructed, Person's correlation and partial correlation coefficients between "Total P" and every new variable are computed. Correlation coefficients are stored in two 99×99 matrices where rows correspond to integration horizons and columns to time delays. Matrices are not presented here because of their size.

This analysis has been done for all three locations S10, S11 and S12, and matrices with correlation coefficients are obtained. Then maximum absolute correlation is sought in these matrices. It has been found that maximum of both Pearson's correlation and partial correlation for all locations is situated somewhere in the middle of the matrix. This means that "Total P" is correlated best with flow which is integrated and time shifted. However, it was noticed that maximum modulus correlation in the first column (without time shift) is quite close to the global maximum for all locations and all correlation coefficients. Hence, to simplify interpretation and analysis, time delays are not studied further. In the Table 5.5 maximum (among integration horizons) correlation coefficients between phosphorus and integrated flow are written.

	Location		
	S10	S11	S12
Pearson's correlation coefficient	+0.42 :1	-0.42 :55	-0.20 :44
Partial correlation coefficient	0.34 :46	0.19 :44	0.24 :0

Table 5.5: Maximal correlation coefficients between "Total P" and integrated flow

In each table cell, maximal correlation coefficient and, behind the colon, corresponding integration horizons are provided. For location S11 the maximum modulus Pearson's correlation coefficient is -0.42 and it is obtained when flow is integrated over 55 days. The maximum partial correlation coefficient is 0.19 with flow integrated over 44 days. It is decided to include 55 day integrated flow into the dataset corresponding to S11. The reason is that partial correlation coefficient is not much less for 55 days integration and that -0.42 is significantly larger number by absolute value. The complete list of variables in missing values matrix for location S11 is presented further in the Table 5.6.

For other two locations situation is different. There are peaks in modulus of correlation coefficient near 44 days integration, however there are other peaks at 1 day integration. This is explicitly seen from the table of S10 location and Pearson's correlation. For location S12 there is also local peak for 1 day integration, but because it is local it is not shown in the table. Partial correlations have also local peaks when flow is integrated over one day. Taking this into account, it is decided to construct two datasets for each location S10 and S12. The only difference is that the first dataset includes flow integrated over previous 55 days, while other includes flow integrated over 1 previous day. The other variables are the same and analogous to those of S11 location.

5.2.3 Smoothing of flow

By looking at the plot of flow in the location S11 which is shown on the Figure 5.2a, one can notice that flow has spiky and very non-smooth shape. It may cause troubles in regression as well in imputation setups. The reason is that if phosphorus depends nonlinearly on the flow, changes in phosphorus concentration may be much less than changes in flow intensity. This is quite typical behavior in real physical systems. Although we use methods of nonlinear analysis, they do not always find the correct dependency, therefore it might be useful to facilitate these methods by smoothing flow intensity.

Smoothing is done by substituting each value of flow by an average of five values $f_{new}(i) = \frac{1}{5}[f(i-2)+f(i-1)+f(i)+f(i+1)+f(i+2)]$. This simple sequential averaging suppresses high frequency components in a sequence of values and result is more smooth. The shape however remains mostly the same. This procedure has been done with flow intensity in each location and this new variable is included into the corresponding dataset 5.6.

5.2.4 Averaging over five day intervals

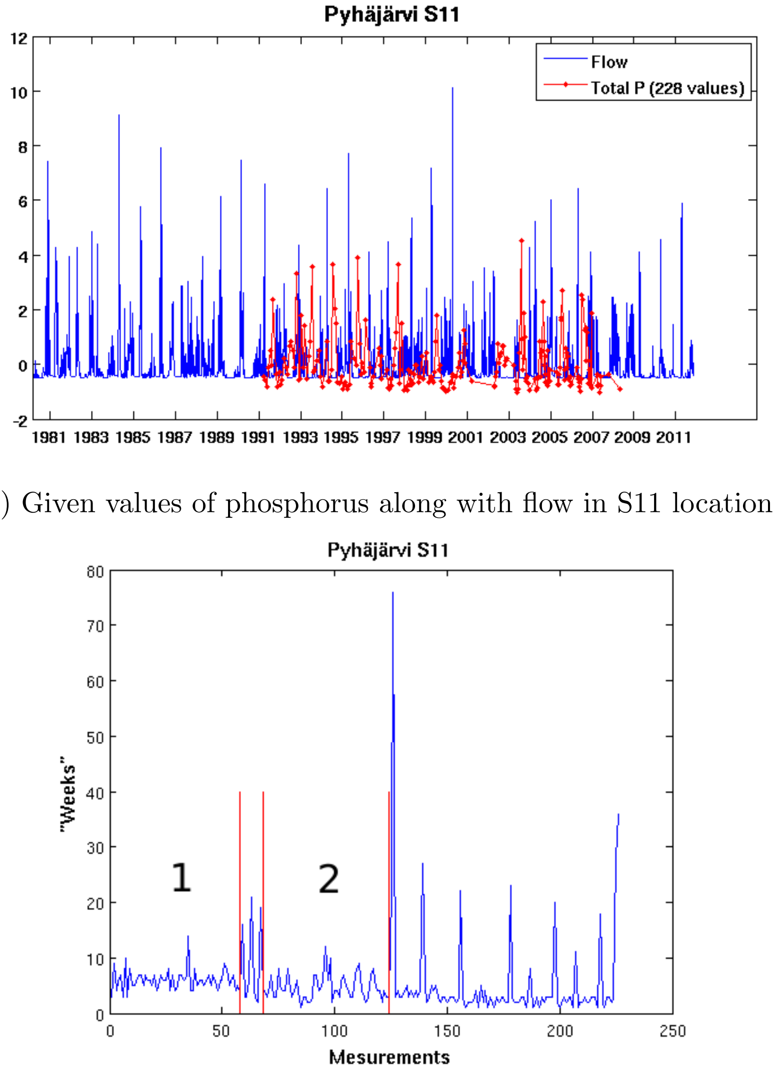
For each location S10, S11, S12 "Total P" variable from other two locations is added to missing values matrix as described in Subsection 5.2.1. Distances between present values of phosphorus (in each location) are quite large. For instance, for location S11 average distance in days between given values of "Total P" is 27 days. Under this high sparsity of a variable "Total P" it is hard to hope that imputed values are satisfactory accurate. Test experiments with such sparse data, cross-validation and MSE as a measure of accuracy, confirmed that suspicion.

In addition, variable "Rains", which provides data about precipitation in the lake region, is given only each five days. Consultation with Pyhäjärvi researchers confirmed that this variable can be useful for phosphorus estimation. The reason is that phosphorus penetrates into the lake from agricultural fields and rain water is a transport. Therefore, decision is made to average all the data over five day intervals. Averaging is done in agreement with Pyhäjärvi researchers, so for them this time resolution is satisfactory. The aim is twofold, first reduce sparsity of the data, and second ability to include "Rains" variable into the dataset. Averaging with missing values is done in a way that only present values contribute to the result. If all five values are missing then the result is missing as well. This five day interval is called "Week" from now on for conciseness and brevity.

Because data is provided for several years, catching correlations between different seasons, months and even weeks seems reasonable here. Therefore each five day interval which is called ”Week” is given a number within a year i.e. first interval has number 1 and last interval’s number is 72. It would be possible to include this variable into the dataset. However, there is a problem, that within one year distances between two dates are correct, but considering two consecutive years this is not true anymore. For example, the last ”Week” of year 1990 has number 72 while the first ”Week” of year 1991 has number 1. These are two consecutive weeks but the distance between them equals 72. This is not desirable property because two actually close ”Weeks” are far away from each other with respect to this new variable.

To overcome this difficulty two new variables are added instead, one is sine of ”Week” and the second one is cosine. This is a standard way of modeling periodic variables [35]. In that case the maximum Euclidean distance between two weeks equals 2 and is obtained when weeks are on the opposite sides of a year. Actually close weeks of two different years appears to be close in this case. Thus two variables ”Sine week” and ”Cosine week” are included into the dataset and their usefulness is investigated below. The complete list of variables in missing values matrix for location S11 is presented further in the Table 5.6

5.3 Exploratory data analysis



(a) Given values of phosphorus along with flow in S11 location.

(b) Consecutive differences in days between given values of phosphorus.

Figure 5.2: Sparsity analysis of Total Phosphorus in S11 location.

The sparsity of phosphorus concentration of averaged dataset is analyzed in this section. On the Figure 5.2a ”Total P” is shown against flow intensity. Variable ”Flow” is a full variable, so it is given for every week. As we can see from the figure there are no phosphorus measurements before year 1991 and after 2008.

It is also noticeable that sometimes there are large gaps between measurements. Analysis of gaps is shown in more details on Figure 5.2b. At this plot first difference of phosphorus concentration is shown *i.e* differences in dates of every two consecutive present values. In the middle there is a huge peak about 70 weeks of missing values which correspond to a year when no phosphorus measurements were taken. There are other relatively large gaps in values - more than 20 weeks, which correspond to absence of measurements during a season.

Imputation of missing values in this thesis is performed only within time period when "Total P" values are given. So, analysis is restricted to years 1991-2008. Missing values approach is applied only to time periods when phosphorus concentration does not contain large gaps. If there are large gaps accuracy of missing values approach is dubious. There are two such periods and they are denoted by numbers 1 and 2 on the Figure 5.2b. For other "Weeks" regression approach is selected for estimation of phosphorus concentration.

5.4 Regression and missing values datasets

5.4.1 Missing values dataset

The resulting datasets for missing values imputation for location S11 are presented in the Table 5.6. There are three datasets as mentioned in the previous Section 5.3. The first one corresponds to complete time interval and is not solved as a missing values problem. It servers as an input to the regression problem after removing incomplete columns. The other two are subsets of the first one and are solved via missing values methods. They correspond to the two parts displayed on the Figure 5.2b.

All variables in these datasets have been described before, except the last six variables. They correspond to the shifted versions of variables 2, 3 and 4. The idea is to make missing values imputation problem more similar to the time series prediction problem. In time series prediction problem predictions are made on the basis of previously known values. More details are given in Part **??** of this thesis. So, for example, variables 11 and 12 are shifted versions of "Total P S11" (2-nd variable) by one and two weeks respectively. By doing so, in one row phosphorus from some "Week" becomes aligned with phosphorus one week and two weeks before. Since, some missing values imputation methods process information row wise, dependency between value of phosphorus and values from previous two weeks is modeled. The similarly shifted "Total P" for locations S10 and S12 are added as variables 13, 14 and

		Dataset		
No.	Variable name	Complete dataset (1230 rows)	Part 1 (351 rows)	Part 2 (271 rows)
1	"Flow S11"	1230 (full)	351 (full)	271 (full)
2	"Total P S11"	227	59	58
3	"Total P S10"	225	60	58
4	"Total P S12"	226	58	72
5	"Temperature"	1228	351 (full)	271 (full)
6	"Integrated Flow S11"	1230 (full)	351 (full)	271 (full)
7	"Smoothed Flow S11"	1230 (full)	351 (full)	271 (full)
8	"Rains"	1230 (full)	351 (full)	271 (full)
9	"Sin Week"	1230 (full)	351 (full)	271 (full)
10	"Cos Week"	1230 (full)	351 (full)	271 (full)
11	"Time shift 1 Ph. S11"	226	58	57
12	"Time shift 2 Ph. S11"	226	58	57
13	"Time shift 1 Ph. S10"	225	59	57
14	"Time shift 2 Ph. S10"	225	59	57
15	"Time shift 1 Ph. S12"	225	57	71
16	"Time shift 2 Ph. S12"	225	57	71

Table 5.6: Datasets for S11 location and amounts of present values in columns

15,16 respectively.

However, there is a negative side of introduction shifted variables. Sparsity of dataset increases because of that. For example, sparsity of the part 1 dataset without addition of shifted variables is 25% while after their introduction it is almost 47%. The same is for part 2 dataset, before shifted variables introduction sparsity is 23% and after 43%. It is one of the questions investigated in this thesis whether it worth adding these shifted variable of not. From the experiments in the next chapter the answer is positive that the addition is beneficial.

Similar datasets as the one shown is the Table 5.6 are constructed for locations S10 and S12. They contain either the common variables like "Temperature", "Rains" or symmetrically substituted variables for a specific location *e.g.* "Flow S10". For each location S10 and S12 the same division on part 1 and part 2 has been done because time lags between present values of phosphorus are approximately the same for all locations. Thus, in total there are six datasets for missing values imputations: part 1 and part 2 for all three locations.

5.4.2 Regression dataset

Regression dataset is constructed from the complete dataset in the Table 5.6. In order to maximize number of training samples only complete variables are taken as regressors (including "Temperature"). The variable to predict is "Total P". The number of training samples for S11 location is 227 and equals the number of present values of "Total P S11". Having trained the regression model, it is possible to estimate phosphorus concentration on all other weeks when it is missing. This is called regression approach and it is compared to missing values approach described in the previous subsection. Since missing values approach is studied only during periods "Part 1" and "Part 2", for other dates regression approach is used to estimate "Total P"

The same process is done for other locations to construct corresponding regression datasets. Variables for regression dataset, this time, for location S10 are presented in Table 5.7

No.	Variable name
To predict	**"Total P S10"**
1	"Flow S10"
2	"Temperature"
3	"Integrated Flow S10"
4	"Smoothed Flow S10"
5	"Rains"
6	"Sin Week"
7	"Cos Week"
8	"Rains Int. 1"
9	"Rains Int. 2"
⋮	⋮
17	"Rains Int. 10"

Table 5.7: Regression dataset for S10 location. Number of training samples is 224

As was said earlier, variable to predict is "Total P S10", regressors No. 1-7 are complete variables from the missing values dataset. For S10 dataset there are 224 training samples. Ten more variables No. 8-17 added to each regression dataset and their usefulness is investigated in Chapter **??**. They are integrated values of "Rains" over 1 week (current one plus one week before), 2 weeks and so forth up to 10 weeks. Integration is done in exactly the same manner as integration of flow in Subsection 5.2.2. The motivation for including these variables is to check possibility that phosphorus concentration depends on accumulated precipitation intensity during a long period.

Analysis and comparison of regression and missing values approaches are presented in the next chapter.

Chapter 6

Estimation of phosphorus concentration

6.1 Regression

Regression problem arises when one wants to model dependency between two variables : $\boldsymbol{x}$ - input variable and $\boldsymbol{y}$ - output variable, based on the observed pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$. The output variable takes continuous values within bounded of unbounded region $\mathcal{Y}$. In this thesis output variable "Total P" is one dimensional, therefore subsequently y is written in a normal, not a bold font. It is assumed that there exists a joint, unknown distribution $P(\boldsymbol{x}, y)$, and the goal is to minimize the risk functional $R = \int L[y, \hat{y}(\boldsymbol{x})]P(\boldsymbol{x}, y)\, d\boldsymbol{x}dy$, where $L[y, \hat{y}(\boldsymbol{x})]$ is, so called, loss function which measures inaccuracy between true value y and the one obtained from the model $\hat{y}(\boldsymbol{x})$. The most common loss function is the quadratic loss $L = (y - \hat{y}(\boldsymbol{x}))^2$. Under the quadratic loss it is possible to show that the optimal model is $\hat{y}(\boldsymbol{x}) = E[y|\boldsymbol{x}]$, which is called regression function, and expectation is taken over $P(y|\boldsymbol{x})$. Since the distribution $P(\boldsymbol{x}, y)$ is unknown it is not possible to find regression function directly. Alternative formulation of regression problem is $y = f(\boldsymbol{x}) + \epsilon(\boldsymbol{x})$, where f - is a deterministic function of input variables while ϵ is random noise which may also depend of input variable $\boldsymbol{x}$.

Regression models can be linear and nonlinear. Linearity leads to fast training algorithms and validation procedures. In linear models, outputs depend linearly on the model parameters. It is worth to emphasize that it is not necessary that outputs depend linearly on inputs, only on model parameters. Hence, linear models may also be nonlinear in inputs, in particular, nonlinear features can be extracted and serve as new inputs. The limitation of linear models is that number of parameters grows exponentially with the number of input dimensions [35]. This is a form of curse of dimensionality. Many nonlinear models are capable to avoid that growth of parameters number. In this thesis one linear model - ridge regression, and two nonlinear SVM and LS-SVM are compared for phosphorus concentration prediction.

Another issue that can deteriorate regression models is the presence of irrelevant, redundant, or too noisy input variables. Those can increase computational time, contribute to the curse of dimensionality and finally, reduce accuracy of the regression. In addition, selecting of only useful variables facilitates interpretability of the model. In this thesis variable selection is applied to the regression datasets in order to determine useful variables and use them in missing values approach and final estimation of phosphorus concentration.

6.1.1 Variable selection

There exist many methods for variable selection. Overview of some of them is presented in [36]. These methods can be divided into three main categories: filters, wrappers and embedded methods. Filter methods optimize some external criteria for each subset of input variables and select a subset which corresponds to the optimum. Criteria is usually computed between output variable and subset of inputs and can be, for instance, mutual information, delta test or gamma test. The advantage of the filter methods is that they are usually faster to compute than other types of methods. However, disadvantage is that they are completely unrelated with the training process, and therefore do not take into account properties of data model.

Wrapper methods utilize learning machine as a black box method to score different subsets of input variables. This is usually done through taking apart validation set and measuring performance of the machine using only a subset of inputs. Therefore, multiple retraining of learning algorithm is required which is a main disadvantage of this class of methods. If the number of input variables is high (≥ 10 depending on the learning algorithm and number of samples) wrapper methods become infeasible. In this case, some search heuristics can be used [2].

Embedded methods for variable selection are built-in into learning machine and selection is done simultaneously with learning. Their properties depend on the particular data model and learning algorithm and, hence they can be very diverse. They are not studied in this thesis; however, they can be potentially useful for these kind of tasks.

Hierarchical variable selection is applied in this thesis. There are three stages of variable selection. Initially, when there are many variables more computationally effective, but less accurate filter methods are used. When substantial part of variables is discarded, finer wrapper methods are applied.

In fact, correlation analysis which is described in Chapter 5 is a first stage of variable selection. It has helped to construct datasets and choose best correlated locations and integration horizons of flow intensity.

Regression datasets, which are similar to the one shown in the Table 5.7, contain 17 variables. Thus, as mentioned above, direct application of more precise but computationally intensive wrapper methods becomes impossible. Hence, at first Delta test method which belongs to the filter category is utilized. Based on its outcome variables are divided onto three groups. The first group is redundant of unimportant variables and is discarded from subsequent investigation. The second group is important and informative variables which are kept in datasets. To the third group attributed variables which are harder to include into two previous groups. All possible their combinations (*i.e.* greedy search) with important variables are analyzed further using wrapper methods and regression modeling.

There are five datasets for which variable selection needs to be done. One of them is presented in Table 5.7 and it corresponds to S11 location. Each of locations S10 and S12 has two datasets associated with it. Hence, datasets names are: "reg_S11", "reg_S10_1", "reg_S10_2", "reg_S12_1", "reg_S12_2". Two datasets corresponding to the same location differs only in one variable "Integrated Flow" as described in Subsection 5.2.2. In the dataset "reg_S10_1" flow is integrated over 55 previous days, while in "reg_S10_2" it is integrated over 1 previous day. The same difference is present between "reg_S12_1" and "reg_S12_2" datasets.

6.1.1.1 Delta test

Delta test has been successfully applied for variable selection, see for example [37],[38]. Originally this method has been developed for noise variance estimation. If regression is considered as

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \quad , i = 1 \cdots N \tag{6.1}$$

where $f(\boldsymbol{x}_i)$ is a smooth function of input $\boldsymbol{x}_i$, and ϵ_i is zero mean, i.i.d. noise. Then estimation of noise variance is given by:

$$Var(\epsilon) \approx \sum_{i=1}^{i=N} \left(y_i - y_{NN(i)}\right) \tag{6.2}$$

Here notation $NN(i)$ means index of the nearest neighbor of $\boldsymbol{x}_i$: $\boldsymbol{x}_j = \arg\min_{j \neq i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, and $NN(i) = j$.

It turns out [39], [38] that Delta test can also be applied for variable selection. The procedure is the following, all subsets of input variables are taken one by one, delta test for each one is calculated and subset with the smallest estimation of noise variance is selected.

Delta test has been applied for every regression dataset, however algorithm has been slightly modified in accordance with [38], to make it more stable and robust. As before delta test is calculated for every subset of variables, but then histogram of delta tests is plotted. Based on the histogram, two types of analysis are held: positive analysis and negative analysis. Let's consider positive analysis first. Instead of selecting subset of variables which minimizes Delta test, some threshold value is selected below which delta test is assumed to be small. All variable subsets which delta test is below the threshold are considered and each variable is scored by number of subsets it presents in. The higher the score the more useful the variable is, because it is found in many subsets with low noise variance estimation.

Negative analysis is almost the same except threshold is selected above which the Delta test is assumed to be too high. Variables are scored in the same way, except this time the higher the score, the worse the variable is. The more "bad" subsets a variable is present, the less relevant it is considered to be.

Delta test analysis has been done for all five regression datasets and results are present in Table 6.1. Since number of variables in each dataset is 17, $(2^{17} - 1)$ Delta tests need to be computed for one dataset. It is done on a Core 2 Quad 2.83Ghz × 4 computer in the Matlab environment. Entire computation for one dataset takes approximately 30 minutes.

As one can notice from the table, only variable "Temperature" is selected to be relevant for all datasets. Importance of other variables does not have significant evidence at this stage. One of the periodic "Week" variables appears among relevant variables almost for all datasets and various "Flow" variables enter relevant group quite often. All variables which are not in this table are discarded and are not used any more for regression modeling.

Variables in the right most column are investigated further through a wrapper approach. All possible combinations of these variables are evaluated via Monte-Carlo 15-fold cross-validation. Important variables are always present in all subsets. There are 50 iterations of Monte-Carlo validation, on each iteration dataset is randomly permuted and 15-fold cross-validation is applied. Number of folds is increased in comparison to standard 10 because number of samples in each dataset is small, and there is a need to increase number of samples for training. In the end, mean square errors (MSE) are averaged over 50 iterations and over 15 folds within one iteration. Best subsets of variables are those with the minimal MSE.

Dataset /(No. of samples)	Relevant variables	Variables to be investigated further
reg_S11 (227)	”Flow S11” , ”Temperature”, ”Integrated Flow S11”	”Smoothed Flow S11”, ”Rains”, ”Sin Week”, ”Cos Week”, ”Rain int 1”
reg_S10_1 (224)	”Temperature”, ”Integrated Flow S10”, ”Smoothed Flow S10”, ”Sin Week”	”Flow S10”, ”Rains”, ”Cos Week”, ”Rain int 1”, ”Rain int 4”, ”Rain int 9”
reg_S10_2 (224)	”Temperature”, ”Sin Week”	”Flow S10”, ”Integrated Flow S10”, ”Smoothed Flow S10”, ”Rains”, ”Cos Week”, ”Rain int 1”
reg_S12_1 (225)	”Temperature”, ”Integrated Flow S12”, ”Cos Week”	”Flow S12”, ”Smoothed Flow S12”, ”Rains”, ”Sin Week”, ”Rain int 2”, ”Rain int 9”
reg_S12_2 (225)	”Flow S12”, ”Temperature”, ”Cos Week”	”Integrated Flow S12”, ”Smoothed Flow S12”, ”Rains”, ”Sin Week”, ”Rain int 1”, ”Rain int 2”

Table 6.1: Variable selection via Delta test for regression datasets

Besides selection of best variables, different regression models are evaluated in the same cycle of validation. Again the one that has lower MSE is selected as the best and used for final estimation of ”Total P”. For each model all variable subsets from the previous paragraph are tried and compared. There are three regression models which are analyzed: Ridge regression, Support Vector Regression (SVR) and Least-Squares Support Vector Regression (LS-SVR). In fact, SVR is split into two models: one with hyperparameters optimization and another one without. Details of these models and results are written in the following subsections.

6.1.2 Ridge regression

Ridge regression is a linear technique for regression where output y_{new} (assumed to be one dimensional) is expressed as:

$$y_{new} = \boldsymbol{x}_{new}^T (X^T X + \lambda I)^{-1} X^T \boldsymbol{y} \tag{6.3}$$

given new input $\boldsymbol{x}_{new}$. Here X and $\boldsymbol{y}$ contains training data of inputs and outputs respectively. Ridge regression provides biased estimates of the coefficients of linear regression model, but the variance is often lower than for least-squares solution [40]. The term λI is a Tikhonov regularization term and makes solution more stable especially if matrix $X^T X$ is close to singular.

There is one parameter to adjust - λ. In this work it is adjusted via a second internal cycle of cross-validation.

6.1.3 Support vector machines

Support Vector Machines has been invented already a long time ago - in 1960s by Vladimir Vapnik. However, useful extensions such as soft-margin SVM, nonlinear SVM and Support Vector Regression (SVR) appeared only in 1990s in the works of Vapnik and his colleagues [41], [42], [43]. Here we are interested in regression problem, but it shares similar properties as SVMs for classification. The solution of binary classification problem depends only on the subset of training points which are close to the separating hyperplane between two classes. Analogously, solution of regression problem depends only on the training points which lie outside ϵ-tube of the model prediction.

In general mathematical formulation this model can be expressed as the following: suppose we want to find solution of regression problem in the form

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \tag{6.4}$$

where ϕ is a mapping from original space to higher (possibly infinite) dimensional space, which is called *reproducing Hilbert Space*. This mapping does not need to be known or estimated in advance because it is not used explicitly. Then, training regression model can be expressed in constrained optimization form, where objective function to be minimized, includes $\|\boldsymbol{w}\|^2$ because we are interested in finding simpler model, and the sum of slack variables which measure violation of ϵ-tube by each training point.

$$\begin{aligned} &\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{i=N}(\xi_i^+ + \xi_i^-) \\ \text{s. t. } &y_i - \boldsymbol{w}^T\boldsymbol{x}_i - b \le \epsilon + \xi_i^+ \\ &\boldsymbol{w}^T\boldsymbol{x}_i + b - y_i \le \epsilon + \xi_i^- \\ &\xi_i^+ \ge 0,\ \xi_i^- \ge 0, \quad 1 \le i \le N \end{aligned} \tag{6.5}$$

This problem can be represented in dual form which is historically used more frequently for SVM training:

$$\begin{aligned} \max -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \epsilon\sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) + \\ +\sum_{i=1}^{N} y_k(\alpha_i^+ - \alpha_i^-) \\ \text{s. t. } \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\ 0 \ge \alpha_i^+, \alpha_i^- \le C \quad 1 \le i \le N \end{aligned} \tag{6.6}$$

Here α_i^+ and α_i^- are Lagrange multipliers corresponding to the first and second constraints in Equation 6.12, $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is called *kernel* and equals $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$ scalar product in reproducing Hilbert space. The regression function is expressed through parameters α in the form:

$$y(\boldsymbol{x}) = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)K(\boldsymbol{x}, \boldsymbol{x}_j) \tag{6.7}$$

The most common algorithms used for SVR training are *Sequential Minimal Optimization (SMO)* algorithm and interior point methods [44]. One of the main chooses of kernel function is Gaussian kernel:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\} \tag{6.8}$$

This form of kernel function is used for experiments in this thesis. A well-known library LIBSVM [45] with additionally written Matlab wrapper has been used.

6.1.4 Hyper-parameters selection for SVR

There are three hyper-parameters to adjust in support vector regression formulation: C - regularization parameter, ϵ - width of a tube inside which no penalty for a point occurs, and σ - width of a Gaussian kernel. Despite the fact that kernel methods have being developed for about two decades there is no unified view about selection of hyper-parameters.

Authors of LIBSVM package propose using grid search where exponential grid in parameters space is formed and for each vertex separate model is trained. Parameters corresponding to the vertex which minimize validation error are selected. Other methods are: random search, Nelder-Mead simplex search, method of Cherkassky and Ma, pattern search, various methods of local search. Some empirical results are given in technical report [46]. There it is shown that for different datasets different methods perform the best; however, pattern search provides good balance between accuracy and computational cost. Pattern search is a simple method which operates on the similar parameter grid. Given an initial point it explores all possible one step moves and if some direction is superior it moves there. On the next step the size of the grid is decreased in two times and the process continues. Superiority is characterized by mean squared error on validation set.

Another appealing approach is the method proposed by Cherkassky and Ma [47]. Parameters are estimated only from the training data and no model training and validating is required.

$$C = \max(|mean_y - 3\sigma_y|, |mean_y + 3\sigma_y|) \tag{6.9}$$

$$\epsilon = 3\nu\sqrt{\frac{\log N}{N}}, \qquad \text{where } \nu - \text{variance of noise} \tag{6.10}$$

$$\sigma = (0.3\; range(\; dist(\; X_{train}\;)\;))^{\frac{1}{d}} \tag{6.11}$$

In the last equation d - is the dimensionality of the data, *dist* - obtains all pairwise distances between training data points, and *range* - obtains range of these distances. Variance of noise ν is estimated via Gamma test with number of neighbors equal 10.

Applicability of Cherkassky and Ma approach is one of the questions of this thesis. Hence, two different parameter selection procedures are applied. The First is pure Cherkassky and Ma approach, the second is Cherkassky and Ma approach with subsequent pattern search. Idea is that if pure Cherkassky and Ma method is not very accurate, then it is refined by the subsequent pattern search, which anyway needs to receive some initial point. These two SVR procedures are called *SVR_1* and SVR_2 respectively.

6.1.5 Least-Squares support vector machines

Least-Squares support vector machines (LS-SVM) are viable modification to the original SVM. For the regression problem it corresponds to the following optimization problem:

$$\begin{gathered}\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{i=N} \xi_i^2 \\ \text{s. t. } y_i - \boldsymbol{w}^T\boldsymbol{x}_i - b = \xi_i, \quad 1 \le i \le N\end{gathered} \tag{6.12}$$

As one can notice there is no ϵ-tube inside which there is no penalty. Now all deviations of regression function $\boldsymbol{w}^T\boldsymbol{x}_i + b$ from actual values y_i are penalized by addition of squared deviation to the cost function. LS-SVMs have proven to have competitive performance [48] and they have an advantage that in the dual space only system of linear equations needs to be solved. However, the sparsity of the solution is sacrificed.

To perform experiments LS-SVM Toolbox for Matlab has been used. Parameter optimization in this toolbox is done through coupled simulated annealing algorithm [49] and fine tuning through simplex method and cross-validation.

6.1.6 Regression Results

Before applying regression modeling all input variables and output variable have been normalized to have zero mean and unit variance. Then, as mentioned earlier, generalization error of different models and different subsets of input variables is measured by Monte-Carlo 15-fold cross-validation. There are 50 loops of validation and on each loop 15-fold cross-validation is performed. Best regression models and best subsets of variables are presented in the Table 6.2.

Not all results are presented here due to space constraint, only the best models and best subset of variables. Only ridge regression performs significantly worse than other methods. This is most likely because data is nonlinear in input variables while ridge regression is a linear model. Other models *i.e.* LS-SVR, SVR_1, SVR_2 usually perform similarly and differences are not high. Also it is notable that in two cases the best model is SVR_1 with hyper-parameters selection purely by Cherkassky and Ma method. It means that subsequent pattern search included in SVR_2 does not improve hyper-parameters selection. The reason might be that intrinsic cross-validation included in pattern search overfits hyper-parameters. Among the all datasets only reg_S11 has reasonably low MSE, others have substantially high taking

Dataset	Best model	Relevant variables	$MSE \pm (std)$
reg_S11	LS-SVR	"Flow S11" , "Temperature", "Integrated Flow S11", "Smoothed flow S11", "Sin Week", "Cos Week"	0.530 ± (0.312)
reg_S10_1	SVR_1	"Temperature", "Integrated Flow S10", "Smoothed Flow S10", "Rains", "Sin Week", "Rain int 1"	0.750 ± (0.548)
reg_S10_2	LS-SVR	"Temperature", "Smoothed flow S10", "Sin Week"	0.783 ± (0.489)
reg_S12_1	SVR_2	"Temperature", "Integrated Flow S12", "Cos Week", "Rain int 2", "Rain int 9"	0.814 ± (0.927)
reg_S12_2	SVR_1	"Flow S12" , "Temperature", "Integrated Flow S12", "Smoothed flow S12", "Cos Week"	0.939 ± (1.172)

Table 6.2: Relevant variables and best models for five regression datasets

into account that mean imputation would give MSE equal to 1 for normalized data.

Concerning variable selection, situation is not straightforward. "Rains" variable (and integrated rains) is selected in "reg_S10_1" and "reg_S12_1" datasets. Moreover, in the latter dataset, integrated over large period "Rain int 9" is selected. However, for the best dataset it terms of MSE - "reg_S11", and for the rest two no "Rains" are selected. Hence, even if "Rains" brings some useful information, its relevance level is not high.

To remind, datasets "reg_S10_1" and "reg_S10_2" differ only by one variable "Integrated Flow S10", so in "reg_S10_1" flow is integrated over 55 previous days while in "reg_S10_2" over 1 previous day. The same is for "reg_S12_1", "reg_S12_2" datasets. From the table it is seen that MSE for "reg_S10_1" and "reg_S12_1" are lower than for "reg_S10_2" and "reg_S12_2", however the difference is rather small. Moreover, standard deviation for "reg_S10_2" is lower than for "reg_S10_1". Thus, it is more likely that integration over 55 days is superior and it is decided to discontinue using datasets "reg_S10_2" and "reg_S12_2". So, for each location only one regression dataset is left.

There is also no clear evidence which of flow variables are is best "Flow", "Integrated Flow" or "Smoothed Flow". For different datasets various subsets of these variables are selected. This issue is analyzed again in missing values imputation approach. Another interesting fact is that both "Sin Week" and "Cos Week" are selected only in "reg_S11" dataset. In other datasets best subset includes either one or the other of these variables. Therefore, definitely some periodicity in dates is important for regression but for two locations only half of the information encoded in periodic variables turns out to be useful.

6.2 Missing values imputation

Missing values datasets have been described in Sections 5.3 and 5.4 of the previous Chapter. The list of variables and dataset sizes for location S11 are presented in the Table 5.6. As mentioned earlier, each location S10, S11, S12 has 2 datasets associated with it. They correspond to the two periods in time when measurements of "Total P" are not very sparse.

There are also several variables which importance is unknown and needed to be checked. They include time shifted versions of "Total P", and discarded variables during regression phase. However, because missing values imputation methods are quite computationally costly, comprehensive variable selection is not possible. Hence, several heuristic sets of variables are constructed and checked. Heuristics are mostly based on results of regression variable selection and general understanding of the problem. For instance, feasibility of including shifted versions of phosphorus is studied by constructing two datasets - one contain those variables and another does not.

In the following subsections description of three methods used for missing values imputation is given. Generally, missing values imputation is a wide area of research with many applications [50], [51], so there is no goal to check all possible methods and compare them. It would be hardly possible. Only a subset from different classes of methods is selected and subsequent ensemble averaging is utilized to lighten possible disadvantages of a single method. Each of the methods presented below takes as input a matrix with missing values, fills missing values and returns the complete matrix.

6.2.1 Mixture of Gaussians

It is assumed that rows of a data matrix come from the mixture of Gaussian distribution:

$$P(\boldsymbol{x}|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k) \tag{6.13}$$

Here K - number of Gaussian components. The set of parameters θ includes π_k, μ_k, and Σ_k. It is possible to introduce *latent variable* $\boldsymbol{z}$ which helps to write distribution of $\boldsymbol{x}$ in marginal form:

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k) \tag{6.14}$$

Latent variable $\boldsymbol{z}$ is a vector of length K each component of it can take either 0 or 1, and there is a constraint $\sum\limits_{j} z_j = 1$. Then, it can be shown that distribution of $\boldsymbol{x}$ can be expressed as a marginal of a distribution $p(\boldsymbol{x}, \boldsymbol{z})$ in the form of Equation 6.14. Hence, for every sample $(\boldsymbol{x}_i)_{i=1}^{N}$ there exist a corresponding sample $(\boldsymbol{z}_i)_{i=1}^{N}$ which is not observed. That is why $\boldsymbol{z}$ is called *latent* variable.

In general, fitting models with latent variables is possible via EM-algorithm [35]. The extension of EM-algorithm to include missing data has been proposed in [52], [53]. Implementation of EM-algorithm with missing values has been taken from [54]. There is one hyper-parameter of the algorithm - number of Gaussian components. The more components we take the more free parameters we have in the models and hence, the more training data is needed. After preliminary tests, it has been established that using more than 2 components is unjustified for this problem. Thus, mixtures of Gaussians with 1 and 2 components are used for imputations and their abbreviations are ”MM 1” and ”MM 2”.

6.2.2 Empirical Orthogonal Functions

Empirical orthogonal functions (EOF) is a widely used method in meteorology and climate research for missing values imputation [55]. It is based on Singular Value Decomposition (SVD), which is valid for any matrix X:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \tag{6.15}$$

Matrix $\boldsymbol{D}$ is a rectangular diagonal matrix and has the same dimensionality as original matrix $\boldsymbol{X}$. Singular values are located in the diagonal of it in the decreasing order. Matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are square and orthogonal matrices. SVD decomposition can be applied directly only to complete matrices. EOF utilizes iterative way to apply singular value decomposition to missing values imputation. The algorithm is presented below:

Algorithm 4 Empirical Orthogonal Functions

```
Given the incomplete matrix $\boldsymbol{X} \in \mathbb{R}^{m,n}$
Make initial imputation $\boldsymbol{X}^0$, for example, by column means
i = 0 (iteration number)
repeat
    Perform SVD: $\boldsymbol{X}^i = \boldsymbol{U}^i \boldsymbol{D}^i (\boldsymbol{V}^i)^T$ to obtain $\boldsymbol{U}^i, \boldsymbol{D}^i$and$(\boldsymbol{V}^i)^T$
    Nullify $K$ smallest singular values of $\boldsymbol{D}^i$. Denote this modified matrix
    as $\boldsymbol{D}_0^i$
    Do inverse transformation: $\boldsymbol{X}_0^i = \boldsymbol{U}^i \boldsymbol{D}_0^i (\boldsymbol{V}^i)^T$
    Restore exactly known values: $known(\boldsymbol{X}_0^i) = known(\boldsymbol{X}^0)$
    i = i + 1        (iteration number)
until Convergence
```

Iterations continue until convergence which is measured by maximal difference between matrix elements on two consecutive iterations. Hyperparameter in this algorithm is K - number of singular values to nullify. Actually this operation serves as data denoising and can be applied even for complete matrices. In this case is is equivalent to Principal Component Analysis. Selection of K has been performed by the same Monte-Carlo 15-fold cross-validation, in the same cycle as model selection for missing values. This validation cycle is described in more details below in Subsection 6.4.1.

6.2.3 Singular Values Thresholding

Singular value thresholding algorithm was proposed in the paper [56] as a fast but approximate algorithm to solve the following optimization problem:

$$\begin{aligned} &\min \|\boldsymbol{X}\|_\star \\ &s.t.\ X_{ij} = M_{ij} \end{aligned} \tag{6.16}$$

In the notations above $\boldsymbol{X}$ - is a complete matrix we want to obtain and M_{ij} are known values. The norm $\|\boldsymbol{X}\|_\star = \sum\limits_k \sigma_k$ is the sum of singular values of a matrix and is called *nuclear norm.* In another paper of the same authors [57] it has been shown that the problem 6.16 under some mild conditions is equivalent to minimizing rank (number of nonzero singular values) under the same constraints. The summary of the algorithm is the following:

Algorithm 5 Singular Value Thresholding

Given the incomplete matrix $\boldsymbol{M} \in \mathbb{R}^{m,n}$, denote matrix $\boldsymbol{X}^0$ which equals $\boldsymbol{M}$ and take matrix $\boldsymbol{Y}^0 = 0 \in \mathbb{R}^{m,n}$

```
repeat
    k = k + 1        (iteration number)
    $\boldsymbol{X}^k = shrink(\boldsymbol{Y}^{k-1}, \tau)$        (shrinking means applying operation $\max\{\sigma - \tau, 0\}$ to all singular values of a matrix)
    $\boldsymbol{Y}^k = \boldsymbol{Y}^{k-1} + \delta_k \mathcal{P}(\boldsymbol{M} - \boldsymbol{X}_k)$        (where operation $\mathcal{P}(\cdot)$ returns zero for incomplete element and element itself if it is known)
until Convergence
```

The implementation used in this thesis is taken from the authors of the original paper [56]. Default values for parameters have been used and there are no hyper-parameters to adjust.

6.3 Combining different models

In the previous subsections three different models have been considered. It is possible to select one of them on the basis of cross-validation outcome. The one that provides the lowest validation error can be assumed superior and others can be discarded. However, there is a reason to keep all of them and take arithmetic mean as a final estimator.

Let's assume that we have M models and we want to estimate value $f(\boldsymbol{x})$ where f might be either regression function or estimation of a missing value. Similarly $\boldsymbol{x}$ might be either regressors or present values of a matrix in an imputation problem. Assume further that each model provides estimation $y_m(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x})$, where $f(\boldsymbol{x})$ is true value and $\epsilon_m(\boldsymbol{x})$ is zero mean noise with variance $\sigma_m^2(\boldsymbol{x})$. If the noise is not zero mean then estimators are biased, but it does not change the following reasoning. Let $y(\boldsymbol{x}) = \frac{1}{M} \sum\limits_{m=1}^{M} y_m(\boldsymbol{x})$, then

$$E[y(\boldsymbol{x})] = E\left[\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right] = \frac{1}{M}\sum_{m=1}^{M} E[y_m(\boldsymbol{x})] = f(\boldsymbol{x}) \tag{6.17}$$

Hence $y(\boldsymbol{x})$ is unbiased estimator, now consider variance of $y(\boldsymbol{x})$:

$$\begin{aligned} &D[y(\boldsymbol{x})] = D\left[\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right] = \\ &\{\text{ if } \forall k, l : k \neq l\ \epsilon_k \text{ and } \epsilon_l \text{ are uncorrelated: } E[\epsilon_k \epsilon_l] = 0\ \} = \\ &= \frac{1}{M^2}\sum_{m=1}^{M} D[y_m(\boldsymbol{x})] = \frac{1}{M^2}\sum_{m=1}^{M} \sigma_m^2(\boldsymbol{x}) = \frac{\overline{\sigma^2(\boldsymbol{x})}}{M} \end{aligned} \tag{6.18}$$

So, we see that under assumption that noise is uncorrelated, variance of the mean estimator equals mean variance divided by M. One benefit is that division by M may reduce the variance significantly. Another benefit is that there is no need to find the best model. Of course, mean variance $\overline{\sigma^2(\boldsymbol{x})}$ is larger than the minimum variance of the best model, but since variances depend on $\boldsymbol{x}$, it is possible that in one part of space one model is the best and in another part of space the other model is best. Thus, by taking average, the variance of estimator is characterized by the mean variance in the given point, and there is no need to select the best model.

The assumption about uncorrelatedness of noise is rather strong and usually is not satisfied in practice. However, it is still possible to make estimation of variance of mean estimator. In Equation 6.19 it is shown that variance of averaged estimator is less (or equal) than average model variances. So, while the division by M has disappeared, but the second argument still holds.

$$\begin{aligned} &D[y(\boldsymbol{x})] = D\left[\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x})\right] = E\left[(\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x}) - f(\boldsymbol{x}))^2\right] = \\ &= E\left[(\frac{1}{M}\sum_{m=1}^{M} \epsilon_m)^2\right] \leq \{\text{using Jensen's inequality}\} \leq E\left[\frac{1}{M}\sum_{m=1}^{M} \epsilon_m^2(\boldsymbol{x})\right] = \\ &= \frac{1}{M}\sum_{m=1}^{M} \sigma_m^2(\boldsymbol{x}) = \overline{\sigma^2(\boldsymbol{x})} \end{aligned} \tag{6.19}$$

From the formulas above it is evident that using averaging is beneficial especially when errors (noise random variables) of the models are uncorrelated and their variance if low. Therefore, in this thesis averaging has been utilized and models which are used for averaging are selected by Monte-Carlo validation and greedy search. There are five models and each of those can predict phosphorus concentration for a given "Week". Models are: "Regression", "Mixture of Gaussians 1" (MM1), "Mixture of Gaussians 2" (MM2), "Empirical Orthogonal Functions" (EOF), "Singular Value Thresholding" (SVT). So, in every iteration of Monte-Carlo validation cycle, all models provide predictions and averaging is performed for all possible combinations of models. Then the best combinations are selected by validation results.

6.4 Model selection results

6.4.1 Experimental setup

Experiments are done in the similar way as regression experiments. Accuracy of imputation is characterized by MSE and is measured by Monte-Carlo 15-fold cross-validation. There are 50 iterations in total, on each of those dataset is randomly permuted and 15-fold cross-validation is performed. The final estimation of MSE is an average over each fold within one iteration and total average over all iterations. Iterations of cross-validation are required because datasets are very small - only around 225 samples. For the same reason number of folds in cross-validation is increased from standard 10 to 15.

There are two missing values datasets for each location as described in Section 5.3. They correspond to different time periods and denoted as "Part 1" and "Part 2". All experiments are conducted in a way that these two datasets are processed simultaneously. Validation is done independently on each Monte-Carlo iteration because datasets have different number of samples. However, when variable selection is performed, different groups of variables are evaluated on exactly the same points of cross-validation.

All matrices have been normalized in the beginning so that each column has zero mean and unit variance. Under this normalization, mean square error of mean imputation equals one. That way is easier to compare accuracies for different datasets. For example, if MSE is below 1 than performance of a method is better than performance of mean imputation.

6.4.2 Usefulness of other locations in missing values datasets

In the previous chapter it is written that missing values dataset for each location includes phosphorus concentration of other locations. The feasibility of that inclusion is tested here. This is done only for location S11 and obtained results are generalized for datasets of other locations. In the Table 6.3 four groups of variables are presented. First group includes all variables, in the second - two other locations are discarded, in the third one - time shifts of phosphorus are also discarder and in the fourth group all locations are kept but time shifts are eliminated.

No.	Variable Name	Group 1	Group 2	Group 3	Group 4
1	"Flow S11"	✓	✓	✓	✓
2	"Total P S11"	✓	✓	✓	✓
3	"Total P S10"	✓			✓
4	"Total P S12"	✓			✓
5	"Temperature"	✓	✓	✓	✓
6	"Integrated Flow S11"	✓	✓	✓	✓
7	"Smoothed Flow S11"	✓	✓	✓	✓
8	"Rains"	✓			
9	"Sin Week"	✓	✓	✓	✓
10	"Cos Week"	✓	✓	✓	✓
11	"Time shift 1 Ph. S11"	✓	✓		
12	"Time shift 2 Ph. S11"	✓	✓		
13	"Time shift 1 Ph. S10"	✓			
14	"Time shift 2 Ph. S10"	✓			
15	"Time shift 1 Ph. S12"	✓			
16	"Time shift 2 Ph. S12"	✓			

Table 6.3: Groups of variables to be tested for usefulness of inclusion

Data matrix	Group 1	Group 2	Group 3	Group 4
Part 1 ($MSE \pm std$)	0.46 ± 0.56	0.62 ± 0.61	0.62 ± 0.58	0.47 ± 0.61
models mask	10001	10000	10000	10001
Part 2 ($MSE \pm std$)	0.40 ± 1.01	0.53 ± 0.95	0.52 ± 0.95	0.40 ± 1.04
models mask	10111	11000	11000	11001

Table 6.4: Performance evaluation for variable groups. Sequence of models in model mask is: "Regression", "MM 1", "MM 2", "SVT", "EOF"

Comparison of these groups of variables is showed the Table 6.4 for both "Part 1" and "Part 2" matrices. Combinations of models which provided the best MSE are also showed in this table in the form of model mask. In the mask 1 means that model has been used in averaging and 0 that it has not. Sequence of models in model mask is: "Regression", "MM 1", "MM 2", "SVT", "EOF". The best regression model and best variables revealed in regression analysis has been utilized for every dataset.

From the table it is clearly seen that using all variables is beneficial in terms of MSE. However, results are only slightly better than for the fourth group, and sometimes only small difference in standard deviation matters. In the second group variables corresponding to other locations are missing, and relatively large growth of MSE is observed. One can notice that "Rains" variable is missing in groups 2 and 3. This is done because this variable is not selected by regression model which is seen form the Table 6.2. Therefore, it is believed that this variable has no relevant information. There is no difference in accuracy (after rounding) between groups 2 and 3. It is also noticeable that in "Part 1" dataset, for groups 2 and 3, only regression model is selected and missing values datasets are not needed at all. It is decided to keep time shifted versions of phosphorus even though their advantage is very minor.

6.4.3 Model selection for missing values imputation

From the previous section it is known that including phosphorus concentration for other locations and time shifted values of phosphorus might be beneficial. However, for final imputation it is necessary to know which model combination and which set of variables to use for each location. To determine this, experiments have been performed and results are presented in Tables 6.5, 6.6, 6.7.

For each location there are two datasets and it is possible to do model selection independently for each of those. However, to facilitate interpretability of the results it is desirable that for each location, there is only one set of useful variables and models in ensemble.

In the Tables 6.5, 6.6, 6.7 groups of variables that have been evaluated are presented. Because the evaluation procedure requires many cycles of cross-validation (Section 6.4.1) it is impossible to test many variable groups. For example, in the experiment for S12 location with four groups, computations take approximately 20 hours. Therefore, variable groups have been selected on the basis of regression results (Table 6.2) and domain knowledge.

Experimental output for one location consist of MSE and STD for "Part 1" and "Part 2" datasets, for all variable groups (which are analyzed) and all model combinations. Then, this results are manually processed to find optimal variable group and model combination. So, sometimes optimality is a compromise between "Part 1" and "Part 2" datasets, because for one of them one model combination is better while for another some different combination. Compromise has been searched in terms of MSE and STD, both are required to be as low as possible. However, as was said earlier, if we allow different variable groups or different model combinations for "Part 1" and "Part 2" interpretability of results is decreased significantly.

Location S10				
No.	Variable Name	Group 1	Group 2	Group 3
1	”Flow S10”	✓		
2	”Total P S10”	✓	✓	✓
3	”Total P S11”	✓	✓	✓
4	”Total P S12”	✓	✓	✓
5	”Temperature”	✓	✓	✓
6	”Integrated Flow S10”	✓	✓	✓
7	”Smoothed Flow S10”	✓	✓	✓
8	”Rains”	✓	✓	✓
9	”Sin Week”	✓	✓	✓
10	”Cos Week”	✓	✓	✓
11	”Time shift 1 Ph. S10”	✓	✓	
12	”Time shift 2 Ph. S10”	✓	✓	
13	”Time shift 1 Ph. S11”	✓	✓	
14	”Time shift 2 Ph. S11”	✓	✓	
15	”Time shift 1 Ph. S12”	✓	✓	
16	”Time shift 2 Ph. S12”	✓	✓	
Best model combination 10111: ”Regression SVR_1”, ”MM 2”,”SVT”,”EOF”				
$MSE \pm \mathit{std}$, Part 1		$\mathbf{0.606 \pm 0.639}$	0.629 ± 0.637	0.667 ± 0.710
$MSE \pm \mathit{std}$, Part 2		$\mathbf{0.314 \pm 0.297}$	0.341 ± 0.368	0.337 ± 0.336

Table 6.5: Groups of variables which have been tested for missing values imputation for location S10

Location S11				
No.	Variable Name	Group 1	Group 2	Group 3
1	"Flow S11"	✓	✓	✓
2	"Total P S11"	✓	✓	✓
3	"Total P S10"	✓	✓	✓
4	"Total P S12"	✓	✓	✓
5	"Temperature"	✓	✓	✓
6	"Integrated Flow S11"	✓	✓	✓
7	"Smoothed Flow S11"	✓	✓	✓
8	"Rains"	✓		
9	"Sin Week"	✓	✓	✓
10	"Cos Week"	✓	✓	✓
11	"Time shift 1 Ph. S11"	✓	✓	
12	"Time shift 2 Ph. S11"	✓	✓	
13	"Time shift 1 Ph. S10"	✓	✓	
14	"Time shift 2 Ph. S10"	✓	✓	
15	"Time shift 1 Ph. S12"	✓	✓	
16	"Time shift 2 Ph. S12"	✓	✓	
Best model combination 10001: "Regression LS-SVM", "EOF"				
$MSE \pm std$, Part 1		$\mathbf{0.503 \pm 0.599}$	0.504 ± 0.634	0.503 ± 0.637
$MSE \pm std$, Part 2		$\mathbf{0.343 \pm 0.611}$	0.340 ± 0.665	0.340 ± 0.662

Table 6.6: Groups of variables which have been tested for missing values imputation for location S11

From results tables it is seen that accuracy for "Part 1" is always better than for "Part 2". For location S12 the difference is very large. This might be due to the fact that time lags between given values of phosphorus are larger for "Part 1" as it is seen for the Figure 5.2. Another noticeable fact is that best groups for locations S10 and S11 are first groups which include all variables. This goes in accordance with the statement in Subsection 6.4.2 that time shifted variables are important. However, variables rejected by regression approach, for instance "Flow S10" for S10 dataset, are not rejected by missing values approach. On the other hand, for S12 location variables that are rejected by regression are also rejected by missing values approach. Thus, it is concluded that variables rejected by regression are of limited relevance. They do not provide significant accuracy improvement.

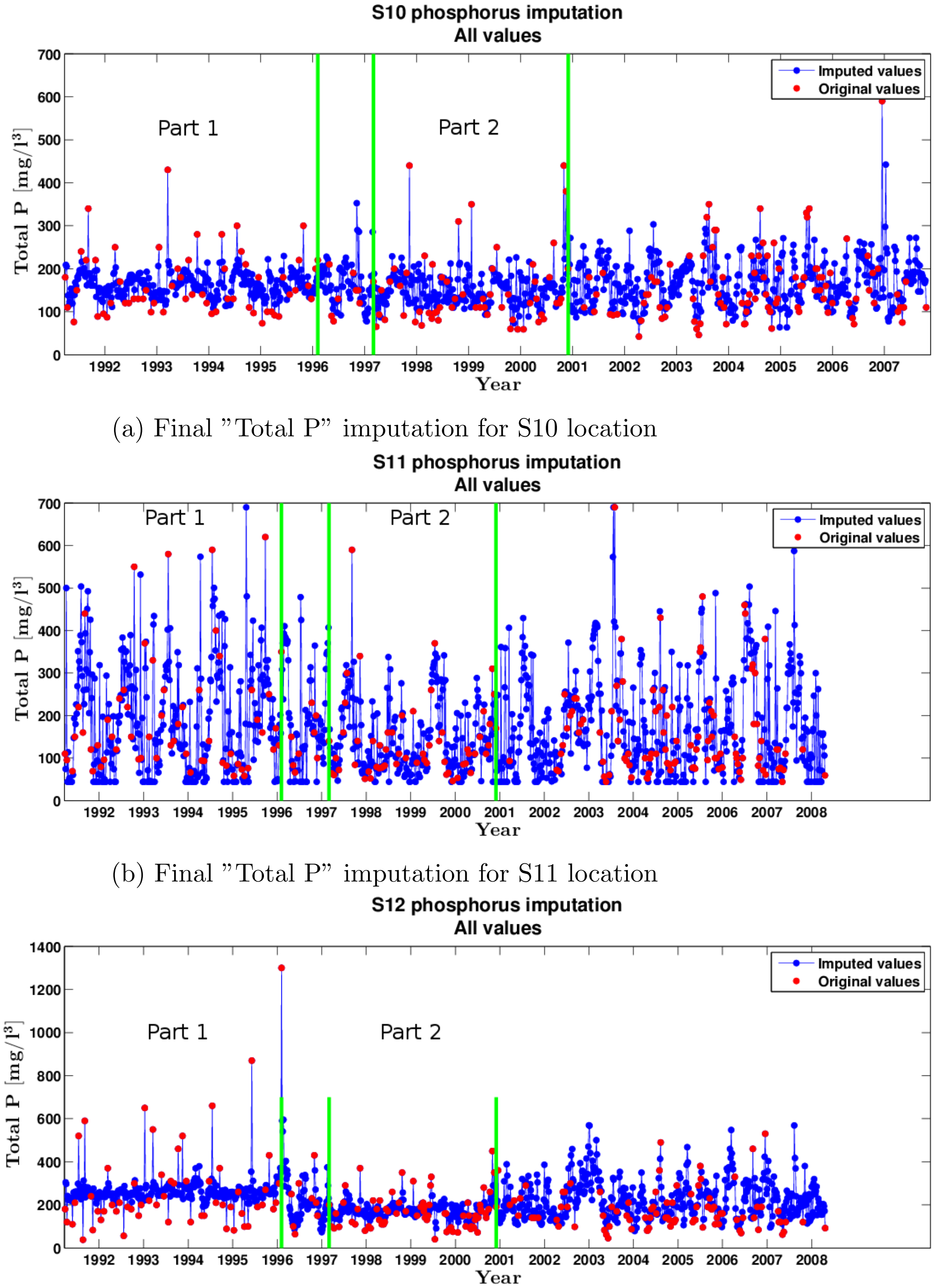
Concerning different models, it is seen from the table that each of them participated at least once in some model combination. For example, for location S12 all models except "MM 2" are selected, for S10 everything except "MM 1" are selected. The important fact is that regression model is always selected, but never alone. This means that using missing values approach improves results of regression.

6.5 Final estimation of phosphorus concentration

The final estimation of phosphorus concentration is done for time period 26.03.1991 - 21.04.2008. Outside this time interval no "Total P" measurements have been done so we decided constrain only to this period. Graphically this time interval is shown on the Figure 5.2. As earlier, three locations S10, S11, S12 have been used.

Inside this large time interval there are two short intervals where phosphorus concentration is not very sparse and missing values approach can be utilized. For example, for S11 location these intervals are (26.03.1991 - 06.02.1996) and (01.03.1997 - 01.12.2000) and correspond to missing values datasets called "Part 1" and "Part 2" as before. For other locations small intervals are almost the same.

At first, for each location, regression model is build. The selection between "LS-SVM", "SVM_1" and "SVM_2" is done looking at results of Table 6.2. Then two datasets - "Part 1" and "Part 2" are build according to the best variable groups from previous section. Missing values imputation is performed for this datasets using best model combination for this location (Tables 6.5, 6.6, 6.7). Therefore, for dates which correspond to dates of missing values datasets, we get an improved ”Total P” estimation in comparison to only regression approach. For other dates regression predictions are taken.



(a) Final "Total P" imputation for S10 location

(b) Final "Total P" imputation for S11 location

(c) Final "Total P" imputation for S12 location

Figure 6.1: "Total P" imputation

Location S12					
No.	Variable Name	Group 1	Group 2	Group 3	Group 4
1	"Flow S12"	✓			
2	"Total P S12"	✓	✓	✓	✓
3	"Total P S10"	✓	✓	✓	✓
4	"Total P S11"	✓	✓	✓	✓
5	"Temperature"	✓	✓	✓	✓
6	"Integrated Flow S12"	✓	✓	✓	✓
7	"Smoothed Flow S12"	✓			
8	"Rains"	✓			
9	"Sin Week"	✓			✓
10	"Cos Week"	✓	✓	✓	✓
11	"Time shift 1 Ph. S12"	✓	✓		
12	"Time shift 2 Ph. S12"	✓	✓		
13	"Time shift 1 Ph. S10"	✓	✓		
14	"Time shift 2 Ph. S10"	✓	✓		
15	"Time shift 1 Ph. S11"	✓	✓		
16	"Time shift 2 Ph. S11"	✓	✓		
Best model combination 11011: "Regression SVR_2", "MM 1","SVT","EOF"					
$MSE \pm std$, Part 1		0.795 ± 1.474	$\mathbf{0.777 \pm 1.469}$	0.753 ± 1.437	0.791 ± 1.531
$MSE \pm std$, Part 2		0.497 ± 0.494	$\mathbf{0.453 \pm 0.457}$	0.567 ± 0.459	0.564 ± 0.563

Table 6.7: Groups of variables which have been tested for missing values imputation for location S12

Chapter 7

Conclusions

The purpose of this thesis is twofold. At first, the problem of pure time series prediction has been addressed. Time series prediction can be applied to variety of problems, including environmental modeling which is practically investigated in this thesis on the Pyhäjärvi case. Lake Pyhäjärvi is large lake which plays crucial role in the local agricultural and fishing industries. It suffers from excessive growth of plants which cause death of animals from the lack of oxygen. Plants grow abundantly because of a large load of nutrients into the lake and the main nutrient is phosphorus. For intelligent planning of preservation activities and better understanding of this ecological system it is necessary to have model which predicts the concentration of phosphorus. Time series modeling seems very natural approach to phosphorus concentration prediction, and methods developed in this thesis can be successive utilized.

However, there is significant impediment to the direct application of time series prediction techniques to the phosphorus concentration prediction in Pyhäjärvi lake. The measurements of phosphorus concentration are done manually and not very regularly. Therefore, weeks when there are no measurements may be considered as missing data. Imputation of missing values constitutes the second part of the thesis. Two approaches to estimate missing values of phosphorus are studied: regression approach and missing values approach. Moreover, datasets provided by Pyhäjärvi institute include many variables importance of which is unclear. So, variable selection is performed and relevance of variables is evaluated.

Conclusions about the first part of the thesis are summarized in detail in the Section 4.4 of Chapter 4. In particular, OP-ELM with DirRec strategy has demonstrated very good performance. In all experiments this combination outperforms linear model with any strategy. In the case, when it does not hold the difference is very small. It has been confirmed that OP-ELM requires much less computations then other nonlinear models like SVM. Ensemble methods are able to provide further substantial improvement of prediction accuracy.

Second part starts with preprocessing and exploratory data analysis. At this stage some important decisions have been made and framework for further analysis has been established. In particular, locations are grouped together on the basis of correlation analysis. Datasets for each of locations S10, S11 and S12 are included phosphorus concentrations from two other locations. Conjecture has been made and investigated that integrated flow could affect phosphorus concentration more significantly than flow from the corresponding week. Smoothed flow intensity has been included into the data sets because sharp changes could negatively influence performance of algorithms. Integrated versions of ”Rains” variable, sine and cosine of week number have been included as well. The important decision has been made about averaging all variables over five day intervals. It helps to avoid extremely sparse datasets and align other variables with precipitation variable which is given only for five day intervals.

Two levels of variable selection have been applied in regression approach. It has been demonstrated that ”Temperature” and ”Integrated Flow” are always selected as relevant features for phosphorus predictions. At least one of two periodic time variables (sine or cosine) is also selected. Relevance of ”Rains” variables and ”Integrated Rains” are dubious at this stage because they are rejected by two datasets out of three. Three regression models *i.e.* LS-SVM, SVM_1 and SVM_2 has been investigated and their performance appears to be very similar. All three datasets select different models as the best. Thus, it is shown that method of Cherkassky and Ma of hyperparameters selection is competitive against greedy cross-validation, at least for Pyhäjärvi data.

Further improvements over regression approach are done via missing values approach. It is applied for time periods where gaps between known values of phosphorus are not very large. It has been shown in Subsection 6.4.2 that including other locations and shifted versions of phosphorus leads to lower mean squared error. Inclusion of shifted versions of a variable into a missing variable dataset is an interesting technique which has been proposed in this thesis.

Missing values imputation has been done by ensemble method using five different models. Comparison of Tables 6.2 and 6.4 shows that results of ensemble of imputation methods outperform the regression approach. Even single model for missing values imputation may be more accurate than results of regression. So, for time periods of missing values datasets, improved estimation of phosphorus concentrations are obtained. Finally, phosphorus concentration is filled for three locations S10, S11, S12 for the time interval (26.03.1991 - 21.04.2008). As a rough estimation of confidence interval of imputed phosphorus, properly normalized mean squared error for corresponding date can be taken.

Methods applied in the second part of the thesis often follow an ad-hoc way. It is believed that more general and systematic approaches can be developed. Plenty of artificially generated variables like integrated flows and rains have been evaluated. Since the generation is done in a systematic way, interesting research direction would be to automate the selection as well. Also interesting technique is a combination of missing value imputation and regression approaches. In environmental domain data is often collected manually which is quite costly, so presence of missing values is a constant problem. Therefore, using all available data *i.e.* complete variables and incomplete ones for solving the task, is frugal and effective way to exploit every bit of information which we possess.

Bibliography

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

[2] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16-18):2861–2869, October 2007.

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

[4] 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*, 21(1):158–162, 2010.

[5] Lennart Ljung. *System identification (2nd ed.): theory for the user.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[6] Yong Yu, Tsan-Ming Choi, and Chi-Leung Hui. An intelligent fast sales forecasting model for fashion products. *Expert Syst. Appl.*, 38(6):7373–7379, 2011.

[7] R. Maclin and D. Opitz. Popular ensemble methods: An empirical study. *Arxiv preprint arXiv:1106.0257*, 2011.

[8] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse. GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, 74(16):2430–2437, 2011.

[9] Y. Lan, Y.C. Soh, and G.B. Huang. Ensemble of online sequential extreme learning machine. *Neurocomputing*, 72(13-15):3391–3395, 2009.

[10] M. van Heeswijk, Y. Miche, T. Lindh-Knuutila, P.A.J. Hilbers, T. Honkela, E. Oja, and A. Lendasse. Adaptive ensemble models of extreme learning machines for time series prediction. In Cesare Alippi,

Marios M. Polycarpou, Christos G. Panayiotou, and Georgios Ellinas, editors, *ICANN 2009, Part II*, volume 5769 of *LNCS*, pages 305–314, Heidelberg, 2009. Springer.

[11] Zhan Li Sun, Tsan Ming Choi, Kin Fan Au, and Yong Yu. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decis. Support Syst.*, 46(1):411–419, 2008.

[12] Jurga Ruksenaite and Pranas Vaitkus. Prediction of composite indicators using combined method of extreme learning machine and locally weighted regression. *Nonlinear Analysis: Modelling and Control*, 17(2):238–251, 2012.

[13] M.H. Hayes. *Statistical digital signal processing and modeling*. Wiley-India, 2009.

[14] A. Sorjamaa and A. Lendasse. Time series prediction using DirRec strategy. In M. Verleysen, editor, *ESANN06, European Symposium on Artificial Neural Networks*, pages 143–148, Bruges, Belgium, April 26-28 2006.

[15] G.B. Huang, Q.Y. Zhu, and C.K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

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

[17] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *Neural Networks, IEEE Transactions on*, 17(4):879–892, July 2006.

[18] T. Similä and J. Tikka. Multiresponse sparse regression with application to multidimensional scaling. In *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005*, volume 3697/2005, pages 97–102. 2005.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. In *Annals of Statistics*, volume 32, pages 407–499. 2004.

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

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

[22] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[23] T. Schreiber. Detecting and analyzing nonstationarity in a time series using nonlinear cross predictions. *Physical Review Letters*, 78(5):843–846, 1997.

[24] L. Cao and Q. Gu. Dynamic support vector machines for non-stationary time series forecasting. *Intelligent Data Analysis*, 6(1):67–83, 2002.

[25] F. Montesino-Pouzols and A. Lendasse. Effect of different detrending approaches on computational intelligence models of time series. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1729–1736, Barcelona, Spain, July 2010.

[26] J. Cao, Z. Lin, G.-B. Huang, and N. Liu. Voting based extreme learning machine. *Information Sciences*, 185(1):66–77, 2012.

[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)*, pages 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] Amaury Lendasse, editor. *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*, 21(7):1056–1067, 2005.

[32] A-M. Ventelä, T. Kirkkala, A. Lendasse, M. Tarvainen, H. Helminen, and J. Sarvala. Climate-related challenges in long-term management of säkylän pyhäjärvi (SW finland). *Hydrobiologia*, 660:49–58, 2011.

[33] A. Stuart, K. Ord, and S. Arnold. *Kendall's Advanced Theory of Statistics, Classical Inference and the Linear Model.* Number nid. 2 in Kendall's library of statistics. John Wiley & Sons, 2009.

[34] Louis Guttman. A note on the derivation of formulae for multiple and partial correlation. *Annals of mathematical statistics*, 9:305–308, 1938.

[35] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[36] D. François. *High-Dimensional Data Analysis.* VDM Publishing, 2008.

[37] A. Guillén, D. Sovilj, F. Mateo, I. Rojas, and A. Lendasse. Minimizing the delta test for variable selection in regression problems. *International Journal of High Performance Systems Architecture*, 1(4):269–281, 2008.

[38] Antonia Jones. New tools in non-linear modelling and prediction. *Computational Management Science*, 1(2):109–149, 07 2004.

[39] E. Eirola, E. Liitiäinen, A. Lendasse, F. Corona, and M. Verleysen. Using the delta test for variable selection. In M. Verleysen, editor, *Proceedings of ESANN 2008, European Symposium on Artificial Neural Networks, Bruges (Belgium)*, pages 25–30. d-side publ. (Evere, Belgium), April 23-25 2008.

[40] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970.

[41] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[42] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[43] Harris Drucker, Chris J C Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Electronic Engineering*, 1(June):155–161, 1997.

[44] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA, 2004.

[45] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011.

[46] Marcos Marin-Galiano, Karsten Luebke, Andreas Christmann, and Stefan Rüping. Determination of hyper-parameters for kernel based classification and regression. Technical Report / Universität Dortmund, SFB 475 Komplexitätsreduktion in Multivariaten Datenstrukturen 2005,38, 2005.

[47] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural Netw.*, 17(1):113–126, 2004.

[48] J.A.K. Suykens. *Least Squares Support Vector Machines.* World Scientific, 2002.

[49] Samuel Xavier de Souza, Johan A. K. Suykens, Joos Vandewalle, and Désiré Bollé. Coupled simulated annealing. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 40(2):320–335, 2010.

[50] Roderick J.A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data.* Wiley, 2 edition, 2002.

[51] Tommi Vatanen. Missing value imputation using subspace methods with applications on survey data. Master's thesis, Aalto University, Espoo, Finland, 2012.

[52] Zoubin Ghahramani and Michael I. Jordan. Learning from incomplete data. Technical report, Cambridge, MA, USA, 1994.

[53] Lynette Hunt and Murray Jorgensen. Mixture model clustering for mixed data with missing information. *Comput. Stat. Data Anal.*, 41(3-4):429–440, January 2003.

[54] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki. Mixture of gaussians for distance estimation with missing data. In *Machine Learning Reports 03/2012*, pages 37–45, 2012. Proceedings of the Workshop - New Challenges in Neural Computation 2012.

[55] R.W. Preisendorfer and C.D. Mobley. *Principal component analysis in meteorology and oceanography.* Developments in atmospheric science. Elsevier, 1988.

[56] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, March 2010.

[57] Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, December 2009.