TKK Dissertations in Information and Computer Science
Espoo 2010 TKK-ICS-D21

METHODOLOGIES FOR TIME SERIES PREDICTION AND MISSING VALUE IMPUTATION

Antti Sorjamaa

Aalto University
School of Science
and Technology

TKK Dissertations in Information and Computer Science

Espoo 2010 TKK-ICS-D21

METHODOLOGIES FOR TIME SERIES PREDICTION AND MISSING VALUE IMPUTATION

Antti Sorjamaa

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T2 at the Aalto University School of Science and Technology (Espoo, Finland) on the 19th of November, 2010, at 12 noon.

Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Aalto-yliopiston teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
PO Box 15400
FI-00076 AALTO
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 470 01
Fax +358 9 470 23369
E-mail: series@ics.tkk.fi

ISBN 978-952-60-3452-2 (Print)
ISBN 978-952-60-3453-9 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)
URL: http://lib.tkk.fi/Diss/2010/isbn9789526034539

Multi-print
Espoo 2010

ABSTRACT: The amount of collected data is increasing all the time in the world. More sophisticated measuring instruments and increase in the computer processing power produce more and more data, which requires more capacity from the collection, transmission and storage.

Even though computers are faster, large databases need also good and accurate methodologies for them to be useful in practice. Some techniques are not feasible to be applied to very large databases or are not able to provide the necessary accuracy.

As the title proclaims, this thesis focuses on two aspects encountered with databases, time series prediction and missing value imputation. The first one is a function approximation and regression problem, but can, in some cases, be formulated also as a classification task. Accurate prediction of future values is heavily dependent not only on a good model, which is well trained and validated, but also preprocessing, input variable selection or projection and output approximation strategy selection. The importance of all these choices made in the approximation process increases when the prediction horizon is extended further into the future.

The second focus area deals with missing values in a database. The missing values can be a nuisance, but can be also be a prohibiting factor in the use of certain methodologies and degrade the performance of others. Hence, missing value imputation is a very necessary part of the preprocessing of a database. This imputation has to be done carefully in order to retain the integrity of the database and not to insert any unwanted artifacts to aggravate the job of the final data analysis methodology. Furthermore, even though the accuracy is always the main requisite for a good methodology, computational time has to be considered alongside the precision.

In this thesis, a large variety of different strategies for output approximation and variable processing for time series prediction are presented. There is also a detailed presentation of new methodologies and tools for solving the problem of missing values. The strategies and methodologies are compared against the state-of-the-art ones and shown to be accurate and useful in practice.

KEYWORDS: Time Series Prediction, Missing Values, Large Databases, Prediction Strategy, Variable Selection, Nonlinear Imputation, EOF Pruning, Ensemble of SOMs

TIIVISTELMÄ: Maailmassa tuotetaan koko ajan enemmän ja enemmän tietoa. Kehittyneemmät mittalaitteet, nopeammat tietokoneet sekä kasvaneet siirto- ja tallennuskapasiteetit mahdollistavat suurien tietomassojen keräämisen, siirtämisen ja varastoinnin.

Vaikka tietokoneiden laskentateho kasvaa jatkuvasti, suurten tietoaineistojen käsittelyssä tarvitaan edelleen hyviä ja tarkkoja menetelmiä. Kaikki menetelmät eivät sovellu valtavien aineistojen käsittelyyn tai eivät tuota tarpeeksi tarkkoja tuloksia.

Tässä työssä keskitytään kahteen tärkeään osa-alueeseen tietokantojen käsittelyssä: aikasarjaennustamiseen ja puuttuvien arvojen täydentämiseen. Ensimmäinen näistä alueista on regressio-ongelma, jossa pyritään arvioimaan aikasarjan tulevaisuutta edeltävien näytteiden pohjalta. Joissain tapauksissa regressio-ongelma voidaan muotoilla myös luokitteluongelmaksi.

Tarkka aikasarjan ennustaminen on riippuvainen hyvästä ja luotettavasta ennustusmallista. Malli on opetettava oikein ja sen oikeellisuus ja tarkkuus on varmistettava. Lisäksi aikasarjan esikäsittely, syötemuuttujien valinta- tai projektiotapa sekä ennustusstrategia täytyy valita huolella ja niiden soveltuvuus mallin yhteyteen on varmistettava huolellisesti. Tehtyjen valintojen tärkeys kasvaa entisestään mitä pidemmälle tulevaisuuteen ennustetaan.

Toinen tämän työn osa-alue käsittelee puuttuvien arvojen ongelmaa. Tietokannasta puuttuvat arvot voivat heikentää data-analyysimenetelmän tuottamia tuloksia tai jopa estää joidenkin menetelmien käytön, joten puuttuvien arvojen arviointi ja täydentäminen esikäsittelyn osana on suositeltavaa. Täydentäminen on kuitenkin tehtävä harkiten, sillä puutteellinen täydentäminen johtaa hyvin todennäköisesti epätarkkuuksiin lopullisessa käyttökohteessa ja ei-toivottuihin rakenteisiin tietokannan sisällä. Koska kyseessä on esikäsittely, eikä varsinainen datan hyötykäyttö, puuttuvien arvojen täydentämiseen käytetty laskenta-aika tulisi minimoida säilyttäen laskentatarkkuus.

Tässä väitöskirjassa on esitelty erilaisia tapoja ennustaa pitkän ajan päähän tulevaisuuteen ja keinoja syötemuuttujien valintaan. Lisäksi uusia menetelmiä puuttuvien arvojen täydentämiseen on kehitetty ja niitä on vertailtu olemassa oleviin menetelmiin.

AVAINSANAT: Aikasarjaennustaminen, puuttuvien arvojen täydentäminen, suuret tietojoukot, ennustusstrategia, muuttujien valinta, Empiiristen ortogonaalifunktioiden valinta, Itseorganisoituvien karttojen yhdistelmä.

Contents

LIST OF FIGURES

LIST OF TABLES

NOTATION

α	Linear model parameter
Θ	Set including all necessary model parameters
c	Number of SOM nodes
d	Number of dimensions or input variables
$\mathbf{D}$	Diagonal matrix of singular values
f	Approximation model
h	Discretization parameter for EFB
k	Number of neighbors in k-NN
l	Number of neighbors in MI
$\mathbf{m}$	Matrix of all SOM node weights
$\mathbf{m_i}$	Vector of weights of SOM node i
N	Number of samples or data points
p	Number of neighbors in GT
p	Projection dimension
$\mathbf{P}$	Projection matrix
S	Set of input variables
$\mathbf{U}$, $\mathbf{V}$	Matrices of left and right singular vectors
$\mathbf{x}_i$	i^{th} data sample
$\mathbf{X}$	Collection of all inputs of all samples
$\mathbf{X}^i$	Dataset with i^{th} variable selected from all samples
y	Time series
y_i	i^{th} value of time series y
$\hat{y}_i$	Approximation of i^{th} value of time series y
Y	Collection of all outputs of all samples
$\mathbf{z}$	Projected sample

ACRONYMS

AIC	Akaike's Information Criterion
ARMA	Autoregression Moving Average
ARX	Autoregression with eXternal variables
BIC	Bayesian Information Criterion
BMU	Best Matching Unit
EFB	Extended Forward-Backward projection strategy
EOF	Empirical Orthogonal Functions
FB	Forward-Backward selection strategy
GA	Genetic Algorithm
GT	Gamma Test search criterion
HQ	Hanna-Quinn information criterion
ICA	Independent Component Analysis
k-NN	k-Nearest Neighbors method
LL	Lazy Learning or Local Learning
LOO	Leave-One-Out validation method
MI	Mutual Information search criterion
MIMO	Multiple-Inputs Multiple-Outputs
MISMO	Multiple-Inputs Several Multiple-Outputs
MSE	Mean Square Error
NAR	Nonlinear Autoregressive
NNE	Nonparametric Noise Estimation
NNLS	NonNegative Least Squares
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
PPCA	Probabilistic PCA
PLS	partial Least Squares
PRESS	PREdiction Sum of Squares
RMS	Root Mean Square error
SOM	Self-Organizing Maps
SVD	Singular Value Decomposition
TSP	Time Series Prediction

PREFACE

This work has has been carried out at the Department of Information and Computer Science (until 31st of December 2008 known as Laboratory of Computer and Information Science) of Aalto University School of Science and Technology (until 31st of December 2009 known as Helsinki University of Technology).

A large part of this work has been funded by the Graduate School of Computer Science and Engineering of Helsinki University of Technology. The work has also been funded by the Adaptive Informatics Research Centre (AIRC), the Academy of Finland national Centre of Excellence. Moreover, personal grants awarded by Tekniikanedistämissäätiö (TES) and Nokia Foundation are gracefully acknowledged. Also Helsinki Graduate School in Computer Science and Engineering (HeCSe) has generously supported conference trips and research visits.

I am very grateful to my supervisor Professor Olli Simula and my instructor Docent Amaury "Momo" Lendasse for their endless help, encouragement and motivation throughout the process of writing this thesis as well as during the research work experience.

I'm also thankful to the director of AIRC, Professor Erkki Oja, and the head of the department, Professor Pekka Orponen, for the first-class facilities to do the research and for arranging all other activities to keep me motivated during the years. It has been very challenging, but also very productive to fulfill the requirements for a good thesis.

I'm also heavily in debt to the pre-examiners Doctor Madalina Olteanu and Professor Vincent Wertz, for their excellent suggestions and comments for improving my manuscript. Large commendation also to my opponent, Professor Guilherme Barreto, it has been an honor to have you as the opponent for my defense.

Very big thanks also to the members of the Environmental and Industrial Machine Learning (EIML) research group, it has been a pleasure to work with you, Ladies and Gentlemen, for all these years. Special thanks for Doctor Yoan Miché and M.Sc. Emil Eirola for their patience during the last few months of the thesis work. I wish you both all the shine and glimmer for the future.

I also want to thank all my coauthors for their excellent work, support and help, without them this thesis would not have been possible. May your paths be paved with fortune and good luck to overcome whatever obstacles there might come about.

Finally, I want to thank my family, my friends and various Gods for keeping me sane, happy and safe, respectively, for all those dark and stressful years.

Antti Sorjamaa

1 INTRODUCTION

1.1 SCOPE OF THE THESIS

Due to the constantly growing computing power and more and more intelligent measurement systems, the amount and size of databases is growing faster and faster. Measurement devices can measure multiple things at the same time, more accurately than before and with higher frequency. These devices require fast data collection, broad bandwidth transmitting and large storage capacities to handle the enormous data flow from the instruments to data analysts.

Since more and more data are measured, transported and stored, especially with new and still unstable devices, the need for verifying and handling the data in a smart way is also emphasized. One does not want to apply a preprocessing on a huge dataset, just to find out later that the result is not good and the processing needs to be corrected and applied again. Even though the computing power increases all the time, the methodological parts need to be improved as well.

In many cases, the collected databases include spatial and / or temporal relativity in their values. These datasets can be called spatially- and / or temporally-related databases and techniques, strategies and methodologies invented to deal with this kind of databases can be used. These methods are numerous with varying applicability to different problems. Some strategies and methodologies are not able to handle large datasets in a reasonable time and some just are not providing the accuracy required.

The size of the databases also leads to the necessity of variable selection. It is not advisable to try to use the whole database for a process that would require only a small piece to be completed adequately. Since larger databases require more memory and processing power, not to mention the applicable methodologies to begin with, it is better to select the necessary variables from the database before the process is started.

On the other hand, since the data collection processes are not perfect, the databases may contain missing values. These missing parts can distort the results, create inaccuracies and even repudiate the usage of several techniques. If the size of the database is large enough, one could just discard the part of the database with missing values, but in many cases this is not possible course of action. The accurate and reliable usage of the database requires that the missing parts are filled beforehand.

As an example of a recently collected database is the temperature database of Lake Tanganyika. Lake Tanganyika is located in the African

Rift in the center of the African continent. The extraordinary size and shape of the lake make it really valuable for the climate research, but the size and the shape of the lake make it hard to adequately measure the bio-geo-physical parameters, such as surface temperature. Given the current political and economical situation in Africa, the satellite is the only valid option to conduct the measurements.

The data measured by satellite includes a vast number of missing values, more than 60 percent of the database, due to clouds, technical difficulties and even heavy smoke from forest fires. The missing values make *a posteriori* modelling a difficult problem and the filling procedure a mandatory preprocessing step before climate modelling.

From the modeling perspective, there are several mandatory settings and parameters to be selected in order to get the optimal combination for the task. The selection is needed for how to fill the missing values, what other preprocessing is needed, which variables should be most related to the problem at hand, which methodology to use and last, but not least, how to train and validate the selected model. Since there are many possibilities in each step, there is no way to test them all, and hence, it is not possible to highlight the only combination to the best of them all. Therefore, it is advisable to test as many combinations as possible and select the best one among them.

What makes the selection problem even harder is that if one of the selections is changed, the whole chain of determined choices have to be re-evaluated. For example, if the preprocessing is changed, it requires the re-evaluation of the selection of the variables, which again implies the re-evaluation of the methodology selection and so on.

1.2 CONTRIBUTIONS PRESENTED IN THE THESIS

This thesis deals with two closely related problems; time series prediction and missing values. Both fields share similarities, but have their own peculiarities as well. A full collection of different strategies and several output approximation methods for time series prediction and for finding missing values in a temporally or spatially related databases are presented in the thesis. Many of the presented methodologies can be used in both fields, including variable selection strategies.

New prediction strategies are presented, namely DirRec, Multiple-Outputs and Several Multiple-Outputs. Also new methodologies are presented, namely EOF Pruning and Ensemble of SOMs. The strategies are mainly used in time series prediction, but are also applicable to some extent for finding missing values. The new methodologies are for finding the missing values.

The above mentioned new strategies and methodologies have also

been compared with the state-of-the-art as well as combined to achieve even higher accuracy. There are two toolboxes, Lazy Learning toolbox and SOM+EOF toolbox, created in the course of the research.

1.3 PUBLICATIONS OF THE THESIS

List of publications included:

1. Antti Sorjamaa, Jin Hao, Nima Reyhani, Yongnan Ji, and Amaury Lendasse. Methodology for long-term prediction of time series. Neurocomputing, Volume 70, Issues 16-18, October 2007, Pages 2861-2869.

2. Antti Sorjamaa, Yoan Miche, Robert Weiss, and Amaury Lendasse. Long-term prediction of time series using NNE-based projection and OP-ELM. In IEEE World Conference on Computational Intelligence, Hong Kong, Research Publishing Services, Chennai, India, June 2008, Pages 2675-2681.

3. Antti Sorjamaa, and Amaury Lendasse. Time Series Prediction using DirRec Strategy. In Michel Verleysen, editor, Proceedings of European Symposium on Artificial Neural Networks - ESANN, d-side publications, Bruges, Belgium, April 26-28, 2006, Pages 143-148.

4. Souhaib Ben Taieb, Antti Sorjamaa, and Gianluca Bontempi, Multiple-Output Modelling for Multi-Step-Ahead Forecasting, Neurocomputing, Volume 73, Issues 10-12, June 2010, Pages 1950-1957.

5. Antti Sorjamaa, Amaury Lendasse, Yves Cornet, and Eric Deleersnijder. An improved methodology for filling missing values in spatiotemporal climate data set. Computational Geosciences, Volume 14, Issue 1, January 2010, Pages 55-64.

6. Antti Sorjamaa, Paul Merlin, Bertrand Maillet, and Amaury Lendasse. A nonlinear approach for the determination of missing values in temporal databases. European Journal of Economic and Social Systems, Volume 22, Issue 1, November 2009, Pages 99-117.

7. Antti Sorjamaa, Elia Liitiäinen, and Amaury Lendasse. Time series prediction as a problem of missing values: Application to ESTSP2007 and NN3 competition benchmarks. In IJCNN, International Joint Conference on Neural Networks, Documation

LLC, Eau Claire, Wisconsin, USA, August 12-17, 2007, Pages 1770-1775.

8. Paul Merlin, Antti Sorjamaa, Bertrand Maillet, Amaury Lendasse, X-SOM and L-SOM: a Double Classification Approach for Missing Value Imputation, Neurocomputing, Volume 73, Issues 7-9, March 2010, Pages 1103-1108.

9. Antti Sorjamaa, Francesco Corona, Yoan Miche, Paul Merlin, Bertrand Maillet, Eric Séverin, and Amaury Lendasse. Sparse linear combination of SOMs for data imputation: Application to financial database. In Risto Miikkulainen, Jose Principe, editors, Lecture Notes in Computer Science - Advances in Self-Organizing Maps - WSOM 2009, volume 5629/2009, Springer Berlin / Heidelberg, June 2009, Pages 290-297.

10. Antti Sorjamaa and Amaury Lendasse. Fast Missing Value Imputation using Ensemble of SOMs. Technical Report in Aalto University series of reports, June 2010.

In the text, these publications are referred using the numbering above, for example the last one by Sorjamaa and Lendasse is referred to as Publication 10.

1.4 CONTENTS OF THE PUBLICATIONS AND AUTHOR'S CONTRIBUTIONS

- **Publication 1.** In this first journal paper, the Direct strategy is introduced and compared against the prevailing state-of-the-art Recursive strategy. Several variable selection strategies and search criteria are compared using Poland Electricity dataset. In order to verify the superior accuracy of Direct strategy, a state-of-the-art nonlinear approximation method called Least Squares Support Vector Machine is used. The author has carried out part of the experiments and is responsible for the writing of the article.

- **Publication 2.** Two projection methodologies used together with new very fast development of MLP [55] is presented in long-term time series prediction context. The prediction scheme is applied to ESTSP 2007 Conference prediction competition dataset, where it reaches to the second position. The author is responsible for a part of the experimental design and computation and most of the writing of the article. The idea to combine

two projection methodologies was developed jointly by the author, Lendasse and Miche.

- **Publication 3.** A new time series prediction strategy, called DirRec, is presented. The DirRec strategy is a combination of Direct and Recursive strategies, hence the name. It is compared against the Direct and the Recursive strategies and the comparison shows the supremacy of the DirRec strategy. The author is responsible for the development of the strategy, experiments and writing of the article.

- **Publication 4.** Several Multiple-Output strategies are developed, introduced and compared with Direct and Recursive strategies. The comparison is performed using NN3 competition dataset, where the proposed methodology would obtain fourth place. The approximation methodology used is local constant model. The author suggested the idea of changing the number of outputs, varying between one output at a time and all at once. Ben Taieb was responsible for carrying out the experiments and the article was written together.

- **Publication 5.** A new algorithm based on Singular Value Decomposition is proposed, called EOF Pruning, for missing value imputation. The new algorithm is compared with the original EOF as well as Objective Analysis, which is considered to be the state-of-the art in the field of geosciences. The comparison is performed using a difficult surface temperature dataset from Tanganyika Lake, which has more than 60 percent of the values missing. The author came up with the improvement, has carried out all experiments and is responsible for the writing of the article.

- **Publication 6.** A combination methodology is introduced, the SOM + EOF. The accuracy of the combination is verified by using three datasets related to finance, more specifically the value of funds. The idea for combining the two methodologies is originally from the author, the experimental design and carrying out the experiments is joint work between the author and Merlin and the author is mainly responsible for the writing of the paper.

- **Publication 7.** An application of SOM + EOF is tested on a problem of time series prediction. The automatic selection of hyperparameters for both SOM and EOF simplify the prediction process. The presented combination achieved the sixth place in the original competition. The author is responsible for most

of the experimental design, carrying out the experiments and writing the article.

- **Publication 8.** A new methodology for missing value imputation relying on Self-Organizing Maps is presented. This methodology uses two different SOMs, which are trained on the original dataset and on a transposed one. The results show that the double imputation improves the accuracy. The author has assisted Merlin in the experimental setup, the original idea for using transposed data is from Merlin and the article was written jointly.

- **Publication 9.** The article presents the first try on combining several Self-organizing Maps using Hanna-Quinn Information Criterion and MultiResponse Sparse Regression by Tikka and Similä [65]. The results compared to the original SOM are promising using a financial dataset with missing values. The idea for combining several SOMs was jointly developed by the author and Lendasse. Lendasse was responsible for the experiments and the article was written jointly by the author, Lendasse and Miche.

- **Publication 10.** A new methodology, called Ensemble of SOMs, is presented. Two very different datasets are used to demonstrate the accuracy and speed of the Ensemble of SOMs. The methodology is compared against Probabilistic Principal Component Analysis. The author is partly responsible for the development of the combination, design of the experiments and conducting most of the experiments. The article was written by the author.

1.5 THESIS ORGANIZATION

This thesis is organized as follows: Chapter 2 is describing the problem areas dealt in the thesis, namely time series prediction and missing value imputation. Chapter 3 summarizes the time series prediction strategies used and developed. Chapter 4 collects all new and improved methodologies created in the course of the research for the prediction as well as for the missing value imputation. A quick overview of the two toolboxes created in the course of research is given in Section 5.

Finally, Chapter 6 summarizes the experimental results contained in the publications and Chapter 7 summarizes the findings of the thesis, presents the conclusions of the thesis and, finally, outlines the further work inspired by the presented research results.

2 PROBLEM AREAS

This is the description of problem areas, namely the division between time series prediction and missing value imputation. Both of the areas have their own specific problems, but are closely related. Similar methodologies can be applied to both problem areas.

Naturally, time series prediction aims at predicting the future by using the knowledge gathered from the past measurement values. This is achieved by using clever input variable selection, finding and validating the most suitable model and finally training the model by obtaining the optimal parameters and hyperparameters using the past data.

One can also think time series prediction as a problem of missing values (see Publication 7). Whereas traditionally missing values are located in the past, in time series prediction the missing values are the ones located in the unknown future.

Both areas also face the same complexity of selecting the output strategy, used methodology as well as variable selection scheme to be used. In the following, the more detailed particulars of both fields are described along with the current state-of-the-art.

2.1 TIME SERIES PREDICTION

Time series forecasting is a challenge in many fields. In finance, experts forecast stock exchange courses or stock market indices; data processing specialists forecast the flow of information on their networks; producers of electricity forecast the load of the following day. The common point to their problems is the following: how can one analyze and use the past to predict the future?

Many techniques exist for the approximation of the underlying process of a time series: Autoregressive Functions [49], Nonlinear Autoregressive Functions [41], Artificial Neural Networks [83], Self-Organizing Maps [6], Fuzzy Inference Systems [82] and Support Vector Machines [72], in order to mention a few. The last one in the previous list has been considered to be the state-of-the-art for several years, but there are several good alternatives to be considered [55]. The developed and improved methodologies are presented in more detail in Chapter 4.

In general, all these methods try to build a model of the underlying process, which created the series of observations. The created model is then used on the last known values of the series to predict the future values. The common difficulty to all the methods is the determination

of sufficient and necessary information for an accurate prediction.

A new challenge in the field of time series prediction is the Long-Term Prediction: several steps ahead have to be predicted. Long-Term Prediction has to face growing uncertainties arising from various sources, for instance, accumulation of errors and the lack of information [83].

When predicting further into the future, the growing uncertainties increase the necessity of reliable and accurate selection of prediction strategy. Is it accurate enough to predict using one model one time step after another and accept the accumulation of errors or is it better to use separate model for each prediction? If multiple models are needed, should one also select or project the variables separately for each model and for each time step?

Currently, the most used approach to deal with the challenges of long-term prediction is the Recursive prediction strategy [25, 41, 59]. Recursive strategy is also known as *rolling, iterative, one-step-ahead, recurrent* and *continuous* prediction strategy. When using Recursive strategy, one is actually repeating one-step-ahead prediction several times using the previous approximation as input. This kind of long-term prediction is obviously not yielding very good results after certain prediction steps, because of the increase in accumulated prediction error.

To overcome the limitations of the Recursive prediction strategy, is to use another existing possibility, the Direct strategy [46, 57, 75, 76]. This strategy is still much less used than the Recursive strategy, even though several papers have shown its superior performance. For instance, Publication 1 and [76] show comparisons of Recursive against Direct strategy.

The Direct strategy might be still relatively unused, because for each prediction step one needs to train separate model and optimize other learning parameters, which in turn increase the need for computational time. On the other hand, rapid increase in computer power and new fast techniques to parallelize computations make the Direct strategy more reasonable and appealing choice for long-term time series prediction.

In this thesis, three new prediction strategies are presented: DirRec as a combination of Direct and Recursive, Multiple-Outputs and Several Multiple-Outputs. More details about the more advanced prediction strategies developed in this thesis are presented in Chapter 3.1.

Another problem arises when selecting a methodology and the proper prediction strategy, is the selection of input variables to be used in the approximation. Different models need different amount of variables in order to be reliable and accurate and that reflects directly to

the problem of validating the used model properly and to the selection of inputs variables.

In the literature, the problem of having large amount of input variables is described as *curse of dimensionality* [11, 79, 34]. More variables, in theory, include more information of the underlying process and enable the approximator to perform with a higher accuracy. On the other hand, less variables, in practice, lead to more efficient usage of the variables, easier interpretability of the chosen set and smaller computational time.

Good selection of variables is heavily dependent on the chosen methodology and the variable selection scheme has to be carefully tested and validated, especially if the selection process is done automatically. The variable selection strategies along with a broader concept of variable projection strategy are presented in Sections 3.2 and 3.3. Section 3.4 presents several Search Criteria, which can be used to perform the variable selection or projection strategies.

The above mentioned numerous different selection problems create very high-dimensional optimization problem. Each of the selections needed to guarantee optimal results are not always straightforward, especially when computational time becomes an issue. Faster methodologies, which still retain the accuracy of the state-of-the-art methodologies, are not only very necessary, but even mandatory, when the sizes of the databases increase all the time.

2.2 MISSING VALUES

The presence of missing values in the underlying databases is a recurrent problem in many different fields. For example, in meteorology and climatology [74, 85], finance, process industry [62] and sosiology [3] have to deal with missing values.

A great number of methods have been already developed for solving the problem by filling the missing values, for example, Kriging [81] and several other Optimal Interpolation methods, such as Objective Analysis [35].

One of the emerging approaches for filling the missing values is the Empirical Orthogonal Functions (EOF) methodology [10, 21, 58]. The EOF is a deterministic methodology, enabling a linear projection to a high-dimensional space. Moreover, the EOF models allow continuous interpolation of missing values even when high percentage of the data is missing.

The EOF is closely related to the Principal Component Analysis (PCA), which was originally introduced by Pearson in [56]. The PCA is closely related to the EOF methodology and it is computed

in a very similar way. Furthermore, the PCA methodology has been lately improved by extending it using statistical formulation of the computation, called generally Probabilistic PCA (PPCA) [17], which can be used to fill the missing values in a database.

Another popular method for finding missing values, called Self-Organizing Maps (SOM) [43], aim to ideally group homogeneous individuals, highlighting a neighborhood structure between classes in a chosen lattice. The SOM algorithm is based on unsupervised learning principle where the training is entirely stochastic, data-driven. No information about the input data is required. Furthermore, the SOM algorithm allows projection of high-dimensional data to a low-dimensional grid. Through this projection and focusing on its property of topology preservation, SOM allows a non-linear interpolation for filling the missing values.

Recently, several new applications, modifications, improvements and combinations with other methods have been presented [7, 44, 52, 80]. The SOM is still an active part of machine learning community and continues to gather new inventions around it.

3 PREDICTION STRATEGIES

This chapter presents strategies for output approximation, variable selection and parameter tuning. The first section presents five different strategies for the output approximation, the second section reviews shortly four approaches to variable selection and the third section explains the projection scheme relevant to the thesis.

3.1 OUTPUT APPROXIMATION IN TIME SERIES PREDICTION

In order to approximate several outputs, one needs to have a strategy to do that. For a single output, like in normal regression task where the output of a sample to be modeled is a scalar value, the choice is easy. But when there are several outputs to be approximated, one needs a suitable strategy for it.

Each presented strategy has its own benefits and drawbacks, which are discussed more deeply with each strategy. The strategies are presented in time series prediction context, but they are applicable to any approximation task, where several output need to be approximated at the same time or separately.

From the five presented approaches, two first ones, Recursive and Direct strategies, can be considered as the state-of-the-art in Time Series Prediction field. The latter three approaches, DirRec, Multiple-Outputs and Several Multiple-Outputs, are recently developed ones.

3.1.1 Recursive

The most intuitive and simple method to approximate several outputs in time series prediction task, is the Recursive strategy. It is also known as *rolling, iterative, one-step-ahead, recurrent* and *continuous* strategy, since it uses the predicted values as known data to predict the next ones. In more detail, the strategy can be explained by first making one-step-ahead prediction:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-d+1}, \Theta), \tag{3.1}$$

where d denotes the number of previous values of the series used, f is the model used for the prediction and Θ denotes the set of parameters needed for the model f. It is possible to use also exogenous variables as inputs, but they are not considered here.

To predict the next value of the series, the same model and the same set of parameters is used:

$$\hat{y}_{t+2} = f(\hat{y}_{t+1}, y_t, y_{t-1}, \ldots, y_{t-d+2}, \Theta). \quad (3.2)$$

In Equation 3.2, the predicted value of $\hat{y}_{t+1}$ is used instead of the true value, which is unknown. Then, for the H-steps ahead prediction, $\hat{y}_{t+2}$ to $\hat{y}_{t+H}$ are predicted iteratively and each step the amount of predicted values used as inputs increases. When the prediction horizon is equal to the number of past values d in the model, the inputs are only approximations and no original observations are used at all.

The benefit of the Recursive Strategy is the simplicity. One needs to build only one model and obtain one set of parameters to predict the series as far as needed. If the model would be perfect, the accumulated prediction error would be zero and the prediction would be as accurate as it can be for infinite-steps-ahead as for one-step-ahead, assuming completely noiseless time series. In cases where the model is not perfect or there is noise present in the series, the accumulation of the errors deteriorates the approximation quality rather quickly, depending on the model accuracy and the amount of noise in the data.

3.1.2 Direct

The Direct strategy (see Publications 1 and 2) approximates each output individually with its own model and set of parameters. For example, for the H-steps ahead prediction, the model is

$$\hat{y}_{t+h} = f_h(y_t, y_{t-1}, \ldots, y_{t-d+1}, \Theta_h) \text{ with } 1 \leq h \leq H. \quad (3.3)$$

This strategy estimates H direct models between the inputs, which do not contain any predicted values, and the H outputs. Each model also has its own set of model parameters Θ_h.

In this strategy, the errors in the approximated outputs are not accumulated. Instead, only the single fold prediction error arising from the model itself is present. The performance of the previous or the next prediction steps have no influence on the performance of the current step.

However, when all the values, from $\hat{y}_{t+1}$ to $\hat{y}_{t+H}$, need to be predicted, H different models must be built. The Direct strategy increases the complexity of the prediction, but for most of the cases, it improves the performance compared to Recursive strategy. There are cases, when the performance is the same compared to Recursive, for example when the data is completely noiseless, but given that the models are properly built and validated the performance should never be worse.

3.1.3 DirRec

The DirRec strategy (see Publication 3) combines aspects from both, the DIRrect and the RECursive strategies. It uses a different model at every time step and introduces the approximations from previous steps into the input set. In the following example, four previous values of the time series are used as inputs. Then, the DirRec Strategy can be written as

$$\begin{aligned} \hat{y}_{t+1} &= f_1(y_t, y_{t-1}, \ldots, y_{t-d+1}, \Theta_1), \\ \hat{y}_{t+2} &= f_2(\hat{y}_{t+1}, y_t, y_{t-1}, \ldots, y_{t-d+1}, \Theta_2), \\ \hat{y}_{t+3} &= f_3(\hat{y}_{t+2}, \hat{y}_{t+1}, y_t, y_{t-1}, \ldots, y_{t-d+1}, \Theta_3), \\ &\vdots \end{aligned} \tag{3.4}$$

Every time step the input set in increased with one more input, the approximation of the previous step, like in Recursive strategy. But each step has its own model and model parameters, like in Direct strategy.

If no input selection is used, the complexity of the model increases linearly and more and more inputs with prediction error are fed into the model. Still, compared to the Recursive strategy, the original measurements are kept as inputs to the model and, if the model is accurate enough, the approximated values bring even more information to the prediction.

In the case where input selection is used, the selection procedure gives extra information of the accuracy of the predictions. If the previous output approximations are selected to be used as inputs for the next one, they are informative enough. If not, the accuracy of the model is not good enough in previous steps and that information could be used to finetune the model in the next step.

3.1.4 Multiple-Outputs

The three previous strategies, Recursive, Direct and DirRec, handled the outputs independently from each other. However, since the inputs are in many cases correlated, the outputs should have similar properties among themselves. In order to remove the conditional independence assumption, the notion of Multiple-Inputs Multiple-Outputs (MIMO) strategy for multi-step-ahead prediction is presented in [19].

The MIMO can be explained by the following formula:

$$(\hat{y}_{t+H}, \ldots, \hat{y}_{t+2}, \hat{y}_{t+1}) = f(y_t, y_{t-1}, \ldots, y_{t-M+1}, \Theta), \tag{3.5}$$

Note that in this case the returned prediction is not a scalar but a time series itself. In case of large horizon H, the number of terms could be enough to allow the use of specific operators of time series analysis, for instance autocorrelation or partial autocorrelation [73].

Naturally, the used model f have to support multiple-outputs in order to have difference between MIMO and Direct strategies. For instance, when using pure linear model, there is no difference between MIMO and Direct, but using some nonlinear method, like Local Linear Models (presented in Section 4.2), the strategies are indeed different and demonstrate different performance.

The MIMO strategy constrains all the horizons to be predicted with the same model structure and the same set of inputs. This constraint greatly reduces the flexibility and the variability compared to the single-output approaches and it could produce the negative effect of biasing the returned model. This happens especially in cases, where the output values have no covariance between themselves. In practice, when dealing with time series, the consecutive values almost always are correlated. Seasonal time series even have correlations that extend to several steps in the past and to the future, but might not have any influence on the steps in between.

On a more global note, many methods designed to find the missing values use the MIMO strategy. Especially for the cases, when there are significant percentage of the values missing, it is advisable to estimate them all at once.

3.1.5 Several Multiple-Outputs

Instead of forcing all H time steps to be predicted individually or altogether, consider an adoption of an intermediate approach. In this approach the constraint of the MIMO is relaxed by tuning an integer parameter s, which calibrates the dimensionality of the output on the basis of a validation criterion. This relaxed strategy is called Multiple-Inputs Several Multiple-Outputs (MISMO) (see Publication 4) and it was first introduced in [73] .

The aim of the MISMO strategy is to learn several multiple-output models from the data, not just a single one as in the MIMO strategy or one for each prediction horizon as in Direct strategy. The following equations describe the MISMO principle:

$$(\hat{y}_{t+ps}, \ldots, \hat{y}_{t+(p-1)s+2}, \hat{y}_{t+(p-1)s+1}) = f_p(y_t, y_{t-1}, \ldots, y_{t-M+1}, \Theta_p), \tag{3.6}$$

where

$$n = \frac{H}{s}, \quad p \in \{1, \ldots, n\} \tag{3.7}$$

In other words, s represents the number of consecutive outputs to be predicted at the same time by the same multiple-output model and p denotes the number of models, their parameters and sets of multiple-outputs.

The MISMO approach addresses the multi-step-ahead problem by taking into consideration two aspects: future predictions are expected to be mutually dependent, because of the stochastic properties of the series, but at the same time their degree of dependency is difficult to set a priori and typically not related to the horizon H fixed by the user.

For example, taking a Nonlinear Autoregressive (NAR) process of order $d = 2$ and that prediction of up to $H = 50$ steps is needed. In this case the MIMO strategy is extremely biased, because the MIMO tries to predict all 50 steps ahead at the same time and forces dependencies in the output with an order much bigger than 2.

But in the MISMO strategy, selecting s which is smaller than H mitigates the large difference between the order of the process and the needed number of prediction steps. Selecting $s = d$ retains the same complexity as in the original process, but probably the optimal choice is somewhere between d and H, due to the persistence of the covariances of an autoregressive process.

Furthermore, since MISMO strategy groups some prediction horizons together, it need less models to be built than the Direct strategy, while still keeping some dependencies between the output variables. The sets of s output variables are still independent from other sets, which enables a straightforward implementation of parallel computation to speed up the calculation process.

3.2 VARIABLE SELECTION

Variable selection is an essential preprocessing stage to guarantee high accuracy, efficiency and scalability [37] in problems such as machine learning, especially when the number of observations is relatively small compared to the number of inputs. It has been the subject in many application domains like pattern recognition [61], process identification [64], time series modeling [78] and econometrics [53]. Problems that occur due to poor selection of input variables are:

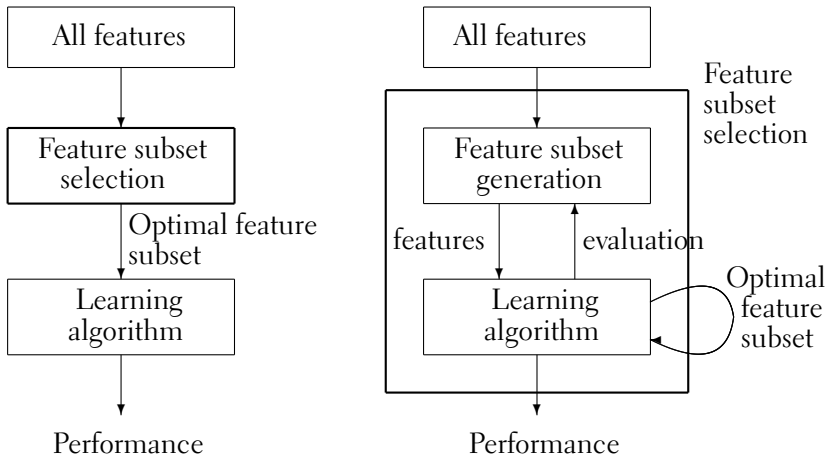
- Too large input dimensionality enforces the *curse of dimensionality* [78]

- Large dimensionality increases the computational complexity, need for computational resources and memory requirements of the learning model

- Unrelated inputs disturb the learning process and lead to poor models due to lack of generalization, whereas comparable performance could be achieved with proper selection of inputs

- Understanding complex models with high number of inputs is more difficult than simple models with less inputs

Usually, the input selection methods can be divided into two broad classes: *Filter* and *Wrapper* techniques, summarized in Figure 3.1.



(a) Filter method (b) Wrapper method

Figure 3.1: Two approaches of input variable subset selection

In Filter technique, the best subset of inputs is selected *a priori* based only on the dataset. The input subset is chosen by a selection criterion, which measures the relationship of each subset of input variables with the output. In the literature, plenty of filter measure methods of different natures [12] exist: distance metrics, dependence measures, scores based on the information theory and so on.

In the case of the Wrapper technique, the best input subset is selected according to the criterion, which is directly defined by the learning algorithm. The Wrappers search for a good subset of inputs using the learning model itself as a part of the evaluation function. This evaluation function is also employed in building the final learning model.

Comparing these two types of input selection techniques, the Wrapper techniques solve the real problem. But it is potentially very time consuming, as the final modeling algorithm has to be included in the cost function. Therefore, thousands of evaluations of the model are performed when searching for the best subset. But the benefit of the Wrapper technique is the guarantee that the found set of variables is proper and performing well with the chosen modeling algorithm.

On the other hand, the Filter technique can be made faster by using a fast algorithm and criterion. Especially when using computationally heavy methodology to model the data after input selection, it is beneficial to select the variables with the Filter technique. Naturally, the filter technique still requires a criterion for the estimation of the quality of the variable sets, but that can be selected to be very fast one. The downside of the filtering is the suitability of the selected set of variables for the modeling algorithm. Since potentially completely different criterion is used to select the best subset of variables, it is not guaranteed that the subset is the best set for the modeling algorithm.

There are also ways to combine the two approaches, see for example [25].

In the following subsections, four basic variable selection strategies are briefly explained and their benefits and pitfalls are discussed. All these strategies assume a predefined number of input variables to start the selection from. Even though one needs to decide the maximum amount of variables considered before starting the actual selection, it is straightforward to add more variables in the course of the selection process, but this is not considered here.

In practice, almost every variable selection strategy can be used as a Filter or a Wrapper. The choice is usually done by estimating the computational time of the final modeling scheme and determining whether it is still reasonable to use Wrapper technique or should one consider using the Filter technique with a fast search criterion.

There are no strict rules to select the starting dimensionality, but as a "rule of thumb" the starting set should contain at least one full season in case of seasonal datasets. In case of non-seasonal ones, the selection is even harder, but it can be done using some kind of heuristics. Naturally, one possibility is to try several choices and validate the best one using a validation methodology [42, 48, 68].

3.2.1 Exhaustive

The optimal algorithm is to compute the criterion with all the possible combinations of inputs, $2^d - 1$ combinations are tested, where d is the number of input variables. Then, the one that gives the best result according to the criterion is selected.

This strategy guarantees to find the optimal subset of variables, but when the amount of variables increases, the computational time increases exponentially, and finally prohibits the use of this strategy. For example, in the field of chemometrics [22] the normal input set is a spectrum, which has hundreds of variables. Computing all possible input subsets for 120 variables means computing the search criterion for $2^{120} - 1$ variable sets (roughly 10^{36}) and that is not feasible in practice.

3.2.2 Forward

Forward strategy belongs to the class of greedy algorithms. Essentially, the algorithm takes the best choice at a time, without ever turning back and goes on until the end, when the best variable set is selected among the evaluated ones.

In this strategy, the procedure is started from an empty set S of input variables and the best available variable is added to the set S one at a time, until the size of S is M. Assuming we have a set of input variables $X^i, i = 1, 2, \cdots, M$ and output Y, the algorithm is summarized in Figure 3.2.

1. (Initialization)
   Set F to be the initial set of original d inputs, and S to be the empty set which will contain the selected inputs.

2. (Forward)
   Find:

   $$X^s \quad = \quad \arg\max_{X^i}\{\textsc{Criterion}(\{S, X^i\}, Y)\}, \qquad X^i \in F.$$

   Save $\textsc{Criterion}(\{S, X^s\}, Y)$ and move X^s from F to S.
   Repeat until the size of S is d.

3. (Result)
   Compare the saved criterion values of all sizes of set S. Then, the selection result is the set S that maximizes the criterion.

Figure 3.2: Forward Selection Strategy.

Naturally, in in Figure 3.2, if an error criterion is used in the search, one must replace arg max with arg min.

In Forward selection, only $d(d+1)/2$ different input sets are evaluated. This is much less than the number of input sets evaluated with the Exhaustive search. Using the same instance than before, if we

have 120 input variables, the Forward strategy computes the criterion on $120(120+1)/2 = 7140$ subsets instead of 10^{36} with the Exhaustive strategy. This is many orders of magnitude smaller computational load and it enables the Forward strategy to use even larger input variable sets.

On the other hand, optimality is not guaranteed. The selected set may not be the global optimal one, but instead the search procedure might get stuck to a local minimum.

3.2.3 Backward

Backward selection strategy, also called Pruning [51], is the opposite of Forward selection process. It is also a greedy strategy and performs very similarly to the previously presented Forward strategy.

In this strategy, the starting set S is initialized to contain all input variables. Then, the input variable, removal of which maximizes the criterion, is removed from set S one at a time, until the size of S is 1.

Basically, Backward strategy is the same procedure as Forward selection presented in the previous section, but reversed. It evaluates the same amount of input sets as Forward selection, $d(d+1)/2$. Also, the same restriction exists, optimality is not guaranteed.

3.2.4 Forward-Backward

Both Forward and Backward selection strategies suffer from an incomplete search and Forward-Backward strategy tries to alleviate the the shortcoming by combining the two selection strategies. It offers the flexibility to reconsider input variables previously discarded and *vice versa*, to discard input variables previously selected. Even though it is still considered as a greedy strategy, it is more flexible than the Forward or the Backward alone.

It can start from any initial input set, including empty, full or randomly initialized input set. Given a set of variables X^i, $i = 1, 2, \cdots, d$ and an output Y, the Forward-Backward strategy is summarized in Figure 3.3.

Naturally, in Figure 3.3, if an error criterion is used in the search, one must replace arg max with arg min and take the lowest values instead of the highest.

It has to be noted that the selection result depends on the initialization of the input set and the nature of the problem. It is not guaranteed that the selection will yield the optimal set of variables and it is still possible to get stuck in a local minimum. However, the combination of the Forward and Backward strategies reduces the risk of suboptimal solutions, which can be even more enforced by redoing the Forward-

1. (Initialization)
   Let set S to be the selected variables set, which can contain any input variables, and set F to be the unselected variables set containing the inputs, which are not in set S. Compute $\text{CRITERION}(S, Y)$.

2. (Forward-Backward)
   Find:

$$X^{s1} = \arg\max_{X^i} \{\text{CRITERION}(\{S, X^i\}, Y)\}, \qquad X^i \in F$$
$$X^{s2} = \arg\max_{X^j} \{\text{CRITERION}(S \backslash X^j, Y)\}, \qquad X^j \in S$$

   If $\text{CRITERION}(S, Y) > \text{CRITERION}(\{S, X^{s1}\}, Y)\}$
   or $\text{CRITERION}(S, Y) > \text{CRITERION}(\{S \backslash X^{s2}\}, Y)\}$,
   go to step 3.
   Otherwise, take the action with the higher Criterion value, update set S and compute $\text{CRITERION}(S, Y)$.
   Repeat step 2.

3. (Result)
   The selection result is set S.

Figure 3.3: Forward-Backward Selection Strategy.

backward strategy several times from different initial starting points. For example, starting from empty set and full set. It is also possible to initialize the search with random selection of input variables.

3.2.5 Others

There are several other strategies available for variable selection. These include ranking strategies [54, 55], different modifications of the Forward-backward [69] and information theoretical approaches [1, 63].

In order to have an optimal selection of variables, several different approaches and criteria should be tested and the applicability verified by the final approximation model. One approach does not guarantee optimal solution for all models and some models are more sensitive to the proper selection of variables.

3.3 VARIABLE PROJECTION

3.3.1 Concept

Another way of decreasing the dimensionality of the input variable set is to use variable projection. In variable projection, the original high-dimensional input space is replaced by lower-dimensional latent space, where the goal is to contain the original information in a smaller space. In many cases, the projection is used as a preprocessing step before the final approximator is utilized.

There are many different ways to project the input variable space into lower dimensional one, for example Partial Least-Squares (PLS), Principal Components Regression (PCR) and Independent Component Analysis (ICA).

When performing regression, classification or other function approximation task, linear projection methods, such as PLS (see Publication 2) and PCR, are standard approaches based on the idea of combining the original variables by projection. The methods project the original input variables onto a latent space with reduced dimensionality. In PCR, the projection is constructed using PCA and the input variables are selected to keep the maximum amount of information. In PLS new inputs are built that are maximize the suitability for approximating the output [84].

But there are only few projection *strategies*, which allow the use of arbitrary search criteria. This section presents one of them, called Extended Forward-Backward (EFB). Another example of such strategy is Genetic Algorithm (GA) [36], but that is not explained here (see for example [70]).

3.3.2 Extended Forward-Backward

This projection strategy is derived from the similarly named variable selection strategy, the Forward-Backward strategy 3.2.4 (see Publication 2). The goal is to decrease the dimensionality of the input variable set, but it is not mandatory to set the final dimensionality of the projected space beforehand. Also, any search criteria can be used to tune the projection parameters.

For N input-output pairs, $(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, a new set of inputs $\mathbf{z}$ is given as:

$$\mathbf{z} = \mathbf{x}\mathbf{P}, \tag{3.8}$$

where $\mathbf{P}$ is the projection matrix with $d \times p$ elements, where p is the projection dimension and d is the dimensionality where the projection is started from, the dimension of the input variables. As in the variable

selection strategy when dealing with time series, the dimensionality d has to fixed before starting the procedure using some heuristics or one can try out several different choices and select the one with the best performance.

The elements of the projection matrix can be any real numbers, but in practice the range of the values is bounded to interval from -1 to 1, given that the input variables are normalized.

Before the EFB can be applied, a degree of discretization h has to be decided. It means, that for each value, the interval has to be evenly divided into parts, which can be selected by the EFB. For example, if we divide the interval from -1 to 1 evenly and take $h = 0.1$, it gives us 21 different possibilities for each element in the projection matrix.

After that, the procedure is similar to the FB selection strategy. Starting from a projection into one-dimensional space, all possibilities for one element at a time are tested and the best one is selected. This is repeated for all elements as long as there is no more improvement obtained according to the search criteria used. When the projection matrix has been optimized, next dimension is added and the process is repeated. The earlier, already optimized projection dimensions, are kept fixed. The procedure is summarized in Figure 3.4.

1. Initialize the first column of $\mathbf{P}$
2. Optimize the first column of $\mathbf{P}$ by EFB using the selected search criteria in the projected space
3. Initialize the next column of $\mathbf{P}$
4. Optimize the intialized column of $\mathbf{P}$ by EFB with the previous columns unchanged
5. Repeat from step 3

Figure 3.4: Extended Forward-Backward projection strategy.

The process can be used to visualize the dataset by setting the projection dimension to two and optimizing the projection. But for an accurate input preprocessing, larger projection dimensions should be used.

In order to have a meaningful projection matrix, the resulting P has to be full rank. This is a simple constraint to fulfill, since the search process can be stopped, if the resulting matrix is not full rank, and the last projection dimension can be removed. Otherwise, there are no clear limitations to the projection dimension p. One could

argue, that the projection to as many or more dimensions as the original dataset does not make much sense when we are speaking of dimensionality reduction. Hence, an intuitive limit for maximum projection dimension is $d - 1$.

As the selection strategy, this projection strategy is not guaranteeing to find the optimal projection. Instead, it can be stuck into local minimum without predefined computational time. As an improvements, the number of steps performed can be limited to control the calculation time and the projection can be started from several different starting points to limit the problem of local minima. Also, the search criteria should be fast and easily computable, since it has to computed vast amount of times.

3.4 SEARCH CRITERIA

This section presents a few Search Criteria used in the methodologies and strategies in the thesis. There are several other criteria available, for example k-NN methodology, Section 4.3, can be used as a search criteria.

3.4.1 Mutual Information

The Mutual Information (MI) can be used to evaluate the dependencies between random variables. The MI between two variables, X and Y, is the amount of information obtained from X in the presence of Y and *vice versa*. In time series prediction problem, if Y is the output and X is a subset of the input variables, the MI between X and Y is one criterion for measuring the dependence between the inputs and the output. Thus, the inputs subset X^i, which gives maximum MI, is chosen to predict the output Y.

The definition of MI originates from the entropy in the information theory. For continuous random variables (scalar or vector), let $\mu^{X,Y}$, μ^X and μ^Y represent the joint probability density function and the two marginal density functions of the variables. The entropy of X is defined by Shannon [8] as:

$$H(X) = -\int_{-\infty}^{\infty} \mu^X(x) \log \mu^X(x) dx, \tag{3.9}$$

where log is the natural logarithm and then, the information is measured in natural units.

The remaining uncertainty of X is measured by the conditional entropy as

$$H(X|Y) = -\int_{-\infty}^{\infty} \mu^Y(y) \int_{-\infty}^{\infty} \mu^X(x|Y=y) \log \mu^X(x|Y=y) dxdy. \tag{3.10}$$

The joint entropy is defined as

$$H(X,Y) = -\int_{-\infty}^{\infty} \mu^{X,Y}(x,y) \log \mu^{X,Y}(x,y) dxdy. \tag{3.11}$$

The MI between variables X and Y is defined as [24]:

$$\mathrm{MI}(X,Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y). \tag{3.12}$$

From Equation 3.9, 3.10, 3.11 and 3.12, MI is computed as:

$$\mathrm{MI}(X,Y) = \int_{-\infty}^{\infty} \mu^{X,Y}(x,y) \log \frac{\mu^{X,Y}(x,y)}{\mu^X(x)\mu^Y(y)} dxdy. \tag{3.13}$$

For computing the MI, only the estimations of the probability density functions $\mu^{X,Y}$, μ^X and μ^Y are required.

$\mathrm{MI}(X,Y)$ is estimated by a k-Nearest Neighbors approach presented in [45]. In order to distinguish the number of neighbors that used in the MI and the one used in the k-NN, the number of neighbors is denoted by l for the estimation of MI.

The novelty of this l-NN based MI estimator consists in its ability to estimate the MI between two variables of any dimensional space. Then, the estimation of MI depends on the predefined value l.

In [45], it is suggested to use a mid-range value $l = 6$. But it has been shown (see Publication 1) that when applied to time series prediction problems, l needs to be tuned for different datasets and different data dimensions in order to obtain better performance.

3.4.2 Nonparametric Noise Estimator using Gamma Test

Gamma Test (GT) is a Nonparametric Noise Estimator (NNE) for estimating the variance of the noise, or the mean square error (MSE), that can be achieved without overfitting [40]. The evaluation of the NNE is done using the GT estimation introduced by Stefansson in [71].

Given N input-output pairs: $(\mathbf{x}_i, y_i) \in \mathbb{R}^M \times \mathbb{R}$, the relationship between $\mathbf{x}_i$ and y_i can be expressed as:

$$y_i = f(\mathbf{x}_i) + r_i, \tag{3.14}$$

where f is the unknown function and r is the noise. The Gamma Test estimates the variance of the noise r.

The GT is useful for evaluating the nonlinear correlation between two random variables, namely, input and output pairs. The GT has been introduced for model selection but also for input selection: the set of inputs that minimizes the GT is the one that is selected. Indeed, according to the GT, the selected set of inputs is the one that represents the relationship between inputs and output in the most deterministic way.

GT is based on hypotheses coming from the continuity of the regression function. If two points $\mathbf{x}$ and $\mathbf{x}'$ are close in the input space, the continuity of regression function implies the outputs $f(\mathbf{x})$ and $f(\mathbf{x}')$ will be close enough in the output space. Alternatively, if the corresponding output values are not close in the output space, this is due to the influence of the noise. The average distance between neighboring samples in the input space is denoted as σ and the average distance between corresponding outputs as γ.

Two versions for evaluating the GT are suggested. The first one evaluates the values of γ, σ in increasing sized sets of data. Then the result for a particular parameter pair is obtained by averaging the results from all set sizes. The new or refined version establishes the estimation based on the k-Nearest Neighbors differences instead of increasing the number of data points gradually. In order to distinguish the k used in the NNE context from the conventional k in k-NN, the number of nearest neighbors is denoted by p.

Let us denote the p^{th} nearest neighbor of the point $\mathbf{x}_i$ in the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ by $\mathbf{x}_{p(i)}$. Then the following variables, γ_N and σ_N are defined as:

$$\gamma_N(p) = \frac{1}{2N} \sum_{i=1}^{N} \left| y_{p(i)} - y_i \right|^2, \tag{3.15}$$

$$\sigma_N(p) = \frac{1}{2N} \sum_{i=1}^{N} \left| \mathbf{x}_{p(i)} - \mathbf{x}_i \right|^2 \tag{3.16}$$

where $|.|$ denotes the Euclidean metric and $y_{p(i)}$ is the output of $\mathbf{x}_{p(i)}$. For correctly selected p [40], the constant term of the linear regression model between the pairs $(\gamma_N(p), \sigma_N(p))$ determines the noise variance estimate. For the proof of the convergence of the Gamma Test, see [40].

The GT assumes the existence of the first and second derivatives of the regression function. Let us denote

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x(i)}\right)_{i=1}^{d}, \mathrm{H}f(\mathbf{x}) = \left(\frac{\partial^2 f}{\partial x(i)\partial x(j)}\right)_{i,j=1}^{d}, \quad (3.17)$$

where $x(i)$ and $x(j)$ are the i^{th} and j^{th} components of $\mathbf{x}$ respectively. d is the number of variables. The GT requires that both $|\mathrm{H}f(\mathbf{x})|$ and $|\nabla f(\mathbf{x})|$ are bounded.

These two conditions are general and are usually satisfied in practical problems. The GT requires no other assumption on the smoothness property of the regression function. Consequently, the method is able to deal with the regression functions of any degree of roughness.

The second assumption is about the noise distribution:

$$\mathrm{E}_\Phi\{r\} = 0 \quad \text{and } \mathrm{E}_\Phi\{r^2\} = \mathrm{var}\{\varepsilon\} < \infty \quad (3.18)$$

$$\mathrm{E}_\Phi\{r^3\} < \infty \quad \text{and } \mathrm{E}_\Phi\{r^4\} < \infty, \quad (3.19)$$

where $\mathrm{E}_\Phi\{r\}$ is the noise density function. Furthermore, it is required that the noisy variable should be independent and identically distributed. In the case of heterogeneous noise, the GT provides the average of noise variance extracted from the whole dataset.

As discussed above (see Equation 3.15), the GT depends on the number of p used to evaluate the regression. It is suggested to use a mid-range value $p = 10$ [40]. But, when applied to time series prediction problems, p needs to be tuned for each dataset and for each set of variables to obtain better performance (see Publication 1).

3.4.3 MultiResponse Sparse Regression

Multiresponse Sparse Regression, proposed by Timo Similä and Jarkko Tikka in [65], is an extension of the Least Angle Regression (LARS) algorithm [27] and hence, it is actually a variable ranking technique, rather than a selection one. After the ranking, another method to actually select the final number of ranked variables is needed. This can performed by any validation method or, for instance, Hanna-Quinn Information Criterion, presented in Section 3.4.4.

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

More specific details of the MRSR algorithm can be found from the original paper [65].

An important detail shared by the MRSR and the LARS is that the ranking obtained is exact, if the problem is linear.

3.4.4 Hannan-Quinn Information Criterion

There are many possible criteria for complexity estimation used in machine learning. Typical examples are Akaike's information criterion (AIC) [1] or the Bayesian Information Criterion (BIC) [63]. Their expressions are usually based on the residual sum of squares (Res) of the considered model (first term of the criterion) plus a penalty term (second term of the criterion). Differences between criteria mostly occur on this penalty term. AIC, Equation 3.20, penalizes only by the number of parameters p of the model, so that not too many free parameters are used to obtain a good fit by the model. On the other hand, BIC, Equation 3.21, takes into account also the number of samples N used for the model training.

$$AIC = N \times \log\left(\frac{Res}{N}\right) + 2 \times p. \tag{3.20}$$

$$BIC = N \times \log\left(\frac{Res}{N}\right) + p \times \log N, \tag{3.21}$$

The AIC is known to have consistency problems: while minimizing the AIC, it is not guaranteed that the complexity selection will converge toward an optima if the number of samples goes to infinity [13]. The main idea raised by this observation is about trying to balance the underfitting and the overfitting when using such criteria. This is achieved through the penalty term, for example, by having a $\log N$ based term in the penalty (where N is the number of samples), which the BIC has.

The Hannan-Quinn Information Criterion (HQ) [38] is defined as

$$HQ = N \times \log\left(\frac{Res}{N}\right) + 2 \times p \times \log\log N. \tag{3.22}$$

The HQ is very close to the other two presented criteria, as can be seen comparing the expressions of the AIC and BIC, Equations 3.20 and 3.21, with the definition of HQ, Equation 3.22.

The idea behind the design of the HQ criterion is to provide a consistent criterion (regarding for example AIC which is not consistent in its standard definition) in which the second term (the penalty) $2 \times p \times \log\log N$ grows but at a very slow rate, regarding the number

of samples. Therefore, the HQ can be considered as a more consistent compromise between the AIC and the BIC.

Examples of the HQ criterion usage can be found from Publication 7 and from [55].

4 METHODOLOGIES FOR PREDICTION AND IMPUTATION

This chapter describes the used and developed methodologies. The methodologies are loosely arranged starting from linear models and moving on to models including more and more nonlinear components. First one is the standard linear model, then local linear models and so on until the SOM [43] and its modifications and combinations are presented.

The following table summarizes the presented methodologies.

Table 4.1: Summary of the presented methodologies. TSP and MV denote that the methodology belongs mainly to time series prediction or missing values context, respectively.

	TSP	MV
Linear Models	x	
Local Linear Models	x	
k-Nearest Neighbors	x	x
Empirical Orthogonal Functions		x
EOF Pruning		x
Self-Organizing Maps	x	x
Ensemble of SOMs		x
SOM + EOF	x	x

All the above methodologies are used in the publications collected into this thesis and they are also presented in the following sections. This thesis proposes as original contribution three methodologies, namely EOF Pruning, Ensemble of SOMs and SOM+EOF, which are presented in this chapter. Other methodologies in the table are considered to be state-of-the-art.

4.1 LINEAR MODELS

Linear models such as ARX and ARMA [49] are the most basic models. They are very fast to compute, but unless the problem is almost completely linear, the results are poor. The approximation of the output $\hat{y}$ is obtained by multiplying each input variable x_i with a variable specific constant weight α_i

$$\hat{y} = \sum_{i=1}^{d} \alpha_i x_i = \mathbf{x}\boldsymbol{\alpha}, \tag{4.1}$$

where d is the number of variables in the input sample. The weights can be obtained by Ordinary Least Squares (OLS) method [60]. Denoting all input variables and samples by a matrix $\mathbf{X}$, all output variables by $\mathbf{Y}$ and the weights by α, the OLS solution is obtained as

$$\alpha = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}. \tag{4.2}$$

4.2 LOCAL LINEAR MODELS

Forecasting methods rely on learning procedures to estimate the temporal stochastic dependencies from data. The use of local learning approaches in forecasting literature dates back to the seminal work of Lorenz [50] on chaotic series. Other classical references are [31, 32, 39].

Considering a local learning approach where the problem of adjusting the size of the neighborhood is solved by a Lazy Learning (LL) algorithm [20]. This algorithm selects on a query-by-query basis and by means of a local cross-validation scheme the best number of neighbors to be used in the linear model.

The PREdiction Sum of Squares (PRESS) [2] gives the tools to calculate the Leave-one-out Cross-Validation errors for linear models without excessive computation. When we add this mathematical derivation to the standard recursive least squares algorithm, introduced by Biermann [14], we have a nice and fast way to validate and select the LL structures.

The computation of the output of the LL model is identical to the previously presented global linear model, Equation 4.1. The difference is that not all samples are used to compute the parameters of the model α, but instead only few nearest neighbors of the query point.

The optimization of the number of neighbors is crucial. When the number of neighbors is small, local linearity assumption is valid. On the contrary, if the number of neighbors is large, the local linearity assumption is not valid anymore and the linear model fails to provide good approximations. Of course the number of neighbors have to be at least as large as the number of local linear model parameters to be estimated. However, it may vary if local input variable selection is also applied to the Lazy Learning model.

The main advantages of the LL models are the simplicity of the model itself and the low computational load. Furthermore, the local model can be built only around the sample for which the approximation is requested. This gives the name to the Lazy Learning: there's no need to do anything before the approximation is requested.

It is also possible to use a *global neighborhood*, where the number of neighbors is the same for all the data points. In order to use the LL with the global neighborhood, the global size of the neighborhood must be determined beforehand. Each neighborhood size for each sample is evaluated and the size minimizing globally the estimation of the generalization error is selected.

If the density of the training samples is fairly constant, the use of global neighborhood really speeds up the calculation process and makes the approximations more accurate. If the density varies, it is wiser to use local neighborhood to get good approximations. More experimental results and discussion on the Lazy Learning paradigm can be found from [67] and Publication 4. There is also a toolbox created for Lazy Learning modeling. More information can be found from Section 5.1.

4.3 K-NEAREST NEIGHBORS

The k-Nearest Neighbors (k-NN) approximation method is a very simple and powerful method. It has been used in many different applications, particularly for classification tasks [16]. The key idea behind the k-NN is that samples with similar inputs have similar output values. Nearest neighbors are selected, according to Euclidean distance, and their corresponding output values are used to obtain the approximation of the desired output. The estimation of the output can be calculated simply by averaging the outputs of the nearest neighbors

$$\hat{y}_i = \frac{\sum_{j=1}^{k} y_{j(i)}}{k}, \tag{4.3}$$

where $\hat{y}_i$ represents the estimate (approximation) of the output, $y_{j(i)}$ is the output of the j^{th} nearest neighbor of sample x_i and k denotes the number of neighbors used. Naturally, it is possible to weight closer neighbors to be more influential than the further away neighbors, but that is not considered here.

The distances between samples are influenced by the input selection. Then, the nearest neighbors and the approximation of the outputs depend on the input selection.

The k-NN is a nonparametric method and only k, the number of neighbors, has to be determined. The selection of k can be performed by many different model structure selection techniques, for example k-fold Cross-Validation [42], Leave-one-out [42], Bootstrap [29] and Bootstrap 632 [28]. These methods estimate the generalization error obtained for each value of k. The selected k is the one that minimizes the generalization error.

In [68] all methods, the Leave-one-out and Bootstraps, select the same input sets. Moreover, the number of neighbors is more efficiently selected by the Bootstraps [68].

Furthermore, the k-NN can be used with locally adaptable number of neighbors. This can be achieved by using the Leave-One-out (LOO) error in the neighborhood to select the optimal number of neighbors for each query point separately. Query point in this case is each sample for which the approximation is needed.

A computationally efficient way to perform LOO cross-validation and to assess the generalization performance of local linear models is the PRESS statistic, proposed in 1974 by Allen [2]. By assessing the performance of each local model, alternative configurations can be tested and compared in order to select the best one in terms of expected prediction performance.

The formulation for constant model from the version for the linear model is straightforward. The idea consists in associating a LOO error $e_{LOO}(k)$ to the estimation

$$\hat{y}_k = \frac{1}{k}\sum_{j=1}^{k} y_{[j]} \tag{4.4}$$

returned by k neighbors. In case of a constant model, the LOO term can be derived as follows [18]:

$$e_{LOO}(k) = \frac{1}{k}\sum_{j=1}^{k}(e_j(k))^2, \tag{4.5}$$

where

$$e_j(k) = y_{[j]} - \frac{\sum_{i=1(i\neq j)}^{k} y_{[i]}}{k-1} = k\frac{y_{[j]} - \hat{y}_k}{k-1}. \tag{4.6}$$

The best number of neighbors is then defined as the number

$$k^* = \arg\min_{k\in\{2,\ldots,K\}} e_{LOO}(k), \tag{4.7}$$

which minimizes the LOO error.

Finally, due to the speed and simplicity of the k-NN, it can be used as a search criteria for variable selection and projection. The usage of the method remains very much the same and both, the local or global selection of k, can be utilized.

4.4 EMPIRICAL ORTHOGONAL FUNCTIONS

Empirical Orthogonal Functions (EOF) [58] is a deterministic method allowing a linear, continuous projection to a high-dimensional space. The EOF has been used in climate research for finding the missing values as well as a denoising tool [5, 4, 9, 10, 21].

Here, the EOF is used as a denoising tool and for finding the missing values at the same time. The method presented here is based on the one presented in [10].

The EOF is calculated using the standard and well-known Singular Value Decomposition (SVD),

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^* = \sum_{k=1}^{K} \rho_k \mathbf{u}_k \mathbf{v}_k, \tag{4.8}$$

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

When the EOF is used to remove the noise from the data, not all singular values and vectors are used to reconstruct the data matrix. Instead, it is assumed that the vectors corresponding to larger singular values have larger signal-to-noise ratio than the ones corresponding to smaller values [58]. Therefore, it is logical to select the q largest singular values and the corresponding vectors and reconstruct the data matrix using only them.

When $q < K$, the reconstructed data matrix is obviously not the same than the original one. The larger q is selected, the more original data, which also includes more noise, is preserved. The optimal q is selected using validation methods; see for example [47, 48].

The EOF (or the SVD) cannot be directly used with databases including missing values. The missing values must be replaced by some initial values in order to use the EOF. This replacement can be for example the mean value of the whole data matrix $\mathbf{X}$, the row or column mean, linear regression or polynomial fitting row wise or column wise, depending on the structure of the data matrix.

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

structed again. The procedure is repeated until the convergence criterion is fulfilled. The procedure is summarized in Table 4.2.

Table 4.2: Summary of the EOF algorithm for finding the missing values.

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

4.5 EOF PRUNING

In some cases, some of the biggest singular values contain so large noise levels that they disturb the selection process described in Table 4.2. For example, if the first n singular values are selected by the validation procedure, but not the $n + 1$, it does not necessarily mean that all the rest from $n + 2$ to K are as noisy as the $n + 1$. Some of the smaller values can still hold some important information vital to the accurate estimation of the missing values (see Publication 5).

Even the assumption of larger singular values holding more signal than noise is still valid; it does not mean that **all** smaller values are completely corrupted with noise. As described above, some smaller values can hold vital information even the amount of noise is increasing compared to the larger singular values.

If the purpose is to solely remove the noise from the dataset with the

cost of accuracy, then the smaller values should not be used. But our goal here is to approximate the missing values as accurately as possible and the denoising is left as a secondary goal.

Therefore, instead of selecting a certain number of largest singular values and vectors to perform the reconstruction, we propose an alternative approach; selecting the values and vectors in a non-continuous fashion.

The selection can be done according to any of the schemes presented in the Section 3.2. In our application, the selection is done by the Forward strategy, explained in Section 3.2.2.

The selection of singular values and vectors is done in each round of the EOF procedure. It means that when the initialization of the missing values is done and the singular values and vectors are calculated, the selection algorithm is used to select the most optimal values and vectors. Then, the initialized missing values are replaced by the reconstruction obtained using the selected set of singular values and vectors. In the next round, the new data matrix is again broken down to singular values and vectors and the selection is performed again.

The revised EOF Pruning algorithm is summarized in Table 4.3.

We have developed a toolbox for the Matlab software to perform the EOF Pruning for the missing value imputation. For more information see Section 5.2.

4.6 SOM

4.6.1 Traditional

The SOM algorithm is based on an unsupervised learning principle, where training is entirely data-driven and no information about the input data is required [43]. Here we use a 2-dimensional network, composed of c units (or code vectors) shaped as a square *lattice*. Each unit of a network has as many weights as the dimension d of the learning data samples, $\mathbf{x}_n,\ n = 1, 2, ..., N$. All units of a network can be collected to a weight matrix $\mathbf{m}(t) = [\mathbf{m}_1(t), \mathbf{m}_2(t), ..., \mathbf{m}_c(t)]$ where $\mathbf{m}_i(t)$ is the d-dimensional weight vector of the unit i at time t and t represents the steps of the learning process. Each unit is connected to its neighboring units through a neighborhood function $\lambda(\mathbf{m}_i, \mathbf{m}_j, t)$, which defines the shape and the size of the neighborhood at time t. The neighborhood can be constant through the entire learning process or it can change in the course of learning.

The learning starts by initializing the network node weights randomly. Then, for a randomly selected sample $\mathbf{x}_{t+1}$, we calculate the

Table 4.3: The EOF Pruning algorithm for finding the missing values.

1. Initial values are substituted into missing values of the original data matrix $\mathbf{X}$
2. Loop until convergence
   (a) K singular values and eigenvectors are calculated using the SVD
   (b) The selection process selects an optimal set of singular values and vectors from the K candidates. The selected set, q_r, is saved, where r represents the number of the current round.
   (c) The values and vectors in the set q_r are used to make the reconstruction
   (d) The missing values from the original data are filled with the values from the reconstruction
   (e) The validation error is calculated and saved. If the convergence criterion is not fulfilled, we continue to the next round from step (a).
3. The selected singular values and vectors in each round are used to reconstruct the final filling of the missing values in $\mathbf{X}$. The final filling uses as many rounds as determined by the validation error. In each round the corresponding set q_r is used.

Best Matching Unit (BMU), which is the neuron whose weights are closest to the sample. The BMU calculation is defined as

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \{\|\mathbf{x}_{t+1} - \mathbf{m}_i(t)\|\}, \tag{4.9}$$

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

If the randomly selected sample includes missing values, the BMU cannot be solved outright. Instead, an adapted SOM algorithm, proposed by Cottrell and Letrémy [23], is used. The randomly drawn sample $\mathbf{x}_{t+1}$ having missing value(s) is split into two subsets $\mathbf{x}_{t+1}^T = NM_{\mathbf{x}_{t+1}} \cup M_{\mathbf{x}_{t+1}}$, where $NM_{\mathbf{x}_{t+1}}$ is the subset where the values of $\mathbf{x}_{t+1}$ are not missing and $M_{\mathbf{x}_{t+1}}$ is the subset, where the values of $\mathbf{x}_{t+1}$ are missing. We define a norm on the subset $NM_{\mathbf{x}_{t+1}}$ as

$$\|\mathbf{x}_{t+1} - \mathbf{m}_i(t)\|_{NM_{\mathbf{x}_{t+1}}} = \sum_{j \in NM_{\mathbf{x}_{t+1}}} (\mathbf{x}_{t+1,j} - \mathbf{m}_{i,j}(t))^2, \quad (4.10)$$

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

Then the BMU is calculated with

$$\mathbf{m}_{BMU(\mathbf{x}_{t+1})} = \arg \min_{\mathbf{m}_i, i \in I} \left\{ \|\mathbf{x}_{t+1} - \mathbf{m}_i(t)\|_{NM_{\mathbf{x}_{t+1}}} \right\}. \quad (4.11)$$

When the BMU is found the network weights are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) - \varepsilon(t)\lambda\left(\mathbf{m}_{BMU(\mathbf{x}_{t+1})}, \mathbf{m}_i, t\right)[\mathbf{m}_i(t) - \mathbf{x}_{t+1}], \quad \forall i \in I, \quad (4.12)$$

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

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

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

$$\pi_{(M_{\mathbf{x}})}(\mathbf{x}) = \pi_{(M_{\mathbf{x}})}\left(\mathbf{m}_{BMU(\mathbf{x})}\right), \quad (4.13)$$

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

The procedure is summarized in Table 4.4. There is a toolbox available for performing the SOM algorithm [26].

Table 4.4: Summary of the SOM algorithm for finding the missing values.

1. SOM node weights are initialized randomly
2. SOM learning process begins
   (a) Input $\mathbf{x}$ is drawn from the learning data set $\mathbf{X}$
      i. If $\mathbf{x}$ does not contain missing values, BMU is found according to Equation 4.9
      ii. If $\mathbf{x}$ contains missing values, BMU is found according to Equation 4.11
   (b) Neuron weights are updated according to Equation 4.13
3. Once the learning process is done, for each observation containing missing values, the weights of the BMU of the observation are substituted for the missing values

4.6.2 Two-Space-SOM

One can mention two main drawbacks arising from the traditional SOM filling procedure. First, the rebuilding process is discrete, missing values of the sample are filled by the corresponding values of the neurons to which the sample is closest to. Thus, for all samples closest to the same node the estimations are the same.

Secondly, when dealing with spatio-temporal databases, where there exists dependencies in two directions, using the data only in one direction leaves some information unused. For example, a financial database consists of separate funds from the same source with their values taken from the same time instances. In this case, there clearly are relationships between different funds at the same time instance as well as between different time instances of the same fund. Hence, one can think the samples as columns or rows in the data matrix. But in both cases, the normalization of the data must be performed, in correct direction, in order to have unbiased training of the SOMs.

Here, two different SOMs are trained in two different input spaces (see Publication 8). Please note that in order not to confuse the approach presented here with the earlier version of DoubleSOM [66], the presented approach is called Two-Space-SOM.

As previously seen in [33], the first network; identified by its code vector weights $\mathbf{m}^1$ (each unit corresponding to a d-dimensional weight vector), uses training samples in the original input space. Then,

for each time series $\mathbf{x}_i$ containing missing values, the weights of the associated BMU are substituted for any missing values as traditional SOM

$$\mathbf{x}_{i,k} = \mathbf{m}_{BMU_{\mathbf{x}_i},k}, \tag{4.14}$$

for $k \in M_{\mathbf{x}}$.

Simultaneously, we run another SOM classification $\mathbf{m}^2$, on the transposed dataset $\mathbf{x}'$, where each unit corresponds to an n-dimensional weight vector, where n is the number of samples in $\mathbf{X}$). Hence, the second uses training samples in the transposed space. Estimation of missing values operates exactly as in Equation 4.13.

We have now, two nonlinear estimations for each missing value $\mathbf{x}_{i,k}$ of the dataset. The first one is accurate when considering spatial dependencies, whereas the second integrates temporal correlations more efficiently. Then, we propose to linearly combine these two candidates according to their distances to their respective BMUs. Let d_1 be the inverse of the distance from the sample $\mathbf{x}_i$ to its associated BMU in $\mathbf{m}^1$,

$$d_1 = \left(\left\| \mathbf{x}_i - \mathbf{m}^1_{BMU_{\mathbf{x}_i}} \right\|_{NM_{\mathbf{x}_i}} \right)^{-1}. \tag{4.15}$$

We define d_2 equivalently as

$$d_2 = \left(\left\| \mathbf{x}'_k - \mathbf{m}^2_{BMU_{\mathbf{x}'_k}} \right\|_{NM_{\mathbf{x}'_k}} \right)^{-1}. \tag{4.16}$$

Then, for each missing value of $\mathbf{x}_{i,k}$, we estimate the missing values contained in the sample through the Two-Space-SOM by

$$\mathbf{x}_{i,k} = d_1 / (d_1 + d_2)\, \mathbf{m}^1_{BMU_{\mathbf{x}_i},k} + d_2 / (d_1 + d_2)\, \mathbf{m}^2_{BMU_{\mathbf{x}'_k},i}. \tag{4.17}$$

For the Two-Space-SOM, we still have to select the optimal grid sizes c^1 and c^2. This is done by using validation and the same validation sets for all combinations of the parameters c^1 and c^2. The Two-Space-SOM that gives the smallest validation error is used to perform the final completion of the data.

This procedure is definitely not guaranteed to work for every dataset. If there are clearly defined samples, which have no meaning or relationships in the transposed space, it is not clear whether one can get any benefit on training SOM in the transposed direction. It is neither clear whether the training on transposed space will deteriorate the results when there is no meaningful interpretation available.

As a practical perspective, if the data matrix dimensions, the numbers of rows and columns, differ heavily, it might not even be possible to train SOM properly. There are memory limitations, when samples start having several tens of thousands variables. In any case, it is possible to test the training of SOMs in both directions of dataset, since it is easy to see from the validation results whether both of the directions make sense.

4.6.3 Ensemble of SOMs

Natural extension from the previous combination of two different SOMs, is to combine several SOMs into a single ensemble. In this case, however, the database is not used in its transposed form, even though it would be possible to mix SOMs trained with the different spaces.

Currently, there are two different ways implemented to create the Ensemble of SOMs. The first is the one presented in Publication 9, where the linear combination of SOM maps is created using MultiResponse Sparse Regression and Hanna-Quinn Information Criterion. Since the SOM nodes are combined linearly, the problem is completely linear and the ranking obtained by the MRSR is exact.

Later, the creation of the ensemble was simplified by using Nonnegative Least-Squares algorithm (NNLS), presented in Publication 10. Both ensembling techniques yield to linear combination using positive weights, but the NNLS is faster and has proven to be more reliable in achieving good accuracy.

Both methodologies are summarized in the following figures, the methodology from Publication 9 is summarized in Figure 4.1 and the one from Publication 10 in Figure 4.2.

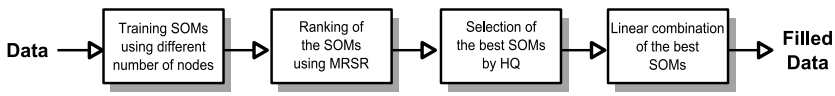


Figure 4.1: Ensemble of SOMs methodology from Publication 9.

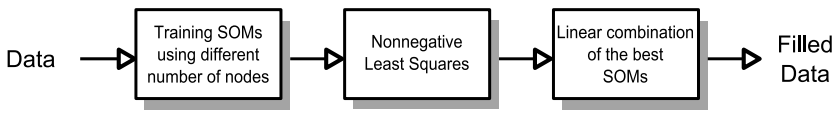


Figure 4.2: Ensemble of SOMs methodology from Publication 10.

In both ensemble techniques, in order to compute the linear combination weights, we have to remove a calibration set from the data before any processing. Then, the SOM estimations of the removed

calibration data are used as the variables of the linear equations and the removed data itself as the outputs of the equations. The linear system is summarized in the following formula:

$$\begin{bmatrix} \hat{s}_{1,1} & \hat{s}_{1,2} & \cdots & \hat{s}_{1,Q} \\ \hat{s}_{2,1} & \hat{s}_{2,2} & \cdots & \hat{s}_{2,Q} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{L,1} & \hat{s}_{L,2} & \cdots & \hat{s}_{L,Q} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_Q \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_L \end{bmatrix}, \qquad (4.18)$$

where s_i denotes the *ith* removed calibration sample, $\hat{s}_{i,j}$ denotes the *ith* calibration data sample estimated by *jth* SOM, L denotes the number of calibration data points, Q the number of the best SOMs used and, finally, the vector α denotes the linear system parameters.

When the α is solved, it can be used to estimate the originally missing values of the dataset from the best SOM estimations selected.

Since the process is not depending on any validation scheme, it is speeding up the traditional SOM imputation considerably. The lengthy validation setup is replaced by a simple calibration, which is orders on magnitude faster than validation, in both ensembling schemes.

4.7 COMBINATION OF SOM AND EOF

The two methodologies presented before, the SOM and the EOF, can be combined (see Publication 6 and 7). The SOM algorithm is first ran through performing a nonlinear projection for finding the missing values. Then, the result of the SOM estimation is used as initialization for the EOF method. The global methodology is summarized in Table 4.3.

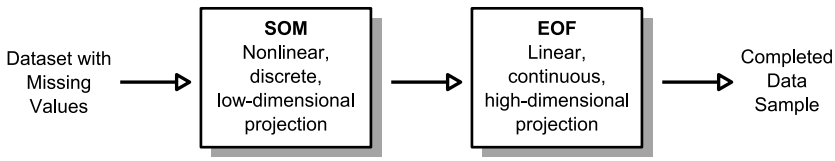


Figure 4.3: Summary of the SOM+EOF combination methodology.

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

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

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

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

Furthermore, the choice of which of the SOM filling methods to use with the EOF or EOF Pruning is arbitrary. Any SOM methodology can serve as initialization for whichever EOF methodology. There is a toolbox available for performing the SOM + EOF Pruning and for more information, see Section 5.2.

5 TOOLBOXES

This chapter presents briefly the two toolboxes created in the course of the thesis research work. Both toolboxes are available for download in [30] and are freely distributable under the GNU General Public License.

5.1 LAZY LEARNING TOOLBOX

Lazy Learning Toolbox for time series prediction is based on the original principle [15], that there's nothing to be done before the actual prediction is needed. Once the query arrives, the local learning procedure is started and all hyperparameters, variable selection as well as model training and validation is performed. However, there are situations where the laziness is not yielding the best performance and some degree of learning is worthwhile to be done beforehand.

The toolbox includes several different models, both lazy and non-lazy versions, several different model validation methods as well as several input variable selection methods. The toolbox also includes few different prediction strategies. Below is a comprehensive list of methodologies included.

Table 5.1: List of methodologies included in the Lazy Learning toolbox.

Model	Input Variable Selection	Validation	Neighbor Selection
Lazy Learning	Continuous Global Backward Local Backward	LOO	Global Local
Global Linear	Continuous Global Backward	LOO Bootstrap Bootstrap 632	
k-NN	Exhaustive Global Backward	LOO Bootstrap Bootstrap 632 LOO+Boot632	Global

Each of the combinations of different methodologies can be applied using Recursive or Direct prediction strategy up to 100 steps ahead.

Because the k-NN model is much faster to calculate than linear models, it is possible to do an Exhaustive search for the optimal input

variable set, given that the initial variable set size is reasonable.

5.2 SOM + EOF TOOLBOX

SOM+EOF Toolbox for missing value imputation is based on the methodologies presented in this thesis. It currently includes SOM, EOF, EOF Pruning, SOM+EOF and SOM+EOF Pruning methodologies. The SOM part of the toolbox takes advantage of the SOM Toolbox [26], which is included in the SOM+EOF Toolbox package.

All methodologies are applied using simple validation procedure, where certain set of data is removed to perform model parameter selection. After the model parameters are selected, the removed validation data is returned and final filling of the database is performed using the trained and validated model.

At the moment, there is no graphical user interface implemented, but all the scripts are well-documented and clearly explained within the codes. Furthermore, the Ensemble of SOMs methodology will be available in the toolbox shortly.

6 RESULTS

This chapter summarizes the main results contained in the included publications. The Publications from 1 to 4 present results related to time series prediction field and Publications from 5 to 10 related to imputation of missing values. Publication 7 shows an application of missing value imputation methodology in time series prediction context.

6.1 TIME SERIES PREDICTION

Publication 1 shows the superiority of Direct prediction strategy against Recursive one in long-term prediction of time series. Figure 6.1 shows the comparison using Poland Electricity dataset.

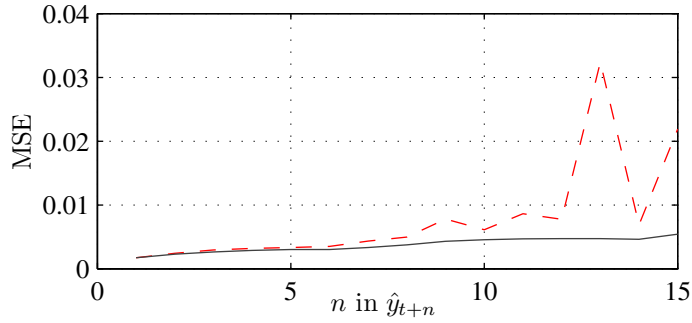


Figure 6.1: MSE of the Direct and Recursive prediction strategies for the test set of Poland Electricity Load data: solid line represents the direct prediction error and dashed line is for the recursive prediction error.

From Figure 6.1 it can be seen, that the Direct strategy gives smaller error than the Recursive one. The error difference increases as the horizon of prediction increases. The error of the Direct strategy is almost linear with respect to the horizon of prediction. This is not the case for the Recursive Strategy, since the accumulation of errors gets worse the further the prediction horizon reaches and around prediction step 13 the error leaps very high.

Publication 2 presents an application of the Direct strategy in a time series prediction contest. The used methodology is a combination of two projection strategies and a fast MLP network. Figure 6.2 shows the accuracy in 50 steps ahead prediction task using ESTSP 2007

Competition dataset.

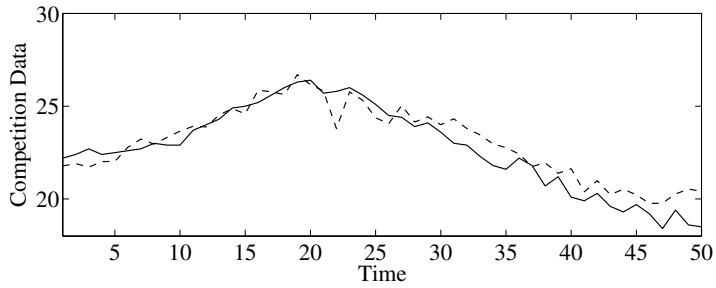


Figure 6.2: ESTSP 2007 Competition dataset, prediction of 50 values. Solid line represents the real value and the dashed one the prediction.

Figure 6.2 shows that the prediction is good in terms of visual inspection. For 15 steps ahead, the Mean Square Error (MSE) is 0.206 and for 50 steps ahead 0.686. The prediction performances are the 6^{th} and the 2^{nd} places, respectively, according to the results of the ESTSP 2007 prediction competition at the time.

Publication 3 demonstrates the DirRec strategy over Direct and Recursive using k-NN prediction methodology with Forward-Backward variable selection strategy and Leave-One-Out validation scheme. Table 6.1 shows test errors in Poland Electricity data and in Santa Fe laser dataset in 100 steps ahead prediction task.

Table 6.1: Average test errors of both datasets. The error values are averages of all 100 timesteps with each strategy through the entire test set.

	Average Test Errors	
	Santa Fe	Electricity Load
Recursive	3379	0.0318
Direct	1057	0.0124
DirRec	850	0.0098

From the table it can clearly be seen that the DirRec strategy performs better than the Direct or Recursive in this very difficult long term prediction task. Also, it is easy to see that the Direct strategy outperforms Recursive one.

Publication 4 presents the results comparing Multiple-Output strategies against Direct and Recursive using k-NN methodology in NN3 competition dataset. The results are summarized in Table 6.2 using Symmetric Mean Absolute Percentage of Error (SMAPE) measure.

Table 6.2 shows that the methods based on the Multiple-Output strategies outperform those based on the Single-Output strategies. The

Table 6.2: Comparison of Multiple-Output and Single-Output prediction strategies. MISMO-L and MISMO-G denote the local and global selection of neighbors, respectively. Minimum, Mean and Combination refer to the selection criterion for the s parameter, value which defines how many outputs are estimated at once.

Multiple-Output Strategy			Single-Output Strategy	
Strategy	Criterion	SMAPE	Strategy	SMAPE
MISMO-G	Minimum	17.63%	Direct	22.57%
	Mean	18.06%		
	Combination	16.50%		
MISMO-L	Minimum	19.50%	Recursive	21.17%
	Mean	18.57%		
	Combination	16.57%		
MIMO		18.19%		

interesting result is that regardless of the amount of outputs approximated at once, the Multiple-Output strategies outperform clearly the Single-Output ones.

Another interesting observation is that the Recursive Strategy seems to be slightly better than the Direct one. The reason is the lack of input selection. All previous publications, Publication 1, 2 and 3, included input variable selection or projection in their experiments. Hence, this definitely shows the importance of variable selection.

There are a few global notes to make from the time series prediction strategies. The first one is that MISMO strategy is the most accurate in long-term prediction. However, it requires selection of an extra parameter, when comparing to other strategies. If one has the time and computational resources to find the optimal parameter s, the MISMO strategy is the best strategy to go with.

The second note is, that if one does not want to deal with the extra parameter of MISMO strategy, the next best choice is DirRec. This strategy however definitely needs variable selection to be performed, before it can be reliably used.

The selection of variable selection strategy is not easy, since none of the strategies are completely satisfactory alone. As noted in Publication 1, the best course of action is to combine the efforts of different strategies and select the best from the results of all of them. Furthermore, since the Forward-Backward strategy allows starting from an arbitrary variable set, it should be repeated several times from different starting sets.

Finally, k-NN search strategy is always a very good choice, because of its simplicity, accuracy and fast computational time.

6.2 IMPUTATION OF MISSING VALUES

Publication 5 reports the accuracy of the improvement EOF Pruning with respect to the original EOF and an Optimal Interpolation method. Table 6.3 shows the summary of validation and test errors of all methods for the southern slice of the Tanganyika Lake dataset.

Table 6.3: Validation and Test Errors for the EOF, the EOF Pruning and the OA using the southern slice of the Tanganyika dataset.

	Validation MSE	nEOF	Test MSE
EOF	0,0839	13	0,0664
EOF Pruning	0,0543		0,0517
Objective Analysis	0,6210		0,6265

From the summary Table 6.3, it is quite clear that the Objective Analysis is not able to fill in the missing values accurately. The error is an order of magnitude larger than those of the EOF and EOF Pruning. Furthermore, according to the table, the EOF Pruning outperforms the original EOF reducing the validation error roughly by one third and the test error by 23 percent.

Publication 6 demonstrates the need to select SOM grid size and number of EOFs to use together, when the two methodologies are combined. In the first database of the paper, the optimal SOM size is validated to 26×26 and optimal number of EOFs to 6. But when the two methodologies are combined, the optimal setting is 18×18 and 40.

The methodologies alone and the combination of them is compared against the ECM, which is considered to be the state-of-the-art in financial field for imputation of missing values. There are several databases utilized and the results are similar with each of them. Table 6.4 shows the results of the comparison with one publicly available database.

Table 6.4: Validation and test RMS errors for all the methods using a publicly available financial dataset.

10^{-2}	Validation Error	Test Error
ECM	4.18	4.34
SOM	4.17	3.92
EOF	3.95	3.88
SOM + EOF	3.89	3.60

Comparing the validation and test errors in the Table 6.4, the

SOM+EOF clearly outperforms the other methodologies. Since this dataset is the hardest one, all the errors are rather close to each other.

Also it can be noted that the SOM, EOf and SOM + EOF have lower test error than validation error. This is many times the case since there is more data available for performing the test, because the validation data is added to the training set when estimating the test performance. However, this is not the case with ECM methodology, the validation result is overoptimistic with respect to the obtained test error. Perhaps the methodology suffers from overfitting to the training data.

Publication 7 presents a cross field experiment, where time series prediction is considered as missing value problem. There are two competition datasets used in the experiments, ESTSP 2007 and NN3.

It is not mentioned in the publication, that the predictions achieved 6th place in ESTSP Competition and 4th place in NN3 competition. The obtained predictions were submitted to the competition at the same time than the publication.

According to the validation errors, the SOM + EOF methodology outperforms the methods alone in two of the example cases, and in one case achieves the same performance. Visual inspection of the prediction results seem to verify the accuracy of the predictions. The predictions of the three examples are shown in Figures 6.3, 6.4 and 6.5.

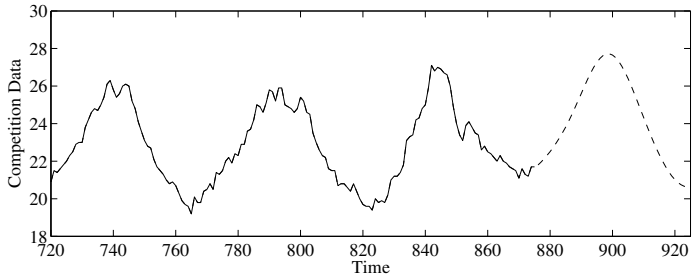


Figure 6.3: Prediction of 50 next values of the ESTSP 2007 Competition dataset. The real values are presented by the solid line and the dashed one presents the prediction.

Publication 8 introduces the Two-Space-SOM and the experiments boil down to Table 6.5. The Two-Space-SOM is effectively a combination of two SOMs, the L-SOM and the X-SOM, and L-SOM denotes the SOM in the original space and X-SOM in the transposed space.

From Table 6.5 we can see that the Two-Space-SOM outperforms the L-SOM and the X-SOM reducing the validation error by 19 and 28 percent, respectively, and the test error by 23 and 31 percent.

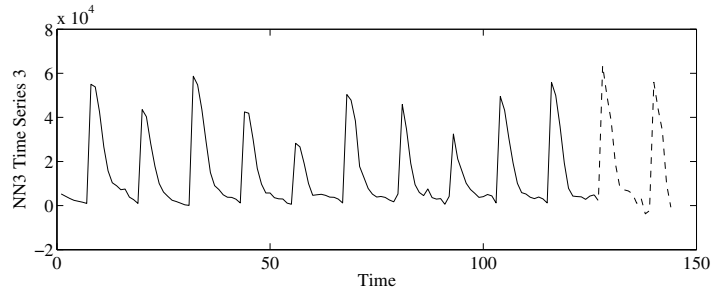


Figure 6.4: Prediction of the 3^{rd} time series of the NN3 prediction competition. Solid line represents the known time series and the dashed one the prediction using the SOM+EOF method.

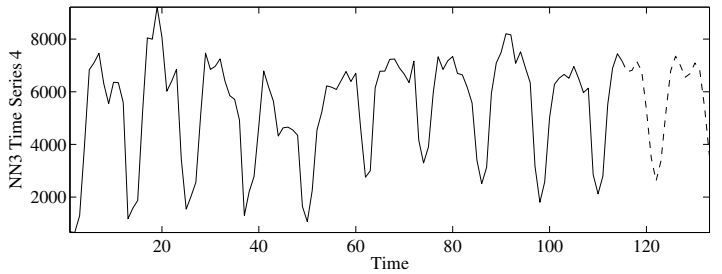


Figure 6.5: Prediction of the 4^{th} time series of the NN3 prediction competition. Solid line represents the known time series and the dashed one the prediction.

Table 6.5: Learning and Test Root Mean Squared Errors for the Expectation Conditional Maximization (ECM), the L-SOM, the X-SOM, the EOF, the SOM+EOF and the Two-Space-SOM using a financial dataset.

10^{-3}	Learning Error	Test Error
ECM	2.8	3.7
L-SOM	1.6	1.7
X-SOM	1.8	1.9
EOF	1.6	1.7
SOM + EOF	1.4	1.6
Two-Space-SOM	1.3	1.4

It can also be seen that SOM + EOF is not as good as Two-Space-SOM, but all individual methods are outpaced. Whether that is due to the incorrect selection of SOM space to be combined with the EOF or incorrect selection of parameters, is unclear.

Inspired by the combination of two SOMs in Publication 8, Publication 9 shows the first try in combining several SOMs. This ensemble is created using the MRSR and HQ criterion. First, Figure 6.6 shows the Hannan-Quinn Information Criterion values with respect to the number of SOMs in the combination.

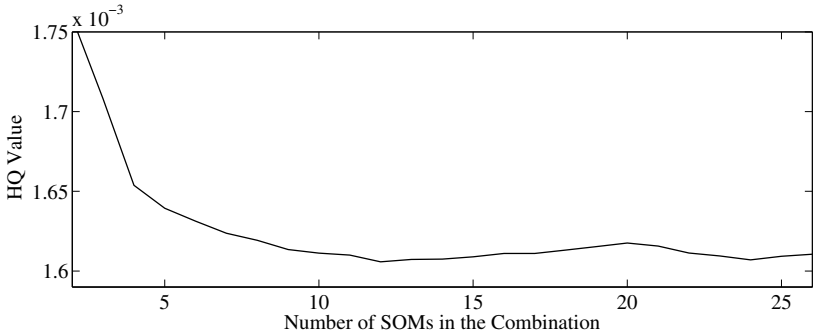


Figure 6.6: Hanna-Quinn Information Criterion values for the selection of SOMs in the combination.

From Figure 6.6 we can see that the most optimal value is reached with 12 SOMs. The selected SOM sizes are 7, 9, 12, 16, 18, 20, 21, 22, 23, 24, 25 and 26. Here the maximum SOM grid size was 26. From the previous list we can clearly see that most of the small SOMs are not accurate enough to be included in the combination, but several larger sizes are. Comparing this to Figure 6.6 it is also clear that after the 12 selected SOMs the HQ value is worse, which means that the rest of the SOMs do not improve the results.

On the other hand, from the list of selected SOM sizes, the smallest one is 7, corresponding to a SOM of $7 \times 7 = 49$ nodes and maximum size $26 \times 26 = 676$ nodes. The database used in the experiments has 120 samples, so the minimum sized SOM has much less nodes than samples in the database and the largest one much more. The effect of the ensembling can be verified from Table 6.6, where the errors are summarized.

Table 6.6: Test Errors for the traditional SOM and the Ensemble of SOMs.

10^{-3}	Training Evaluation Error	Test Error
Traditional SOM	1.8	1.6
Ensemble of SOMs		1.3

From Table 6.6 we can see that the Ensemble of SOMs clearly outperforms the single SOM decreasing the test error by 18 percent.

Publication 10 presents the other variant of the Ensemble of SOMs, including imputation results using two very different databases. Table 6.7 shows one of the two sets of results and the first try on combination of the Ensemble of SOMs + EOF Pruning. The combination is not presented in Publication 10, but it is the start of testing further combinations of methodologies.

Table 6.7: The results of all methods using Tanganyika dataset.

	Validation MSE	Test MSE	Computational Time
EOF	0.0553	0.0595	30.3 hours
EOF Pruning	0.0442	0.0426	23.3 hours
SOM	0.0393	0.0379	3.14 hours
Ensemble of SOMs		0.0280	0.65 hours
Ens.SOM+EOFPrun	0.0296	0.0259	1.57 hours
PPCA	0.0700	0.0818	$>$ 8 days

From Table 6.7 we can see that according to the test error, Ensemble of SOMs together with EOF Pruning is the best. The EOF Pruning performed after the Ensemble of SOMs reduces the test error by 7.5 percent. Ensemble of SOMs is close to the combination performance with computational time of just fractions compared to other methodologies.

The EOF methodologies alone are not able to fill the missing values as accurately as the methodologies related to the SOM.

Computational time is also the smallest when using the Ensemble of SOMs, largely due to the lack of lengthy validation procedure.

Comparing the two methods of creating the Ensemble of SOMs, in Table 6.8 there are both combination techniques compared using two dataset, Anthrokids and one finance dataset. In both cases, the test error is obtained as average of 10 different test sets.

Table 6.8: Comparing the two Ensembles of SOMs.

Dataset	Method	Test MSE	Variance
Anthokids	NNLS	0.3058	0.0002
	MRSR+HQ	0.3085	0.0003
Finance	NNLS	0.417	0.044
	MRSR+HQ	0.423	0.046

From table 6.8, we can see that there seems to be small benefit towards NNLS combination technique. However, the differences between the 10-fold test errors are really small with both datasets.

But the difference becomes more apparent when we compare each test set individually and count the number of times when the NNLS technique outperforms the MRSR+HQ combination. In Anthrokids dataset the NNLS is better in 7 test sets and in finance dataset all sets except one. This clearly shows the NNLS to be the better combination technique in terms of the test error.

Furthermore, the testing also showed that the NNLS is faster than the MRSR+HQ combination and that the NNLS selected generally fewer SOMs in the combination than MRSR+HQ.

Considering the presented results, the Ensemble of SOMs is the most viable option for imputation of missing values. That is, if only one methodology has to be selected. The Ensemble of SOMs is very accurate with respect to other compared methodologies and the speed is superb, thanks to the removal of lengthy cross-validation procedure. Furthermore, it is advisable to use the NNLS technique for creating the ensemble, since it is accurate and fast at the same time.

On the other hand, the combination of two very different methodologies provides the best accuracy. Even though the computation takes a little longer when using two consecutive imputation methodologies, the computational time can be compensated by using faster ones, like the Ensemble of SOMs. Therefore, the combination of the Ensemble of SOMs and the EOF Pruning is a very good choice for solving the imputation problem. These two methodologies complete each other. The same applies to the earlier experiments with the SOM + EOF combination methodology.

Still, considering that there are so many different databases available, it is not possible to guarantee that one imputation scheme would always be the best one. So, the best course of action is to always try several possibilities and validate the methodologies carefully, before performing any final filling of the database, especially in the cases where the imputation is a preprocessing method for some other analyzing technique.

7 SUMMARY, CONCLUSIONS AND FURTHER WORK

7.1 SUMMARY OF THE WORK

According to the results of the comparisons, it can be noted that Direct strategy is more accurate in long-term prediction than Recursive one. This is due to the accumulation of prediction errors, which is nonexistent in Direct strategy.

Of course one has to remember that if the time series to be predicted is completely noiseless, then, in theory, the Recursive strategy should provide very good results, even in long-term prediction, given that the trained model is not making prediction error. However, Direct strategy should not be any worse than Recursive strategy in the case of noiseless data, but the accumulation of prediction error is avoided and hence the accuracy should be better.

The same observation can be done regarding the DirRec strategy, which surpasses both previously mentioned ones in terms of accuracy in the long run. However, DirRec strategy has a downside. The size of the input variable set keeps growing the further to the prediction horizon we go and this makes computational time rise. Furthermore, one definitely needs a valid variable selection strategy and search criteria to overcome the constantly increasing amount of input variables. This increases the computational time and decreases the benefit received in accuracy.

Moving to multiple-output strategies, MIMO and MISMO, the results are convincing that considering the prediction horizon points together is a valid choice. But considering all of them at once, is not as good as selecting a portion of points at a time to be predicted simultaneously. Therefore, MISMO strategy is more accurate than MIMO, which still surpasses Direct and Recursive. Even though MISMO strategy has an extra parameter to tune, it is the best option for long-term prediction at the moment.

Depending on the amount of input variables to begin with, the choice of variable selection strategy should be considered differently. Naturally, if there are reasonable amount of variables, the Exhaustive search provides the optimal selection, given the search criteria. The amount of variables after which it is not reasonable to perform the Exhaustive search, depends greatly on the computational resources available, the amount of samples in the training dataset and the selected search criteria. At this time, for a dataset with a thousand samples in 20 dimensional space, the Exhaustive search is still reasonable using a fast search criteria.

If the amount of variables is very high, other variable selection

strategies or projection strategy might come in handy. Since none of the other strategies than the Exhaustive search guarantee the optimal solution to be found, it is best to use a combination of them, even all, if possible. It is also good to note that if the projection strategy is used, the interpretability of the variables is lost in the mapping process.

With reliable prediction strategy, a clever variable selection or projection strategy and fast search criteria, linear models can achieve very good performance, even in nonlinear problems. It can be thought that the variable selection or projection transforms the nonlinear problem to a more linear state and then even the linear model is able to perform well.

Even though Local Linear Models are linear in the local space, globally they are not anymore linear. That means that the problem does not need to be globally linear, it suffices that the problem is linear in a local neighborhood of the prediction point. Combining this with variable selection or projection strategy, it is possible to enhance the local linearity and obtain good performance by using local linear models.

Moving to missing values, the EOF Pruning not only enhances the accuracy of the original EOF, but also speeds up the process. This is due to the heavy decrease in the number of rounds performed in the methodology. Even though more time is spent each round, the computational time is decreased.

Then, the improvements of SOM, Two-Space-SOM and the Ensemble of SOMs, increase the accuracy over the traditional one. The Two-Space-SOM works with the original input space as well as with the transposed one, which improves the accuracy. On the other hand, it requires roughly the same amount of samples in both spaces, which decreases the applicability of the methodology.

The Ensemble of SOMs not only increases accuracy over the traditional SOM, but also works much faster, since the lengthy validation procedure is removed. The increase in accuracy is due to the cooperation of SOMs with different amount of nodes, which makes each SOM represent the input space differently.

The needed performance compared against the required response time is also very relevant, especially when working with more online type systems. Also, many industrial applications require fast response time in order to be able to take advantage of the modern strategies and computational models. This makes the Ensemble of SOMs even more valuable and appealing choice for filling the missing values, because the validation procedure is not slowing down the process and the accuracy is very good.

If ultimate accuracy is required, a combination of the SOM and the EOF methodologies is a valid option. Any of the SOM and

EOF methodologies can be combined, but the best one in terms of computational time as well as accuracy, is deemed to be the Ensemble of SOMs + EOF Pruning. Both methodologies are faster and more accurate versions of the original ones and even better when working together.

7.2 CONCLUSIONS

As a main conclusion, the speed of any methodology has to be acceptable, while providing reasonably accurate results. There is no point in wasting time and resources by using slow or inaccurate methodologies. If the computation of the needed outcome takes weeks, the methodology responsible for the computation can be forgotten and the focus of the researcher should be moved to faster methods. On the other hand, inaccurate methodologies are irrelevant in practice, no matter how fast they are. There is no place for compromises.

Secondly, even older methods should be kept available and in use. There is no point in inventing new methodologies that cannot beat the old ones. This makes it really valuable to compare the previous versions with the current ones, in order to quantify the value of the improvement.

Finally, simple methods are always the best. When using a simple method, which is also fast and reasonably accurate, it is easy and fast to try several preprocessing schemes, variable selection strategies and prediction strategies. With a complex, hence slow, method one needs to have a good knowledge of the most optimal selections before applying the method, or otherwise the computational time is too long to test all the possibilities.

With simpler models, it is straightforward to ensemble them into more accurate modeling machine, even using parallel processing in the obvious way. Furthermore, using one single model to solve all problems is a thing of the past, ensembling several models into one is the thing of the future.

7.3 FURTHER WORK

In the course of this thesis work, several new methodologies and techniques have been developed and compared in real world test cases. When combining the methodologies with, for example, variable selection, scaling or projection techniques, there are very large number of possible combinations to be tested. In many cases, the optimal choice cannot be defined *a priori*, but several combinations have to

searched through and validated in order to select a good one. This procedure takes a vast amount of time and computational resources.

Regarding the developed methodology, the Ensemble of SOMs, there is a clear need for further testing, since several open questions remain: How much calibration data is needed with certain amount of missing values present in the dataset? Is there strict lower limit to the amount of calibration data other than the number of SOMs in the ensemble? Is it beneficial to include SOMs trained with the transposed data space? How about size of the SOMS, which ones to include? what happens to the performance, if several same sized SOMs are included? All these questions demand further investigation.

Another interesting challenge transpired in the course of the research, is the creation of new intelligent parallel algorithms, which are also scalable into large computing systems. Using smart way to parallelize the computation would make it more easier to test more combination, parameters values and different algorithms in less time. There are more and more computing power available, why not take the full advantage out of it?

At the same time, the already existing methodologies and techniques need to be improved, not only to enable the intelligent parallel and scalable computing possible, but also to speed up the computation and make the achieved results more accurate and reliable. Which combinations of methodologies are good in which sense, need to be determined in order to select the most appropriate ones for each problem.

Furthermore, the data collection processes are improving all the time and larger and larger datasets are available in many fields. The author of the thesis is currently in the process of moving to a field of bioinformatics, where computational resources and the problem complexity are not yet fully meeting each other in an optimal way. The bioinformatics research affects the whole human kind with the endless possibilities to make the life better for everyone. It is very important and necessary research area and it is my feeling that machine learning and intelligent parallelization can help the bioinformatics researchers significantly.

BIBLIOGRAPHY

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

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

[3] P. Allison. *Missing Data*. Sage Publications Inc., New York, 2001.

[4] A. Alvera-Azcarate, A. Barth, J.M. Beckers, and R. H. Weisberg. Multivariate reconstruction of missing data in sea surface temperature, chlorophyll and wind satellite fields. *Journal of Geophysical Research*, (C03008), 2007.

[5] A. Alvera-Azcarate, A. Barth, M. Rixen, and J.M. Beckers. Reconstruction of incomplete oceanographic data sets using empirical orthogonal functions. application to the adriatic sea. *Ocean Modelling*, 9:325–346, 2005.

[6] G. Barreto. Time series prediction with the self-organizing map: A review. In Barbara Hammer and Pascal Hitzler, editors, *Perspectives of Neural-Symbolic Integration*, volume 77 of *Studies in Computational Intelligence*, pages 135–158. Springer Berlin / Heidelberg, 2007.

[7] G.A. Barreto and L.G.M. Souza. Adaptive filtering with the self-organizing map: A performance comparison. *Neural Networks*, 19(6-7):785 – 798, 2006. Advances in Self Organising Maps - WSOM'05.

[8] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Network*, 50:537–550, 1994.

[9] J.M. Beckers, A. Barth, and A. Alvera-Azcarate. Dineof reconstruction of clouded images including error maps. application to the sea surface temperature around corsican island. *Ocean Science*, 2(2):183–199, 2006.

[10] J.M. Beckers and M. Rixen. Eof calculations and data filling from incomplete oceanographic datasets. *Journal of atmospheric and oceanic technology*, 20(12):1839–1856, 2003.

[11] R.E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[12] M. Ben-Bassat. Pattern recognition and reduction of dimensionality. *Handbook of statistics II*, pages 773–910, 1982.

[13] R. J. Bhansali and D. Y. Downham. Some properties of the order of an autoregressive model selected by a generalization of akaike's epf criterion. *Biometrika*, 64(3):547–551, 1977.

[14] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York, 1977.

[15] M. Birattari, G. Bontempi, and H. Bersini. Lazy learning meets the recursive least-squares algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *NIPS 11*, pages 375–381, Cambridge, 1999. MIT Press.

[16] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[17] C.M. Bishop. Variational principal components. In *In Proceedings Ninth International Conference on Artificial Neural Networks, ICANNŠ99*, pages 509–514, 1999.

[18] G. Bontempi. *Local Learning Techniques for Modeling, Prediction and Control*. Ph.d., IRIDIA-Université Libre de Bruxelles, Louvain-la-Neuve, BELGIUM, 1999.

[19] G. Bontempi. Long term time series prediction with multi-input multi-output local learning. In *Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP08*, 2008.

[20] G. Bontempi, M. Birattari, and H. Bersini. Lazy learning for modeling and control design. *International Journal of Control*, 72(7/8):643–658, 1999.

[21] J. Boyd, E. Kennelly, and P. Pistek. Estimation of eof expansion coefficients from incomplete data. *Deep Sea Research*, 41:1479–1488, 1994.

[22] F. Corona, E. Liitiäinen, A. Lendasse, L. Sassu, S. Melis, and R. Baratti. A SOM-based approach to estimating product properties from spectroscopic measurements. *Neurocomputing*, 73(1–3):71–79, December 2009.

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

[24] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley, New York, 1991.

[25] S.F. Crone and N. Kourentzes. Feature selection for time series prediction - a combined filter and wrapper approach for neural networks. *Neurocomput.*, 73(10-12):1923–1936, 2010.

[26] J. Parhankangas E. Alhoniemi, J. Himberg and J. Vesanto. SOM Toolbox: http://www.cis.hut.fi/projects/somtoolbox/.

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

[28] B. Efron and R. J. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92:548–560, 1997.

[29] B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, 1993.

[30] EIML Research Group, 2010. Download Toolboxes: http://www.cis.hut.fi/projects/eiml/research/downloads.

[31] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Physical Review Letters*, 8(59):845–848, 1987.

[32] J. D. Farmer and J. J. Sidorowich. Exploiting chaos to predict the future and reduce noise. Technical report, Los Alamos National Laboratory, 1988.

[33] F. Fessant and S. Midenet. Self-organising map for data imputation and correction in surveys. *Neural Computing & Applications*, 10(4):300–310, 2002.

[34] D. François. High-dimensional data analysis : optimal metrics and feature selection. 2007.

[35] L.S. Gandin. Objective analysis of meteorological fields. *Israel Program for Scientific Translations, Jerusalem*, page 242, 1969.

[36] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[37] J.W. Han and M. Kamber. *Data mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, 2001.

[38] E. J. Hannan and B. G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society, B*, 41:190–195, 1979.

[39] T. Ikeguchi and K. Aihara. Prediction of chaotic time series with noise. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E78-A(10), 1995.

[40] A. J. Jones. New tools in non-linear modeling and prediction. *Computational Management Science*, 1:109–149, 2004.

[41] J.M.P. Menezes Junior and G.A. Barreto. Long-term time series prediction with narx network: An empirical evaluation. *'Neurocomputing*, 2008.

[42] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 2, 1995.

[43] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.

[44] T. Kohonen. Data management by self-organizing maps. (5050):309–332, 2008.

[45] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Phys. Rev.*, 69:066138, 2004.

[46] A. Lendasse and E. Liitiäinen. Variable scaling for time series prediction: Application to the ESTSP'07 and the NN3 forecasting competitions. pages 2812 –2816, aug. 2007.

[47] A. Lendasse, G. Simon, V. Wertz, and M. Verleysen. Fast bootstrap methodology for model selection. 2005.

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

[49] L. Ljung. *System identification theory for User*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[50] E. N. Lorenz. Atmospheric predictability as revealed by naturally occurring analogues. *Journal of the Atmospheric Sciences*, 26:636–646, 1969.

[51] G. Manzini. Perimeter search in restricted memory. *Computers and Mathematics with Applications*, 32:37–45, 1996.

[52] S. Massoni, M. Olteanu, and P. Rousset. Career-path analysis using optimal matching and self-organizing maps. In *WSOM '09: Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps*, pages 154–162, Berlin, Heidelberg, 2009. Springer-Verlag.

[53] R. Meiri and J. Zahavi. Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171:842–858, June 2006.

[54] Y. Miche, P. Bas, C. Jutten, O. Simula, and A. Lendasse. A methodology for building regression models using extreme learning machine: OP-ELM. In M. Verleysen, editor, *ESANN 2008, European Symposium on Artificial Neural Networks, Bruges, Belgium*, pages 247–252. d-side publ. (Evere, Belgium), April 23-25 2008.

[55] 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, January 2010.

[56] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[57] F.M. Pouzols, A. Lendasse, and A.B. Barros. Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation. *Fuzzy Sets Syst.*, 161(4):471–497, 2010.

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

[59] W.J. Puma-Villanueva, E.P. dos Santos, and F.J. Von Zuben. Long-term time series prediction using wrappers for variable selection and clustering for data partition. pages 3068 –3073, aug. 2007.

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

[61] E. Rasek. A contribution to the problem of feature selection with similarity functionals in pattern recognition. *Pattern Recognition*, 3(1):31–36, April 1971.

[62] S. L. Shah S. A. Imtiaz. Treatment of missing values in process data analysis. *The Canadian Journal of Chemical Engineering*, 86.

[63] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[64] Q. Shen and R. Jensen. Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring. *Pattern Recognition*, 37(7):1351–1363, July 2004.

[65] 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.

[66] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, and M. Verleysen. Double som for long-term time series prediction. Workshop on Self-Organizing Maps, Kitakyushu, Japan, 11-14 September.

[67] A. Sorjamaa. Strategies for the long-term prediction of time series using local models. Master's thesis, Helsinki University of Technology, October 14 2005. Master Thesis obtained with the grade 5.

[68] A. Sorjamaa, N. Reyhani, and A. Lendasse. Input and structure selection for k-nn approximator. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval Hernandez, editors, *Lecture Notes in Computer Science*, volume 3512, pages 985–991. Computational Intelligence and Bioinspired Systems: 8^{th} International Work-Conference on Artificial Neural Networks, IWANN 2005, Barcelona, Spain, Springer Berlin / Heidelberg, 2005.

[69] D. Sovilj, A. Sorjamaa, and Y. Miche. Tabu search with delta test for time series prediction using OP-KNN. In Amaury Lendasse, editor, *ESTSP, European Symposium on Time Series Prediction*, pages 187–196, Porvoo, Finland, September 17-19 2008. Multiprint Oy / Otamedia , Espoo, Finland.

[70] D. Sovilj, A. Sorjamaa, Q. Yu, Y. Miche, and E. Séverin. OPELM and OPKNN in long-term prediction of time series using projected input data. *Neurocomputing*, 73(10-12):1976–1986, June 2010.

[71] A. Stefansson, N. Koncar, and A.J. Jones. A note on the gamma test. *Neural Computing & Applications*, (5(3)):131–133, 1997.

[72] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing Co., Pte, Ltd. (Singapore), November 2002.

[73] S. Ben Taieb, G. Bontempi, A. Sorjamaa, and A. Lendasse. Long-term prediction of time series by combining direct and mimo strategies. In *Proceedings of the 2009 IEEE International Joint Conference on Neural Networks*, 2009.

[74] F.T. Tangang, B. Tang, A.H. Monahan, and W.W. Hsieh. Forecasting enso events: A neural network-extended eof approach. *Journal of Climate*, 11:29Ű41, January 1998.

[75] J. Tikka and J. Hollmén. Long-term prediction of time series using a parsimonious set of inputs and ls-svm. 2008.

[76] J. Tikka, J. Hollmén, and A. Lendasse. Input selection for long-term prediction of time series. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval, editors, *Computational Intelligence and Bioinspired Systems*, volume 3512 of *Lecture Notes in Computer Science*, pages 1002–1009. Springer Berlin / Heidelberg, 2005.

[77] Unknown Author. *The Book of The Dead, The Papyrus of Ani*. 1240 BC. translated by E.A. Wallin Budge and Allen and Faulkner.

[78] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In Joan Cabestany, Alberto Prieto, and Francisco Sandoval Hernandez, editors, *Lecture Notes in Computer Science*, volume 3512, pages 758–770. Invited talk in Computational Intelligence and Bioinspired Systems: 8^{th} International Work-Conference on Artificial Neural Networks, IWANN 2005, Barcelona, Spain, Springer Berlin / Heidelberg, 2005.

[79] M. Verleysen, D. François, G. Simon, and V. Wertz. On the effects of dimensionality on data analysis with neural networks. In *IWANN '03: Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks*, pages 105–112, Berlin, Heidelberg, 2003. Springer-Verlag.

[80] T. Villmann and E. Merényi. Extensions and modifications of the kohenen-som and applications in remote sensing image analysis. pages 121–144, 2002.

[81] H. Wackernagel. *Multivariate Geostatistics - An Introduction with Applications*. Springer, Berlin, 1995.

[82] L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pages 263–268, aug 1991.

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

[84] H. Wold. Partial least squares. In *Encyclopedia of Statistical Sciences*, volume 6, pages 581–591. Wiley, New York, 1985.

[85] Y. Xia, P. Fabian, A. Stohl, and M. Winterhalter. Forest climatology: Estimation of missing values for bavaria, germany. *Agricultural and Forest Meteorology*, 96(1-3):131 – 144, 1999.

I have knitted myself together,
I have made myself whole and complete.
I shall renew my youth.
I am Osiris Himself, the Lord of Eternity

– Osiris Ani, The Book of The Dead [77]