Year : 2018

Environmental Data Mining using Machine Learning Algorithms: Methodological Developments and Case Studies

Leuenberger Michael

UNIL | Université de Lausanne

Faculté des Géosciences et de l'Environnement
Institut des Dynamiques de la Surface Terrestre

Environmental Data Mining using Machine Learning Algorithms: Methodological Developments and Case Studies

Thèse de doctorat

Présentée à la

Faculté des Géosciences et de l'Environnement
de l'Université de Lausanne
par

Michael Leuenberger
B.Sc., M.Sc. Université de Neuchâtel, Suisse

diplômé en

Docteur en Sciences de l'Environnement

Jury

Directeur de Thèse	Prof. Mikhail Kanevski
Expert interne	Prof. François Bavaud
Expert externe	Prof. Vasily Demyanov
Expert externe	Dr. Stéphane Joost
Président du colloque	Prof. Michel Jaboyedoff

Lausanne, 2018

UNIL | Université de Lausanne
Faculté des géosciences et de l'environnement
bâtiment Géopolis bureau 4631

IMPRIMATUR

Vu le rapport présenté par le jury d'examen, composé de

Président de la séance publique :	M. le Professeur Michel Jaboyedoff
Président du colloque :	M. le Professeur Michel Jaboyedoff
Directeur de thèse :	M. le Professeur Mikhail Kanevski
Expert interne:	M. le Professeur François Bavaud
Expert externe:	M. le Professeur Vasily Demyanov
Expert externe:	M. le Docteur Stéphane Joost

Le Doyen de la Faculté des géosciences et de l'environnement autorise l'impression de la thèse de

Monsieur Michael LEUENBERGER

Titulaire d'une
Maîtrise universitaire ès sciences en mathématiques
de l'Université de Neuchâtel

intitulée

Environmental Data Mining using Machine Learning Algorithms: Methodological Developments and Case Studies

Lausanne, le 19 janvier 2018

Pour le Doyen de la Faculté des géosciences et de l'environnement

Professeur Michel Jaboyedoff

“*La simplicité est la sophistication suprême.*”

Léonard de Vinci

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Mikhail Kanevski. From the beginning of this PhD, he always found the right words to motivate me and to give me the desire for learning more. I am very grateful to have had the opportunity to explore the fields of machine learning with him.

I would also like to acknowledge the jury members Prof. François Bavaud of the University of Lausanne, Prof. Vasily Demyanov of the Heriot-Watt University (Edinbourg) and Dr. Stéphane Joost of the Swiss Federal Institute of Technology in Lausanne, for their constructive comments and suggestions, which greatly helped to improve the quality of the thesis.

I would like to express my sincere gratitude to Prof. Jorge Mário Gonzalez Pereira and Joana Parente of the University of Trás-os-Montes and Alto Douro (portugal), and Dr. Antonino Marvuglia of the Luxembourg Institute of Science and Technology for the scientific discussions and the prosperous collaborations.

I would like to thank all the colleagues and friends from the University of Lausanne. In particular Carmen, Jean, Mohamed and Fabian who shared the 3141 office with me throughout the thesis, Mary for the agreeable research collaboration, and Zhivko for the fun time spent here (and in Madrid). I really appreciated every moments spent with you all.

From outside the university, I would like to thank my friends from La Montagnarde who helped me to free my mind in our sport. A grateful thank to Yann, Sven and Sarah for the enjoyable moments spent in Switzerland or in Luxembourg playing coinch. And a big thank to all the Old Skulls guy for the countless wipes: GG, and see you IG.

A special thanks to my parents Gilbert and Catherine for your support, encouragement and love received during my PhD. Also, I want to thank my sisters Aurelie and Sabrine, my brother Hakim, and my brother-in-law Christophe for your laughs and humour. All of you helped me so much during these five years, thank you!

Finally, I deeply thank Nadia. From the bottom of my heart I thank you for always being there for all these beautiful days we spent and will spent together. Your encouragement through these years was invaluable.

No combination of words can express how grateful I am to all of you!

Michael Leuenberger, January 2018

Environmental Data Mining using Machine Learning Algorithms: Methodological Developments and Case Studies

Michael Leuenberger

Institute of Earth Surface Dynamics

Abstract

Due to the large amount and complexity of data available nowadays in geo- and environmental sciences, we face the need to develop and incorporate more robust and efficient methods for their analysis, modelling and visualization. An important part of these developments deals with an elaboration and application of a contemporary and coherent methodology following the process from data collection to the justification and communication of the results. Recent fundamental progress in machine learning can considerably contribute to the development of this emerging field – environmental data science.

The main purpose of this Thesis is to develop coherent and self-consistent methodologies for the analysis of environmental phenomenon using machine learning algorithms. In particular, this Thesis gives an overview of machine learning algorithms for environmental data mining. It highlights and investigates the different issues that can occur when dealing with complex and high dimensional environmental data using cutting-edge machine learning algorithms. In addition, several important topics of data driven modelling, including data splitting, complexity analysis, residuals assessment, feature selection and uncertainties are discussed.

Moreover, a special attention is paid to the Extreme Learning Machine algorithm (ELM). Being an efficient artificial neural network, it gained recently a great popularity in the domain of machine learning. By taking advantage of its quickness, another objective of this Thesis is to extract the potential of ELM for the tasks of feature selection and uncertainty quantification. Both of these approaches can give valuable information about the hidden relationship between input and output variables, which indirectly reflects the behaviour of the studied phenomenon.

In this regard, the general leitmotif of this Thesis is focused on the development of coherent methodologies for the analysis of environmental phenomenon using machine learning algorithms. The applied part of the research deals with an application of the methodology and the developed methods for simulated, modelled and real environmental data, such as forest fires, pollution and wind fields.

Environmental Data Mining using Machine Learning Algorithms: Methodological Developments and Case Studies

Michael Leuenberger

Institut des Dynamiques de la Surface Terrestre

Résumé

En raison de la grande quantité et de la complexité des données disponibles de nos jours dans les géosciences et sciences de l'environnement, nous sommes confronté.e.s à la nécessité de développer et d'intégrer des méthodes plus robustes et plus efficaces pour leurs analyses, modélisations et visualisations. Une partie importante de ces développements traite de l'élaboration et de l'application d'une méthodologie cohérente depuis la récolte des données jusqu'à la justification des résultats en passant par leur divulgation. Les progrès fondamentaux ayant récemment eu lieu dans le domaine des apprentissages automatiques (*machine learning*) contribuent à l'émergence du domaine appelé *environmental data science.*

Le principal objectif de cette Thèse est le développement de méthodologies cohérentes utilisant les algorithmes d'apprentissage automatique pour l'analyse de phénomènes environnementaux. En particulier, cette Thèse fournit une vue d'ensemble des algorithmes d'apprentissage automatique pour l'extraction d'information dans les données environnementales (*environmental data mining*). Elle met en évidence et examine les différents problèmes qui peuvent survenir lors de l'application d'algorithmes d'apprentissage automatique sur des données environnementales complexes à hautes dimensions. Plusieurs problématiques liées à ces algorithmes sont discutées, telles que le fractionnement des données, l'analyse de la complexité, l'évaluation des résidus, la sélection de variables et la quantification des incertitudes.

Une attention toute particulière est consacrée à l'algorithme *Extreme Learning Machine* (ELM). Ce dernier constitue en effet un réseau de neurones artificiels efficace et a récemment gagné une grande popularité dans le domaine des apprentissages automatiques. En profitant de sa rapidité d'application, un autre objectif de cette Thèse est d'extraire le potentiel d'ELM pour les tâches de sélection de variables et de quantification des incertitudes. Ces deux approches peuvent donner des informations essentielles sur la relation cachée entre les variables d'entrées et de sortie, lesquelles reflètent indirectement le comportement du phénomène étudié.

A cet égard, le thème principal de cette Thèse est axé sur le développement de méthodologies cohérentes pour l'analyse de phénomènes environnementaux avec l'aide des algorithmes d'apprentissage automatique. Une mise en application des méthodologies et méthodes développées dans cette recherche a été réalisée sur des données simulées ainsi que sur des données réelles de phénomènes tels que les feux de forêts, la pollution et les champs du vent.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Contributions of the Thesis 3

1.3.1 Chapter 2 3

1.3.2 Chapter 3 5

1.3.3 Other Contributions and Collaborations 5

1.3.4 Summary of the Contributions 7

1.4 State of the Art 7

1.5 Organisation of the Manuscript 9

2 Machine Learning of Environmental Data **11**

2.1 Methodology of Machine Learning Application in Environmental Data Modelling . . 11

2.2 Definitions and Notions 13

2.3 Parameters versus Hyper-parameters 14

2.4 True Validation versus Cross-validation 15

2.4.1 True Validation 15

2.4.2 Cross-validation 18

2.4.3 Remarks and Recommendations 19

2.5 Data Splitting 20

2.5.1 Random Sampling 21

2.5.2 Stratified Sampling 21

2.5.3 Assessment of the Subsets 22

2.5.4 Issues with Spatial Autocorrelation 23

2.6 Measures of Evaluation 24

2.7 Complexity Analysis 27

2.7.1 Hyper-parameter related Complexity 27

2.7.2 Extreme Learning Machine Complexity 28

2.8 Concluding Remarks 31

3 Extreme Learning Machine 33
3.1 Presentation and Theory 33
3.2 Feature Selection 34
3.2.1 Exhaustive search 35
3.2.2 Simulated Annealing based Feature Selection 38
3.2.3 Data and Results 41
3.3 Uncertainty 45
3.3.1 Introduction 46
3.3.2 General Framework 48
3.3.3 Methodology 50
3.3.4 Case Study and Results 51
3.3.5 Discussion 53
3.3.6 Concluding Remarks 53

4 Extreme Learning Machines for Spatial Environmental Data 57
4.1 Introduction 57
4.2 Method and Data 59
4.2.1 General ELM Theory 59
4.2.2 Dataset Description 60
4.3 Methodology 62
4.3.1 Data Preprocessing 62
4.3.2 Model Training 63
4.3.3 Model Evaluation 64
4.3.4 Learning the Residuals 64
4.3.5 Multivariate ELM 65
4.4 Results and Discussion 67
4.5 Conclusions 75

5 Wildfire Susceptibility Mapping: Deterministic vs. Stochastic Approaches 77
5.1 Introduction 77
5.2 Material: Study Area and Datasets 79
5.2.1 Study Area 79
5.2.2 The Datasets 80
Topography 81
Land Use and Vegetation Cover 81
The Fire Dataset 81
5.3 Methodology 82
5.3.1 Deterministic Method 83
5.3.2 Machine Learning Algorithms 84

Random Forest 84
Extreme Learning Machine 85
Parameters Optimisation 87
5.4 Results and Discussion 88
5.4.1 Major Variables affecting Wildfires Occurrence in Portugal 88
5.4.2 Susceptibility Maps 89
5.4.3 Methods' Assessment 91
5.5 Conclusion 94

6 Conclusion **97**
6.1 Conclusions and future directions 97
6.2 Contributions 99

A Proceeding **101**
A.1 Feature selection combining Simulated Annealing and Extreme Learning Machine . 101
A.1.1 Introduction 101
A.1.2 Method 102
A.1.3 Data and Results 104
A.1.4 Conclusion 105

Bibliography **107**

List of Figures

2.1 Presentation of the general methodology. Each link represents a dependence which should be taken into account during the process. 12
2.2 Example of a validation curve (in black) and of a training curve (in red) for different hyper-parameter values by using the extreme learning machine as a model and the mean squared error (MSE) as a measurement of error. 17
2.3 Example of boxplots for the quality control of the splitting procedure. Case study of wind field prediction in Switzerland. 22
2.4 Example of boxplots when the 1-D distribution is identical, and the 2-D distribution is different. 23
2.5 According to the data available, the complexity can be highlighted by the presence of linear or non-linear relationship between input and output variables. 28
2.6 Difference in MSE between linear and non-linear models by considering the set of variables $\{X1, X2, Y\}$. The green arrow shows the difference between MSE_l (the linear model computed on the validation subset in red dashed line) and MSE_{nl} (the non-linear model computed on the validation subset in red solid line) when they reach their minimum. 30
2.7 Visualization of the butterfly dataset $\{X1, X2, Y\}$ with the linear ELM model (on the left), and the non-linear ELM model (on the right). 31

3.1 Structure of Extreme Learning Machine (ELM) following a single-hidden layer feed-forward neural network (SLFN). 34
3.2 Study area with training (black dots) and testing (red triangles) data points. Colour scale displays the elevation in meters. 36
3.3 Wind speed predictions (m/s) for January with the best subset of features (variable X (1) and the medium DoG (5)). 38
3.4 State space with $r = 1$ for the neighbourhood definition, and 4 input variables. 41
3.5 Boxplots of the optimal number of nodes of ELM model for each data set 43
3.6 Each dashed line corresponds to one random subset of features. The graph shows the MSE of the ELM for these different subsets of features according to the number of hidden nodes (Ionosphere database). 45

3.7 Behaviour of the optimal number of nodes among different θ in Ionosphere database. Although different θ are used, the range of optimal number of nodes is between 15 and 30. 46

3.8 Behaviour of the cost function through one run of the SANELM algorithm (Ionosphere database). 47

3.9 Methodology of the ELM bootstrap-based uncertainty. The dataset is split into training (TRN) and testing (TST) subsets. Then, from the TRN subset, a step by step process is proposed in order to extract the different kind of variances. Finally, TST subset is used to validate the obtained result. 51

3.10 Original data with added noise (on the left), and mean prediction of the 100 ELM bootstrap models (on the right). Scale color on the left is proportional to the level of noise, while on the right it is proportional to the output variable. 52

3.11 Visualization of the variance of the model uncertainty (on the top left), the variance of the data noise (on the top right), and the variance of the total prediction errors (at the bottom). 54

3.12 Mean prediction of the 100 ELM bootstrap models on the top left. Variance of the model uncertainty on the top right. And variance of the data noise at the bottom. The maps are linearly projected into the square $[0,1]^2$. 55

3.13 Variance of the total prediction error on the left. Distribution curves based on the 100 ELM bootstrap models on the right. Vertical continuous lines are the predicted mean value of the 100 ELM models, and dotted lines are the true value. These three curves are based on three test points denoted on the map in black, red and blue. The left map is linearly projected into the square $[0,1]^2$. 56

4.1 Extreme Learning Machine Structure. 59

4.2 Lake Geneva with elevation map (m) and measurment points. 61

4.3 Scatterplot of heavy metal concentration (normalized). 61

4.4 ELM structure with two output variables. 66

4.5 Results of the 3D-Ni, 3DSH-Ni, 6D-Ni and 3D-Zn scenarios using a 5-fold cross-validation (20 runs). Black and red colors correspond to the training and validation sets respectively. Solid lines show the averages of 20 runs, and dashed lines the corresponding standard deviations. 68

4.6 Accuracy plot of the scenarios 3D-Ni (on the left) and 3DSH-Ni (on the right). Validation and training data are in black (circle) and testing data in red (triangle). 69

4.7 Cross-validation plot of ELM model for the residuals of the scenarios 3D-Ni (on the left) and 3DSH-Ni (on the right). Training data are in black and validation data in red. 70

4.8 Cross-validation curves of ELM Zn (triangles) and Ni (open circles) residuals using kNN algorithm. 70

4.9 Omnidirectional variography of raw data (triangles) and Ni ELM residuals (open circles). 71

4.10 Prediction of the Nickel (on the left) and Zinc (on the right) normalized pollutants. Validation and training data are in black and testing data in red. The maps are linearly projected into the square $[0,1]^2$. 71

4.11 Visualization of the normalized MSE according to the number of hidden nodes. Elements are: 1-Hg, 2-Zn, 3-Cu, 4-Ti, 5-Cr, 6-V, 7-Ni and 8-Cd. 72

4.12 Pairwise visualization of the normalized β-distance according to the number of hidden nodes. Index are described in table 4.4. 73

4.13 Evaluation of all possible models with ELM algorithm. Abscissa indicates the selected variables according to the following index: 1-X, 2-Y, 3-Z, 4-X sh., 5-Y sh. and 6-Z sh. 74

5.1 Location of Mainland Portugal and its 21 PROF regions protruding the study region of Dão-Lafões. 80

5.2 (a) Slope, (b) land cover according to Corine Land Cover 2006 inventory, and (c) fire frequency for the 2000-2013 period in the Portuguese PROF region of Dão-Lafões. . . 82

5.3 Annual burnt area polygons in the calibration (2000-2009) and validation (2010-2013) periods in the Portuguese PROF region of Dão-Lafões. 83

5.4 Structure of RF based on an ensemble of single decision trees. 85

5.5 Structure of the ELM following a single-hidden layer feedforward neural network (SLFN). In this configuration, slope and CLC classes are used as input variables, and SC_{ELM} stands for the susceptible value of the model. 86

5.6 Susceptibility maps for the three models based on the 5 generated classes, which are very high, high, medium, low and very low. 90

5.7 Differences of the susceptibility classes between the three methods. For each susceptibility class of Figure 5.6, a value is assigned (very low=0, low=1, medium=2, high=3 and very high=4) and the differences between classes for each model were computed. 90

5.8 Proportion of total BA explained by each of the three models in each susceptibility class. 92

5.9 Variable importance computed with random forest algorithm over 20 runs. The 9 top variables are displayed with the corresponding % increase of mean square error. . . . 93

A.1 Each dashed line correspond to one random subset of features and the solid line coincides with the best subset of features. The graph shows the MSE of the ELM for these different subsets of features according to the number of hidden nodes. 106

List of Tables

2.1 Example of model functions with their corresponding parameters and hyper-parameters. 15

2.2 Structure of a confusion matrix for two groups only. Here the right predictions are determined by the number in TP and TN, while wrong prediction are in FN and FP. . 26

2.3 Different scenarios are highlight according to the set of variables and the number of points used. The mean square error (MSE), the optimal number of node ($\tilde{N}$) as well as the I_C index are computed for each scenario. The optimal number of node in parenthesis denotes the optimal one found by visual evaluation. 32

3.1 Results of the feature selection. 37

3.2 Names of variables for each code number. DoG means Differences of Gaussians and DD means directional derivative 37

3.3 Database from the UCI repository used for the evaluation of SANELM 42

3.4 Results of the pre-analyis showing the optimal number of nodes $\tilde{N}$ and the temperature parameter T_0 42

3.5 Results of the SANELM algorithm with the number of hidden nodes used ($\tilde{N}$), the mean number of features selected (MNF), the mean error with all features (Err_{all}) and the mean error with the best subset of features (Err_{best}). Mean (and standard deviation) are evaluated over 50 runs 44

4.1 Composition of the different scenarios. 62

4.2 Results of the 3D-Ni, 3DSH-Ni, 6D-Ni and 3D-Zn scenarios. 67

4.3 Details on the residuals for scenarios 3D-Ni and 3DSH-Ni. 69

4.4 Details on the index used in figure 4.12. 72

4.5 10 best models according to the 6D-Ni scenario evaluated with ELM. 73

5.1 Total number of pixels for the entire study area (All dataset) with the corresponding proportion of pixel which burnt 0, 1, 2, 3, 4 and 5 times. The column *Testing and training subsets* displays the proportion which was used in order to generate the 20 training subsets and the testing subset. 87

5.2 Ratio between the size of each class and the proportion of BA for the testing period for the determinictic approach. 91

5.3 Ratio between the size of each class and the proportion of BA for the testing period for ELM and RF. 93

5.4 Variable importance for the top 6 feature for the random forest and the corresponding rank in deterministic model. Favorability scores computed in the deterministic model are used for ranking the variables. CLC classes are identified by the level code. . . . 94

List of Abbreviations

ACC	**ACC**uracy
BA	**B**urnt **A**rea
CIPEL	**C**omission for the **P**rotection of Lake Geneva
CLC	**C**orine **L**and **C**over
DD	**D**irectional **D**erivative
DEM	**D**igital **E**levation **M**odel
DoG	**D**ifference **o**f **G**aussians
EDA	**E**xploratory **D**ata **A**nalysis
ELM	**E**xtreme **L**earning **M**achine
ESDA	**E**xploratory **S**patial **D**ata **A**nalysis
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GIS	**G**eographic **I**nformation **S**ystems
GRNN	**G**eneral **R**egression **N**eural **N**etworks
k-NN	**k**-**N**earest **N**eighbours
MAE	**M**ean **A**bsolute **E**rror
MAPE	**M**ean **A**bsolute **P**ercentage **E**rror
MLP	**M**ulti**L**ayer **P**erceptron
MSE	**M**ean **S**quared **E**rror
NMBA	**N**ational **M**apping **B**urnt **A**reas
OP-ELM	**O**ptimally **P**runed **E**xtreme **L**earning **M**achine
PROF	**P**lanos **R**egionais de **O**rdenamento **F**lorestal
PSO	**P**article **S**warm **O**ptimization
RF	**R**andom **F**orest
RMSE	**R**oot **M**ean **S**quared **E**rror
SAN	**S**imulated **AN**neling
SANELM	**S**imulated **AN**neling with **E**xtreme **L**earning **M**achine
SLFN	**S**ingle-hidden **L**ayer **F**eedforward **N**eural network
SLT	**S**tatistical **L**earning **T**heory
SVD	**S**ingular **V**alue **D**ecomposition
SVM	**S**upport **V**ector **M**achine
TN	**T**rue **N**egative

TP	**T**rue **P**ositive
TRN	**TR**ai**N**ing subset
TST	**T**e**ST**ing subset
VAL	**VAL**idation subset
VALTRN	**VAL**idation and **TR**ai**N**ing subset

à Nadia

Chapter 1

Introduction

1.1 Motivation

Due to the large amount and complexity of data available nowadays in geo- and environmental sciences, we face the need to develop and incorporate more robust and efficient methods for their analysis, modelling and visualization. An important part of these developments deals with an elaboration and application of a contemporary and coherent methodology following the process from data collection to the justification and communication of the results. Recent fundamental progress in machine learning can considerably contribute to the development of the emerging field - *environmental data science*.

When speaking of environmental data science, it should be seen as a new emerging field, where several areas, like informatics, statistics, data management, geo- and environmental science (in a broad way) among others, try to merge and share knowledge in order to solve, model, understand or visualize the increasing amount of data available nowadays. It is important to understand that it is, for the time being, too early to clearly define this emerging field, where, for sure, application of machine learning algorithms to environmental data is part of it.

Machine learning algorithms, considered in a wide sense, are used in many different research fields and applications: data mining and knowledge discovery, biocomputing, business and finance, text mining, socio-economic data mining and many others. Recently they gained a great popularity in geoscience and environmental science.

Environmental data share different properties which are important for analysis and modelling, and which should be taken into account during the machine learning process. In particular, they can be clustered in space and can show a high variability at several spatio-temporal scales (e.g. extreme values, outliers). They are complex and most of the time multivariate, what makes the analysis and the prediction difficult. Finally, they are not homogeneous (i.e. datasets can be composed of a mixture of categorical and continuous variables) and the phenomenon under study and the related data are not linear.

For this reason, the use of machine learning algorithms for environmental applications can efficiently answer some of the fundamental problems mentioned above. This can be explained by the fact that machine learning algorithms are known to be universal and non-linear. Moreover, as a data driven based method, they are able to adapt to any phenomenon. In principle, these algorithms have the capacity to extract information from any dataset with any desired precision, and can efficiently work in high dimensional spaces. Namely, machine learning algorithms can support environmental sciences in solving basic and complex problems of learning from data (i.e. classification or regression modelling).

During the last years, many environmental phenomena were analysed and modelled using machine learning algorithms, but in most cases the coherence of the methodology or the reproducibility of the results were lacking. Moreover, in many applications (of such methods), important questions were not considered. For example, the quantification of the uncertainties from raw data to the results, the analysis of multivariate and dependent variables for modelling purposes, the validation of the training process, and the coherence of the validation and the testing results.

In this regard, the present research highlights and proposes coherent methodologies for problems of analysis, modelling and prediction of environmental data.

1.2 Objectives

The main purpose of this thesis, in a broad way, is to develop coherent and self-consistent methodologies for the analysis of environmental phenomena using machine learning algorithms. In particular, this thesis gives an overview of machine learning algorithms for environmental data mining. It highlights and investigates the different issues that can occur when dealing with complex and high dimensional environmental data using cutting-edge machine learning algorithms. In addition, several important topics of data driven modelling, including data splitting, complexity analysis, residual assessment, feature selection and uncertainties are discussed.

More precisely, a special attention was paid to the definition of a model in the framework of machine learning. From this definition and the resulting notation, one of the objectives of this thesis is to propose a consistent reflection about the process of model selection and model evaluation. In this regard, differentiation between the model parameters and the model hyper-parameters was proposed. Explanations of the different methods for the model validation, the data splitting and the measurement of evaluation was highlighted.

In many cases, the machine learning algorithm can be seen as a black box, in which data are inserted, parameters are tuned, and results are given with a prediction map and a number which tries to reflect the error of the model. According to this aspect, a major objective of this thesis is to break the black box idea of machine learning algorithms. For this purpose complexity analysis and residuals assessment were presented in order to support the understanding of machine learning results and, indirectly, the comprehension of the phenomenon under study.

Finally, a special attention was paid to the Extreme Learning Machine algorithm (ELM). Being an efficient artificial neural network, it gained recently a great popularity in the domain of machine learning. By taking advantage of the quickness of ELM process, another objective of this thesis is to extract the potential of ELM for the tasks of feature selection and uncertainty quantification. Both of these approaches can provide valuable information about the possible hidden relationships between input and output variables (which can indirectly reflect the behaviour of the phenomenon studied).

Consequently, the general leitmotif of this thesis is the development of coherent methodologies for the analysis of environmental phenomena using machine learning algorithms, and their application to simulated and real data.

1.3 Contributions of the Thesis

In the following sections, the different contributions of the thesis are presented along with their related publications or conference researches. It is worth mentioning that several publications or conference researches are present in several chapters. It is essentially due to the fact that the different steps highlighted in chapter 2 were used in the development of ELM-based methods in chapter 3.

1.3.1 Chapter 2

This chapter proposes the general process of an application of machine learning algorithms for environmental data mining. For this purpose, mathematical formalism was adopted in order to introduce the definitions and the basic notions (section 2.2). Then, a development of this formalism was extended to the description of parameters vs. hyper-parameters (section 2.3), the cross-validation vs. the true validation definitions (section 2.4), and for the measures of evaluation (section 2.6). It is worth noting that for the corresponding sections, there are no specific publications or conference researches dealing with these topics. This is essentially due to the fact that all researches done in the field of environmental data mining using machine learning algorithms should deal somehow with these topics. In other words, it means that the different concepts of parameters optimization, validation and measures of the errors should be taken into consideration in all researches. For this reason, all the researches (publications and conferences) carried out during this thesis contribute in a certain way to the development of the mentioned sections.

Regarding the data splitting methods presented in section 2.5, several publications and conference researches contributed to this domain. In particular, a comparison of two sampling strategies was performed in order to counter the effect of spatial auto-correlation in the case of landslide susceptibility maps (Micheletti et al., 2014). Then, the assessment of the generated subset was proposed in Leuenberger and Kanevski (2015) and then applied in different studies (e.g. wind fields, lake pollutant, forest fires). Finally, in Leuenberger et al. (2018), the use of stratified sampling for complex and non-homogeneous datasets was tested for a case study of forest fires in Portugal.

The complexity analysis presented in section 2.7 exposes two different ways of quantifying the hidden relationship complexity between input and output variables. In particular, a new approach based on ELM was proposed and presented at the European Geosciences Union 2017.

The referred works were presented and published as follows:

- **M. Leuenberger** and M. Kanevski, Extreme learning of environmental pollution, In *11th Swiss Geoscience Meeting, Lausanne (Switzerland)*, 2013
- N. Micheletti, L. Foresti, S. Robert, **M. Leuenberger**, A. Pedrazzini, M. Jaboyedoff and M. Kanevski, Machine learning feature selection methods for landslide susceptibility mapping, *Mathematical Geosciences*, 46 (1), 33-57, 2014
- **M. Leuenberger** and M. Kanevski, Overview of machine learning applications in environmental data mining, In *Data analysis and modelling in Earth sciences, Milan (Italy)*, 2014
- M. Kanevski and **M. Leuenberger**, Environmental Data Modelling Using Extreme Learning Machines, In *10th International Conference on Geostatistics for Environmental Application, Paris (France)*, 2014
- **M. Leuenberger** and M. Kanevski, Extreme Learning Machines for spatial environmental data, *Computers and Geosciences*, 85, 64-73, 2015
- **M. Leuenberger** and M. Kanevski, Recent advances in environmental data mining, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 18, page 6137, 2016
- M. Pereira, **M. Leuenberger**, J. Parente and M. Tonini, wildfire susceptibility mappings: comparing deterministic and stochastic approaches, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 18, page 7395, 2016
- **M. Leuenberger** and M. Kanevski, Feature Selection and Modelling with Extreme Learning Machine. Case study: Wind Fields in Complex Regions, In *11th International Conference on Geostatistics for Environmental Application, Lisbon (Portugal)*, 2016
- **M. Leuenberger**, J. Parente, M. Tonini, M. Pereira and M. Kanevski, Wildfire susceptibility: Comparing deterministic approach with machine learning, In *14th Swiss Geoscience Meeting, Geneva (Switzerland)*, 2016
- **M. Leuenberger** and M. Kanevski, Study of Environmental Data Complexity using Extreme Learning Machine, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 19, page 14015, 2017
- **M. Leuenberger**, J. Parente, M. Tonini, M.G. Pereira and M. Kanevski, Wildfire susceptibility mapping: deterministic vs. stochastic approaches, *submitted to Environmental Modelling and Software*, 2017

1.3.2 Chapter 3

In this chapter the main attention is focused on the Extreme Learning Machine algorithm. By using the properties of ELM, feature selection task, which consists of selecting the most suitable variables for a particular problem, was considered (section 3.2). In particular, the use of exhaustive search and simulated annealing as heuristic model was proposed. The major contributions of this research were published in Leuenberger and Kanevski (2014) and Leuenberger and Kanevski (2015) and presented at various conferences.

The Uncertainty analysis with extreme learning machine was first presented at the conference of the European Geosciences Union (EGU), 2015. It was adapted then for the International Association for Mathematical Geosciences (IAMG), 2015, and finally improved in section 3.3. This contribution allows a better understanding of the limits of the model and the dataset by identifying and quantifying the uncertainties.

The referred works were presented and published as follows:

- **M. Leuenberger** and M. Kanevski, Multivariate Mapping of Environmental Data Using Extreme Learning Machines, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 16, page 4206, 2014

- **M. Leuenberger** and M. Kanevski, Feature selection in environmental data mining combining Simulated Annealing and Extreme Learning Machine, *Proceedings, European Symposium on Artificial Neural Networks*, 22, 601-606, 2014

- **M. Leuenberger** and M. Kanevski, Extreme Learning Machines for spatial environmental data, *Computers and Geosciences*, 85, 64-73, 2015

- **M. Leuenberger** and M. Kanevski, Mapping of Estimations and Prediction Intervals Using Extreme Learning Machines, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 17, page 6127, 2015

- **M. Leuenberger** and M. Kanevski, Decision-Oriented Mapping Using Extreme Learning Machines,*Proceedings of the 17th annual conference of the International Association for Mathematical Geosciences*, 597-601, 2015

- **M. Leuenberger** and M. Kanevski, Feature Selection and Modelling with Extreme Learning Machine. Case study: Wind Fields in Complex Regions, In *11th International Conference on Geostatistics for Environmental Application, Lisbon (Portugal)*, 2016

1.3.3 Other Contributions and Collaborations

During the thesis, several collaborations have been made in different fields (permafrost, forest fire, chemical emissions, and fractal-based theory), but always with a connection to machine learning.

Although they are not mentioned in previous chapters, they greatly contributed in the development and the reflection of how to construct a coherent methodology around machine learning algorithm and environmental data.

The referred works were presented and published as follows:

- C. D. Vega Orozco, **M. Leuenberger**, M. Tonini and M. Kanevski, Anthropogenic forest fires susceptibility mapping using Random Forest algorithm, In *International conference on forest fire risk modelling and mapping, Aix en Provence (France)*, 2013

- **M. Leuenberger**, M. Kanevski and C. D. Vega Orozco, Forest Fires in a Random Forest, In *European Geosciences Union General Assembly, Copernicus Publication, Vienna (Austria)*, Vol. 15, page 3238, 2013

- **M. Leuenberger**, M. Kanevski and N. Deluigi, Permafrost in a Random Forest, In *15th conference of International Association for Mathematical Geosciences, Madrid (Spain)*, 2013

- N. Deluigi, **M. Leuenberger**, M. Kanevski and C. Lambiel, Alpine permafrost data analysis and mapping with Support Vector Machines, In *15th conference of International Association for Mathematical Geosciences, Madrid (Spain)*, 2013

- **M. Leuenberger**, C. D. Vega Orozco, M. Tonini and M. Kanevski, Random Forest for susceptibility mapping of natural hazards, In *11th Swiss Geoscience Meeting, Lausanne (Switzerland)*, 2013

- J. Golay, M. Kanevski, C. D. Vega Orozco and **M. Leuenberger**, The multipoint Morisita index for the analysis of spatial patterns, *Physica A: Statistical Mechanics and its Applications*, 406, 191-202, 2014

- **M. Leuenberger** and M. Kanevski, Application of Random Forest Algorithm for Environmental Data, In *Data analysis and modelling in Earth sciences, Milan (Italy)*, 2014

- J. Golay, **M. Leuenberger** and M. Kanevski, Morisita-based feature selection for regression problems, *Proceedings, European Symposium on Artificial Neural Networks*, 23, 279-284, 2015

- A. Marvuglia, **M. Leuenberger**, M. Kanevski and E. Benetto, Random forest for toxicity of chemical emissions: features selection and uncertainty quantification, *Journal of Environmental Accounting and Management*, 3 (3), 229-241, 2015

- M. Conedera, M. Tonini, L. Oleggini, C. D. Vega Orozco and **M. Leuenberger**, Geospatial approach for defining the Wildland-Urban Interface in Alpine environment, *Computers, Environment and Urban Systems*, 52, 10-20, 2015

- J. Golay, **M. Leuenberger** and M. Kanevski, Feature Selection for Regression Problems Based on the Morisita Estimator of Intrinsic Dimension, *Pattern Recognition*, 70, 126-138, 2017

1.3.4 Summary of the Contributions

Within the framework of the proposed methodology and its application to environmental data, the main contributions of this thesis can be stated as follows:

- Development of a consistent and coherent methodology for machine learning algorithms applied to environmental data.
- Elaboration of different methods for complexity analysis and residuals assessment tasks.
- Presentation and development of new analytic tools based on Extreme Learning Machine, such as: multivariate ELM, feature selection, and uncertainty.
- Application of the proposed methodology to various environmental phenomena, such as: landslides, pollutions, chemical emissions, forest fires, permafrost and wind fields.

1.4 State of the Art

The literature on machine learning algorithms is extremely rich and covers many topics at different levels. It can be ranged from textbooks to advanced presentations which include theories, algorithms and programming details. Nevertheless, the kernel based methods from the statistical learning theory (SLT) continue to dominate in both the application and the development (Vapnik, 1998; Shawe-Taylor and Cristianini, 2004; Cherkassky and Mulier, 2007; Hastie, Tibshirani, and Friedman, 2009). They affect almost all scientific disciplines and especially those which heavily relied on data analysis. In the framework of statistical learning theory, two algorithms, which are Support Vector Machines and Support Vector Regression, became common tools for the analysis and modelling of complex data in high dimensional spaces. At present, the "family" of SLT methods includes non-linear extensions of principal component analysis, several supervised and unsupervised dimensionality reduction techniques, semi-supervised learning, etc.

Regarding more classical machine learning algorithms for data analysis in a broad sense, we can mention among others: multilayer perceptrons (Rosenblatt, 1961), general regression and probabilistic neural networks (Specht, 1991), self-organizing maps (Kohonen, 2001), decision trees (Breiman et al., 1984), Bayesian networks (Pearl, 1985), Gaussian processes (Bishop, 2006), etc. Some of them form the basis of a new trend, called "visual analytics". This aspect gained a great interest in environmental science. It is due to the fact that in reality, many environmental phenomena should be considered in a higher dimensional feature spaces, and not only in the two-dimensional geographical coordinate system (Kanevski and Maignan, 2004). Therefore, visual analytics are becoming a natural tool to explore and understand the related data. In that sense, the combination of machine learning algorithms and intelligent visualization of data can be considered as visual data mining (Anderson, 2013).

In the feature selection domain, it is well known that the selection of good predictors is more important (in many cases) than the modelling tool. For example, in time series, a good selection of the time delay and the number of delays is of great importance for the modelling part. In that sense, appropriate inputs with even a simple model can produce better results than a complex and non-linear model which considers all the features. The same is valid for spatial data. Although simple problems can only consider geographical space (e.g. for interpolation purpose), most of the real environmental problems require the construction of complex input feature spaces. As an example, the modelling of monthly wind fields in complex region should be carried out in a high dimensional space, which includes geographical coordinates, digital elevation model and its derivatives (e.g. slope, curvature, etc. Robert, Foresti, and Kanevski (2012)). The same is even more important for natural hazards problems, which should take into account land use, geology and many other factors (Brenning, 2005; Micheletti et al., 2014). Usually, the construction of the input space is performed with the experts in the domain (e.g. geologists, risk analysts, environmentalists, etc.). They can propose a collection of potentially relevant variables, but as their number increases, the problem becomes quite difficult. In other words, for a fix number of data points, as the number of variable increases (i.e., the dimensionality of the input space), the data points tend to be "isolated", which modify our representation of the distances in this high dimensional space. Also called " curse of dimensionality", it can be a real issue for all distance-based methods (Guyon et al., 2006; Hastie, Tibshirani, and Friedman, 2009; Lee and Verleysen, 2007). For this reason, the reduction of the dimensionality of the input space is of great importance. In most publications on machine learning applications to environmental data, this problem is very rarely considered. Nevertheless, different feature selection algorithms can be well adapted to environmental risks and natural hazards, such as: multiple kernel learning (Rakotomamonjy et al., 2008), adaptive general regression neural networks, adaptive probabilistic neural networks (Specht and Romsdahl, 1994; Gheyas and Smith, 2010; Robert, Foresti, and Kanevski, 2012), and random forest (Breiman, 2001; Amatulli, Camia, and San-Miguel-Ayanz, 2013).

An important question of intelligent data analysis and modelling deals with the treatment of uncertainties. In real decision making process, the uncertainties (i.e. confidence and prediction error bars) around unknown values often play even more important role than the predictions themselves. There are different sources of uncertainties (e.g. model uncertainties, uncertainties in the parameters of the model, or data uncertainties). Many of them can be reduced by improving the modelling and the calibration process. The approaches to estimate the uncertainties are specific to the accepted assumptions, methods used, and to the type of data (i.e., according to the addressed topic). For example, probabilistic models provide inherent treatment of uncertainties like the variance of predictive distribution (Hastie, Tibshirani, and Friedman, 2009; Murphy, 2012; Bishop, 2006). Many traditional regression models of machine learning, such as neural networks, do not provide directly such outputs and further efforts are required to obtain the estimated uncertainty. On the other hand, the decision function of support vector machine can be transformed into probabilities, which is very important for natural hazard analysis (Platt, 2000; Pozdnoukhov et al., 2011). This method is usually used for susceptibility mapping. Recently, an interesting approach for uncertainties modelling was proposed in Shrestha, Kayastha, and Solomatine (2009), where a general framework was proposed in the field of hydro-informatics (Abrahart, Kneale, and See, 2004).

It should be noted that the applications of machine learning algorithms cover a wide range of environmental topics, which are: air, water and soil pollutions (Dubois, 2005; Kanevski and Maignan, 2004; Nagendra and Khare, 2005; Cervone et al., 2008; Pasero and Mesin, 2010; Hassan and Li, 2010; Bnanankhah and Nejadkoorki, 2012); earth and environmental sciences, including natural hazards analysis (Abrahart, Kneale, and See, 2004; Amatulli, Camia, and San-Miguel-Ayanz, 2013; Brenning, 2005; Cheng and Wang, 2008; Cherkassky et al., 2006; Gardnera and Dorlinga, 1998; Haupt, Pasini, and Marzban, 2009; Hsieh, 2009; Krasnopolsky and Lin, 2012; Pradhan, 2013); renewable resources assessments (Marvuglia and Messineo, 2012; Robert, Foresti, and Kanevski, 2012; Xu et al., 2012).

Unfortunately, many environmental data studies using machine learning algorithms were carried out without a deep understanding of the non-linear modelling part, but rather like black-boxes, which try to find some relationships between input and output variables. As a consequence, the justification and interpretability of the results are not clear. Therefore, there is a room for developing a coherent and self-consistent methodology for machine learning algorithms in environmental sciences.

1.5 Organisation of the Manuscript

This thesis is divided into two parts. The first part is composed of chapters 2 and 3. Chapter 2 presents the general methodology for model selection and model evaluation in machine learning. In particular, the notions of parameters, hyper-parameters, cross-validation, data splitting among others are presented. In chapter 3 the theoretical part of extreme learning machine with the development of feature selection and uncertainty quantification are presented.

The second part, composed of chapters 4 and 5, presents two articles (Leuenberger and Kanevski, 2015; Leuenberger et al., 2018). Being key publications for the methodological aspect, they are of great importance in this thesis. In particular, main attention was paid on the application of the proposed methods on real datasets such as pollutions and forest fires in complex and high-dimensional spaces.

The first article *Extreme Learning Machine for Spatial Environmental Data* (Leuenberger and Kanevski, 2015) highlights a methodology for the application of Extreme Learning Machine on environmental data (chapter 4). In particular, analysis of the residuals and results on multivariate ELM are presented and discussed. In chapter 5, the second article *Wildfire Susceptibility Mapping: Deterministic vs. Stochastic Approaches* (Leuenberger et al., 2018) shows a comparison between a standard method for wildfire susceptibility mapping and two machine learning algorithms (i.e., extreme learning machine and random forest). A special attention was paid on the description of the methodology for both the application of each model, and the comparison purposes.

Finally, chapter 6 concludes the thesis, appendix A highlights a proceeding which represents a special case of section 3.2, and bibliography completes the recent researches and studies in the field of machine learning for environmental data mining.

Chapter 2

Machine Learning of Environmental Data

This chapter focuses on all aspects that can affect the performance of machine learning algorithms in environmental data mining. Mainly, special attention is paid to the definitions and the distinction between parameters and hyper-parameters. Then, the data splitting process and the choice of validation or cross-validation are discussed and compared. Finally, specific problems related to spatial environmental data, mainly the issues with spatial autocorrelation, are highlighted and explained with recommendations.

2.1 Methodology of Machine Learning Application in Environmental Data Modelling

When dealing with environmental data, a lot of methods are available in the literature of machine learning algorithms. In particular, several programs, libraries, plugins or home made scripts (or codes) are accessible, which gives to the practitioner a higher freedom. Although the use of the proposed default parameters provides always a result (either good or bad), the application of such algorithms is not straightforward, in particular when the practitioner need to understand and validate the obtained result. For this reason, a special attention is paid, in this thesis, to the elaboration of a consistent and coherent methodology for the application of machine learning algorithms to environmental data.

In this regard, a generic methodology is summarized and presented in the flowchart of figure 2.1. The main points are as follows:

- From the available data, and according to the main goal of the research (which should be very clear), the validity domain should be defined. With the help of expert knowledge, the validity domain should take into account the variables (also called features) of interest, the geographical space, or more widely the high dimensional space where the phenomenon under study resides.

- By taking into account the output variable (i.e., the measured variable, or the variable of interest related to the phenomenon), several pre-processing steps can be performed. One of them focuses on the detection of pattern. It consists of analysing the output variable in order to detect whether or not information are present in the data (i.e., determine if the output variable is just

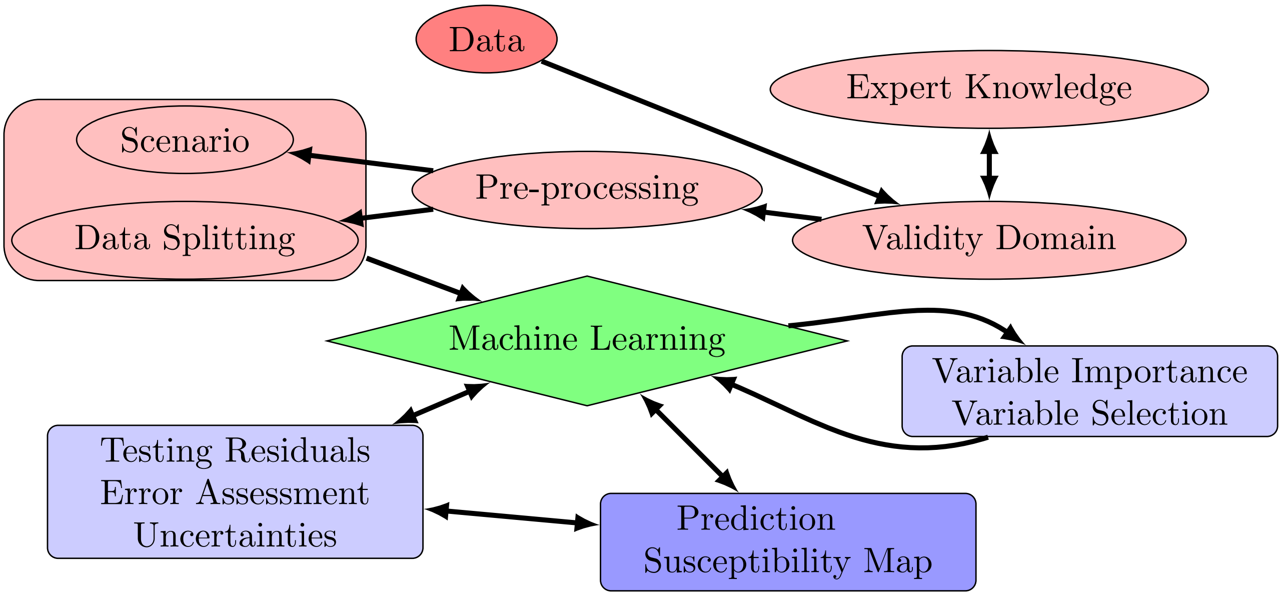


FIGURE 2.1: Presentation of the general methodology. Each link represents a dependence which should be taken into account during the process.

composed of noise ore not) by using permutation tests or clustering analysis. Another aspect of the pre-processing resides in the normalization procedure. Depending on the choice of the machine learning algorithm used, the normalization of the data should be taken into account in order to fit the "range of application". This range of application can vary according to the selected algorithm, but most of the time it will consist on normalizing all variables between $[0,1]$. By considering different kind of analytical and visualization techniques, exploratory data analysis (EDA) can provide valuable information about the nature of the data, and can help for the processing and the understanding of the results.

- Then, different scenarios can be generated. According to the nature of the data and the goal of the research, these scenarios should help to understand and validate (or not) the different assumptions about the phenomenon. At this stage, the data splitting procedure should be considered and adapted to the objectives and selected scenarios.

- Only after these considerations, a machine learning algorithms can be applied (i.e., train, validate and test a model, see the next sections for more details).

- In addition to the standard process, different tasks can be performed. For example, feature selection (also called variable selection or variable importance), which try to select the optimal subset of variables, can be applied independently or with the help of the machine learning algorithm. Results of the feature selection present a great interest for the understanding of the studied phenomenon, and most of the time, by decreasing the dataset dimensionality, can improve model accuracy.

- Before the production of the final prediction map, others post-processing task can be performed. Among them, the residuals assessment consists of testing the presence of remaining information in the residuals. It can give a good indicator about the performance of the model and the quality of the results. Then, the general error can be computed and compared among the different scenarios. And finally, an estimation of the uncertainties can be performed. All these "by-products", although they can differ from the main goal of the research, can provide valuable information about the hidden mechanism of the phenomenon under study.

- Finally, by selecting the best model (according to the practitioner criteria and the preliminary results), predictions and susceptibility maps can be performed on new data. It is important to note that the obtained results will depend on the whole process, and especially of the selected validity domain.

One important aspect of the whole process is the visualization. The visualization of the raw data, the preliminary results, and then the final result can help to understand the phenomenon under study, but also to validate and avoid errors. For this reason, whenever possible visualization of the proposed procedure steps is recommended.

The next sections of the chapter will focus on the different aspects linked to the model selection and model evaluation in machine learning.

2.2 Definitions and Notions

It is worth to mention the definition and the meaning of *model selection* and *model evaluation*, not only in the environmental data field but also in general terms.

Let us consider the typical problem of finding the hidden relationship between a set of input variables $X = (X_1, X_2, ..., X_d)$ and the output variable Y, and let $(\mathbf{x}_i, y_i)_{i=1,...,n}$ be n data points, where $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^d)^T \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The general machine learning problem aims to find an estimator of the function f which connects X to Y:

$$f(X) = Y. \tag{2.1}$$

In this context a standard definition of a *model* can be generalized with the following three levels structure:

$$f(\tilde{X}, \theta) = Y, \tag{2.2}$$

where $f \in \mathfrak{F}$ is a function among the set of all possible algorithms $\mathfrak{F}$ (e.g., Multilayer Perceptron - MLP, Extreme Learning Machine - ELM, Random Forest - RF, Support Vector Machine - SVM, General Regression Neural Networks - GRNN,...), $\tilde{X} \in \mathfrak{X}$ denotes a subset among all possible combinations of variables (called $\mathfrak{X}$) from the set X, and $\theta \in \Theta_f$ represents the selected parameters among the family of parameters of the function f.

For a particular real case study, the first choice would be to construct the set of function $\mathfrak{F}$ with which the study would be conducted. Then, the practitioner should determine whether he performs a feature selection task or not. This choice will determine the nature of $\mathfrak{X}$ (the set of subsets of X). At the end, for a determined function f and a fixed subset $\tilde{X}$, an optimization procedure should be used in order to reach the optimal value of $\theta^* \in \Theta$ for the current model:

$$\theta^* = \arg\min_{\theta \in \Theta} L(f(\tilde{X}, \theta), Y), \tag{2.3}$$

where the L function symbolizes a particular cost function (see section 2.6) which will evaluate the current model according to the true value Y. This last procedure will be closely dependant on the choice of a measure L for the evaluation purpose.

According to this notation, the forthcoming sections address the connected issues of *model selection* and *model evaluation*.

2.3 Parameters versus Hyper-parameters

The distinction between parameters and hyper-parameters is always ambiguous. In a broad way, parameters are optimized by internal mechanisms of each algorithms. On the other hand, hyper-parameters are optimized by the user. Nevertheless, heuristic methods can be applied in order to find the optimal hyper-parameters (see later in the cross-validation section). By keeping the notation of section 2.2, the distinction between the two kinds of parameters can be written as follows:

$$\theta = (\theta_p, \theta_{hp}) \in \Theta, \tag{2.4}$$

where θ_p and θ_{hp} represent respectively the parameter and hyper-parameter parts of the model f.

Let us consider the example of polynomial functions as a model (this example will be also used in the forthcoming sections). For a considered problem, where the main goal is to find the hidden relationship between an input variable $x \in \mathbb{R}$ and an output variable $y \in \mathbb{R}$, a standard polynomial regression model of degree k can be written as follows:

$$f(x) = a_k x^k + a_{k-1} x^{k-1} + ... + a_2 x^2 + a_1 x + a_0 = \sum_{i=0}^{k} a_i x^i. \tag{2.5}$$

In this setting, $a_0, a_1, ..., a_k$ denote the parameters of the model, while k represents the hyper-parameter (i.e. $\theta_p = \{a_0, a_1, ..., a_k\}$ and $\theta_{hp} = k$).

Models	$f(x)$	θ_p	θ_{hp}
Polynomial Function	$\sum_{i=0}^{k} a_i x^i$	$\{a_0, a_1, ..., a_k\}$	k
Extreme Learning Machine (ELM)	$\sum_{i=1}^{\tilde{N}} \beta_i g(x \cdot w_i + b_i)$	$\{\beta_i, w_i, b_i\}_{i=1,...,\tilde{N}}$	$\tilde{N}$
Support Vector Machine (SVM)	$\sum_{i=1}^{N} y_i \alpha_i K(x_i, x) + b$	$\{\alpha_i, b\}$	$\{\sigma, \mathrm{C}\}$
Random Forest (RF)	$\frac{1}{\text{nbtree}} \sum_{i=1}^{\text{nbtree}} \hat{y}_i$	thresholds	$\{\text{nbtree}, \text{nbtry}\}$

TABLE 2.1: Example of model functions with their corresponding parameters and hyper-parameters.

It is worth mentioning that different aspects of the parameters can vary according to the value of the hyper-parameters. For example, for the polynomial model, when the hyper-parameter k increases, the number of parameters increases as well. Moreover, when the hyper-parameter takes a new value the dependent parameters need to be retuned. For this reason, it is always better to optimize both parameters and hyper-parameters in an iterative way (because of the dependence between hyper-parameters and parameters). This aspect will be presented in the next section 2.4 where different ways of optimization are presented.

In table 2.1 a selection of model with its parameters and hyper-parameters are displayed. While the general decision function f shows similar aspects (e.g. ELM and MLP have exactly the same decision function), the main difference between algorithms are located in the parameters and the hyper-parameters, and in the way each algorithm will optimize them.

For this purpose, different subsets of the original dataset should be generated. Ideally, one subset should be assigned to optimize the parameters θ_p (known as the training procedure), another one for the hyper-parameters optimization θ_{hp} (denoted as the validation process), and finally a third subset for quantifying the generalization ability of the final optimal model.

In section 2.4 different ways of optimizing the hyper-parameters θ_{hp} are exposed, and section 2.5 introduces several methods of data splitting.

2.4 True Validation versus Cross-validation

In this section, the main objective is to explore the general framework for training, validating and testing a model. As mentioned in section 2.3, each step of the process will focus on a part of the optimization. In particular, each parameter (i.e. θ_p and θ_{hp}) should have its own optimization steps. In addition to optimizing all parameters, special attention should be paid to the estimation of the generalization error of the final optimal model.

2.4.1 True Validation

First of all, let us consider the most standard and "text book" example, which consists of having a dataset with enough data points, an homogeneous distribution over each input and output variables, and with absolutely no a priori knowledge about the phenomenon and the relationship between input and output variables. Let also consider that the optimization is only on the parameters θ of a defined model f and not on the subset of features $\tilde{X} \in \mathfrak{X}$. In this context, the traditional way to optimize a model with its corresponding parameters $\theta = (\theta_p, \theta_{hp})$ is to generate three distinct subsets (training - TRN, validation - VAL and testing - TST). As already mentioned in section 2.3, the training set will be used by the algorithm in order to optimize each parameter in θ_p. Then, the validation set will focus on the calibration of the hyper-parameters θ_{hp}. And finally, the testing set will be used for computing the generalization error of the final optimal model.

In practice, the application of a standard true validation method (also called holdout method (Hastie, Tibshirani, and Friedman, 2009)) for training, validation and testing is as follows:

Step 1: Randomly generate three subsets with 50% of the data for the training subset, 25% for the validation subset and 25% for the testing subset.

Step 2: According to the number of hyper-parameters θ_{hp} for a considered model f, fix the values of the hyper-parameters which represent the lowest degree of complexity of the model f. For example, in the case of a polynomial regression function (see equation 2.5), the value of the hyper-parameter with the lowest complexity is $k = 0$.

Step 3: Train the model (in other words, find the optimal parameter $\theta_p^*(\theta_{hp})$ which depends on the hyper-parameter) with the training dataset and with the fixed hyper-parameters θ_{hp}.

Step 4: Evaluate the present model $f(\theta_p^*(\theta_{hp}))$ with the validation subset. For this purpose, use one of the measurement error highlighted in section 2.6.

Step 5: Repeat step 3 and 4 by iteratively changing the value of the hyper-parameters θ_{hp} from the lowest to the highest degree of complexity. For the example of polynomial functions, this means that step 3 and 4 are repeated with $k = 1, 2, 3, \ldots$.

At the end, it is recommended to stop the iteration part of step 5 when the measurement error of step 4 reaches a minimum. In practice, it is not always simple to stop the iteration at the location of the minimum. This is essentially due to the fact that some models can have stochastic components (like extreme learning machine or random forest algorithms, but not like polynomial function or k-nearest neighbour) and in these cases the measurement error can fluctuate around the minimum value with more or less variability. For this reason, it is better to fix the general range where the hyper-parameters θ_{hp} will evolve before the procedure. For example, for the case of a polynomial model, a range from 0 to 20 (for the hyper-parameter k) can be defined. Then the five step of the procedure can be run and finally a plot showing the relationship between the different hyper-parameter values and the measurement error can be generated. If no minimum is detected on the plot, an adaptation of the range can be extended in order to explore more complex models.

In figure 2.2 an example of such a plot is shown for the case where extreme learning machine (see section 3.1) and mean squared error are used. In this case, a practitioner will select a hyper-parameter

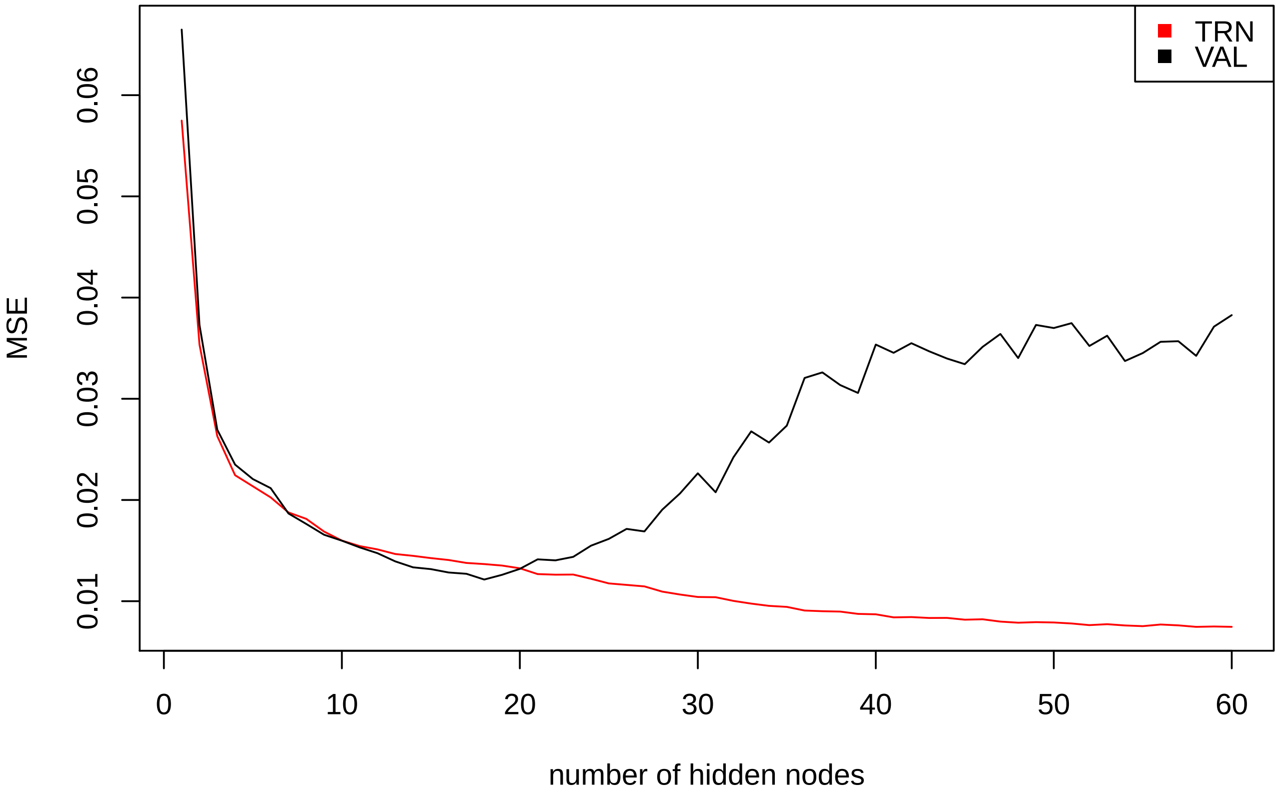


FIGURE 2.2: Example of a validation curve (in black) and of a training curve (in red) for different hyper-parameter values by using the extreme learning machine as a model and the mean squared error (MSE) as a measurement of error.

(for this algorithm it is the number of hidden nodes) with a value around 18, which is the value of the hyper-parameter θ_{hp} with the lowest error for the validation subset. Notice that sometimes the validation curves (here in black) can be below the training curve (here in red). This is essentially due to the fact that, ELM being a stochastic algorithm with a higher degree of variability than other machine learning algorithms, it can happen that the validation curve, for some hyper-parameters, is below the training curve. The final step resides in selecting and generating the best model and to apply this model on the testing data. This last test will provide the generalisation error for any new predicted data point.

It is worth mentioning that the numbers presented in step 1 are recommendations for a standard dataset. In some cases the subdivision could be $(60,20,20)$, or even $(80,10,10)$. The only recommendation is to have enough data points in each subset, in order to be able to characterize the phenomenon under study. Moreover, step 3 is generally proper for each machine learning algorithm.

The advantages of this method are diverse. First of all, the application of this procedure is quite simple. The generation of each subset is done randomly once. It can easily be implemented in various programming languages and does not require intense iterative or merging process. On the other hand, the true validation method, as presented here, is very sensitive to the number and the quality of the data available. In the cases where very few data points are available (less than 100), it is recommended to perform a cross-validation method (see next section 2.4.2). Another drawback can occur when the spatial distribution of the data points is not homogeneous over the study area. In this context, the use of the true validation method can lead to inappropriate parameters and a suboptimal model. This last point can be explained by the fact that when clusters are present in the data, the generated training, validation and testing subsets can show quite different spatial distribution. In that sense, the validation or testing subset can differ from the training subset and not completely represent the phenomenon. More aspects on this topic are discussed in section 2.5.4.

2.4.2 Cross-validation

As mentioned in section 2.4.1, there are some cases where the use of true validation is not feasible and requires instead the use of cross-validation methods. It can happen when very few data points are available but not only (see section 2.4.3 for more details).

Cross-validation (or more particularly K-fold cross-validation) aims to partition the entire dataset into K subsets in order to generate several training and validation subsets. More precisely, here are the mains steps of the K-fold cross-validation procedure:

Step 1: Randomly generate two subsets. The VALTRN set with 70% of the data available and the TST set with the remaining 30% data points. As for true validation, the percentage of the subsets should be selected in order to keep enough data points to "represent" the phenomenon (Hastie, Tibshirani, and Friedman, 2009).

Step 2: According to the chosen K (more explanations for the choice of K are given in section 2.4.3), randomly partition the VALTRN set into K equal subsets (also called folds).

Step 3: From the K generated folds, assign the first fold as the validation subset (VAL) and the remaining $K-1$ folds as the training subset (TRN).

Step 4: Apply the steps 2 to 5 of the true validation method (section 2.4.1) for the TRN and VAL subsets, and save the results of the measurement error in L_1

Step 5: iteratively assign the next fold as the validation subset (VAL) and the remaining $K-1$ folds as the training subset (TRN), and iteratively apply step 4 to TRN and VAL subsets. Save the results in L_i for $i = 1,...,K$.

Step 6: Average the obtained results $L = \frac{1}{K}\sum_{i=1}^{K} L_i$ and find the hyper-parameters θ_{hp} which minimize the value of L.

The proposed process can be adapted for different number of folds and can also be used several times in an embedded way. It is worth mentioning that when the number of folds is equal to the number of data points available, the method is then called leave-one-out cross-validation (Hastie, Tibshirani, and Friedman, 2009). More remarks are given in the next section 2.4.3 where special cases are developed.

2.4.3 Remarks and Recommendations

In the process of validation and cross-validation there are no rules of thumb. As the main objectives of a data driven model (in particular for machine leaning algorithms) is to extract the maximum of information, the use of the available data is subject to compromise for both the learning and the validation procedure. In that sense, here is a non exhaustive list of recurrent questions:

Why do we have to generate subsets?

The generation of subsets is essential for the optimization process. As the learning step and the validation step should be performed with independent data points, the generation of subsets is the first step in order to reach this goal but does not guarantee the independence of the subsets. This latter aspect is developed in the next section 2.5, and in section 2.5.4.

How many subsets to use?

For the case of a true validation method, the number of subsets is related to the number of "parameters" (here "parameters" should be understood in the broadest sense of the word). In all cases, regardless of the number of parameters for the selected method, the final optimal generated model has to be evaluated in order to estimate the generalization error. This evaluation needs its own subset usually called test (TST) subset. In the standard case, where the analysis is performed with a machine learning algorithm including parameters and hyper-parameters (i.e. $\theta = (\theta_p, \theta_{hp})$), two subsets are required: traditionally the training subset (TRN) for the optimization of the parameters θ_p, and the validation subset (VAL) for the optimization of the hyper-parameters θ_{hp}. Up to now it is a standard procedure with 3 subsets.

Let us consider that the practitioner wants to include the feature selection task in the modelling process. As mentioned in section 2.2 the model becomes a three levels structure $f(\tilde{X}, \theta)$ where an additional "parameter" (i.e. $\tilde{X}$ the subset of features) should be optimized. With this setting, a fourth subset needs to be generated (VALX). The optimization of the whole model does not change from the process proposed in section 2.4.1. An additional iteration should be performed around step 2 to 5 in order to evaluate different subsets of features, but the way of iterating will depend on the search strategy used to explore the combinations of features. More details on this topic are developed in section 3.2.

When should we use K-fold cross-validation?

The use of cross-validation instead of true validation method is really related to the number of available data points. When very few data points are present, the generation of distinct subsets, as in true validation method, can lead to misrepresentation of the phenomenon under study. In other words, the distribution of each generated subset can diverge from the original distribution of the whole dataset. If it is the case, either the training or the validation parts can fail and lead to a misinterpretation of the general behaviour of the phenomenon. For this reason, it is recommended to use K-fold cross-validation when few data points are available.

How many K-fold?

There are no fixed rules for the number K. In general, it can vary between 5, 10 or even 20 (Hastie, Tibshirani, and Friedman, 2009). The best way to choose this number is to calculate the size of each fold according to the number of available data points. For example, if the VALTRN dataset is composed of 100 points, by using a 5-fold cross-validation, each fold will contain 20 points. On the other hand, if a 20-fold cross-validation is used, the size of each fold will decrease to 5 data points. As mentioned in the process of section 2.4.2, this means that for one iteration the algorithm will train the model with 95 points, and evaluate this model only with 5 points. According to my point of view, the use of only 5 points to evaluate the model can lead to underestimate the true validation error of the model for this iteration. In that sense, it is better to use a 5-fold (or even a 10-fold) which will allow the algorithm to learn with 80 points and validate with 20 points. In the extreme case, K could be equal to the number of data points (i.e., the leave-one-out cross-validation). In this case, for each iteration, the algorithm will learn over 99 points and evaluate the model over 1 point. Of course, at the end, it is only the average result which will be taken into account (i.e. $L = \frac{1}{K}\sum_{i=1}^{K} L_i$). But as each L_i could underestimate the general validation error, L may also underestimate this value. Using less folds will lead to a better estimation of the error and also increase the computational speed.

2.5 Data Splitting

In section 2.4 the decomposition of the model parameters $\theta = (\theta_p, \theta_{hp})$ are shown and the use of different subsets (TRN, VAL and TST) are highlighted. Connected to this decomposition, the data splitting procedure is in charge of the generation of subsets, which will be used for the calibration of both the parameters and the hyper-parameters. For this reason, the data splitting is considered as one of the central and most important part of the machine learning pre-processing step. Results can significantly change according to the way the splitting procedure is done.

As the main goal of data splitting is to generate (or partition) subsets of the complete dataset, it can be seen and considered as being a part of the statistical sampling method. Another objective of the splitting procedure is to provide independent subsets which keep a good representation of the phenomenon under study. Although the latter point seems contradictory (i.e., being independent and representative of the phenomenon), the practitioner should keep in mind this aspect in order to understand and validate the results and the whole methodology.

According to a standard setting (the same as for the true validation section 2.4.1), the rest of this section will concentrate on the different manners of generating training, validation and testing subsets, or in a broader way, on the creation of a subset, and on the assessment of such splitting procedure.

2.5.1 Random Sampling

The random sampling (or simple random sampling) represents the easiest and the most commonly used method for generating subsets (Thompson, 2012). According to the chosen size of each subset (TRN, VAL and TST), it can randomly split the complete dataset in a straightforward and non-parametric way. This method gives good results when enough data points are allocated in each subset, and when the distribution of the whole dataset is homogeneous. In the case where clusters or complex distributions are present in the dataset, the use of random sampling can lead to a poor coverage of the phenomenon. And as mentioned in section 2.4.3, the training and the validation process can lead to a non-optimal model.

It is worth noting that most of the time the use of random sampling method for performing a K-fold cross-validation is preferred among other methods. This is mainly due to the fact that when several subsets need to be generated, the use of self-consistent methods like random sampling allows a simpler implementation in programming languages without loss of accuracy of the prediction.

2.5.2 Stratified Sampling

The stratified sampling consists in identifying (or generating) partitions of the whole dataset, which will be called strata, and then applying a random sampling within each stratum and proportional to the size of each stratum. It can be very useful when the output variable is imbalanced (for the case of classification problems, the distribution can vary among the classes) or when the distribution of the output variable is heterogeneous (i.e., non-normal or non-uniform). In the first case, the use of stratified sampling based on the different proportions of category (for imbalanced classes) allows the generation of subsets (mainly the TRN, VAL and TST subsets) with the same properties.

For example, for a study on forest fires in Portugal (Leuenberger et al., 2018) two machine learning algorithms were applied for a classification problem. In this case, the output variable was a categorical variable composed of 6 classes. As the classes were very imbalanced (i.e. class 1 was composed of more than 4’000’000 data points, while class 6 contained only 327 data points) the use of random sampling led to an under-estimation of the higher classes. For this reason, we decided to use a stratified sampling methods in order to generate the different subsets by taking into account the proportion of each classes. It allowed to generate models with a better consideration of the higher classes.

In some cases, when complex distributions are present in the input or in the output variables, the generation of strata may not be straightforward. It is the case when there are no evident or natural partitions in the dataset. In these circumstances, the generation of strata can be performed by using a clustering algorithm (e.g. K-means or K-medoids). By applying a clustering algorithm on the input variable (as in unsupervised learning), the original heterogeneous dataset will be divided into homogeneous clusters (at least more consistent subsets). Then, by considering each cluster as

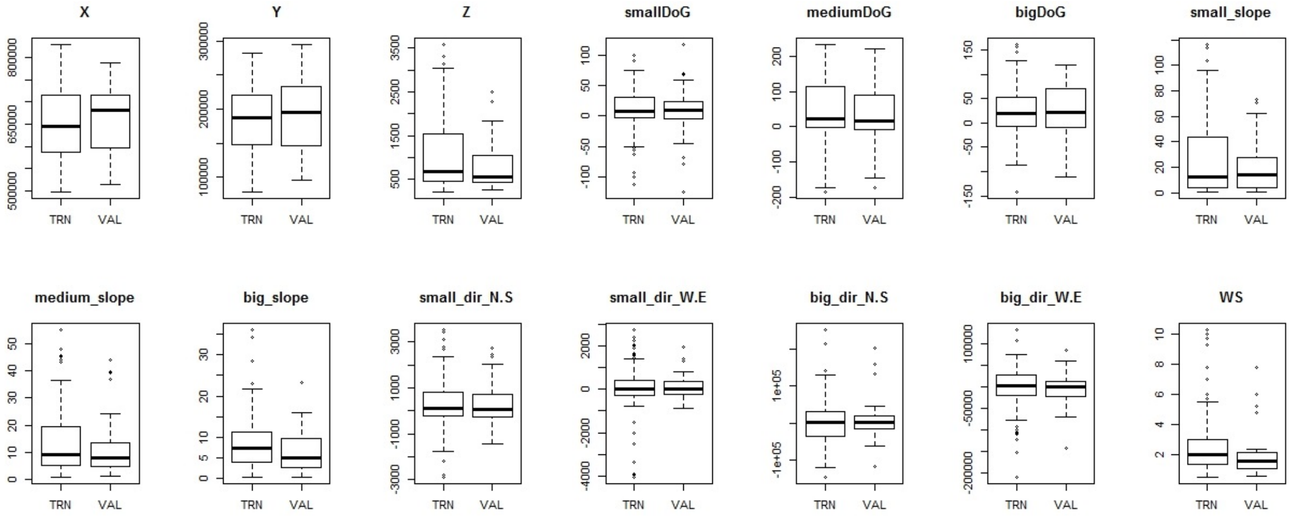


FIGURE 2.3: Example of boxplots for the quality control of the splitting procedure. Case study of wind field prediction in Switzerland.

a stratum, TRN, VAL and TST subsets can be generated with a stratified sampling by taking into account the proportion of each stratum.

2.5.3 Assessment of the Subsets

Once the different subsets have been generated, one may ask if they are consistent and representative of the phenomenon under study, or at least of the entire dataset. In this regard, and essentially when generating TRN, VAL and TST subsets with the true validation method, it is important to check the distribution of each subset for each variable (Leuenberger and Kanevski, 2015).

Figure 2.3 illustrates a practical example with the wind field dataset in Switzerland (Robert, Foresti, and Kanevski, 2012). In Leuenberger and Kanevski (2016), presented at the GeoENV conference (see also section 3.2.1), the generation of the TRN and VAL subsets was controlled with the use of boxplots. Several attempts were performed in order to find subsets with similar distributions for each variables.

The limitation of this technique resides in the fact that only 1-D representation of the distribution is checked. It means that two subsets (e.g. TRN and VAL) can show the same "1-D" distribution for two variables, but by looking at a "2-D" representation of their distribution, they can show a complete different behaviour (see figure 2.4). For this reason, when applying true validation with a complex dataset, it is recommended to perform several times the splitting procedure, for example 20 times, and to apply the desired algorithm to these 20 different splitting processes. By considering an average of theses 20 results, instead of only one result, we can avoid a misrepresentation of the phenomenon under study.

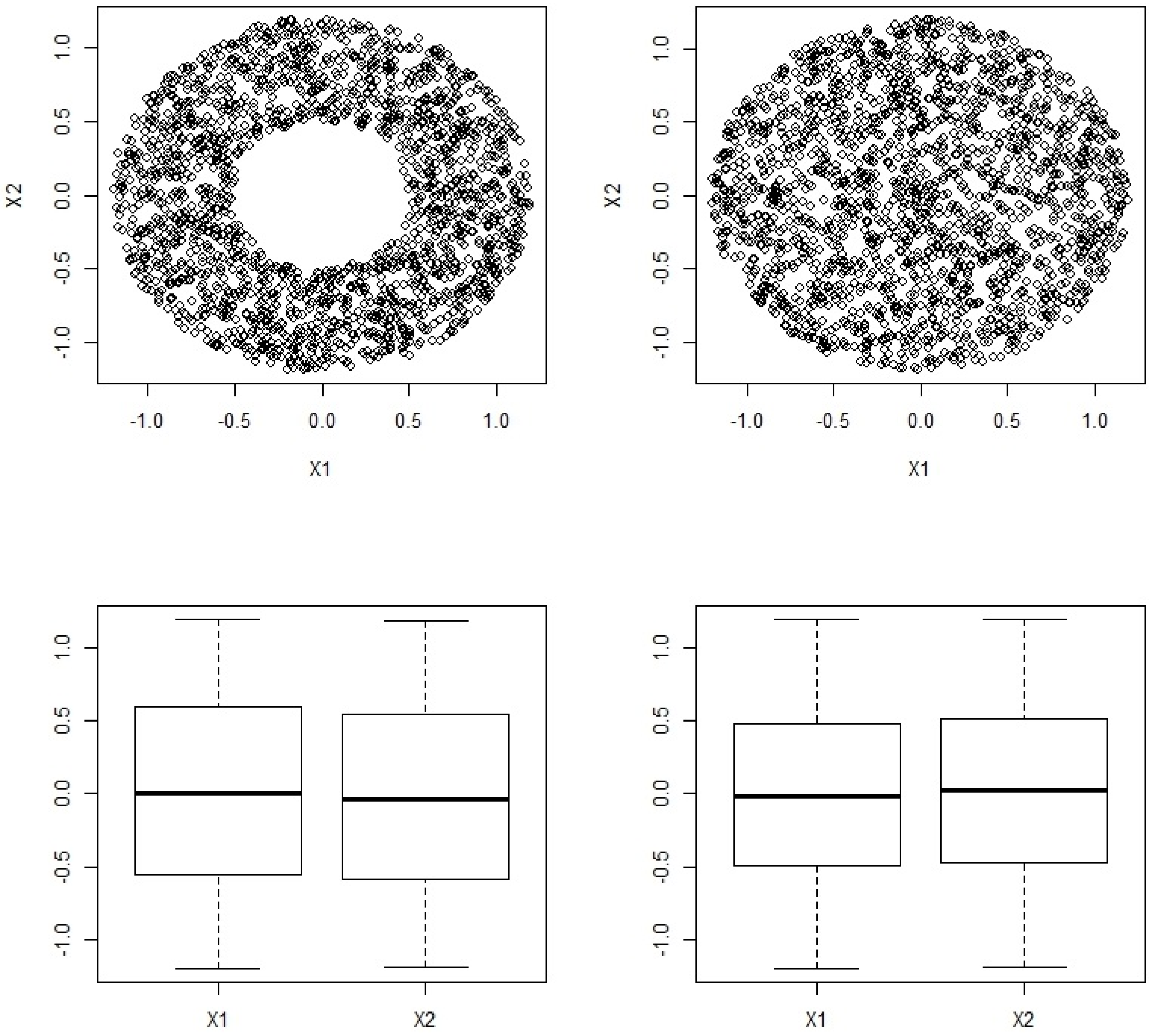


FIGURE 2.4: Example of boxplots when the 1-D distribution is identical, and the 2-D distribution is different.

2.5.4 Issues with Spatial Autocorrelation

When dealing with a huge dataset (with a lot of measurement points), it can happen that the generated TRN, VAL and TST subsets have very close data points between them. The closeness of these points associated with the spatial autocorrelation of the output variable can lead to an underestimation of the true model error. This is essentially due to the fact that most of the data used for the validation or the test of the model were already used by "similar" data during the training process.

In order to minimize this effect, a special attention should be paid during the exploratory data analysis, where the level of spatial autocorrelation can be estimated with the use of a variogram. For the splitting process, the use of a clustering method (with a high number of clusters) can generate homogeneous and small areas. Then splitting the areas instead of the data points can lead to TRN, VAL and TST subsets with less nearby points.

In some cases, external means can be used in order to split the data with less "similarity" between TRN, VAL and TST subsets. For example, in the case study of landslide susceptibility maps, each polygon of landslide was composed of pixels. Instead of splitting the pixels, the use of an object-based sampling strategy (the splitting was performed on the polygons) allows us to preserve from an under estimation of the true classification error (pages 42 to 44 in Micheletti et al. (2014)).

2.6 Measures of Evaluation

In section 2.2 -equation 2.3- a cost function symbolized by L was introduced. In this section different cost functions are presented for both regression and classification problems. They all have the same purpose of computing errors between predicted and observed data points. They are used in the validation process in order to evaluate and find optimal parameters, and in the testing process in order to estimate the generalization error of the final model. However, there is no clear guideline on when to use what cost function, but a good knowledge on the nature of the data (i.e., the general range, distribution, etc.) can help to select the appropriate cost function (Hastie, Tibshirani, and Friedman, 2009; Cherkassky and Mulier, 2007).

For the case of a regression problem, let us consider the following labelled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^d$ as input values and $y_i \in \mathbb{R}$ as output value for an observed data point i. The predicted value of a model (either optimal or not) f with parameter $\hat{\theta}$ for the point $\mathbf{x}_i$ is defined as follows:

$$\hat{y}_i = f(\mathbf{x}_i, \hat{\theta}) \tag{2.6}$$

In this context, the most common measurements of errors are:

Mean Squared Error (MSE)

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{2.7}$$

It has the particularity to be one of the most widely used cost function in regression problems, but suffers of a high sensitivity in outliers or extreme values. In those cases, outliers detection or non-linear normalization (e.g. log-transformation on the output variable) are recommended.

Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{2.8}$$

Also sensible to outliers or extreme values as MSE, RMSE differs from MSE by the square root, which allows the value of RMSE to be at the same scale as the original output variable y.

Mean Absolute Error (MAE)

$$\frac{1}{N}\sum_{i=1}^{N} \mid \hat{y}_i - y_i \mid \tag{2.9}$$

Close to the RMSE measure, MAE has the advantages of being easier to interpret than MSE or RMSE (the measure is in the same unit as the original variable y), and to be less sensitive to outliers or extreme values.

Mean Absolute Percentage Error (MAPE)

$$\frac{100}{N}\sum_{i=1}^{N} \mid \frac{\hat{y}_i - y_i}{y_i} \mid \tag{2.10}$$

Close to the mean absolute error, the mean absolute percentage error can be applied and expresses the error as a percentage. Nevertheless, this measure of error has several drawbacks which can make it difficult to apply. For example, the measure cannot be used when there are zero values in the output y_i. Moreover, the measure is biased in the sense that it will privilege lower prediction (i.e., $\hat{y}_i < y_i$) than higher prediction (i.e., $\hat{y}_i > y_i$).

Coefficient of Determination (R^2)

$$1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.11}$$

Here $\bar{y}$ denotes the mean value of the observed data $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. This measure of error between observed and predicted data lies normally in the interval $[0,1]$. It can be seen as the ratio between the variance of the model error and the variance of the observed data. A result close to 1 indicates that the considered model works well on the data. On the other hand, a results close to 0 indicates a poor performance of the model. Let us note, that in some cases the computed value of R^2 can be negative. In these cases, the model used is worse than the mean value of the data.

In the case of a classification problem, the only difference is the nature of the output variable y. In this case the labelled dataset is $\left\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathscr{C}\right\}_{i=1}^{N}$ where $\mathscr{C} = \{0,1\}$ when considering a binary problem, or $\mathscr{C} = \{1,2,...,C\}$ for a multi-class problem. Here are the descriptions of the most common measures of error for classification model:

Accuracy (ACC)

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{\hat{y}_i = y_i\}} \tag{2.12}$$

Here $\mathbf{1}_{\{\hat{y}_i = y_i\}}$ is an indicator function with value 1 when $\hat{y}_i = y_i$ and 0 when $\hat{y}_i \neq y_i$. The measure of accuracy is one of the most standard measure for classification problems. It counts the number of good predictions and normalizes this number by the total number of points present in the dataset. This measure can also be derived from a confusion matrix (i.e. a special case of the contingency table) with the followed equation:

$$ACC = \frac{TP + TN}{\#P + \#N}, \tag{2.13}$$

		Observed classes	
		1	0
Predicted classes	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

TABLE 2.2: Structure of a confusion matrix for two groups only. Here the right predictions are determined by the number in TP and TN, while wrong prediction are in FN and FP.

where $\#P$ and $\#N$ denote the number of real positives (class 1) and real negatives (class 0) in the considered dataset (table 2.2).

It is worth noting that this measure works well when the classes are balanced in the dataset. In the case of imbalanced classes, the value of ACC can misrepresent the real accuracy of the model. For example let consider a dataset with 95 data points as class 1 and 5 data points as class 0. If a model classified the 100 data points as class 1, the value of ACC will be 0.95 (which is a good value), but this value does not reflect the inability of the model to classify at least once the class 0. In these cases, other metrics should be investigated (e.g. true positive rate, true negative rate, precision, recall, or F-score) or at least an analysis of the confusion matrix should be performed.

Cohen's kappa (κ)

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.14}$$

where p_o is the observed accuracy and p_e the expected accuracy. According to table 2.2, p_o and p_e are computed as follows:

$$p_o = \frac{TP + TN}{TP + FP + FN + TN} = ACC, \tag{2.15}$$

$$p_e = p_{class0} + p_{class1}, \tag{2.16}$$

where,

$$p_{class0} = \frac{TN + FN}{TP + FP + FN + TN} \cdot \frac{TN + FP}{TP + FP + FN + TN}, \tag{2.17}$$

$$p_{class1} = \frac{TP + FN}{TP + FP + FN + TN} \cdot \frac{TP + FP}{TP + FP + FN + TN}. \tag{2.18}$$

With this notation p_e can be considered as the expected accuracy that a random classifier can reach by taking into account the number of points in each class. The kappa index can be a good alternative when the studied dataset contains imbalanced classes. But in any case, the use of such an index should always be interpreted along with the corresponding confusion matrix.

2.7 Complexity Analysis

The complexity of an algorithm, and more generally of a model, can provide valuable information about the hidden relationship between input and output variables, and more generally about the phenomenon under study. For example, knowing the degree of complexity among different sets of variable can explain and help to understand a phenomenon much more than just having a prediction map. Moreover, the detection of linearity or non-linearity into the relationship between input and output variables can highlight new way to understand a phenomenon. In this section, two methods for evaluating and interpreting the complexity are highlighted. The notion of complexity is, for the first method, based on the hyper-parameter value, and for the second one, based on the performance of extreme learning machine with different activation functions.

2.7.1 Hyper-parameter related Complexity

When applying a machine learning algorithm on a dataset, there is always an optimization phase, which determines the optimal values for the parameters $\theta = (\theta_p, \theta_{hp})$. During this process, several values of hyper-parameter θ_{hp} are trained and then tested (either by the use of true validation or cross-validation). As it was mentioned in section 2.4.1, these values are iteratively changing from the lowest to the highest degree of complexity. This process is finally stopped when the validation curve reaches a minimum and the corresponding optimal hyper-parameter θ^*_{hp} is selected for the final model.

In this context, the optimal hyper-parameter θ^*_{hp} can be considered as an index of complexity. It can be easily understood in the case of polynomial functions, where high degree of polynomial functions implies a more complex relationship between input and output variable. In the case of artificial neural networks, the hyper-parameter, which is related to the structure of the network (i.e. number of layers and number of neurones), suggests also a degree of complexity, in the sense that more layers and more neurones imply more flexibility for the model to learn the data.

As an example, let consider a dataset where there is no structure in the output variable (i.e. no relationship between input and output variables). In this case, the best possible model is the mean value of the available data points. When applying a polynomial model on this dataset (with true validation or cross-validation), the best hyper-parameter, which is k, the degree of the polynomial function, will be equal to 0, which is the model of a constant value ($f(x) = \sum_{i=0}^{k=0} a_i x^i = a_0$). For the case of artificial neural networks, the hyper-parameter will select the value of 1 layer with 1 neuron, which is the simplest network structure.

It is worth mentioning that all these considerations are not applicable on all machine learning algorithms. For example, the hyper-parameters of random forest algorithms (Breiman, 2001), which are the number of trees and the number of selected variables in each splitting node, are difficult to interpret in term of complexity. It is due to the fact that these hyper-parameters do not have a clear order relationship (or partially ordered set). It is the same for the k-nearest neighbours (k-NN) model, where the number of neighbours k (which is the hyper-parameter of this model) does not express directly a "complexity index" (in a broad sense). Because, as the 1-NN model can be seen as the most simple model of this family, the "∞-NN" model is the optimal one for the case where there is no structure in the output variable.

In addition to the proposed tools for quantifying the complexity (here according to hyper-parameter related complexity and in section 2.7.2 according to Extreme Learning Machine Complexity), it should be mentioned that other means of model complexity analysis exist, such as: Shannon information, minimum description length, Bayes factor, or the number of support vectors in the support vector machine algorithms (Cherkassky and Mulier, 2007; Hastie, Tibshirani, and Friedman, 2009; Rissanen, 1978).

2.7.2 Extreme Learning Machine Complexity

This section highlights and investigates the different issues that can occur when identifying the complexity (linear/non-linear) of environmental data using machine learning algorithms. In particular, the main attention is paid to the description of a self-consistent methodology for the use of Extreme Learning Machine (see section 3.1), which recently gained a great popularity. By applying two ELM models (with linear and non-linear activation functions) and by comparing their efficiency, quantification of the linearity can be evaluated.

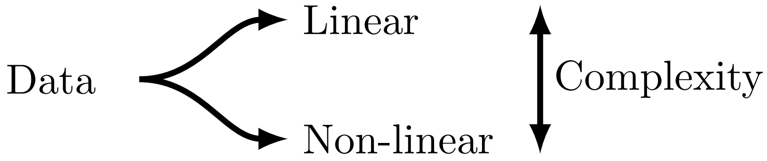


FIGURE 2.5: According to the data available, the complexity can be highlighted by the presence of linear or non-linear relationship between input and output variables.

It is worth mentioning that the combination of Extreme Learning Machine with two different activation functions is very promising. Compared to classical machine learning algorithm like multilayer perceptron (Rosenblatt, 1961), ELM has the ability to learn faster without loss of accuracy, and needs only one hyper-parameter to be fitted. With these good properties, it allows us to generate large amount of models for both the linear and the non-linear ELM, which for standard algorithms is not feasible.

In particular, the following activation functions for the ELM model are considered:

$$g(x) = \begin{cases} x, & \text{linear.} \\ \frac{1}{1+e^{-x}}, & \text{sigmoid,} \end{cases} \tag{2.19}$$

and are used in the model function of ELM (see section 3.1 for more details):

$$\sum_{j=1}^{\tilde{N}} \beta_j g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j) = y_i. \tag{2.20}$$

According to this setting, the following methodology is applied:

Step 1: Split the data between training (TRN) and testing (TST) sets.

Step 2: Apply a 10 fold cross-validation with the training set, in order to find the hyper-parameter of ELM θ_{hp}, that is the number of nodes in the hidden layer $\tilde{N}$.

Step 3: For each iteration of the cross-validation procedure and for each number of nodes between 1 to 200, train two ELM models. The first one with a linear activation function and the second one with a non-linear (in this case a sigmoid function,eq. 2.19).

Step 4: According to the training and validation subsets (generated by the cross-validation procedure), compute the mean value of the Mean Square Error (MSE) for the two ELM models.

Step 5: Select the linear and the non-linear models with the lowest MSE.

Step 6: Compute the index I_C (an index of complexity) defined as follow:

$$I_C = \frac{MSE_l - MSE_{nl}}{MSE_l}, \tag{2.21}$$

where MSE_l corresponds to the optimal MSE of ELM with a linear activation function, and MSE_{nl} is the best MSE for the non-linear ELM model. If multiple minima are detected, select the one with the smallest number of nodes (essentially for visualisation purpose, because the MSE are equal for multiple minima).

This index is included into the interval $[0,1]$ when $MSE_{nl} \leq MSE_l$. If it is not the case (due to stochastic variation on the ELM model), a bound can be fixed at 0 as follow:

$$I_C = \max\left(\frac{MSE_l - MSE_{nl}}{MSE_l}, 0\right). \tag{2.22}$$

In order to understand the behaviour of this new index I_C, the Butterfly dataset (Golay, Leuenberger, and Kanevski, 2017) was used. It contains a total of 8 input variables, which are $\{X1, X2, J3, J4, J5, I6, I7, I8\}$, for one output variable $\{Y\}$. $X1$ and $X2$ correspond to the two original variables from which Y was built (combination of sigmoid functions). $J3, J4$ and $J5$ are redundant variables (non-linear relationship with $X1$ and $X2$), and $I6, I7$ and $I8$ are irrelevant variables with non-linear relationship between them. According to this dataset, five scenarios including different subsets of input variables (i.e. $\{X1,X2,J3,J4,J5,I6,I7,I8,Y\}$, $\{X1,X2,Y\}$, $\{X1,X2,I6,Y\}$, $\{X1,X2,I6,I7,I8,Y\}$ and $\{X1,X2,I6,I6,I6,Y\}$) with different number of data points (i.e. 100, 200 and 1000) were compared. In the last scenario (i.e. $\{X1,X2,I6,I6,I6,Y\}$) the three variables $I6$ are generated randomly with the same distribution.

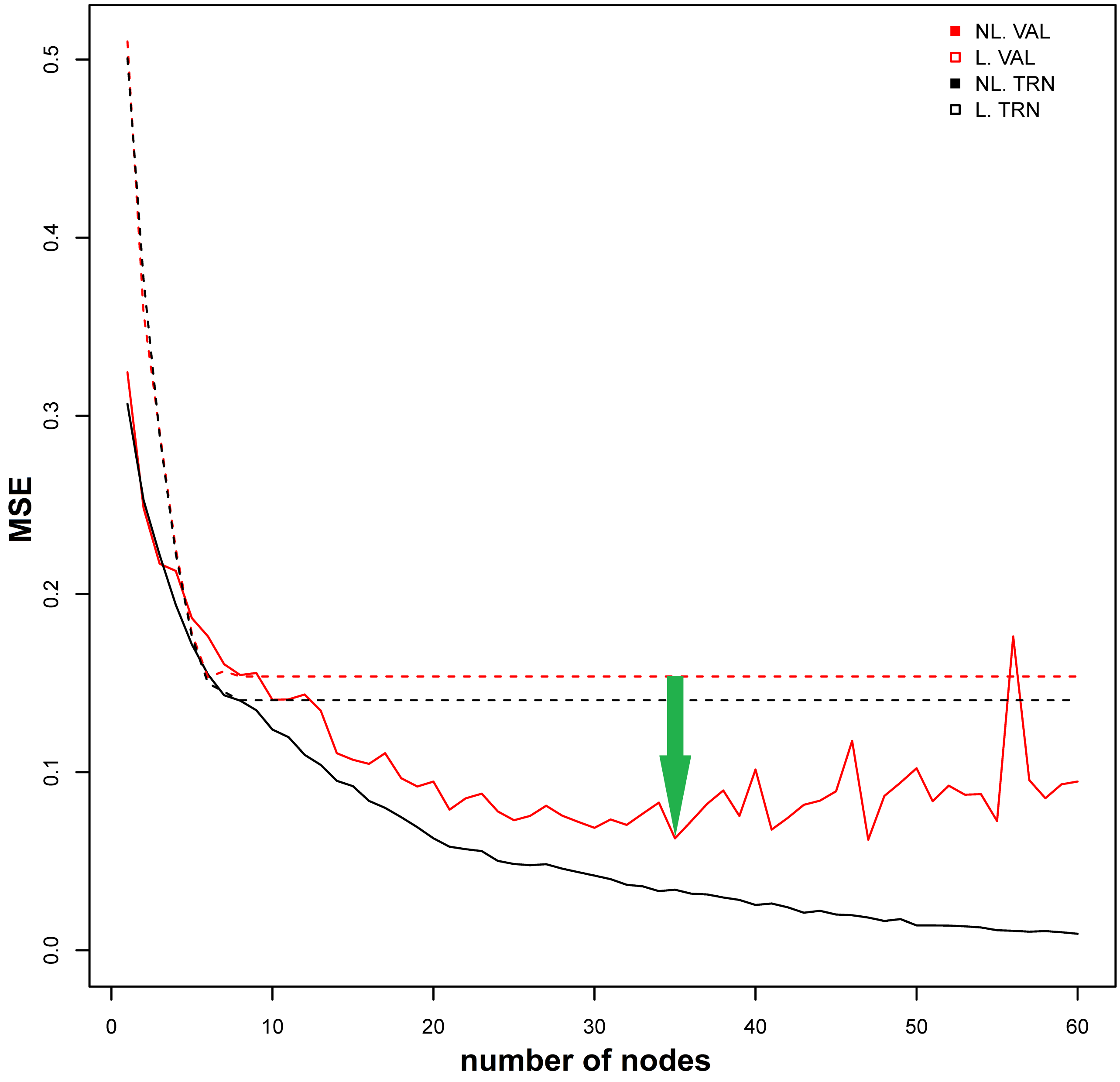


FIGURE 2.6: Difference in MSE between linear and non-linear models by considering the set of variables $\{X1, X2, Y\}$. The green arrow shows the difference between MSE_l (the linear model computed on the validation subset in red dashed line) and MSE_{nl} (the non-linear model computed on the validation subset in red solid line) when they reach their minimum.

Figure 2.6 illustrates the behaviour of this index I_C for the case where the phenomenon is purely non-linear (by considering only the set of variables $\{X1, X2, Y\}$). The green arrow shows the difference between MSE_l (the linear model in dashed line) and MSE_{nl} (the non-linear model in solid line) when they reach their minimum. And figure 2.7 shows the two optimal solutions for the linear and non-linear ELM models. It is worth mentioning that the non-linear validation curve (i.e., solid red line in figure 2.6) shows a stochastic behaviour because of the variability nature of ELM. Moreover, when multiple minima are detected, the one with the smallest number of nodes (i.e., the less complex

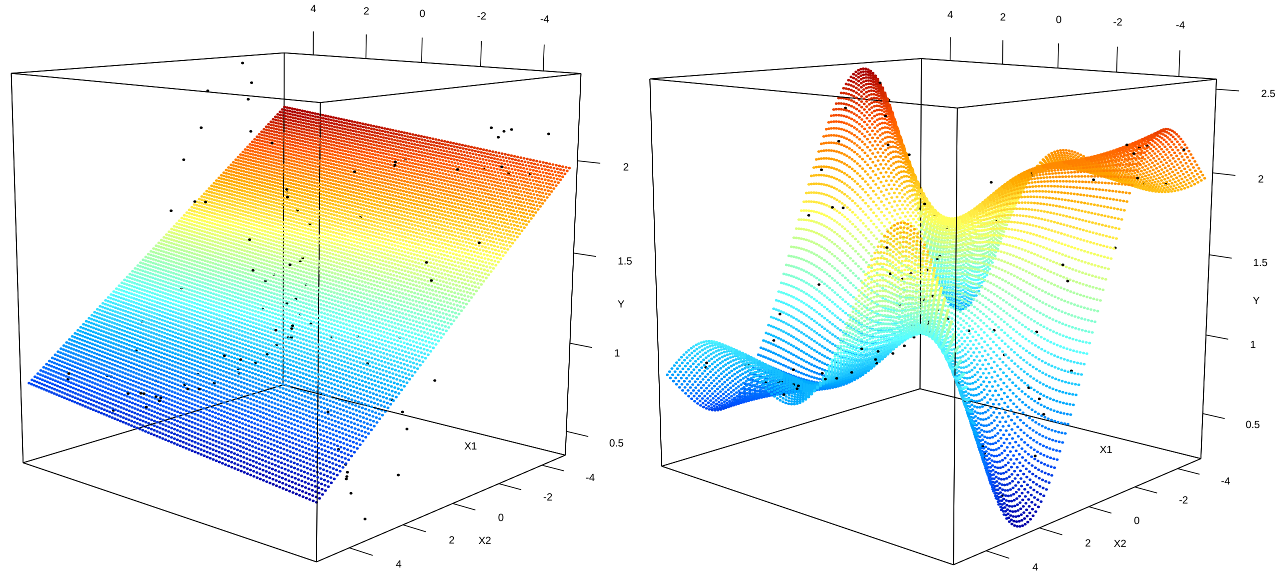


FIGURE 2.7: Visualization of the butterfly dataset $\{X1, X2, Y\}$ with the linear ELM model (on the left), and the non-linear ELM model (on the right).

model) should be selected.

Table 2.3 highlights the different scenarios according to the set of variables and the number of points used. When considering all the input variables with different number of data points (the first tree rows of the table), the index I_C increases as the number of points increases. This aspect reflects the fact that phenomena with a low number of points are harder to be modelled and, in some extreme cases, the best predictor could be the mean value which can be easily caught by linear models.

In the case where relevant information ($X1$ and $X2$) is mixed with irrelevant variables ($I6, I7$ and $I8$) the performance of the non-linear ELM model decreases while the linear model stays quite stable. This observation represents the main characteristic of a feature selection paradigm (i.e. the addition of redundant or irrelevant variables tends to increase the general error of the model).

In the opposite case, where the relationship between input and output variables is linear, both models reach the optimal MSE at the same level, which results in an index close to 0.

Finally, the characterization of the complexity (at different levels) is an important part of the machine learning process. Moreover, the identification of the linear or non-linear behaviour between input and output variables adds valuable information for the knowledge of the phenomenon complexity.

2.8 Concluding Remarks

The main goals of environmental data science using machine learning algorithms deal, in a broad sense, around the calibration, the prediction and the visualization of hidden relationship between input and output variables. In order to optimize the models and to understand the phenomenon under study, different aspects (presented in a broad way in section 2.1, and then in detail in the remaining

		linear		non-linear		
Variables	# points	MSE	$\tilde{N}$	MSE	$\tilde{N}$	I_C
X1, X2, J3, J4, J5, I6, I7, I8	100	0.141	27 (8)	0.0981	24 (20)	0.3043
X1, X2, J3, J4, J5, I6, I7, I8	200	0.158	6	0.0564	62 (30)	0.6430
X1, X2, J3, J4, J5, I6, I7, I8	1000	0.1528	9	0.0157	85	0.8973
X1, X2	100	0.1327	3	0.0168	30	0.8734
X1, X2	200	0.1507	3	0.0149	38	0.9011
X1, X2	1000	0.1535	10 (3)	0.0125	55 (35)	0.9186
X1, X2, I6	100	0.1327	6 (3)	0.0784	21	0.4092
X1, X2, I6	200	0.1521	5	0.0257	79	0.831
X1, X2, I6	1000	0.154	6	0.0143	98	0.9071
X1, X2, I6, I7, I8	100	0.1394	5	0.1361	24 (8)	0.0236
X1, X2, I6, I7, I8	200	0.1547	5	0.0839	32 (10)	0.4577
X1, X2, I6, I7, I8	1000	0.154	8 (5)	0.0523	43	0.6604
X1, X2, I6, I6, I6	100	0.136	28 (6)	0.1362	8	0
X1, X2, I6, I6, I6	200	0.1532	103 (6)	0.1273	71	0.1691
X1, X2, I6, I6, I6	1000	0.154	194 (6)	0.0546	109	0.6455

TABLE 2.3: Different scenarios are highlight according to the set of variables and the number of points used. The mean square error (MSE), the optimal number of node ($\tilde{N}$) as well as the I_C index are computed for each scenario. The optimal number of node in parenthesis denotes the optimal one found by visual evaluation.

section of chapter 2) have to be considered. Among them, parameters versus hyper-parameters, data splitting, measurement for evaluation and complexity analysis are discussed.

An important contribution of this research deals with an elaboration of a self-consistent methodology that can be used for intelligent decision making process. It should be noted that the methodology used does not depend on particular machine learning model and can be applied for any data driven modelling tools.

Chapter 3

Extreme Learning Machine

3.1 Presentation and Theory

Extreme Learning Machine is based on the artificial neural network concept. Following the structure of a single-hidden layer feedforward neural network (SLFN), it connects all input variables to the hidden layer, computes the neuron value, and then calculates a weighted average of all neurons with optimal weights, which will be assign to the output layer (Huang, Zhu, and Siew, 2006; Leuenberger and Kanevski, 2015). More formally, composed of *nbnode* neurons ($\tilde{N}$) and by using an activation function $g : \mathbb{R} \to \mathbb{R}$, the ELM network connecting the input ($\mathbf{x}_i$) to the output (y_i) value can be written in the following form:

$$\sum_{j=1}^{\tilde{N}} \beta_j g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j) = y_i, \tag{3.1}$$

where $\mathbf{x}_i \cdot \mathbf{w}_j$ is an inner product between the input ($\mathbf{x}_i$) and the weight vector ($\mathbf{w}_j$) which connects the input layer to the j^{th} neuron, b_j is the bias of the j^{th} neuron, and β is a weight vector connecting the hidden layer to the output layer. In a more compact way, ELM can be written as:

$$H\beta = \mathbf{y}, \tag{3.2}$$

where $H_{i,j} = g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j)$ is the output matrix of the hidden layer (Fig. 3.1).

According to this notation, ELM algorithm applies the following steps without iteration:

1. randomly generate the input weight $\mathbf{w}_j$ and the bias b_j (by using a uniform distribution);

2. compute the matrix H;

3. compute the output weight $\beta = H^{\dagger}\mathbf{y}$, where $H^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix H (Moore, 1920).

It is worth mentioning that the only parameter which should be fitted is the number of hidden neuron ($\tilde{N} = nbnode$). This latter can easily be optimized by applying the procedure highlighted in section 2.4. Finally, when all weights and biases of the network are defined, new data points can be predicted and the testing error evaluated.

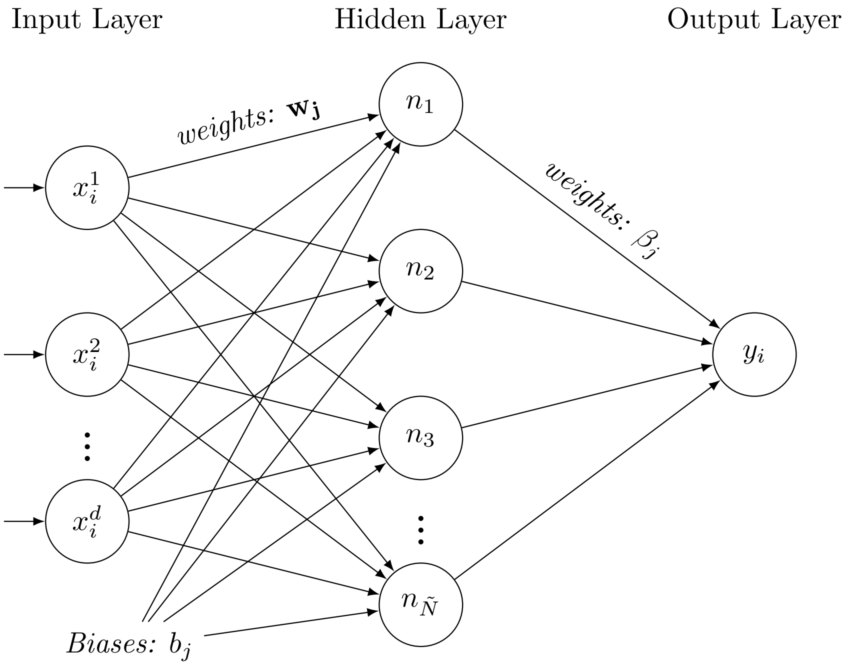


FIGURE 3.1: Structure of Extreme Learning Machine (ELM) following a single-hidden layer feedforward neural network (SLFN).

Notice that ELM allows fast and accurate predictions and it was proven to be a universal modelling tool (Huang, Zhu, and Siew, 2006). Furthermore, the algorithmic complexity of ELM lies in the computation of the Moore-Penrose generalized inverse of matrix H, which in this case uses the singular value decomposition algorithms (SVD) (Huang, Zhu, and Siew, 2006). This means that the quickness of ELM refers just to the computation of weights. The optimization phase, which resides in finding the optimal hyper-parameter (the number $\tilde{N}$ of hidden neurons for ELM), does not differ from other machine learning algorithms. For these reason, ELM algorithms make a suitable candidate compare to the traditional multilayer perceptron (MLP), which can fall in local minimum during the learning process, and which can take much more time for the same accuracy.

In this thesis, all computations of ELM were performed by using the *elmNN* R package (R Core Team, 2016).

3.2 Feature Selection

In many fields of data-driven sciences (e.g. geo- and environmental sciences, biocomputing, finance, astronomy, ect...), the need for efficient methods to carry out features selection tends to increase dramatically (Donalek et al., 2013; Meiri and Zahavi, 2006; Micheletti et al., 2014). Because phenomena under study lie in high dimensional spaces (e.g. for natural hazards: $d \approx 10-100$), it is a challenging task to reach the real dimension (i.e., finding the relevant features or variables) where the true phenomena can be understood, explained and predicted (Kanevski, Pozdnoukhov, and Timonin, 2009; Hastie, Tibshirani, and Friedman, 2009). Moreover, in most real data cases the relationships between features and phenomena are non-linear and complex. Keeping in mind that these relationships involve not only one but several features, the main goal is to select relevant subsets of features according to their potential non-linear ability to explain or predict relationships between input and output variables.

There are a lot of methods in wrapper, filter and embedded methodologies (Guyon and Elisseeff, 2003; Guyon et al., 2006; Lee and Verleysen, 2007; Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos, 2013; van der Maaten, Postma, and van den Herik, 2009). On the one hand, filter methods are faster but do not necessarily take into account the combinations of various features simultaneously (a feature can be irrelevant alone but may be relevant combined with other features). Because of the method speed, they allow applications with high dimensional datasets, but hardly deal with the real complexity of the phenomena. On the other hand, wrapper methods allow complex associations of features but suffer from the curse of dimensionality when considering all possible combinations of features. In the latter, the use of heuristic models is usually advised (Kohavi and John, 1997). Recent publications show new developments by merging global optimization algorithms with machine learning algorithms for classification or regression problems. For this purpose, here are some of the key publications according to the algorithms used: simulated annealing with backpropagation networks (Lin et al., 2008a); simulated annealing with support vector machines (SVM) (Lin et al., 2008b); particle swarm optimization (PSO) with logistic regression (Unler and Murat, 2010); PSO with SVM (Huang and Dun, 2008; Liu et al., 2011); extreme learning machine (ELM-based feature selection) (Frénay et al., 2013; Leuenberger and Kanevski, 2014). All these methods show good aptitude for different tasks and data, and new combination of algorithms should be investigated.

3.2.1 Exhaustive search

In this section, a standard method for feature selection called exhaustive search, is presented. For a considered dataset composed of d input variables and one output variable, the exhaustive search strategy aims to successively generate all combinations of input variable subsets. During the iterations, each subset is evaluated by a classifier (or according to the nature of the output data, a regression model) and the general error is computed. At the end, the subset with the lowest error is selected. The selected subset is therefore the best subset of input variable which can explain the output variable. But it is worth mentioning that the obtained result is sensitive to the model used and also to the way the general error is computed. In this sense, it is reasonable to say that the obtained result represents the input variables which have the best hidden relationship with the output variable, but only according to the method and model used.

The following section presents an applications of ELM for feature selection by using exhaustive search.

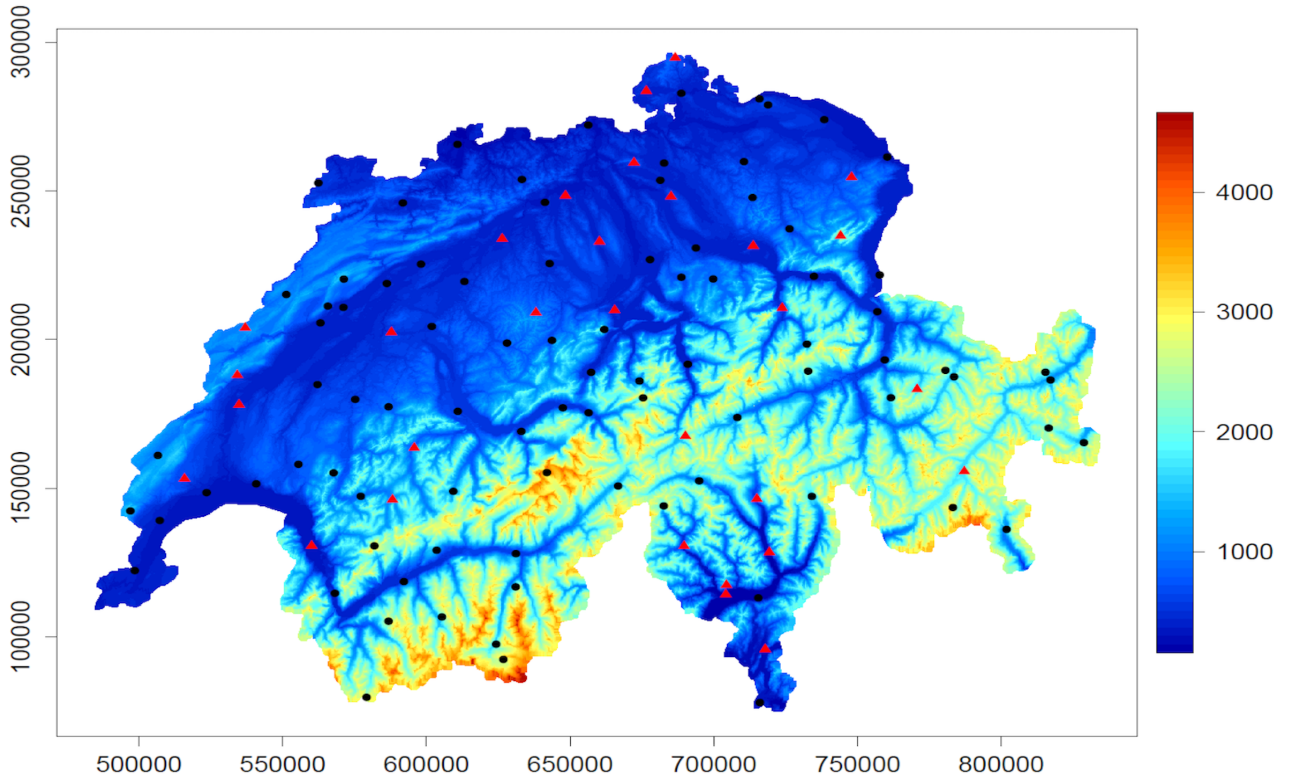


FIGURE 3.2: Study area with training (black dots) and testing (red triangles) data points. Colour scale displays the elevation in meters.

Wind Fields in Complex Regions

The present research highlights and investigates an application of Extreme Learning Machine to model monthly wind speed in Switzerland for the year 2008. Based on 118 measurement points, the input space was constructed using a Digital Elevation Model (DEM) and complementary geo-features, containing information about slope, North and West aspects, difference of Gaussians (DoG), etc. constituting a set of 13 independent variables. A description of all variables as well as predictions using general regression neural networks are presented in Robert, Foresti, and Kanevski (2012).

The following methodology was applied for both the optimization of the algorithm and the feature selection task.

Step 1: Split the data between training and testing sets.

Step 2: Apply a 5-fold cross-validation with the training set, in order to find the hyper-parameter of ELM, that is the number of node $\tilde{N}$ in the hidden layer (see section 3.1 for more details).

Step 3: Generate 100 bootstrap subsets from the training data and use the optimal hyper-parameter $\tilde{N}$ in order to generate 100 ELM models. Then, compute a mean prediction value.

Repeat steps 2 and 3 with all the combinations of subsets, that is $2^{13} - 1 = 8191$, and select the subset of features with the lowest mean squared error (MSE).

months	best subsets	MSE best	$\tilde{N}$ best	MSE all	$\tilde{N}$ all
January	1,5	1.6425	9	2.6953	13
February	1,2,3,4,5	0.9628	12	1.5277	13
March	1,2,3,4,6,7	1.5229	14	2.0787	16
April	1,2,3,5	0.9159	14	1.5880	11
May	1,2,5	0.5764	14	1.0365	7
June	1,3,5	0.6136	13	0.9477	6
July	1,2,5,7	0.5522	19	0.6976	5
August	1,3,5,11	0.6526	11	1.0866	8
September	1,3,4,6	0.8349	11	1.2081	16
October	1,2,3,4,5	0.7628	13	1.2213	13
November	1,3,5,11	1.1740	13	2.0178	14
December	1,2,3,4,5,6	1.3434	12	2.3923	12

TABLE 3.1: Results of the feature selection.

Code	Variables	Code	Variable
1	X	8	medium slope
2	Y	9	big slope
3	Z	10	small DD North-South
4	small DoG	11	small DD West-East
5	medium DoG	12	big DD North-South
6	big DoG	13	big DD West-East
7	small slope		

TABLE 3.2: Names of variables for each code number. DoG means Differences of Gaussians and DD means directional derivative

Table 3.1 highlights the best subset of feature for each month of the year 2008. For each month, the MSE and the optimal number of hidden nodes ($\tilde{N}$) are recorded for both the best subset of features and the subsets containing all the features. The number code for the 13 features are the ones presented in table 3.2.

Figure 3.3 displays the mean prediction of the 100 ELM bootstrap models (from step 3) for January taking into account only the best subset of features, which is for this case the variable X (1) and the medium DoG (5).

In this context ELM is well adapted for a feature selection task. As it can be performed very quickly, it allows the evaluation of all combination of subsets (8191 in total for this study) and gives, with $\tilde{N}$, valuable information about the complexity of the phenomenon under study.

Moreover, the best subsets selected consider much less features among the 13 ones, and the MSE can be significantly reduced, which means that the remaining features do not provide relevant information for this case study. This provides a clearer representation of the wind behaviour for the different considered months.

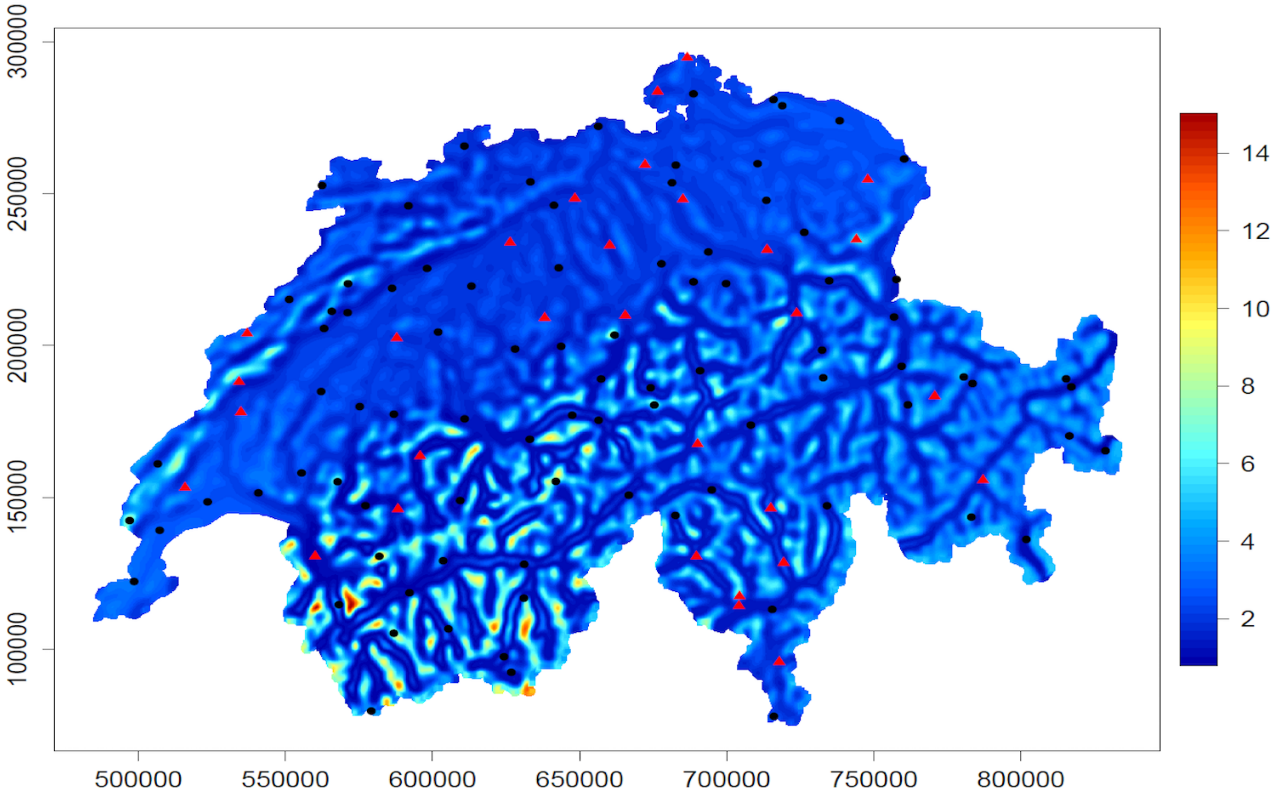


FIGURE 3.3: Wind speed predictions (m/s) for January with the best subset of features (variable X (1) and the medium DoG (5)).

3.2.2 Simulated Annealing based Feature Selection

The following section proposes a methodology combining Extreme Learning Machine (ELM, Huang, Zhu, and Siew (2006)) and Simulated Annealing (SAN, Kirkpatrick, Gelatt, and Vecchi (1983)) algorithms. The ELM algorithm has showed good capability for merging methods (Frénay and Verleysen, 2010) and provides fast and accurate prediction on various sorts of data from complex to highly non-linear. The proposed SAN algorithm remains a good optimization algorithm despite the fact that some new studies showed better performance (principally for computational time) by combining with a genetic algorithm (Gheyas and Smith, 2010).

The principal advantages of this method are the following:

1. ELM allows the quick evaluation of the non-linear potential of subsets of features,

2. SAN alows the optimal subset of features to be reached without using an exhaustive search.

The use of ELM instead of the more robust and accurate OP-ELM (Miche et al., 2010) resides in the fact that current version of OP-ELM cancel out the wrapper ability to detect irrelevant features.

Simulated Annealing Algorithm

SAN is a metaheuristic algorithm for optimization problems inspired by the field of metallurgy (Kirkpatrick, Gelatt, and Vecchi, 1983). Initialized with a high temperature parameter, it performs a global random search from neighbour to neighbour. In a second stage, temperature decreases progressively and the search becomes local. Based on the Metropolis criterion (Metropolis et al., 1953) it has the capability to accept bad solutions according to the level of the current temperature T.

Let θ_{cur} and θ_{new} respectively be the current and new states of the research, and let $f(\theta)$ be a cost function to minimize. If the difference between the new state and the current state is less than zero

$$\Delta f = f(\theta_{new}) - f(\theta_{cur}) \leq 0, \tag{3.3}$$

the new state θ_{new} is accepted, else θ_{new} is accepted with a probability:

$$P = \exp(-\Delta f / T) \tag{3.4}$$

In a theoretical way, the ability to accept bad solutions allows to find the global minimum of any problem. In a practical way, it cannot guarantee to find the optimal solution but it can approach it. The success of this convergence lies in a good parametrization of the initial temperature and in the annealing schedule (Filippone, Masulli, and Rovetta, 2011; Press et al., 2007).

Feature Selection Methodology

Let n be the number of features available and $\Theta = \{\theta \mid \theta = \{0,1\}^n\}$ be the state space of the whole combination of features, where θ_i indicates if feature i is considered or not (e.g. if the number of features is 3, $\theta = (1,0,1)$ means that features 1 and 3 are selected but not feature 2). The goal is to find the best subset of features $\theta^* \in \Theta$ that minimizes the cost function $f(\theta)$ defined as follows:

$$f(\theta) = MSE(\mathbf{y}_{val}, \hat{\mathbf{y}}_{val}) + \rho|\theta| \tag{3.5}$$

where,

$$\hat{\mathbf{y}}_{val} = ELM(\theta, \tilde{N}, Z_{trn}, Z_{val}) \tag{3.6}$$

and Z_{trn} and Z_{val} correspond to two separate training and validation sets, $\tilde{N}$ is the number of hidden nodes, and ρ is a regularisation parameter which will penalise large subsets of features.

Applying this notation and using the simulated annealing algorithm, the proposed new feature selection algorithm is shown in algorithm 1 (Simulated Annealing with Extreme Learning Machine - SANELM).

Pre-analysis for Hyper-parameters Determination

Before applying algorithm 1, some important parameters have to be accurately tuned. For example, the number of hidden nodes $\tilde{N}$ and the initial temperature T_0. For this task, a pre-analysis of the data

Algorithm 1 SANELM

```
Require: Initialize $\theta_0 \in \Theta$ and $T_0$ the initial temperature
Generate a model with $ELM(\theta_0, \tilde{N}, Z_{trn}, Z_{val})$
Compute $f(\theta_0)$, and put $\theta_{cur} = \theta_0$
for $i = 1$ to $STOP$ do
    Compute $T_{new} = Ann(T_0, i)$
    Generate $\theta_{new}$ in the neighbourhood of $\theta_{cur}$
    Compute $f(\theta_{new})$ and $\Delta f = f(\theta_{new}) - f(\theta_{cur})$
    if $\Delta f \leq 0$ then
        Accept $\theta_{new}$: $\theta_{cur} \leftarrow \theta_{new}$
    else
        Generate $U$ uniformly in $[0,1]$, and compute $P = \exp(-\Delta f / T_{new})$
        if $U \leq P$ then
            Accept $\theta_{new}$: $\theta_{cur} \leftarrow \theta_{new}$
        else
            Reject $\theta_{new}$
        end if
    end if
end for
```

sets is proposed in the following form:

- generate a random state $\theta \in \Theta$,
- evaluate the MSE of ELM using this θ for different numbers of hidden nodes $\tilde{N}$ (the range of the space search for $\tilde{N}$ can be determined during the process by trial and error).

Performing this process many times (about 1000 times) allows to find optimal number of hidden nodes for each particular data set. Then, SANELM algorithm is applied using only the optimal number of hidden nodes $\tilde{N}^*$.

An important by-product of this pre-analysis is the evaluation of the data variability which gives valuable information about the initialization of the temperature parameter T_0. As shown in Press et al. (2007) and applied in Filippone, Masulli, and Rovetta (2011), the pre-analysis step allows to determine T_0 according to the data set.

Some SANELM Details

Once the preprocessing task is completed, several SAN parameters have to be fitted. The first one is the annealing schedule $Ann(T_0, i)$. Written as a function of the initial temperature T_0 and the iteration index i, the schedule can take different forms. No preferential function exists, but as the optimization space Θ is discrete and not continuous, a basic schedule can be considered such as:

$$Ann(T_0, i) = \frac{T_0}{c \cdot i} \qquad \text{or} \qquad Ann(T_0, i) = \frac{T_0}{c \cdot \log(i)}, \tag{3.7}$$

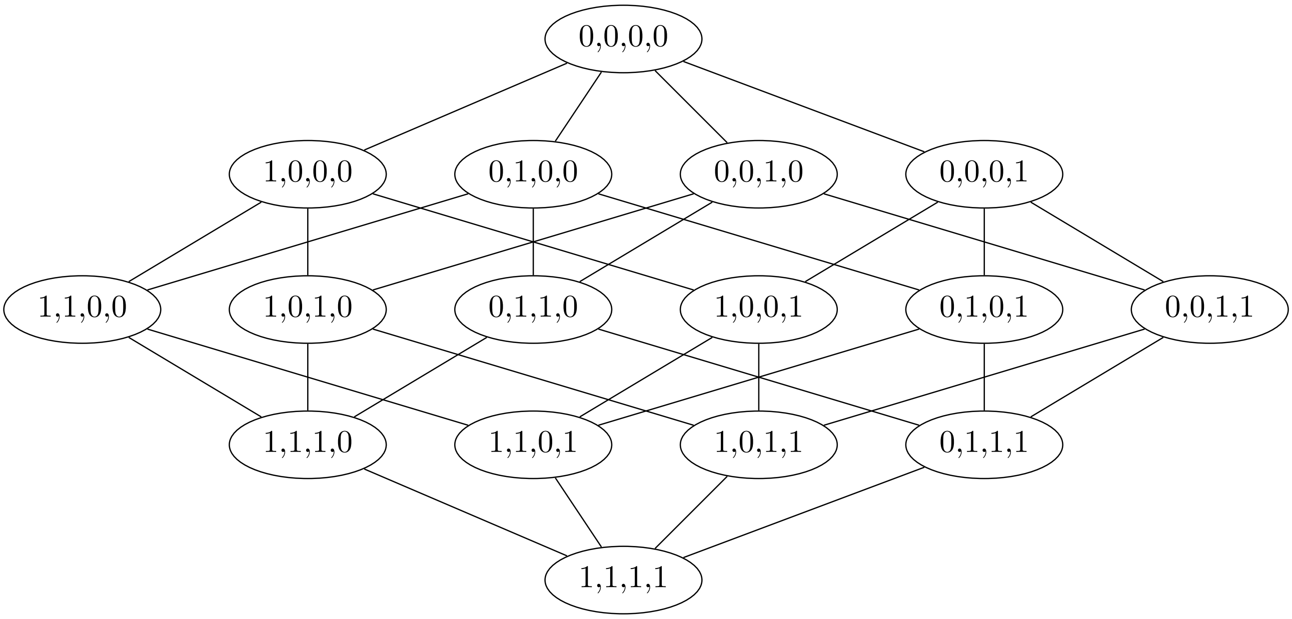


FIGURE 3.4: State space with $r = 1$ for the neighbourhood definition, and 4 input variables.

where c is the parameter of the schedule. In practice, since T_0 and c have to be parametrized, the most simple way is to fix $c = 1$ and to fit the parameter T_0 according to the pre-analysis results.

Another important process in the algorithm is the generation of a new state $\theta_{new} \in \Theta$ in the neighbourhood of the current state θ_{cur}. For this purpose, it is recommended to use the following Hamming distance (Hamming, 1950)

$$d(\theta_{new}, \theta_{cur}) = \# \left\{ i \mid \theta^i_{new} \neq \theta^i_{cur} \right\}, \tag{3.8}$$

and to consider as neighbourhood of θ_{cur} the following set:

$$B_r(\theta_{cur}) = \{\theta \in \Theta \mid d(\theta, \theta_{cur}) \leq r\}, \tag{3.9}$$

with $r = 1$. With this definition, it is possible to reach any state of the Θ space in at least n steps (where n is the number of input variables). Figure 3.4 highlights the state space when considering $r = 1$ for the definition of neighbourhood with 4 input variables.

3.2.3 Data and Results

Data

The data used for performing and evaluating the proposed methodology were taken from the UCI (University of California, Irvine) machine learning repository (Bache and Lichman, 2013). The databases were selected according to their relevance for feature selection tasks and for their prevalence in other publications related to feature selection methods.

Data sets	# Attributes	# Instances	# Output labels
Sonar	60	208	2
Ionosphere	33	351	2
Diabetes	8	768	2
Heart	44	267	2
Parkinsons	22	197	2
Diabetes SIM	16	768	2

TABLE 3.3: Database from the UCI repository used for the evaluation of SANELM

Data sets	Best $\tilde{N}$	T_0
Sonar	23	5
Ionosphere	23	5
Diabetes	14	1.5
Heart	7	2
Parkinsons	22	2
Diabetes SIM	20	1.5

TABLE 3.4: Results of the pre-analyis showing the optimal number of nodes $\tilde{N}$ and the temperature parameter T_0

Table 3.3 shows details on the following data sets: Sonar mines vs rocks (Gorman and Sejnowski, 1988) (Sonar), Johns Hopkins University ionosphere database (Ionosphere), Pima Indians diabetes database (Diabetes), SPECTF heart data (Heart) and Parkinsons disease data set (Parkinsons). An additional "simulated" database was considered (Diabetes SIM). Composed of the 8 original variables of Diabetes data set, 8 simulated variables were added by shuffling the original 8 variables.

Experimental Setup

First of all, the whole database must be normalized in order to fit to the range $[0,1]$ within which ELM works (Huang, Zhu, and Siew, 2006). Secondly, because of the need to assess the ELM model at each iteration of the SAN algorithm and to evaluate the performance of the final subset of features, the database must be split into three subsets. To address this task, two k-fold cross-validations were implemented. First, 10-fold cross-validation splits the database and alternatively used onefold as a test (TST) and the remaining 9 folds as a validation-training set ($VALTRN$). Then, with the $VALTRN$ set a second 5 fold cross-validation is performed in order to create alternatively one validation set (VAL) with one fold and one training set (TRN) with the remaining 4 folds. Finally 50 runs of SANELM algorithm are performed using respectively the TRN set for training the model, the VAL set to validate each new state θ, and the TST set to evaluate the final reached stateθ.

Results

Results from the pre-analysis of each database are shown in table 3.4. According to the 1000 random states evaluated by ELM, boxplots of the optimal number of nodes are shown in figure 3.5. Applying

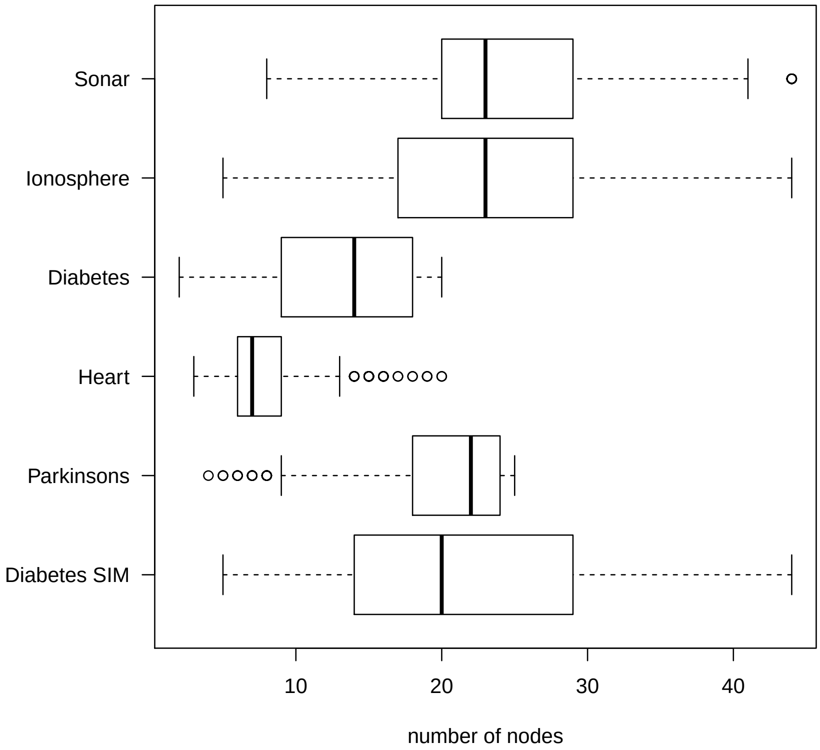


FIGURE 3.5: Boxplots of the optimal number of nodes of ELM model for each data set

parameters from table 3.4, general results of the SANELM algorithm are highlighted in table 3.5, where $\tilde{N}$, MNF, Err_{all} and Err_{best} indicate respectively the number of hidden nodes used, the number of features reached at the end of the search, the error of the model considering all the features and the error of the model with only the best features found. Due to the use of a two k-fold cross-validation process, the results show the mean ant the standard deviation.

As a representative example, detailed results from the Ionosphere database are shown in figures 3.6, 3.8 and 3.7. In figure 3.6 each dashed line corresponds to one random subset of features and the solid line coincides with the best subset of features. Examining 1000 random subsets of features reveals that the range of the number of hidden nodes where they reach the minimum value of MSE is approximately $[15,30]$ (highlighted in figure 3.7 with normalized MSE). Figure 3.8 shows the behaviour of the cost function $f(\theta)$ through one run of the SANELM algorithm.

Discussion

Results from table 3.5 highlight the fact that the proposed SANELM algorithm can reduce significantly the number of features without affecting the accuracy of the models. Moreover, in 4 cases (i.e. Sonar, Ionosphere, Parkinsons and Diabetes SIM) the models with the best subsets of features show an error lower than the models with all features. For the other databases, the errors computed with the best subset of features are equal to or greater than models with all features. But these differences are negligible according to the fact that the reduction of features is of primary importance. Let us

Data sets	$\tilde{N}$	MNF	Err_{all}	Err_{best}
Sonar	23	19.9 (3.8)	0.27 (0.07)	0.26 (0.07)
Ionosphere	23	10.2 (3.1)	0.14 (0.04)	0.12 (0.04)
Diabetes	14	3.4 (0.9)	0.23 (0.03)	0.24 (0.04)
Heart	7	3.1 (1.6)	0.2 (0.06)	0.21 (0.06)
Parkinsons	22	5.2 (2.2)	0.13 (0.06)	0.12 (0.06)
Diabetes SIM	20	3.9 (0.9)	0.25 (0.05)	0.24 (0.05)

TABLE 3.5: Results of the SANELM algorithm with the number of hidden nodes used ($\tilde{N}$), the mean number of features selected (MNF), the mean error with all features (Err_{all}) and the mean error with the best subset of features (Err_{best}). Mean (and standard deviation) are evaluated over 50 runs

note that the SANELM model applied on simulated data (Diabetes SIM with the added 8 shuffled variables) retrieves the same features than those with the original Diabetes database.

Another important result is the adjustment of the number of hiden nodes $\tilde{N}$. In the first stage of this research and in Leuenberger and Kanevski (2014) an additional loop was added in the SANELM algorithm in order to find the optimal number of hidden nodes for each new θ. Because this process is time consuming, an experimental analysis was carried out in order to determine the sensitivity of the SANELM algorithm according to the parameter $\tilde{N}$. For this purpose, 1000 ELM models with different subsets of features were analysed for each database. Results reveal that the optimal number of hidden nodes for each model are in the same range (an example is shown in figure 3.7 with the Ionosphere database, where the range of $\tilde{N}$ is between 15 and 30). According to these results, a comparison between SANELM algorithm with a fixed $\tilde{N}$ in this optimal range and SANELM algorithm without fixed $\tilde{N}$ was performed. Results show the same accuracy for both methods (for the number of selected features and for the error), but as mentioned before, the algorithm without fixed $\tilde{N}$ spends much more time. Therefore, the proposed pre-analysis in section 3.2.2 provides a suitable trade-off between computational time and effectiveness.

This section is focused on classification task. The regression problem with real and simulated data was considered in detail in Leuenberger and Kanevski (2014) and in Appendix A.1. The regression study dealt with environmental pollution data: 21 dimensional case study, 200 measured points with 3 known relevant variables. Using SANELM algorithm, within few thousand of iterations SANELM has converged to the true subset of features. For more details, see Appendix A.1.

Concluding Remarks

In this research, a new methodology for feature selection based on two algorithms, the Extreme Learning Machine as a wrapper method and the Simulated Annealing as an optimization algorithm, was developed. Comprehensive analyses were performed in order to investigate the behaviour of both ELM and SAN parameters. As the optimization space is a discrete one, the annealing schedule of SAN can be standard. For the remaining T_0 parameter, trial and error are needed according to the complexity

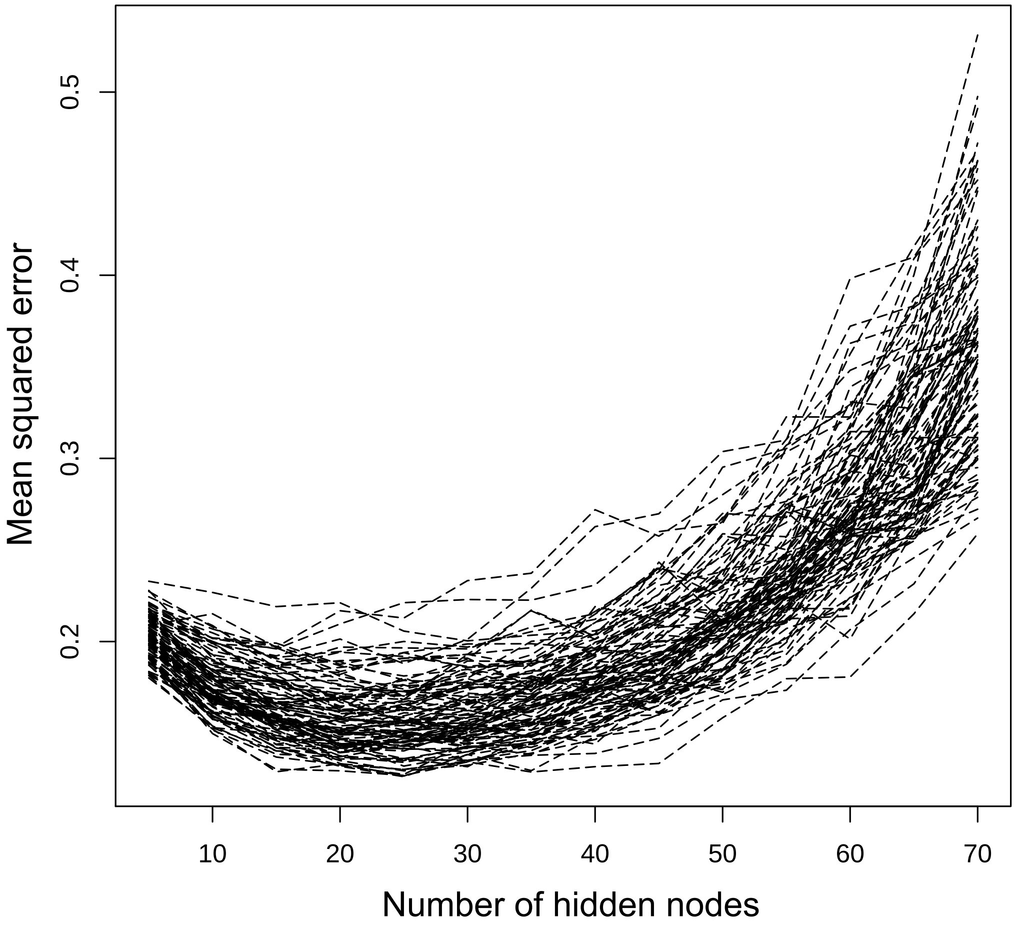


FIGURE 3.6: Each dashed line corresponds to one random subset of features. The graph shows the MSE of the ELM for these different subsets of features according to the number of hidden nodes (Ionosphere database).

and dimensionality of the problem. For the unique ELM parameter $\tilde{N}$ (the number of hidden nodes), it has been shown that it is quite stable within the range determined by the problem. Therefore, $\tilde{N}$ can be fixed during the process and computational time can be reduced. Final results show significant reduction of the number of features without affecting the general performance of the models. In future research, this benefit will allow to investigate more complex phenomena in high dimensional space and multivariate data, as well as to develop and compare other ELM-wrapper based models using different methods of global optimization.

3.3 Uncertainty

This section highlights a new theoretical development in the domain of uncertainty based machine learning. It collects several researches done for various conferences, such as the International Association for Mathematical Geosciences 2015 conference (IAMG) and the European Geosciences Union 2015 conference (EGU).

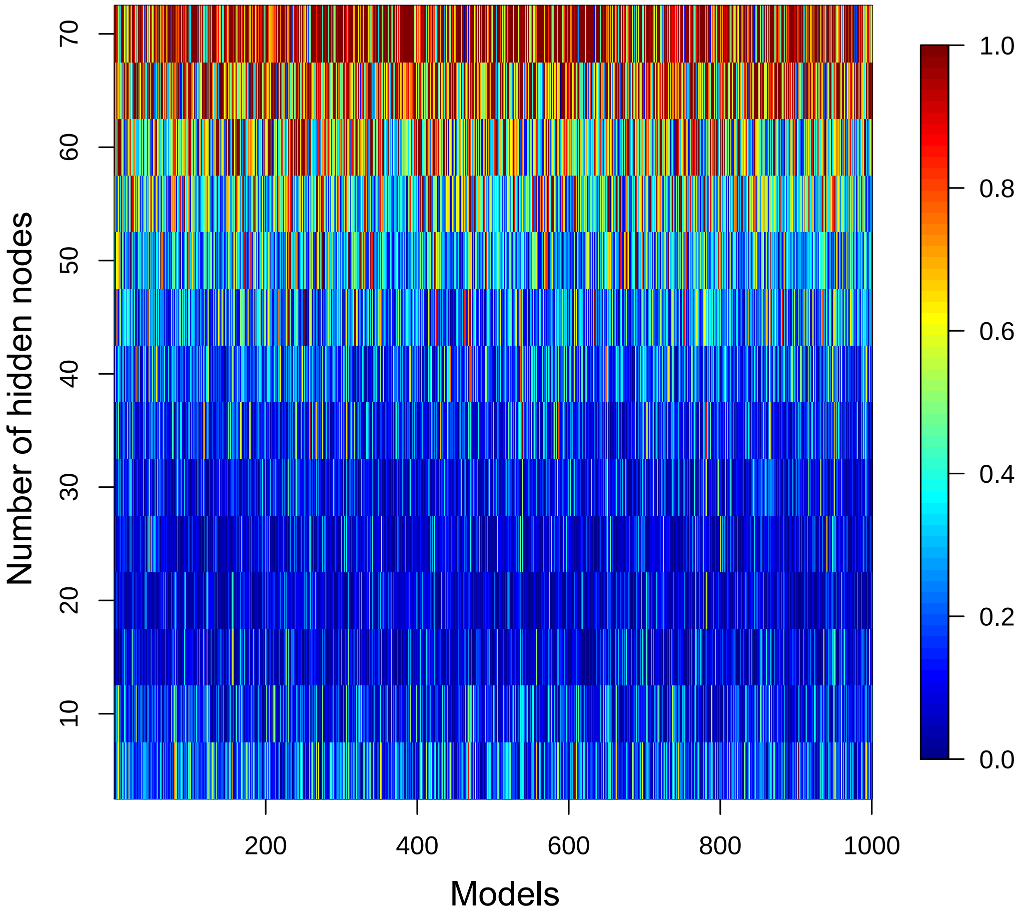


FIGURE 3.7: Behaviour of the optimal number of nodes among different θ in Ionosphere database. Although different θ are used, the range of optimal number of nodes is between 15 and 30.

Nowadays in environmental sciences we face the need to investigate more robust methodology and methods in order to analyse and understand the complex and non-linear phenomena under study. In addition to existing prediction tools, recent developments try to quantify the reliability of the generated predictions. It is well known that real and intelligent decision making process heavily resides on the quantification of uncertainties in data, models and predictions. In this regard, the main objective of the current research is to identify and quantify the different sources of uncertainties that can occur during the prediction process using Extreme Learning Machine along with a bootstrap-based approach. These different sources can be of two types: the uncertainty in the data and the uncertainty in the model.

3.3.1 Introduction

Due to large amount and complexity of data available nowadays in environmental sciences, we face the need to develop and apply more efficient and robust methodology, using both geostatistical and machine learning approaches (Kanevski and Maignan, 2004; Kanevski, Pozdnoukhov, and Timonin,

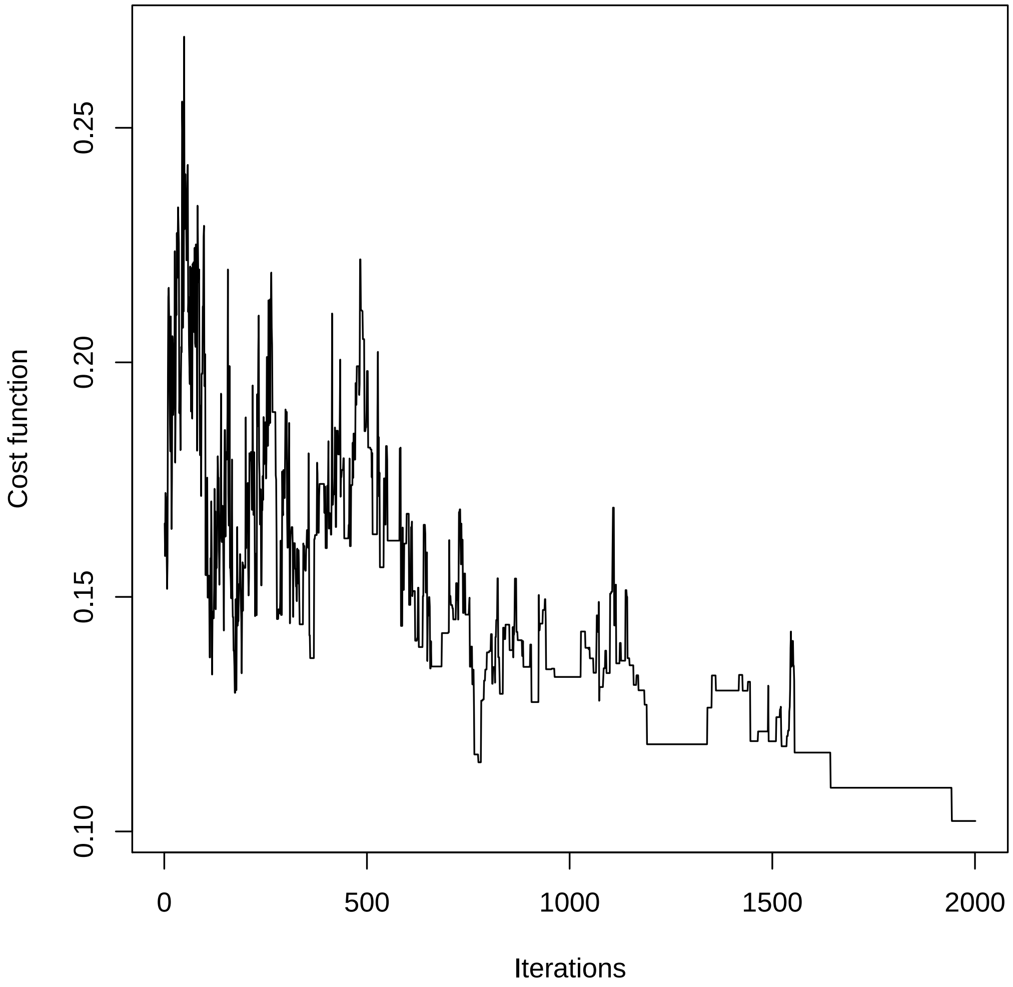


FIGURE 3.8: Behaviour of the cost function through one run of the SANELM algorithm (Ionosphere database).

2009). This will improve the design of monitoring networks, the data processing and the decision making process. One particular, but very important task of this development, is the reliability of generated prediction models, characterized, for example by confidence and prediction intervals. From the data collection to the prediction map, several sources of error can occur and affect the final results. These sources are mainly identified as uncertainty in data (data noise), and uncertainty in model. Their combination leads to the quantification of probabilistic prediction interval (essentially based on testing data). Understanding of these two categories of uncertainty allows a finer analysis of data and comprehension of phenomena under study and, finally, a better assessment of the prediction accuracy.

The main objective of this research deals with a development and adaptation of the methodology combining Extreme Learning Machine (ELM) with a bootstrap-based procedure (Efron and Tibshirani, 1986). Proposed by Wan et al. (2014) for the case of time series forecasting of wind power generation, this method is extended and adapted to the spatial environmental data.

Developed by Huang, Zhu, and Siew (2006) and detailed in section 3.1, ELM is an artificial neural network following the structure of a multilayer perception (MLP) with one single hidden layer. The particularity of ELM is that the weights between inputs and hidden layer are generated randomly and then a linear projection on the output space is performed. Compared to classical MLP, ELM is much faster and has the same property of being a universal approximator. There is only one hyper-parameter to be learned during the training procedure that is the number of nodes in the hidden layer.

The key step of the proposed approach to quantify the uncertainties are the following:

- sample from the original data a variety of subsets using bootstrapping,
- train and validate ELM models from these subsets,
- compute the ELM residuals.

Then, the same procedure is performed a second time but with the squared training residuals. Finally, taking into account the two modelling levels (on original and residual data) allows developing the prediction map, the model uncertainty variance, and the data noise variance. In order to better understand the method, the proposed approach was applied on simulated data with known uncertainties.

3.3.2 General Framework

For a given labelled set $Z = \left\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}\right\}_{i=1}^{N}$, let $\mathbf{x}_i$ be the input value and y_i the output value of the measured data points. The measured value can be modelled in the following form:

$$y_i = f(\mathbf{x}_i, \theta) + \varepsilon(\mathbf{x}_i) = t(\mathbf{x}_i) + \varepsilon(\mathbf{x}_i), \tag{3.10}$$

where $f(\mathbf{x}_i, \theta)$ is an optimal ELM model with parameter θ, $t(\mathbf{x}_i) = f(\mathbf{x}_i, \theta)$ is the true hidden function of the phenomenon, and $\varepsilon(\mathbf{x}_i)$ is the data noise at location $\mathbf{x}_i$. Now the first step resides in finding the optimal ELM parameter θ in order to have the best estimation of the true regression $t(\mathbf{x}_i)$:

$$\hat{t}(\mathbf{x}_i) = f(\mathbf{x}_i, \hat{\theta}) = \mathbb{E}\left[y_i \mid \mathbf{x}_i\right]. \tag{3.11}$$

Then, by considering the difference between the measured value and the estimation of the true regression, the following prediction error can be deduced:

$$y_i - \hat{t}(\mathbf{x}_i) = [t(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)] + \varepsilon(\mathbf{x}_i), \tag{3.12}$$

where $t(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)$ is the error of the ELM estimation with the true regression, $\varepsilon(\mathbf{x}_i)$ is the data noise, and $y_i - \hat{t}(\mathbf{x}_i)$ is the total prediction error. By using this notation, the total prediction error can be used for estimating the variance of the total prediction errors $\sigma_y^2(\mathbf{x}_i)$ as:

$$\sigma_y^2(\mathbf{x}_i) = \sigma_{\hat{t}}^2(\mathbf{x}_i) + \sigma_{\varepsilon}^2(\mathbf{x}_i), \tag{3.13}$$

where $\sigma_{\hat{t}}^2(\mathbf{x}_i)$ is the variance of the model uncertainty $t(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)$, and $\sigma_{\varepsilon}^2(\mathbf{x}_i)$ is the variance of the data noise $\varepsilon(\mathbf{x}_i)$.

From a practical point of view, we need some assumptions on the phenomenon in order to catch all types of uncertainty. These assumptions are the following:

- We assume that the noise in the data follows a normal distribution with variance $\sigma_{\varepsilon}^2(\mathbf{x}_i)$, i.e:

$$\varepsilon(\mathbf{x}_i) \sim \mathcal{N}(0, \sigma_{\varepsilon}^2(\mathbf{x}_i)). \tag{3.14}$$

- We assume that the two error components $t(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)$ and $\sigma_{\varepsilon}^2(\mathbf{x}_i)$ are independent.

According to this notation and theory, the next step resides in developing robust and coherent procedure in order to extract the required uncertainties.

In order to compute each component of equation 3.13, *BM* ELM models are generated by using an optimal parameter $\hat{\theta}$ and *BM* bootstrap samples from the training dataset. From this *BM* models the following values can be computed:

$$\hat{t}(\mathbf{x}_i) = \frac{1}{BM} \sum_{k=1}^{BM} ELM_k(\mathbf{x}_i, \hat{\theta}), \tag{3.15}$$

$$\sigma_{\hat{t}}^2(\mathbf{x}_i) = \frac{1}{BM-1} \sum_{k=1}^{BM} \left(ELM_k(\mathbf{x}_i, \hat{\theta}) - \hat{t}(\mathbf{x}_i)\right)^2. \tag{3.16}$$

It remains to find $\sigma_{\varepsilon}^2(\mathbf{x}_i)$. From equation 3.10 the following equation can be extracted:

$$\varepsilon(\mathbf{x}_i) = y_i - t(\mathbf{x}_i). \tag{3.17}$$

As $\varepsilon(\mathbf{x}_i)$ is not known, an estimation of this value can be computed:

$$\hat{\varepsilon}(\mathbf{x}_i) = y_i - \hat{t}(\mathbf{x}_i). \tag{3.18}$$

In this case, the variance of $\varepsilon(\mathbf{x}_i)$ can be estimated by the variance of $\hat{\varepsilon}(\mathbf{x}_i)$ with a correction factor:

$$\sigma_{\varepsilon}^2(\mathbf{x}_i) = \sigma_{\hat{\varepsilon}}^2(\mathbf{x}_i) - \sigma_{\hat{t}}^2(\mathbf{x}_i). \tag{3.19}$$

This correction is due to the fact that when we estimate $\varepsilon(\mathbf{x}_i)$ by $\hat{\varepsilon}(\mathbf{x}_i)$ we introduce a new source of variability with $\hat{t}(\mathbf{x}_i)$ in equation 3.18. Then, by definition of $\sigma_{\hat{\varepsilon}}^2(\mathbf{x}_i)$ we have:

$$\sigma_{\hat{\varepsilon}}^2(\mathbf{x}_i) = \frac{1}{n-1} \sum_{j=1}^{n} \left[(y_i - \hat{t}(\mathbf{x}_i))_j - \overline{(y_i - \hat{t}(\mathbf{x}_i))}\right]^2, \tag{3.20}$$

where

$$\overline{y_i - \hat{t}(\mathbf{x}_i)} = \overline{t(\mathbf{x}_i) + \varepsilon(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)} = \overline{t(\mathbf{x}_i) - \hat{t}(\mathbf{x}_i)} \cong 0. \tag{3.21}$$

Equation 3.21 can be explained by the fact that according to equation 3.14 the mean value of $\varepsilon(\mathbf{x}_i)$ is equal to zero, and the mean value of the best estimate model $\hat{t}(\mathbf{x}_i)$ is unbiased for a considered level. It is worth mentioning that the computation of the means in equation 3.21 and 3.20 should be seen as a new realisation of the measured values y_i. As mentioned in Heskes (1997), the bias component is negligible compared to the variance. Finally equation 3.20 can be estimated as:

$$\sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i) = \frac{1}{n-1}\sum_{j=1}^{n}(y_i - \hat{t}(\mathbf{x}_i))^2_j. \tag{3.22}$$

Now, instead of considering the measured data points $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, let consider $\{\mathbf{x}_i, R_i\}_{i=1}^{N}$, where $R_i = (y_i - \hat{t}(\mathbf{x}_i))^2$. By using BM_R ELM models with an optimal parameter $\hat{\theta}_R$, the following value can be computed:

$$\sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i) = \frac{1}{BM_R - 1}\sum_{k=1}^{BM_R} ELM_k(\mathbf{x}_i, \hat{\theta}_R), \tag{3.23}$$

In practice, as the predicted value of ELM can reach negative values (due to edge effect or in area with few points, as ELM is a continuous model, it can reach negative values where data points are close to zero), it is recommended to use the following equation:

$$\sigma^2_{\varepsilon}(\mathbf{x}_i) = \max\left(\sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i) - \sigma^2_{\hat{t}}(\mathbf{x}_i), 0\right) \tag{3.24}$$

3.3.3 Methodology

The following steps help to take into account all aspects highlighted in the theoretical part. The whole procedure is illustrated in figure 3.9

Step 1: Split the data between training (TRN) and testing (TST) sets.

Step 2: Apply a 10-fold cross-validation (CV) with the training set, in order to find the hyper-parameter of ELM, that is the number of node in the hidden layer $\tilde{N}_1$.

Step 3: Create 100 bootstrap subsets from the training data and use the optimal hyper-parameter $\tilde{N}_1$ in order to generate 100 ELM models (one for each bootstrap subset). These 100 models can be used latter for the prediction purpose, and the variance of the 100 models (i.e. $\sigma^2_{\hat{t}}(\mathbf{x}_i)$) is used for quantifying the model uncertainty.

Step 4: Use the squared residuals of the mean prediction of TRN (Res^{2}TRN) and apply a 10-fold cross-validation on these squared residuals in order to find $\tilde{N}_2$.

Step 5: Create 100 bootstrap subsets from the training data with the squared residuals and use $\tilde{N}_2$ in order to generate 100 ELM models. From these 100 models, an adapted mean value of the squared residuals prediction $\sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i)$ is retained for quantifying the uncertainty in the data.

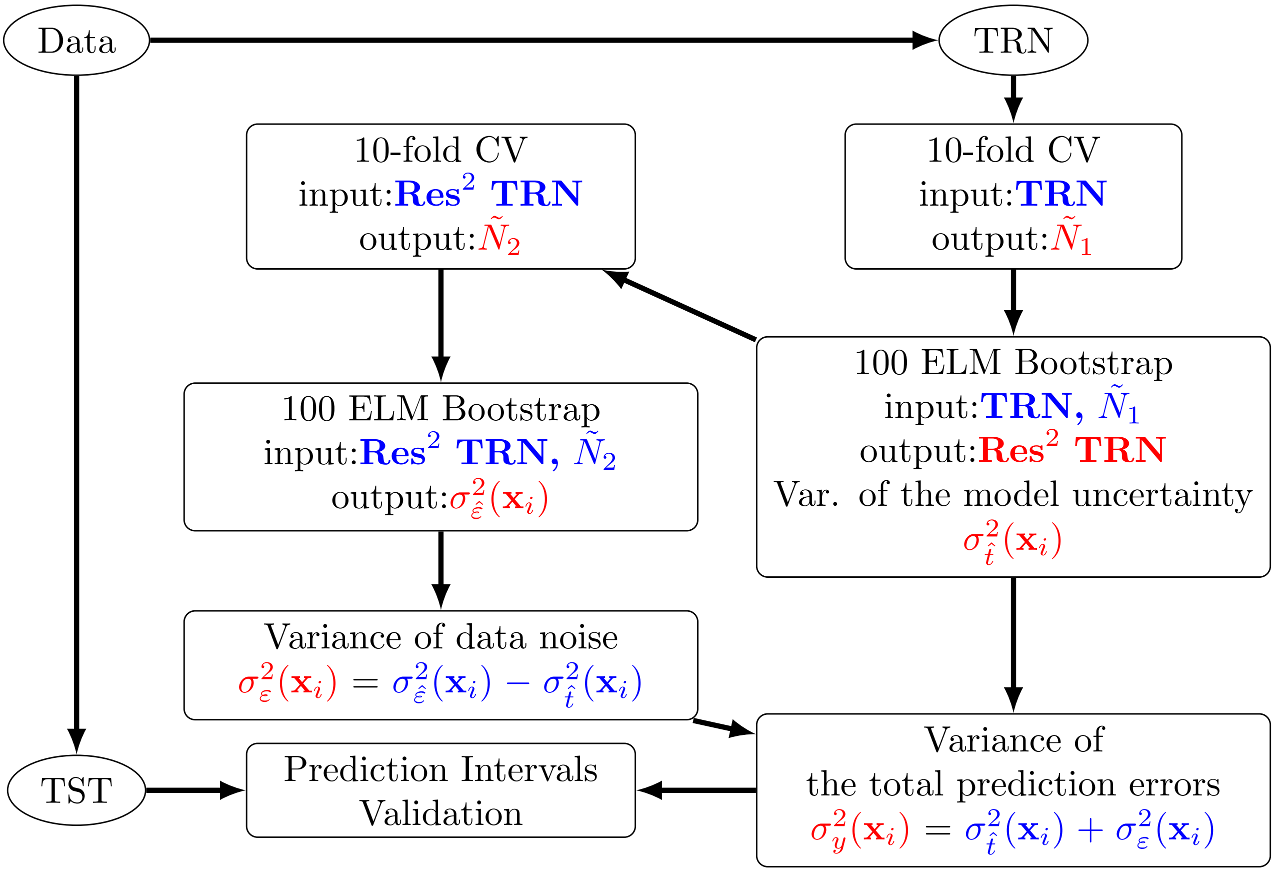

Data
TRN
10-fold CV
input:Res^2 TRN
output:$\tilde{N}_2$
10-fold CV
input:TRN
output:$\tilde{N}_1$
100 ELM Bootstrap
input:TRN, $\tilde{N}_1$
output:Res^2 TRN
Var. of the model uncertainty
$\sigma^2_{\hat{t}}(\mathbf{x}_i)$
100 ELM Bootstrap
input:Res^2 TRN, $\tilde{N}_2$
output:$\sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i)$
Variance of data noise
$\sigma^2_{\varepsilon}(\mathbf{x}_i) = \sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i) - \sigma^2_{\hat{t}}(\mathbf{x}_i)$
TST
Prediction Intervals
Validation
Variance of
the total prediction errors
$\sigma^2_{y}(\mathbf{x}_i) = \sigma^2_{\hat{t}}(\mathbf{x}_i) + \sigma^2_{\varepsilon}(\mathbf{x}_i)$

FIGURE 3.9: Methodology of the ELM bootstrap-based uncertainty. The dataset is split into training (TRN) and testing (TST) subsets. Then, from the TRN subset, a step by step process is proposed in order to extract the different kind of variances. Finally, TST subset is used to validate the obtained result.

Step 6: Compute the variance of data noise $\sigma^2_{\varepsilon}(\mathbf{x}_i) = \sigma^2_{\hat{\varepsilon}}(\mathbf{x}_i) - \sigma^2_{\hat{t}}(\mathbf{x}_i)$, and the variance of the total prediction errors $\sigma^2_{y}(\mathbf{x}_i) = \sigma^2_{\hat{t}}(\mathbf{x}_i) - \sigma^2_{\varepsilon}(\mathbf{x}_i)$. Finally, the testing set is used in order to validate and to quantify the reliability of different uncertainties.

3.3.4 Case Study and Results

The data used for this study is the butterfly dataset (Golay, Leuenberger, and Kanevski, 2017) and has been generated with a linear combination of non-linear functions (e.g. the sigmoid function). On this continuous and regular manifold, two sources of noise in coordinates (2,2) and (-2,-2) were added (see figure 3.10 on the left). In particular, the noise follows a normal distribution with a variance that is proportional to the distance of the two sources.

The main objective of the proposed methodology is to be able to model the background surface and to detect the two sources of noise.

The variance of the model uncertainty is displayed on the left part in figure 3.11. According to the color scale, we see that areas with few points of measurements affect significantly the variance of

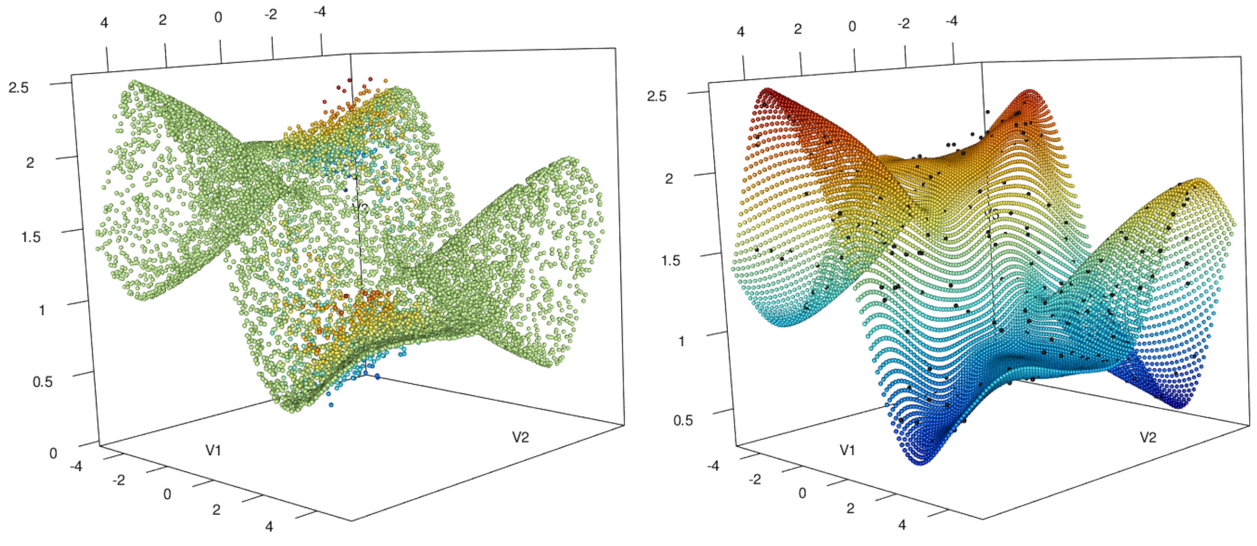


FIGURE 3.10: Original data with added noise (on the left), and mean prediction of the 100 ELM bootstrap models (on the right). Scale color on the left is proportional to the level of noise, while on the right it is proportional to the output variable.

the model uncertainty. For the variance of the data noise, which is displayed on the top right of figure 3.11, the proposed method can efficiently detect and highlight the generated noise. Finally, by taking into account both the variance of the model uncertainty and the variance of the data noise, we can see on the bottom of figure 3.11 the variance of the total prediction errors. In this final plot the degree of importance of each kind of uncertainty can be analysed and compared with the same color scale.

Note that the variance of the model uncertainty is highly sensitive to the edge effect. This is mainly due to the fact that at the edges there are less measurement points, which tends to quickly increase the variance of the model.

In figures 3.12 and 3.13 the same methodology was applied on the 198 measurement points of Lake Geneva. In this study, X, Y and Z coordinates were used as input variables in order to model the Nickel sediment pollutant as an output variable. Displayed on the top left of figure 3.12, the mean prediction of the 100 ELM bootstrap models can efficiently catch the main structure of the phenomenon. The variance of the model uncertainty (figure 3.12 on the top right) highlights with the high values (in red on the edge of the lake) the spatial area where the density of measurement points are low. The last map at the bottom displays the variance of the data noise which is essentially based on the quantification of the squared residuals.

Finally, figure 3.13 displays the variance of the total prediction error which is a composition of the two last variances. In order to visualize the relevance of the results, three test points have been extracted and the corresponding distribution curve (by considering the 100 ELM bootstrap models) are shown in figure 3.13 on the right.

3.3.5 Discussion

The combination of Extreme Learning Machine with a bootstrap-based procedure is very promising for modelling and quantification of uncertainties. Compared to classical machine learning algorithm like multilayer perceptron, ELM has the ability to learn faster without loss of accuracy, and needs only one hyper-parameter to be fitted. With these good properties, it allows us to generate 100 models for the normal training data and another 100 models for the squared residuals, which for standard algorithms is a training and computational problem.

One particular aspect of the proposed method is the use of the first assumption (noise that follows a normal distribution). As the variance is not a constant, it allows us to deal with heteroscedastic noise. But as this assumption is quite strong, further research needs to be carried out for different kinds of noise distributions.

According to the first results (figure 3.10 on the right), we see that the mean prediction of the 100 ELM bootstrap models can efficiently predict the desired surface without being affected by the added noise. Furthermore, this added noise has been correctly detected and highlighted in the variance of the data noise (figure 3.11 on the top right). For the variance of the model uncertainty, we can see that the density of the measurement points affects this variance. This aspect is certainly enhanced by the bootstrap-based procedure which generates subsets with some missing measurement points. Therefore, areas with low density of measurement points show higher variance for the model uncertainty.

3.3.6 Concluding Remarks

This study develops a new methodology which combines the Extreme Learning Machine with a bootstrap-based procedure for the quantification of the uncertainties in the data and in the model. Analyses were performed in order to investigate the behaviour of both ELM and bootstrap-based procedure for noisy data. It was shown that this method allows fast and accurate prediction and quantification of uncertainty.

An important contribution of this research deals with an elaboration of a self-consistent methodology that can be used for intelligent decision making process. It is worth mentioning that at this stage of the research the generated maps should be seen as an indicator based on visualization for identifying the spatial area where the data or the model have uncertainties. This means that future researches need to be done in order to investigate other noise sources and noise distributions.

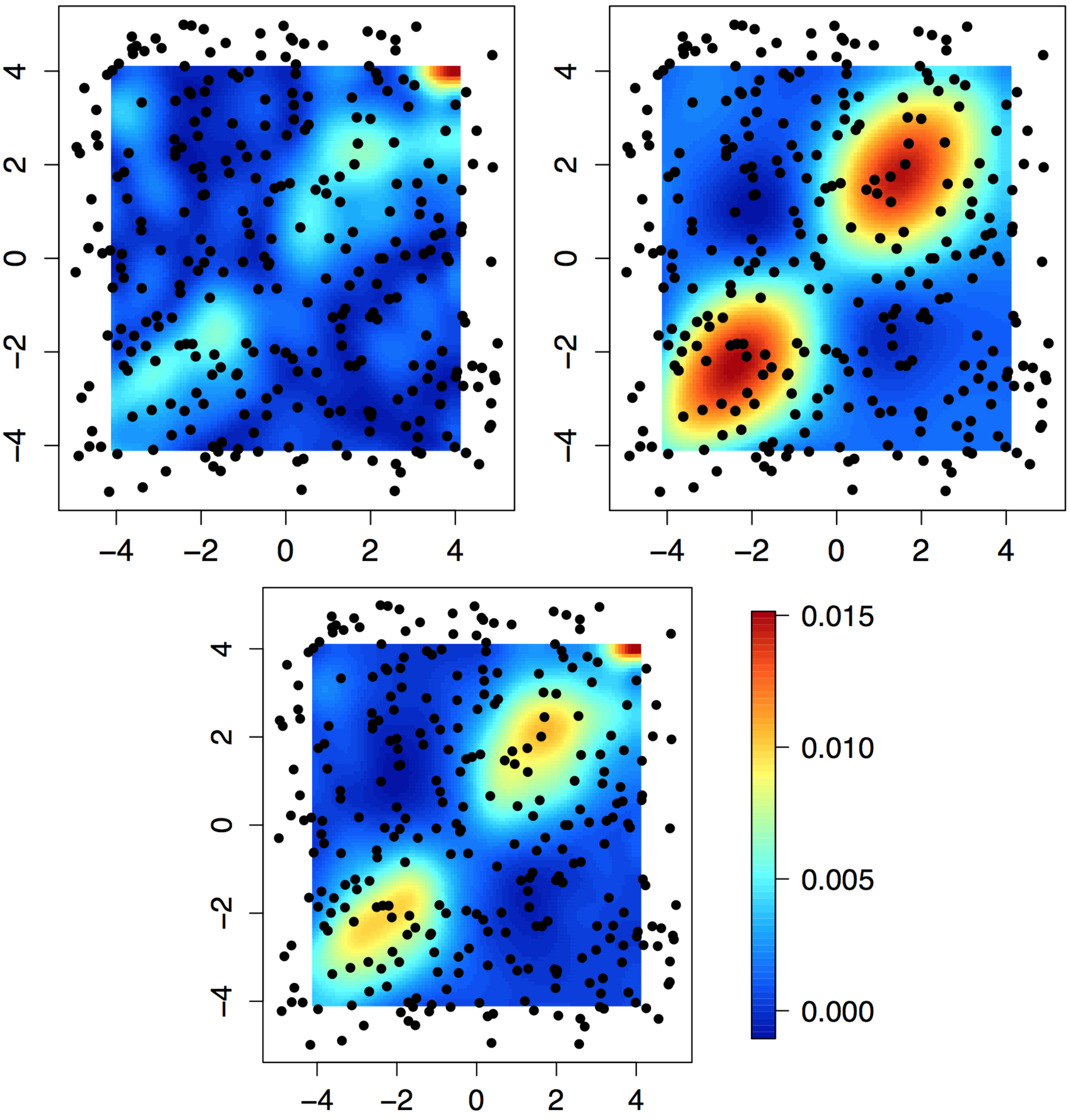


FIGURE 3.11: Visualization of the variance of the model uncertainty (on the top left), the variance of the data noise (on the top right), and the variance of the total prediction errors (at the bottom).

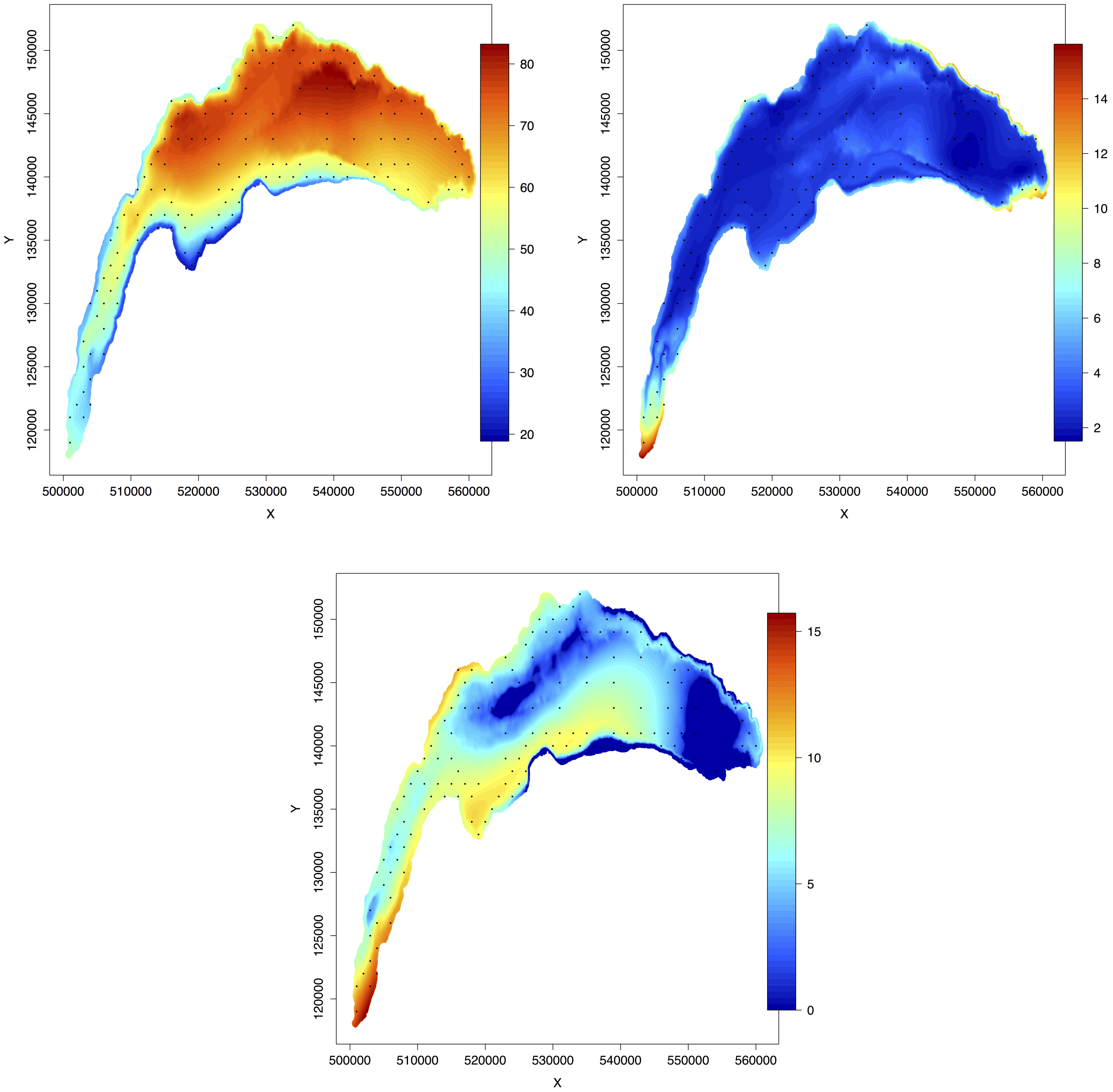


FIGURE 3.12: Mean prediction of the 100 ELM bootstrap models on the top left. Variance of the model uncertainty on the top right. And variance of the data noise at the bottom. The maps are linearly projected into the square $[0,1]^2$.

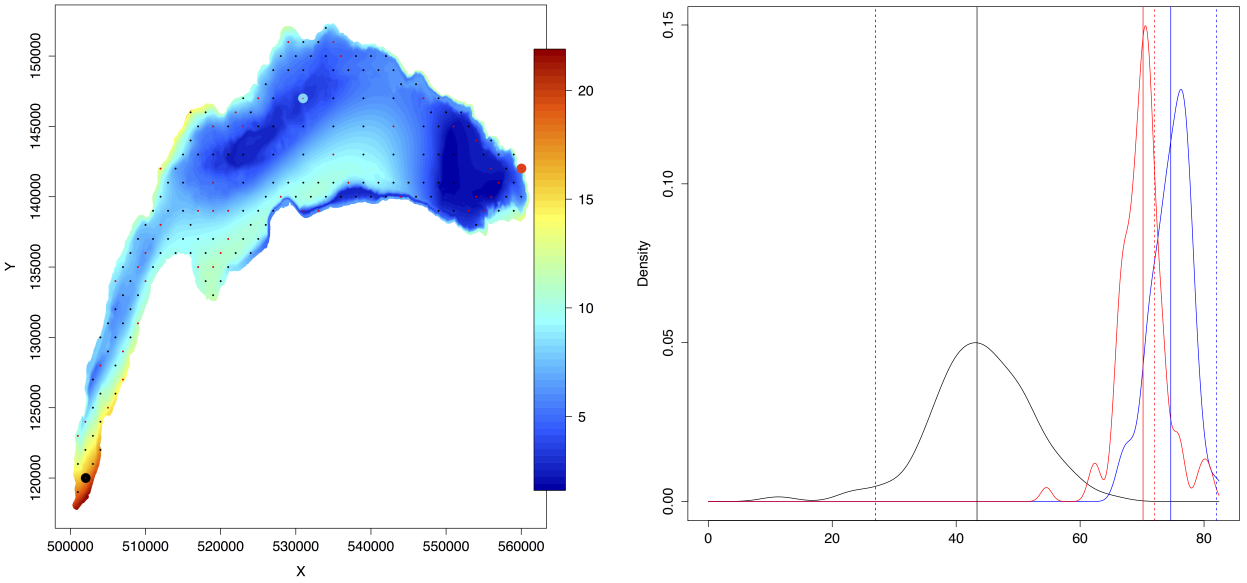


FIGURE 3.13: Variance of the total prediction error on the left. Distribution curves based on the 100 ELM bootstrap models on the right. Vertical continuous lines are the predicted mean value of the 100 ELM models, and dotted lines are the true value. These three curves are based on three test points denoted on the map in black, red and blue. The left map is linearly projected into the square $[0,1]^2$.

Chapter 4

Extreme Learning Machines for Spatial Environmental Data

M. Leuenberger and M. Kanevski, Extreme Learning Machines for spatial environmental data, Computers and Geosciences, 85, 64-73, 2015

Abstract

The use of machine learning algorithms has increased in a wide variety of domains (from finance to biocomputing and astronomy), and nowadays has a significant impact on the geoscience community. In most real cases geoscience data modelling problems are multivariate, high dimensional, variable at several spatial scales, and are generated by non-linear processes. For such complex data, the spatial prediction of continuous (or categorical) variables is a challenging task. The aim of this paper is to investigate the potential of the recently developed Extreme Learning Machine (ELM) for environmental data analysis, modelling and spatial prediction purposes. An important contribution of this study deals with an application of a generic self-consistent methodology for environmental data driven modelling based on Extreme Learning Machine. Both real and simulated data are used to demonstrate applicability of ELM at different stages of the study to understand and justify the results.

Keyword: Extreme Learning Machine, Spatial Environmental Data.

4.1 Introduction

Machine learning algorithms, principally based on statistical learning theory (Hastie, Tibshirani, and Friedman, 2009; Vapnik, 1998), being a universal non-linear modelling tools, play an important role in the modelling of environmental spatial data (Cracknell and Reading, 2014; Kanevski et al., 2004; Kanevski, Pozdnoukhov, and Timonin, 2009; Melchiorre and Abella, 2011; Micheletti et al., 2014; Nefeslioglu, Gokceoglu, and Sonmez, 2008). Recently, a new approach in machine learning, Extreme Learning Machine (ELM) (Huang, Zhu, and Siew, 2006), has gained a great popularity in the computer science community. For instance, it shows classification accuracies similar to Support Vector Machines (Chorowski, Wang, and Zurada, 2015), presents efficient capability with hyperspectral and uncertain data (Moreno et al., 2014; Sun, Yuan, and Wang, 2014), as well as in feature selection (Leuenberger and Kanevski, 2014).

ELM is a fast and powerful machine learning algorithm. It follows the structure of a multilayer perceptron (MLP) with just one single-hidden layer. The learning step of classical artificial neural networks, like MLP, deals with the optimization of weights and biases by using a variety of gradient-based or other (more complex) learning algorithms. Opposed to this optimization phase, which can fall into local minima and can have difficulty in converging, ELM generates randomly the weights between the input layer and the hidden layer and also the biases in the hidden layer. After this initialization, it just optimizes the weight vector between the hidden and the output layers by solving a relatively simple mathematical problem (see details below).

The main advantage of this algorithm is the speed of the training step and the ability to learn complex non-linear phenomena. Furthermore, by optimizing the number of hidden nodes during the training step, the algorithm can learn any set of training data with the desired precision (Huang, Zhu, and Siew, 2006). To avoid over-fitting, cross-validation methods or "true validation" (by splitting data into training, validation and testing subsets) are recommended in order to find the optimal number of hidden nodes (also called neurons). With its universal property and solid theoretical basis, ELM is an efficient machine learning algorithm which can push forward the field of environmental data analysis and modelling.

The present research addresses several essential methodological and applied problems of multivariate and high-dimensional spatial environmental data predictions using ELM. The first methodological part is focused on the influence of different numbers of hidden nodes when optimizing different tasks and on the analysis of the distances (similarities or dissimilarities) between the output weight vectors β in studying multivariate patterns. This new development can improve not only the accuracy of the prediction but also the comprehension of the phenomenon. The second methodological contribution deals with the analysis of ELM robustness and efficiency when working with noisy multivariate data and noisy features (noisy or non-relevant independent variables which, in case of simulated data, can be generated, for example, by permuting or shuffling raw data).

A real data case study was performed on three dimensional modelling of sediments pollution by heavy metals in Lake Geneva (CIPEL, 2008; Kanevski, Pozdnoukhov, and Timonin, 2009). The relationships between pollutants are quite complex (from linear correlations to non-linear dependences) which makes this problem quite challenging and useful for calibrating and validating the proposed approach.

First results on the analysis of the optimal number of nodes show in detail the range of complexity for each pollutant. Considered as a complexity spectrum, these values allow combining tasks (multivariate modelling) which can improve both the prediction accuracy and the understanding of hidden interactions between dependant variables. One of the results is the comparison of the output weight vector β between pairs of pollutants. When two pollutants show close behaviour in terms of complexity (the optimal number of hidden nodes), computing the so-called β-distance can disclose clear interrelations at different scales of complexities for each type of dependences encountered in data.

In order to better present and understand the methodology and the results, several simulated data with pre-defined structures were generated and a geostatistical tool (variography) was widely applied for both raw data and the residuals of ELM modelling.

The applied part of the research (real and simulated data case studies) follows the generic approach, consisting in several important steps, proposed in (Kanevski, 2013): exploratory (spatial) data analysis; data preprocessing and data splitting; training and selection of models; models evaluation/testing including comprehensive analysis of the residuals; predictions and uncertainty characterization.

The remainder of the paper is organized as follows. Section 2 briefly introduces the general theory of ELM, the datasets and the scenarios used. Section 3 presents the methodology step-by-step. In Section 4 the main results as well as the discussion are presented and Section 5 concludes the paper and provides future research directions.

4.2 Method and Data

4.2.1 General ELM Theory

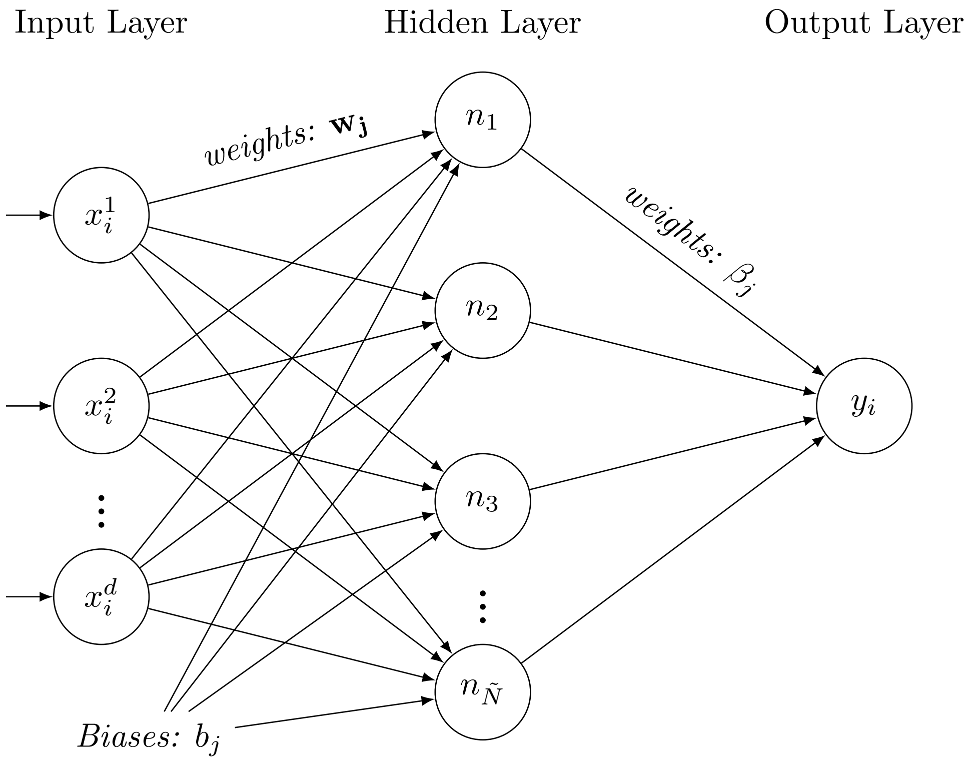


FIGURE 4.1: Extreme Learning Machine Structure.

Developed by Huang, Zhu, and Siew (2006), the Extreme Learning Machine algorithm (ELM) follows the structure of a single-hidden layer feedforward neural network (SLFN) (Fig. 4.1).

Let $(\mathbf{x}_i, y_i)_{i=1,...,n}$ be n data points, where $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^d)^T \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. For a fixed number of hidden nodes $\tilde{N}$, ELM generates randomly the weights $\mathbf{w}_j$ $(j = 1, .., \tilde{N})$ connecting the input layer to the hidden layer and the biases b_j ($j = 1,..,\tilde{N}$) of each node. The next step resides in computing the $n \times \tilde{N}$ matrix consisting of the ouputs from the hidden layer. This matrix has the following form:

$$H_{ij} = g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j),$$

where $g : \mathbb{R} \to \mathbb{R}$ is an infinitely differentiable activation function of hidden nodes.

Each row of the matrix H corresponds to a hidden layer's output for one of the n input data vectors. As the matrix H is completely defined, vector β connecting the hidden layer to the output layer is estimated by using the Moore-Penrose generalized inverse of the matrix H:

$$\hat{\beta} = H^{\dagger}\mathbf{y}.$$

Finally, when all weights and biases of the network are defined, new data points can be predicted and the testing error evaluated.

Notice that ELM allows fast and accurate predictions and it was proven to be a universal modelling tool (Huang, Zhu, and Siew, 2006). Furthermore, the algorithmic complexity of ELM lies in the computation of the Moore-Penrose generalized inverse of matrix H, which in this case uses the singular value decomposition algorithm (SVD). This means that the quickness of ELM refers just for the computation of weights. The optimization phase, which resides in finding the optimal hyperparameter (the number $\tilde{N}$ of hidden nodes for ELM), does not differ from other machine learning algorithms.

The following study was performed by using the *elmNN* package of the R language (R Core Team, 2016).

4.2.2 Dataset Description

The data used for this study were provided by CIPEL (2008) and they are composed of 200 measurements of sediment pollution by heavy metals in Lake Geneva. In addition to the information about X, Y coordinates and the elevation (Fig.4.2), each point of the dataset has the concentration (micro g/g) of the following pollutants: Mercury (Hg), Zinc (Zn), Copper (Cu), Titanium (Ti), Chromium (Cr), Vanadium(V), Nickel (Ni) and Cadmium (Cd).

Because of the wide variety of relationships in the dataset between pollutants (Fig.4.3) it provides an interesting and challenging case study.

Four scenarios have been created in order to validate and test the proposed approach. These scenarios are presented in detail in table 4.1, where X and Y correspond to the geographical coordinates, Z denotes the elevation, and shuffled or "sh." means that data for the corresponding variable has been randomly permuted. The shuffling (or random permutation) was carried out for input or output variables (see scenarios in table 4.1).

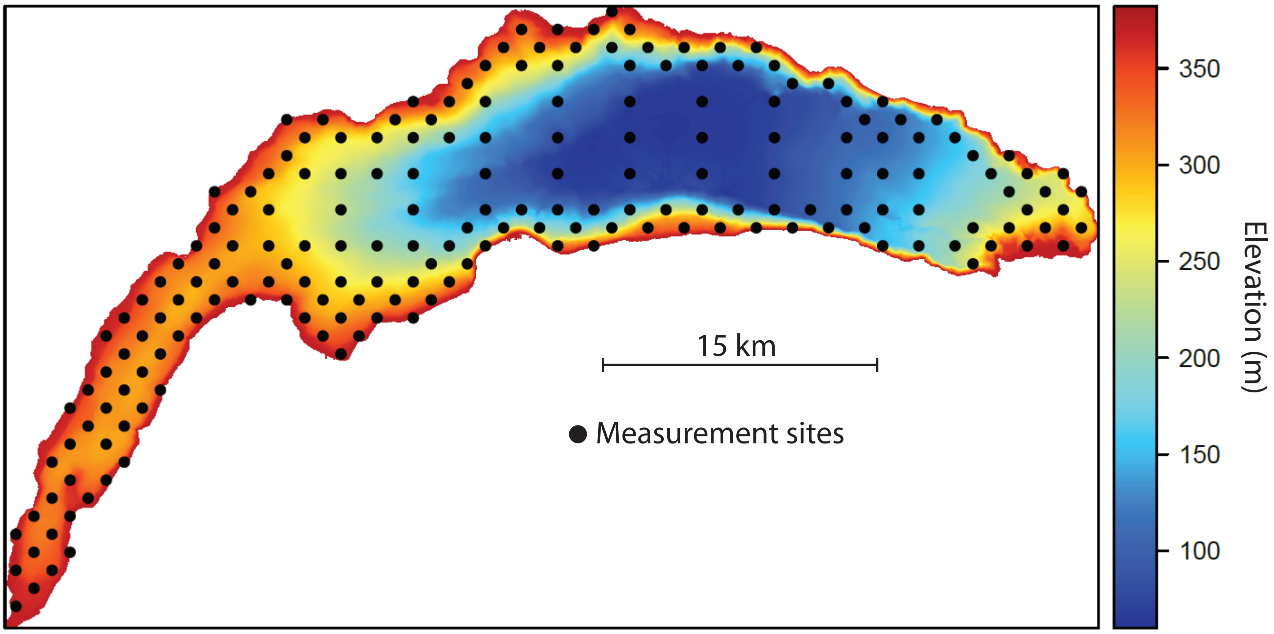


FIGURE 4.2: Lake Geneva with elevation map (m) and measurment points.

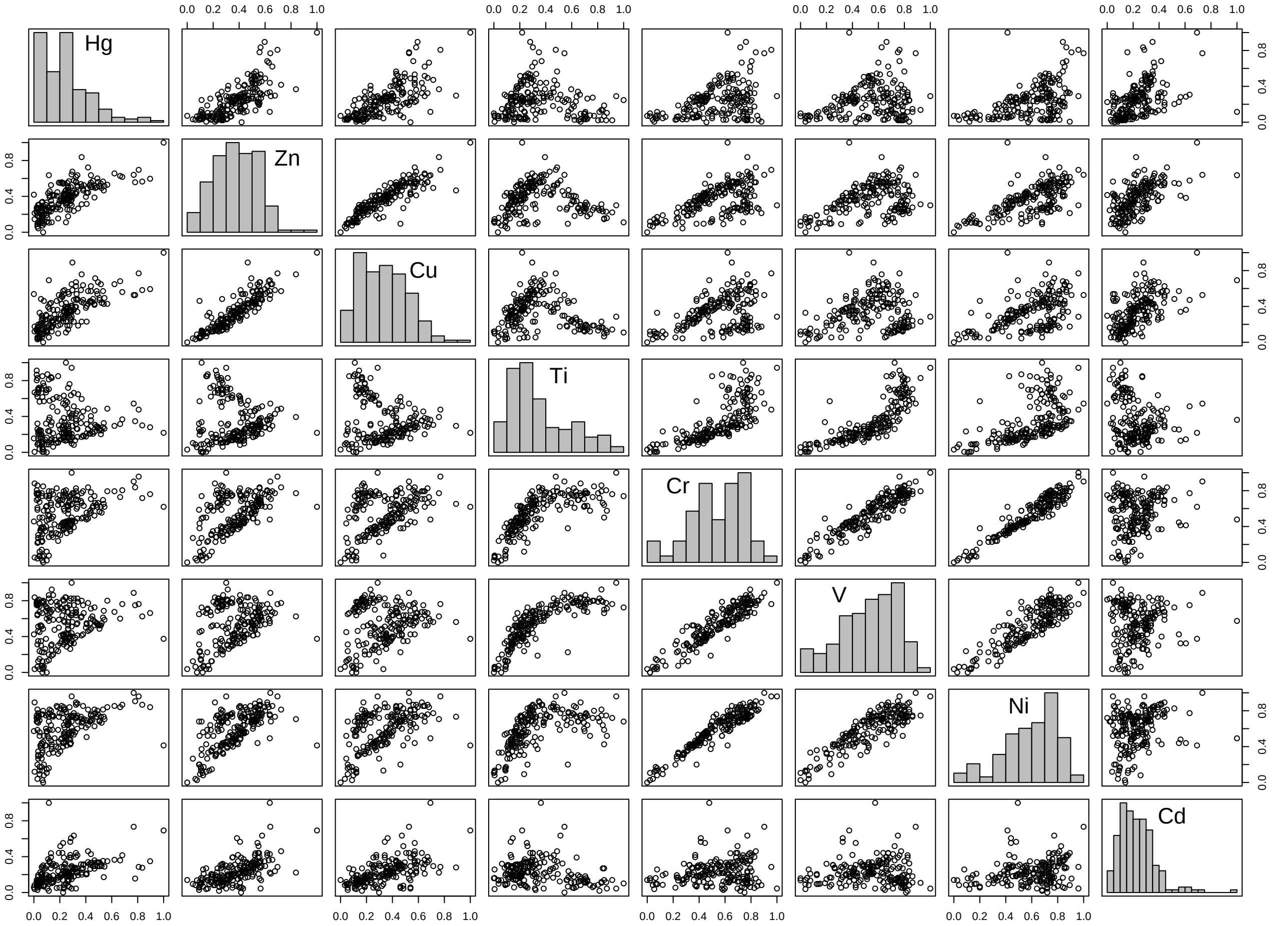


FIGURE 4.3: Scatterplot of heavy metal concentration (normalized).

Scenarios	Input variables	Output variables
3D-Ni	X, Y, Z	Nickel
3DSH-Ni	X, Y, Z	Nickel shuffled
6D-Ni	X, Y, Z, X sh., Y sh., Z sh.	Nickel
3D-Zn	X, Y, Z	Zinc

TABLE 4.1: Composition of the different scenarios.

The 3D-Ni scenario was chosen in order to apply the standard methodology in a straightforward way. The use of 3DSH-Ni (with shuffled Nickel) allows us to assess the behaviour of ELM when there is no dependence between input and output variables.

Then, 6D-Ni scenario is used to evaluate the efficiency of ELM with noisy and non-relevant features, as well as to assess the capability of ELM dimensionality reduction by evaluating different combinations of input variables. Finally, the 3D-Zn scenario is considered in order to compare it with the 3D-Ni scenario. Also some results were generalized using multi-output ELM.

4.3 Methodology

The research follows a generic methodology of the application of machine learning algorithms for spatial multivariate and, in general, high dimensional data presented in Kanevski (2013). The methodology consists of several important steps from exploratory (spatial) data analysis (EDA/ESDA) to predictions and uncertainties characterization.

In general, the preparation of the input space (independent variables or features), when considering real complex environmental problems, for example, landslides, avalanches, forest fires, etc., is a difficult task. Usually this space is constructed using expert and science-based knowledge and can be either incomplete or redundant. Therefore, feature selection is an important task (Micheletti et al., 2014). In the next sections, by following the complete methodology, we concentrate the presentation only on the new and the most relevant properties of ELM and corresponding results.

4.3.1 Data Preprocessing

The first steps, as it was mentioned above, deal with the analysis of monitoring network (clustering and preferential sampling); exploratory (spatial) data analysis (raw data visualization, analysis of distributions and relationships between variables, variography, detection of patterns, noise estimation, etc.); data pre-processing (transformations of data); data splitting (random, stratified or using more complex tools); construction of input/feature space. In our case, the input space was constructed according to the scenarios presented above: 3d and 6d case studies covering proposed topics of the research. The variography was carried out only for 3d data. Let us remind that here the variography is used to complete understanding of data and the ELM results.

In the data pre-processing step, each variable was normalized into $[0,1]$ interval in order to fit the functional range where ELM efficiently works (Huang, Zhu, and Siew, 2006). This is achieved using the following equation:

$$X_{new} = \frac{X - min(X)}{max(X) - min(X)} \quad .$$

In general, pre-processing can include non-linear transformations of data and more complex tools, like principal component analysis (Haykin, 2008).

The next step in the pre-processing stage deals with the creation of several subsets: training, validation and testing (Hastie, Tibshirani, and Friedman, 2009; Kanevski, Pozdnoukhov, and Timonin, 2009). The splitting of the data is not a simple task. There are many approaches how to do it taking into account some properties of data and the objectives of the study: random and stratified splitting, application of self-organizing maps, etc. (May, Maier, and Dandy, 2010). Taking into account that the monitoring network is not clustered, in this research only a random splitting of data was applied. Thus, in order to evaluate the accuracy of the model generalization, a testing set (TST) was randomly extracted from the data. This set was only used at the end of the modelling procedure to assess the trained optimal ELM model. Then, if the amount of the remaining data is large enough, the creation of a second separate set is preferred. Called validation set (VAL), this second set aims to select the model by finding the best hyper-parameters that give the minimum validation error. The remaining data, called training set (TRN), are used to train the model. According to the literature (Hastie, Tibshirani, and Friedman, 2009), the split into three separate subsets TRN, VAL and TST can be performed respectively with the following proportions of the amount of available data: 50%, 25% and 25%. On the other hand, if a few data are available, the two sets TRN and VAL can be merged to form the $VALTRN$ set, and cross-validation approaches applied on the $VALTRN$ set are considered. For both cases, it is recommended to check the pertinence of the generated subsets compared to the original dataset, especially if the number of available data is not large enough. This can be performed by comparing, variable by variable, general statistics in a quantitative way (e.g. mean, standard deviation, kurtosis and skewness), or in a qualitative way by plotting histograms and variograms.

In this study two subsets have been randomly generated with respectively 150 data points for the $VALTRN$ set, and 50 for the TST set.

4.3.2 Model Training

The training part in machine learning deals with the search of optimal hyper-parameters. In the case of the ELM algorithm the only hyper-parameter to fit, which controls the complexity of the model, is the number of nodes $\tilde{N}$ in the hidden layer. In order to find this optimal number of nodes, the prevailing method resides in training several ELM models with different number of nodes $\tilde{N}$ in a predefined range, and in selecting the model with the lowest mean squared error. In a more formal way, when true-validation is used (i.e. with the TRN and VAL sets), the following steps are executed:

- By using the TRN set, train $\mathrm{ELM}_{\tilde{N}}$ with $\tilde{N}$ number of hidden nodes, where $\tilde{N} \in \{\tilde{N}_{min}, ..., \tilde{N}_{max}\}$ (in this study the parameters were $\tilde{N}_{min} = 1$ and $\tilde{N}_{max} = 100$).
- By using the VAL set, evaluate each $\mathrm{ELM}_{\tilde{N}}$ by computing the mean squared error (MSE):

  $$MSE(\mathrm{ELM}_{\tilde{N}}) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} (\hat{y}_{val,i} - y_{val,i})^2,$$

  where n_{val} is the number of data points in the validation set, $y_{val,i}$ are the output values of the validation set and $\hat{y}_{val,i}$ are the $\mathrm{ELM}_{\tilde{N}}$ estimated values of the validation set.
- Finally, the optimal number of hidden nodes $\tilde{N}^*$ can be defined as:

  $$\tilde{N}^* = \underset{\tilde{N} \in \{\tilde{N}_{min}, ..., \tilde{N}_{max}\}}{\arg\min} MSE(\mathrm{ELM}_{\tilde{N}}).$$

For the case where cross-validation is used (i.e. with the unique $VALTRN$ set), k folds are generated (i.e. the $VALTRN$ set is randomly split into k distinct subsets). Then, in an iterative way, one fold is selected as a VAL set and the remaining $\mathrm{k}-1$ folds as a TRN set. The above two first steps are repeated for each of the k pairs of TRN and VAL sets generated by the k folds, and the mean values of the k $MSE(\mathrm{ELM}_{\tilde{N}})$ are used in order to find the optimal number of nodes $\tilde{N}^*$.

4.3.3 Model Evaluation

Once the optimal number of nodes $\tilde{N}^*$ is determined, a final ELM model with $\tilde{N}^*$ nodes can be generated using both TRN and VAL sets. This final and optimal ELM model is evaluated with the TST set and the following residuals are analysed:

$$Res_{tst,i} = \hat{y}_{tst,i} - y_{tst,i},$$

where $y_{tst,i}$ are the output values of the testing set and $\hat{y}_{tst,i}$ are the ELM estimated values of the testing set. Basic statistics, such as minimum, maximum, mean values, standard deviation, skewness and kurtosis, can be computed from the residuals. Furthermore, accuracy plot ($\hat{y}_{tst}$ versus y_{tst}), mean square error and correlation measures can provide valuable information about the performance of the ELM model.

4.3.4 Learning the Residuals

The analysis of the residuals (training, validation, testing) is an extremely important step of the study. This procedure allows us to understand and to quantify the quality of data-driven modelling by analysing the distributions and the presence or absence of structures (patterns) in the residuals. Such analyses also help to avoid over-fitting of data and to have good predictions (Kanevski et al., 2004;

Kanevski, Pozdnoukhov, and Timonin, 2009). It is worth noting that the analysis of the residuals is not specific to ELM, but it is an essential step for all machine learning algorithms or data-driven models (Kanevski, Pozdnoukhov, and Timonin, 2009; Kanevski, 2013).

There are many tools which can be used to discriminate between patterns (structured information) and non-structured noise. For the purpose of learning the residuals, methods which can learn quickly with low complexity (0 or 1 hyper-parameter) are advised.

For example, k-nearest neighbours (kNN) and general regression neural networks are easy to implement and to understand (Kanevski, Pozdnoukhov, and Timonin, 2009), and therefore can be applied in order to detect remaining information in the residuals. In geostatistics it is the variogram which efficiently discriminates spatially correlated data from noise (pure nugget effect). In a more general framework, different methods exist and are well established like the gamma test, the delta test and others (Liitiäinen et al., 2009). In the present research kNN and variography are used, as well as ELM trained on non-structured data.

When using k-nearest neighbours algorithm on the residual data, the following two cases can occur (Kanevski et al., 2004):

- cross-validation plot (k versus MSE) shows a minimum. It means that kNN algorithm reveals some structures in the residuals.
- cross-validation plot shows no minimum.

In the first case, the fact that kNN observes information in the residuals means that the ELM model used is not optimal, ELM under-fits the data. On the other hand, when the cross-validation plot shows no minimum, the ELM model is complex enough to apprehend latent pattern in the data. However, this test does not prevent from over-fitting. But when the kNN cross-validation curve for large number of neighbours fluctuates around the level of nugget of raw data, the model did not over-fit. Therefore, it is very important to estimate nugget (noise) in original data. In general, this is a very difficult problem, especially in high dimensional spaces. Recently, some studies were carried out and some general recommendations were given (Liitiäinen et al., 2009).

In the same way as kNN, the use of a second ELM model on the residuals can provide more accurate knowledge about the remaining information in the residuals. Using a cross-validation plot ($\tilde{N}$ versus MSE), the same criteria as in kNN (presence/absence of minimum) can be used in order to determine if the first ELM model under-fits the data or not. Thus, the use of different methods and tools on the residuals help to validate and assess the accuracy of the model developed.

The last step consists in computing spatial predictions either on some study area or on a grid by using trained and tested ELM model.

4.3.5 Multivariate ELM

The big advantage of the ELM algorithm is the fact that it can learn quickly and it has just one hyper-parameter. In the special case where the analysis is focused on several output variables according to the same input space, ELM model can provide valuable information about the nature of the relationships between output variables.

One additional analysis, which can be performed with ELM, is the investigation of the optimal number of hidden nodes $\tilde{N}^*$.

Because $\tilde{N}^*$ represents the structure complexity of the neural network, it also reflects the complexity of the relationship between input and output variables. Now, if two problems (for example 3D-Ni and 3D-Zn) have the same or relatively close $\tilde{N}^*$, it would mean that they have the same level of complexity. Thus, $\tilde{N}^*$ can be described as a complexity index of the case study.

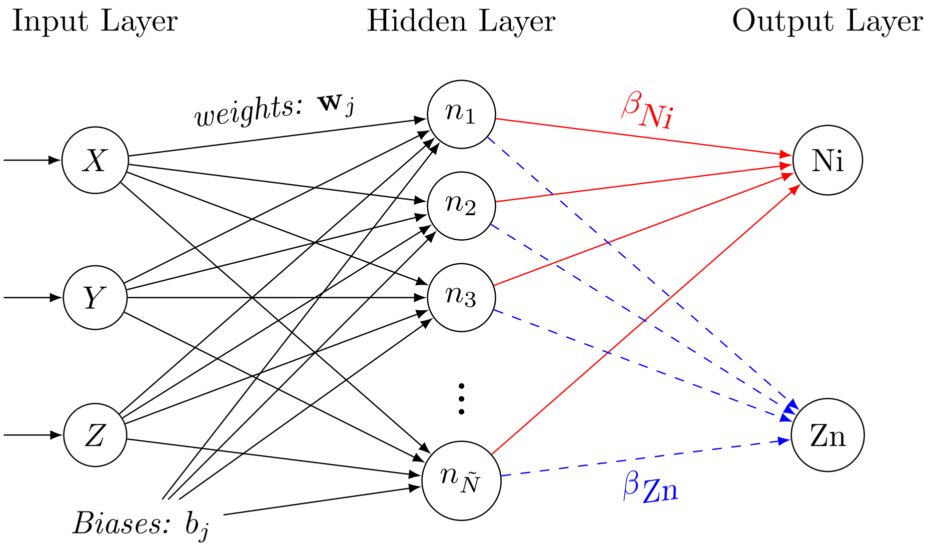


FIGURE 4.4: ELM structure with two output variables.

Related to this complexity index, an analysis of the β-distance is proposed.

When two output variables have relatively close optimal $\tilde{N}^*$, the use of one identical ELM model can be investigated. By generating the same matrix H (which is the output matrix of the hidden layer) for both output variables, the resulting β vectors can be analysed, and, in particular, the distance between them. For example, if 3D-Ni and 3D-Zn have the same optimal $\tilde{N}^*$ (figure 4.4), then the following β can be computed using the Moore-Penrose generalized inverse of matrix H:

$$\beta_{\mathrm{Ni}} = H^{\dagger}\mathbf{y}_{\mathrm{Ni}},$$

$$\beta_{\mathrm{Zn}} = H^{\dagger}\mathbf{y}_{\mathrm{Zn}}.$$

And, the Euclidean distance between β_{Ni} and β_{Zn} can be described as a dissimilarity index. If the index of dissimilarity is close to zero, this would mean that the two output variables are in a sense related in a non-linear or linear way. On the other hand, if the index shows large value, this would mean that the considered two output variables have independent behaviour relative to each other.

Another analysis which can take advantage of the performances of ELM is feature selection.

When dealing with environmental data, it can occur that the available input variables are redundant or irrelevant. In this case, feature selection methods can be applied in order to detect and remove non relevant input variables. There are a lot of methods for the task of feature selection (Guyon and Elisseeff, 2003). On the one hand, filter methods are fast but do not necessarily take into account the combinations of various features simultaneously (a feature can be irrelevant alone but may be relevant with other features together). On the other hand wrapper methods allow the evaluation of different subsets of input variables (Kohavi and John, 1997).

Then, optimal subset can be selected when the mean squared error is minimum. In this study, capabilities of ELM as a wrapper method are explored and presented using the 6D-Ni scenario.

4.4 Results and Discussion

This section highlights and discusses the results obtained with the proposed methodology applied to the different scenarios of the Lake Geneva dataset.

Figure 4.5 shows the results obtained by using a 5-fold cross-validation repeated 20 times for 3D-Ni, 3DSH-Ni, 6D-Ni and 3D-Zn scenarios. For these scenarios, the optimal number of nodes and the mean square error for the validation and training sets are given in table 4.2. The results from the 3D-Ni and 3D-Zn scenarios show a conventional behaviour. For both scenarios, ELM reaches a precise minimum with the validation set. In the case of 6D-Ni scenario (where 3 irrelevant variables were added), the validation curve shows similar trend as in 3D-Ni scenario, but with a slightly higher MSE. On the other hand, validation curve of the scenario 3DSH-Ni presents scarcely a minimum when the number of nodes is small, but according to the high standard deviation for this scenario, the minimum can be neglected. Thus, no structure in data means no minimum on the curve which corresponds to the scenario selected.

Scenarios	3D-Ni	3DSH-Ni	6D-Ni	3D-Zn
$\tilde{N}^*$	15	3	14	16
TRN MSE	0.01390	0.04322	0.01985	0.00794
VAL MSE	0.01735	0.04503	0.02410	0.01232

TABLE 4.2: Results of the 3D-Ni, 3DSH-Ni, 6D-Ni and 3D-Zn scenarios.

For the scenarios 3D-Ni and 3DSH-Ni basic statistics on the residuals are given in table 4.3, as well as graphic representations in figure 4.6. Note that the residuals are present in figure 4.6 by considering the vertical distance between each points and the diagonal. From this point of view, results show two distinctive tendencies. Especially for the 3DSH-Ni scenario, figure 4.6 demonstrates that, when the data have no structure, the best predicted value is the mean value of the available data. According to these residuals, the second ELM model has been trained in order to detect if there is still an information in the data.

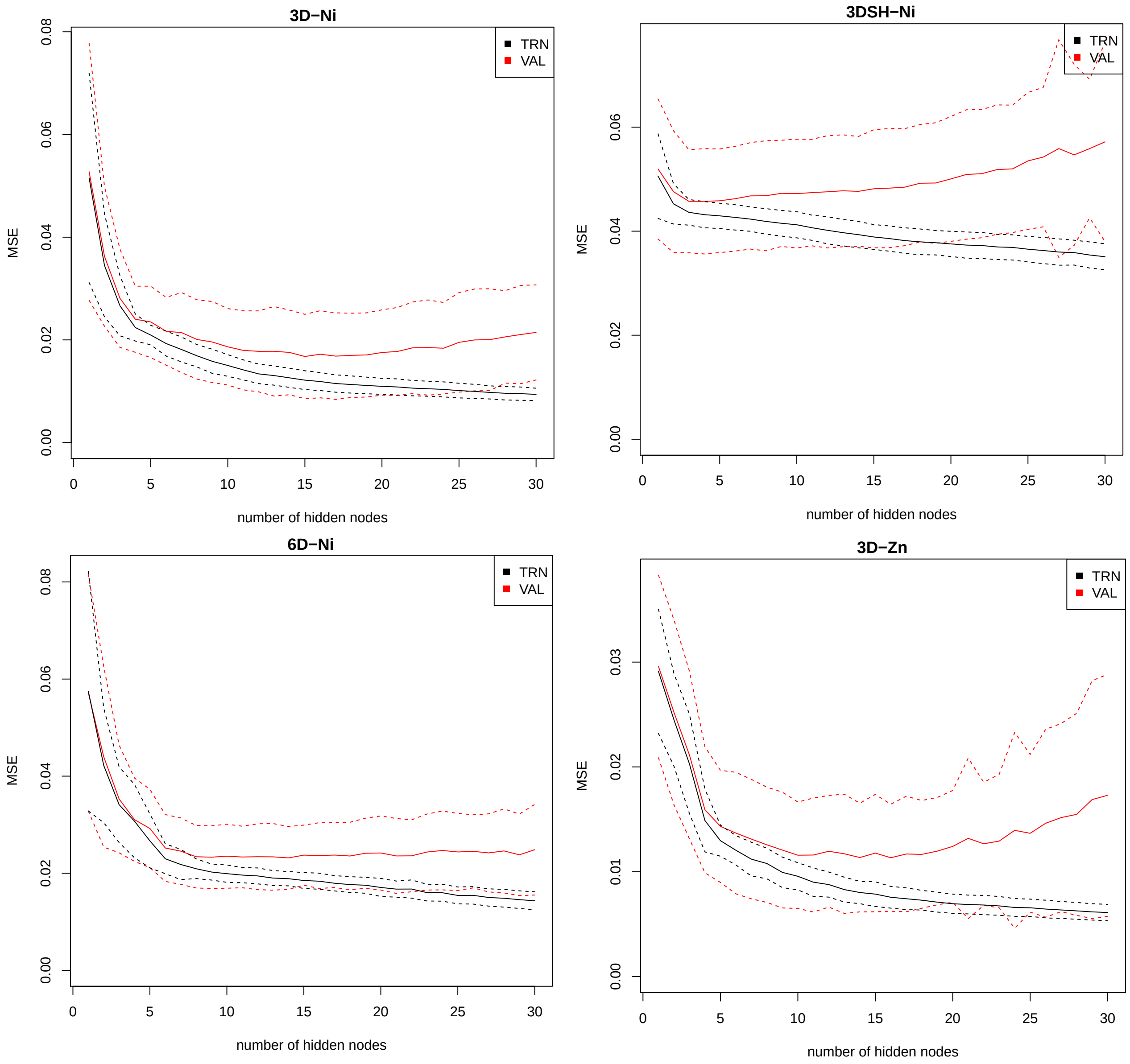


FIGURE 4.5: Results of the 3D-Ni, 3DSH-Ni, 6D-Ni and 3D-Zn scenarios using a 5-fold cross-validation (20 runs). Black and red colors correspond to the training and validation sets respectively. Solid lines show the averages of 20 runs, and dashed lines the corresponding standard deviations.

Results of this second ELM trained on the residuals are shown in figure 4.7. Because of the absence of minimum in the validation curves, the analysis of the residuals by this ELM model displays explicit evidence of no structure in the residuals for both scenarios. Moreover, results of kNN model on the residuals of 3D-Zn and 3D-Ni scenarios are demonstrated in figure 4.8, and an example of 3d omnidirectional variogram of scenario 3D-Ni for both raw data and the residuals in figure 4.9. In the same way as for the second ELM built on the residuals, the variogram of the residuals for scenario 3D-Ni (figure 4.9) demonstrates no spatial structure (pure nugget effect) and, according to the spatial resolution of the available data, it sill is close to the nugget value in raw data. Moreover,

Scenarios	3D-Ni	3DSH-Ni
Res min	-0.48260	-0.29930
Res 1^{st}Q.	-0.08571	-0.14220
Res median	0.00555	0.02313
Res mean	-0.00493	0.02622
Res 3^{rd}Q.	0.07478	0.16390
Res max	0.40450	0.55550
Res sd	0.12998	0.20795
Res Kurt.	6.61479	2.83107
Res Skew.	-0.38955	0.63421

TABLE 4.3: Details on the residuals for scenarios 3D-Ni and 3DSH-Ni.

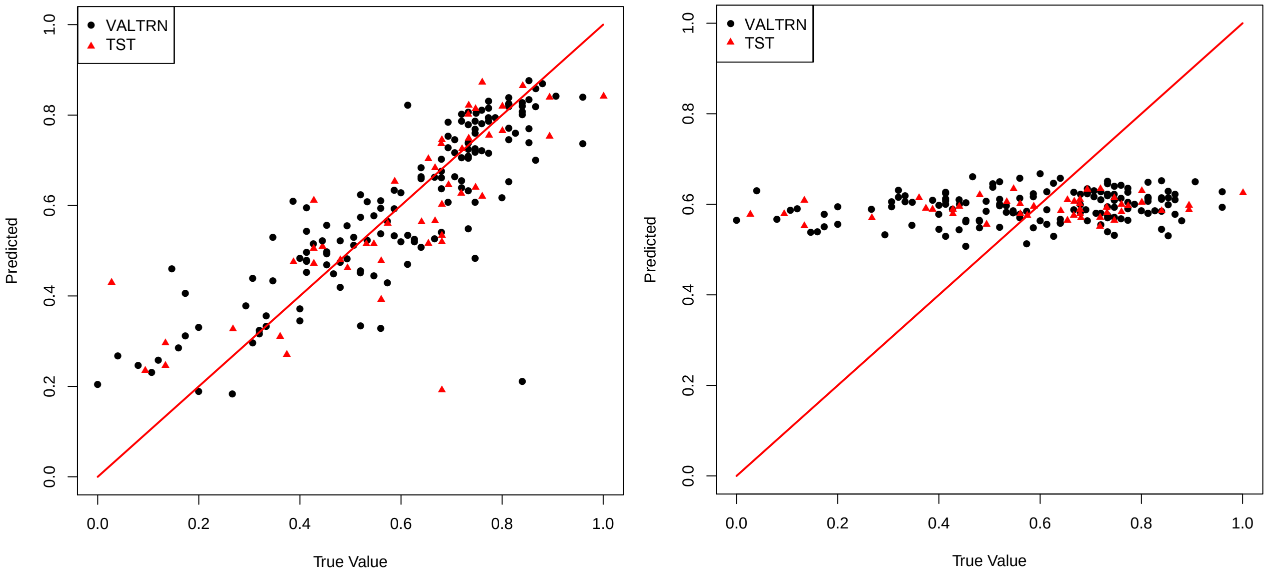


FIGURE 4.6: Accuracy plot of the scenarios 3D-Ni (on the left) and 3DSH-Ni (on the right). Validation and training data are in black (circle) and testing data in red (triangle).

this nugget value (estimated around 0.012) can be observed in both cross-validation curve of kNN for large number of neighbours (for Ni residuals in figure 4.8), and in the cross-validation curve of ELM for small number of hidden nodes (figure 4.7 on the left). This observation reflects that all structured information, according to ELM, kNN and geostatistical criteria, was extracted by the first ELM model from raw data without over-fitting. The latter result provides a self-consistent ELM capability of preventing from over-fitting and under-fitting.

Figure 4.10 shows the predictions of the Nickel (on the left) and Zinc (on the right) pollutants according to the 3D-Ni and 3D-Zn scenarios.

The last results deal with some properties of ELM, namely the number of hidden nodes, the β-distance, and the feature selection.

The analysis of the different number of nodes that optimize different pollutants can reflect a complexity index that highlights hidden relationships between pollutants (two pollutants are more similar if they have close $\tilde{N}^*$). Figure 4.11 shows the normalized mean squared error of ELM models for

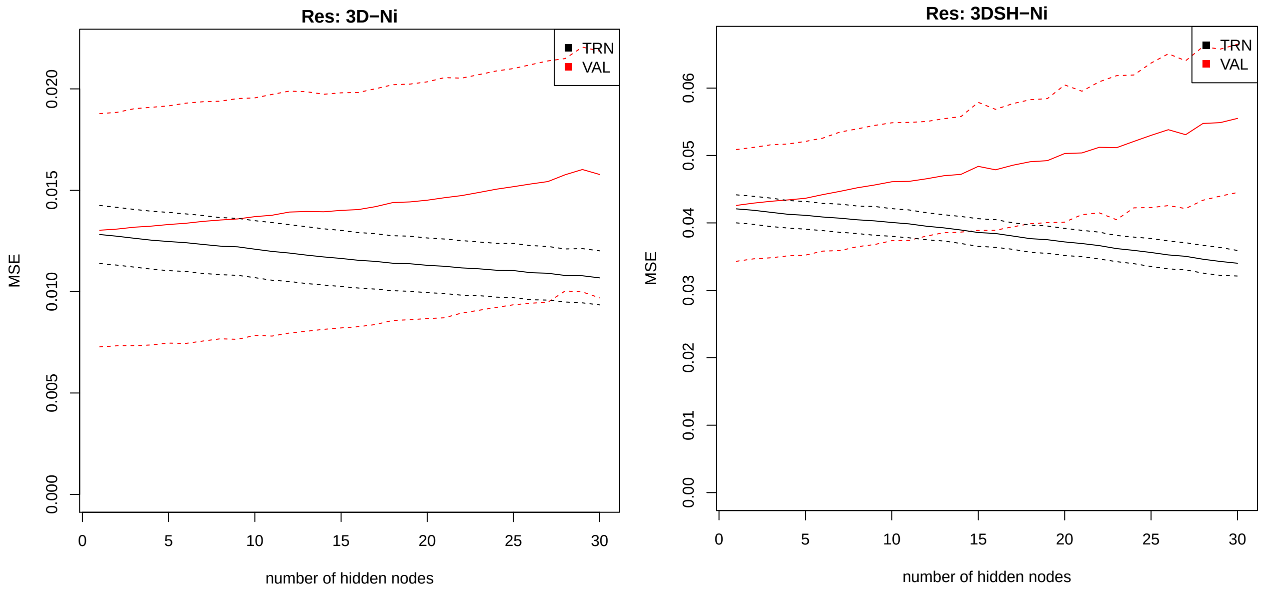


FIGURE 4.7: Cross-validation plot of ELM model for the residuals of the scenarios 3D-Ni (on the left) and 3DSH-Ni (on the right). Training data are in black and validation data in red.

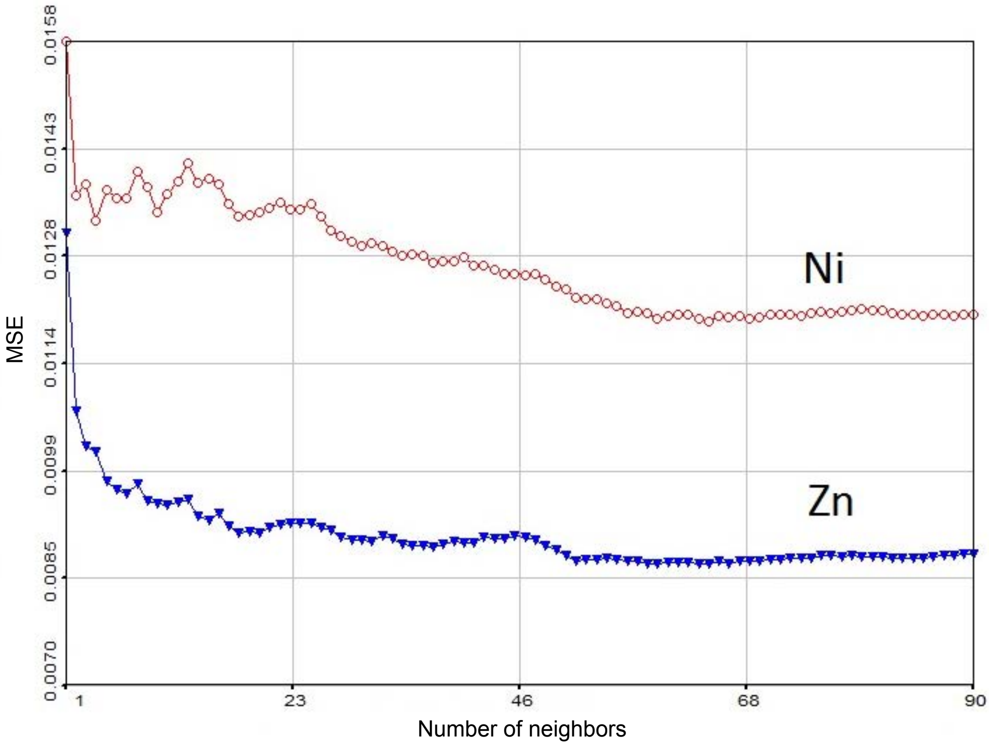


FIGURE 4.8: Cross-validation curves of ELM Zn (triangles) and Ni (open circles) residuals using kNN algorithm.

the eight pollutants according to the number of hidden nodes. The use of normalized value of MSE instead of real value simplifies the visualization of the minimum range for each pollutant. In figure 4.11 all pollutants show general range between 10 and 20 for the optimal number of hidden nodes,

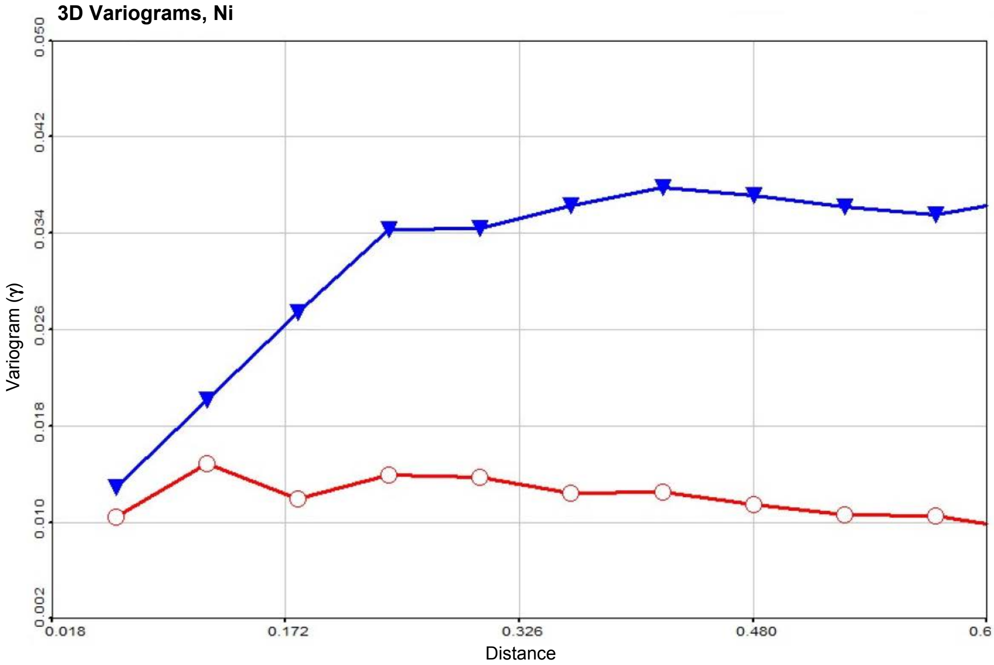


FIGURE 4.9: Omnidirectional variography of raw data (triangles) and Ni ELM residuals (open circles).

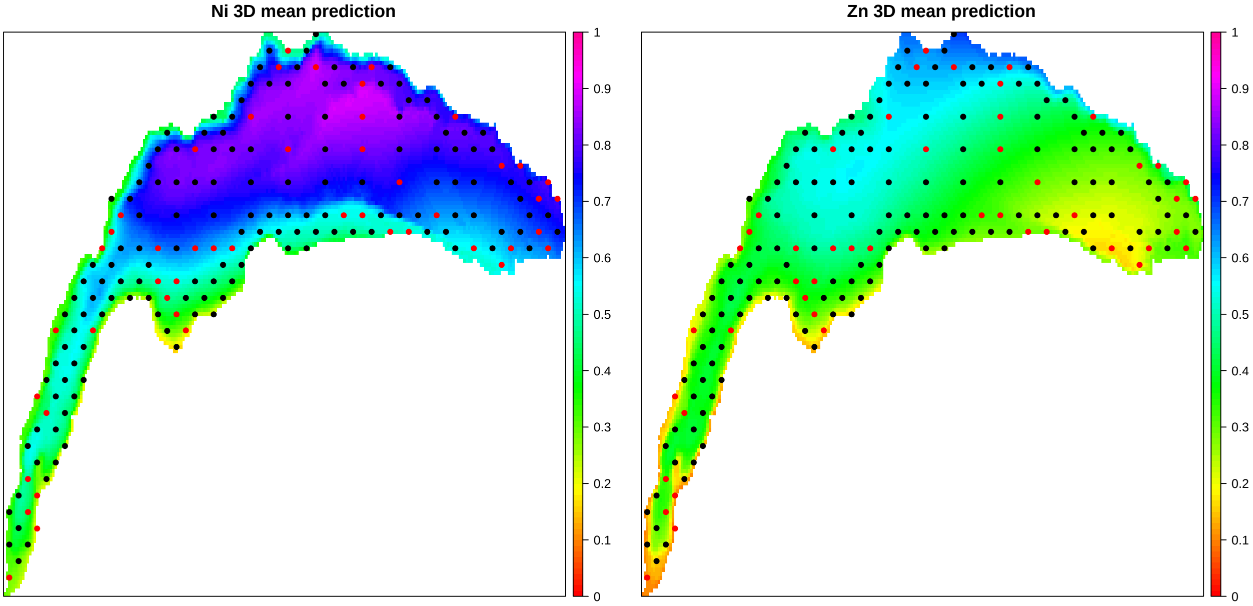


FIGURE 4.10: Prediction of the Nickel (on the left) and Zinc (on the right) normalized pollutants. Validation and training data are in black and testing data in red. The maps are linearly projected into the square $[0, 1]^2$.

except for the element 1 (i.e. Hg) which presents a minimum a bit above 20. But according to the whole spectrum, the pair 4 and 6 (that are Ti and V) shows similar behaviour. The correspondence between Titanium and Vanadium is clear enough according to the scatterplot in figure 4.3.

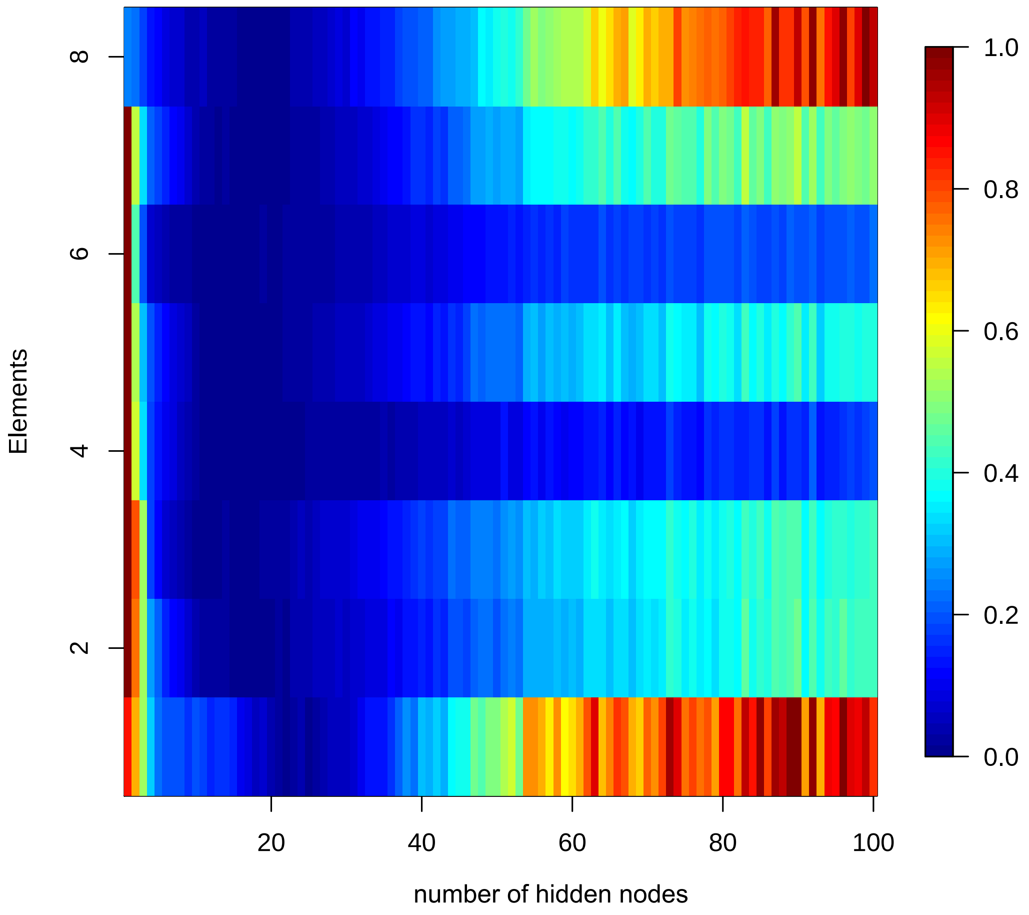


FIGURE 4.11: Visualization of the normalized MSE according to the number of hidden nodes. Elements are: 1-Hg, 2-Zn, 3-Cu, 4-Ti, 5-Cr, 6-V, 7-Ni and 8-Cd.

Index	Elements	Index	Elements
1	Zn-Hg	15	Cr-Cu
2	Cu-Hg	16	V-Cu
3	Ti-Hg	17	Ni-Cu
4	Cr-Hg	18	Cd-Cu
5	V-Hg	19	Cr-Ti
6	Ni-Hg	20	V-Ti
7	Cd-Hg	21	Ni-Ti
8	Cu-Zn	22	Cd-Ti
9	Ti-Zn	23	V-Cr
10	Cr-Zn	24	Ni-Cr
11	V-Zn	25	Cd-Cr
12	Ni-Zn	26	Ni-V
13	Cd-Zn	27	Cd-V
14	Ti-Cu	28	Cd-Ni

TABLE 4.4: Details on the index used in figure 4.12.

In addition to this, figure 4.12 displays the normalised β-distance
of pairwise pollutants indexed in table 4.4, and reveals detailed relationships between pollutants.

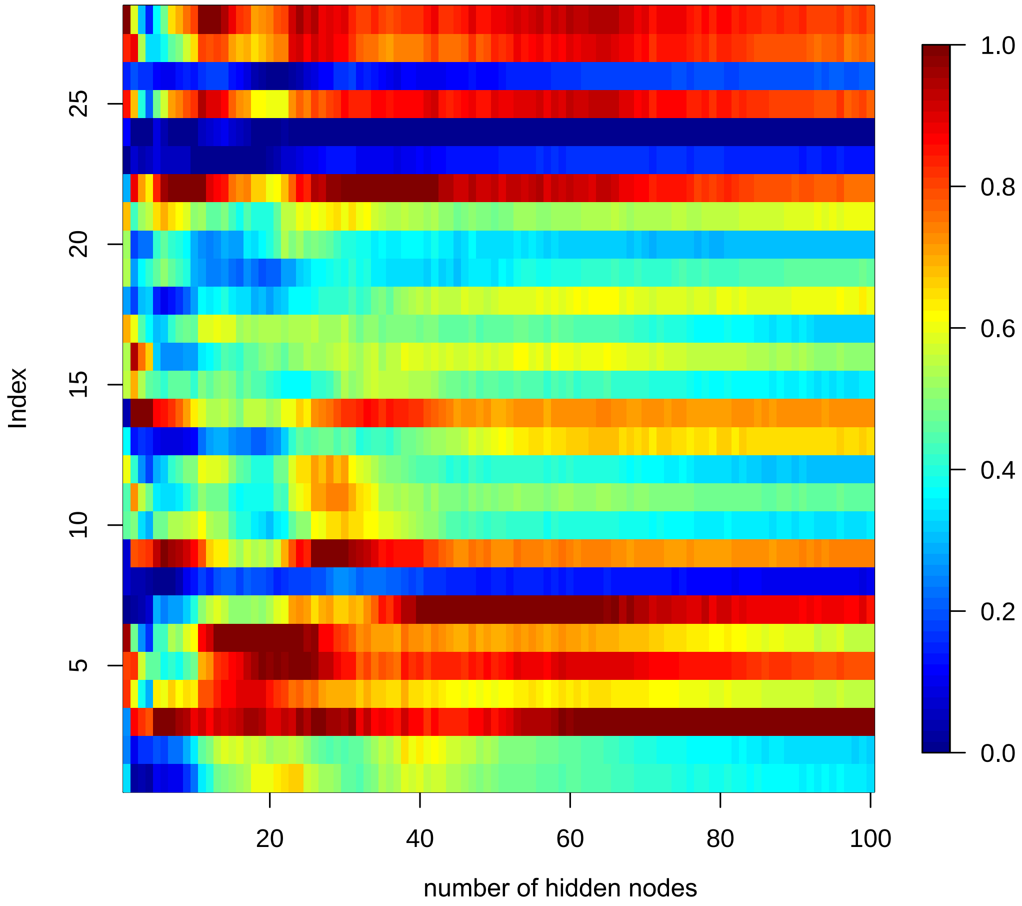


FIGURE 4.12: Pairwise visualization of the normalized β-distance according to the number of hidden nodes. Index are described in table 4.4.

Order	Selected Features
1	X, Y and Z
2	X, Y, Z and X sh.
3	X, Y, Z and Z sh.
4	Y and Z
5	X and Y
6	X, Y, Z and Y sh.
7	Y, Z and X sh.
8	X, Y, Z, X sh. and Z sh.
9	Y, Z and Z sh.
10	Y

TABLE 4.5: 10 best models according to the 6D-Ni scenario evaluated with ELM.

According to the range of optimal number of nodes, that is around 20, indexes 3, 4, 5 and 6 show evident dissimilarities. The corresponding pairwise pollutants are Hg versus Ti, Cr, V and Ni (table 4.4), and as shown in figure 4.3, no sign of relatedness is present. On the other hand, indexes 8, 23, 24 and 26 (that are Cu-Zn, V-Cr, Ni-Cr and Ni-V) show clear similarities in the β vectors (figure 4.12) and are corroborated by linear relationships in figure 4.3. Other affinity between pollutants are present

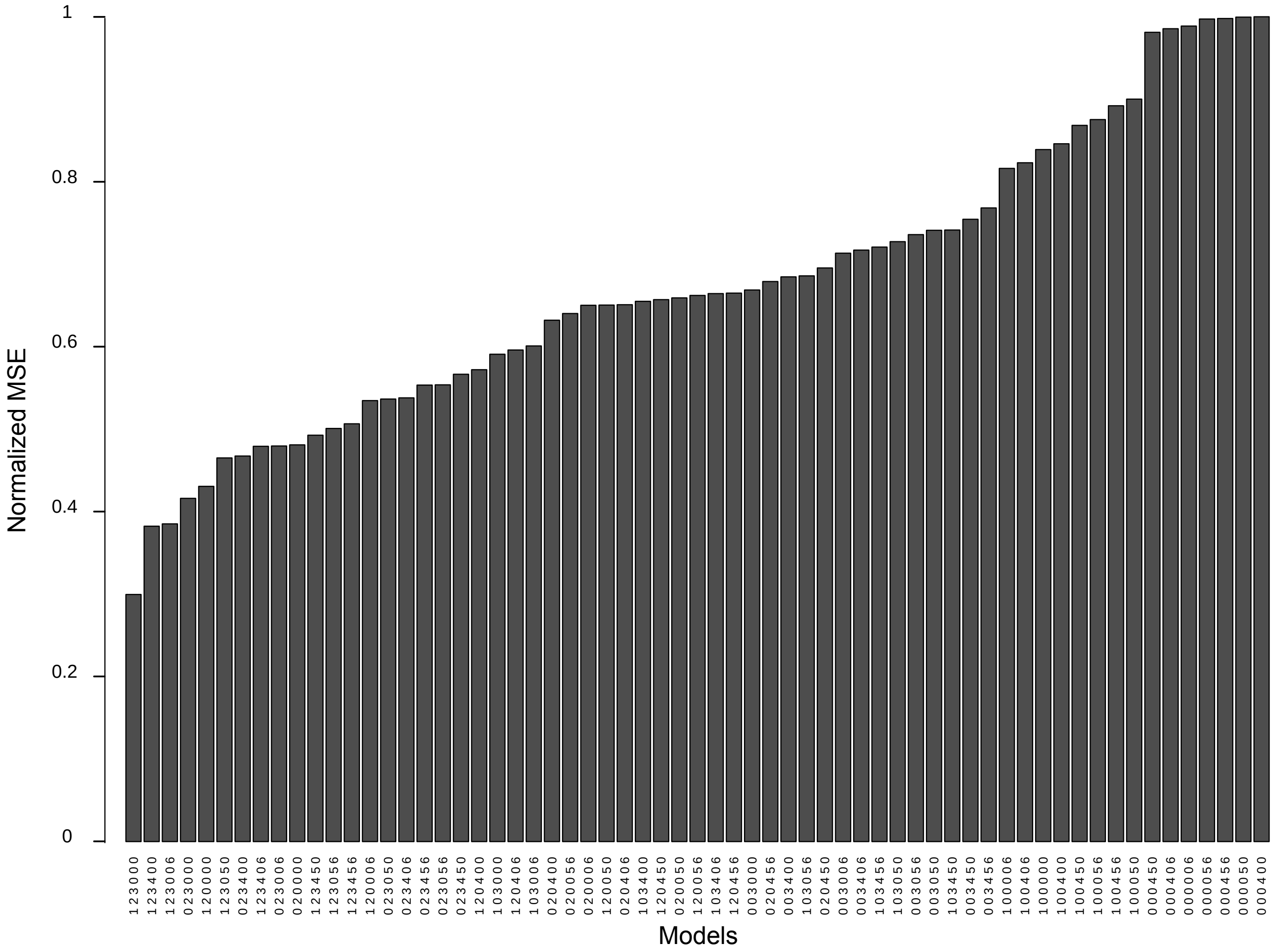


FIGURE 4.13: Evaluation of all possible models with ELM algorithm. Abscissa indicates the selected variables according to the following index: 1-X, 2-Y, 3-Z, 4-X sh., 5-Y sh. and 6-Z sh.

but are not so evident as the latter. For example, indexes 19 and 20 (that are Cr-Ti and V-Ti) show a small β-distance in the range between 10 and 20 number of hidden nodes, and are indeed related by a non-linear relationship in figure 4.3.

Finally, results of feature selection based on ELM are presented in figure 4.13 and table 4.5.

According to the scenario 6D-Ni, figure 4.13 shows the performance of ELM for all the possible combinations of input space ($2^6 - 1 = 63$), and table 4.5 highlights the ten best models. Notice that each bar corresponds to one subset of features (input variables), which has been trained and optimized. As it is highlighted in table 4.5, the best subset of input variables for the prediction of the pollutant Nickel is $\{X,Y,Z\}$. For the remaining subsets of features shown in table 4.5, they have either a majority of real variables, or have only real variables. In both cases, using the ELM performance to evaluate the relevance of features in a model is efficient. Let us note that if the number of variables is considerably higher, the use of a meta-heuristic model for improving the search of optimal subsets of features (e.g. simulated annealing, particle swarm optimization, or others) is recommended (Leuenberger and Kanevski, 2014).

4.5 Conclusions

A methodology for spatial environmental data modelling based on the Extreme Learning Machine algorithm was developed and applied to real and simulated environmental data. Comprehensive and self-consistent analysis (from raw exploratory data analysis to the analysis and justification of the results) were performed on the Lake Geneva dataset in order to investigate the behaviour of ELM when dealing with noisy and irrelevant data or features. The unique ELM hyper-parameter $\tilde{N}$ shows relevant information about the complexity of data and spatial patterns studied. It was shown, that trained ELM was able to extract and to model structured information in data. The analysis of the residuals using various techniques (ELM, kNN and variography) has proved that the residuals are not spatially structured and optimally trained ELM did not over-fit the data. It should be noted, that the methodology used does not depend on particular machine learning model and can be applied for any data driven modelling tools.

In addition, for the specific ELM properties, efficient capability of ELM has been shown in both the study of hidden relationships in environmental data using the β-distance, and in the feature selection task as a wrapper method.

The main future researches of ELM application to environmental data will deal with further elaboration of feature selection by ELM in higher dimensional spaces and quantification of modelling and prediction uncertainties.

Acknowledgments

The authors would like to thank the International Commission for the Protection of Lake Geneva (CIPEL, Nyon, Switzerland) for providing the data. They also are grateful to Jean Golay and the anonymous reviewers for their constructive comments that contributed to improving the manuscript.

Chapter 5

Wildfire Susceptibility Mapping: Deterministic vs. Stochastic Approaches

M. Leuenberger, J. Parente, M. Tonini, M.G. Pereira and M. Kanevski, Wildfire susceptibility mapping: Deterministic vs. stochastic approaches, Environmental Modelling and Software, 101, 194-203, 2018

Abstract

Wildfire susceptibility is a measure of land propensity for the occurrence of wildfires based on terrain's intrinsic characteristics. In the present study, two stochastic approaches (i.e., extreme learning machine and random forest) for wildfire susceptibility mapping are compared versus a well established deterministic method. The same predisposing variables were combined and used as predictors in all models. The Portuguese region of Dão-Lafões was selected as a pilot site since it presents national average values of fire incidence and a high heterogeneity in land cover and slope. Maps representing the susceptibility of the study area to wildfires were finally elaborated. Two measures were used to compare the different methods, namely the location of the pixels with similar standardized susceptibility and total validation burnt area. Results obtained with the stochastic methods are very alike with the deterministic ones, with the advantage of not depending on a priori knowledge of the phenomenon.

Keyword: Susceptibility mapping, Wildfires, Random Forest, Extreme Learning Machines, Portugal

5.1 Introduction

Wildfires are defined as unwanted fires occurring in countryside or rural area and burning forest and wild lands, included abandoned agricultural lands and rural vegetated areas. Wildfires, as undesirable as often uncontrolled events, represent a hazardous and harmful phenomena to people and environment. Natural fires, caused by lightning, appeared on the Earth surface in concomitance with the first plant communities, well before the appearance of humans, and played a key role in plant adaptation and the ecosystems' equilibrium (Pausas and Keeley, 2009). Nowadays, the primary cause of wildfires in populated areas is related to the human activities that voluntary (arsonism) or involuntary (accidental or negligent causes) can initiate fire. A recent analysis of fire data from the European Forest Fire Information System shows that over 95% of wildfires are human induced (San-Miguel-Ayanz et al., 2012) and this percentage is even higher in the Mediterranean regions.

Estimating the probability of wildfire occurrence in a certain area under particular environmental and anthropogenic conditions is a modern tool to support forest protection plans and to reduce fires' consequences, which can also affect the neighbouring or intermingled urban areas. In this context, the implementation of wildfire susceptibility maps and the investigation of the main driving factors inducing wildfires is fundamental. A good review of these factors can be found in (Ganteaume et al., 2013): they included human factors and related variables (such as distance to road or to urban area) as well as environmental factors. More or less sophisticated models have been applied to combine the predisposing variables into a geographic information systems (GIS) (Chuvieco et al., 2010; Chuvieco and Salas, 1996; Bonazountas et al., 2005; Jaiswal et al., 2002). The most reliable analyses applied statistical models to assess the importance of different variables influencing fire occurrences and the obtained results are used to produce the risk maps (Beverly, Herd, and Conner, 2009; Soto et al., 2013; Pourtaghi, Pourghasemi, and Rossi, 2015). Recent analyses compared different statistical models for variable selection (Pourghasemi, 2016; Pourghasemi, Beheshtirad, and Pradhan, 2016; Pourtaghi et al., 2016; Rodrigues, de la Riva, and Fotheringham, 2014; Eugenio et al., 2016) but most of the studies relied on expert knowledge to pre-select most important drivers or on the results of linear (deterministic) statistical models.

Portugal is unequivocally the European country most affected by wildfires, due to its favorable climatic conditions, topography and vegetation (Amraoui et al., 2015; Pereira et al., 2013). Investigations of driven factors and the elaborations of wildfires density and risk maps were latterly performed for this highly affected country. Tonini et al., 2017 analysed the spatio-temporal density distribution of these hazardous events in the last decades and produced a 3D graphical output of the results, which highlights areas and frame-periods more affected by wildfires. Nunes, Lourenço, and Meira, 2016 used geographically weighted regression to identify relevant municipal drivers of fires. It results that topography and population density were significant factors in municipal ignitions, while topography and uncultivated land were significant factors in municipal burnt area (BA). Verde and Zêzere, 2010 assessed forest fire susceptibility, testing and using variables of strong spatial correlation (i.e. elevation, slope, land cover, rainfall and temperature) and, more recently, Parente and Pereira, 2016 adopted this method, updating the selected variables, to map the structural fire risk in the vegetated area of Portugal.

In the present study, the authors refer to the wildfire susceptibility mapping as an estimation of the probability that fire occurs in a specific area without considering a temporal scale, assessed on the basis of predisposing factors related to terrain's intrinsic characteristics. The revised literature misses the use of stochastic models to elaborate accurate susceptibility maps of wildfires, which can be compared with the results obtained by applying deterministic approaches. These latter methods usually assume a priori knowledge of predisposing factors, or they are evaluated by applying linear methods, which implies that every set of variable states is uniquely determined by the parameters used in the model and by the sets of previous states. Therefore, a deterministic model always performs the same way for a given set of initial conditions. Contrary to the deterministic approach, the stochastic methods assume that results obtained by the combination of independent factors (i.e. variables), affecting the investigated phenomenon, can be slightly different due to the randomness of the process. This aspect is particularly useful to model environmental and anthropogenic hazard, which naturally present a complex behaviours and patterns.

Therefore, the objective of the present study is to compare stochastic approaches vs a well established deterministic method for wildfire susceptibility mapping. A first assessment of the susceptibility and hazard wildfire performed for Portugal (Verde and Zêzere, 2010; Parente and Pereira, 2016) is used as benchmarking while extreme learning machine (ELM) and random forest (RF) are the two applied stochastic methods. We restricted our investigation to a pilot area, namely the region of Dão-Lafões, characterized by a high variability and heterogeneity of environmental features and fire incidence similar to the national average, which makes it a good representative of the general characteristics of Continental Portugal.

5.2 Material: Study Area and Datasets

5.2.1 Study Area

Portugal is the European country more to the southwest, with a Mediterranean type of climate, but suffering of the influence of the Atlantic Ocean that bathes its western and southern coasts (Parente, Pereira, and Tonini, 2016). Mainland Portugal has a total land area of about 90’000 km^2, which, according to the Corine Land Cover (CLC) 2006 inventory, is predominantly used for agriculture (47%), followed by forests coverings (23%), scrub and/or herbaceous vegetation associations (23%) and open spaces with little or no vegetation (2%) (Pereira, Aranha, and Amraoui, 2014).

According to the Planos Regionais de Ordenamento Florestal (PROF), Continental Portugal is divided into 21 PROF regions (Figure 5.1). The PROF establish specific rules for the use and exploitation of its forest spaces, in order to ensure sustainable production of all goods and services associated with them (ICNF, 2016).

In the present study, Dão-Lafões region was selected as the case study area for the following reasons: (i) it is located in the Northern half of Portugal, which presents, by far, the highest wildfire incidence (Parente and Pereira, 2016); (ii) this region presents an annual average number of fires and BA very similar to the national average and; (iii) its area is very heterogeneous in terms of topography, land use and vegetation cover (Figure 5.2).

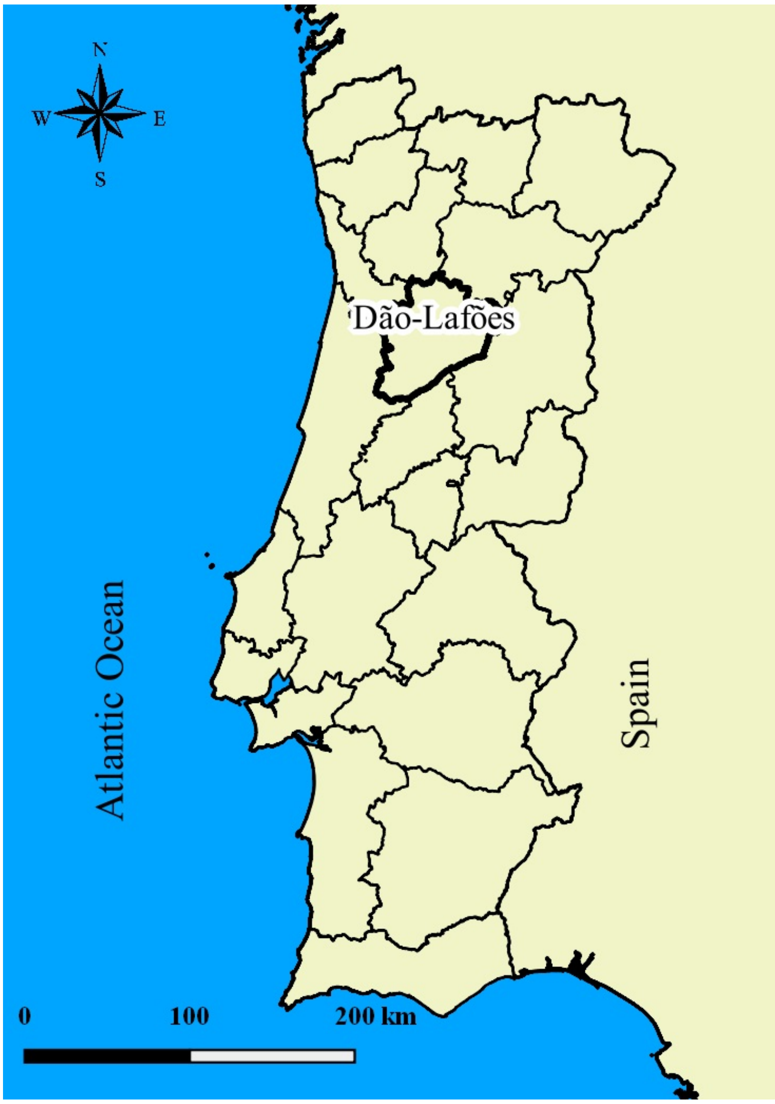


FIGURE 5.1: Location of Mainland Portugal and its 21 PROF regions protruding the study region of Dão-Lafões.

5.2.2 The Datasets

Raw data used in this study include: (i) Digital Elevation Model (DEM) derived from the Shuttle Radar Topographic Mission with a resolution of 1 arc-second (DEM-SRTM $\sim$ 25 m), used to compute elevation and slope (Gonçalves and Morgado, 2008); (ii) CLC 2006 inventory, produced by the European Environment Agency, which provides the land use and land cover maps; and, (iii) the National Mapping Burnt Areas (NMBA) implemented by the Institute for the Conservation of Nature and Forests (ICNF, 2016), which provides a detailed description of the shape and the size of the area burnt by fires in each year of occurrence. The data pre- and post- processing, as well as the mapping elaboration, were performed by Quantum GIS free software (QGIS Development Team, 2016).

Topography

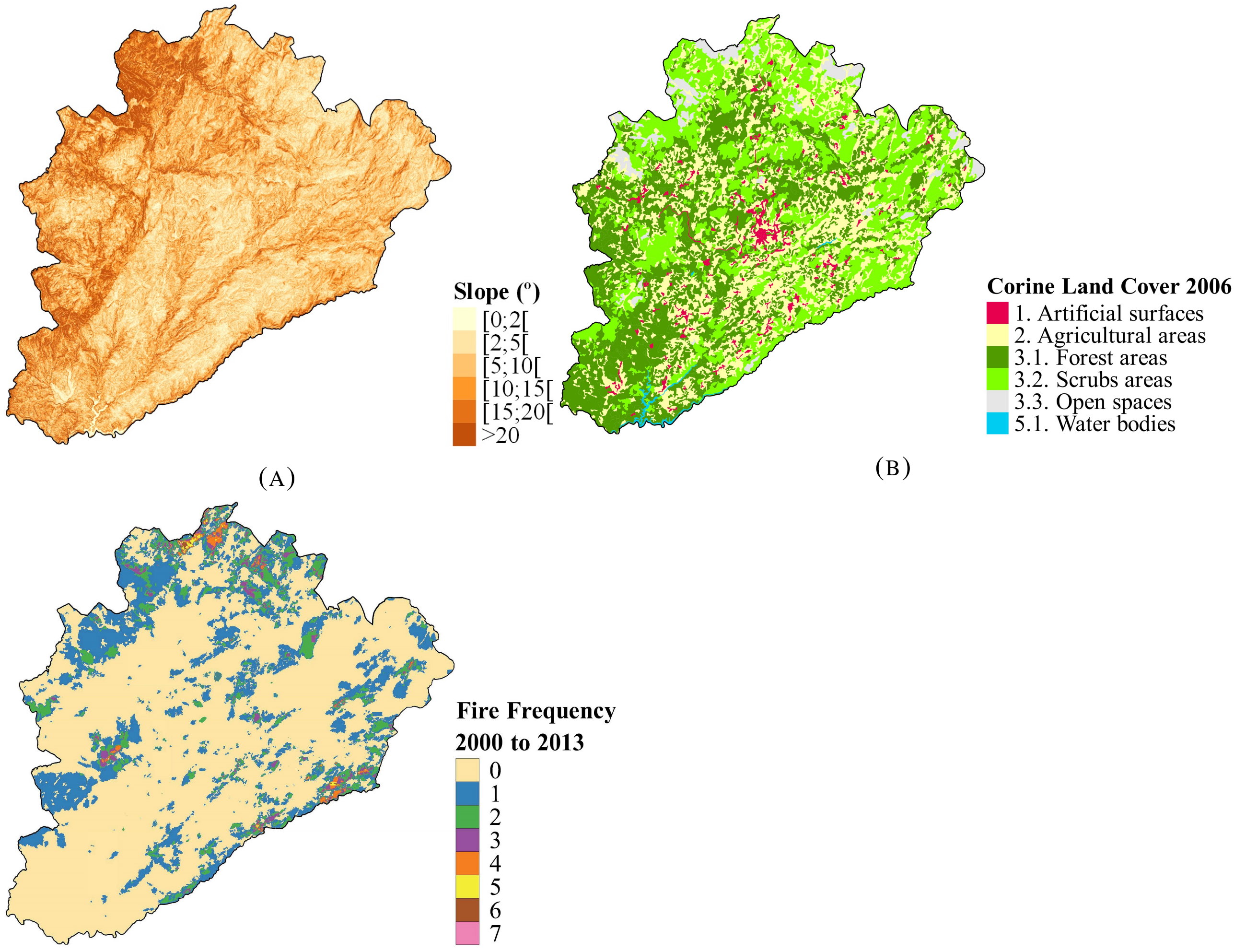
Topography, characterized by the altitude, slope and exposure, constitutes one of the most important factors to define the type of the climate of a region such as the average weather conditions and the space-time variability of the climatic elements (e.g., air temperature, precipitation, solar radiation). These factors control the life cycle of the vegetation cover and land use and have a profound influence on the fire incidence (Chuvieco and Congalton, 1989; Freire, Carrão, and Caetano, 2002; Verde and Zêzere, 2010; Parente and Pereira, 2016; Parente, Pereira, and Tonini, 2016). In this study, we considered the slope as the main topographic variable influencing the susceptibility to wildfires in the study area. This value was derived from the DEM and categorized in the same 6 classes used by Verde and Zêzere, 2010, namely: 0-2%, 2-5%, 5-10%, 10-15%, 15-20% and > 20%.

Land Use and Vegetation Cover

The CLC consists of an inventory of land cover in 44 classes with a minimum map unit of 25 ha for areal phenomena. The main classes are: artificial surfaces, agricultural, forest and semi-natural areas, wetlands and water bodies (Büttner, 2014; Caetano, Nunes, and Nunes, 2009). The 2006 version of CLC was used in the present study (Figure 5.2), since this date is in the middle of the investigated period (2000-2013). In investigated region (Dão-Lafões), the different classes of land cover and land use are quite homogeneously distributed within the area. However, it is possible to identify some patterns: higher concentration of forest cover may be found in the southwest and middle-class slopes; agricultural areas are mostly located in the southeast, away from the highest slopes, while scrubs are predominant in the southeast and northwest borders as well as in high slopes.

The Fire Dataset

The NMBA is an official Portuguese fire dataset based on satellite imagery, acquired once per year at the end of the fire season, and delivered in vector format, as polygons of the BA allowing a detailed description of the location, size and shape of the fire scars, which is fundamental for the present study. This dataset was recently reviewed to correct a minor number of missing values and data inconsistencies. It contains 17’903 fire events between 2000 and 2013, where 1’114 of which occurred on Dão-Lafões (Parente, Pereira, and Tonini, 2016). In this region, most of the fire incidences are located far in the north and in the southeast (Figure 5.2), affecting mostly agricultural areas (10%), scrublands (62%) and open spaces (13%) as well as areas with slopes ranging from 5-10° (32%) and 10-15° (23%). The location and size of the BA for the investigated period (2000-2013) is represented in Figure 5.3 in the form of fire frequency (*ff*), which is the number of times each pixel burnt over the fourteen years. The year with the highest fire incidence was 2005 (with 11% of the total number of fires and 29% of the total BA in the study period), followed by 2012 (11% of the total number of fires) and 2013 (8% of the total number of fires and 20% of the total BA). Only 18% of total number of pixels burnt at least once and the fire frequency is mostly low or very low (Figure 5.2 and Figure



(A)

(B)

(C)

FIGURE 5.2: (a) Slope, (b) land cover according to Corine Land Cover 2006 inventory, and (c) fire frequency for the 2000-2013 period in the Portuguese PROF region of Dão-Lafões.

5.3), with 97% of the total number of burnt pixels (TNBP) with $ff < 3/14$ (namely, 72% of TNBP with $ff = 1/14$, 20% of TNBP with $ff = 2/14$ and 5% of TNBP with $ff = 3/14$).

5.3 Methodology

Both deterministic and stochastic models for wildfire susceptibility mapping were applied in the present study. The deterministic model, used as benchmark, was developed by Verde and Zêzere, 2010 and further adopted and updated by Parente and Pereira, 2016. The model includes the computation of fire occurrence probability and favorability scores for each predisposing variable (land cover and slope). Two stochastic methods from the machine learning field were then applied: RF and ELM. Generally speaking, stochastic models account for the uncertainty in modelling processes that have some kind of randomness and, therefore, are useful to represent phenomena with random variability.

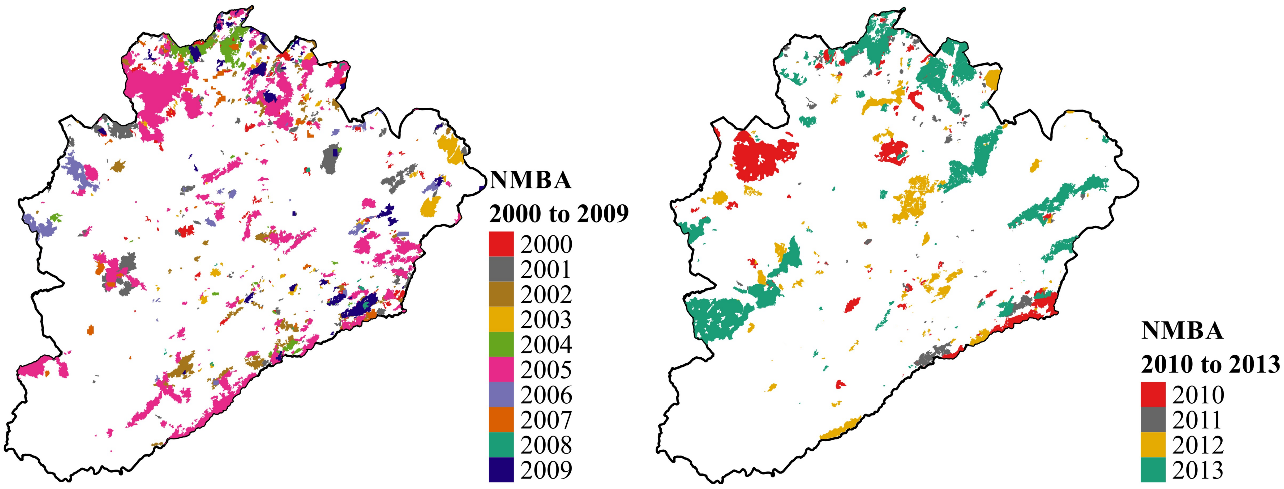


FIGURE 5.3: Annual burnt area polygons in the calibration (2000-2009) and validation (2010-2013) periods in the Portuguese PROF region of Dão-Lafões.

In the case of machine learning algorithms, the models produce susceptibility maps based on input data (variables) without the need of a priori knowledge of the investigated phenomena, but simply learning from experience. Once the model is fitted according to the training data, it allows to generate predictions over the entire study area. In the present study, data were splitted into training (2000-2009) and validation periods (2010-2013): the first was used to fit and calibrate the three models and the second to assess and compare susceptibility maps.

The susceptibility maps were elaborated by means of GIS procedures and organized into 5 classes, in agreement with the Portuguese law (DL, 2006). The classes were defined as in the reference works (Verde and Zêzere, 2010; Parente and Pereira, 2016) using the quintiles of the susceptibility, computed as explained below. The applied methods were assessed by computing the matching, pixel by pixel, between the standardized susceptibility maps obtained for the training period (2000-2009) and the effective BA over the validation period (2010-2013). These values were finally evaluated as a percentage for each susceptibility class.

The next two subsections are devoted to the brief description of the deterministic and stochastic applied models.

5.3.1 Deterministic Method

In Portugal, national authorities, such as Forest Service (ICNF) and the Meteorological Office (*Instituto Português do Mar e da Atmosfera*, IPMA) adopted the wildfire susceptibility map proposed by Verde and Zêzere, 2010 which was developed using a deterministic approach and based on just three factors. The susceptible values for each regular unit-area (i.e., pixel) is computed by integrating the favorability scores (*Fav*) of the two variables (slope and vegetation cover) and the fire probability (*fp*) as:

$$SP = fp \cdot Fav_{\text{slope}} \cdot Fav_{\text{vegetation}}. \tag{5.1}$$

The favorability scores for each class x ($Fav(x)$) of slope and vegetation cover are computed by:

$$Fav(x) = \frac{NBP(x)}{TNP(x)} \times 100, \tag{5.2}$$

where $NBP(x)$ is the number of burnt pixels in class x and $TNP(x)$ is the total number of pixels in the class x. The fire probability of each pixel is estimated using the fire database and the classic definition of probability according to:

$$fp = \frac{\text{(the number of times the pixel burned in the study period, in years)}}{\text{(duration of the study, in years)}} \times 100. \tag{5.3}$$

It is important to note that, due to the yearly temporal acquisition of the fire database (NMBA), each pixel can only burn once in each year. In addition, due to the multiplicative nature of susceptibility equation, all the null favorability scores were reclassified to one, thus becoming neutral values in the equation. Therefore, the obtained value in each pixel is a consequence of all the possible combinations of the variables found in that pixel.

5.3.2 Machine Learning Algorithms

At present, machine learning algorithms are important tools for the analysis, modelling and visualization of environmental data (Kanevski, Pozdnoukhov, and Timonin, 2009). They have good generalization abilities when modelling high dimensional and complex nonlinear phenomena, are universal modelling methods and many of them have solid roots in statistical learning theory (Hastie, Tibshirani, and Friedman, 2009). In predictive learning, they focus on modelling the hidden relationship between a set of input and output variables by trying to minimize both the errors and the complexity of the model. After a training procedure, to calibrate the model's parameters, prediction maps of the susceptibility can be computed and displayed with the corresponding uncertainty quantification. In this study, two machine learning algorithms, based on two different concepts, are used for comparison purposes: RF, which is based on decision trees, and ELM, which is based on traditional artificial neural networks. Detailed application of the RF and ELM for environmental data modelling along with the description of consistent methodology are presented in literature (Micheletti et al., 2014; Leuenberger and Kanevski, 2015). Analysis were performed using R free software (R Core Team, 2016). The packages *randomForest* and *elmNN* were employed for RF and ELM respectively.

Random Forest

Developed by Breiman, 2001, RF is an ensemble machine learning algorithm based on decision trees. It contains two hyper-parameters: the number of decision trees generated (*nbtree*), and the number of selected variables for each split node (*nbvar*).

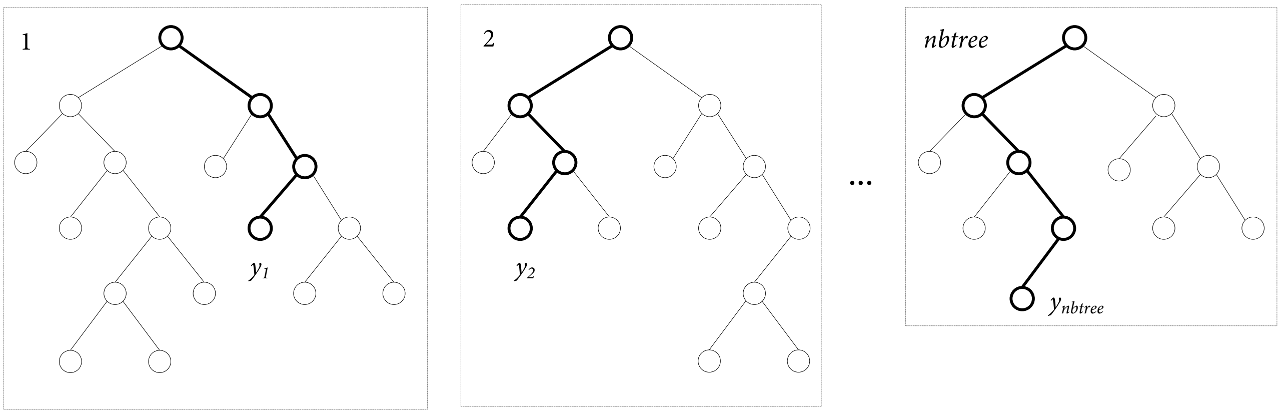


FIGURE 5.4: Structure of RF based on an ensemble of single decision trees.

The random forest algorithm first generates *nbtree* subsets of the training dataset by bootstrapping (i.e. random sampling with replacement). Then, for each subset, it will grow a decision tree by iterating the following rules up to the maximum level (when each final node contains less than 5 data points):

1. for each split, the algorithm selects randomly *nbvar* variables,

2. according to these *nbvar* variables and the output variable, it computes the Gini index (Hastie, Tibshirani, and Friedman, 2009) and selects the best variable with the best threshold in order to minimize the error of the prediction.

Once the *nbtree* decision trees have been grown, prediction of new data points is performed by taking the average value of all decision trees (Figure 5.4):

$$y_{\text{pred}} = \frac{1}{nbtree} \sum_{i=1}^{nbtree} y_i. \tag{5.4}$$

This procedure leads to a robust mean value of prediction as well as a measure of uncertainty by considering the standard deviation among all trees.

Extreme Learning Machine

ELM is based on artificial neural network concept. Following the structure of a single-hidden layer feedforward neural network (SLFN), it connects all input variables to the hidden layer, computes the neurone value and averages all neurons, with optimal weights, to the output layer (Huang, Zhu, and Siew, 2006; Leuenberger and Kanevski, 2015). More formally, composed of *nbnode* neurons ($\tilde{N}$) and by using an activation function $g : \mathbb{R} \to \mathbb{R}$, the ELM network, connecting the inputs ($\mathbf{x}_i$) to the output (y_i) value, can be written in the following form:

$$\sum_{j=1}^{\tilde{N}} \beta_j g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j) = y_i, \tag{5.5}$$

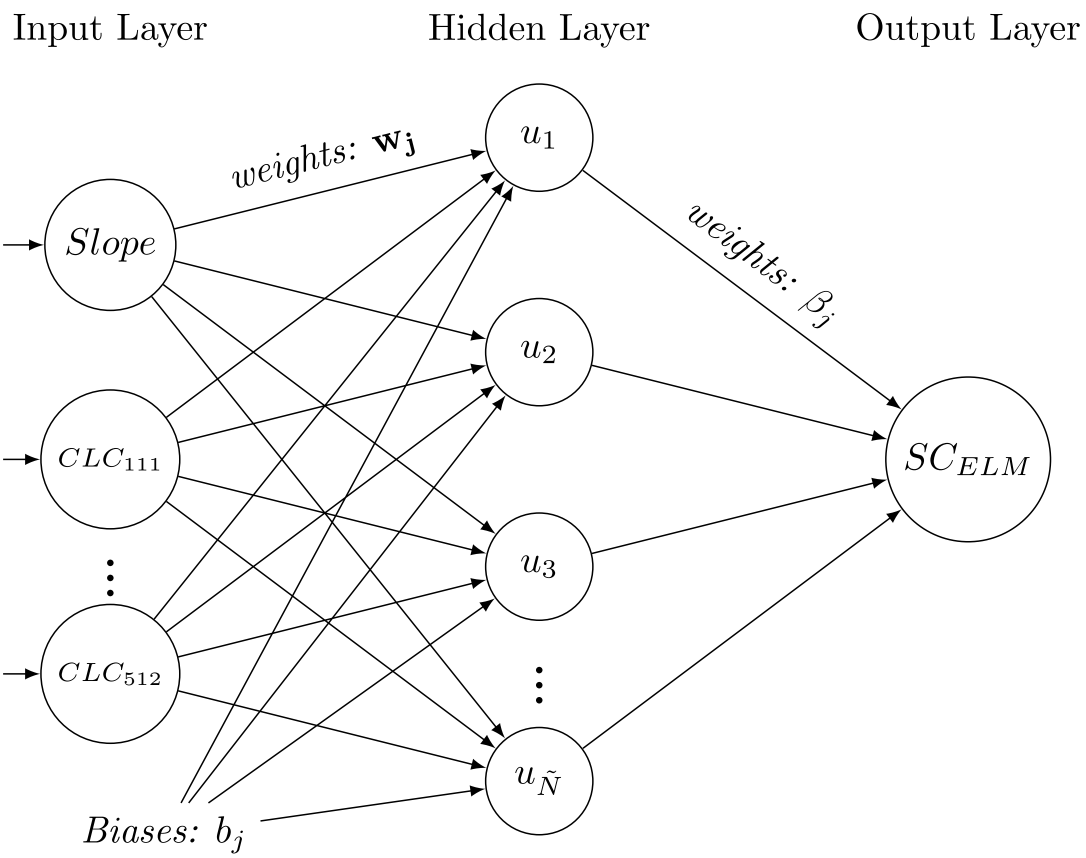


FIGURE 5.5: Structure of the ELM following a single-hidden layer feedforward neural network (SLFN). In this configuration, slope and CLC classes are used as input variables, and SC_{ELM} stands for the susceptible value of the model.

where $\mathbf{x}_i \cdot \mathbf{w}_j$ is an inner product between the input ($\mathbf{x}_i$) and the weight vector ($\mathbf{w}_j$) which connects the input layer to the jth neuron, b_j is the bias of the jth neuron, and β is a weight vector connecting the hidden layer to the output layer. In a more compact way, ELM can be written as:

$$H\beta = \mathbf{y}, \tag{5.6}$$

where $H_{i,j} = (u_j)_{j=1,\ldots,\tilde{N}} = g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j)$ is the output matrix of the hidden layer (Figure 5.5).

According to this notation, ELM algorithm applies the following steps:

1. randomly generates the input weight $\mathbf{w}_j$ and the bias b_j;

2. computes the matrix H;

3. computes the output weight $\beta = H^{\dagger}\mathbf{y}$, where $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H.

At the end, the only fitted parameter is the number of hidden neuron ($\tilde{N} = nbnode$).

	All dataset	Testing and training subsets
# of pixels which never burnt	4’535’233	81’254
# of pixels which burnt once	856’554	15’346
# of pixels which burnt twice	166’232	2’978
# of pixels which burnt 3 times	20’127	361
# of pixels which burnt 4 times	3’049	55
# of pixels which burnt 5 times	327	6
Total number of pixels	5’581’522	100’000

TABLE 5.1: Total number of pixels for the entire study area (All dataset) with the corresponding proportion of pixel which burnt 0, 1, 2, 3, 4 and 5 times. The column *Testing and training subsets* displays the proportion which was used in order to generate the 20 training subsets and the testing subset.

Parameters Optimisation

In order to optimize the learning process of RF and ELM, different pre-processing steps must be considered. First of all, the CLC classes were converted into 27 dummy variables (one for each class of the CLC variable within the study area). Then, the complete dataset, which is composed of 28 input variables (slope + CLC variables) and 1 output variable (presence or absence of forest fire), was normalized into the $[0, 1]$ interval. This transformation was performed in order to fit the functional range where ELM works in an optimal way (Huang, Zhu, and Siew, 2006). After that, from the 5’581’522 raster cells covering the study area, 100’000 points (approximately 1.8% of the total area) were randomly selected using a stratified sampling for the construction of the testing subset (Table 5.1). Namely, 6 strata were used by considering the number of time each pixel burnt (between 0 and 5 times, in this case). This process was reiterated 20 times in order to generate 20 training subsets, but without considering already selected testing points.

The optimization of the *nbtree* and *nbvar* hyper-parameters of RF was performed by using a trial and error process. The choice of this method is justified by the fact that both types of RF hyper-parameters are not highly sensitive to changes and optimized values are close to the default ones. In this study, hyper-parameters of RF were set to 1000 and 9 for *nbtree* and *nbvar*, respectively. For ELM, the *nbnode* hyper-parameter, which is the number of nodes in the hidden layer, a 5-fold cross-validation approach was performed. Minimum mean squared error (MSE) values obtained for the validation sets were retained, which lead to an optimal number of 40 nodes for this dataset.

Once each machine learning algorithm was fitted, 20 models were built by using the 20 training subsets. Finally, susceptibility maps were generated by averaging the prediction values of the 20 models for the whole study area. In addition to the mean prediction values, standard deviation maps could be extracted from this process and analysed in order to eventually detect areas with high variability in fire susceptibility. A useful by-product of RF algorithm is the variable importance ranking (Breiman, 2001). From an internal evaluation of each variable (based on random shuffling), it computes the percentage of mean square error increase (%IncMSE) by comparing the difference of

RF performance when considering both the raw variables and the shuffled variables (Breiman, 2001). As a result, each variable can be ranked according to their %IncMSE score with the following meaning: high %IncMSE score indicates an important contribution in the relationship between input and output variables, while low %IncMSE score (close to 0) indicate that the variable is not a valuable contributor to the model.

5.4 Results and Discussion

The following section (4.1) discusses the selection of CLC and slope variables as the only parameters influencing the wildfire susceptibility in our study area. Then, results and comparisons on the susceptibility maps generated by the three proposed methods are presented in section 4.2. Finally, section 4.3 assesses the different methods by using data from a testing period. Moreover, it presents the variable importance measurement for both the deterministic and the random forest approaches and discusses on the relevance of the obtained results according to the literature in this field.

5.4.1 Major Variables affecting Wildfires Occurrence in Portugal

The deterministic model was first proposed by Verde and Zêzere, 2010, further discussed in Verde, 2015, and then updated and used by Parente and Pereira, 2016 to map the structural fire risk. This model is based on the combination of geographic variables that do not change much in the short period. This is in line with the wildfire susceptibility, being a measure of the terrain/land propensity for the occurrence of wildfires based on the terrain's intrinsic characteristics (Parente and Pereira, 2016).

Although it is a quite simple model, parsimoniously based on just two variables, it is very robust. Its robustness was recently assessed (Verde, 2015) in respect to the use of single or multiple CLC inventories as well as to rely the calibration and validation on different CLC inventories. The obtained results point to a relative independence of the model performance in relation to how many or which CLC inventories are used to access the favourability scores. Parente and Pereira, 2016 test the impacts of using a high-resolution DEM and, besides mapping the susceptibility with higher spatial accuracy, the obtained patterns were very similar. Obviously, changes in vegetation cover and different fire history can induce different susceptibility patterns due to changes fire probability and vegetation dynamics.

Many researchers have studied the fire processes/mechanisms and tried to identify the underlying factors in Portugal including topography, land use land cover, climate, man-made features, demographic and socio-economic information. For example, Nunes, Lourenço, and Meira, 2016 found that topography, land cover, population density and livestock are significant in both ignition density and BA. Variables such as altitude, slope and land cover help to explain the existence of space-time clusters of fires in Portugal (Parente and Pereira, 2016; Parente, Pereira, and Tonini, 2016; Tonini et al., 2017).

Verde and Zêzere, 2010 tested the usefulness of other variables such as altitude, temperature and precipitation in the deterministic model, but they did not found any significant increase in the prediction rates. This may be due to several reasons. First, some of these variables can be proxies of each other. Slope is a measure of the altitude change (Chang and Tsai, 1991; Parente and Pereira, 2016) while altitude regulates the rainfall and temperature (Li et al., 2010; Neteler et al., 2011; Parente and Pereira, 2016). Climate/weather conditions determines the existence, type and state of the vegetation at each location, which means that the information about vegetation cover implicitly considers climate information (Parente and Pereira, 2016). Second, all fires tend to occur associated to high air temperature, low humidity and relatively long periods of drought (Amraoui et al., 2015; Parente and Pereira, 2016; Trigo et al., 2006). Third, vegetation cover can be viewed as a set of different variables instead of just one. For example, to model fire ignition probabilities, Vasconcelos et al., 2001 test the usefulness of CLC related variables such as distance to urban areas, distance to agricultural areas, distance to forests, distance to scrublands, etc., which can be viewed as a different use of vegetation cover. Oliveira et al., 2012 adopted a similar procedure to study the spatial distribution of large fires, considering the proportion of forest area, of scrubs, of agricultural areas, etc. Finally, in a very recent study, Fernandes et al., 2016 identifies fuels and topography as the major determinants of large-size BA in Portugal and in the Western Mediterranean Basin, which is consistent with previous findings on the characterization of wildfires in Portugal (Marques et al., 2011).

Another aspect that must be pointed out in deterministic model is the double use of the BA/fire probability, namely: (i) to compute the favourability scores to rank CLC and slope classes in terms of fire proneness; and, (ii) in the expression of susceptibility, in the form of fire probability in each pixel, i.e., to discriminate where, within the country, each class is more or less affected. In addition, fire probability is also a proxy for the human behaviour since the large majority of the fires are caused by humans (Parente and Pereira, 2016; Verde and Zêzere, 2010).

5.4.2 Susceptibility Maps

Figure 5.6 shows the susceptibility maps obtained by applying the three models. In a broad sense, the three models lead to relatively similar maps. The main areas with high/very high and low/very low susceptibility classes are detected and highlighted on similar locations. The very high susceptibility class shows a common pattern for the three models and is mainly located on the North of the region and on the South border.

In order to evaluate the two stochastic models, assuming the deterministic one as reference, maps of the differences of susceptibility were generated (Figure 5.7). These maps were produced by assigning each class of susceptibility to a unique value (very low=0, low=1, medium=2, high=3 and very high=4), and by computing the differences pixel by pixel. These maps are predominantly characterized by light colours (Figure 5.7), which means that differences, when they exist, occur between

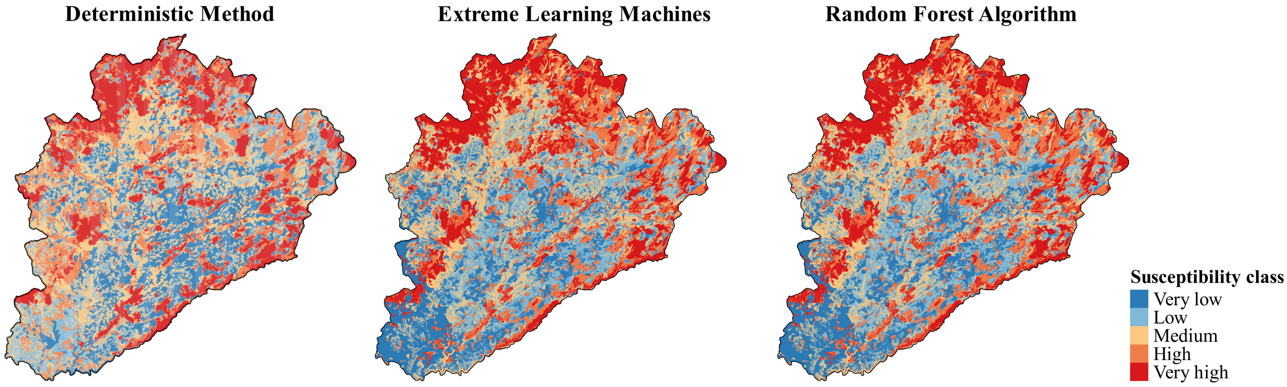


FIGURE 5.6: Susceptibility maps for the three models based on the 5 generated classes, which are very high, high, medium, low and very low.

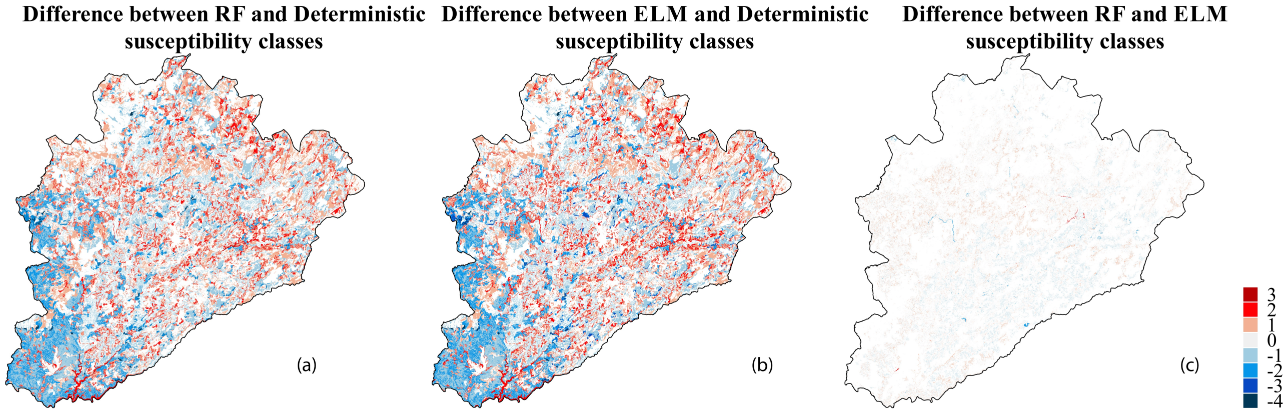

Difference between RF and Deterministic susceptibility classes
Difference between ELM and Deterministic susceptibility classes
Difference between RF and ELM susceptibility classes
(a)
(b)
(c)
3
2
1
0
-1
-2
-3
-4

FIGURE 5.7: Differences of the susceptibility classes between the three methods. For each susceptibility class of Figure 5.6, a value is assigned (very low=0, low=1, medium=2, high=3 and very high=4) and the differences between classes for each model were computed.

successive classes ($-1 \leq \mathit{diff} \leq 1$). The southwest part of the study area shows an apparent and systematic underestimation of the susceptibility classes for RF and ELM models compared with the deterministic model (Figure 5.7a and Figure 5.7b). Nevertheless, this difference is not problematic since it concerns essentially of classifying a pixel in the very low class instead of low or medium classes. For the rest of the region, differences between the stochastic and deterministic models are insignificant. Differences between the two machine learning algorithms (RF and ELM) are shown on Figure 5.7c: these differences are only slightly present but without significant spatial variations. This result is mainly due to the same pre-processing and similar methodological procedure featuring the two stochastic methods. Moreover, from the machine learning point of view, the use of 100’000 training points contributes to the general stability of both RF and ELM models.

Finally, it is important to note that standard deviation maps (not shown) computed from the 20 RF and ELM models built by using the 20 training subsets to eventually detect areas with high variability in fire susceptibility, reveal, on the contrary, very low variability of both models. In addition to this, a general evaluation of both methods was performed by computing the mean squared error (MSE) on

Susceptibility classes	ratio	Nb Pixels	%
Very high	33.1	360’161	42.8
High	22.6	267’356	31.8
Medium	10.4	136’410	16.2
Low	6.6	58’392	6.9
Very low	1.7	18’875	2.3
Total	15.1	841’194	100.0

TABLE 5.2: Ratio between the size of each class and the proportion of BA for the testing period for the determinictic approach.

the testing set (which has never been used during the learning process). Unsurprisingly, ELM and RF algorithms show highly similar results with a MSE of 0.1115 for ELM and 0.1117 for RF.

5.4.3 Methods’ Assessment

Figure 5.8, Table 5.2 and Table 5.3 show the proportion of BA within each susceptibility class obtained for each method and assessed for the testing period (2010-2013). Moreover, the ratio between the size of each class and the proportion of BA were also computed. By considering the deterministic model as the reference, the ELM and RF susceptibility maps reveal proportions of BA close to the benchmark model. Differences in the percentage of total BA in each susceptibility class is always less than 7.2%. Apparently, the percentage of total BA in the two first susceptibility classes (very low and low) is higher for ELM and RF (approximately 4%), but in the medium and high susceptibility classes this value is higher for the deterministic approach (about 3%-7%). For the last susceptibility class (very high), both ELM and RF algorithms show a percentage of total BA higher than the deterministic one (3% higher).

As it was already mentioned in section 4.1, the susceptibility maps generated by ELM and RF are very similar. As shown in Figure 5.8, Table 5.2 and Table 5.3, the maximum difference between the percentage of total BA in each susceptibility class for both methods is 0.5% in low and medium classes, and almost zero in the others classes. Generally, and by considering the three approaches, the obtained results are promising in the sense that less than 20% of the total BA of the testing period was classified as very low or low susceptibility (by summing the very low and low scores). This evaluation over the testing period allows to validate the proposed new approach in this field through machine learning algorithms, and to compare the stochastic and deterministic approaches on non-used dataset.

The RF algorithm allows an internal evaluation of each input variable, which leads to a variable importance ranking (Breiman, 2001). This last result constitutes a significant added value to the understanding of the phenomenon. In Figure 5.9, the top 9 variables for RF are listed by decreasing order of their respective %IncMSE score.

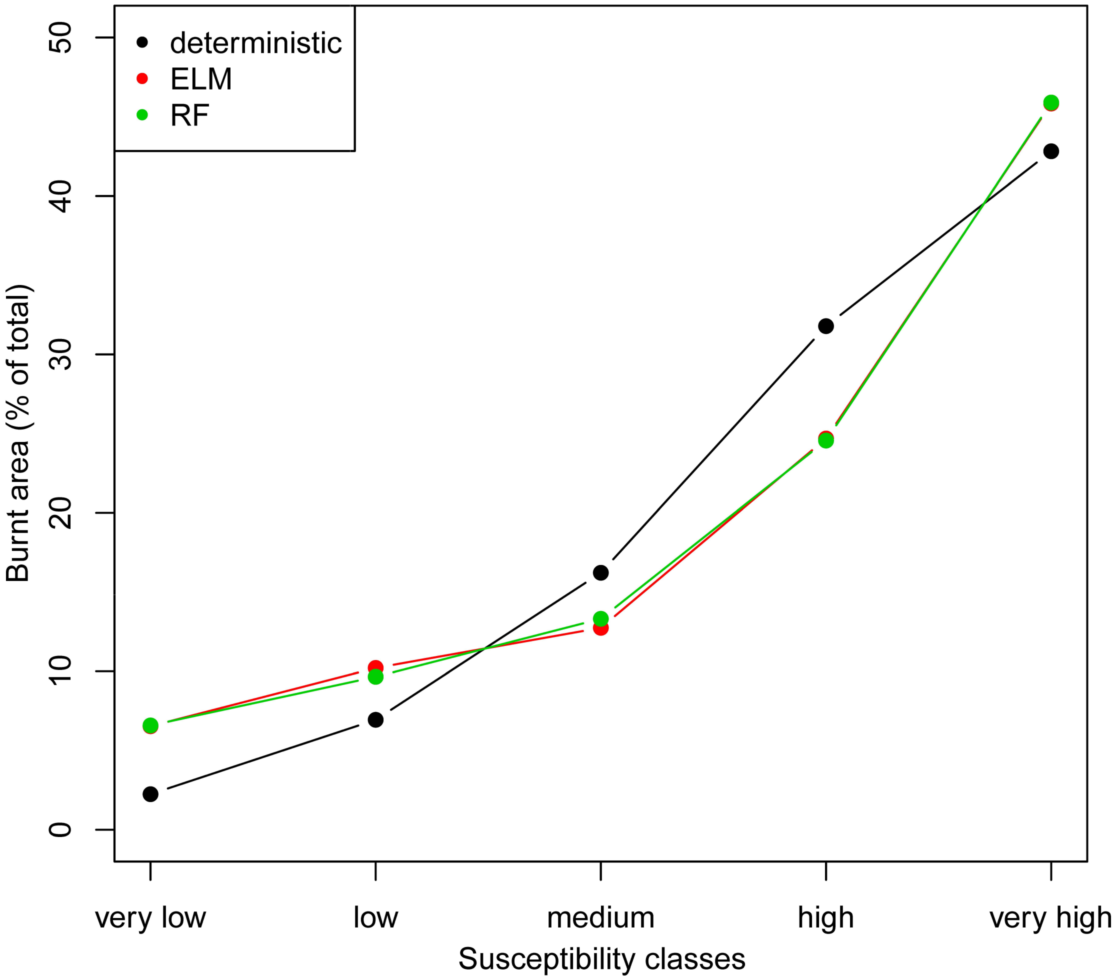


FIGURE 5.8: Proportion of total BA explained by each of the three models in each susceptibility class.

The first six land cover variables (CLC324, CLC322, CLC334, CLC333, CLC312 and CLC321) represent the variables that most contribute to model and explain the observed variance (higher %IncMSE score). These correspond to: transitional woodland-shrub, moors and heathland, burnt areas, sparsely vegetated areas, coniferous forest and natural grasslands. This short list is dominated by scrub and/or herbaceous vegetation associations (level 32 of CLC classes) followed by open spaces with little or no vegetation (level 33) and forests (level 31). These results are in accordance with fire selectivity studies performed for Portugal where fire selectivity is generally higher for scrublands, pine stands and eucalyptus plantations than for evergreen oak woodlands, annual and rainfed crops and agroforestry lands (Barros and Pereira, 2014). Similar findings were recently obtained for fire proneness studies, also performed for Portugal (Moreira et al., 2009; Silva et al., 2009). In general, agricultural areas are excluded from this list because it includes well managed arable lands (both irrigated and non irrigated), permanent crops (vineyards, olive groves, fruit trees and berry plantations and even pastures). However, heterogeneous agricultural areas (CLC level 24), especially those corresponding to complex cultivation patterns with significant areas of natural vegetation, present higher

Susceptibility classes	ELM			RF		
	ratio	Nb Pixels	%	ratio	Nb Pixels	%
Very high	34.5	385’519	45.8	34.6	386’051	45.9
High	18.6	207’640	24.7	18.5	206’570	24.6
Medium	9.6	107’178	12.8	10	111’970	13.3
Low	7.7	85’877	10.2	7.3	81’211	9.7
Very low	4.9	54’980	6.5	5	55’392	6.5
Total	15.1	841’194	100.0	15.1	841’194	100.0

TABLE 5.3: Ratio between the size of each class and the proportion of BA for the testing period for ELM and RF.

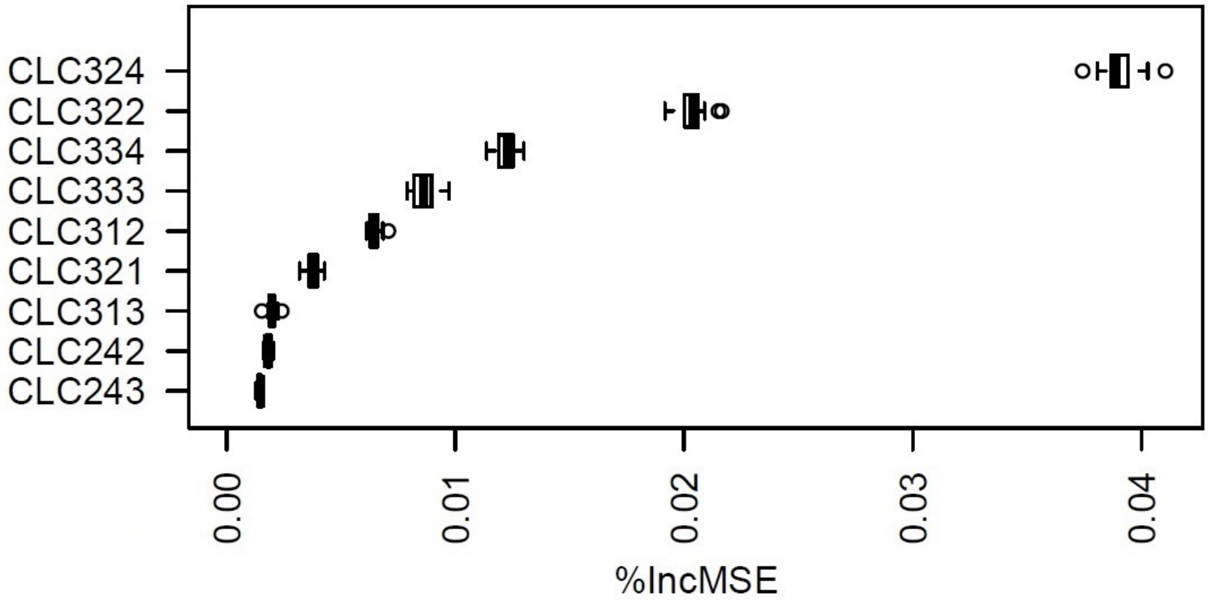


FIGURE 5.9: Variable importance computed with random forest algorithm over 20 runs. The 9 top variables are displayed with the corresponding % increase of mean square error.

relative importance in RF stochastic models. The slope is one of the most important factors of fire spread, acting on different aspects of the fuel combustion (Rothermel, 1972). However, per se, i.e. without the other aspects of the fire environment/controls usually conceptualized in fire triangles (e.g, Whitlock et al., 2010), the slope is not able to independently determine the terrain/land propensity for the occurrence or spread of a wildfire. For example, terrain parcel with high slope can be free from vegetation. Therefore, it is not surprising that the ranking of the most important variables is dominated by the land cover variables with 9 classes in the top 10 variables.

On Table 5.4, the top 6 variables for both random forest and deterministic methods are shown. For comparison purposes the favorability score of the deterministic model (computed based on eq.5.2) are retained. As highlighted, 5 of the 6 top variables selected by random forest are also among the most important variables of the deterministic model even if with a different order. This fact underlines

Variables description	Random Forest		Deterministic approach	
	CLC classes	%IncMSE	Rank position	Favorability score
Traditional woodland-shrub	324	0.03906	6	48.45520
Moors and heathland	322	0.02036	3	68.22487
Burnt areas	334	0.01219	1	95.71093
Sparsely vegetated areas	333	0.00862	2	83.87631
Coniferous forest	312	0.00646	15	4.86921
Natural grasslands	321	0.00378	4	59.39358

TABLE 5.4: Variable importance for the top 6 feature for the random forest and the corresponding rank in deterministic model. Favorability scores computed in the deterministic model are used for ranking the variables. CLC classes are identified by the level code.

that, in spite of the differences between the methods (random forest being able to detect non-linear relationship), the matching between the most relevant variables is highly satisfactory and validates the use of the new approach based on machine learning algorithm.

The apparent greater importance of conifers (CLC312) in relation to the mixed (CLC313) and broadleaf forest/hardwoods (CLC311) for the RF (Figure 5.9) is also worth noting for two reasons: (1) these variables present the same relative importance in both methods; and, (2) it is in good agreement with previous studies for vegetation fire proneness performed for Portugal (Silva and Harrison, 2010; Pereira, Aranha, and Amraoui, 2014). In fact, the increase in conifer tree component tends to increase the difficulties to control the fire (Rowe and Scotter, 1973), BA and fire proneness (Moreira et al., 2009; Silva et al., 2009; Silva and Harrison, 2010) and fire risk in WUIs (Lampin-Maillet et al., 2010).

5.5 Conclusion

In the present paper, susceptibility maps of wildfires obtained by applying stochastic methods, namely Random Forest and Extreme Learning Machine, were compared with the correspondent map elaborated by applying a validated standard deterministic method, here considered as a benchmark. The study was performed for the Dão-Lafões region of Portugal, which is a representative region of a country highly prone to wildfires. The variables, implemented into the model, considered as favorable factors for wildfires, are the slope, the land use and vegetation covers, provided by the Corine Land Cover 2006 inventory. The official dataset of the national mapping BA was considered to train (2000-2009 period) and test (2010-2013 period) the models. Comparison of the obtained results clearly suggests that the two stochastic models perform in an equal manner in terms of susceptibility areas and classes as well as that these results are broadly consistent with susceptibility maps obtained with the benchmarking model. The main benefit of using stochastic models is that these approaches are data driven, meaning that they do not need a priori knowledge of the process. Moreover, random forest directly provides the measurement of the importance of each variable. On this respect, the RF and the deterministic models present similar top variable importance ranking. Results of the present analysis are encouraging for further applications of stochastic models to elaborate susceptibility maps considering more variables and larger areas.

Software and Data Availability

The following software and data were used to perform the analysis presented in this paper:

- **QGIS** (QGIS Development Team, 2016), an open source geospatial software, was mainly used for the pre- and post- processing and the elaboration of maps.
- **R** language (R Core Team, 2016) is an open source statistical software. It was used with the packages *randomForest* and *elmNN* for computing the random forest and the extreme learning machine algorithms.
- **Digital Elevation Model** (DEM) derived from the Shuttle Radar Topographic Mission (STRM - NASA) was used to compute the slope.
- **Corine Land Cover** (CLC 2006) is an inventory provided by the European Environment Agency. It was used in order to extract the land use and land cover map.
- **National Mapping Burnt Areas** (NMBA) is an official Portuguese fire dataset and provides a detailed description of the shape and the size of BA. It was provided by the Institute for the Conservation of Nature and Forests (ICNF, 2016).

Acknowledgements

Michael Leuenberger wants to thank the Institute of Earth Surface Dynamics (University of Lausanne), where he works on this paper during his PhD. This work was supported by: (i) the Herbette Foundation of the University of Lausanne; (ii) the FIREXTR project, PTDC/ATPGEO/0462/2014; (iii) the project Interact - Integrative Research in Environment, Agro-Chain and Technology, NORTE-01-0145-FEDER-000017, research line BEST, co-financed by FEDER/NORTE 2020; and, (iv) European Investment Funds by FEDER/COMPETE/POCI - Operacional Competitiveness and Internacionalization Programme, under Project POCI-01-0145-FEDER-006958 and National Funds by FCT - Portuguese Foundation for Science and Technology, under the project UID/AGR/04033/2013. We are especially grateful to ICNF for providing the fire data and to João Pereira for the final spelling and grammar review of the manuscript.

Chapter 6

Conclusion

6.1 Conclusions and future directions

The major goals of data driven modelling of environmental data using predictive machine learning algorithms concern, in a broad sense, the calibration, prediction, testing, and the visualization of hidden relationships between input (independent) and output (dependent) variables. In this regard, this thesis investigates and combines methodologies and methods in order to better understand the black-box of machine learning algorithms, namely, data preprocessing, feature selection, validation and testing, decision-oriented mapping taking into account uncertainties quantification. Let us recall the research carried out in the thesis and its potential future development.

Chapter 2 presents the main bases and definitions for the general understanding of machine learning algorithms applied in the research. In particular, parameters versus hyper-parameters, data splitting, measures of evaluation and complexity analysis are discussed. Although the elements developed in this chapter are part of the general basis, a good understanding of the different issues helps and improves the interpretation of the results. It should be noted that the methodology used in chapter 2 (except for the section 2.7.2) does not depend on particular machine learning model and can be applied for any data driven modelling tools. An important future methodological development could be an extension (or more precisely, scaling) of the proposed methodology to the domains of big and non-homogeneous high dimensional multivariate data. Also, more attention should be paid in the future to the incorporation (assimilation/integration) of science-based models (physical, meteorological, pollution, etc.) depending on the phenomena under study, and to a direct incorporation of expert knowledge.

In chapter 3, the Extreme Learning Machine (ELM) algorithm is presented. Due to its solid theoretical basis (universal modelling tool), computational and implementation efficiencies, ELM nowadays has gained a great popularity. The thesis contributes not only to the new environmental case studies using ELM but also to the development of the ELM application to the generic feature selection problems in machine learning. A new method, which combines ELM with simulated annealing is introduced and studied in detail. Applied and tested on different benchmark classification and regression studies (appendix A), it was demonstrated, that the proposed method was able to extract the relevant information (i.e., optimal subset of features) without the loss of the accuracy in predictive learning. The obtained results are very promising and confirm the potential of ELM to become an important new tool in computational feature selection. It will allow the investigation of complex and high dimensional problems in the domain of environmental risks, natural hazards, renewable resources assessments and others. It should be noted, that the proposed simulated annealing algorithms can be replaced by other optimization algorithms, for example, genetic algorithms or particle swarm optimization. However, the general methodological approach remains the same as it was proposed with the simulated annealing.

In the second part of chapter 3, a new methodology, which combines the Extreme Learning Machine with a bootstrap-based procedure for the quantification of the uncertainties in the data and in the model, is proposed. Comprehensive analyses were carried out in order to investigate the behaviour of both ELM and bootstrap-based procedures for noisy data. It was shown, that this method allows fast and accurate prediction and quantification of the different kinds of uncertainties. It is worth mentioning, that at this stage of the research, the generated maps should be considered as a visual indicator for identifying the spatial area where the data or the model have significant uncertainties. This means that future research needs to be carried out in order to investigate other types of noise sources and noise distributions. However, already the first results allow the practitioners to identify and visualize areas where the prediction or the susceptibility map can have a high degree of uncertainty. Knowing these areas is, in most of the cases, even more essential than knowing the prediction itself. For this reason, the proposed method is of great interest for the environmental decision-oriented mapping. An important future development of the uncertainty quantification can be related to the monitoring networks optimization based on the estimated sources of the uncertainties and to active learning of environmental data.

Chapter 4 presents an article mainly focused on the methodological aspects of the Extreme Learning Machine applied to the simulated and real environmental pollution data (Leuenberger and Kanevski, 2015). From raw exploratory data analysis to the analysis and justification of the results, this chapter proposes a comprehensive and self-consistent data driven analysis with ELM algorithm. In particular, it investigates the behaviour of ELM when dealing with noisy and irrelevant data or features. It was shown, that trained ELM was able to extract and to correctly model structured information in data. Moreover, the analysis of the residuals using different techniques (like ELM, kNN or variography) highlights the ability of trained ELM to not over-fit the data and to provide residuals which are not spatially structured. In addition to the specific ELM properties mentioned above, an efficient capability of ELM has been shown in both the study of hidden relationships in multivariate environmental data, and in the feature selection task. Although the chapter 4 is mainly focused on the ELM algorithm, the proposed methods and methodologies can be adapted for any machine learning algorithm. In fact, this chapter is a demonstration of the proposed methodology application to real data case study. The natural future developments can be in the application of the approach to new challenging case studies on natural hazards and renewable energy resources assessments.

Finally, chapter 5 presents a paper on wildfire susceptibility maps obtained by applying stochastic methods (i.e., the Random Forest and Extreme Learning Machine algorithms) and standard deterministic method (Leuenberger et al., 2018). The main objective was to compare the different methods and the resulting susceptibility maps by considering the deterministic methods as a benchmark. In this regard, the Dão-Lafões region of Portugal, which is a representative region of a country highly prone to wildfires, was selected in order to perform the comparison. The variables, implemented into the model and considered as favourable factors for wildfires, are the slope, the land use and vegetation covers, provided by the Corine Land Cover 2006 inventory. The official dataset of the national mapping BA was considered to train (2000-2009 period) and test (2010-2013 period) the models. The obtained results show that the two machine learning algorithms perform in an equal manner in terms of susceptibility areas and classes. Moreover, their susceptibility maps are broadly consistent with the one obtained with the benchmark model developed by the experts. However, the main benefit of the stochastic models resides on the fact, that they are data driven, meaning that they do not need a priori knowledge of the process and they can provide uncertainties assessments. Furthermore, the random forest algorithm has the ability to directly provide the measure of the importance of each variable. It is important to note that RF was able to automatically detect the relevant variables consistent with the expert knowledge. By presenting and applying both stochastic and deterministic methods to a highly sensitive and socially important case study, chapter 6 highlights new methods and approaches, which help in decision making and in understanding of both the phenomenon itself and its consequences. It would be important to apply and to test the approach for other regions sensible to the forest fires.

Results presented in the thesis demonstrate that the use of machine learning algorithms for environmental data analysis and modelling is not straightforward. A good understanding of the objectives and the limitations of such methods are essential for their correct application and interpretation of the results. An important contribution of this thesis deals with an elaboration of a self-consistent methodology that can be used for intelligent decision making process. The perspective of future researches in machine learning algorithms application to environmental data will deal with further elaboration of feature selection methods and quantification of modelling and prediction uncertainties in higher dimensional spaces. These achievements will have to be realized with the collaboration and communication between different environmental scientists, practitioners and decision makers.

6.2 Contributions

The main contributions of the thesis can be summarized as follows:

- Development and adaptation of the methodology for data driven environmental modelling based on machine learning.

- The application and adaptation of pre-processing, residuals analysis, validation and testing procedure based on Extreme Learning Machine.

- Development and investigation of multi-output Extreme Learning Machine algorithms.
- Complexity analysis on linear and non-linear Extreme Learning Machine models.
- Development and implementation of the efficient hybrid feature selection approach based on simulated annealing and Extreme Learning Machine modelling algorithms.
- Uncertainties quantification and visualization based on the combination of Extreme Learning Machine algorithm and a bootstrap-based procedure, which significantly contributes to the understanding of the phenomena under study.
- Challenging real data case studies dealing with environmental pollution, natural hazards, and renewable resources data.

Appendix A

Proceeding

A.1 Feature selection in environmental data mining combining Simulated Annealing and Extreme Learning Machine

This section presents the paper published in the proceedings of the ESANN 2014 conference (European Symposium on Artificial Neural Networks), which proposes a regression task of the section 3.2.2.

A.1.1 Introduction

Environmental science is a field in constant development. Because environmental phenomena lie in high dimensional spaces (e.g. for natural hazards: $d \approx 10 - 100$), it is challenging to reach the real dimension where the phenomena under study can be understood, explained and predicted (Kanevski, Pozdnoukhov, and Timonin, 2009). Moreover, in most real data cases the relationships between features and phenomena are non-linear. Keeping in mind that these relationships involve not only one but several features, the main goal is to select relevant subsets of features according to their potential non-linear ability to explain or predict environmental phenomena.

There are a lot of methods in wrapper, filter and embedded methodologies (Guyon and Elisseeff, 2003; Guyon et al., 2006; Lee and Verleysen, 2007). On the one hand filter methods are faster but do not necessarily take into account the combinations of various features simultaneously (a feature can be irrelevant alone but may be relevant with other features together). On the other hand wrapper methods allow complex associations of features but suffer from the curse of dimensionality when considering all possible combinations of features.

To address this challenge, this paper proposes a new methodology based on combining Extreme Learning Machine (ELM, Huang, Zhu, and Siew (2006)) and Simulated Annealing (SAN, Kirkpatrick, Gelatt, and Vecchi (1983)) algorithms. ELM has showed good capability for merging methods (Frénay and Verleysen, 2010) and SAN remains a good optimization algorithm despite the fact that it can perform faster by combining with a genetic algorithm (Gheyas and Smith, 2010). The principal advantages of this new method are the following: (1) ELM allows the quick evaluation of the non-linear potential of subsets of features, (2) SAN allows the optimal subset of features to be reached without using an exhaustive search. The use of ELM instead of the more robust and accurate OP-ELM (Miche et al., 2010) resides in the fact that current version of OP-ELM cancel out the wrapper ability to detect irrelevant feature. The methodology is described in Section 2. Section 3 presents the results using real and simulated data, and Section 4 concludes the paper.

A.1.2 Method

Extreme Learning Machine

The ELM algorithm follows the structure of a single-hidden layer feedforward neural network (SLFN) (Huang, Zhu, and Siew, 2006). For a given labelled training set $Z_{trn} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}\}_{i=1}^{N}$ and for a number of hidden nodes $\tilde{N}$, it computes the output matrix $N \times \tilde{N}$ of the hidden layer:

$$H_{ij} \quad = \quad g(\mathbf{x}_i \cdot \mathbf{w}_j + b_j)$$

where $\mathbf{w}_j$ (the vector of weights connecting the input layer with the j^{th} neuron) and b_j (the bias of the j^{th} neuron) are randomly generated. Then, the vector β (connecting the hidden layer with the output layer) is estimated using the Moore-Penrose generalized inverse of the matrix H:

$$\hat{\beta} \quad = \quad H^{\dagger}\mathbf{y}$$

Once all weights of the network are known, new data can be evaluated and error assessed using a hold-out validation set. Extremely fast, the only parameter that requires tuning is the number of hidden node $\tilde{N}$. See in Section A.1.3 how to deal with this parameter in order to preserve the computational time.

Simulated Annealing

SAN is a metaheuristic algorithm for optimization problems inspired by the field of metallurgy. Initialized with a high temperature parameter, it performs a global random search from neighbour to neighbour. In a second stage, temperature decreases progressively and the search becomes local. Based on the following Metropolis criterion (Metropolis et al., 1953), it has the capability to accept bad solutions according to the level of the current temperature T.

Let θ_{cur} and θ_{new} respectively be the current and new states of the research, and f the function to minimize. If $\Delta f = f(\theta_{cur}) - f(\theta_{new}) \leq 0$ the new state θ_{new} is accepted, else θ_{new} is accepted with a probability:

$$P = \exp(-\Delta f / T)$$

In a theoretical way, the ability to accept bad solutions allows us to find the global minimum of any kind of problem. In a practical way, it cannot guarantee finding the optimal solution but it can approach it. The success of this convergence lies in a good parametrization of the initial temperature and in the annealing process.

Feature Selection Methodology

Let n be the number of features available and $\Theta = \{\theta \mid \theta = \{0,1\}^n\}$ the set of the whole combination of features, where θ_i indicates if we consider feature i or not. The goal is to find $\theta^* \in \Theta$ that minimizes the cost function f defined as follows:

$$\begin{aligned} f(\theta) &= MSE(\mathbf{y}_{val}, \hat{\mathbf{y}}_{val}) \\ \text{where,} \qquad \hat{\mathbf{y}}_{val} &= ELM(\theta, \tilde{N}, Z_{trn}, Z_{val}) \end{aligned}$$

Z_{trn} and Z_{val} correspond to two separate training and validation sets, and $\tilde{N}$ is the number of hidden nodes. Without loss of generality, $\tilde{N}$ can be defined a priori (see experimental part A.1.3).

Applying this notation and using the simulated annealing algorithm, the proposed new feature selection algorithm is as follows:

Algorithm 2 SANELM

```
Require: Initialize $\theta_0 \in \Theta$ and $T_0$ the initial temperature
Generate a model with $ELM(\theta_0, \tilde{N}, Z_{trn}, Z_{val})$
Compute $f(\theta_0)$, and put $\theta_{cur} = \theta_0$
for $i = 1$ to $STOP$ do
    Compute $T_{new} = Ann(T_0, i)$
    Generate $\theta_{new}$ in the neighbourhood of $\theta_{cur}$
    Compute $f(\theta_{new})$ and $\Delta f = f(\theta_{cur}) - f(\theta_{new})$
    if $\Delta f \leq 0$ then
        Accept $\theta_{new}$: $\theta_{cur} \leftarrow \theta_{new}$
    else
        Generate $U$ uniformly in $[0,1]$, and compute $P = \exp(-\Delta f / T_{cur})$
        if $U \leq P$ then
            Accept $\theta_{new}$: $\theta_{cur} \leftarrow \theta_{new}$
        else
            Reject $\theta_{new}$
        end if
    end if
end for
```

For more details of the methodology, see section A.1.3.

A.1.3 Data and Results

Data

The data used for this application come from 200 measurement points in Lake Geneva. Composed of 3 real input variables (i.e. X, Y and Z coordinates), 21 simulated variables were added to the database. These additional input variables are composed of 3 shuffled variables from the original X, Y and Z coordinates, and of 18 random variables following a uniform distribution. Finally, the database was composed of 21 input variables and 1 output variable which is the pollutant, Nickel.

The principal objective is to investigate the parameter of the SANELM for this particular database, important for environmental risk studies and to evaluate the robustness and the accuracy of such methodology according to the parameters. The expected result is to find the original features, that is the X, Y and Z coordinates.

Experimental setup

First of all, the whole database must be normalized in order to fit to the range $[0,1]$ within which ELM works. Secondly, because of the need to assess the ELM model at each iteration of the SAN algorithm, the database must be split into two subsets. About 75 per cent of the data are allocated to the training set and the remaining 25 to the validation set.

Once the preprocessing task is completed, several SAN parameters have to be fitted. The first one is the annealing schedule $Ann(T_0,i)$. Written as a function of the initial temperature T_0 and the iteration index i, the schedule can take different forms. No preferential function exists, but as the optimization space Θ is discrete and not continuous, a basic schedule can be considered such as:

$$Ann(T_0,i) = \frac{T_0}{c \cdot i} \qquad \text{or} \qquad Ann(T_0,i) = \frac{T_0}{c \cdot \log(i)}$$

where c is the parameter of the schedule. In practice, since T_0 and c have to be parametrized, the most simple way is to fix $c = 1$ and to fit the parameter T_0 by trial and error.

Another important proceeding in the algorithm is the generation of a new state $\theta_{new} \in \Theta$ in the neighbourhood of the current state θ_{cur}. For this purpose, θ_{new} is defined as a neighbour of θ_{cur} if and only if the Hamming distance between the two is equal to 1 (i.e. θ_{new} and θ_{cur} differ in just one coefficient). This allows them to reach any state of the Θ space in at least n steps (where n is the number of input variable).

In order to complete the parameter setup, it remains to tune the number of hidden nodes $\tilde{N}$. In the first stage of the paper, an additional loop was added in the algorithm in order to compute $f(\theta_{new})$ with the optimal number of hidden nodes. Because this process is time consuming, an analysis of the distribution of the optimal number of node was carried out. It appears that this distribution shows the same range of optimal number of nodes for any kind of $\theta \in \Theta$. Furthermore, if we fix the number of nodes $\tilde{N}$ that is not necessary the optimal one for the desired best subset of features θ^*, it appears that

$$f(\theta^*) \leq f(\theta) \qquad \forall\theta \in \Theta$$

In other words, even if the model f is not perfect for a fixed number of hidden nodes $\tilde{N}$, it would be minimal for subset of relevant features.

Results

The first results show the stability of the methodology according to the choice of the number of hidden nodes $\tilde{N}$. For this purpose, 1000 subsets of features were generated randomly and all are evaluated with ELM for $\tilde{N} \in \{5, 10, 15, ..., 70\}$. In Figure A.1 each dashed line correspond to one random subset of features and the solid line coincides with the best subset of features. Examining 1000 random subsets of features reveals that the range of the number of hidden nodes where they reach the minimum value of MSE is approximately $[15, 30]$.

According to this first result, it is recommended that for each new problem the behaviour of $\tilde{N}$ is explored through randomly generative several subsets of features. By doing this, the range of the minimum number of nodes can be determined, and the SANELM algorithm can be performed using a fixed $\tilde{N}$ in that range.

By using the Lake Geneva database with the additional 18 irrelevant variables and with a fixed $\tilde{N} = 20$, the SANELM algorithm reaches the optimal subset of feature, that is the original X, Y and Z coordinates, in less than 4000 iterations. By comparison, the exhaustive search need $2^n - 1$ iterations (in this case more than 2 million) to evaluate all the possible combinations of features. The same results are obtained using different $\tilde{N} \in [15, 30]$.

A.1.4 Conclusion

This paper develops a combination of two algorithms, the Extreme Learning Machine as a wrapper method and the Simulated Annealing as an optimization algorithm. Analyses were performed in order to investigate the behaviour of both ELM and SAN parameters. As the optimization space is a discrete one, the annealing schedule of SAN can be standard. For the remaining T_0 and c parameters, trial and error are needed according to the complexity and dimensionality of the problem. For the unique ELM parameter $\tilde{N}$ (the number of hidden nodes), it has been shown that it is quite stable within the range determined by the problem. Therefore, $\tilde{N}$ can be fixed during the process and computational time can be reduced. In future research, this benefit will allow to investigate more complex phenomena in high dimensional space and multivariate data, as well as to perform a comprehensive comparison in computational time and accuracy with other feature selection algorithms.

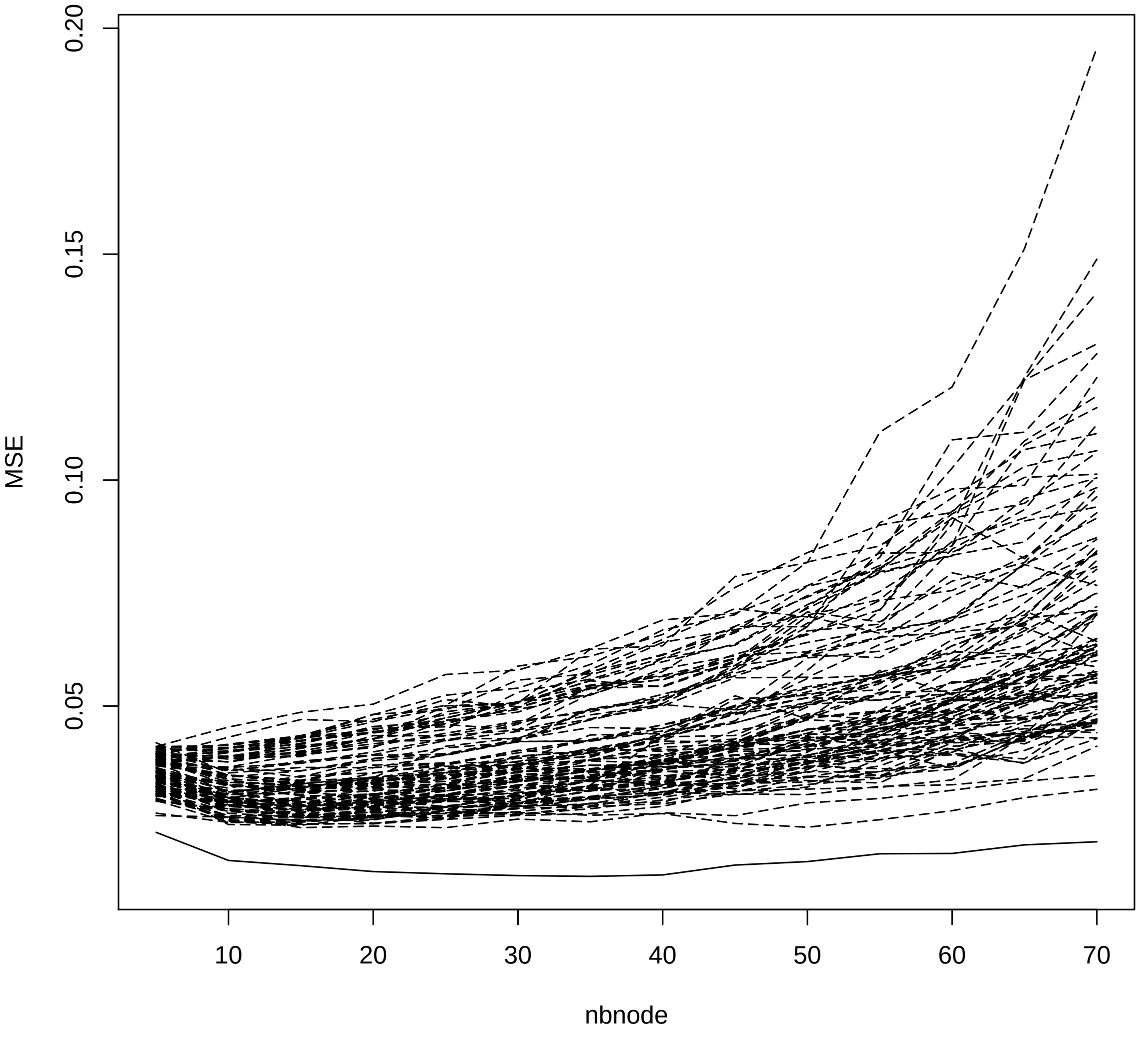


FIGURE A.1: Each dashed line correspond to one random subset of features and the solid line coincides with the best subset of features. The graph shows the MSE of the ELM for these different subsets of features according to the number of hidden nodes.

Bibliography

Abrahart, R., P. Kneale, and L See (2004). *Neural Networks for Hydrological Modelling*. Balkema Publishers.

Amatulli, G., A. Camia, and J. San-Miguel-Ayanz (2013). Estimating of future burned areas under changing climate in the EU-Mediterranean countries. In: *Science of the Total Environment* 450-451, pp. 209–222. DOI: 10.1016/j.scitotenv.2013.02.014.

Amraoui, M. et al. (2015). Atmospheric conditions associated with extreme fire activity in the Western Mediterranean region. In: *Science of the Total Environment* 524, pp. 32–39. DOI: 10.1016/j.scitotenv.2015.04.032.

Anderson, R. (2013). *Visual Data Mining. The VisMiner Approach*. Wiley.

Bache, K. and M. Lichman (2013). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. URL: http://archive.ics.uci.edu/ml.

Barros, A. M. and J. M Pereira (2014). Wildfire Selectivity for Land Cover Type: Does Size Matter? In: *PLoS ONE* 9.1. DOI: 10.1371/journal.pone.0084760.

Beverly, J. L., E. P. K. Herd, and J. C. R. Conner (2009). Modeling fire susceptibility in west central Alberta, Canada. In: *Forest Ecology and Management* 258.7, pp. 1465–1478. DOI: 10.1016/j.foreco.2009.06.052.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bnanankhah, A. and F. Nejadkoorki (2012). Artificial Neural Networks: A Non-Linear Tool for Air Quality Modeling and Monitoring. In: *International Conference on Applied Life Sciences (ICALS2012)*, pp. 81–85.

Bolón-Canedo, V., N. Sánchez-Maroño, and A. Alonso-Betanzos (2013). A review of feature selection methods on synthetic data. In: *Knowledge and Information Systems* 34.3, pp. 483–519.

Bonazountas, M. et al. (2005). Forest Fire Risk Analysis. In: *Human and Ecological Risk Assessment: An International Journal* 11.3, pp. 617–626. DOI: 10.1080/10807030590949717.

Breiman, L. (2001). Random forests. In: *Machine learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324.

Breiman, L. et al. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth, Brooks/Cole Advanced Books, and Software.

Brenning, A. (2005). Spatial prediction models for landslide hazards: review, comparison and evaluation. In: *Natural Hazards and Earth System Sciences* 5, pp. 853–862.

Büttner, G. (2014). “CORINE Land Cover and Land Cover Change Products”. In: *Land Use and Land Cover Mapping in Europe: Practices & Trends*. Ed. by Ioannis Manakos and Matthias Braun. Dordrecht: Springer Netherlands. Chap. 5, pp. 55–74. ISBN: 978-94-007-7969-3. DOI: 10.1007/978-94-007-7969-3_5.

Caetano, M., V. Nunes, and A. Nunes (2009). *CORINE land cover 2006 for continental Portugal*. Instituto Geográfico Português.

Cervone, G. et al. (2008). Risk assessment of atmospheric emissions using machine learning. In: *Natural Hazard and Earth Sciences* 8, pp. 991–1000. DOI: 10.5194/nhess-8-991-2008.

Chang, K. and B. Tsai (1991). The effect of DEM resolution on slope and aspect mapping. In: *Cartography and Geographic Information Systems* 18.1, pp. 69–77. DOI: 10.1559/152304091783-805626.

Cheng, T. and J. Wang (2008). Integrated Spatio-Temporal Data Mining for Forest Fire Prediction. In: *Transactions in GIS* 12.5, pp. 591–611.

Cherkassky, V. and F. M. Mulier (2007). *Learning from Data: Concepts, Theory and Methods*. 2nd ed. Wiley, p. 538.

Cherkassky, V. et al. (2006). Computational intelligence in earth sciences and environmental applications: Issues and challeenges. In: *Neural Networks* 19.2, pp. 113–121.

Chorowski, J., J. Wang, and J. M. Zurada (2015). Review and performance comparison of SVM- and ELM-based classifiers. In: *Neurocomputing* 128, pp. 507–516. DOI: 10.1016/j.neucom.2013.08.009.

Chuvieco, E. and J. Salas (1996). Mapping the spatial distribution of forest fire danger using GIS. In: *International journal of geographical information systems* 10.3, pp. 333–345. DOI: 10.1080/02693799608902082.

Chuvieco, E. et al. (2010). Development of a framework for fire risk assessment using remote sensing and geographic information system technologies. In: *Ecological Modelling* 221.1, pp. 46–58. DOI: 10.1016/j.ecomodel.2008.11.017.

Chuvieco, Emilio and Russell G. Congalton (1989). Application of remote sensing and geographic information systems to forest fire hazard mapping. In: *Remote Sensing of Environment* 29.2, pp. 147–159. DOI: 10.1016/0034-4257(89)90023-0.

CIPEL (2008). *CIPEL*. International Commission for the Protection of Lake Geneva. URL: http://www.cipel.org/sp/.

Cracknell, M. J. and A. M. Reading (2014). Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. In: *Computers & Geosciences* 63, pp. 22–33. DOI: 10.1016/j.cageo.2013.10.008.

DL (2006). *Decreto-Lei no. 124/2006 de 28 de Junho*. Diário Da República - I SERIE-A, p. 123.

Donalek, C. et al. (2013). Feature selection strategies for classifying high dimensional astronomical data sets. In: *IEEE International Conference on Big Data*, pp. 35–41.

Dubois, G. (2005). *Automatic mapping algorithms for routine and emergency data*. European Commission, JRC Ispra, EUR 21595, p. 123.

Efron, B. and R. Tibshirani (1986). Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical accuracy. In: *Statistical Science* 1, pp. 54–75.

Eugenio, F. C. et al. (2016). Applying GIS to develop a model for forest fire risk: A case study in Espírito Santo, Brazil. In: *Journal of Environmental Management* 173, pp. 65–71. DOI: 10.1016/j.jenvman.2016.02.021.

Fernandes, P. M. et al. (2016). Bottom-up variables govern large-fire size in Portugal. In: *Ecosystems* 19.8, pp. 1362–1375. DOI: 10.1007/s10021-016-0010-2.

Filippone, M., F. Masulli, and S. Rovetta (2011). Simulated annealing for supervised gene selection. In: *Soft Computing* 15, pp. 1471–1482.

Freire, S., H. Carrão, and M. R. Caetano (2002). *Produção de cartografia de risco de incêndio florestal com recurso a imagens de satélite e dados auxiliares*. Lisboa, IGP.

Frénay, B. and M. Verleysen (2010). “Using SVMs with randomised feature spaces: an extreme learning approach”. In: *Proceedings of the 18th European Symposium on Artificial Neural Networks*. Ed. by M. Verleysen. d-side pub., pp. 315–320.

Frénay, B. et al. (2013). Feature selection for nonlinear models with extreme learning machines. In: *Neurocomputing* 102, pp. 111–124.

Ganteaume, A. et al. (2013). A Review of the Main Driving Factors of Forest Fire Ignition Over Europe. In: *Environmental Management* 51.3, pp. 651–662. DOI: 10.1007/s00267-012-9961-z.

Gardnera, M. W. and S. R. Dorlinga (1998). Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. In: *Atmospheric Environment* 32.14-15, pp. 2627–2636. DOI: 10.1016/S1352-2310(97)00447-0.

Gheyas, I. and L. Smith (2010). Feature subset selection in large dimensionality domains. In: *Pattern Recognition* 43, pp. 5–13.

Golay, J., M. Leuenberger, and M. Kanevski (2017). Feature selection for regression problems based on the Morisita estimator of intrinsic dimension. In: *Pattern Recognition* 70, pp. 126–138.

Gonçalves, J. A. and A. M. Morgado (2008). Use of the SRTM DEM as a Geo-referencing Tool by Elevation Matching. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37, pp. 879–883.

Gorman, R. P. and T. J. Sejnowski (1988). Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. In: *Neural Network* 1, pp. 75–89.

Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. In: *The Journal of Machine Learning Research* 3, pp. 1157–1182.

Guyon, I. et al. (2006). *Feaure extraction: foundations and applications*. Springer.

Hamming, R. W. (1950). Error detecting and error correcting codes. In: *Bell System Technical Journal* 29.2, pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

Hassan, R. and M. Li (2010). “Urban Air Pollution Forecasting Using Artificial Intelligence-Based Tools”. In: *Air Pollution*. Ed. by V. Villanyi. InTech. Chap. 9. ISBN: 978-953-307-143-5.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, p. 754. URL: `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`.

Haupt, S., A. Pasini, and C. Marzban (2009). *Artificial Intelligence Methods in the Environmental Sciences*. Springer.

Haykin, S. (2008). *Neural Networks and Learning Machines*. 3rd ed. Prentice Hall, p. 936.

Heskes, T. (1997). “Practical Confidence and Prediction Intervals”. In: *Advances in Neural Information Processing Systems 9*. Ed. by M. C. Mozer, M. I. Jordan, and T. Petsche. MIT Press, pp. 176–182.

Hsieh, W. (2009). *Machine Learning Methods in the Environmental Sciences*. Cambridge University Press, p. 364.

Huang, C.-L. and J.-F. Dun (2008). A distributed PSO-SVM hybrid system with feature selection and parameter optimization. In: *Applied Soft Computing* 8.4, pp. 1381–1391.

Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew (2006). Extreme learning machine: theory and applications. In: *Neurocomputing* 70.1-3, pp. 489–501. DOI: `10.1016/j.neucom.2005.12.126`.

ICNF (2016). *O que são os PROF?* URL: `http://www.icnf.pt/portal/florestas/profs/obj`.

Jaiswal, R. K. et al. (2002). Forest fire risk zone mapping from satellite imagery and GIS. In: *International Journal of Applied Earth Observation and Geoinformation* 4.1, pp. 1–10. DOI: `10.1016/S0303-2434(02)00006-5`.

Kanevski, M. (2013). A Methodology for Automatic Analysis and Modeling of Spatial Environmental Data. In: *ScGEOProcessing 2013: The Fifth International Conference on Advanced Geographic Information Systems, Applications, and Servises* 5, pp. 105–107.

Kanevski, M. and M. Maignan (2004). *Analysis and Modelling of Spatial Environmental Data*. EPFL Press; Lausanne, Switzerland.

Kanevski, M., A. Pozdnoukhov, and V. Timonin (2009). *Machine Learning for Spatial Environmental Data*. EPFL Press; Lausanne, Switzerland, p. 392.

Kanevski, M. et al. (2004). Environmental data mining and modeling based on machine learning algorithms and geostatistics. In: *Environmental Modelling and Software* 19, pp. 845–855. DOI: `10.1016/j.envsoft.2003.03.004`.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. In: *Science* 220, pp. 671–680.

Kohavi, R. and G. H. John (1997). Wrappers for feature subset selection. In: *Artificial Intelligence* 97, pp. 273–324. DOI: `10.1016/S0004-3702(97)00043-X`.

Kohonen, T. (2001). *Self-Organizing Maps*. 3rd ed. Springer-Verlag, Berlin.

Krasnopolsky, V. M. and Y. Lin (2012). A Neural Network Nonlinear Multimodel Ensemble to Improve Precipitation Forecasts over Continental US. In: *Advances in Meteorology*. DOI: `10.1155/2012/649450`.

Lampin-Maillet, C. et al. (2010). "Wildland urban interfaces, fire behaviour and vulnerability: characterization, mapping and assessment". In: *Towards Integrated Fire Management - Outcomes of the European Project Fire Paradox*. Ed. by J. S. Silva et al. European Forest Institute. Chap. 3.3, pp. 71–92. ISBN: 970-952-5453-48-5.

Lee, J. A. and M. Verleysen (2007). *Nonlinear Dimensionality Reduction*. Information Science and Statistics, Springer.

Leuenberger, M. and M. Kanevski (2014). "Feature selection in environmental data mining combining Simulated Annealing and Extreme Learning Machine". In: *Proceedings of the 22th European Symposium on Artificial Neural Networks*. Ed. by M. Verleysen. d-side pub., pp. 601–606. ISBN: 978-287419095-7.

— (2015). Extreme Learning Machines for spatial environmental data. In: *Computers and Geosciences* 85, pp. 64–73. DOI: `10.1016/j.cageo.2015.06.020`.

— (2016). *Feature Selection and Modelling with Extreme Learning Machine. Case study: Wind Fields in Complex Regions*. 11th International Conference on Geostatistics for Environmental Application (GeoENV) , Lisbon (Portugal).

Leuenberger, M. et al. (2018). Wildfire susceptibility mapping: deterministic vs. stochastic approaches. In: *Environmental Modelling & Software* 101, pp. 194–203. DOI: `10.1016/j.envsoft.2017.12.019`.

Li, S. et al. (2010). Investigating spatial non-stationary and scale-dependent relationships between urban surface temperature and environmental factors using geographically weighted regression. In: *Environmental Modelling and Software* 25.12, pp. 1789–1800. DOI: `10.1016/j.envsoft.2010.06.011`.

Liitiäinen, E. et al. (2009). Residual variance estimation in machine learning. In: *Neurocomputing* 72, pp. 3692–3703. DOI: `10.1016/j.neucom.2009.07.004`.

Lin, S.-W. et al. (2008a). A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks. In: *Expert Systems with Applications* 34, pp. 1491–1499.

Lin, S.-W. et al. (2008b). Parameter determination of support vector machine and feature selection using simulated annealing approach. In: *Applied Soft Computing* 8, pp. 1505–1512.

Liu, Y. et al. (2011). An Improved Particle Swarm Optimization for Feature Selection. In: *Journal of Bionic Engineering* 8.2, pp. 191–200.

Marques, S. et al. (2011). Characterization of wildfires in Portugal. In: *European Journal of Forest Research* 130.5, pp. 775–784. DOI: `10.1007/s10342-010-0470-4`.

Marvuglia, A. and A. Messineo (2012). Monitoring of wind farms' power curves using machine learning techniques. In: *Applied Energy* 98, pp. 574–583. DOI: 10.1016/j.apenergy.2012.04.037.

May, R. J., H. R. Maier, and G. C. Dandy (2010). Data splitting for artificial neural networks using SOM-based stratified sampling. In: *Neural Networks* 23, pp. 283–294. DOI: 10.1016/j.neunet.2009.11.009.

Meiri, R. and J. Zahavi (2006). Using simulated annealing to optimize the feature selection problem in marketing applications. In: *European Journal of Operational Research* 171, pp. 842–858.

Melchiorre, C. and E. A. Castellanos Abella (2011). Evaluation of prediction capability, robustness, and sensitivity in non-linear landslide susceptibility models, Guantánamo, Cuba. In: *Computers and Geosciences* 37, pp. 410–425. DOI: 10.1016/j.cageo.2010.10.004.

Metropolis, N. et al. (1953). Equation of State Calculations by Fast Computing Machines. In: *The Journal of Chemical Physics* 21, pp. 1087–1092.

Miche, Y. et al. (2010). OP-ELM Optimally Pruned Extreme Learning Machine. In: *IEEE Transactions on Neural Networks* 21.1, pp. 158–162.

Micheletti, N. et al. (2014). Machine learning feature selection methods for landslide susceptibility mapping. In: *Mathematical and Geosciences* 46.1, pp. 33–57. DOI: 10.1007/s11004-013-9511-0.

Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. In: *Bulletin of the American Mathematical Society* 26.9, pp. 394–395. DOI: 10.1090/S0002-9904-1920-03322-7.

Moreira, F. et al. (2009). Regional variation in wildfire susceptibility of land-cover types in Portugal: implications for landscape management to minimize fire hazard. In: *International Journal of Wildland Fire* 18.5, pp. 563–574. DOI: 10.1071/WF07098.

Moreno, R. et al. (2014). Extreme learning machines for soybean classification in remote sensing hyperspectral images. In: *Neurocomputing* 128, pp. 207–216. DOI: 10.1016/j.neucom.2013.03.057.

Murphy, K. (2012). *Machine Learning. A Probabilistic Perspective*. MIT Press.

Nagendra, S. M. S. and M. Khare (2005). Modelling urban qir quality using artificial neural network. In: *Clean Technologies and Environmental Policy* 7.2, pp. 119–126.

Nefeslioglu, H. A., C. Gokceoglu, and H. Sonmez (2008). An assessment on the use of logistic regression and artificial neural networks woth different sampling strategies for the preparation of landslide susceptibility maps. In: *Engineering Geology* 97, pp. 171–191. DOI: 10.1016/j.enggeo.2008.01.004.

Neteler, M. et al. (2011). Terra and Aqua satellites track tiger mosquito invasion: modelling the potential distribution of Aedes albopictus in north-eastern Italy. In: *International Journal of Health Geographics* 10.1. DOI: 10.1186/1476-072X-10-49.

Nunes, A. N., L. Lourenço, and A. C. C. Meira (2016). Exploring spatial patterns and drivers of forest fires in Portugal (1980-2014). In: *Science of the Total Environment* 573, pp. 1190–1202. DOI: 10.1016/j.scitotenv.2016.03.121.

Oliveira, S. et al. (2012). Modeling spatial patterns of fire occurrence in Mediterranean Europe using Multiple Regression and Random Forest. In: *Forest Ecology and Management* 275, pp. 117–129. DOI: 10.1016/j.foreco.2012.03.003.

Parente, J. and M. G. Pereira (2016). Structural fire risk: The case of Portugal. In: *Science of the Total Environment* 573, pp. 883–893. DOI: 10.1016/j.scitotenv.2016.08.164.

Parente, J., M. G. Pereira, and M. Tonini (2016). Space-time clustering analysis of wildfires: The influence of dataset characteristics, fire prevention policy decisions, weather and climate. In: *Science of the Total Environment* 559, pp. 151–165. DOI: 10.1016/j.scitotenv.2016.03.129.

Pasero, E. and L. Mesin (2010). "Artificial Neural Networks for Pollution Forecast". In: *Air Pollution*. Ed. by V. Villanyi. InTech. Chap. 10. ISBN: 978-953-307-143-5.

Pausas, J. G. and J. E. Keeley (2009). A Burning Story: The Role of Fire in the History of Life. In: *BioScience* 59.7, pp. 593–601. DOI: 10.1525/bio.2009.59.7.10.

Pearl, J. (1985). Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning. In: *Proceedings of the 7th Conference of the Cognitive Science Society, University of California* 7, pp. 329–334.

Pereira, M. G., J. Aranha, and M. Amraoui (2014). Land cover fire proneness in Europe. In: *Forest Systems* 23.3, pp. 598–610. DOI: 10.5424/fs/2014233-06115.

Pereira, M. G. et al. (2013). Effects of regional climate change on rural fires in Portugal. In: *Climate research* 57.3, pp. 187–200. DOI: 10.3354/cr01176.

Platt, J. (2000). "Probabilities for SV Machines". In: *Advances in Large Margin Classifiers*. Ed. by A. J. Smola. MIT Press, pp. 61–74.

Pourghasemi, H. R. (2016). GIS-based forest fire susceptibility mapping in Iran: a comparison between evidential belief function and binary logistic regression models. In: *Scandinavian Journal of Forest Research* 31.1, pp. 80–98. DOI: 10.1080/02827581.2015.1052750.

Pourghasemi, H. R., M. Beheshtirad, and B. Pradhan (2016). A comparative assessment of prediction capabilities of modified analytical hierarchy process (M-AHP) and Mamdani fuzzy logic models using Netcad-GIS for forest fire susceptibility mapping. In: *Geomatics, Natural Hazards and Risk* 7.2, pp. 861–885. DOI: 10.1080/19475705.2014.984247.

Pourtaghi, Z. S., H. R. Pourghasemi, and M. Rossi (2015). Forest fire susceptibility mapping in the Minudasht forests, Golestan province, Iran. In: *Environmental Earth Sciences* 73.4, pp. 1515–1533. DOI: 10.1007/s12665-014-3502-4.

Pourtaghi, Z. S. et al. (2016). Investigation of general indicators influencing on forest fire and its susceptibility modeling using different data mining techniques. In: *Ecological Indicators* 64, pp. 72–84. DOI: 10.1016/j.ecolind.2015.12.030.

Pozdnoukhov, A. et al. (2011). Spatio-temporal avalanche forecasting with Support Vector Machines. In: *Natural Hazards and Earth System Sciences* 11, pp. 367–382. DOI: 10.5194/nhess-11-367-2011.

Pradhan, B. (2013). A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS. In: *Computers & Geosciences* 51, pp. 350–365. DOI: 10.1016/j.cageo.2012.08.023.

Press, W. L. et al. (2007). *Numerical Recipes*. 3rd ed. Cambridge University Press, New York.

QGIS Development Team (2016). *QGIS Geographic Information System*. Open Source Geospatial Foundation. URL: http://qgis.osgeo.org.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rakotomamonjy, A. et al. (2008). SimpleMKL. In: *Journal of Machine Learning Research* 9, pp. 2491–2521.

Rissanen, J. (1978). Modeling by shortest data description. In: *Automatica* 14.5, pp. 465–658. DOI: 10.1016/0005-1098(78)90005-5.

Robert, S., L. Foresti, and M. Kanevski (2012). Spatial prediction of monthly wind speeds in complex terrain with adaptive general regression neural networks. In: *International Journal of Climatology* 33.7, pp. 1793–1804. DOI: 10.1002/joc.3550.

Rodrigues, M., J. de la Riva, and S. Fotheringham (2014). Modeling the spatial variation of the explanatory factors of human-caused wildfires in Spain using geographically weighted logistic regression. In: *Applied Geography* 48, pp. 52–63. DOI: 10.1016/j.apgeog.2014.01.011.

Rosenblatt, F. (1961). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington DC.

Rothermel, R. C. (1972). *A mathematical model for predicting fire spread in wildland fuels*.

Rowe, J. S. and G. W. Scotter (1973). Fire in the boreal forest. In: *Quaternary Research* 3.3, pp. 444–464. DOI: 10.1016/0033-5894(73)90008-2.

San-Miguel-Ayanz, Jesús et al. (2012). "Comprehensive Monitoring of Wildfires in Europe: The European Forest Fire Information System (EFFIS)". In: *Approaches to Managing Disaster - Assessing Hazards, Emergencies and Disaster Impacts*. Ed. by John Tiefenbacher. InTech. Chap. 5, pp. 87–108. ISBN: 978-953-51-0294-6. DOI: 10.5772/28441.

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. cambridge University Press.

Shrestha, D., N. Kayastha, and D. Solomatine (2009). ANNs and other Machine Learning Techniques in Modelling Models' Uncertainty. In: *ICANN* 2, pp. 387–396.

Silva, J. S. and S. P. Harrison (2010). "Humans, Climate and Land Cover as Controls on European Fire Regimes". In: *Towards Integrated Fire Management - Outcomes of the European Project Fire Paradox*. Ed. by J. S. Silva et al. European Forest Institute. Chap. 3.1, pp. 49–59. ISBN: 970-952-5453-48-5.

Silva, J. S. et al. (2009). Assessing the relative fire proneness of different forest types in Portugal. In: *Plant Biosystems - An International Journal Dealing with all Aspects of Plant Biology* 143.3, pp. 597–609. DOI: 10.1080/11263500903233250.

Soto, M. E. C. et al. (2013). A territorial fire vulnerability model for Mediterranean ecosystems in South America. In: *Ecological Informatics* 13, pp. 106–113. DOI: 10.1016/j.ecoinf.2012.06.004.

Specht, D. (1991). A General Regression Neural Network. In: *IEEE Transactions on Neural Networks* 2, pp. 568–576.

Specht, D. and H. Romsdahl (1994). Experience with Adaptive Probabilistic Neural Networks and Adaptive General Regression Neural Networks. In: *Proceedings of the IEEE World Congress on Computational Intelligence* 2, pp. 1203–1208.

Sun, Y., Y. Yuan, and G. Wang (2014). Extreme learning machine for classification over uncertain data. In: *Neurocomputing* 128, pp. 500–506. DOI: 10.1016/j.neucom.2013.08.011.

Thompson, S. (2012). *Sampling*. 3rd ed. Wiley.

Tonini, M. et al. (2017). Evolution of forest fires in Portugal: from spatio-temporal point events to smoothed density maps. In: *Natural Hazards* 85.3, pp. 1489–1510. DOI: 10.1007/s11069-016-2637-x.

Trigo, R. M. et al. (2006). Atmospheric conditions associated with the exceptional fire season of 2003 in Portugal. In: *International Journal of Climatology* 26.13, pp. 1741–1757. DOI: 10.1002/joc.1333.

Unler, A. and A. Murat (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. In: *European Journal of Operational research* 206.3, pp. 528–539.

van der Maaten, L. J. P., E. O. Postma, and H. J. van den Herik (2009). *Dimensionality reduction: a comparative review*. Technical Report TiCC-TR 2009-005, Tilburg University.

Vapnik, V. (1998). *Statistical Learning Theory*. 1st ed. Wiley, p. 768.

Vasconcelos, M. P. et al. (2001). Spatial prediction of fire ignition probabilities: Comparing logistic regression and neural networks. In: *Photogrammetric engineering and remote sensing* 67.1, pp. 73–81.

Verde, J. C. (2015). *Wildfire Susceptibility Modelling in Mainland Portugal*.

Verde, J. C. and J. L. Zêzere (2010). Assessment and validation of wildfire susceptibility and hazard in Portugal. In: *Natural Hazards ans Earth System Sciences* 10.3, pp. 485–497. DOI: 10.5194/nhess-10-485-2010.

Wan, C. et al. (2014). Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine. In: *IEEE Transactions on Power Systems* 29.3, pp. 1033–1044.

Whitlock, C. et al. (2010). Paleoecological Perspectives on Fire Ecology: Revisiting the Fire-Regime Concept. In: *The Open Ecology Journal* 3, pp. 6–23. DOI: 10.2174/1874213001003020006.

Xu, C. et al. (2012). GIS-based support vector machine modeling of earthquake-triggered landslide susceptibility in the Jianjiang River watershed, China. In: *Geomorphology* 145-146, pp. 70–80. DOI: 10.1016/j.geomorph.2011.12.040.