Aleksei Shkurin

WATER QUALITY ANALYSIS USING MACHINE LEARNING ALGORITHMS

Bachelor's Thesis
in Environmental Engineering

December 2015

MAMK
University of Applied Sciences

DESCRIPTION

MAMK University of Applied Sciences

Date of the bachelor's thesis

29.03.2016

Author(s)

Aleksei Shkurin

Degree programme and option

Environmental Engineering

Name of the bachelor's thesis

Water Quality Analysis Using Machine Learning Algorithms

Abstract

Data analysis is one of the key engines of progress in most of areas of the research in natural sciences, including environmental sciences. Nowadays, continuous development and technological progress provide us with universal and advanced tools for data analysis, such as machine learning algorithms.

The purpose of the research behind this thesis is to provide examples to the engineers and scientists working in environmental field of how these models can be implemented towards the environmental tasks.

The data used for the research is water quality data, produced during the STREAMES (STream REAch Management, an Expert System) project, initially aiming at producing tools for increasing the quality of European rivers.

In this report one will find examples of data imputation, regression, classification, clusterization and feature selection tasks using machine learning algorithms, such as: random forest, support vector machines, neural networks, k-nearest neighbours, and k-means clustering.

Subject headings, (keywords)
Classification, Regression, Water Quality, Machine Learning, Random Forest, Neural Network, SVM, Feature Selection, Imputation

Pages	**Language**	**URN**
54	English	

Remarks, notes on appendices

Appendix: R code

Tutor	**Bachelor’s thesis assigned by**
Eino Sarkola	

CONTENTS

1 INTRODUCTION AND BACKGROUND 1

1.1 Data Mining 1

1.1.1 Environmental Informatics 2

1.2 Machine Learning 2

1.2.1 Overfitting 4

1.2.2 Calculating Accuracy 5

2 MATERIALS AND METHODS 8

2.1 Data 8

2.2 Models and Software 10

2.2.1 Support Vector Machine (SVM) 11

2.2.2 Random Forest 12

2.2.3 Artificial Neural Network (ANN) 14

2.2.4 K-nearest Neighbours (KNN) 16

2.2.5 K-means Clustering 18

2.2.6 Cross-Validation 19

2.2.7 R 21

3 RESULTS 22

3.1 Data Imputation 22

3.2 Regression and Variable Importance 22

3.2.1 NH_4 23

3.2.2 NO_3 25

3.2.3 PO_4 26

3.2.4 Overfitting 28

3.3 Classification 29

3.4 Cluster Analysis 29

4 DISCUSSION 31

5 CONCLUSION 35

6 BIBLIOGRAPHY 37

APPENDIX

R code

1 INTRODUCTION AND BACKGROUND

Since the early beginning of the development of natural sciences, collecting and assay of huge amounts of data was one of the leading analytical tools. The same goes for environmental sciences and environmental engineering, which produce higher demand for efficient and productive approaches to work with continuously increasing sizes of the collected data from a huge variety of research fields every day. (Kendall and Costello 2006)

Nowadays, machine learning algorithms have proven themselves as a universal tool for different types of tasks, giving advanced possibilities for dealing with analysed data, including such types of tasks as *data imputation*, *unsupervised clusterization*, *classification* and *regression*. They are commonly used in many research areas; however, they are yet less common among environmental engineering workers, though such tools may provide an extremely efficient alternative to the traditional analytical approaches. (Wilcox, Woon and Aung 2013)

The purpose of the research behind this thesis was in presenting of examples of how such advanced tools may be used on a particular data set meant for increasing water quality in european region. In the following chapters one will go through the presentation of the machine learning, it's origins and possibilities in general, explanation of the data and models used during the research, results of the application of algorithms, discussion (covering obstacles one can face while working with this kind of models) and conclusion, which will cover the presented material, give advices for engineers and scientists who would like to use this models for their environmental tasks and finally and give some words about the possbile future of the development of these tools in environmental field.

1.1 Data Mining

Formally, the beginning of data analysis field had begun at the moment humanity started making simplest analysis of the surrounding environment by watching and manually interacting with nature. For data mining itself, there are some more or less consistent and defined events in history that are associated with the birth of the discipline, such as publishing of Bayes' theorem (which describes the probability of an event, based on conditions that might be related to the event) by Thomas Bayes' in 1763 and first regression analysis by Adrien-Marie Legendre and Carl Friedrich Gauss in 1805 (Figure 1.1). (Li 2015)

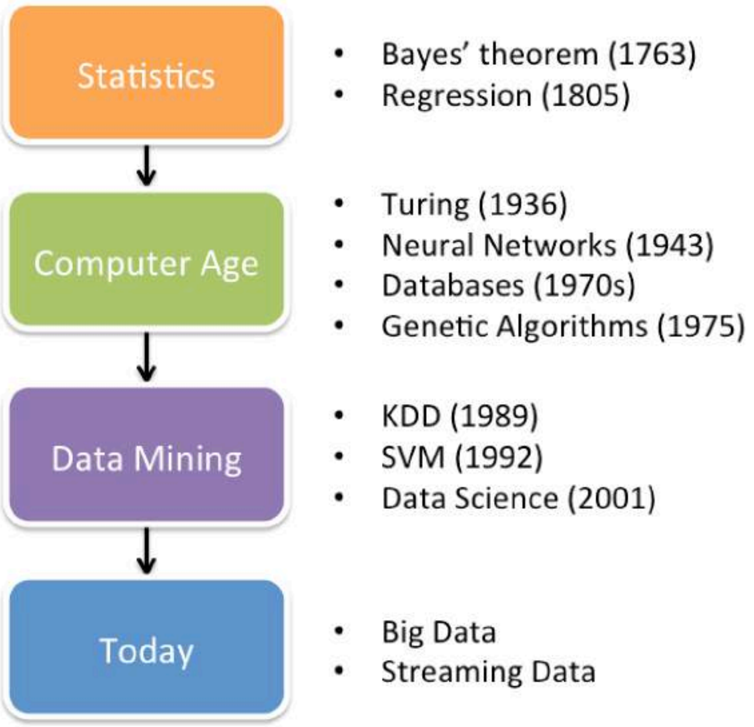

Figure 1.1 Stages of Data Mining development (Li 2015)

Data mining has changed far and away ever since, the most known and general models, such as Neural Networks and SVM were presented and nowadays one of the most advanced directions of the development of the field is working with Big and Streaming Data. In the following subchapter, one will go deeper in the concept of machine learning, types of the tasks it can help dealing with, review some of the common obstacles and finally go through the most common techniques used for evaluating the performance of the models.

1.1.1 Environmental Informatics

Generally, this research may be associated with the new and currently rapidly growing field of environmental informatics. Being one of the directions of the development of data sciences, environmental informatics covers researches that work with data about the state of Earth's biosphere (and associated spheres) and those processes affecting it. Thus, being interested in reviewing and analysing more projects and articles in this field, one should consider searching for information primarily in this particular area. (Frew and Dozier 2012)

1.2 Machine Learning

Since times of Bayes' theorem, data mining has greatly developed (especially since the beginning of the computer age) and Machine Learning separated from it as an independent scientific field. There are two the most common definitions of this term. First is provided by

Arthur Samuel In 1959, who described it as a "Field of study that gives computers the ability to learn without being explicitly programmed" (Simon 2013). And the more formal definition by Tom M. Mitchell:"A computer program, said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell 1998).

One can define also the following main steps of the analysis using machine learning models:

1. *Data Understanding* – before defining the possible approaches to work with data, it is necessary to analyse the raw data itself first. What kind of measurements are included, is there any missing data (and in case of natural sciences research, usually there is plenty), which kind of models it is possible to apply to the data and defining the initial goal of the research
2. *Data Preparation* – merging data, imputing missing values or excluding variables with too many missing values, sorting data, etc.
3. *Model Training* – actually training the models and analyzing data
4. *Results Evaluation* – an important stage of the results understanding, which makes possible adjustment of the models and correction of the initial research plan

(Chapman, et al. 2000)

Additionaly, it is worth defining and explaining the main types of models one can apply:

1. *Supervised learning* - these are methods where a given set of independent variables are to be matched to one or more dependent variables. During this kind of analysis, model is given a “labled data”, where it can find the real values of the parameter it is working with for some certain measurement and values of other parameters for the same measurement, thus it can fit a function. These can be *regression* tasks (working with continuous values) and *classification* tasks (working with class labeled data)
2. *Unsupervised learning* - in contrast, with unsupervised methods there is no prior “correct” data and the purpose of this kind of analysis is to search for the underlying patterns in the data
3. *Optimization* - techniques for finding the optimal set of parameters which minimize a pre-defined cost function

(Wilcox, Woon and Aung 2013)

1.2.1 Overfitting

Overfitting is a common problem among a vast majority of the machine learning algorithms and it comes from the origins of these tools. While fitting a model, algorithm tries to minimize the error function of the model, which is the difference between the estimator and what is estimated (Lebanon 2010). On Figure 1.2 one can see an example of the three typical types of a model fit:

1. *Predictor too inflexible* – these models are underfitted, which means that the formula describes data poorly and mean square error (MSE) is significantly high. These models can't provide sufficial degree of accuracy
2. *Middle pictures* – this model has nearly a perfect fit. The MSE is not absolutely zero; however, this model will provide an appropriate function to the given data and will be able to make suitable predictions based on new data (or data not included in training set)
3. *Predictor is too flexible* – in these cases, we can see a typical example of overfitting. The model is too complicated and it is able to minimize MSE nearly to zero, so it seems to fit the training set alone perfectly. However, it will describe random errors or noise and thus show pretty bad results on the new data

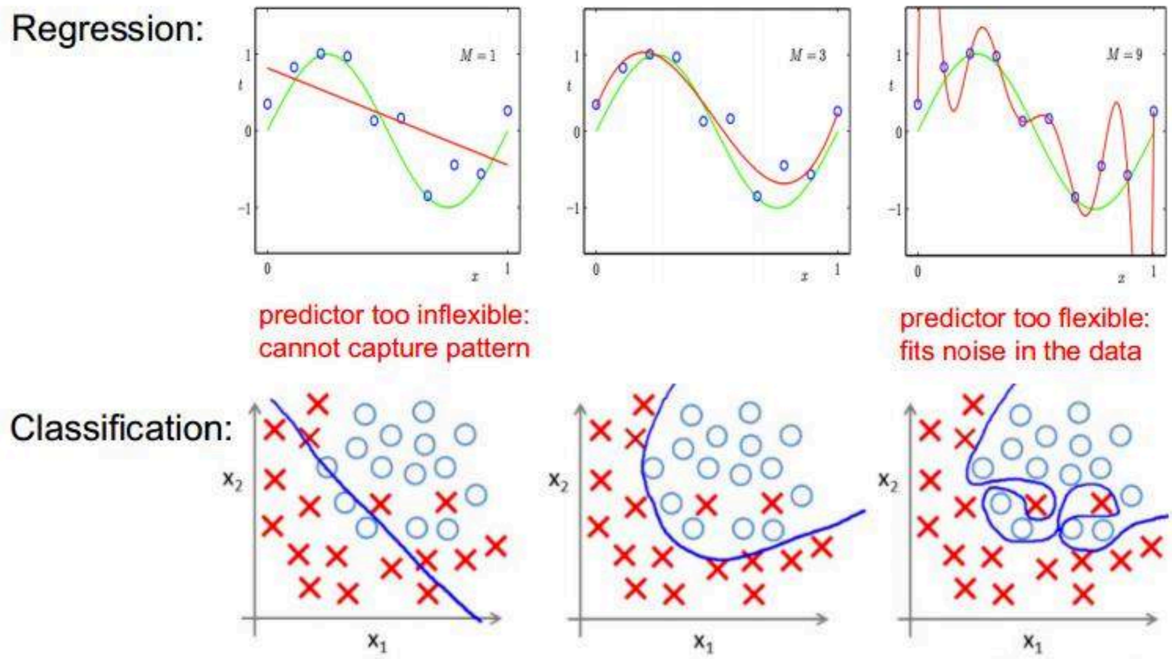

Figure 1.2 Examples of *overfitting* and *underfitting* for *regression* and *classification* tasks (Lavrenko and Goddard 2015)

In this thesis, problem of the overfitting will be met; in addition to that, the most common and the most useful models for the kind of data used for the research were used: *supervised regression* and *classification*, *unsupervised clustering,* together with *data imputation* and *feature selection*. In the following subchapter one can find information about possible ways to evaluate performance of these models.

1.2.2 Calculating Accuracy

Goals of the evaluation stage can't be effectively met without proper ways to interpret the achieved results. For different models and different types of tasks, various techniques may be used for this purpose. During this research, the following ones were used: confusion matrix, root mean square error and node purity.

Confusion matrixes are one of the most commonly used tools for evaluating a performance of classification models. It contains information about the actual and predicted classes. Table 1.1 shows the basic idea of a confusion matrix. (Hamilton 2012)

Table 1.1 Concept of a confusion matrix; *a* is the number of correct predictions that an instance is negative; *b* is the number of incorrect predictions that an instance is positive; *c* is the number of incorrect of predictions that an instance negative; *d* is the number of correct predictions that an instance is positive

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

There are several calculations one can do using these matrixes:

1. The *accuracy* (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a + d}{a + c + b + d} \qquad [1]$$

2. The *recall* or *true positive rate* (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{d}{c + d} \qquad [2]$$

3. The *false positive rate* (FP) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = \frac{b}{a + b} \qquad [3]$$

4. The *true negative rate* (TN) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$TN = \frac{a}{a + b} \qquad [4]$$

5. The *false negative rate* (FN) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{c}{c + d} \qquad [5]$$

6. *Precision* (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{d}{b + d} \qquad [6]$$

7. Finally, the accuracy determined using equation for AC might not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases. Suppose there are 1000 cases, 995 of which are negative cases and 5 of which are positive cases. If the system classifies them all as negative, the accuracy would be 99.5%, even though the classifier missed all positive cases. Other performance measures account for this by including TP in a product: for example, geometric mean (g-mean):

$$g - mean_1 = \sqrt{TP * P} \qquad [7]$$

$$g - mean_2 = \sqrt{TP * TN} \qquad [8]$$

or by using F-Measure, where *ß* has a value from 0 to infinity and is used to control the weight assigned to TP and P:

$$F = \frac{(\beta^2 + 1) * P * TP}{\beta^2 * P + TP} \qquad [9]$$

(Hamilton 2012)

Mean square error, or MSE of an estimator θ' of a parameter θ is the function of θ defined b

$$\sum (\theta' - \theta)^2 \qquad [10]$$

This is also called the risk function of an estimator, with $(\theta' - \theta)^2$ called the quadratic loss function. It is used primarily while working with continuous numbers and fitting regression models. (Lebanon 2010, Blackwell, Honaker and King 2015)

Also, during the analysis of model's performance, root mean square error was used (RMSE).

$$\sqrt{1/n \sum (\theta' - \theta)^2} \qquad [11]$$

(Jensen 2013)

Finally, node purity is a measurement typical for decision trees, such as the ones behind the concept of a random forest used during this research. The idea is to calculate how much the particular split reduces node impurity (for regression trees, the difference between residual sum of squares (RSS) before and after the split), where RSS is calculated by the formula:

$$\sum e_i^2 = \sum (\theta' - \theta)^2 \qquad [12]$$

(Hastie, Tibshirani and Friedman 2009)

2 MATERIALS AND METHODS

This chapter gives information about the data, followed by the explanation of the algorithms and overview of the software used during the analysis.

2.1 Data

The data used for this research was generated during European STREAMES (STream REAch Management, an Expert System) project, which is an international enterprise for the development of a knowledge-based environmental decision support system to assist water managers with their decision-making tasks. The core of the project itself involved the evaluation of the effect of substantial nutrient loads on the overall water quality and ecological status of stream ecosystems. Empirical data for the knowledge base come from several streams located throughout Europe and Israel, with emphasis on streams from the Mediterranean region. These data comprise several types of variables, including physical, chemical and biological parameters. (Vellido, et al. 2007)

The original data contains 110 measurements of 52 variables for 9 streams. The characteristics of the streams chosen for the research can be found in Table 2.1.

Table 2.1 Catchment characteristics of the chosen streams (Vellido, et al. 2007)

Stream	Dominant Geology	Climate	Catch ment area (km^2)	Stream length (km)	Altitudinal range (m a.s.l.)	Land-use (%)		
						Arable and grass-land	Forest and open land	Urban
Tordera (Spain)	Siliceous	Mediterra-nean	80.2	21.7	1100-190	10.8	87.4	1.8
Grandola (Portugal)	Siliceous	Mediterra-nean	54.9	40.1	258-11	15.8	83.0	1.2
Apose-lemis (Greece)	Calcareous	Mediterra-nean	19.6	4.6	902-240	42.3	57.1	0.4
Montagut (France)	Calcareous	Atlantic	12.9	8.0	620-320	49.4	50.6	0
Bagnatore (Italy)	Calcareous	Mediterra-nean	11.0	5.0	828-470	59.7	36.0	4.4
Erpe (Germany)	Siliceous	Sub-continental	207	20.0	65-38	60.0	21.0	19.0

Gurri (Spain)	Calcareous	Mediterra-nean	37.7	14.3	1140-503	60.7	35.2	4.0
Lezat (France)	Calcareous	Atlantic	226.1	44.0	620-207	79.0	20.9	0.1
Demnitzer (Germany)	Siliceous	Mediterra-nean	15.0	6.2	67-60	100	0	0

For this particular study, out of all 52 variables, the most significant 29 variables were chosen during the data preparation process. These variables are presented in Table 2.2.

Table 2.2 List of the 29 variables selected for the study, grouped by their topology (Vellido, et al. 2007)

Type	Variable	Description
Ion Concentrations (chemical)	Cations	$Na^{+} + K^{+} + Mg^{2+} + Ca^{2+} + NH^{+}4$ (Concentration in meq/l)
	Anions	$Cl- + SO-_{4} + NO-3$ (Concentration in meq/l)
	Alkalinity	(Concentration in meq/l)
Nutrient Concentration (chemical)	NH_4^+-N	Ammonium (concentration in mgN/l)
	NO_3^--N	Nitrate (concentration in mgN/l)
	PO_4^3-N	Phosphate (concentration in mgP/l)
	D.O.C.	Dissolved Organic Carbon (Concentration in mg/l)
	Conductivity	In µS/cm
	D.I.N.	Dissolved Inorganic Nitrogen (in mgN/l)
Hydrological, Hydrau-lic & Morphologic (physical)	Depth	Wet channel average depth (m)
	Wet Perimeter	Cross-sectional area divided by depth
	Substrate Ratio	Percentage of (Cobbles þ Pebbles) substrata, divided by percentage of (Gravel þ Sand þ Silt) substrata
	Wet Perimeter: Depth Ration	Ratio between Wet Perimeter and average Depth (unitless)
	K1	Water transient storage exchange coefficient: from water column to transient storage zone (in s^{-1})
	K2	Water transient storage exchange coefficient: from transient storage zone to water column (in s^{-1})
	Transient Storage Ratio	K1/K2

	Froude number	$v/(g*D)^{1/2}$, where *v* is Average Water Velocity as defined below, *g* is the gravitational acceleration and *D* is the hydraulic depth
	Reynolds number	$(v*D)/KV$, where *v* and *D* as above and *KV* is the kinematic viscosity
	Discharge	In m^3/s
	Average Water Velocity	In m/s
	Manning's Coefficient	$(h^{2/3}*s^{1/2})/v$, where *v* as above, *h* is the wet channel depth and *s* is the reach slope
Stream Metabolism & Biofilm (biological)	Respiration	Daily rate of ecosystem respiration (in g O2/m^2)
	G.P.P.	Daily rate of gross primary production (in g O2/m^2)
	G.P.P.:R	G.P.P. to Respiration ratio (unitless) per day
	Daily Light (P.A.R.)	In mol/m^2
	Temperature	Average temperature at midday (in □^{O}C)
	D.O. Range	Daily variation in dissolved oxygen concentration (in mg O2/l)
	Chlorophyll	In mg/m^2
	Biomass	In mgAFDM/m^2 (AFDM: Ash-Free Dry Mass)

The chosen dataset contains an average level of 5.3% missing values. Some of the variables were dropped due to high amount of the missing data, which makes imputation process useless and some of the variables were dropped due to their obvious meaningless for the analysis. In addition to the explained variables, some basic information about the measurements was included in order to show the examples of fitting the classification algorithm: *season* of the measurement and *land use* (forested or agricultural).

2.2 Models and Software

There is a huge variety of machine learning algorithms and tools existing nowadays. In the following subchapters, the following algorithms used in this research are covered: support vector machines, random forests, artificial neural networks (used for classification, regression, variable importance tasks), k-nearest neighbours (used for data imputation) and k-means clustering (used for unsupervised classification).

2.2.1 Support Vector Machine (SVM)

Support vector machine is one of the basic algorithms and in this research is used mostly as a baseline in order to be able to compare the performances of the models. The core of this algorithm refers to the family of linear models. The model is trained by transfering of the original vector in the space of higher dimension and search for *dividing hyperplane* with the maximum gap in this space. Two parallel hyperplanes are constructed on both sides of the hyperplane separating classes. Separating hyperplane is a hyperplane that maximizes the distance to two parallel hyperplanes. The algorithm works on the assumption that the greater the difference and the distance between these parallel hyperplanes, the smaller the average error of the classifier (Figure 2.1). (Cortes and Vapnik 1995)

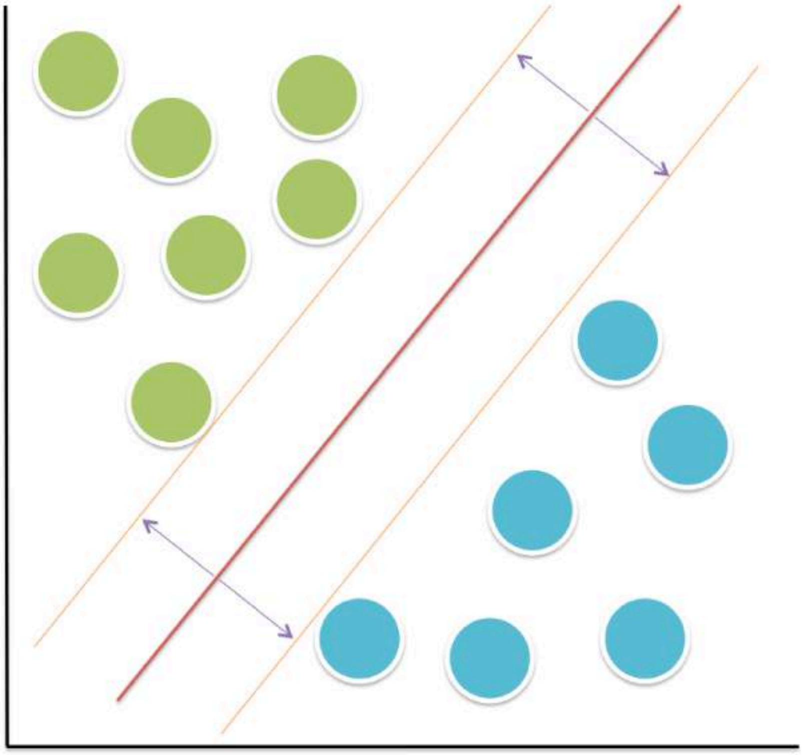

Figure 2.1 General graphical scheme of SVM algorithm. The model based on two hyperplanes with the maximum distance separates *green* and *blue* classes

The representation of the basic SVM algorithm from mathematical point of view is shown below:

1. Let's assume that one has input and output datasets X and Y, where $x \in X$, $y \in Y$ and the training set (x_1, y_1) ... (x_n, y_n)
2. The goal is to learn a classifier $y = f\ (x, \alpha)$, where α are the parameters of the function

3. Function $f\ (x,\alpha)$ learns by choosing a function that has minimum error rate, which is calculated by

$$R_{emp}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i,\alpha),y_i) \qquad [13]$$

where l is zero-one loss-function $l(y,\acute{y}) = 1$, if $y \neq \acute{y}$ and 0 otherwise. R_{emp} is also called the empirical risk

4. This helps us to minimize the overall risk

$$R(\alpha) = \int l(f(x,\alpha),y)\ dP(x,y) \qquad [14]$$

Where $P\ (x,y)$ is unknown joint distribution function of x and y

5. Finally, we choose the set of hyperplanes, so

$$\frac{1}{n}\sum_{i=1}^{n} l(f(w \cdot x_i + b, y_i) + ||w||^2 \qquad [15]$$

Is a subject to $min_i|w \cdot x_i| = 1$, where b is bias unit and $||w||^2$ is a complexity term and can be optimized in different ways depending on the actual task

(Weston 2004)

2.2.2 Random Forest

Random forest (RF) is a relatively new model, developed by Leo Breiman and Adele Cutler in the end of 90s. The general graphical scheme of the RF algorithm is sketched in Figure 2.2. At each split of the observed sample data, a random subset of variables is selected and the process is repeated until the specified number of *decision trees* is generated. Each tree is built from a bootstrap sample drawn with replacement from the observed data, and the predictions of all trees are finally aggregated through majority voting. A feature of RFs is the definition of an out-of-bag (OOB) error, which is calculated from observations that were not used to build a particular tree; it can thus be considered as an internal cross-validation error measure. This is an important feature for the type of experiments carried out in this study, because it simplifies the otherwise cumbersome cross-validation procedures that would be required if alternative classification methods such as, for instance, support vector machines or artificial neural networks were used. (Breiman 2001, Shkurin and Vellido 2016)

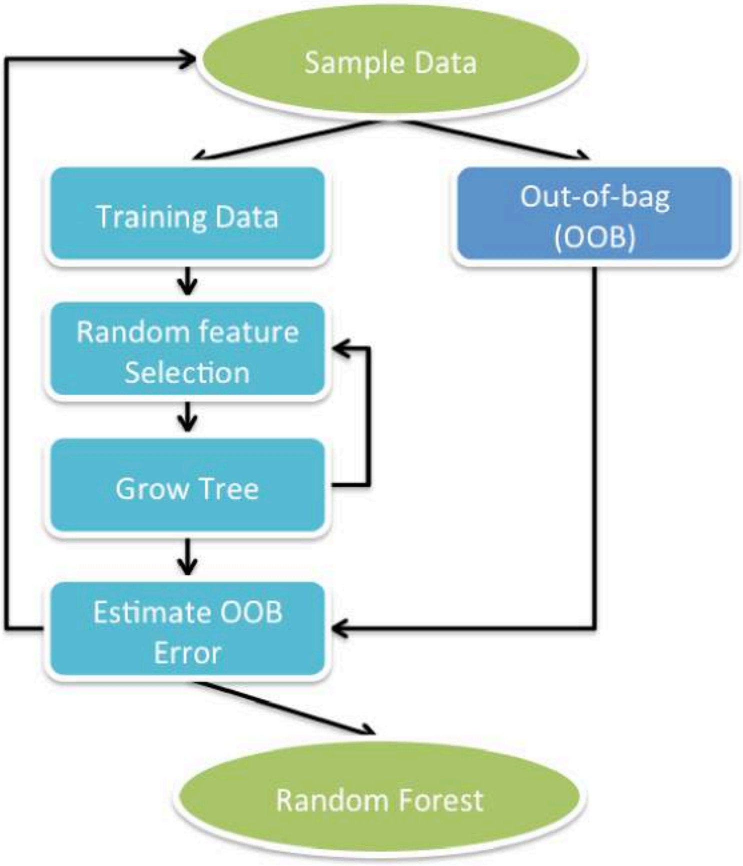

Figure 2.2 General graphical scheme of RF algorithm

One of the capabilities this model provides is performing a feature selection, which helps not only to simplify the procedure and reduce computational costs of the analysis, but also helps to understand the relation between variables and dependency of one variable with another. The variable importance measure here is based on weighted sums of the absolute regression coefficients. The weights are a function of the reduction of the sums of squares across the number of PLS (Partial Least Squares) components and are computed separately for each outcome. Therefore, the contribution of the coefficients is weighted proportionally to the reduction in the sums of squares (Kuhn 2015). It is a significant advantage especially for environmentally related tasks, since these processes are significantly complicated and involve a huge amount of matter and reactions of different kind. Collecting of large amounts of data and performing feature selection may help to identify the reasons of some event and find the related processes.

Also, following is the mathematical representation of the RF algorithm.

1. The following procedure is repeated $\lceil \log_2 k_n \rceil$ times, where log$_2$ is the base-2 logarithm, $\lceil . \rceil$ the ceiling function and $k_n \geq 2$ is a deterministic parameter defined by user and usually depending on *n* size of the data set
2. At each node, a coordinate of $x = \{x_1 \dots x_n\}$ is selected, with the j-th feature having a probability of $p_{nj} \in \ (0,1)$ to be selected
3. At each node, the split is performed at the midpoint of the chosen site, after the coordinate is selected
4. Each random decision tree $r_n = (\mathrm{x}, \Theta)$ (where Θ is an output of the randomized variable), gives average over all variables y_i that were selected to the same cell as x_i during random partition. Or in other words, letting $A_n(x, \Theta)$ be the rectangular cell of the random partition, which includes x_i:

$$r_n(\mathrm{x}, \Theta) = \frac{\sum_{i=1}^{n} y_i [x_i \in A_n(x, \Theta)]}{\sum_{i=1}^{n} [x_i \in A_n(x, \Theta)]} E_n(\mathrm{x}, \Theta) \qquad [16]$$

   where $E_n(\mathrm{x}, \Theta)$ is defined as:

$$E_n(\mathrm{x}, \Theta) = \left[\sum_{i=1}^{n} [x_i \in A_n(x, \Theta)] \right] \qquad [17]$$

5. Finally, the discrimination function (or combination function) is to be applied to all the outputs produced during the previous stages of all the single randomly generated trees. Assuming that *t* trees were generated and one is fitting a classification model, the following function can be applied. Let the probability of *x* belonging to some certain class c $(c = 1 \dots m)$, be defined by $P\ (c|v_j(x))$, where $v_j(x)$ is the terminal node *x* is assigned to:

$$P\left(c \middle| v_j(x)\right) = \frac{P(c, v_j(x))}{\sum_{l=1}^{m} P(c, v_j(x))} \qquad [18]$$

   Then, discrimination function $g_c(x)$ can be defined as:

$$g_c(x) = \frac{1}{t} \sum_{j=1}^{t} \acute{P}\ (c|v_j(x)) \qquad [19]$$

   And the decision criterion is: *x* belongs to the class *c*, for which $g_c(x)$ is the maximum

(Ho 1995, Biau 2012)

2.2.3 Artificial Neural Network (ANN)

The artificial neural networks (ANN) are one of the most popular machine learning models nowadays, with a huge variety of possible applications, including regression, classification, image recognition etc, introduced in 1943 by neurophysiologist Warren McCulloch and mathematician Walter Pitts (Warren and Pitts 1943). The basic of this model is in building several *layers* that are made up of a number of *interconnected nodes*, containing the *activation function*. The training set is presented to a model through input layer; one or more hidden layers perform processing by the system of weighted connections, taking each of the inputs for calculation, and finally output layer gives the fitted function (see Figure 2.3). (Cheng and Titterington 1994)

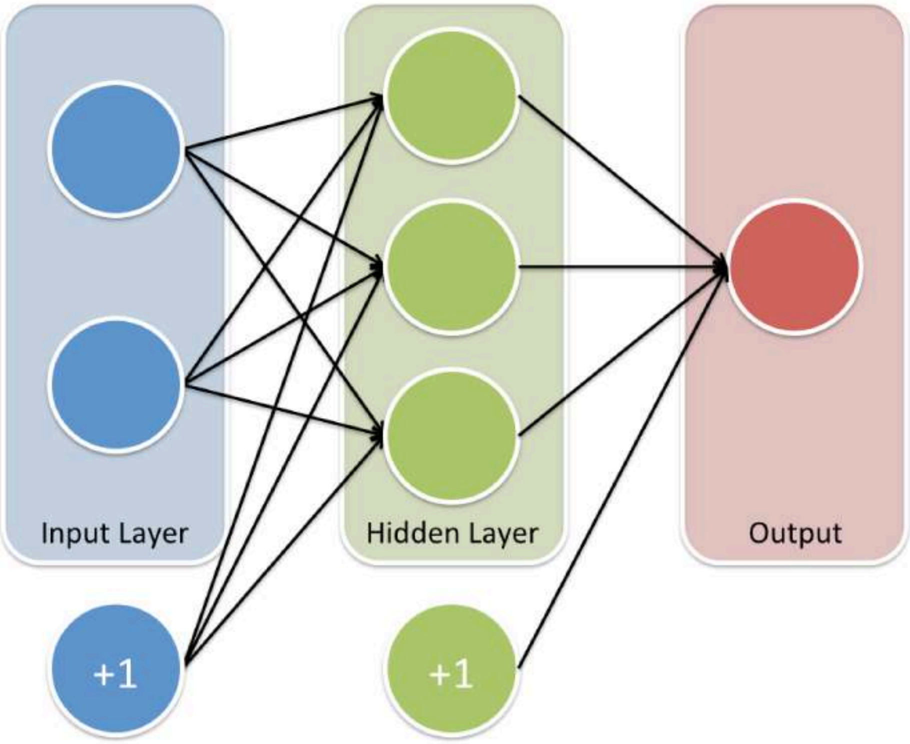

Figure 2.3 General graphical scheme of ANN model; *+1 nodes* represent bias units that help shifting the activation function depending on the task

The following is a step-by-step algorithm of ANN:

1. Assuming that one has the same datasets as in example with SVM, neural network will provide a way to define a complex, non-linear form of hypotheses $h_{W,b}(x)$, with parameters W - weight and b - bias are to be fitted to the data
2. Each neuron from $1 \dots m$, where m is the amount of neurons in the model, takes $x_1 \dots x_n$ inputs +1 bias term (or intercept term, b) and outputs

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{m} W_i x_i + b) \qquad [20]$$

where f is an activation function. Let's assume for this example $f\ (\cdot)$ is a sigmoid function (one of the most common functions to use), defined by

$$f(z) = \frac{1}{1 + \exp{(-z)}} \qquad [21]$$

3. Let's denote number of layers in network by n_l and we label layer l as L_l. Weights $W_{ij}^{(l)}$ denotes weight associated with the connection between unit j in layer l and unit i in layer $l + 1$; bias $b_i^{(l)}$ is associated with unit i in layer l and $a_i^{(l)}$ will denote the activation (or output) value of unit i in layer l. Then, performance of each node in each layer except for the input layer can be described by equation:

$$a_i^{(l)} = f(W_{i1}^{l-1}x_1 \dots W_{ij}^{l-1}x_j + b_i^{l-1}) \qquad [22]$$

   As outputs of layer l become inputs of layer $l + 1$, the final model $h_{W,b}(x)$ is calculated the same way, using inputs of a previous layer

(Hagan, et al. 2002)

2.2.4 K-nearest Neighbours (KNN)

A k-nearest neighbours is one of the basic and naturally obvious algorithms for tasks such as classification; in this particular study it was used for the missing data imputation (replacing missing values with the closest possible value). Generally, it is possible to use any model for imputation purposes; however, KNN is used during this study, since it is able to give sufficient results while keeping computational costs in acceptable limits, thus it is commonly used for such purposes (Batista and Monard 2002). The idea behind the model is that the missing value of a measurement is assigned to the value, which is the most common among the specified amount of neighbours of a given measurement. (Lu, et al. 2012)

The graphical representation of the algorithm can be seen on Figure 2.4.

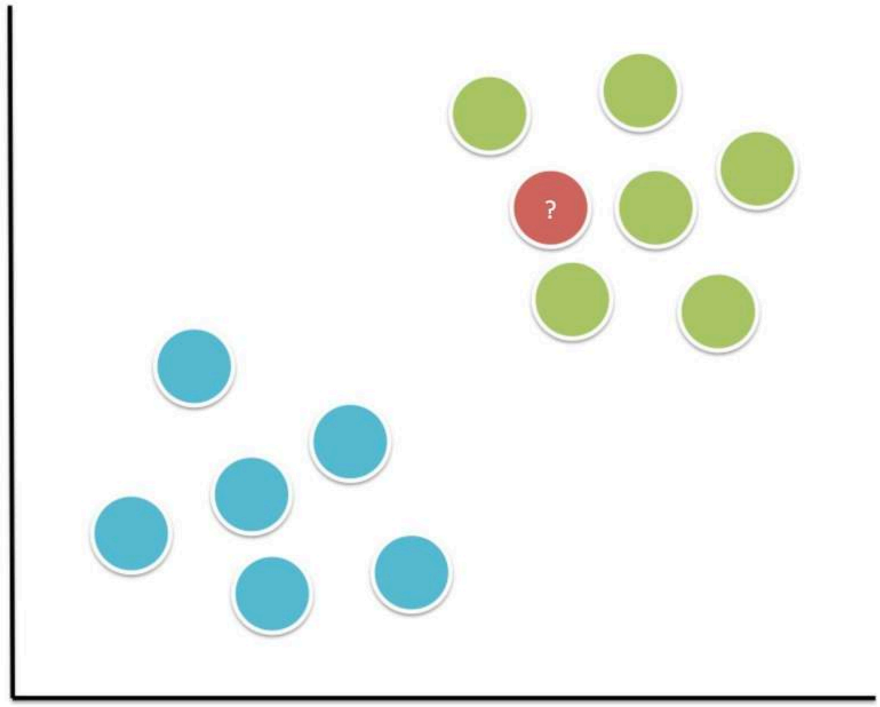

Figure 2.4 Graphical representation of KNN algorithm; the *red* value will be assigned to the same class as the *green* values in this case

The model can be used for both, classification and regression tasks:

1. *KNN classification* – the value is assigned to a class the most common among it's k neighbours
2. *KNN regression* – the value is assigned to a mean value of it's k neighbours

Mathematical representation of the algorithm is given below:

1. Let's firstly define the distance function $d\ (x_i, x_j)$, where x is the same features as in example with SVM. One of the examples of distance function is Euclidean distance measuring

$$d_E\ (x_i, x_j) = \sum_{i=1}^{n} \sqrt{x_i^2 - x_j^2} \qquad [23]$$

2. The result y of the model is calculated by majority voting of k neighbours with the lowest result of distance function

$$y = \frac{1}{k} \sum_{i=1}^{k} y_i \qquad [24]$$

where y_i is the i-th case of the examples sample

(Lu, et al. 2012)

In this research, KNN is used for imputing all the missing values that are presented in the data. These are the values from: cation, anion, alkaline, substrate ratio, K1, K2, NO_3, PO_4 and chlorophyll measurements.

2.2.5 K-means Clustering

One of the useful analytical tools is unsupervised clusterization, where the data is classified by the algorithm into specified amount of classes based on internal patterns. It can be used to search for the subtypes and subclasses for researched process, value or compound. (Likas, Vlassis and Verbeek 2003)

The data is classified firstly by setting k centroids, which will be the core to the searched classes. Then, the grouping is done by minimizing the sum of squares of distances (analogue to MSE) between data and the corresponding cluster centroid, as shown on the Figure 2.5. At each iteration cluster centre is recalculated until the best position is reached. (Teknomo 2007)

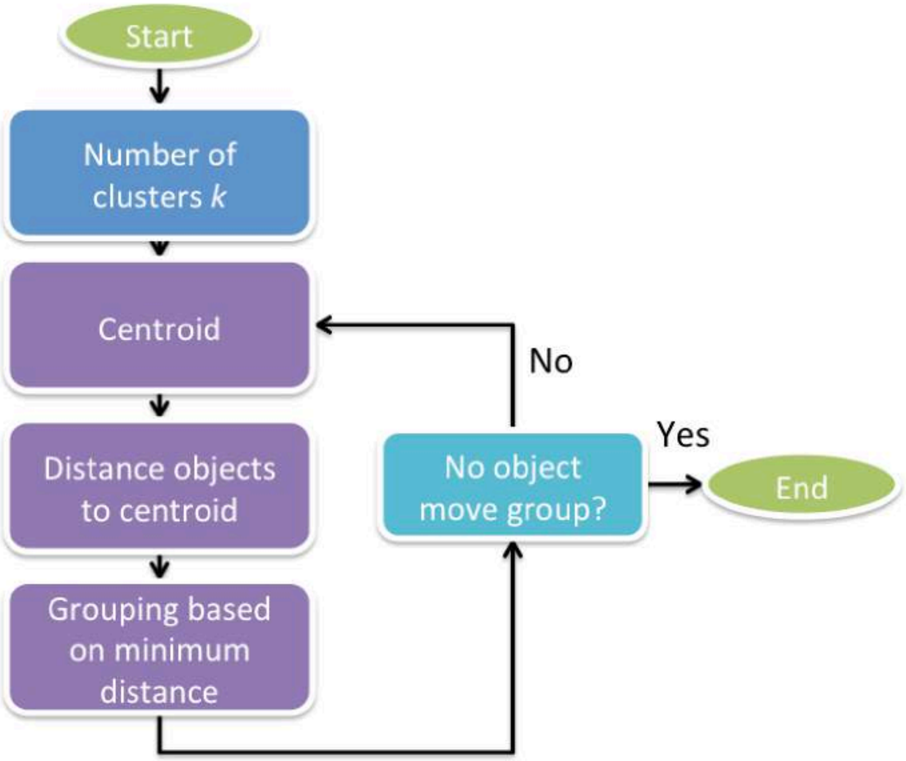

Figure 2.5 k-means clustering algorithm

Below is the mathematical representation of the algorithm:

1. Specified amount of centroids c is defined randomly

2. Distance function (i.e. Euclidean distance measuring shown in K-nearest Neighbours (KNN) chapter) is used to assign objects to their closest centroids
3. At each iteration objective function J is recalculated in order to minimize total intra-cluster variance

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2 \qquad [25]$$

Where k is number of classes, n is number of cases; $x_i^{(j)}$ is case i; c_j is a centroid for cluster j and $|| \cdot ||$ is a distance function

(Wagstaff, et al. 2001)

2.2.6 Cross-Validation

Cross-validation is a commonly used model evaluation method for reducing scales of overfitting and increasing accuracy of the model. It is one of the methods that is applied in addition while training other models. The result is an assessment of the effectiveness of the chosen model with the most even usage of available data. (Starkweather 2011)

There are several types of cross-validation, which all have their pros and cons:

8. *Hold-Out validation* – the idea here is in dividing the data into training and test set (usually 70% is training and 30% is test). The model is trained based on first 70% and then results of the prediction for the remaining 30% is compared with the actual results, possibly calculating MSE or RMSE. It is a naturally obvious way to estimate the performance, though it can’t be used while adjusting the models and it has obvious cons, since the amount of data used for training is reduced
9. *k-Fold Cross-Validation* – the general idea of this technique is in dividing the data in *k* parts. Then model is trained using *n-1* parts of data and the rest of data is used for testing. The procedure is repeated *n* times; as a result, each of *n* pieces of data used for testing. This method is commonly used and was used for this research as well. It makes possible using all the data for training and adjusting the model after each iteration
10. *Leave-one-out cross-validation (LOOCV)* – is a special case of k-Fold validation, where on each iteration all data except for one observation is used for training and assessment is done using this one observation

(Refaeilzadeh, Tang and Liu 2009)

The performance of these models is compared in Table 2.3.

Table 2.3 Comparison of Cross-Validation techniques (Refaeilzadeh, Tang and Liu 2009)

Validation Method	Pros	Cons
Hold-out Validation	Independent training and test	Reduced data for training and testing; Large variance
k-Fold Cross-Validation	Accurate performance estimation	Small samples of performance estimation; Overlapped training data; Elevated Type I error for comparison; Underestimated performance variance or overestimated degree of freedom for comparison
Leave-one-out cross-validation (LOOCV)	Unbiased performance estimation	Very large variance

Below is the procedure of performing k-Fold cross-validation:

1. Divide the set $(x_1 \dots x_n)$ into K subsets (i.e. folds) of roughly similar size – $F_1 \dots F_K$, randomly or stratified (with the respect to the sizes of the classes in case of classification task)
2. For k=1...K:
    - Consider training on $(x_i; y_i), i \notin F_k$, and validating on $(x_i; y_i), i \in F_k$
    - For each value of the tuning parameter $\theta \in (\theta_1 \dots \theta_m)$, compute the estimate $\acute{f}_\theta^{-k}$ on the training set and record the total error on validation set:

$$e_k(\theta) = \sum_{i \in F_k} (y_i - \acute{f}_\theta^{-k}(x_i))^2 \qquad [26]$$

3. For each tuning parameter value θ, compute the average error over all folds:

$$CV(\theta) = \frac{1}{n} \sum_{k=1}^{K} e_k(\theta) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in F_k} (y_i - \acute{f}_\theta^{-k}(x_i))^2 \qquad [27]$$

(Browne 2000, Krstajic, et al. 2014)

This is the procedure behind the 10-fold cross-validation used for most of the models in this research.

2.2.7 R

All the calculations were done using R, which is a free object-oriented data analysis language and software environment for statistical computing and graphics, distributed under GNU General Public License.

The packages used while working on this research are: *caret*, *DMwR*, *randomForest*, *GGally*, *RGtk2*, *rattle*, *rpart, rpart.plot*, *e1071, MASS, nnet* and *devtools.*

In order to ensure reproducibility, one has to set the random number generator (RNG) state to a value of 42. Each model of a random forest was stratified and included 500 trees.

3 RESULTS

The research was done in several stages: the data was prepared and imputed; regression models were made for NH_4, NO_3 and PO_4 values using both: imputed data and the data with missing values; feature selection was done for all the models of NH_4, NO_3 and PO_4 values; new regression models were built using the most significant values found during feature selection process; classification models were done for season and land use values, making a models that can predict these values for water quality data; clusterization was performed for the whole dataset and it's decision tree for NH_4, NO_3 and PO_4 values generated. In the following subchapters one will go through all these stages.

3.1 Data Imputation

First of all, the data was imputed. In the original data, missing values are presented as "0" and "-1" values. After these were assigned to NA, KNN with 2, 3 and 4 neighbours were used for imputation of the results. The accuracies of the models based on these numbers of neighbours for NH_4, NO_3 and PO_4 values are presented in Table 3.1.

Table 3.1 Results of data imputation with different values for *k*

Measurement	*k* value of neighbours	Accuracy of variables explained (in %)
NH_4	2	48.73
	3	48.6
	4	48.76
NO_3	2	76.18
	3	79
	4	78.71
PO_4	2	64.17
	3	64.23
	4	61.94

3.2 Regression and Variable Importance

Regression models were made for NH_4, NO_3 and PO_4 values using data imputed with *k*=3. After fitting the model with "importance=TRUE", one can subtract the key variables for each model. The following subchapters will go through this procedure for the chosen measurements.

3.2.1 NH_4

For NH_4, the initial model using random forest had an RMSE of 1.8449 and accuracy of 48.6%. The variable importance of all the measurements were subtracted (Figure 3.1)

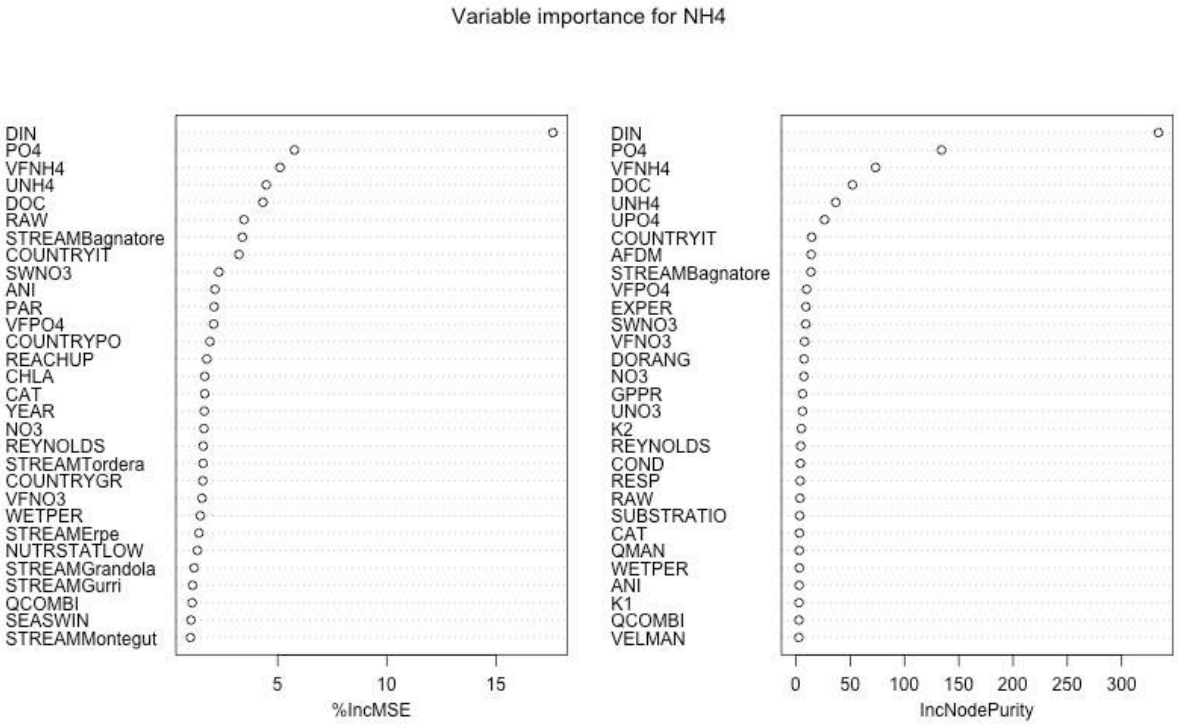

Figure 3.1 Variable importance for NH_4; The left graphs represent the rate of the value in mean decrease in MSE, right graph shows the mean decrease in node impurity

For the next step, the mean of the mean decrease in MSE of each variable was calculated, which in the case of NH_4 equals to 0.9727. Then, all the variables with the value higher than that were subtracted for further analysis.

The following step was to fit a model using these values only. The accuracy of the model was found to be 54.49%, comparing to the initial 48.6% and RMSE to be 1.7888, comparing to the initial 1.8449.

The Table 3.2 contains the same values (before and after variable importance analysis) for all 3 models: RF, SVM and ANN.

Table 3.2 Results of the regression training before and after variable importance analysis for all the algorithms in RMSE for NH_4

Model	RMSE of the model before Variable Importance analysis (relative)	RMSE of the model after Variable Importance analysis (relative)
RF	1.8449	1.7888
ANN	2.9297	2.8804
SVM	5.1370	4.7132

In addition to that, *devtools* package in R provides the possibility to visualize neural network models, showing its scheme and weights on each step. On Figure 3.2 one can see an example of such model for the dataset created after variable importance analysis.

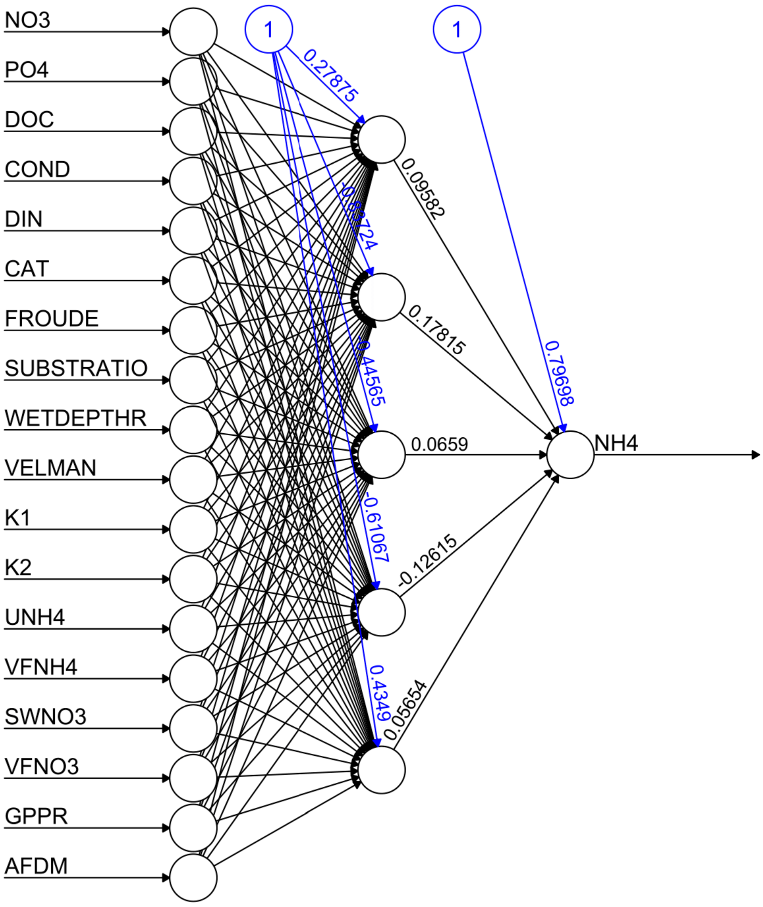

Figure 3.2 Scheme of the neural network algorithm fitted for NH_4 data, generated by *devtools* package

3.2.2 NO_3

The same was done for the NO_3. The initial model has accuracy of 79% and RMSE of 2.3704. The variable importance is shown on the Figure 3.3.

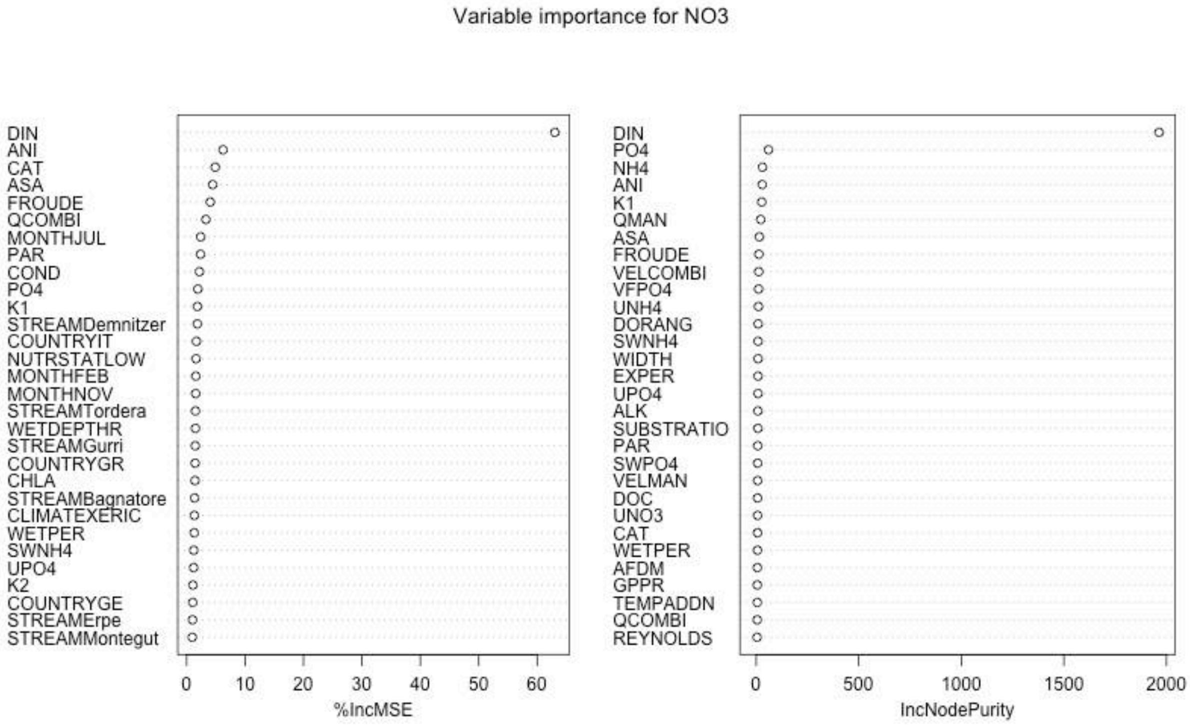

Figure 3.3 Variable importance for NO_3; The left graphs represent the rate of the value in mean decrease in MSE, right graph shows the mean decrease in node impurity

The mean decrease in MSE for NO_3 was found to be 1.2455. After the measurements with the decrease in accuracy higher than mean value were subtracted, new model was fitted. The accuracy of the new model is 83.6%, comparing to 79% of the initial model and new RMSE 1.9533, compared to initial 2.3704.

Table 3.3 contains RMSE of all the models before and after variable importance analysis.

Table 3.3 Results of the regression training before and after variable importance analysis for all the algorithms in RMSE for NO_3

Model	RMSE of the model before Variable Importance analysis (relative)	RMSE of the model after Variable Importance analysis (relative)
RF	2.3704	1.9533
ANN	4.7612	4.7387
SVM	7.6743	5.9287

Similarly to NH_4 results, scheme of neural network for dataset generated after variable importance is shown on Figure 3.4.

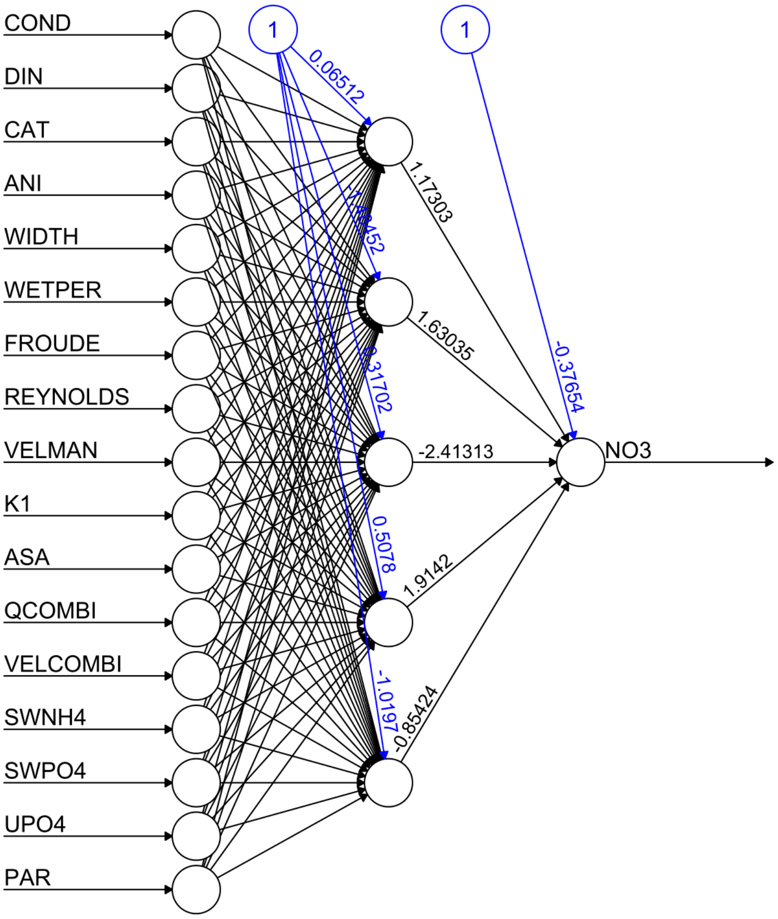

Figure 3.4 Scheme of the neural network algorithm fitted for NO_3 data, generated by *devtools* package

3.2.3 PO_4

Finally, for the PO_4 the initial accuracy of the model was sound to be 64.23% and RMSE to be 0.1678. The variable importance is shown on Figure 3.5.

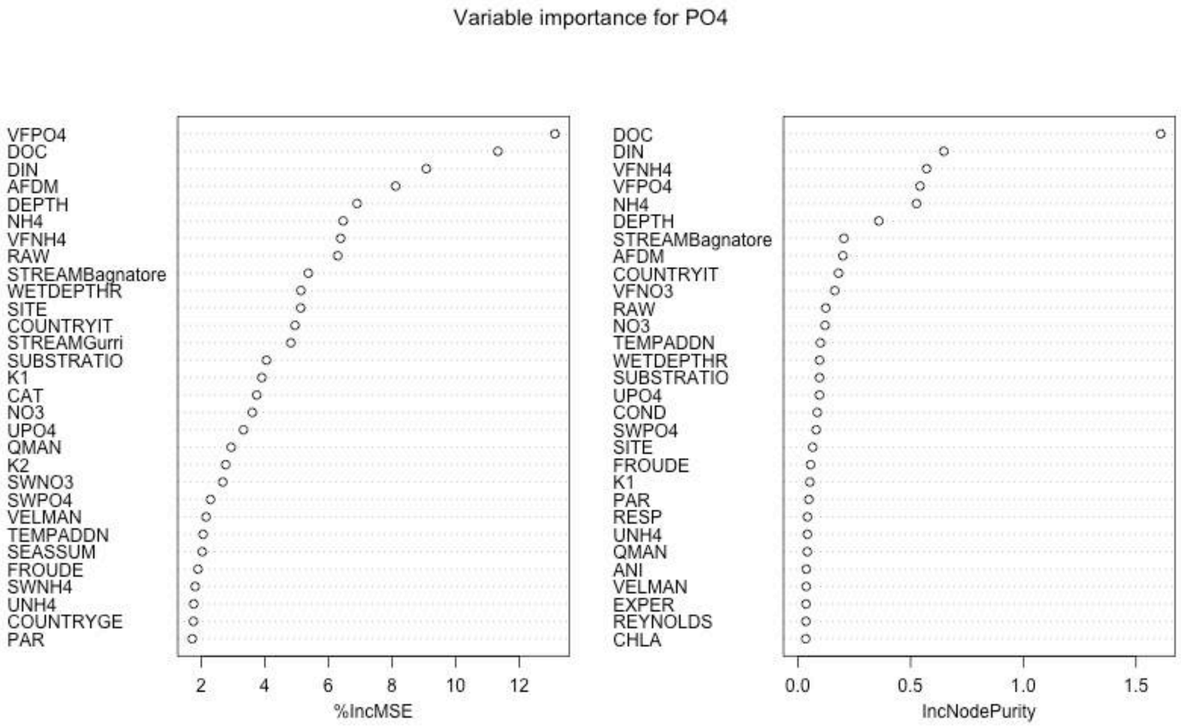

Figure 3.5 Variable importance for PO_4; The left graphs represent the rate of the value in mean decrease in MSE, right graph shows the mean decrease in node impurity

The mean decrease in MSE for PO_4 was found to be 2.0643. The model based on the variables with the higher rate of the decrease has shown the accuracy of 64.31% and RMSE of 0.1786.

Table 3.4 contains RMSE of all the models before and after variable importance analysis.

Table 3.4 Results of the regression training before and after variable importance analysis for all the algorithms in RMSE for PO_4

Model	**RMSE of the model before Variable Importance analysis (relative)**	**RMSE of the model after Variable Importance analysis (relative)**
RF	1.8449	1.7888
ANN	0.2767	0.1724
SVM	0.0380	0.0412

Finally, on Figure 3.6 one can see the scheme of the ANN generated for PO_4 data.

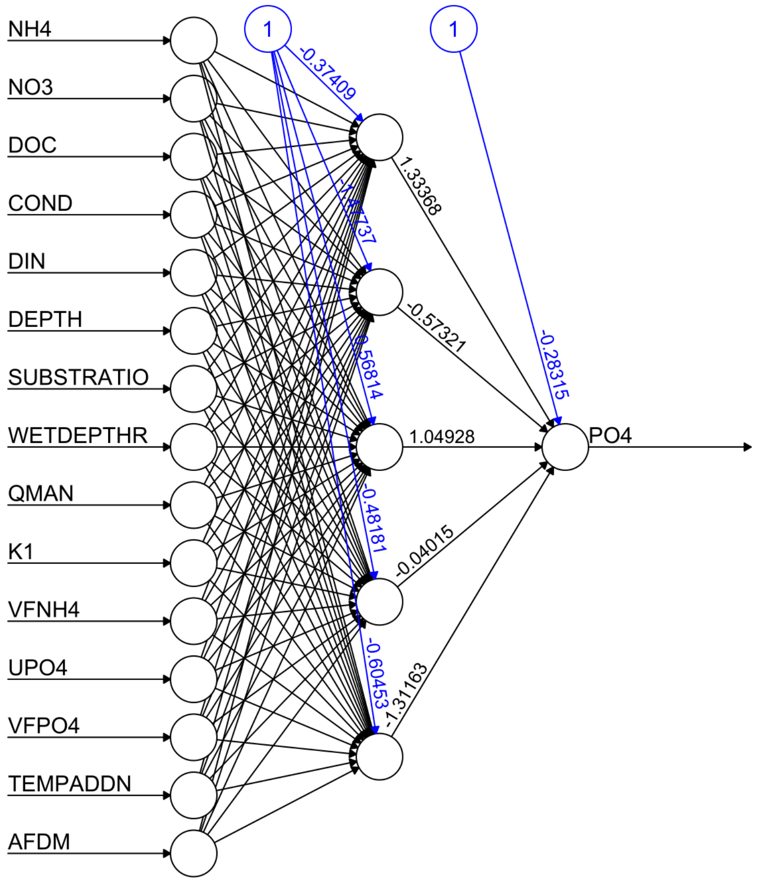

Figure 3.6 Scheme of the neural network algorithm fitted for PO_4 data, generated by *devtools* package

3.2.4 Overfitting

In the Table 3.5 one can find the accuracy of the model for PO_4 without data imputation and after data imputation. Also, for these models the variable importance was done and the corresponding accuracy is shown.

Table 3.5 Accuracies of the models for PO_4, before and after variable importance analysis

Data set of PO_4	**Accuracy of the model before Variable Importance analysis (in %)**	**Accuracy of the model after Variable Importance analysis (in %)**
Non-imputed	66.84	35.92
Imputed	64.23	64.31

3.3 Classification

The next step was to apply classification models to the used data. Two models were fit using Random Forest algorithm for *season* and *land use* measurements. The confusion matrices are presented in Table 3.6 and Table 3.7 respectively.

Table 3.6 Confusion matrix for season classification

	Autumn	**Spring**	**Summer**	**Winter**	**Class error**
Autumn	16	0	0	0	**0**
Spring	0	34	0	6	**0.15**
Summer	0	0	27	0	**0**
Winter	0	3	0	24	**0.11**

Table 3.7 Confusion matrix for land use classification

	Agricultural	**Forested**	**Class error**
Agricultural	57	0	**0**
Forested	0	53	**0**

The accuracy of the models is calculated to be 91.82% for season classification and 100% for land use classification

3.4 Cluster Analysis

Finally, clusterization based on internal patterns of the data was done using k-means clustering with *k*=3. The initial purpose was to define 3 typical types of overall water quality, which may be called i.e. *poor*, *good*, and *normal*. The model was made based on all the available data. In the following figures some details regarding NH_4, NO_3 and PO_4 are shown. On Figure 3.7 one can see the results of the clustering and overall sizes of the classes for these measurements. Similarly, on Figure 3.8 the decision tree is shown, which can be used for distributing the samples to one of three new classes.

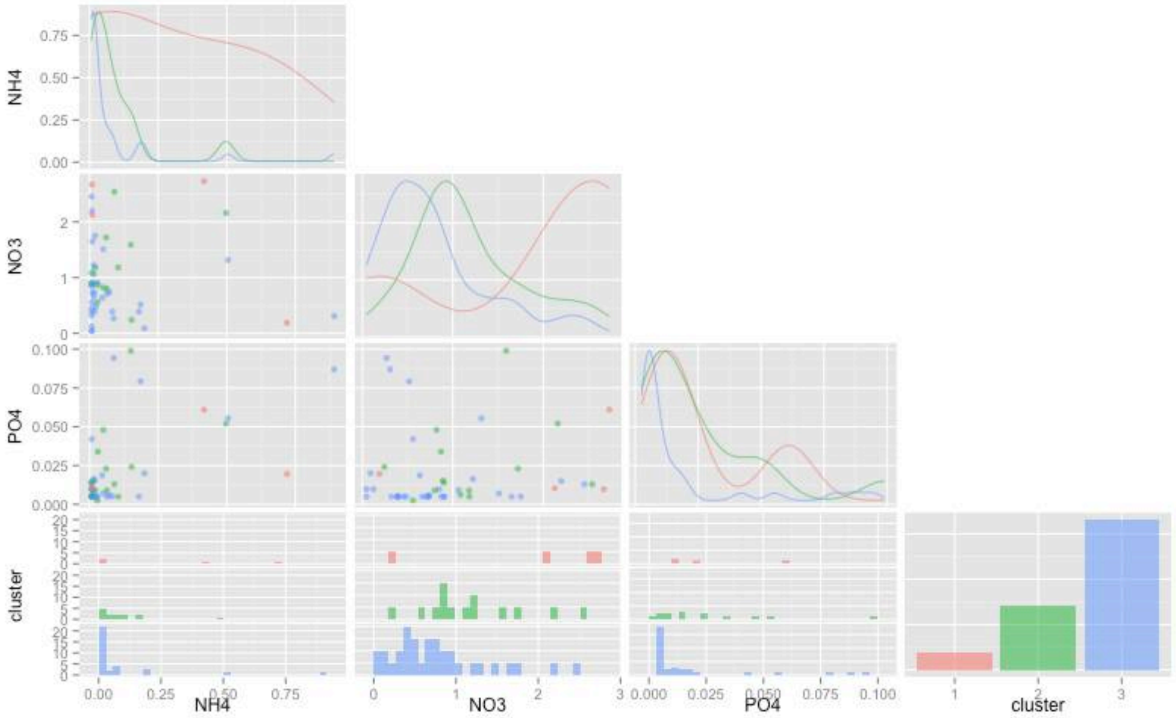

Figure 3.7 Results of k-means clustering of the data for NH_4, NO_3 and PO_4 measurements

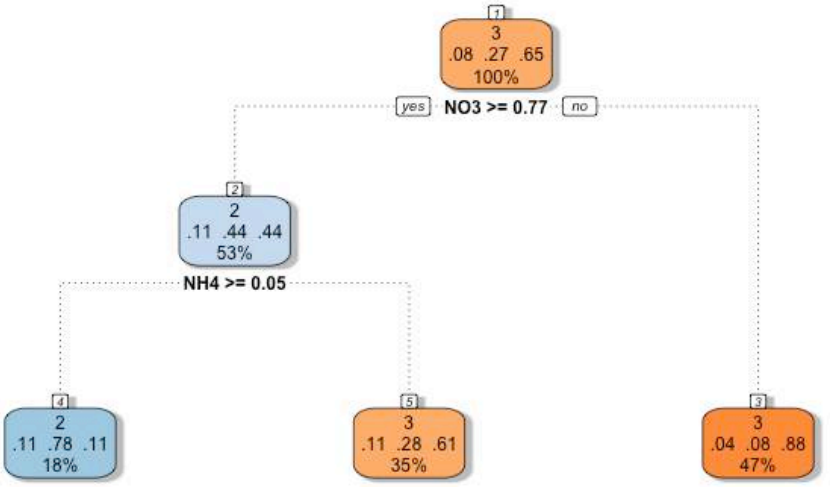

Figure 3.8 Decision tree for distribution of samples to three classes for NH_4, NO_3 and PO_4 measurements

4 DISCUSSION

Following the goals set for this research, all the common types of data reparation and models' fitting was done. Firstly, after the NA observations were specified, data imputation was performed using KNN algorithm with 2, 3 and 4 neighbours (Table 3.1); since imputation with 3 neighbours gave better results, it was used for further analysis. The results presented in Table 3.1 also yield several important messages. Firstly, data gives quite different accuracies for different compounds (around 48%, 76% and 64% for NH_4, NO_3 and PO_4 respectively). Generally, accuracy of >70% could be considered suitable for prediction tasks in fields like environmental sciences, since origins of the processes are extremely diverse and complicated, while for analytical chemistry it would be completely unacceptable (Domingos 2012). We can evaluate that the most probably the data set won't give suitable results for NH_4; however, they were still checked in order to give more accurate representation of model's performance. The most possible reason for such poor accuracy is the size of the data set and relatively huge amount of NA values for some of the measurements. Fitting a regression model usually requires huge amount of cases to be analysed, while classification models may give better performance on such data sets (examples will be explained below). The same was noticed for this data set by other research groups working with STREAMES data (Vellido, et al. 2007). This is the reason why it is always important to consult data-analysts before planning environmental monitoring activities.

Next, regression models for NH_3, NO_3 and PO_4 were fitted. As it is seen in Table 3.2, Table 3.3 and Table 3.4, random forests tend to give consistently better results (except for PO_4 models, where the problem of overfitting is likely to occur). Taking in accounts that due to it's origins, RF model was less prone to overfitting, it can be considered the best suit for this kind of data sets based on this particular study and is hereby recommended for scientists and engineers planning to work with similar environmental data. As was expected, SVM gave the poorest performance consistently and ANN shows slightly lower performance, than RF.

Variable importance analysis meanwhile tends to increase the accuracies of the models; as it is also reducing computational costs, it is highly recommended especially for large data sets with huge variety of variables presented. Also, analysing variable importance helps to find the relations between compounds and reactions. For example, amount of NO_3 is strongly related to amount of amount of dissolved organic nitrogen, while accuracy of the model (83,6%) is relatively suitable, which makes it possible to simplify the procedure of evaluating the quality of the water by calculating only one measurement. The same is true for NH_4 and is quite obvious from the origins of the measurements (Van Kessel, Clough and Van Groenigen 2009). However, it gives a clear example of how these calculations may be used for searching for the patterns and relations of one measurement to another.

Good examples of a clear overfitting problem are shown in Table 3.5. While the model based on non-imputed data was able to show better performance, after variable importance analysis, its accuracy reduced significantly, while the model based on imputed data has shown better performance. The reason for that, as was explained in Overfitting chapter, is that model was able to fit pretty well to the training data, but since it starts explaining random errors and noise, as there is not enough data presented; after variable importance analysis this issue reveals and performance is significantly lowered.

While regression models show average performance for specified chemical compounds, this data set tends to give quite good patterns for defining land use and season. It is clear that chemical composition of water in agricultural areas is strongly different to water composition in forested areas due to the serious effects caused by cultivation process. For example, agricultural lands usually lack organic layer (O) of the soil, while mineral surface (A) is often mixed with subsurface mineral horizon with features of accumulation (B), while B and C (regolith layer) may be themselves broken up by ripping of deep cultivation (see Figure 4.1). It causes erosion of the land, alteration of soil microorganisms and loss of organic material; in addition to that, such scenario changes the proportions of rainfall absorbed by the soil. All together, these factors cause significant changes to groundwater composition, which itself leads to noticeable changes in the water compositions of the rivers (Brooks 2005). Thus, classification models are able to find these patterns, so it can perfectly define the case.

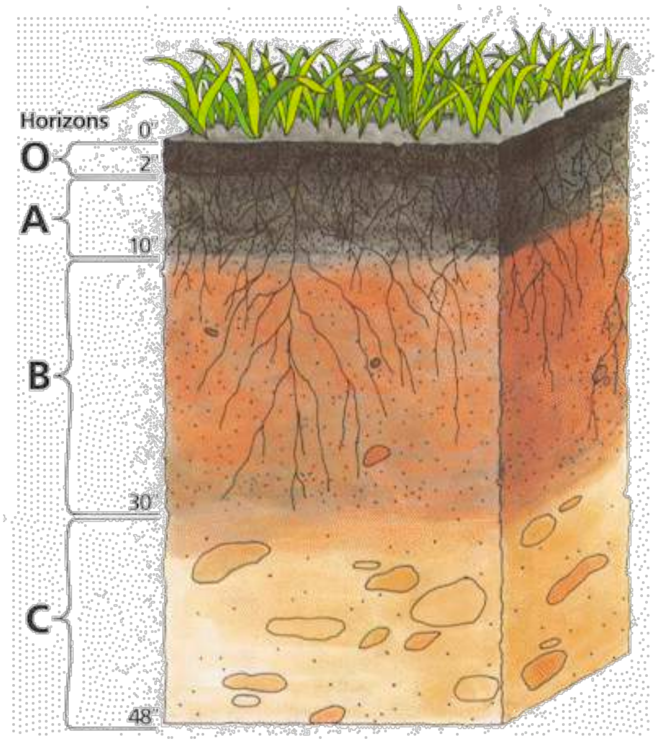

Figure 4.1 Soil profile; *O* - organic layer; *A* – topsoil, mineral surface layer; *B* – layer of illuviation, a mineral horizon penetrable by roots; *C* – regolith layer, containing rocks and little organic material (Krech 2010)

The same goes for the classification of the seasons. Interesting is the misclassification behaviour in case of seasons data. *Winter* results tend to be confused with *spring* results and vice versa. The possible reason for that is that waters take time to stabilize after winter season, especially when weather is unstable or cold spring occurs in the region. Also, as stated in microbiology works, some microorganisms have their bloom peaks and overall activities peaks in winter-spring seasons, which could also affect water composition (Chenier, et al. 2003). Overall, these models already give possibility to predict the season of the water sample and land use of the area where the sample was taken with noticeable accuracy of 90-100%.

Finally, examples of how this data could be classified in 3 major classes based on water composition are presented using clusterizaion algorithm in Cluster Analysis chapter. Figure 3.7 shows results of this process for NH_4, NO_3 and PO_4 measurements. Diagonal plots are of special interest, as they show the distribution of samples included in different classes. Season and land use values were excluded before fitting the model in order to exclude the possibility of classification based on these patterns. The huge difference in the sized of the classes, shown on cluster-cluster bar chart (lower right) on Figure 3.7 suggests that the classes could be defined as indicating *good* (the biggest class, class 3), *average* (class 2) and *poor* (class 1) quality of water. The same can be clearly seen on NO_3-NO_3 plot (middle). While class 3 contains samples with the lowest amount of NO_3, class 2 contains samples where this value is higher and class 1 mostly contains samples with enormously high concentration of NO_3. For the water quality, concentration of nitrates in river water of <1 mg/L are considered natural (Behar 1996). As it is also seen on Figure 3.7, classes 2 and 3 are below 1 mg/L NO_3 and class 1 has samples with NO_3 >2 mg/L. Figure 3.8 with decision tree for defining sample into class 1, class 2 and class 3 based on NH_4, NO_3 and PO_4 measurements shows the algorithm suggested by k-means clustering model for the given data set. Firstly, samples are separated by concentrations of NO_3 by extracting once where it is higher than 0.77 mg/L, which correlates with a conventional standard of water quality. The next decision criteria is higher amount of NH_4 and finally samples which do not fall under these criteria go to the last class. Since building the decision tree was based on NH_4, NO_3 and PO_4 values *alone*, the sizes of the classes do not correlate to the sizes of the classes shown on Figure 3.7. It is important to notify that this figure is meant to show the patterns of NH_4, NO_3 and PO_4 values found by the model *alone* and the distribution may be different in that case, as other values are not taken into consideration; thus it is not to be confused with the real distribution based on *all* the patterns in the data shown on Figure 3.7. So, all in all clustering techniques give quite good results on providing a researcher with possibly ways to classify given data set and may assist process of analysing the water quality.

5 CONCLUSION

Overall, the goals defined for this research were reached and the examples of the application of machine learning models are presented, covering most of the aspects of the average research working in the field of artificial intelligence for environmental sciences tasks. This work also reveals the importance of consulting data scientists before starting of the monitoring, since data sets unsuitable for requested tasks is a common problem.

Generally, regression models were able to show the consistent trend and overall correlation between each other, even though for some of the measurements they give models of poor quality. Random forests (RF) show the best performance and are advised for scientists and engineers working with environmental data. Artificial neural networks (ANN) are another alternative, though their performance is inferior and they are prone to overfitting. Support vector machines (SVM) are the good example for the cases where a baseline model is needed, being one of the basic algorithms.

K-nearest neighbours (KNN) model was successfully used for data imputation and is also suggested for this task for other researchers. Though, and it is worth noticing, amount of neighbours used for this research (3) is not universal and another amount may be found suitable for different data sets.

Classification models show good performance and are able to make highly accurate prediction models for identifying season of the sample and land use of the area where it was taken. Meanwhile, clusterization techniques, such as k-means clustering, may assist data scientist with possible algorithms to classify given data, for example defining good, average and bad conditions of the water based on various chemical, biological and physical parameters.

Future prospective of the development of this research may be seen in several ways. Firstly, consistent misclassification of season values between winter and spring may be studied further using this data set by extracting and analysing the samples, which tend to be often misclassified. On the other hand, models generated during this research may be used by IT students for producing software meant to help environmental specialists in analysing collected water quality data.

All in all, following the technological progress and taking the best from what it provides us from day to day ensures continuous development of the research field. The same goes for environmental sciences and machine-learning algorithms are one of the tools that can contribute to this field a lot and may be used to keep the progress on-going.

6 BIBLIOGRAPHY

Batista, G. E. A. P. A., and M. C. Monard. “A Study of K-Nearest Neighbour as an Imputation Method .” *Soft Computing Systems: Design, Management and Applications* , 2002: pp. 251–260 .

Behar, Sharon. *Testing the Waters: Chemical and Physical Vital Signs of a River.* River Watch Network, 1996.

Biau, Gerard. “Analysis of a Random Forests Model.” *The Journal of Machine Learning Research* 13, no. 1 (2012): 1063-1095.

Blackwell, Matthew, James Honaker, and Gary King. “A Unified Approach to Measurement Error and Missing Data: Overview and Applications.” *Sociological Methods & Research*, 2015: 1-39.

Breiman, Leo. “Random Forests.” *Machine Learning* 45, no. 1 (2001): 5–32.

Brooks, Randy. *Forest Soils versus Agricultural Soils.* UI Extension Forestry Information Series, University of Idaho, University of Idaho, 2005.

Browne, Michael W. “Cross-Validation Methods.” *Journal of Mathematical Psychology* 44, no. 1 (2000): pp 108–132.

Chapman, Pete, et al. *CRISP-DM 1.0* . Step-by-step data mining guide, SPSS, 2000.

Cheng, Bing, and D. M. Titterington. “Neural Networks: A Review from a Statistical Perspective.” *Statistical Science* 9, no. 1 (1994): 2-30.

Chenier, Martin R., Danielle Beaumier, Real Roy, Brian T. Driscoll, John R. Lawrence, and Charles W. Greer. “Impact of Seasonal Variations and Nutrient Inputs on Nitrogen Cycling and Degradation of Hexadecane by Replicated River Biofilms.” *Applied and Environmental Microbiology* 69, no. 9 (2003): pp 5170-5177.

Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” *Machine Learning* 20, no. 3 (1995): pp 273-297.

Domingos, Pedro. “A few useful things to know about machine learning.” *Communications of the ACM* 55, no. 10 (2012): pp 78-87.

Frew, James E., and Jeff Dozier. “Environmental Informatics .” *Annual Review of Environment and Resources* 37 (2012): pp. 449-472.

Hagan, Martin T., Howard B. Demuth, Mark Hudson Beale, and Orlando De Jesus. *Neural Network Design.* University of Colorado, 2002.

Hamilton, Howard. “Confusion Matrix.” *Computer Science 831: Knowledge Discovery in Databases.* 8 June 2012.

http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html (accessed December 27, 2015).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New-York: Springer, 2009.

Ho, Tin Han. "Random decision forests." *Proceedings of the Third International Conference on Document Analysis and Recognition* 1 (1995): 278 - 282.

Jensen, Thomas. "Cross Validation." *link.ch.* 10 2013. http://www.link.ch/wp-content/uploads/2013/10/CrossValidationFinal.pdf (accessed 01 15, 2016).

Kendall, Bruce, and Chris Costello. *Data Analysis for Environmental Science and Management.* Santa Barbara: University of California, 2006.

Krech, M. J. "Soil Horizons." *slideshare.net.* 06 December 2010. http://www.slideshare.net/Teach5ch/soil-horizons (accessed January 24, 2016).

Krstajic, Damjan, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. "Cross-validation pitfalls when selecting and assessing regression and classification models." *Journal of Cheminformatics* 6, no. 10 (2014).

Kuhn, Max. "Package 'caret'." *cran.r-project.org.* 23 November 2015. https://cran.r-project.org/web/packages/caret/caret.pdf (accessed 12 25, 2015).

Lavrenko, Victor, and Nigel Goddard. "Introductory Applied Machine Learning." *inf.ed.ac.uk.* 2015. http://www.inf.ed.ac.uk/teaching/courses/iaml/slides/eval-2x2.pdf (accessed December 27, 2015).

Lebanon, Guy. *Bias, Variance, and MSE of Estimators.* Technical Notes, Georgia: Georgia Institute of Technology, 2010.

Li, Ray. *History of data mining.* 26 April 2015. http://rayli.net/blog/data/history-of-data-mining/ (accessed December 25, 2015).

Likas, Aristidis, Nikos Vlassis, and Jakob J. Verbeek. "The global k-means clustering algorithm." *Pattern Recognition* (Elsevier) 36, no. 2 (February 2003): 451–461.

Lu, Wei, Yanyan Shen, Su Chen, and Beng Chin Ooi. "Efficient Processing of k Nearest Neighbor Joins using MapReduce." *Proceedings of the VLDB Endowment* 5, no. 10 (2012): 1016-1027.

Mitchell, T. *Machine Learning.* McGraw Hill, 1998.

Refaeilzadeh, Payam, Lei Tang, and Huan Liu. "Cross-Validation." *Encyclopedia of Database Systems*, 2009: pp 532-538.

Shkurin, Aleksei, and Alfredo Vellido. "Random Forests for quality control in G-Protein Coupled Receptor databases ." 2016.

Simon, Phil. *Too Big to Ignore: The Business Case for Big Data.* New Jersey: Wiley, 2013.

Starkweather, Jon. *Cross Validation techniques in R: A brief overview of some methods, packages, and functions for assessing prediction models.* . Review, University of North Texas, 2011.

Teknomo, Kardi. "Kardi Teknomo – K Mean Clustering Tutorial ." *croce.ggf.br.* July 2007. http://croce.ggf.br/dados/K%20mean%20Clustering1.pdf (accessed December 27, 2015).

Van Kessel, Chris, Tim Clough, and Jan Willem Van Groenigen. "Dissolved Organic Nitrogen: An Overlooked Pathway of Nitrogen Loss from Agricultural Systems?" *Journal of Environmental Quality* 38, no. 2 (2009): pp 393-401.

Vellido, A., E. Marti, J. Comas, I. Rodriguez-Roda, and F. Sabater. "Exploring the ecological status of human altered streams through Generative Topographic Mapping ." *Environmental Modelling & Software* (Elsevier), no. 22 (2007): 1053e1065.

Wagstaff, Kiri, Claire Cardie, Seth Rogers, and Stefan Schrödl. "Constrained K-means Clustering with Background Knowledge." *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001: 577-584.

Warren, McCulloch, and Walter Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5, no. 4 (1943): pp 115–133.

Weston, Jason. "Support Vector Machine (and Statistical Learning Theory)." *cs.columbia.edu.* 2004. http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf (accessed January 15, 2016).

Wilcox, Catherine, Wei Lee Woon, and Zeyar Aung. *Applications of Machine Learning in Environmental Engineering.* Technical Report, Abu Dhabi: Masdar Institute of Science and Technology, 2013.

R code

```r
#Welcome
#Firstly, necessary libraries
library(caret)
library(DMwR)
library(randomForest)
library(GGally)
library(RGtk2)
library(rattle)
library(rpart)
library(rpart.plot)
library(e1071)
library(MASS)
library(devtools)

#set seed to ensure reproducibility and set working directory
set.seed(42)
setwd("/Users/Aleksei/Thesis")

#Load original data
stream<-read.csv("streames.csv")

##Starting data imputation
#firstly, set all the missing values to NA

stream[stream=="0"]<-NA
stream[stream=="-1"]<-NA

#For imputing, use DMwR package
```

```r
#KNN imputation performed with 2, 3 and 4 neighbours
prep2stream<-knnImputation(stream,k=2)
prep3stream<-knnImputation(stream,k=3)
prep4stream<-knnImputation(stream,k=4)

#Preparing data for unsupervised learning for NH4, NO3 and PO4
plot(prep3stream$NH4)
prep3streamUN<-subset(prep3stream,!prep3stream$NH4>sum(mean(prep3stream$NH4)))
prep3streamUN<-subset(prep3streamUN,!prep3streamUN$PO4>sum(mean(prep3streamUN$PO4)))
prep3streamUN<-subset(prep3streamUN,!prep3streamUN$NO3>sum(mean(prep3streamUN$NO3)))

#For the examples of overfitting, make a prediction model for NH4
#based on non-imputed data using RF
crStreamNH4<-train(NH4~.,data=stream,method="rf",importance=TRUE)

#Make a prediction model for NH4, NO3 and PO4 with prep3stream
```

```
#(best results so far, prep2 and prep4 were also checked)
#For NH4
crPrep3StreamNH4<-train(NH4~.,data=prep3stream,method="rf",importance=TRUE)
getTrainPerf(crPrep3StreamNH4)

#For NO3
crPrep3StreamNO3<-train(NO3~.,data=prep3stream,method="rf",importance=TRUE)
getTrainPerf(crPrep3StreamNO3)

#For PO4
crPrep3StreamPO4<-train(PO4~.,data=prep3stream,method="rf",importance=TRUE)
getTrainPerf(crPrep3StreamPO4)

#Analyze Varible Importance for NH4 initial model based on non-imputed data
#VI is extracted and new data set is created based on results of VI
VarImpNH4<-as.matrix(varImp(crStreamNH4$finalModel))
mainVarImpNH4<-subset(VarImpNH4,VarImpNH4>mean(as.matrix(varImp(crStreamNH4$finalModel))))
```

```
mainStreamNH4<-stream[,(names(stream) %in% attributes(mainVarImpNH4)$dimnames[[1]])]
newStreamNH4<-data.frame(mainStreamNH4,NH4=stream$NH4)

#Train new model based on VarImp data for NH4
crVIStreamNH4<-train(NH4~.,data=newStreamNH4,method="rf",importance=TRUE)

#Analyze VarImp for NH4 model based on prep3stream
VarImpPrep3NH4<-as.matrix(varImp(crPrep3StreamNH4$finalModel))
mainVarImpPrep3NH4<-subset(VarImpPrep3NH4,VarImpPrep3NH4>mean(as.matrix(varImp(crPrep3StreamNH4$finalModel))))
mainPrep3StreamNH4<-prep3stream[,(names(prep3stream) %in% attributes(mainVarImpPrep3NH4)$dimnames[[1]])]
newPrep3StreamNH4<-data.frame(mainPrep3StreamNH4,NH4=prep3stream$NH4)

#Train new model
crVIPrep3StreamNH4<-train(NH4~.,data=newPrep3StreamNH4,method="rf",importance=TRUE)
getTrainPerf(crVIPrep3StreamNH4)

#Same for NO3
```

```r
VarImpPrep3NO3<-as.matrix(varImp(crPrep3StreamNO3$finalModel))
mainVarImpPrep3NO3<-subset(VarImpPrep3NO3,VarImpPrep3NO3>mean(as.matrix(varImp(crPrep3StreamNO3$finalModel))))
mainPrep3StreamNO3<-prep3stream[,(names(prep3stream) %in% attributes(mainVarImpPrep3NO3)$dimnames[[1]])]
newPrep3StreamNO3<-data.frame(mainPrep3StreamNO3,NO3=prep3stream$NO3)

crVIPrep3StreamNO3<-train(NO3~.,data=newPrep3StreamNO3,method="rf",importance=TRUE)
getTrainPerf(crVIPrep3StreamNO3)

#Same for PO4
VarImpPrep3PO4<-as.matrix(varImp(crPrep3StreamPO4$finalModel))
mainVarImpPrep3PO4<-subset(VarImpPrep3PO4,VarImpPrep3PO4>mean(as.matrix(varImp(crPrep3StreamPO4$finalModel))))
mainPrep3StreamPO4<-prep3stream[,(names(prep3stream) %in% attributes(mainVarImpPrep3PO4)$dimnames[[1]])]
newPrep3StreamPO4<-data.frame(mainPrep3StreamPO4,PO4=prep3stream$PO4)

```

```
crVIPrep3StreamPO4<-train(PO4~.,data=newPrep3StreamPO4,method="rf",importance=TRUE)
getTrainPerf(crVIPrep3StreamPO4)

##Classification tasks
#Fitting classification model based on RF algorithm
crPrep3StreamSEA<-train(SEAS~.,data=prep3stream,method="rf",importance=TRUE)
crPrep3StreamLAND<-train(LANDUSE~.,data=prep3stream,method="rf",importance=TRUE)

#Unsupervised classification using k-means clustering with 3 clusters
#Excluding 1 sample with obviously wrong measurements
prep3streamkm<-prep3streamUN[-77,]
drop<-c("RAW","YEAR","MONTH","SEAS","CLIMATE","COUNTRY","SITE","EXPER","STREAM","REACH","NUTRSTAT","LANDUSE")
kmPrep3Stream<-prep3streamkm[,!(names(prep3streamkm) %in% drop)]
kmFitPrep3StreamCen3<-kmeans(kmPrep3Stream,centers = 3)

##Now, time to plot
plot(prep3streamkm$RAW, prep3streamkm$NH4, col=kmFitPrep3Stream$cluster)
```

```

#Save plot to working directory
png(filename = "VarImpInitNH4.png",width = 800, height = 500)
varImpPlot(crPrep3StreamNH4$finalModel,main="Variable importance for NH4")
dev.off()

##Plotting the decision tree for the results of k-means clusterization
#Firstly, prepare data using NH4, NO3 and PO4 samples
chose<-c("NH4","NO3","PO4")
kmFitPrep3StreamCen3$cluster->clu
kmPlotPrep3Stream<-prep3streamkm[,(names(prep3streamkm) %in% chose)]
 kmPlotPrep3Stream$cluster<-factor(clu)
 ggpairs(kmPlotPrep3Stream, colour='cluster', alpha=0.4)

 #Decision tree for the results of clusterization
 RpartPrep3 = rpart(cluster~.,data = kmPlotPrep3Stream)
 fancyRpartPlot(RpartPrep3)

 ##Fitting SVM using package e1071 based on the same data, as with RF
 #With 10-folds cross-validation
 #Firstly, models based on imputed data (initial models in report)
```

```
#For NH4
svmPrep3StreamNH4<-svm(NH4~.,prep3stream,cross=10)
summary(svmPrep3StreamNH4)

#For NO3
svmPrep3StreamNO3<-svm(NO3~.,prep3stream,cross=10)
summary(svmPrep3StreamNO3)

#For PO4
svmPrep3StreamPO4<-svm(PO4~.,prep3stream,cross=10)
summary(svmPrep3StreamPO4)

#Now, fitting models based on the data of VarImp procedure
#For NH4
svmVIPrep3StreamNH4<-svm(NH4~.,newPrep3StreamNH4,cross=10)
summary(svmVIPrep3StreamNH4)

#For NO3
svmVIPrep3StreamNO3<-svm(NO3~.,newPrep3StreamNO3,cross=10)
summary(svmVIPrep3StreamNO3)

#For PO4
svmVIPrep3StreamPO4<-svm(PO4~.,newPrep3StreamPO4,cross=10)
summary(svmVIPrep3StreamPO4)

```

```
##Finally, ANN
#Data preparation, setting conditions for 10-folds cross-validation and
#Training and test sets to calculate RMSE
control <- trainControl(method="repeatedcv", number=10, repeats=3)
inTrainNH4         = createDataPartition(prep3stream$NH4, p = 0.8)[[1]]
trainingNH4         <- prep3stream[inTrainNH4,]
testingNH4         <- prep3stream[-inTrainNH4,]

inTrainNO3         = createDataPartition(prep3stream$NO3, p = 0.8)[[1]]
trainingNO3         <- prep3stream[inTrainNO3,]
testingNO3         <- prep3stream[-inTrainNO3,]

inTrainPO4         = createDataPartition(prep3stream$PO4, p = 0.8)[[1]]
trainingPO4         <- prep3stream[inTrainPO4,]
testingPO4         <- prep3stream[-inTrainPO4,]

#Now, models based on imputed data and calculating RMSE
#For NH4
nnPrep3StreamNH4<-train(NH4~.,data=trainingNH4,method="nnet",trControl=control)
predNH4<-predict(nnPrep3StreamNH4,newdata=testingNH4)
NH4rmse<-sqrt(mean((predNH4 - testingNH4$NH4)^2))
```

```

#For NO3
nnPrep3StreamNO3<-train(NO3~.,data=trainingNO3,method="nnet",trControl=control)
predNO3<-predict(nnPrep3StreamNO3,newdata=testingNO3)
NO3rmse<-sqrt(mean((predNO3 - testingNO3$NO3)^2))

#For PO4
nnPrep3StreamPO4<-train(PO4~.,data=trainingPO4,method="nnet",trControl=control)
predPO4<-predict(nnPrep3StreamPO4,newdata=testingPO4)
PO4rmse<-sqrt(mean((predPO4 - testingPO4$PO4)^2))

#Now, models based on the data generated after VarImp and calculating RMSE
#For NH4
newinTrainNH4 = createDataPartition(newPrep3StreamNH4$NH4, p = 0.8)[[1]]
newtrainingNH4 <- newPrep3StreamNH4[newinTrainNH4,]
newtestingNH4 <- newPrep3StreamNH4[-newinTrainNH4,]

nnVIPrep3StreamNH4<-train(NH4~.,data=newtrainingNH4,method="nnet",trControl=control)
```

```
newpredNH4<-predict(nnVIPrep3StreamNH4,newdata=newtestingNH4)
NH4VIrmse<-sqrt(mean((newpredNH4 - newtestingNH4$NH4)^2))

#For NO3
newinTrainNO3         = createDataPartition(newPrep3StreamNO3$NO3, p = 0.8)[[1]]
newtrainingNO3        <- newPrep3StreamNO3[newinTrainNO3,]
newtestingNO3        <- newPrep3StreamNO3[-newinTrainNO3,]

nnVIPrep3StreamNO3<-train(NO3~.,data=newtrainingNO3,method="nnet",trControl=control)
newpredNO3<-predict(nnVIPrep3StreamNO3,newdata=newtestingNO3)
NO3VIrmse<-sqrt(mean((newpredNO3 - newtestingNO3$NO3)^2))

#For PO4
newinTrainPO4         = createDataPartition(newPrep3StreamPO4$PO4, p = 0.8)[[1]]
newtrainingPO4        <- newPrep3StreamPO4[newinTrainPO4,]
newtestingPO4        <- newPrep3StreamPO4[-newinTrainPO4,]

```

```
nnVIPrep3StreamPO4<-train(PO4~.,data=newtrainingPO4,method="nnet",trControl=control)
newpredPO4<-predict(nnVIPrep3StreamPO4,newdata=newtestingPO4)
PO4VIrmse<-sqrt(mean((newpredPO4 - newtestingPO4$PO4)^2))

#Visualizing of the ANN models based on VarImp data
#NH4
nnPlotNH4<-neuralnet(NH4~NO3+PO4+DOC+COND+DIN+CAT+FROUDE+SUBSTRATIO+WETDEPTHR+VELMAN+K1+K2+UNH4+VFNH4+SWNO3+VFNO3+GPPR+AFDM,prep3stream, hidden=5,threshold = 0.01)

#NO3
nnPlotNO3<-neuralnet(NO3~COND+DIN+CAT+ANI+WIDTH+WETPER+FROUDE+REYNOLDS+VELMAN+K1+ASA+QCOMBI+VELCOMBI+SWNH4+SWPO4+UPO4+PAR,prep3stream, hidden=5,threshold = 0.01)

#PO4
nnPlotPO4<-neuralnet(PO4~NH4+NO3+DOC+COND+DIN+DEPTH+SUBSTRATIO+WETDEPTHR+QMAN+K1+VFNH4+UPO4+VFPO4+TEMPADDN+AFDM,prep3stream, hidden=5,threshold = 0.01)
```