Avertissements au lecteur

Ce document est le produit d'un projet de master réalisé à l'EPFL dans le laboratoire d'Ecohydrologie sur une période de quatre mois. Son contenu n'engage que son auteur et n'est reproductible qu'avec le consentement écrits de ce dernier.

Warning to the reader

This document is the product of a master's project done at EPFL in the laboratory of Ecohydrology during a four month period. Its content commits solely it author and can be reproduced only with the written consent of the latter.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

SIE Master Thesis	Start: 16.02.2015 Finish: 19.06.2015

Rainfall Forecasting in Burkina Faso Using Bayesian-Wavelet Neural Networks

Jonathan Giezendanner

Laboratory of Ecohydrology ECHO

Professor:	Andrea Rinaldo
Assistant:	Francisco-Javier Perez-Saez

Foreword

The following project comes in the framework of the project " *Understanding schistosomiasis transmission in Burkina Faso*"[1] hosted at the Laboratory of Ecohydrology (ECHO), EPFL, which is under the supervision of Professor Andrea Rinaldo.

The project serves as Master Thesis for the author, and is followed and supervised by PhD Student Francisco-Javier Perez-Saez (ECHO).

The time allocated for the project is of 17 weeks, starting the 16^{th} of February 2015, and ending the 19^{th} of June 2015.

[1] `http://echo.epfl.ch/page-113944-en.html`

Abstract

English

This work aims to forecast rain locally in Tambarga, Burkina Faso, to be able to fight against a worm inducing the disease called schistosomiasis. The chosen approach relies on a machine-leaning technique called Artificial Neural Networks, which simulates the synapses of a brain, with climatic parameters as inputs, activation functions and outputs in the form of rain prediction. A special case of Neural Networks using Bayesian Computations is used, along with as a transform allowing to capture the changes in climatic conditions, called Wavelet Transform. The precipitation is forecasted in different manners: binary forecast on the presence or absence of rain, linear forecast on the daily and weekly intensity, as well as a rain-class forecast. The most successful predictions have been found to be the binary forecast, as well as the weekly windowed cumulative rain forecast. The daily cumulative rain, as well has the classes forecast have not produced satisfying results, mainly because of the high temporal variability of the observations, as well as the very unequal distribution of observations in the different rain classes. In the end, it has been shown that it is possible to use Bayesian Networks to forecast precipitation in some extent, and that the wavelet transform of the inputs has a positive impact on the accuracy of the prediction.

Français

Le but de ce travail est de prédire la pluie localement à Tambarga, Burkina Faso, afin de lutter de façon efficace et ciblée contre un ver induisant une maladie appelée schistosomiase. L'approche choisie repose sur une technique d'apprentissage automatisée appelée Artificial Neural Networks. Celle-ci simule les synapses du cerveau, avec des paramètres climatiques en tant que flux entrant, des fonctions d'activations et des flux sortants sous forme de prédictions de précipitations. Un cas particulier de ces réseaux utilisant le calcul bayésien est utilisé, ainsi qu'une transformée permettant de capturer les changements dans les conditions climatiques, appelée décomposition en ondelettes. La précipitation est prédite de différentes

manières: prévisions binaires sur la présence ou l'absence de pluie, prévisions linéaires de l'intensité quotidienne et hebdomadaire, ainsi que prévisions de classes de pluie. Les prévisions les plus prometteuses ont été réalisée avec la prévision binaire ainsi que sur les prévisions de pluie hebdomadaire cumulée. La prédiction de la pluie cumulée quotidienne et celle par classes n'ont pas produit de résultats satisfaisants, principalement en raison de la grande variabilité temporelle des observations, ainsi que de la répartition très inégale des observations dans les différentes classes de pluie. En fin de compte, il a été montré qu'il est possible d'utiliser des réseaux bayésiens afin de prédire la précipitation dans une certaine mesure, et que la transformée en ondelettes des paramètres a un impact positif sur l'exactitude de la prédiction.

Deutsch

Das Ziel dieser Arbeit ist, Regen in Tambarga, Burkina Faso, lokal zu prognostizieren um effizient und gezielt einen Wurm, welcher Schistosomiasis überträgt, zu bekämpfen. Der gewählte Ansatz beruht auf einer machine-learning Technik namens Artificial Neural Networks welche die Synapsen des Gehirns simuliert, mit klimatischen Parametern als eingehende Daten, Aktivierungsfunktionen und der Regenvorhersage als Output. Es wird ein Spezialfall dieser Netze verwendet, der auf Bayes-Berechnungen basiert, sowie eine Transformation, genannt Wavelet-Transformation, welche die Veränderungen der Klimabedingungen erfasst. Der Niederschlag wird auf unterschiedliche Weise prognostiziert: binäre Vorhersagen über die An- oder Abwesenheit von Regen, lineare Vorhersage über die Tages- und Wochenintensität, sowie eine Prognostik der Regenklassen. Als erfolgreichste Vorhersagen haben sich die binären Vorhersagen sowie die wöchentlichen kumulativen Regenprognosen erwiesen. Der täglich kumulierte Regen sowie die Klassenprognose haben sich als nicht ausreichend erwiesen, vor allem wegen der hohen zeitlichen Variabilität der Beobachtungen, sowie der sehr ungleichen Verteilung der Beobachtungen in den verschiedenen Regenklassen. Schlussendlich hat sich gezeigt, dass es möglich ist, Bayes Netzwerke zu verwenden, um Regen bis zu einem gewissen Grad zu prognostizieren, und dass die Wavelet-Transformation der inputs eine positive Auswirkung auf die Genauigkeit der Vorhersage hat.

Acknowledgements

I would like to thank Professor Andrea Rinaldo for allowing and supporting this project, as well as the entire ECHO team for providing this nice working environment. I would specially like to thank my assistant Francisco-Javier Perez-Saez, without whom this project could not have been made, who always provided excellent feedback and motivational speeches, and for the time he devoted to guidance of the project. I would also like to thank Dr. Natalie Ceperley and Dr. Théophile Mande for their insight, and pertinent comments on the data. Additionally, I would also like to thank my office-mates, Tina Genolet and Fabian Bernhard, for being there to provide a nice mood and funny working environment, as well as support in the darker hours. I would also like to thank the association of the bar Satellite for always providing enough quantities of coffee, and the whole association of Balélec to which I belong for the welcome changes in daily routines. Finally a special thanks to my family and friends who always supported me during these years.

Data

The meteorological data for Tambarga used for this thesis was collected as part of the Info4Dourou project, a collaboration between the "*Ecole Polytechnique Fédérale de Lausanne (CH)*" and the "*Institut International d'Ingénierie de l'Eau et de l'Environnement (BF)*". The project was coordinated by Dr. Natalie Ceperley and Dr. Théophile Mande in the Environmental Fluid Mechanics and Hydrology Laboratory of Marc Parlange and supported by the cooperation unit directed by Jean-Claude Bolay and with the help of the Laboratory of Audiovisual Communications directed by Martin Vetterli and the Sensorscope Sàrl team, as well as all the students, funding agencies and field assistants whose hard work made this possible. Financial support was received from the Foundation for the Third Millennium, Velux Foundation, KFPE, NCCR MICS, and the SDC in addition to the support institutions.

The meteorlogical data in the villages of Tougou, Lioulgou and Panamasso used for this thesis were collected as part of the 3E program, a collaboration between the "*Ecole Polytechnique Fédérale de Lausanne (CH)*" and the "*Institut International d'Ingénierie de l'Eau et de l'Environnement (BF)*". The project is coordinated by Javier Perez Saez and supported by Dr. Natalie Ceperley and Dr. Théophile Mande in the Ecohydrology laborartory of Andrea Rinaldo and the Sensorscope Sarl team. Financial support was received from the Swiss Development and Cooperation Agency.

Contents

Variables Symbols and Explanation

Symbol	Explanation
β	Bias (addition constant) of the hidden and output layer.
D	Data, input and taget data.
ϵ	Model error, residual.
ξ	Threshold.
I	Precipitation.
$\mathcal{N}\left(\mu, \sigma^2\right)$	Gaussian distribution, with mean μ and standard deviation σ.
o_i	Output of hidden node i, in the ANN's hidden layer.
p	Momentum in the Hamiltonian equation.
q	Parameter state (or position of the particle) in the Hamiltonian equation.
τ	Time step or time interval.
t	Time
θ	Network Weight. Note that the bold version represents a vector of weights, and in the BNN chapter the ensemble of weights and biases.
w	Wind, usually characterized by an angle and a velocity.
x	Artificial Neural Network's input.
$\hat{y}$	Network output, prediction.
y, y_{tr}, y_{ts}	Target data (observations), used for training and testing.

Chapter 1 Introduction

1.1 Background and Motivation

In Burkina Faso, water resources have to be handled carefully. As stated in Ceperley et al. [3], water in the form of rainfall makes up for a large part of the water needs of agriculture. As agriculture strongly relies on precipitation, it is interesting to be able to forecast the intensity and the presence or absence of rain.

In Burkina Faso, two different diseases are transported by water: cholera to small extents, and schistosomiasis. In both cases, water is a dominant factor in the life-cycle of the disease. The main goal of the ECHO project

1 Egg excretion into the environment through urine and faeces. 2 hatching into the first larval stage - 0.1mm long miracidia. 3 Infection of the snail intermediate host (IH), *Bulinus* or *Biomphalaria*, within 12h of hatching. 4 Maturation and asexual reproduction of parasite in the IH leading to the emission of the second larval stage - 1mm long cercaria - after a pre-patent period of 1 month. 5 Human infection through skin penetration during prolongated water contacts within 24h of cercarial emission. 6 Parasite migration and maturation in the human final host (FH) during 1-3 months 7 Mature schistosomes lodged in the capillary veins around the bladder or the intestinal lumen undergo sexual reproduction resulting in a daily release of 300 to 3000 eggs during the 3-5yr lifespan of the schistosomes.

Figure 1.1: Schematic representation of the cycle of schistosomiasis. Source *F.-J. Perez-Saez*.

"*Understanding Schistosomiasis Transmission in Burkina Faso*" is to understand, model and develop strategies against schistosomiasis. The vector of the disease is a worm which needs two main factors for its development: humans in contact with water, and aquatic snails (see figure 1.1 for more details). The idea is to fight against the second vector of interest, the snails. To actively fight against these snails, one can either carry out large scale exterminations, or perform targeted treatments when the snails appear. This second solution requires a certain knowledge on the snails' habit. It has been shown (Poda et al. [1][2]) that the appearance of snails is largely related to the rainfall events in the region.

For these reasons, it might be interesting to be able to predict the rainfall, as a model could then potentially be developed to forecast the apparition of snails.

The focus of the current work is to develop a method to forecast local rainfalls in Burkina Faso. The idea is to use a machine learning technique called Artificial Neural Network to forecast the precipitation, along with Wavelet transforms and Bayesian probabilities.

1.2 Local Precipitation Forecasting

Bayesian Neural Networks have been applied in a few cases to hydrological problems. Most of the time, though, it is used to predict variables that do not vary in time as much as precipitation. The focus is more on modeling runoff (Khan et al. [35]), or the salinity and concentration of cyanobacteria in a river (Kingston [36]). Artificial Neural Networks coupled with Wavelet transforms seem to benefit from a large appreciation in the hydrological machine learning community (Nourani et al. [21]), but there seems to be no trace of Bayesian Neural Networks using Wavelet transforms to forecast hydrological events.

Coupling wavelet transforms to a Bayesian Neural Network for local precipitation prediction is therefore something that has not been tested yet. The advantage of the Bayesian method over a classical ANN approach is that it is able to estimate the parameter uncertainty, which means that each forecast is associated to a range of values which describes how confident the model is on the prediction. This allows the user to evaluate if he trusts the forecast, or if the model is too vague to be trusted.

Predicting local precipitation on the basis of small low-cost weather-stations in a climatic region where rainfall has a high temporal variability using these techniques is also something that has only been pursued in a few cases. Precipitation prediction often relies on large networks of sensors or satellite imaging [17] to predict precipitation at large scales, but not necessarily on local instruments for local precipitation forecasts.

1.3 Problematic and Objectives

As stated before, the focus will be on predicting precipitation. This will occur in different manners: forecast of the presence or absence of rain (binary output), forecast of the daily intensity, or forecast of the rain-class in which the event might occur.

The objective of this work is to test a Bayesian Network coupled with Wavelet Transform to predict short-term precipitation. This will be achieved in several steps. The first step is to acquire a large enough theoretical background so that it is possible to understand the principle in use. The second objective is to select a suitable code/toolbox/program to run the simulations. The third step requires to create and process a data-set which can be used in the frame of this work. The fourth part consists of adjusting the model according to the needs of this work. The fifth and last part will be to assess the quality of the model and suggest improvements.

1.4 Structure

This report will first present the theoretical background for the implementation of the forecasting, then explain the methodology, present the data, and explain the different choices of the implementation. The results will then be presented, and discussed. The last part will be a conclusion of the work achieved, as well as a discussion about the future improvements.

Chapter 2 Theoretical Background

This section serves as theoretical background to the notions used in the framework of this project, namely Artificial Neural Networks, Bayesian computation, and Wavelet transforms.

2.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are a class of probabilistic and statistical models used to link inputs to their corresponding output in a non-linear way. This section explains the mechanism of ANNs.

2.1.1 General

Artificial Neural Networks use weights, as well as biases, to act on inputs. The result is then passed through an activation function, which can either be a linear function, a tangent hyperbolic, or a sigmoid function. This layer of activation functions is called hidden layer, and each of these activation functions is called node. A neural network can be composed of several hidden layers, all linked together with different weight sets and biases. The results from these nodes are then again multiplied by weights and summed together with a final bias to produce the output of the net. Figure 2.1 shows the way an ANN with one hidden layer works (as used throughout this work).

As can be seen, the first step consists of multiplying the chosen inputs by adequate weights, which are then passed through an activation function, tanh in this case (Neal [23]):

$$o_i = \tanh\left(\beta_i^h + \sum_j^{N_x} \theta_{i,j}^h \cdot x_j\right) \tag{2.1}$$

where o_i denotes the output of hidden node i, β_i^h, the addition constant (bias) of hidden node i, x_j the j^{th} input, and $\theta_{i,j}^h$ the weight multiplying input j at hidden node i. The activation function (tanh) has the particularity to take values between -1 and 1, therefore allowing inputs to either influence

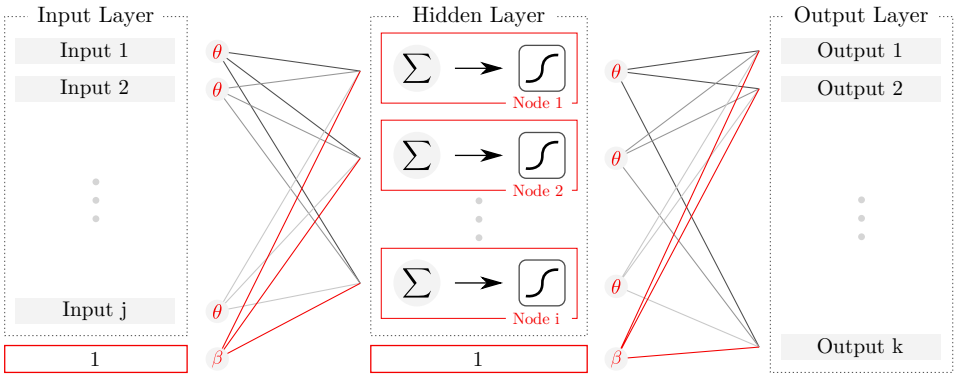

Figure 2.1: Schematic representation of the mode of operation of an ANN. Note that the β and θ symbols represent vectors containing weights, respectively biases, for every line they are connected to. The first symbol of each hidden node represents the sum of the inputs multiplied with their respective weights, and the second symbol represents the activation function.

positively, negatively or not at all the given nodes. The result from each node is then used to produce the net's outputs:

$$\hat{y}_k = \beta_k^o + \sum_i^{N_o} \theta_{i,k}^o \cdot o_i \tag{2.2}$$

where $\hat{y}_k$ denotes the output of node k. Depending on the final purpose of this output, different functions can then be used (Lampinen et al. [26]):

Logistic Output: The purpose of this type of output is to create a binary result which can be interpreted as "*true*" or "*false*". The output of the net is simply passed through a Sigmoid function:

$$P\left(\hat{y} = 1 | \mathbf{x}, \boldsymbol{\theta}\right) = \frac{1}{1 + \exp\left(-\hat{y}\right)} \tag{2.3}$$

This function ranges from 0 to 1. By applying a threshold to the result, the value takes a binary form.

Linear Output: In this case, the output is simply taken as is. Sometimes, if the observations are scaled for training (explained below), the output has to be scaled back:

$$\hat{y}_{unscaled} = \alpha_0 + \alpha_1 \cdot \hat{y} \tag{2.4}$$

where α is the parameter used for scaling.

Classification Output: This type of function is used when several outputs are computed to differentiate between multiple classes. The outputs

are transformed in such a way that they sum to 1 when put together, using a "*softmax*" function:

$$P\left(\hat{y} = k|\mathbf{x}, \boldsymbol{\theta}\right) = \frac{\mathrm{e}^{\hat{y}_k}}{\sum_l^{N_{\hat{y}}} \mathrm{e}^{\hat{y}_l}} \tag{2.5}$$

This allows to create a probability of a given output to be part of class k.

2.1.2 Training

Of course, to produce a correct estimation of the network's output, the weight and biases of the network need to be fitted. This process, called "*training*", consists of minimizing the error, or residual, of the model [26, 36]:

$$\begin{aligned} y &= f\left(\mathbf{x}, \boldsymbol{\theta}\right) + \epsilon & \text{(2.6a)} \\ &= \hat{y} + \epsilon & \text{(2.6b)} \end{aligned}$$

where ϵ is the model residual, and $\hat{y}$ the model prediction for the observation y, using the inputs $\mathbf{x}$, and the network's weights (including the biases) $\boldsymbol{\theta}$. For classical ANNs, there are different ways of fitting the weights. The one most used is backpropagation, and consists of assessing the error at the output of the network, and then "*backpropagate*" the error to each weight and bias (see ASCE Task Committee [17] for a more detailed explanation on this process).

For the training phase, the data-set is separated in two sub-sets: the training and testing set (y_{tr}, respectively y_{ts}). The training set is used to train the data. The testing set, also called validation set, is then used to see if the computed model also explains data outside of the computation domain, which tests the prediction capacity, or if is is only valid for the training data, which would probably mean that the model has has been overfitted (further explained in section 2.2.4).

2.1.3 ANN Applied to Hydrology

ANNs have been applied to hydrology for different problems. The ASCE Task Committee [18] gives a good overview on the different applications in the context of hydrology. These vary from rainfall-runoff modeling, to streamflows modeling, over water quality, ground water and finally precipitation modeling. Most works on precipitation have been focused on predicting short term precipitation of a couple of hours. Precipitation is also often forecasted over a region, rather than locally, using multiple stations or satellite imaging. The biggest hustle in predicting rainfall is the high temporal variability of the data. Indeed, rain might fall on one day, but be

completely absent on the next day, producing extreme peaks in the observations, and therefore in the data to predict. To overcome this problem, Partal et al. [19] and Nourani et al. [21] propose to use wavelet transforms associated with neural networks. The wavelets allow to characterize the changes in climatic conditions and therefore to capture the moment at which the conditions might assemble for rain to fall. Section 2.3 will further introduce this concept.

2.2 Bayesian Neural Networks

As said before, the ANN in a classical way is not used in this work. Instead, a variation of this principle using Bayesian probabilities is used. These networks are called Bayesian Artificial Neural Networks, or simply Bayesian Neural Networks (BNN).

2.2.1 General

Bayesian Neural Networks differ from ANNs in the way that instead of producing one specific value, they produce a range of possible values for each output, allowing therefore to assess the uncertainty, or the confidence interval which the prediction can handle. Each of the weights and biases described in 2.1.1 are expressed as a distribution, producing multiple possible outcomes for each network output. The probability density distribution of the weights given the data is called the "*posterior*" and is expressed using Bayes' theorem (Neal [23], Kingston [36], Khan et al. [35]):

$$P\left(\boldsymbol{\theta}|\mathbf{D}\right) = \frac{P\left(\mathbf{D}|\boldsymbol{\theta}\right)P\left(\boldsymbol{\theta}\right)}{P\left(\mathbf{D}\right)} = \frac{P\left(\mathbf{D}|\boldsymbol{\theta}\right)P\left(\boldsymbol{\theta}\right)}{\int P\left(\mathbf{D}|\boldsymbol{\theta}\right)P\left(\boldsymbol{\theta}\right)d\theta} \tag{2.7}$$

where D is the available data, which consists of the inputs of the net, as well as the target data. Note that in this chapter, $\boldsymbol{\theta}$ represents the weights as well as the biases to simplify the notation.

This equation contains three important concepts:

- $P\left(\mathbf{D}|\boldsymbol{\theta}\right)$ is called the "*likelihood*" and describes the error of the model (ϵ in equation 2.6).
- $P\left(\boldsymbol{\theta}\right)$ is called the "*prior*", and describes the a priori belief that one has on the weight distribution.
- $P\left(\boldsymbol{D}\right)$ is called the "*normalization factor*", or evidence of the model considered, and can be expressed as $\int P\left(\mathbf{D}|\boldsymbol{\theta}\right)P\left(\boldsymbol{\theta}\right)d\theta$. According to Khan et al. [35], this allows to make sure that "*the left hand side gives unity when integrated over all weight space*".

When expressed in terms of the input x to the net and the observations y used for training, equation 2.7 takes the following form [12, 28]:

$$P\left(\boldsymbol{\theta}|\mathbf{x_{tr}}, \mathbf{y_{tr}}\right) = \frac{P\left(\mathbf{y_{tr}}|\mathbf{x_{tr}}, \boldsymbol{\theta}\right) P\left(\boldsymbol{\theta}\right)}{\int P\left(\mathbf{y_{tr}}|\mathbf{x_{tr}}, \boldsymbol{\theta}\right) P\left(\boldsymbol{\theta}\right) d\theta} \tag{2.8}$$

The result from this equation can then be used to predict a value using new inputs to the system [12, 28, 34]:

$$P\left(\hat{y}_{new}|x_{new}, \mathbf{x_{tr}}, \hat{\mathbf{y}}_{\mathbf{tr}}\right) = \int P\left(\hat{y}_{new}|x_{new}, \boldsymbol{\theta}\right) P\left(\boldsymbol{\theta}|\mathbf{x_{tr}}, \mathbf{y_{tr}}\right) d\theta \tag{2.9}$$

This process is called "*marginalization*", and consists in drawing a new prediction with regards to the knowledge about the sampled weights, as well as the data and previous predictions.

2.2.2 Gaussian Approximation

The prior and likelihood in equation 2.7 need to be given a suitable distribution. This is, in the simplest case, done by Gaussian approximation. The following section explains how this is done.

2.2.2.1 Modeling of the Residual and Weights Distribution

The likelihood, which is also the noise of the model, is generally modeled by a Gaussian with 0 mean and standard deviation σ_y [36, 23]:

$$P\left(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}\right) = \prod_k \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(y_k - f\left(x_k, \boldsymbol{\theta}\right)\right)^2}{2\sigma_y^2}\right) \tag{2.10a}$$

$$= \prod_k \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\epsilon_k^2}{2\sigma_y^2}\right) \tag{2.10b}$$

In the same way, the weight's prior can also be modeled as a Gaussian [36]:

$$P\left(\boldsymbol{\theta}\right) = \prod_j P\left(\theta_j\right) = \prod_j \frac{1}{\sqrt{2\pi\sigma_\theta^2}} \exp\left(-\frac{\theta_j^2}{2\sigma_\theta^2}\right) \tag{2.11}$$

with mean 0 and standard deviation σ_θ.

2.2.2.2 Hyperparameters

Both standard deviations in equations 2.10 and 2.11 can be expressed by a fixed value, or can be modeled by a distribution. These standard deviations are called "*hyperparameters*" as they are parameters influencing the way the parameters (weights) of the model are distributed, but are a level above

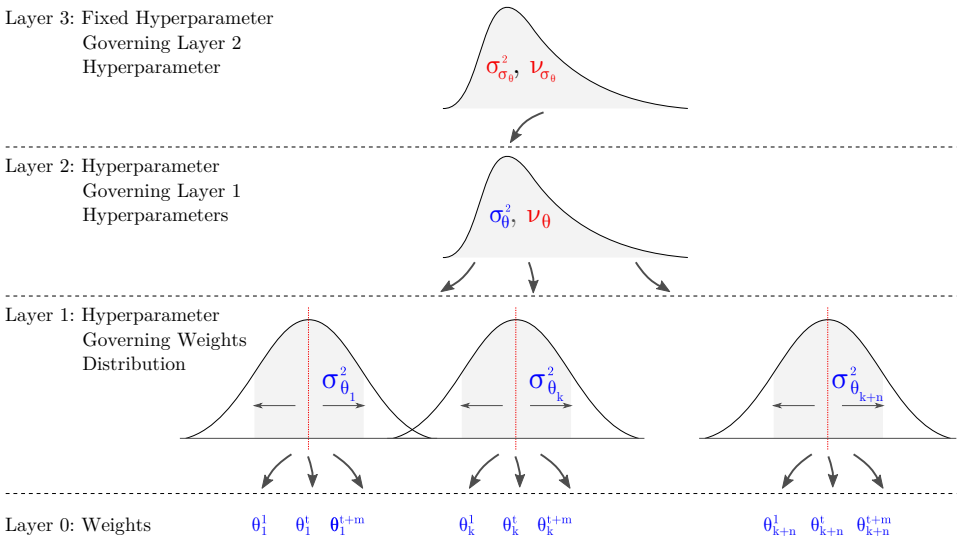

Figure 2.2: Schematic representation of the hierarchical model for the weights. The blue parameters indicate values which are modeled and vary with sampling. The red parameter indicate fixed constants.

the other parameters. Of course, expressing these hyperparameters as a distribution makes more sense, since the exact value of the residual and weights is not known from the start. The natural prior for this distribution is the conjugate of the Gaussian distribution with known mean (with known variance it's a gaussian), the inverse Gamma distribution (which is the same as the scaled inverse chi-squared distribution) [26, 36]:

$$\sigma_{\theta,y}^2 \sim \left(\sigma^2\right)^{-(\nu_\sigma/2+1)} \exp\left(-\frac{1}{2}\frac{\nu_\sigma \sigma_\sigma^2}{\sigma_{\theta,y}^2}\right) \tag{2.12}$$

where ν_σ is the number of degrees of freedom, and σ_σ the scale parameter. When choosing a value for these two parameters, one defines the prior distribution of the possible values that the standard deviation of the underlying model can adopt. Lampinen et al. [26] and Vanhatalo et al. [29], show that taking one standard deviation to express the distribution of all the parameters may result in poor modeling of the posterior distribution of the weights. To assess this problem, they propose to express each feature's weight distribution by its own standard deviation. The distribution of this resulting collection of standard deviations is then as well modeled using an inverse Gamma distribution. Depending on the implementation, there is one more level of hyperparamter modeling the scaling parameter of the inverse Gamma distribution.

Figure 2.2 is a schematic representation of the way these different layers of modelisation work together when considering weights. The first layer describes how each weight is modeled separately by a Gaussian, where the

standard deviations are then modeled by an inverse Gamma distribution, which is then in turn modeled by an inverse Gamma [26]:

$$\theta_k \sim \mathcal{N}\left(0, \sigma^2_{\theta_k}\right) \tag{2.13a}$$
$$\sigma^2_{\theta_k} \sim Inv-gamma\left(\nu_\theta, \sigma^2_\theta\right) \tag{2.13b}$$
$$\sigma^2_\theta \sim Inv-gamma\left(\nu_{\sigma_\theta}, \sigma^2_{\sigma_\theta}\right) \tag{2.13c}$$

The hyperparameters ν_θ, ν_{σ_θ} and σ_{σ_θ} have fixed chosen values, while the other parameters are picked from the governing distributions. This way of handling the weights distribution is called "*Automatic Relevance Determination*" (ARD) [26, 23].

The modeling of the error distribution is done in the same way[26]:

$$\epsilon \sim \mathcal{N}\left(0, \sigma^2_{\epsilon_k}\right) \tag{2.14a}$$
$$\sigma^2_{\epsilon_k} \sim Inv-gamma\left(\nu_\epsilon, \sigma^2_\epsilon\right) \tag{2.14b}$$
$$\sigma^2_\epsilon \sim Inv-gamma\left(\nu_{\sigma_\epsilon}, \sigma^2_{\sigma_\epsilon}\right) \tag{2.14c}$$

The usage of a Gaussian as a model for the error and the weights distribution is the simplest possible way of doing. Lampinen et al. [26] propose other distributions for the residual modeling, like the student's t distribution, but this has not been used in this work and will not be further explained.

2.2.3 Markov Chain Monte Carlo

For complex multidimensional problems, solving equation 2.9 analytically is not possible. As described in Neal [23], Andrieu et al. [27] as well as Khan et al. [35], the Markov Chain Monte Carlo method allows for a numerical integration of this integral:

$$P\left(y_{new}|x_{new}, \mathbf{x_{tr}}, \mathbf{y_{tr}}\right) = \frac{1}{N}\sum_{n=1}^{N} P\left(y_{new}|x_{new}, \boldsymbol{\theta}\right) \tag{2.15}$$

where the network's weights are generated by sampling from the posterior distribution of $P\left(\theta|D\right)$ knowing the prior distribution. This process of sampling and then analytically integrating is called the Markov Chain Monte Carlo Method. As it is impossible to sample directly from the posterior distribution, the samples are drawn from the prior distribution, until the distribution has reached an equilibrium.

2.2.3.1 Training

The training process in Bayesian Networks is very different than a classical ANN. Instead of trying to reach the perfect set of weights allowing for an exact modeling of the observed process, the Bayesian approach generates

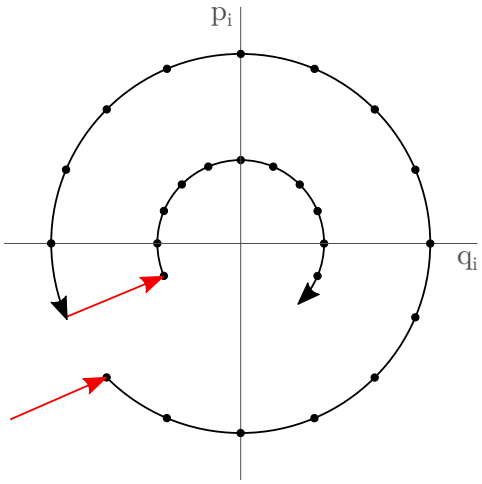

Figure 2.3: Schematic representation of the way the Hybrid Monte Carlo sampling and the Gibbs sampling work together. The red arrows represent the Gibbs sampling when the hyperparameters are changed and thus the total energy. The black dots and arrow represent the sampling of the HMC. The schema is inspired by the figures in Neal [30] and Hanson [24].

samples from the solution space, which are then all together solutions to the problem. In the methodology followed in this work, which is mainly influenced by the choice of software (see section 3.3.1), the sampling is done in two parts, using two different Markov Chain Monte Carlo Methods. Gibbs sampling to sample the hyperparameters, and Hybrid, also called Hamiltonian, Monte Carlo to sample the weights (see figure 2.3).

2.2.3.2 Gibbs Sampling

As mentioned, Gibbs is used to sample the hyperparameters. As stated in Neal [23], Gibbs allows to "*sample from a distribution over a multi-dimensional parameter,* $\theta = \{\theta_1, ..., \theta_p\}$". What this means, is that Gibbs allows to generate a new sample for each dimension given the other dimensions and the data, one dimension at a time [36]:

$$\theta^1_{t+1} \sim P\left(\theta^1_{t+1}|\theta^2_t, ..., \theta^p_t, \mathbf{y}\right) \tag{2.16a}$$
$$\theta^2_{t+1} \sim P\left(\theta^2_{t+1}|\theta^1_{t+1}, \theta^3_t, ..., \theta^p_t, \mathbf{y}\right) \tag{2.16b}$$
$$\vdots$$
$$\theta^p_{t+1} \sim P\left(\theta^p_{t+1}|\theta^1_{t+1}, \theta^2_{t+1}, ..., \theta^{p-1}_{t+1}, \mathbf{y}\right) \tag{2.16c}$$

Expressed in the case of the hyperparameters, this means that the sampling for each hyperparameter is done with regards to the other hyperparameters, each of which is changed in turn. This is possible because the joint

conditional distribution of the hyperparameters can be expressed by the inverse-gamma distribution.

2.2.3.3 Hybrid Monte Carlo

The two main problems faced when optimising parameters for Neural Nets are the danger to get trapped in local maxima, or to undergo random walk. The first problem is solved with the sole fact of using BNNs. As the purpose of the training is not to find one perfect solution, but a set of possible solutions, the state space is continued to be sampled even when an optimal solution is found.

A possible solution to the second problem has been developed by Neal [23], and is called the Hybrid Monte Carlo Method (HMC). The method takes advantage of the prior samplings as well as the direction of the gradient and the momentum to move along the state space in directions which are decorrelated from the previous samples. This makes it a very efficient way of avoiding random walk. The full concept is explained in this section.

The Hybrid Monte Carlo Method is based on a physical concept of energy conservation, a Hamiltonian system. A Hamiltonian describes a system where the total energy of the system is always constant and is the sum of two components, the potential and the kinetic energy:

$$H\left(q,p\right) = E_{pot}\left(q\right) + E_{kin}\left(p\right) \tag{2.17}$$

In such a system, the variation of one of the component is described by the variation of the total energy with respect to the other component:

$$\frac{dq}{dt} = \frac{\partial H\left(q,p\right)}{\partial p} \tag{2.18a}$$

$$\frac{dp}{dt} = -\frac{\partial H\left(q,p\right)}{\partial q} \tag{2.18b}$$

As explained in Neal [23], in the framework of Bayesian Networks, the variable q represents the parameters of the network. The potential energy of the Hamiltonian system is then expressed as [23, 30]:

$$E_{pot}\left(q\right) \propto -\log\left(P\left(q\right)\right) \tag{2.19}$$

The variable p is an artificial variable introduced to describe the momentum of the sampling, and has one component per parameter of the system. When combining equation 2.17 and 2.19, it is possible to see that the distributions for q and p are independent (Neal [30]):

$$P\left(q,p\right) \propto \exp\left(-H\left(q,p\right)\right) \tag{2.20a}$$

$$= \exp\left(-E_{pot}\left(q\right) - E_{kin}\left(p\right)\right)\right) \tag{2.20b}$$

$$\propto P\left(q\right)P\left(p\right) \tag{2.20c}$$

which makes it possible to sample from the joint distribution and then ignore the momentum to get the posterior network parameters. The kinetic energy is expressed as follow [30]:

$$E_{kin}\left(p\right) = p^{\top} M p = \sum \frac{p_i^2}{2m_i} \tag{2.21}$$

where M, repectively m_i, represents the mass of the system, respectively the importance of each component, usually one, as the same importance is accorded to each component. Using this information, equation 2.18a can be rewritten as follow:

$$\frac{dq_i}{dt} = \frac{\partial H\left(q,p\right)}{\partial p_i} = \frac{p_i}{m_i} \tag{2.22}$$

As can be read from equation 2.19, the potential energy is the negative log of the posterior distribution, which has been defined in section 2.2.2. Recalling this, the potential energy can be expressed as follow:

$$\begin{aligned}
E_{pot}\left(q\right) &\propto -\log\left[P\left(\boldsymbol{\theta}|\mathbf{D}\right)\right] && \text{(2.23a)}\\
&\propto -\log\left[P\left(\mathbf{D}|\boldsymbol{\theta}\right)P\left(\boldsymbol{\theta}\right)\right] && \text{(2.23b)}\\
\overset{\text{Gaussian approx.}}{} &\propto -\log\left[\exp\left(-\sum_k \frac{\epsilon_k^2}{2\sigma_{y_k}^2}\right)\exp\left(-\sum_j \frac{\theta_j^2}{2\sigma_{\theta_j}^2}\right)\right] && \text{(2.23c)}\\
&= \sum_k \frac{\epsilon_k^2}{2\sigma_{y_k}^2} + \sum_j \frac{\theta_j^2}{2\sigma_{\theta_j}^2} && \text{(2.23d)}
\end{aligned}$$

where the normalization constant in equation 2.7, as well as the constants in equations 2.10 and 2.11 are neglected. Equation 2.18 is of course a continuous equation. To be able to update (or sample) the momentum and parameters, one needs to discretise this equation. This is done with a method called "*leapfrog*" discretisation, which is derived from the Euler scheme. The process consists of performing half a step on the momentum equation in one direction, then to perform one step for the parameter equation using the result of the momentum equation, and finally to perform a second half step for the momentum [23, 29]:

$$\begin{aligned}
p_i\left(t+\tau/2\right) &= p_i\left(t\right) - \frac{\tau}{2}\frac{\partial E_{pot}\left(q\left(t\right)\right)}{\partial q_i} && \text{(2.24a)}\\
q_i\left(t+\tau\right) &= q_i\left(t\right) + \tau\frac{p_i\left(t+\tau/2\right)}{m_i} && \text{(2.24b)}\\
p_i\left(t+\tau\right) &= p_i\left(t+\tau/2\right) - \frac{\tau}{2}\frac{\partial E_{pot}\left(q\left(t+\tau\right)\right)}{\partial q_i} && \text{(2.24c)}
\end{aligned}$$

with the partial derivation of equation 2.23d being:

$$\frac{\partial E\left(q\left(\tau\right)\right)}{\partial q_i} = \frac{\epsilon_j}{\sigma_y^2} + \frac{\theta_j}{\sigma_{\theta_j}^2} \tag{2.25}$$

where ϵ_j corresponds to the network error backpropagated to each weight. The initial momentum p_0 is chosen randomly from a standard normal distribution. The leapfrog can be performed L times, so that it reaches the target time $t + L \cdot \tau$ (Neal [23]). This discrete method will produce a numerical error, leading to a variation in the total energy which is not zero, as it ought to be in a Hamiltonian system. To reduce this error, Neal [23] proposes to introduce two more steps: negate the momentum ($p = -p$), and evaluate if the new state is good enough using a method introduced in the Metropolis algorithm, by accepting the new state with probability:

$$\min\left(1, \exp\left(-\left(H\left(q\left(t + L \cdot \tau\right), p\left(t + L \cdot \tau\right)\right) - H\left(q\left(t\right), p\left(t\right)\right)\right)\right)\right) \tag{2.26}$$

otherwise the old state is reused.

As can be taken from all of this, the Hamiltonian Monte Carlo method intends to sample from a distribution with same total energy, which is defined by the hyperparameters. As stated before, the hyperparameters are sampled using the Gibbs sampler, and it is therefore important to follow this two-steps scheme, where the hyperparameters are sampled using Gibbs, and then the parameters using Hybrid Monte Carlo, and to do this multiple times.

2.2.4 Overfitting

Overfitting is characterised by the fact that a model fits the training data well, but performs very poorly on the test set. As stated in Neal [23], because of the way they sample the data, BNNs should not overfit, no matter the number of hidden nodes, nor the number of input features. In classical Artificial Neural Networks, to avoid overfitting the user is forced to introduce the concept of weight regularization. The weight regularisation adds a term to the error of the model which penalises the complexity of the network[36]:

$$E_{regularised} = E_y + \alpha E_w \tag{2.27}$$

This regularization occurs naturally in Bayesian Neural Networks, as the probability distribution of the weights with respect to the data already accounts for these two components (see equation 2.23d).

2.2.5 Feature Selection

Although overfitting should not occur in Bayesian Networks, selecting the correct inputs can still be of advantage, as it can speed up convergence. Too many inputs can lead to a slow process, as the network complexity increases, and certain inputs might just add more noise, and not necessarily much more information. As introduced in Neal [23] and elaborated in Vanhatalo et al. [29], to choose the right parameters, two processes have been developed:

Automatic Relevance Determination and Reversible Jump Markov Chain Monte Carlo (RJMCMC).

2.2.5.1 Automatic Relevance Determination

As described in section 2.2.2.2, ARD consists of assigning a Gaussian distribution with mean zero and standard deviation σ_{θ_k} to each parameter. These standard deviations are then controlled by a hyperparameter, which can increase or decrease the range on which the weights can be drawn from. The narrower the Gaussian, the smaller the weights are going to be, and therefore the less impact they will have. This process is active during the training, and can thus account for more or less useful inputs on the fly. A posteriori, it is possible to assess the importance of an input by observing the distribution of the standard deviation for each weight. The larger the distribution, the more important the input. Removing the less significant inputs is not necessary, but can improve the computation speed.

2.2.5.2 Reversible Jump Markov Chain Monte Carlo

RJMCMC is a modified version of the MCMC which allows to sample in a space with changing dimensionality. This makes it possible to change the number of inputs on the fly. Each input is then granted a probability to be used in the model based on similar concepts of energy as previously explained. As RJMCMC has not shown significant improvements in the forecasting capability of the model, it will not be explained in greater lengths here.

2.2.6 BNN Applied to Hydrology

Kingston [36] has dedicated a whole thesis to the problem of predicting hydrological phenomena with Bayesian Networks. As she states, the most considerable advantage of BNNs compared to ANNs is the capacity to assess the confidence interval of the model. This allows to predict a value, and to explain to which extent the model can actually be trusted. The bigger the confidence interval, the more certain the chance to mispredict is. Her applications are the forecast of river salinity and the concentration of cyanobacteria in different rivers in Australia. Both these forecast might have a lesser variability than precipitation, but are strongly related to climatic and hydrological processes, and show that it is possible to work with BNNs in that framework.

Khan et al. [35] have also developed a Bayesian Network to model a hydrological process. In their work, they successfully elaborated a rainfall-runoff model, which overperforms the physical, as well as the ANN model that had been developed for the same case-study. In their work they use Gaussian priors in the same way than described before.

2.3 Wavelet Transform

As mentioned in section 2.1.3, multiple studies seem to corroborate the use of wavelet transform to improve the forecasting capacity of the net. This section will serve as a brief introduction to this transform.

2.3.1 General

As stated in Nourani et al. [21], the Fourier analysis of a signal has a major drawback: the loss of time information. After performing a Fourier analysis, one might know which frequencies characterise the signal the best, but it is not possible to know where these frequencies actually influence the signal most. Wavelet transform allows to perform a pointed analysis at a given

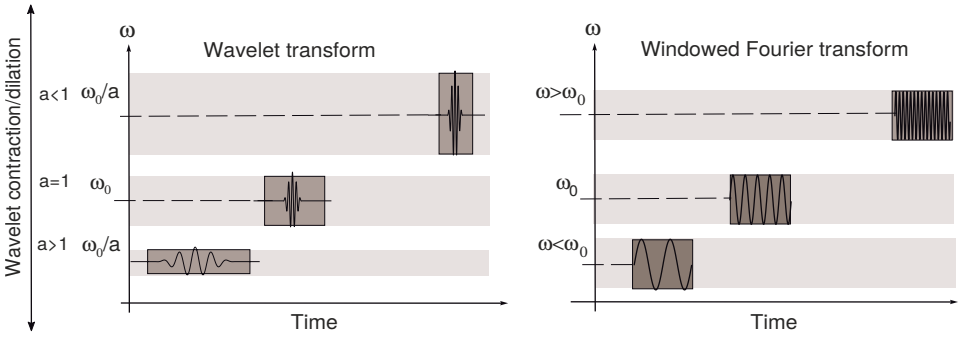

Figure 2.4: Schematic representation of the difference between wavelet transform (left) and windowed Fourier transorm (right), by making the frequency of the underlying data vary. Reproduced from *Cazelles et al. [11]*.

time step. Given the lower, or higher frequency at which the analysis needs to be performed, the window, or domain of influence of the transform is adapted, leading to a specific information on what frequencies play a major role at the given time step. Figure 2.4 shows how the scale and dilatation of the wavelet transform are influenced when observing signals with different frequencies. While the domain and amplitude of the Fourier transform stays the same for all the cases, the wavelet transform adapts the scale and the dilatation of the wavelet to the underlying frequency. The discrete wavelet transform is computed as follow [21]:

$$T_{m,n} = 2^{-m/2} \sum_{i=0}^{N-1} g\big(2^{-m} i - n\big) x_i \tag{2.28}$$

where g describes the "*mother wavelet*", the function, for example a *sinc*, that is used to sample the signal. $T_{m,n}$ represents the wavelet coefficient for a scale 2^m, and a location $n2^m$. The wavelet can have different scales, called levels, which will embrace more data, and therefore explain the variation at different time-scales.

2.3.2 Causal Wavelets

As the focus of the current experiment is on forecasting an event, it is of importance to consider that the wavelets need to be computed using only the time frame $t \in [-\infty, t]$, as no information should be available on future observations. Such a case is called causal wavelet. To do so, the wavelet transform for each time-step is computed using only the data gathered until then. The problem with this method is that the coefficient of interest is always located at the end of the window. and therefore exactly in the border effect. Indeed, the closer to the border, the harder it is to compute a correct wavelet transform. The technique used to compute the wavelet transform anyway, is to perform a symmetry on the data available, and to then perform the transform on this new data-set. Chaplais et al. [9] have tried to reduce this border effect using various filters, but have not been able to reduce the delay (the distance to the border) to zero, which makes it not of much use in the current application.

2.3.3 Wavelet Transform and Hydrology

As stated before, the wavelet transform captures changes in the signal structure. As rain is related to changes in atmospheric conditions, assessing these state alterations might be useful.

Nourani et al. [21] explain that the wavelets are especially important for rainfall forecasting since it is characterized by variations between zero-precipitation and high peaks of rain. The wavelets help the fitting of the model in this case. The difficulty relies in extracting the non-linearity of the process. They also state that the improvement in performance when using wavelet is "*greater for large scales such as monthly or seasonal data compared to hourly, daily or weekly*".

As stated before, Partal et al. [19] praise the usage of wavelets to predict daily precipitation. In their work, they state that higher levels (bigger windows) of wavelet transforms seem to have a great impact on the quality of the prediction. The fact that they do not talk about the usage of causal wavelets seem to indicate that the whole time-series has been used for the wavelet transform, which might allow for a model to use information that it is not suppose to have for a prediction. Their results vary from very satisfying to inconclusive, rendering the assessment of the utility of the wavelet difficult. A further statement they make is that the wavelet model allows to significantly improve the quality of the prediction during times with no rain, which might come in handy for discriminating between dry- and rainy-season.

Chapter 3 Materials and Methods

This section starts by presenting the methodology used in the project, continues with a review and a description of the available data, as well as the different ways they have been treated. The last part will explain the different steps involved in performing the experiments.

3.1 Methodology

The methodology for this work is separated in different steps, which are presented in figure 3.1.

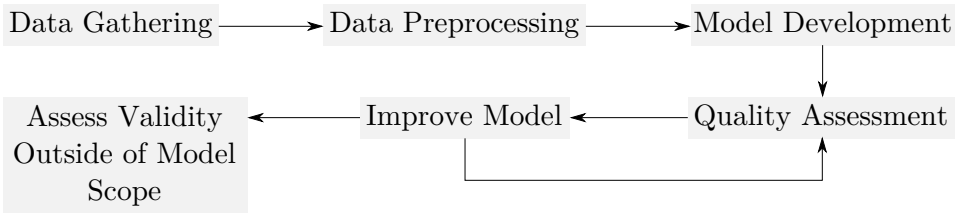

Figure 3.1: Schematic representation of methodology of the project.

The different steps will involve collecting and treating, as well as preprocessing the data necessary for the experiment. How this is done is explained in the next section. The next steps will involve developing the correct model for the experiment at hand, which strongly relies on the theory explained in last chapter, and a toolbox described later. This model will then be fine-tuned, as well as assessed using test sets in an iterative process. The last step will involve assessing the quality of the model outside of the scope of the fitting, meaning on stations or data sets which have not been used in the process of fine-tuning the model.

3.2 Data

This section describes the different locations and sensors deployed in Burkina Faso. The Focus will then be on how the data is treated to be used in the

experiment. Some statistics and characteristics of the data will then be shown.

3.2.1 Climate in Burkina Faso

The climatic condition in Burkina Faso is characterized by a strong North to South gradient of temperature and rainfall intensity. As explained in Couttet [6], Burkina Faso is characterized by three distinct climatic regions: the Sahelian zone in the North, the Sudoan-Sahelian zone in central Burkina Faso, and the Sudanian zone in the south. The avarage rainfall as well as the rainy-season duration increase when moving down to the south of the country. Ceperley et al. [3], as well as Mande [5] mention that the climate is strongly characterized by seasonality, meaning that there are mainly two seasons: a rainy season which makes up for most of the rain during the year, and a dry-season. The rainy season comes from May to September [6]. A small rain event called "*pluie des mangues*" often occurs in may and is a precursor of the start of the real rainy-season which starts around a month later.

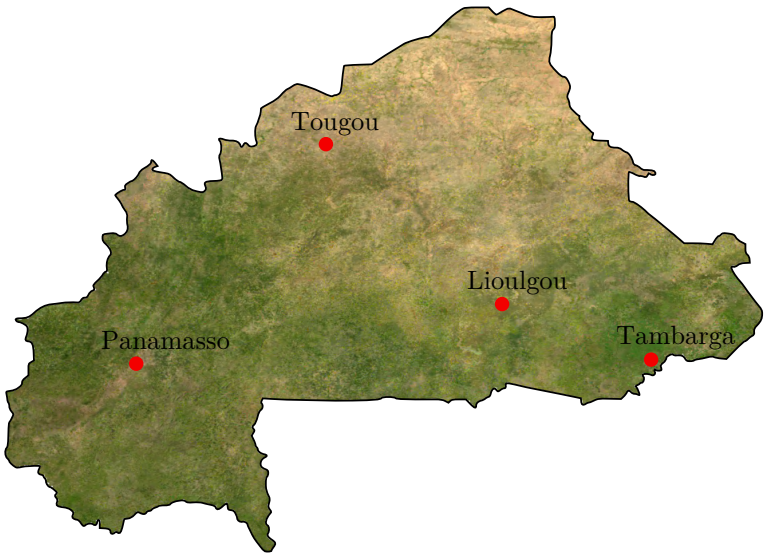

Figure 3.2: Satellite image of Burkina Faso with the four different villages of interest[1]. Note that the three different climatic regions can be guessed by the degree of vegetation displayed in the image.

[1]Modified from `http://commons.wikimedia.org/wiki/File:Burkina_sat.png`

3.2.2 Situation

Together with 2iE[2] and Sensorscope[3], over the years, the ECHO and EFLUM[4] laboratories have deployed several small weather stations in Burkina Faso (see Acknowledgments for more information about the project's partners). Figure 3.2 shows the four locations at which mobile stations have been deployed, and table 3.1 the number of active and inactive stations per locations, as well as the commissioning date of the oldest station. The different locations and stations have varying instruments, but a common denominator of interest for most of the stations is given as the following environmental parameters:

Table 3.1: Overview of the number of active (AS) and inactive (IS) stations for each locations, as well as the commissioning date of the oldest station (stand: 14.06.2015).

Place	# AS	# IS	Oldest Station
Tambarga	4	23	June 2009
Lioulgou	3	-	July 2014
Tougou	7	2	April 2014
Panamasso	2	1	June 2014

- Rain
- Solar Radiation
- Air Temperature
- Humidity
- Wind Direction and Speed

Depending on the station, several other values are measured, as for example soil moisture and temperature, vapor pressure, etc.

3.2.3 Stations

From the four locations available, only the stations in Tambarga are in place since more than a year, namely around six years (see table 3.1). Since for the experiment at hand, the requirement is to have several years worth of data to be able to perform the tests, Tambarga was chosen as a place of

[2]International Institute for Water and Environmental Engineering - `http://www.2ie-edu.org/`

[3]Sensoroscope - `http://www.sensorscope.ch/`

[4]Laboratory of Environmental Fluid Mechanics and Hydrology - `http://eflum.epfl.ch/`

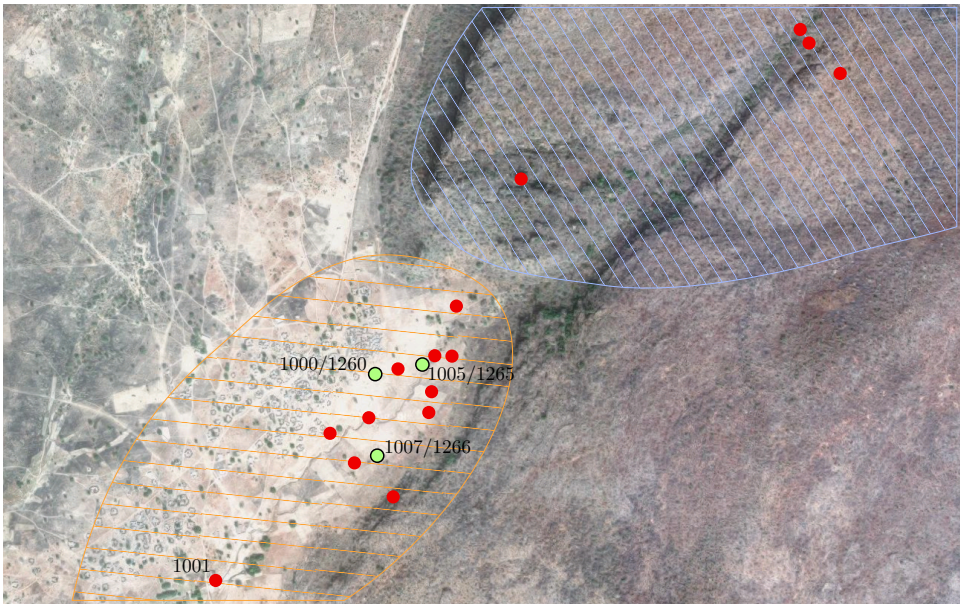

Figure 3.3: Situation Map of the region around the village of Tambarga[5]. The blueish dashed region indicates the location uphill, and the orange dashed region the location in the valley. The selected stations are marked in green.

study. The twenty-seven stations mentioned in table 3.1 are not all still operational. Most of them have been replaced in 2011 - 2012. The sensors have therefore changed, but a continuity of the measures is somewhat granted.
The stations in Tambarga are located in two different environments. The first is down in the valley, where agriculture is pursued. As described in Ceperley et al. [3], "*the landscape is a patchwork of tall grasses, meadows, interspersed large African ebony [..] and Baobab [..] trees, and shrub woodlands dominated by Combretaceae, wetlands, marshes, and riparian gallery forests*". Mande et al. [8] describe the location as a "*large agroforestry field used for millet and rice plantation during the rainy season*".

The second location is uphill, and is mainly covered by savanna forest [8]. As this second site has different climatic characteristics than the first location, the stations from this location are disregarded.

Not all the stations can be used for the experiment, because some of them differ too much in climatic characteristics, or have been damaged over the years. For instance, station 1001 was in a rice-field, and fell down, leading to wrong records until it was fixed, and station 1005 was trampled down by a donkey, also leading to unusable data (source: Dr. N. Ceperley). Other do not have any rain measurement, which make them unsuited for the experiment. Lastly, some of the stations are placed near, or under

[5]Satellite image from `https://www.google.com/maps/@11.4453628,1.2173366,2196m/`

trees, which may result in very different rain, solar and temperature readings (stations 1012, 1013, 1003, 1262, 1263).

All in all, from all the stations located in the valley, the stations used are:

- Station 1000, and its replacement, station 1260.
- Station 1007, and its replacement, station 1266.
- Station 1265, which has no reliable previous station, as it would be station 1005 which was trampled down.

These stations are not too far apart (less than a hundred meters), which makes them ideal candidates to be used together.

3.2.4 Data Aggregation and Filtering

The main problem faced with the data, is that all the stations contain missing values. As can be seen in figure 3.4, there are several gaps in the recordings, as well as some data that seem very suspicious at some stations. To be

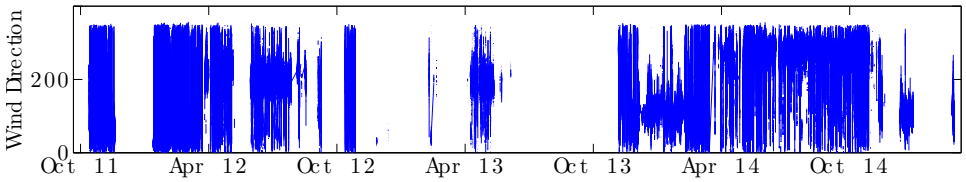

Figure 3.4: Wind measurements for station 1266 given in [°]. The gaps in the measures indicate that there has not been any recording at this moment. As can be seen there are several gaps of relatively consequent size, which are not always covered by other stations, leading to holes in the time-series continuity.

able to perform the simulations without too much trouble, it seemed necessary to aggregate all the usable stations together and to use the resulting time series. The following section explains how the data was first treated and then aggregated.

3.2.4.1 Precipitation

The precipitation data are composed of filtered data of three different sources. Dr. T. Mande and Dr. N. Ceperley both already did some extensive work on these data sets and created some aggregated and filtered time series composed of several stations. The data from Dr. Ceperley covers a time-period from early 2009 to late 2013. Dr. Mande's data cover the time from early 2010 to late 2012. Since the goal was to create a time series as long as possible, the data from the stations from early 2012 to early 2015 described

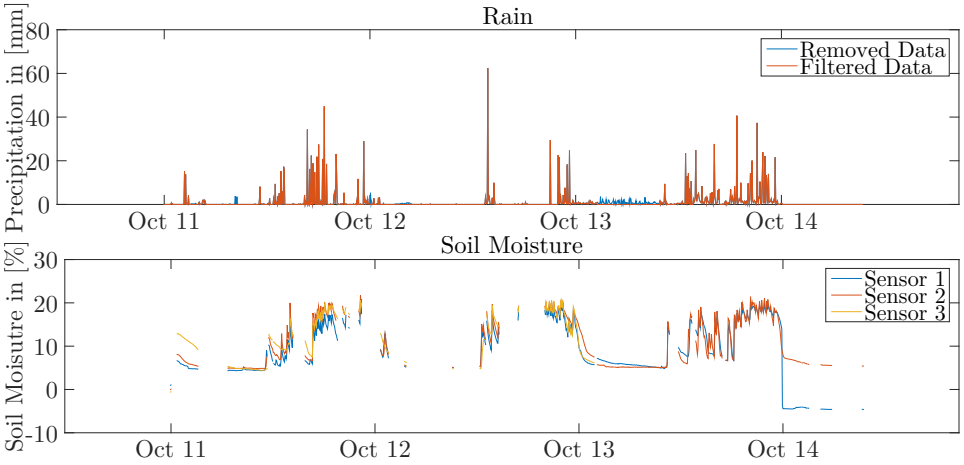

Figure 3.5: Example on how the rain is filtered for Station 1260. The first plot shows the rain measurements, and the second plot the soil moisture at different depths.

before were also included, after aggregation and filtering. This process is explained now.

The stations have a sampling frequency of $1/[minute]$. The decision was taken to use the daily cumulative rainfall as a value to be predicted, leading to the necessity to transform these observations into a daily cumulative rainfall time-series. This was done by summing the precipitation over 24 hours, ranging from midnight to midnight. To make sure that small events due to wind, or other nuisances, are not present when merging the observations, it is important to filter the data. To achieve this, the observations from each station are compared to the soil moisture readings from the same station. As shown in figure 3.5, the precipitation events are most of the time followed by strong variations in moisture readings. This knowledge is used to filter out "*ghost*" events, as for example the rain events on Jun 2014 (see blue oscillations in the upper plot of figure 3.5), where the data seem to indicate a precipitation event, but the soil moisture is constantly decreasing, showing that no rain occurred in this period of time. The opposite effect can be observed in May 2012 and June 2013, when strong rain events occurred, followed by a response of the soil moisture. To filter out these "*ghost*" events, a moving window over $2 \cdot \tau$ days with the following rule is used:

$$I(t) = \begin{cases} I(t) & \text{if } \Delta(SM(t \pm \tau)) > \xi_1 \text{ and } \sigma(SM(t \pm \tau)) > \xi_2 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where I denotes the precipitation reading at time t, SM the soil moisture, ξ the chosen thresholds, Δ the difference between the maximum and the minimum in the interval, and σ the standard deviation.

After different tests, the seemingly working parameters have been found

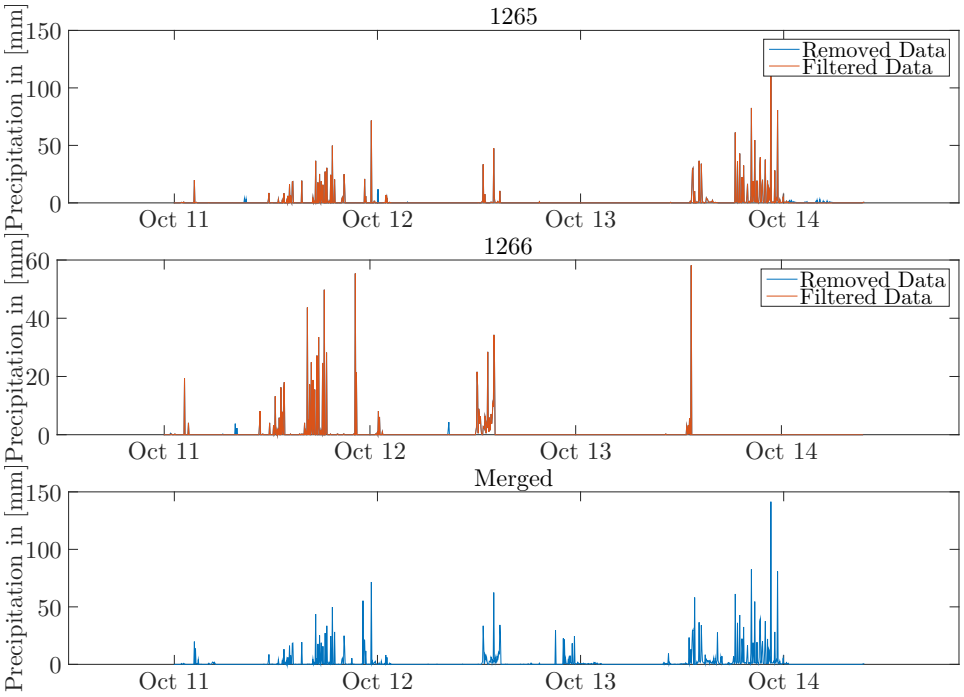

Figure 3.6: Un- and filtered rain by stations and merged as explained in the section.

to be a window of 20 $[days]$, an ξ_1 of 5 and an ξ_2 of 0.4. Figure 3.6 shows the filtering and the merging of the three stations of interest. The noisy data seem to be well filtered out, which is also confirmed by comparing the result to the other sources, as can be seen in the upper plot of figure 3.7. The merging is done by taking the maximum of the three stations for each

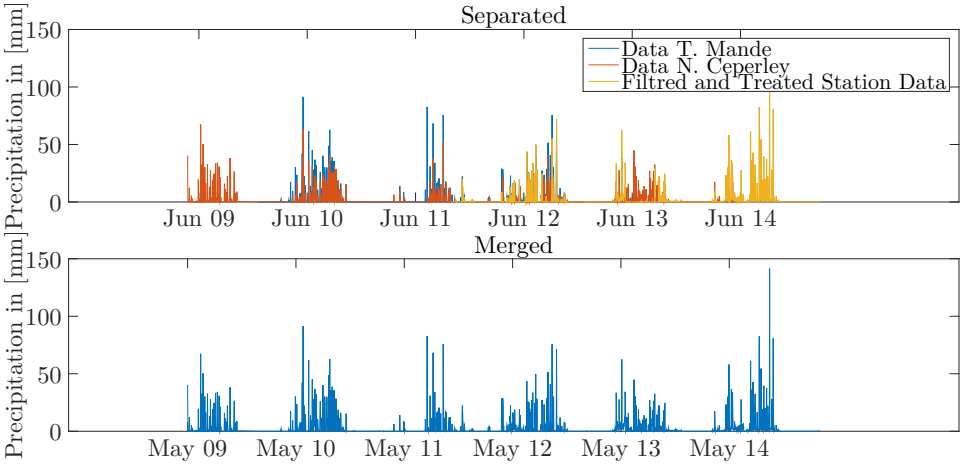

Figure 3.7: Separate and merged data from the three sources (Dr. Mande, Dr. Ceperley and the filtered stations).

day. Finally the data from the three sources are merged together, again by taking the maximum for each day. The result can be seen in the lower plot of figure 3.7.

3.2.4.2 Climatic Parameters

Several parameters were chosen, selected for their potential precipitation forecasting capability. The different way each of them is treated and how the stations are aggregated is explained in the following section.

Temperature, Humidity and Solar Radiation

Temperature, Humidity and Solar Radiation are treated in the same way. For each of the stations, the mean, min, max and standard deviations are computed for a window of 24 hours. The stations are then merged together. The global mean between stations is computed with regard to the number of measurements of each station, minimizing the effect of one single measurement:

$$\bar{x}_{i,j} = \frac{N_i \cdot \bar{x}_i + N_j \cdot \bar{x}_j}{N_i + N_j} \tag{3.2}$$

where $\bar{x}$ denotes the averaged climatic parameter over 24 hours, i, j station i, respectively j, and N the number of measurements for that laps of time for the given station.

The merged standard deviation is computed in a similar fashion. Headrick [13] shows a way of computing the global true standard deviation given multiple subgroups of data:

$$\begin{aligned}\sigma_{i,j}^2 = \Big(&{N_i}^2\,{\sigma_i}^2 - N_i\,{\sigma_j}^2 - N_j\,{\sigma_i}^2 - N_j\,{\sigma_j}^2 - N_i\,{\sigma_i}^2 + {N_j}^2\,{\sigma_j}^2 + N_i\,N_j\,{\sigma_i}^2 \\ &+ N_i\,N_j\,{\sigma_j}^2 + N_i\,N_j\,(\bar{x}_i - \bar{x}_j)^2\Big) \Big/ \left((N_i + N_j)\,(N_i + N_j - 1)\right)\end{aligned} \tag{3.3}$$

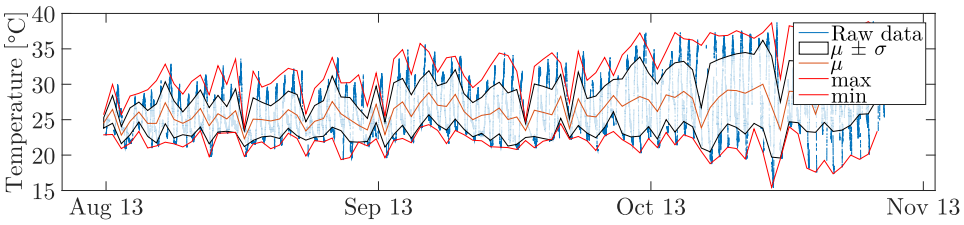

Figure 3.8: Example on how the temperature data is aggregated by day. The data originates from station 1266 for a chosen period of three months.

The min and max are computed by taking the minimum, respectively the maximum of the stations for each time window.

Wind

The wind is recorded as an angle ranging between 0 and 360 [*degrees*], as well as an intensity given in [m/s]. The average wind speed and direction can not just be computed by a mathematical mean as done for the other parameters. Instead, as explained in Olson [14], the wind angle and intensity are transformed into vectors, and then added together:

$$\mathbf{w} = \sum w_{speed} \cdot \begin{bmatrix} \cos(w_{angle}) \\ \sin(w_{angle}) \end{bmatrix} \tag{3.4}$$

This creates a vector with a certain speed and direction, which is then to be translated back to an average daily angle and speed:

$$\bar{w}_{angle} = \text{atan2}(w_y, w_x) \tag{3.5}$$

$$\bar{w}_{speed} = \sqrt{w_x^2 + w_y^2} \tag{3.6}$$

This way, a wind blowing in one direction for most of the day with a low speed, will have less impact than a strong wind blowing in the opposite direction for one hour.

3.2.5 General Data Analysis

This section serves as a general explanation of the rain data, extracting some general trends as well as correlations between parameters and precipitation, which will allow for an initial understanding of the parameter selection.

3.2.5.1 Fourier Transform

The discrete Fourier transform allows to extract the characteristic frequencies

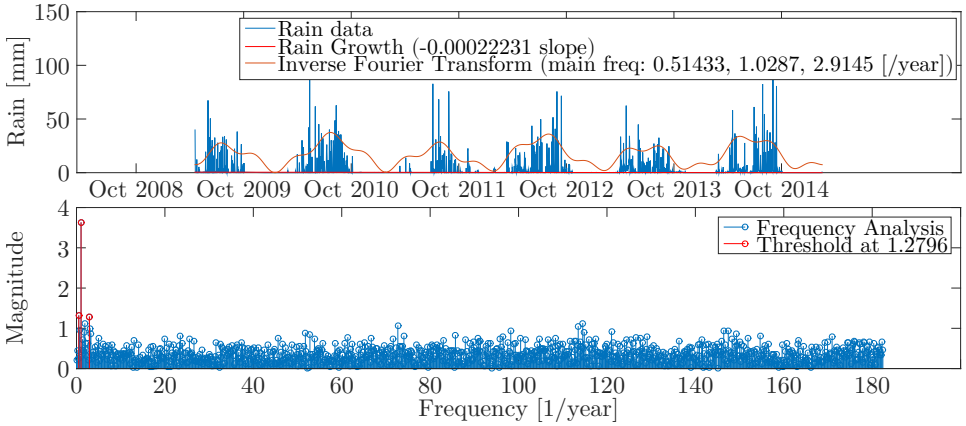

Figure 3.9: Upper plot: rain data (blue) with trend (red line), and inverse transform of the three main frequencies (orange). Lower plot: Discrete Fourier transform with the two main frequencies highlighted in red.

of a signal, as well as their corresponding magnitudes. This permits to understand the general trends behind the data. As can be seen in figure 3.9, the main frequencies of the rain data is around 1, 0.5 and 2.9 [/year], which clearly shows a seasonal trend. The yearly frequency explains that the rain intensity is similar from one year to the other. It is however interesting to notice that the frequency is not of exactly one, but 1.0287, which could be explained by a slight shift in the rain season. The half-year period (frequency of around 2.9) can be explained by the variation of rain season - dry season. The last two-year period (frequency of 0.5) seems to indicate that every second year is prone to have less rain. The negative general slope indicates a generally downward trend in the amount of rain, but given the very low value of the slope, this does not seem very relevant.

3.2.5.2 Wavelet Transform

The wavelet transform as explained in the previous chapter has been applied to the rain, as well as the temperature time-series. Figure 3.10 shows this decomposition. As can be seen, the wavelet transform captures the tran-

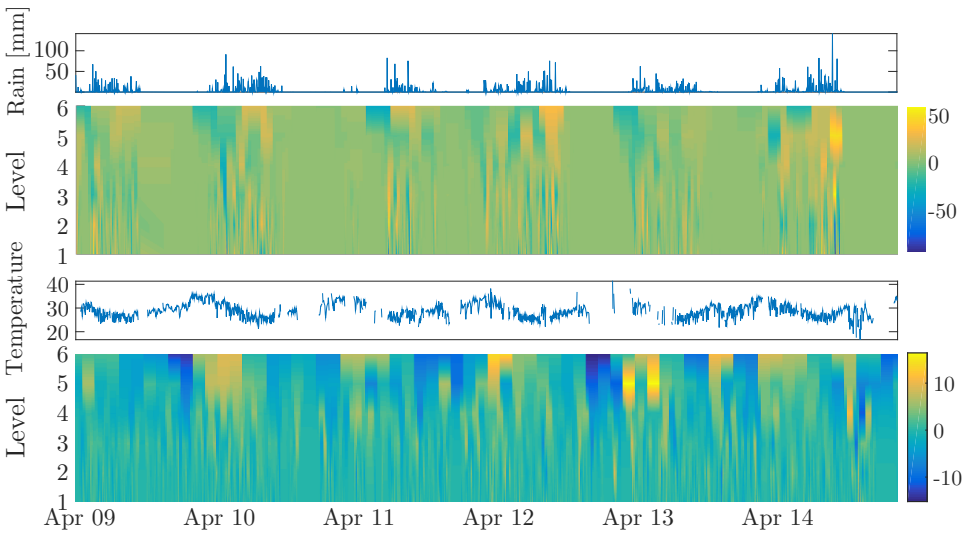

Figure 3.10: From top to bottom: Rain data and the resulting wavelet transform, temperature in [°] and the resulting wavelet transform.

sition between dry- and rainy-season in the wavelet transform for the rain data. The wavelet transform for the temperature also seem to follow this logic. In the temperature transform the seasonal periodicity is clearly to be seen. At small time-scale (lower level), the wavelet coefficients seem to be more present when rain occurs, leading to the possible conclusion that this transform might capture the changes occurring when rain starts.

3.2.5.3 Cross-Correlations

The cross-correlation expresses the linear correlation between two variables when shifted in time. It is computed as follow:

$$\mathbf{R}_{x_1,x_2}(t) = \mathbf{E}\left[x_{1,t+\tau} x_{2,t}\right] \tag{3.7}$$

where $x_{1,2}$ are two parameters. The resulting value can be useful to assess the linear relations between different parameters. Figure 3.11 shows the cross-correlation between the rain measurements and the different climatic parameters described previously. After a first analysis of figure 3.11,

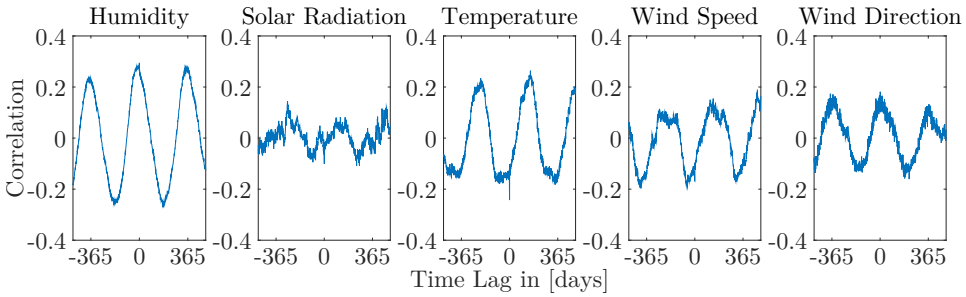

Figure 3.11: Cross-correlation between the rain measurements and the Humidity, Solar Radiation, Temperature, Wind Speed and Direction readings for the aggregated data.

the conclusion that can be drawn is that all the parameters have a cyclic correlation to the rain. Humidity and temperature seem to be the most correlated with rain, but interestingly, the wind speed and direction seem to oscillate a lot at fine time scale. This could possibly mean that at large time-scale, or for general trends, humidity and temperature may play an important factor, but that for the small time variations, wind, as well as solar radiation might be of more use. Generally speaking, this cyclic trend tends towards the conclusion that if a model might explain the precipitation for one year, it might very well be able to predict it for an other year. Most of the parameters seem to express the highest absolute valued correlation around zero time-lag. This means that it might be optimal to take inputs around the day of interest.

The attention has to be drawn to the fact that this cross-correlation is a linear understating of the relations between parameters. Neural Networks do not model in a linear way, but model non-linear problems. This means that these information can be interesting, but might not be the only possible interpretation of the relevance of the parameters, as well as the optimal time-lag.

3.2.6 Prior Knowledge from literature

Mande [5] has done some extensive work on the rainfall characteristics in Tambarga. In his works, he states that the "*rainfall regime is mono-modal*", meaning that there is one rainy season which ranges from May to October, followed by one dry season. Rain is the main component of precipitation at Tambarga. He also states that the temporal variability of rain in Tambarga is enormous. As he says, there is a difference of around 50% rainfall between 2010 and 2011. He also states that extreme rainfalls are dependent on high temperatures, and that "*rain at Tambarga is triggered by the convective mechanism, which is mainly controlled by the sensible heat flux. The daytime rain events [...] are convective and are characterized by their high intensities and short durations*". In an other work, Mande et al. [8] reinforce that "*the rainfalls are short in duration, intense, and occur mostly during daytime primarily due to convective activities*".

With respect to this information, it it safe to assume that temperature and humidity, as well as a method to capture the changes occurring might be of importance to model the underlying phenomena of rainfall.

3.3 Experimental Setup

This section explains the different steps involved before the actual experiment can start.

3.3.1 Software

This section will briefly explain the choice of the software and the way it works.

3.3.1.1 Software Choice

As the focus of this work is on forecasting rain, and not on implementing a Bayesian Neural Network, a prior selection of software had to be done. The choice was made to use the Matlab toolbox developed by *Aki Vehtari* and his team, called "*MCMC Methods for MLP and GP and Stuff (for Matlab) V2.1*"[6]. The toolbox comes with a comprehensive manual/publication: *"MCMC methods for MLP-network and Gaussian process and stuff – a documentation for matlab toolbox MCMCstuff"* [29]. The toolbox has a complete implementation of the Bayesian Neural Network and MCMC algorithm described in section 2.2. Matlab itself has good tools for data processing, as well as a toolbox for Wavelet Transforms[7], and therefore meets all the requirements for this project.

[6] `http://becs.aalto.fi/en/research/bayes/mcmcstuff/`
[7] `http://www.mathworks.com/help/wavelet/`

3.3.1.2 Toolbox Operating Mode

The algorithms used in the toolbox are implemented according to the theory explained in section 2.2. Figure 3.12 shows how the different parts fit

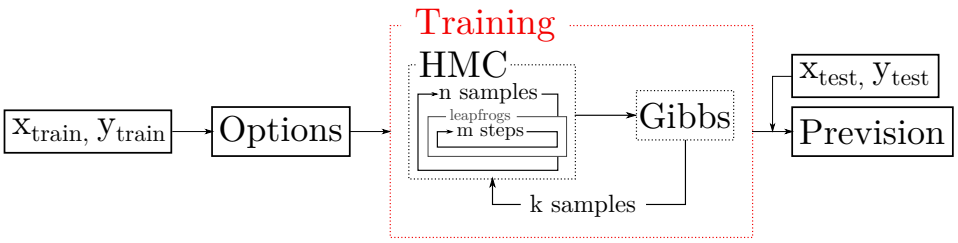

Figure 3.12: Schematic representation of the operating mode of the software.

together in the code. It is important to notice that there are two different constants governing the sample process. The first is a global sampling of k samples, and the second a sampling which is done only for the Hybrid Monte Carlo, of n samples, which keeps the same hyperparameters (sampled with Gibbs) between rounds.

3.3.2 Data Preprocessing

Neural Networks have technically the ability to predict any value given any inputs by changing the weights accordingly. That being said, it has been shown (Kingston [36]) that it helps a lot if the input, as well as the target output are in the same range of values. The complexity of the solution surface is also of great importance. The more unpredictable and the harsher the change between time-steps, the harder it will be for the net to perform well when a prediction has to be made. There are different solutions to transform the data into a smoother solution space. Some of these transformations have to accuse a loss or a transformation of the information, which might result in the necessity to interpret the output differently, and can make it less important for prediction. The following subsection briefly explains the different possibilities to achieve these desired effects.

3.3.2.1 Normalization and Standardization

One of the solutions to get all the data in the same range is to normalize the data. A special case of normalization is standardization, which transforms the data so that they have an average of zero and a standard deviation of one:

$$D_{std} = \frac{D - \mu(D)}{\sigma(D)} \tag{3.8}$$

where D denotes the data, and the symbols σ and μ the standard deviation, respectively the average of the data.

An other case of normalization is to map the data between a minimum and a maximum (usually -1 and 1):

$$D_{norm} = a + (b - a) \frac{D - \min(D)}{\max(D) - \min(D)} \tag{3.9}$$

where a and b are the lower, respectively the upper limits of the distribution. To distinguish between these two transforms, this second transform will simply be referred to as normalization in the rest of the document (as opposed to standardization).

Both of these transformations allow for an easier fitting of the parameter, as the weights multiplying the different inputs will be distributed around the same values.

3.3.2.2 Logarithm

When the target or input data is not normally distributed, it can help to perform a logarithmic transformation on the data (Kingston [36]). This will reduce the "*non-normality*" of the data and allow for an easier modelling when using normally distributed data as input.

3.3.2.3 Cumulative Sum

A further way to smoothen the solution surface it to apply a window on the data which will perform a given operation. In the current context, taking the sum of the windowed values seems to make sense, since it would then mean taking the cumulative rainfall over a period of time dt.

$$D_{windowed}[t] = \sum_{\tau=0}^{T=dt} D[t + \tau] \tag{3.10}$$

For the current experiment, this would, for example, mean to take the cumulative rainfall over a time laps of one week. The solution would then move from predicting the value of a single day to the cumulated rainfall of one week.

3.3.3 Output Choices

In the context of the current experiment, three different outputs are tested.

3.3.3.1 Binary Output

The first type of forecasting is defined as a binary output predicting the presence or absence of rain. The statement is simply as follow:

$$I_{binary} = \begin{cases} 1 \text{ if } p > \xi \\ 0 \text{ else} \end{cases} \tag{3.11}$$

The threshold can be chosen to be the precision of the rain measuring instrument, which can be regarded as one tick of the tipping bucket of the Davis Rain Collector[8]. One tick of this instrument corresponds to 0.2 $[mm]$.

3.3.3.2 Linear Output

The second choice of output is on the linear forecast of the cumulated rain over a given time-laps. Different time-rain setups are tested:

- Forecast on the next day (24 $[hours]$ forecast).
- Forecast on the next week (7 $[days]$ days forecast).
- Forecast of the daily cumulative rain ($[mm]$ of rain fallen in 24 $[hours]$).
- Forecast on the weekly cumulative rain ($[mm]$ of rain fallen in the next 7 $[days]$).

One of these possibilities would for example be to predict the rain fallen over one week in 7 days, which would mean the cumulated rain fallen from $t + 7\,[days]$ to $t + 14\,[days]$.

3.3.3.3 Class Output

The last output is a classification of the different intensities of rain. The rain values are separated in different classes using the three quartiles, and the value of one tick (abbreviated OT) explained before (see table 3.2). Figure

Table 3.2: Different categories of rain and their corresponding distribution. Note that the quantile ($\hat{q}$) refer to a distribution where the "*no rain*" values are removed, like shown in the boxplot of figure 3.13.

Class Name	Statistical Limit	Range $[mm]$	Count	[%]
No Rain	0.0 - OT	0.0 - 0.2	1309	64.87
Small Rain	OT - $\hat{q}(25\%)$	0.2 - 0.5	180	8.92
Medium Rain	$\hat{q}(25\%)$ - $\hat{q}(50\%)$	0.5 - 2.5	177	8.77
Heavy Rain	$\hat{q}(50\%)$ - $\hat{q}(75\%)$	2.5 - 11.9	172	8.52
Extreme Event	$> \hat{q}(75\%)$	> 11.9	180	8.92

3.13 shows how the data used for the current experiment are divided in classes. As can be seen, most of the data is either contained in low, or no rain classes.

[8] http://www.davisnet.com/product_documents/weather/manuals/07395-275_IM_07852.pdf

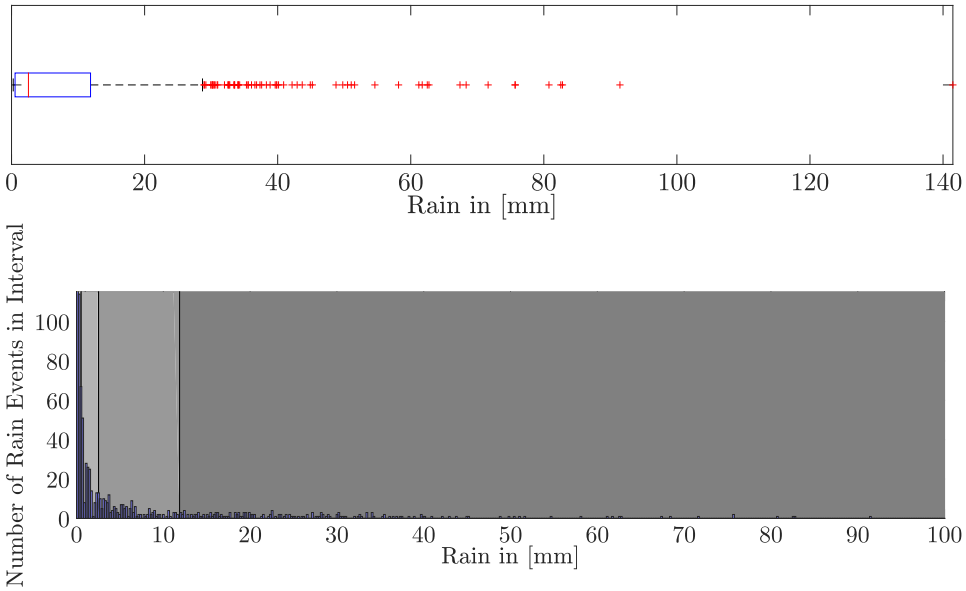

Figure 3.13: Boxplot of the distribution of rain measurements for values $> 0.2\,[mm]$ (upper figure). Histogram of the rain measurements (lower figure). Note that for the histogram, the limits of the x and y axis have been modified so to improve the readability. The limits for the Boxplot remain unchanged. The bar corresponding to the values below $0.2\,[mm]$ contains 1309 values.

3.3.4 Error Assessment

To assess the error of the predicted data compared to the observed data, different statistical tools are used.

3.3.4.1 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC), as described in Fawcett [10], is used to assess the result of binary forecast, and is a measure on how the prediction is better or worse than a random guess. ROC is defined as a plot of the false positive rate (FPR) against the true positive rate (TPR). The FPR is the ratio between the false positive, the number of predictions that give a "*true*" but should actually give "*false*", and the number of negative observations, which is the same as the number of false positive added to the number of true negative, the number of "*false*" forecasts that are actually observed as "*false*"[10]:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{3.12}$$

The TPR is the ratio between the number of true positive, the number of predictions that give a "*true*" and have the same observation, and the number of positive observations, which is equal to the number of true positive

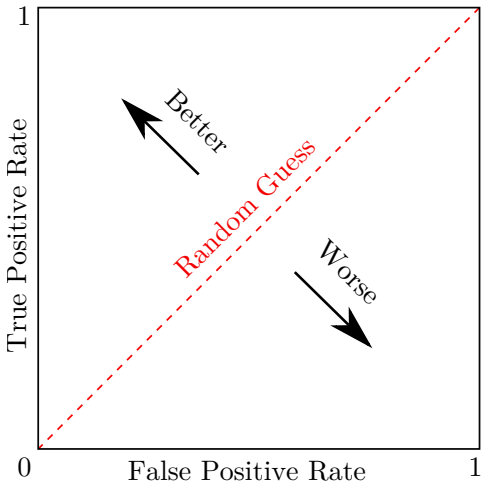

Figure 3.14: Schematic representation of a Receiver Operating Characteristic graph[9]. The arrow showing the direction of "*worse*" states the direction in which the prediction is worse than a random guess (red dashed line), and the arrow showing the direction of "*better*" indicates the opposite.

added to the number of false negative, which is the number of prediction stating "*false*" but that should actually be "*true*" [10]:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3.13}$$

The different values of FPR and TPR are then obtained by letting the threshold between "*true*" and "*false*" vary. An indicator derived from the ROC is the area under the curve (AUC). The more the area tends towards one, the better the prediction.

3.3.4.2 Coefficient of determination

The coefficient of determination, also called r-squared (r^2) is a measure on how much the model is performing better than the average of the data, and is computed as [36]:

$$r^2 = \left[\frac{\sum_i (y_i - \bar{y}) \left(\hat{y}_i - \bar{\hat{y}}\right)}{\sqrt{\sum_i (y_i - \bar{y})^2 \left(\hat{y}_i - \bar{\hat{y}}\right)^2}} \right]^2 \tag{3.14}$$

The possible outputs are between $]-\infty, 1]$, where 1 corresponds to a 100% accuracy of the model. This measure is used in the case of a linear output.

[9]Inspired from `http://commons.wikimedia.org/wiki/File:ROC_space-2.png`

3.3.4.3 Relative Root Mean Square Error

The relative root mean square error (RRMSE), is a measure of the difference between the observed and predicted data. The value is divided by the number of elements so that the number of data point does not influence the result, which allows for a comparison between sets of varying sizes. The value is also divided by the average of the observed data, so that the scaling factor does not matter anymore. Results with different amplitudes can be compared. The equation reads as follow:

$$\text{RRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}}{\bar{y}} \tag{3.15}$$

The possible readings are comprised between $[0, \infty[$, where 0 is the perfect fit between observed and predicted data.

3.3.4.4 Confusion Matrix

The confusion matrix is used in the case of a class forecast. It describes how the data can correctly predict a class. The computation is done in a similar way than for the ROC. The different predictions are compared

Table 3.3: Confusion Matrix Example as presented in Tuia [16]. Note that UA stands for "*User Accuracy*", PA for "*Producer Accuracy*" and OA for "*Overall Accuracy*". $n_{i,j}$ stands for the number of predictions labelled as i and with true label j.

		True Class				PA
		A	B	...	I	
Predicted	A	$n_{A,A}$	$n_{A,B}$	...	$n_{A,I}$	$n_{A,A}/n_{A,A..I}$
	B	$n_{B,A}$	$n_{B,B}$			$n_{B,B}/n_{B,A..I}$
	⋮	⋮		⋱		⋮
	I	$n_{I,A}$			$n_{I,I}$	$n_{I,I}/n_{I,A..I}$
UA		$n_{A,A}/n_{A..I,A}$	$n_{B,B}/n_{A..I,B}$	...	$n_{I,I}/n_{A..I,I}$	OA: $\sum_{i=A..I} n_{i,i}/N_{obs}$

against the true label of each class, and are summarized in a table (see table 3.3 for an example). The table can be summarized with two values per class: the *user* and the *producer* accuracy. The first term stands for the percentage of correct classifications for one class given all the predictions that have this class as true label. The second term stands for the percentage of correct classifications for one class given all the predictions that have been assigned this label. In table 3.3, this is resumed by each element of the diagonal divided by the sum of the corresponding column or row. The last value which can be taken from this table is the *Overall* accuracy of the prediction, with is the number of correct classifications given the total

number of observations. In table 3.3, this is the sum of the diagonal divided by the number of elements.

Overall, these different values give precious information on the accuracy with which each class is predicted.

3.3.4.5 Quantitative Assessment of the Confidence Interval

As explained in section 2.2, the Bayesian Neural Networks allow for a prediction with a confidence interval. As this is one of the main interest of Bayesian Networks, it is of importance to quantitatively assess the quality of this interval. The idea developed is to compute three values. Firstly, the

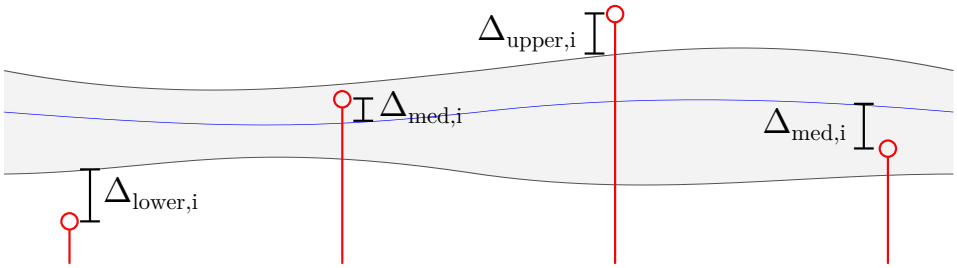

Figure 3.15: Schematic representation of the three characteristic dimensions of the confidence interval. The red dots represent the observations, the blue line the median, and the greyed surface the 95% confidence interval.

difference between the observations bigger than the confidence interval and the border. The second value is the difference between the median and the observations that are within the confidence interval, and the third value is the difference between the observations which are smaller than the lower boundary, and the boundary. The comptuation is done as follow:

$$\Delta_{upper} = \frac{\sqrt{\sum \left(\hat{y}_{97.5\%,i} - y_i\right)^2}}{\bar{y}}, \; y_i > \hat{y}_{97.5\%,i} \tag{3.16a}$$

$$\Delta_{median} = \frac{\sqrt{\sum \left(\hat{y}_{50\%,i} - y_i\right)^2}}{\bar{y}}, \; \hat{y}_{2.5\%,i} < y_i < \hat{y}_{97.5\%,i} \tag{3.16b}$$

$$\Delta_{lower} = \frac{\sqrt{\sum \left(\hat{y}_{2.5\%,i} - y_i\right)^2}}{\bar{y}}, \; y_i < \hat{y}_{2.5\%,i} \tag{3.16c}$$

where N is the total number of observations, and $\bar{y}$ the average over all observations. By dividing by the mean, the scaling of the observation does not influence the value too much. Figure 3.15 shows how the values are computed.

3.3.5 Inputs Choice

As specified in section 2.2.5.1, the relevance of the different inputs is done automatically. Knowing this, it is therefore interesting to use multiple inputs and let the algorithm decide whether the given input is relevant or not. In this optic the following inputs have been chosen:

- Temperature, humidity, solar radiation, wind speed and wind direction, using the mean, standard deviation, minimum and maximum for all of these observations.
- Wavelet transforms of the temperature, humidity and solar radiation.
- Time lag of 0, 1 and 2 days for each parameter.

A specific analysis on which input is necessary will be made in the discussion.

3.3.6 Training and Test Sets

The entire time-series has been divided in two sub-time series, a training an a test set. The choice has been made to take $2/3$ of the data for training and $1/3$ for the testing. Every third observation is therefore assigned to the testing.

3.3.7 Choice of Algorithm Parameters

As can be taken from section 2.2.3, as well as from figure 3.12, different parameters influence the way the weights are sampled:

- Number of samples.
- Number of steps for the Leapfrog.
- Step size in HMC.
- Number of samples in HMC.

To choose these constants, Neal [23] proposes to do different runs with the same number of "*supertransitions*", which is the number of steps multiplied by the number of samples in HMC. The total number of supertransitions stays the same, but the number of steps is varied, and the number of samples proportionally. The process is then repeated for different steps sizes, and the results are analysed on rejection rate, training and testing error, as well as spreading of the weights. Figure 3.16 shows such an analysis for the current experiment. The first analysis that can be drawn, is on the spreading of the parameters. The spreading is computed as the root mean square of the parameters. The larger the value, the larger the spectrum of the analyzed weights, which means that more different combinations are

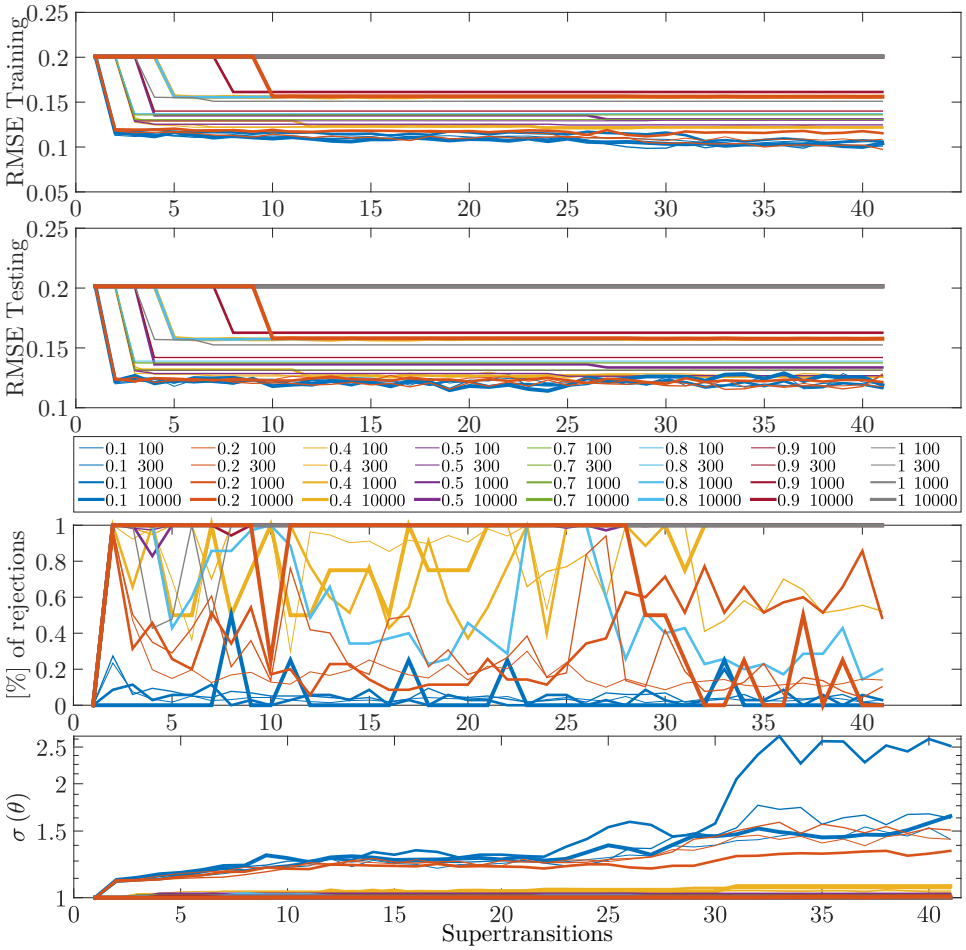

Figure 3.16: From top to bottom: RMSE on the training set, RMSE on the testing set, rejection rate and spreading of the parameters for a network with 10 nodes and a total number of (steps · hmc-samples) of 35'000. The legend indicates the step size (first value) and the number of leapfrog - steps (second value).

tried out. Small values which stay small the entire process do not display an efficient exploration of the state space. In figure 3.16, it is relatively clear that the only settings efficiently exploring the space are the ones using step sizes between 0.1 and 0.4.

The second interesting information is the number of rejections. The rejection is defined as explained in section 2.2.3.3, and the ratio is drawn from the number of samples:

$$ratio_{rejects} = {}^{\#\ rejections}/_{\#\ samples} \tag{3.17}$$

The higher the ratio, the fewer new samples are drawn. According to Neal [30], the ratio should be around 10 - 40%. Settings featuring a step-size of 0.1 or 0.2, and a number of steps around 100 - 300 seem to works best.

The last information that can be drawn is the value of root mean square of the training and test set. Of course, the lower this value, the better the fitting will be. The resulting plots seem to indicate that a too low number of steps might result in a bad exploration of the state space, and that a higher number is required.

All in all, it seems that a number of leapfrog-steps around 300, with steps sizes around 0.1 and 0.2 might be the best solution. During the experiment, this values are taken as starting point, but are then fine-tuned according to the values of rejection, RMSE and spreading explained here.

3.3.8 Convergence Analysis

It is important to assess the convergence of the Markov Chain to be able to use the resulting weights. In this framework, this is done as suggested in the toolbox presented before, by using an other toolbox, "*MCMC Diagnostics for Matlab*[10]". The toolbox has been developed by the same team, and is based on the analysis of the decorrelation time through autocorrelation of the chain. The results which are presented in next section have all been analyzed on convergence, and are only presented if they do converge. Additionally to that, a burn-in time is computed, which is the time the chain requires before the samples are uncorrelated from each-other. The burn-in samples are then removed from the weight-set.

[10]MCMC Diagnostics for Matlab - `http://becs.aalto.fi/en/research/bayes/mcmcdiag/`

Chapter 4 Results

This section presents the different results obtained. As several hundred runs with fine tuning of the parameters have been launched and processed, only the best results are presented here for each category.

4.1 Binary Forecast

As described before, the binary output is defined as the forecast of the presence or absence of rain. The best result obtained has an area under the

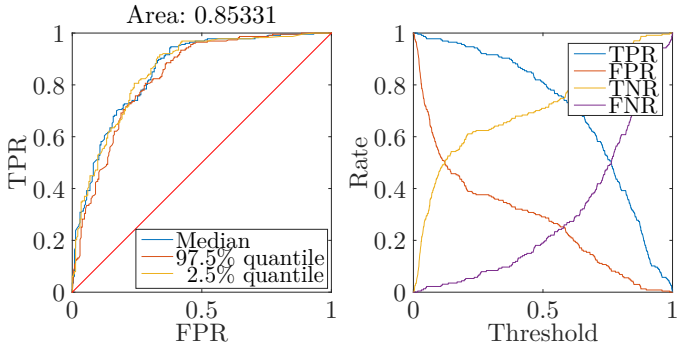

Figure 4.1: Receiver Operating Characteristic graph on the left, with area under the curve. On the right, the different rates are plotted against the thresholds, which allows to find the optimal threshold.

curve of 0.85 as displayed in figure 4.1. This figure also allows to obtain the optimal threshold value, as it is the intersection between the True Positive rate and the True Negative rate. For this case of study, the optimal threshold

Table 4.1: Confusion matrix for the binary output, for a threshold of 0.577. The table on the left is for training, and the table on the right for testing.

		Training		**Observed**	*Testing*		
		False	True		False	True	
Predicted	False	384	76	0.83	175	54	0.76
	True	84	376	0.82	59	173	0.75
		0.82	0.83	**0.82**	0.75	0.76	**0.75**

is found at approximately 0.577. Figure 4.2 shows the result of the forecast after applying a threshold of 0.577 to the probability output. As can be seen, the forecast is relatively well done during the dry-season, but seems to struggle with the high variability of the rain measurements during the rainy season. Table 4.1 shows the confusion matrix for the binary output. The

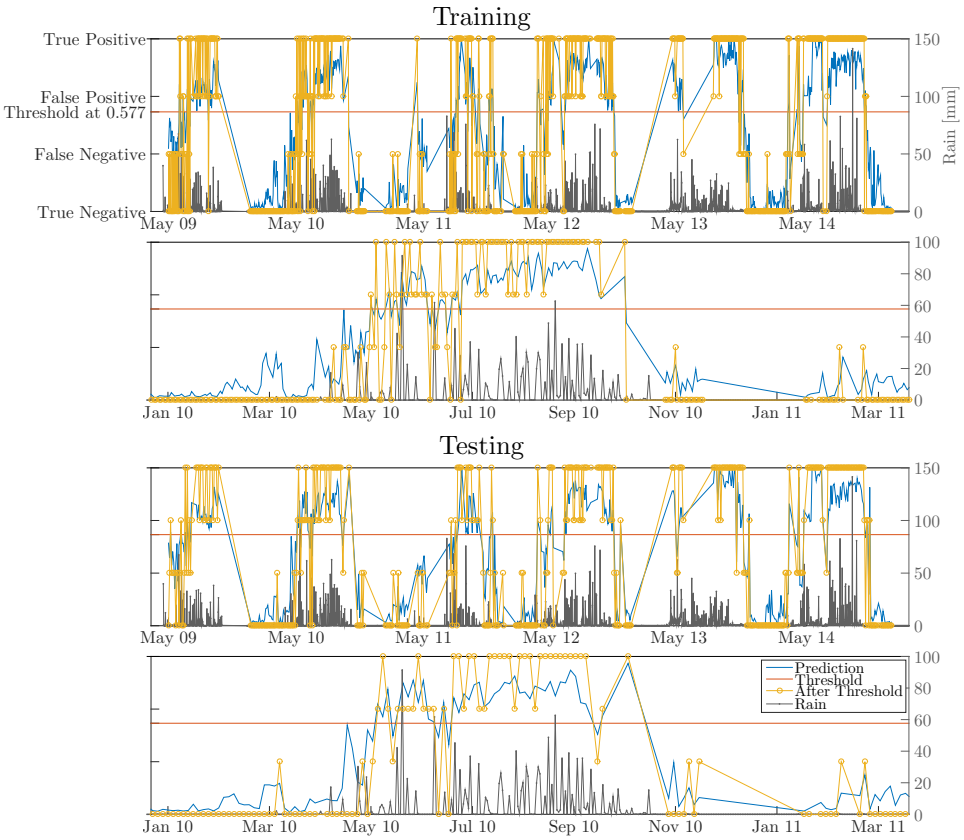

Figure 4.2: Binary forecast of the presence of absence of rain. The first two plots show the results for the training set, the other two for the test set. The first plot of each set shows an overview of the six years of data, the second plot show a closeup of one year.

two results of 0.82 and 0.75 for the training, respectively the testing set are quite satisfying. The majority of the false observations come from a miss-prediction during the rainy-season. To test the accuracy during this season, the confidence matrix is recomputed with the data from this period of time. By comparing table 4.1 with table 4.2, it is possible to see that most of the miss-predicted observations occur during the rainy-season. There are three observations which are predicted true, but should be false which are outside of the rainy-season. The rest of the error is related to observations occurring during this time. The overall number of false observations is drastically reduced, but the number of miss-predictions stays around the same, which

Table 4.2: Confusion matrix for the binary output, for a threshold of 0.577. The table on the left is for training, and the table on the right for testing. The data selected all occur during the rainy season.

		Training		**Observed**	*Testing*		
Predicted		False	True		False	True	
	False	101	66	0.60	42	42	0.50
	True	81	373	0.82	56	173	0.75
		0.55	0.85	**0.76**	0.43	0.80	**0.69**

leads to a much poorer prediction quality during the rainy season. Still the overall accuracy during the rainy season is around 69% for testing, which is not too bad. At least, the prediction on when rain actually occurs is relatively high, which is good. The other conclusion that can be drawn from these two matrices is the fact that the model is able to discriminate between rainy- and dry-season.

4.2 Intensity Forecast

The forecasting of a linear output has been separated in four different outputs: daily cumulative rain, weekly cumulative rain, forecast at one day and one week. This section shows the best results obtained for each configuration.

4.2.1 Daily Cumulative Rain

This section presents the results for the forecast of the Daily Cumulative Rain, the forecast of the quantity of rain fallen in one day. Figure 4.3 shows the result for such a forecast. As can be seen, the model grossly overfits the data, leading to a very poor test result. The tests have been done with fewer nodes, as well as fewer inputs, but the results are always either a very poor prediction on the training, as well as the testing set, or a complete overfit of the model. Table 4.3 shows the different statistics for the setup. As all the

Table 4.3: Summary of the statistics for the daily forecast.

τ $[days]$	Set	RRMSE	r^2	Δ_{Upper}	Δ_{Median}	Δ_{Lower}
1	tr	0.60	0.95	3.22	15.33	1.19
	ts	3.23	0.00	44.94	20.85	11.62

setups were performing very poorly, as well for the seven day forecast than for the one day forecast, only one result is shown here.

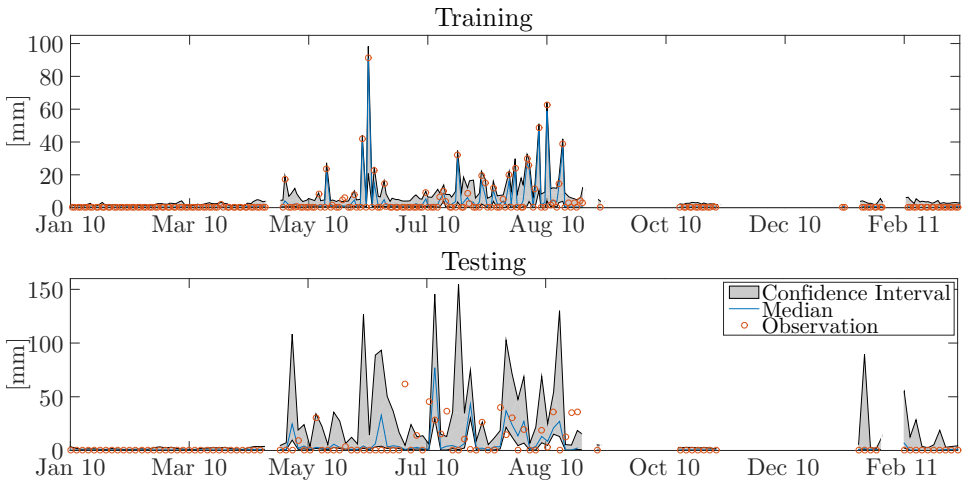

Figure 4.3: The plots show the rainfall forecasting of one day of cumulative rain for a year worth of data. The first plot shows the result for training, and the second for testing.

4.2.2 Weekly Cumulative Rain

The results for the weekly cumulative rain intensity forecast are separated in different setups: normal wavelets (transform over the whole time-series), causal wavelets, no wavelets, for a forecast time of one day, and for a forecast time of one week. For all the sets, the computation was stopped when the error for the testing set seemed to increase too much for too long. Table 4.4 shows the statistical results for the different configurations. As it can be

Table 4.4: Statistical summary of the different configurations for the weekly cumulative rain. "τ" stands for the time at which the predictions is done in the future, "tr" stands for training set and "ts" for testing set.

Wavelet	τ $[days]$	Set	RRMSE	r^2	Δ_{Upper}	Δ_{Median}	Δ_{Lower}
Normal	1	tr	0.13	0.99	0.08	3.67	0.13
		ts	0.57	0.79	1.63	6.68	4.28
	7	tr	0.30	0.94	1.50	7.39	0.83
		ts	0.74	0.65	5.02	5.61	6.90
Causal	1	tr	0.60	0.76	5.74	8.72	3.68
		ts	0.78	0.61	7.12	5.93	6.42
	7	tr	0.56	0.81	3.66	10.54	2.30
		ts	0.85	0.55	5.53	7.86	7.90
None	1	tr	0.87	0.52	16.17	5.29	8.07
		ts	0.95	0.43	11.94	2.62	9.86
	7	tr	0.72	0.66	5.37	11.08	3.48
		ts	0.94	0.44	8.92	5.31	8.30

seen, the best configuration is the one involving a normal wavelet transform performed over the whole data-set. The one setup involving causal wavelets is close to the best prediction. The configurations without wavelets clearly misses the mark. Generally speaking, the predictions seem of better quality for a one day in the future prediction, except for the cases without wavelets, which are around the same values. The normal wavelet transform seems to

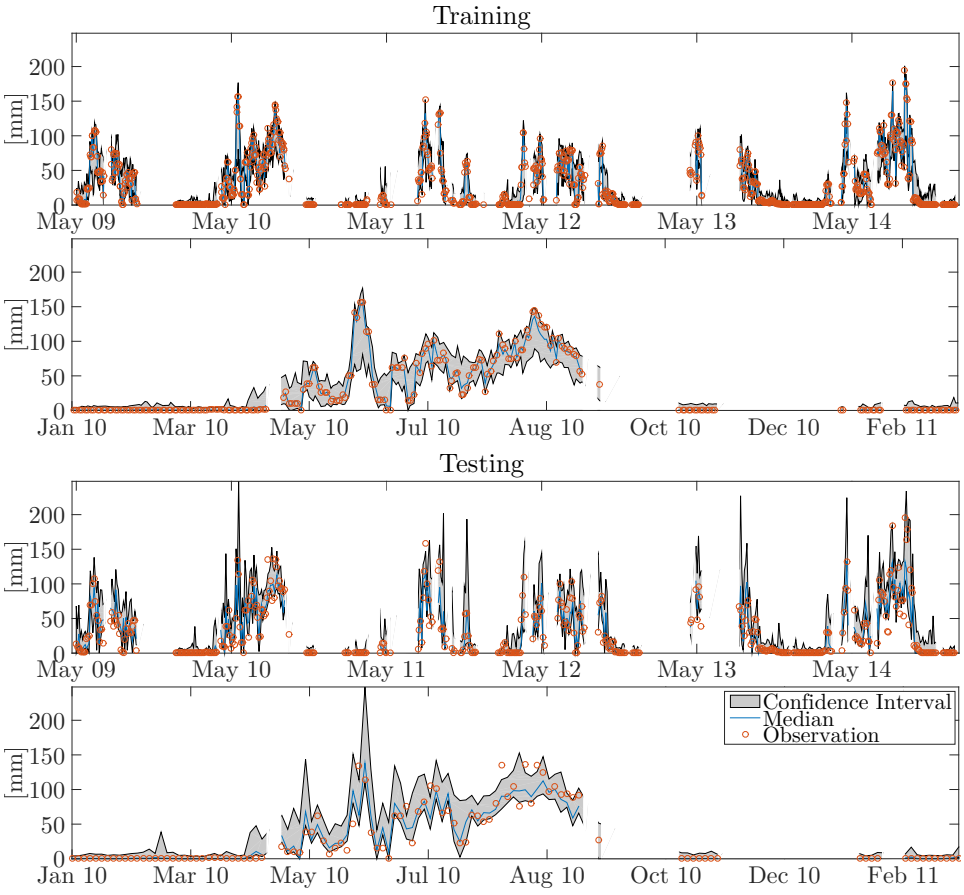

Figure 4.4: Results for the normal wavelet decomposition. As indicated, the first two plots show the results for the training set, while the two last show the result for the test set. The first plots from each set is an overview of the whole time-series. The second plot is a zoom on one year of the time-series which allows for a better view on the results.

be more able to embrace the observations in the confidence interval than the other configurations, as it can be taken from the different values of Δ. Generally the normal wavelet transform seems to produce a better fitting. The fitting for the causal wavelet on a one day prediction is acceptable, but it is to notice that the training set hasn't the same values than for the normal set. The set without wavelets seems to perform much worse than the

other sets, for both training and testing. Figure 4.4 shows graphically the results for the training and testing set for the normal Wavelet transform. The figure shows how the model seems to perform a good transition between dry- and rainy-season. Most of the observations for the six years seem to be explained by the model, within the confidence interval. The zoom on the year shows that the model nicely bounds the observations most of the time. As said before, the transition between the dry- and rainy-season seems to be very well modeled.

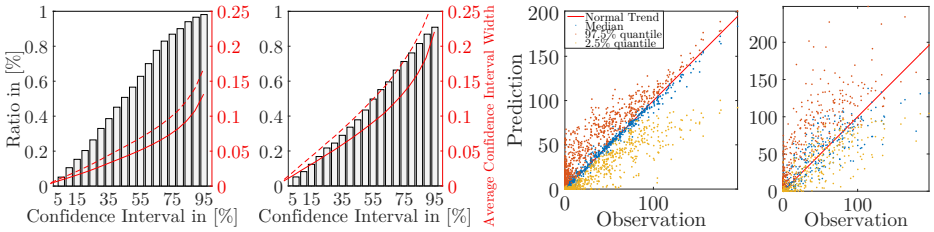

Figure 4.5: The first two plots show the percentage of observations that the given confidence interval (x-axis) can explain. The red curve shows the mean width of the confidence interval (in $[mm]$ of rain), and the dashed red curve the mean width but only during the rainy season. The first plot is for training, the second for testing. The third and fourth plots show the distribution of forecasted vs observed data, again for training and testing. The values are given in $[mm]$ of rain.

Figure 4.5 on the left shows the percentage of the observations that can be explained by an increasing size of confidence interval. It is interesting to see that the model can account for around 90% of the data for the training set with an average confidence interval width of around $0.25\,[mm]$ of rain. The dashed red curve indicates the size of the confidence interval when only the rainy-season is considered. It is interesting to see that the confidence of the model during this time is lower than when considering the whole time-series. The model seems to be able to predict with better accuracy the intensity during the dry season, when no rain occurs. The figure on the right shows the distribution of the observations when plotted against the modelled values for the median and the 95% confidence interval. Here the difference between a model that perfectly fits the data (the training set) and a model which performs a bit less well can be assessed. For the training set, the values of the mean of the model plotted against the observations nicely follow a 45° slope which indicates that the model performs fairly well. The confidence interval seems to be the most spread around observations with low, or no intensity of rain, which indicates that the difficulty resides in assessing the absence or presence of rain during the rainy-season. The confidence interval narrows towards higher values which indicates that the certainty of the forecast raises with the intensity of the rain. This can be associated to the fact that very different conditions arise during dry and

rainy-seasons, but both have periods without rain. The model has to account for these differences and find weights which can predict with certainty that there won't be any rain during a rainy season day, but as well during the dry-season.

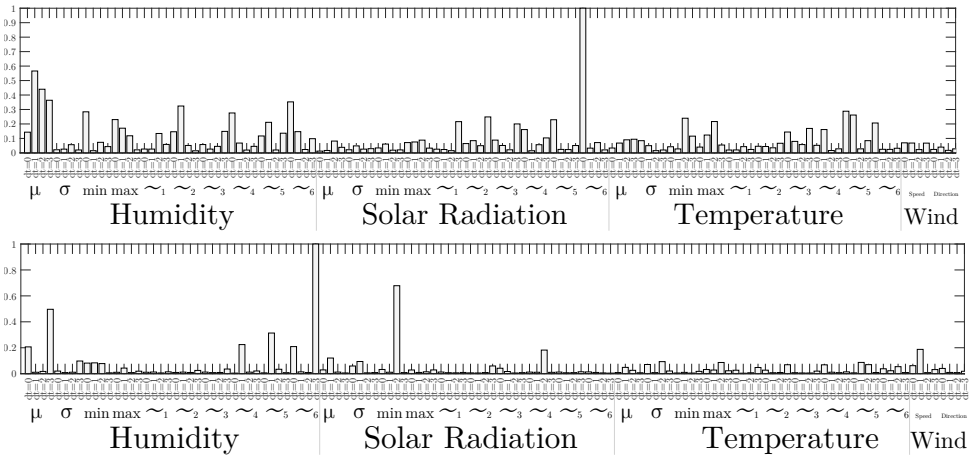

Figure 4.6: Figure showing the relative relevance of each input. The relevance is computed from the variance of the kernel density of the first layer of hyperparameters. The first plot shows the case of a one day prediction with all wavelets, the second with causal wavelets. The $\sim_n$ symbol represents the wavelet transform of level n, μ the average, and the σ symbol the standard deviation.

Figure 4.6 shows the relative relevance of each input for the two systems using wavelets (normal and causal) and forecasting on one day. As can be seen in the first plot, the main feature seems to be, by far, the solar radiation wavelet transform of level 6. The other important inputs are the mean humidity with lagged values of up to three days. This may indicate that the variation in humidity, as well as the long term variation in solar radiation may significatively improve the model, as they allow to understand the difference in atmospheric conditions.

The second plot shows the importance of the parameters for the causal wavelets case. Here the wavelet transform of level 6 with a time-lag of three days seems to make the difference, as well as the minimum solar radiation with a lag of two days. The first result confirms that inputs which can assess the changes in conditions seem important. The second parameter is harder to interpret, as the other time-lags for the same category are not deemed important.

To assess the importance of the other parameters, the simulations have been rerun with only the parameters exceeding a threshold of 0.2 relative relevance for the two scenarios forecasting on a one day time-lag. Table 4.5 shows the results of these re-runs. As can be seen, the results for the first system is approximately the same, which points in the direction that the

Table 4.5: Statistical summary of the runs with reduced number of inputs. The runs correspond to a threshold of 0.2 for the relative importance of parameters as presented in figure 4.6.

Wavelet	τ $[days]$	Set	RRMSE	r^2	Δ_{Upper}	Δ_{Median}	Δ_{Lower}
Normal	1	tr	0.42	0.88	3.64	7.69	2.50
		ts	0.61	0.76	3.85	5.02	6.01
Causal	1	tr	0.78	0.61	12.47	7.46	6.99
		ts	0.90	0.50	10.64	3.59	9.07

result strongly relies on these parameters.

The result for the second setup is further away from the setup with full range of parameters than the other result, which tends to show that the model with causal wavelets relies more on other parameters to fit the observations.

4.3 Rainfall Classes Forecast

The last possibility of forecast is the prediction of intensity classes in which the rain might be. The rain observations have been separated according to the description in section 3.3.3.3.

Table 4.6: Confusion matrix for the classes output. "*N.R.*" means "*No Rain*", "*R.C.#*" means rain class 1-5, which correspond to the values described in section 3.3.3.3.

		Observed					
		N.R.	R.C. 1	R.C. 2	R.C. 3	R.C. 4	
Predicted	N.R.	498	43	23	57	55	0.74
	R.C. 1	1	10	5	0	0	0.63
	R.C. 2	3	30	60	22	20	0.44
	R.C. 3	1	1	5	9	3	0.47
	R.C. 4	1	0	1	1	6	0.67
		0.99	0.12	0.64	0.10	0.07	**0.68**

Training

Predicted	N.R.	238	26	20	20	39	0.69
	R.C. 1	0	3	2	0	1	0.50
	R.C. 2	4	16	29	10	10	0.42
	R.C. 3	1	0	0	5	2	0.63
	R.C. 4	1	1	0	0	0	0.00
		0.98	0.07	0.57	0.14	0.00	**0.64**

Testing

Figure 4.7 shows the results for a rainfall class forecast. As can be seen, the forecast of the "*No Rain*" class almost always benefits from a higher probability than the other classes. The class forecast is done based on these probabilities, where the class with the highest probability is taken as the class being predicted. Table 4.6 shows the confusion matrix for the different

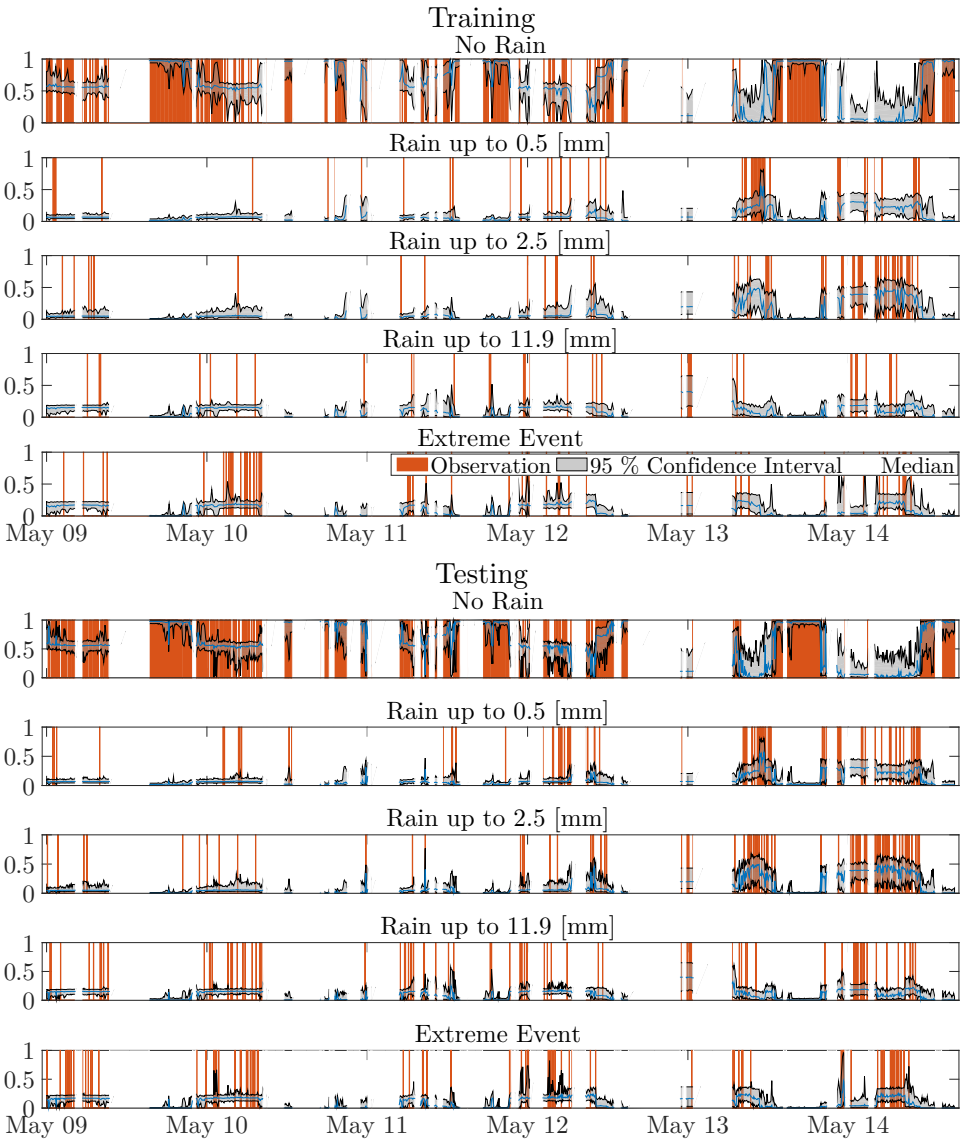

Figure 4.7: Training and testing results for the five classes of rain predicted. The orange bars show where the prediction should be assigned to this class. The prediction is assigned to the class displaying the maximum value for the median.

classes. As can be seen, and as it was already to be assumed from figure 4.7, the "*No Rain*" class has the most forecasted data, with also the most false

predictions which are assigned to this class. The other classes are predicted in a way worse manner, with the "*Extreme Event*" class exhibiting a 0% accuracy for the testing set. Globally the overall accuracy of around 60% can deemed relatively bad, especially since most of the correctly forecasted data is in the same class.

4.4 Testing of the Results

To assess the the model outside of the scope of training and testing, and to see if a the model can also explain rain with a completely different climatic setup, different tests are performed. The first test is to run the training on the same data-set, but with a different division of the two sets. As expressed before, the data has been divided in two by using every third observation for the testing set. To test the capability of the algorithm to train the model, the data has been subdivided in a different manner: the set consists of then consecutive measurements assigned for testing, followed by twenty measurements assigned to training, and so on. This allows to assess how much the training relies on a continuous time-series to perform a good prediction.

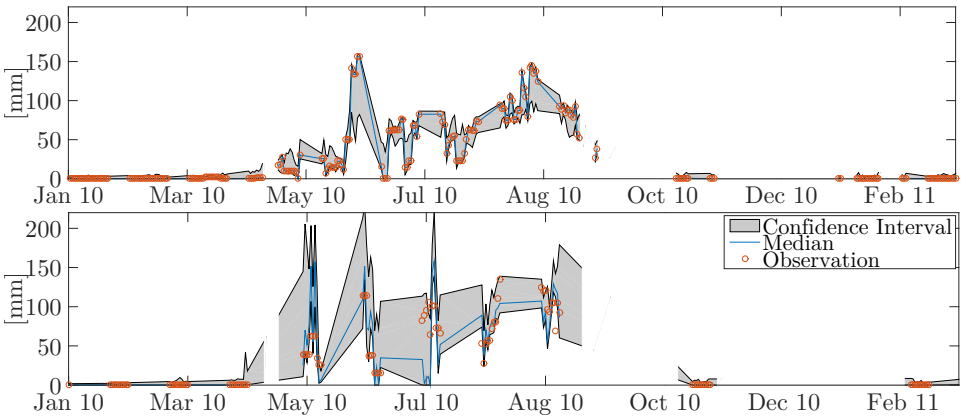

Figure 4.8: Zoom on one year of the result when training using 10-20 subdivision of testing and training. r^2 for training is around 0.99, and for testing around 0.15.

Figure 4.8 shows the result when training the model with this new division. As can be seen, the model has much more difficulty to predict the correct values for the testing set. The algorithm might rely too much on continuous data to be able to train the model with data that sparse, and produce a model which can explain the test set. The effect of this subdivision is that a large portion of the test set is not included in the model fitting, and this might remove relevant information for the training. In any case, due to this result, the forecasting capability of the best model might

have to be rethought.

The second test that is proposed is to apply the weights and configuration of the best model on a data-set outside of Tambarga, which has a completely different climatic setup. The model has been applied on data from Tougou (see 3.2), which is located in the North of Burkina Faso, and is subject to much higher temperature and lower rain intensities. Figure 4.9 shows the result from this experiment. As can be seen, the model does not predict the

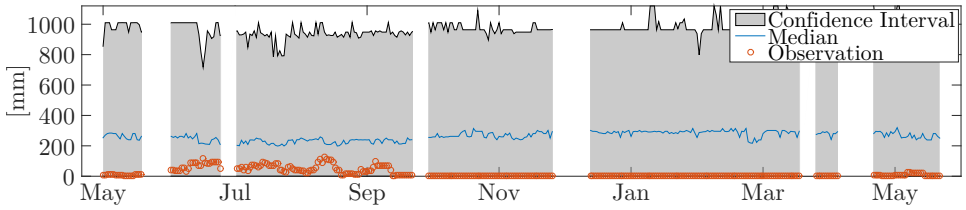

Figure 4.9: Best model applied to data from the location of Tougou for the year 2014 - 2015.

precipitation values correctly. The model is therefore subject to strong local conditions and not a generally valid physical model.

Chapter 5 Discussion

This section discusses the different results obtained and assesses the strength and weaknesses of the approach.

5.1 Overall

From the different results displayed in the previous chapter, the best working ones were the binary output, as well as the weekly windowed cumulative rain forecast. The daily rain forecast performed very poorly, which renders it unusable. The classes output seems to be unable to efficiently discriminate between rain classes, which makes it not very promising. The classes forecast might be subject to the same issue than the daily forecast; the high rain values variability between days might be of too high complexity for the model to accurately make a prediction. One might ask the question on why the binary output was able to mostly successfully forecast the absence or presence of rain. The main difference between the class, as well as the daily forecast, and the binary forecast is the range of data which has to be taken into account. In the binary forecast, there is no discrimination between an extreme event and a small rain event, so if a combination of parameters allows to generally explain the absence or presence of rain, the forecast can be of good quality. For both the classes forecast and the daily prediction, the task is much more complicated. The prediction has to account for all the variations. The daily forecast has an even greater burden, as it has to predict the exact value. The other problem with the classes forecast compared to the binary, is that the class "*No Rain*" has much more observations than all the other classes (see table 3.2). The training algorithm, in its eagerness to reduce the error, will be able to greatly reduce it by fitting the model to the "*No Rain*" class. This is why the confusion matrix displays such a high correctness for this class, and not the others. To overcome this problem, the algorithm should take into account the number of observations, by weighting the error proportionally to the number observations per class.

Generally, the discrimination between dry- and rainy-season seems to have worked well, as can be taken from the evaluation of the binary output. For the linear output, the size of the confidence interval only varies in a small manner when taking into account all the data or only the rainy-seaon,

which indicates that the error on the dry-season is really low, and therefore well modeled.

5.2 Solution Surface

The characteristics of the solution surface seem to matter a lot in the quality of the prediction. Figure 5.1 shows how the daily variability and the smoothness of the observations vary when the daily cumulled rain is taken, or when it is passed through a moving weekly filter, as used in this work. As

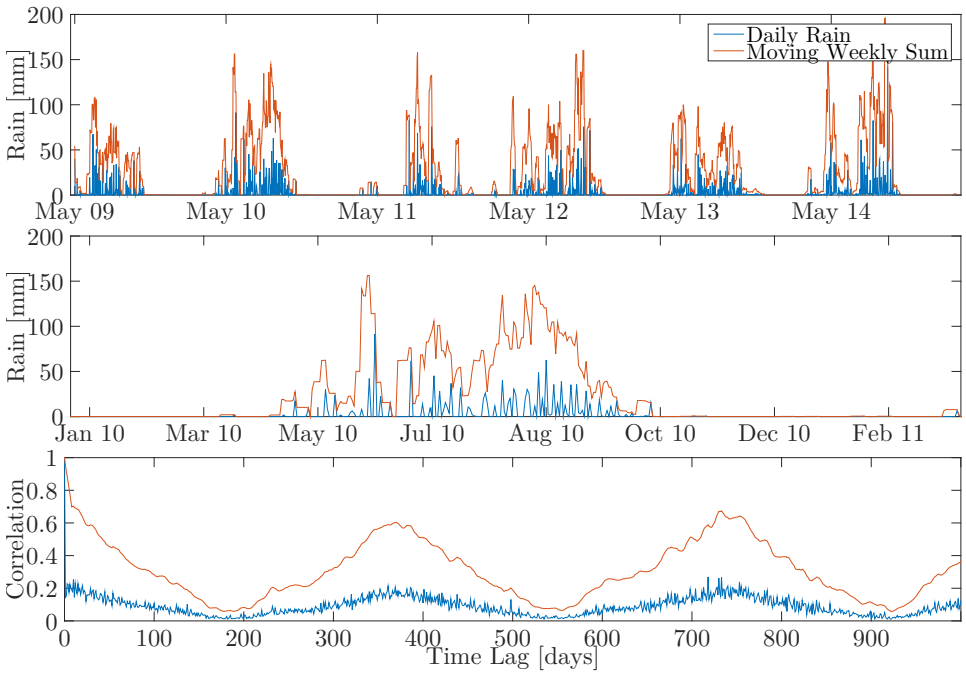

Figure 5.1: Plot of the daily rain and the rain passed through the moving summing window, overview and close-up. The third plots shows the autocorrelation of each of the two sets.

it can be seen, the data after filtering is much smoother, and the temporal variability much lower. The third plot shows the autocorrelation of both time-series. As shown, the daily values have a much higher variation at fine time-scales than the filtered, which indicates that the changes between days are much harsher than for the filtered data. Predicting a smoother surface seems definitely easier than a rough surface, as it can be seen from the results. Indeed, the lower variability in the observations makes it easier to express general trends as well as general assumptions about the data. The inputs can be better generalized by the net and therefore allow to express configurations as cases which have approximately already occurred, therefore allowing a prediction of much better quality.

The scaling of the inputs as well as the observations using normalization has shown a significant improvement in the model accuracy. The other modification of the solution space introduced previously (logarithm, etc.) have not shown particularly better results. These transform only rescale the solution space, which means that they do not allow for a smoother solution space. The normalization has helped because it brings the different time-series in the same range.

5.3 Input Relevance

Generally speaking, as stated in Neal [23] and Partal et al. [19], the more inputs, the better the result. As can be taken from the runs with reduced number of inputs (see table 4.5), the quality of the result is not quite as satisfactory as the results with all the parameters, even if close. The additional parameter seem not to add unwanted noise in the model, but rather explain small variability in the solution surface. By comparing figure 4.4 with figure A.1, it can be seen that the setup using fewer inputs has a harder time modelling the peaks, as well as the succession of low and high values. The assumption can therefore be made that by using more inputs, the extremes during the rainy-season are better modeled. This is confirmed when comparing the case of causal wavelets, figure A.3 with figure A.5, where the succession of low and high values is better modeled with more inputs. The fact that the causal wavelets with fewer inputs is unable to successfully model the extremities of the distribution is confirmed in figure A.6, where it can be seen in the scatter plots that the values are grandly modeled as mid-values, and the lower and higher values mostly ignored. The histograms on the right, showing the confidence interval quality shows that the interval needs to get relatively large until around 60% of the data can be explained, which confirmes the previously made hypothesis.

5.3.1 Wavelets

As suggested in the results, the usage of wavelets seem to significantly improve the performance of the model, at least for the weekly cumulated data. The wavelet transform seems to efficiently capture the changes in atmospheric conditions and translate them into the model. When comparing figure 4.4 (normal wavelet transform) and A.3 (causal wavelet transform) to figure A.7 (no wavelet transform), the inability to move between high and low predictions seem to rise again. In the case of no wavelets, the predictions are all in-between the minimum and the maximum of the observations, without being able to capture the alterations during the rainy-season. The wavelet transform seems to be able to express these variations. By comparing the figures, an other statement can be made, which is that the model performs significantly better during the dry-season when wavelets are used.

Again, this can be taken back to the fact that the wavelets allow to capture the variations in climatic conditions and therefore "understand" that there can not be rain during the dry-season.

There is still a gap in performance between the normal and causal wavelet transform. It would be interesting to improve the quality of the causal wavelet transform to improve the quality of the model.

5.4 Overfitting, Relevance of the Prior and Automatic Relevance Determination

As seen in the results, the prediction of precipitation for one day of cumulative rain overfits grossly. As described previously, this might be because of the complexity of the underlying non-linear phenomena. The precipitation might be so complex to predict, that only certain combinations of weights and inputs exactly explaining the observation produce an adequate result. These setups, working perfectly for the training set, do not express a general model valid for more than this set, leading to gross overfitting.

As mentioned before, Nourani et al. [21] stated that the wavelet might not significantly improve the results for a weekly or daily forecast. The results show that they were right for the daily forecast, but the wavelets definitely made a huge difference for the weekly windowed case, even when only using causal wavelets.

Neal [23] states that overfitting should not occur, except for a bad choice of priors. As explained in Lampinen et al. [26] and Vehtari et al. [25], selecting an appropriate prior requires a lot of experience or a high understanding of the underlying processes, both of which the author might not have. The experiments have been run with different priors, which includes large non-informative priors (priors with high scale and standard deviation). When using large enough priors, the model did not overfit anymore, but the sampling did not produce any interesting outputs, as the weights continuously change without any improvement. With large priors, the degree of liberty of the weights might be too big to find appropriate solutions to the problem, and the sampling wanders around in the solution space without finding any acceptable solution, which might correspond to a random walk behaviour.

Automatic Relevance Determination, as well as using two level of hyperparameters for the error modeling described in Lampinen et al. [26], made a huge difference in the quality of the prediction for the weekly cumulative sum, as well as the binary prediction. The model was of very poor quality prior to introducing these concepts, and the values obtained in the results section all use these techniques. ARD and the error hyperparameters allow to deal with large quantity of inputs without having to worry about the relevance of the inputs, as they are autoregulated.

5.5 Size of the Net

The size of the net has not made a big difference as the weights multiplying the outputs of the hidden nodes are also regulated by hyperparameters. The size of the net greatly negatively impacts the computational time, because for each node, $(\#_{inputs} + \#_{outputs} + 1)$ new weights are added to the net. For this reason, nets with a small number of hidden nodes are preferred.

The number of weights does not either seem to be the cause of overfitting, as greatly reducing it does not lead to better results for the linear output. On the other hand, for the classes output, limiting the number of nodes to a low value (5-10) seemed to produce better results.

5.6 Forecasting Capability and Physicality of the Model

With regard to the results presented in section 4.4, the forecasting capability of the model has to be discussed. The question that arises is whether the model actually forecasts the precipitation in the test set, or if these values are just found by chance. In the training process, the data used for testing is not "*seen*" by the algorithm. The algorithm therefore trains the model to the best of its possibilities, regardless of whether the test set is actually respected or not. The weights are therefore optimized for given situations, or characteristical combinations of inputs. The training algorithm operates the training regardless of the fact that the data is given as a time-series, except for the use of lagged values and wavelet transforms. To the algorithm everything is just input and corresponding output. The fact that the model performs well on the training set is therefore enough to justify the forecasting capability of the net.

The question then arises to why the net performs significantly worse in the new selection of training set. The training algorithm optimizes its weights so that it works for the set it is allowed to use. The problem of this approach, is that the model can only perform well on situations that are known. Situations that have no statistical similarity to other setups, and climatic configurations which are unknown to the model will perform significantly worse, as they are out of scope of the model. If a year of data outside of training were completely different from the other years, the model would maybe be able to show the general trend, but not the exact rainfall. Additionally to that, using a test set which has observations that are that much apart, the data are not representative of the overall processes, which does not allow the model to be well trained. The model assumes that the input and output data do not change much outside of the scope of the training, and are therefore valid for situations which are similar. The problem is that in the present case, the climatic parameters vary much in

space and time, which is one reason why the best model performs very poorly on the data from Tougou. The model is optimized and valid for local data, with local physical phenomena, not globally for global phenomena. It would be interesting to train a net on data from different locations to predict spatially distributed data.

Chapter 6 Conclusion

This chapter serves as conclusion to the project, discusses the achievements, the lows and the highs of the implementation and provides an outlook into the possible future work that could be done. The last part is a personal conclusion from the author on the project.

6.1 General

The work achieved provides a good introduction and background to enable a future use of Bayesian networks in the context of the prediction of local rainfall, and later of snails for the fight against schistosomiasis. The potential of the binary output is relatively high, as it allows to forecast with a satisfying certainty if rain is going to fall on the next day. The weekly cumulative rain output as well provides some insight in the precipitation forecast. When combining both outputs, it might and should be possible to make a decision on when to intervene on the snail population.

The potential of the Bayesian Neural Networks in terms of uncertainty has not been used to its full extent when binary forecasts were done. It would be interesting to develop a method to include this in the binary decision.

The goal to exactly predict the daily precipitation one day in the future has not been achieved, but as already stated, when combining the two working outputs, this can give good indications on future precipitation.

The forecast provided a good discrimination between dry- and rainy-season, as most of the error observed originated from the rainy-season. This has mostly been achieved by using the wavelet transform. Wavelet transform is also one of the reason for the good forecast on the binary output, as well as on the weekly cumulative rain.

6.2 Outlook

The quality of the forecast still needs to be verified in a live implementation. For this, it could be interesting to train the network on all available data, and to forecast one day at a time, renewing the training every day. For classical ANNs, a technique called on-line learning has been developed, where the

model is constantly adapted to the new inputs that are given, as opposed to batch-learning, which is the way it is done in this framework. This way, a model which has only to be valid for the next day could be developed, which might be easier to train correctly than a model valid for six years. The prediction would then be: the probability, or intensity, of rain for next day knowing all the data available.

As stated before, wavelets did improve the quality of the forecast significantly, and the normal wavelet transform even more. A next step to be able to extend the power of normal wavelets to causal wavelets would be to model the parameters used for precipitation forecasting (temperature, solar radiation, humidity, etc.), and then to perform a wavelet transform on these new data. This could eventually reduce the border effect, but it might also be that the error on the parameter estimation propagates to the wavelets transform, and therefor introduces a new error in the precipitation model.

A further development that could be made is to include other parameters in the data set. At some stations, especially the more recent ones, the pressure is for instance measured. It would be interesting to include this parameter in the model, as precipitation is strongly related to pressure. Variations in the use of the same parameter could then possibly be removed, for instance only use the mean and standard deviation of each parameter.

For the classes output, it would be interesting to implement a condition on the number of observations per class so that the model does not try to only fit this class.

For the linear output, it might interesting to test other time-scales, like hourly precipitation, or an even higher time-resolution, as the transition between rain- and no-rain-states might be smoother. A problem with this approach might then come from even bigger gaps in the time-series, which are already a problem at the currently used time-scale.

As soon as enough data has been gathered, it would be interesting to perform the same experiences on the data from the other locations. As the climatic conditions vary a lot, it would be interesting to see the performance of the different forecasts in this situation. Once this is done, the natural next step would be to forecast in a spatially distributed manner, over several locations at the same time.

6.3 Personal Conclusion

In the short time allocated to this project (4 months), I grasped at the surface of understanding the full extent of these incredible tools which are Bayesian Neural Networks. The power and applications of these techniques are incredible and very large if well implemented. Getting to understand these concepts was a real struggle, but in the end it is very satisfactory to get to further understand a technique in machine learning as powerful as

Bayesian Neural Networks. Getting to work with these machine-learning tools on environmental problems is something I wanted to be doing for a long time. The use of time-series and real-world data in this context brings a right amount of challenges to the table, as problems like missing values, discontinuity in the time-series, error in the measurements etc. have to be dealt with which are usually less part of the problem in robotics or when using synthetic data like I was used to. But the harder the challenge, the greater it feels when the techniques applied seem to start to work. The time I was involved in this project was very pleasant, and I enjoyed every part of it.

Bibliography

Water Resources Engineering and Hydrology

[1] J.N. Poda, B. Sellin, and L. Swadago. "Dynamique des populations de *Bulinus senegalensis* Müller 1781 dans une mare temporaire située dans une zone climatique nord-soudanienne au Burkina Faso". fre. In: *Revue d'élevage et de médecine vétérinaire des pays tropicaux* 47.4 (1994), pp. 375–378. ISSN: 0035-1865. URL: http://cat.inist.fr/?aModele=afficheN\&cpsidt=3508258.

[2] J.N. Poda, L.L. Sawadogo, Bertrand Sellin, and S. Sanogo. "Dynamique des populations de *Bulinus truncatus rohlfsi Clessin*, 1886, dans le barrage de Dyoro en zone nord soudanienne du Burkina Faso". In: *Agronomie Africaine* 8.1 (1996), pp. 61–68. ISSN: 1015-2288. URL: http://www.documentation.ird.fr/hor/fdi:010008056.

[3] Natalie Ceperley, Alexandre Repetti, and Marc Parlange. "Application of soil moisture model to Marula (Sclerocarya birrea): Millet (Pennisetum glaucum) agroforestry system in Burkina Faso". In: *Technologies and Innovations for Development*. Springer Paris, (2012), pp. 211–229.

[4] Christine Wiedmann. "Variability of Rainfall in a Semi-Arid Catchment in Burkina Faso". (2012).

[5] Theophile Mande. "Hydrology of the Sudanian Savannah in West Africa, Burkina Faso". eng. PhD thesis. Lausanne: ENAC, (2013). DOI: 10.5075/epfl-thesis-6011.

[6] Margaux Couttet. "SIE – Project: Precipitation Pattern Analysis and Inter Villages Comparison in Burkina Faso". 2015.

[7] Armel T. Kaptué, Niall P. Hanan, Lara Prihodko, and Jorge A. Ramirez. "Spatial and temporal characteristics of rainfall in Africa: Summary statistics for temporal downscaling". In: *Water Resources Research* (2015). ISSN: 1944-7973. DOI: 10.1002/2014WR015918. URL: http://dx.doi.org/10.1002/2014WR015918.

[8] T. Mande, N. C. Ceperley, G.G. Katul, S. Tyler, H. Yacouba, and M. B. Parlange. "Suppressed convective rainfall by agricultural expansion in southeastern Burkina Faso." In: *Water Resources Research* (2015).

General Statistics and Data Analysis

[9] François Chaplais, Panagiotis Tsiotras, and Dongwon Jung. "Redundant wavelet processing on the half-axis with applications to signal denoising with small delays: theory and experiments". In: *International Journal of Adaptive Control and Signal Processing* 20.9 (2006), pp. 447–474.

[10] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[11] Bernard Cazelles, Mario Chavez, Dominique Berteaux, Frédéric Ménard, Jon Olav Vik, Stéphanie Jenouvrier, and Nils C Stenseth. "Wavelet analysis of ecological time series". In: *Oecologia* 156.2 (2008), pp. 287–304.

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, (2009).

[13] Todd C. Headrick. *Statistical simulation: power method polynomials and other transformations*. CRC Press, (2009).

[14] Edwin Olson. "On computing the average orientation of vectors and lines". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. (2011), pp. 3869–3874.

[15] Patrick Thiran and Martin Hasler. "Course notes on "Dynamical Systems for Engineers"". (2014).

[16] Devis Tuia. "Course notes on "Imagery of Territory"". (2014).

Artificial Neural Networks Applied to Hydrology

[17] ASCE Task Committee et al. "Artificial neural networks in hydrology. I: Preliminary concepts". In: *Journal of Hydrologic Engineering* 5.2 (2000), pp. 115–123.

[18] ASCE Task Committee et al. "Artificial neural networks in hydrology. II: hydrologic applications". In: *Journal of Hydrologic Engineering* 5.2 (2000), pp. 124–137.

[19] Turgay Partal and H Kerem Cigizoglu. "Prediction of daily precipitation using wavelet–neural networks". In: *Hydrological sciences journal* 54.2 (2009), pp. 234–246.

[20] R Venkata Ramana, B Krishna, SR Kumar, and NG Pandey. "Monthly rainfall prediction using wavelet neural network analysis". In: *Water resources management* 27.10 (2013), pp. 3697–3711.

[21] Vahid Nourani, Aida Hosseini Baghanam, Jan Adamowski, and Ozgur Kisi. "Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review". In: *Journal of Hydrology* 514 (2014), pp. 358–377.

Bayesian Neural Networks

[22] Radford M. Neal. "Probabilistic inference using Markov chain Monte Carlo methods". PhD thesis. (1993).

[23] Radford M. Neal. "Bayesian Learning for Neural Networks". In: Lecture Notes in Statistics 118. Springer New York, (1996). ISBN: 978-0-387-94724-2, 978-1-4612-0745-0.

[24] Kenneth M. Hanson. "Tutorial on Markov Chain Monte Carlo". In: *Workshop for Maximum Entropy and Bayesian Methods*. (2000), pp. 9–13.

[25] Aki Vehtari, Simo Sarkka, and Jouko Lampinen. "On MCMC sampling in Bayesian MLP neural networks". In: *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. Vol. 1. IEEE. (2000), pp. 317–322.

[26] Jouko Lampinen and Aki Vehtari. "Bayesian approach for neural networks – review and case studies". In: *Neural networks* 14.3 (2001), pp. 257–274.

[27] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. "An introduction to MCMC for machine learning". In: *Machine learning* 50.1-2 (2003), pp. 5–43.

[28] Radford M. Neal and Jianguo Zhang. "High dimensional classification with Bayesian neural networks and Dirichlet diffusion trees". In: *Feature Extraction*. Springer, (2006), pp. 265–296.

[29] Jarno Vanhatalo and Aki Vehtari. "MCMC methods for MLP-network and Gaussian process and stuff – a documentation for matlab toolbox MCMCstuff". In: *Laboratory of computational engineering, Helsinki university of technology* (2006).

[30] Radford M. Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* 2 (2011).

[31] Nando de Freitas. "Lecture on Machine Learning, University Of British Columbia". `http://www.cs.ubc.ca/~nando/540-2013/lectures.html`. (2013).

[32] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, (2013). ISBN: 9781439840955.

Bayesian Neural Networks Applied to Hydrology

[33] G. Humphrey, M. Lambert, and H. Maier. "Development of stochastic artificial neural networks for hydrological prediction". In: *International Congress on Modelling and Simulation (15th: 2003: Townsville, Queensland)*. 2003.

[34] Greer B. Kingston, Martin F. Lambert, and Holger R. Maier. "Bayesian training of artificial neural networks used for water resources modeling". In: *Water Resources Research* 41.12 (2005).

[35] Mohammad Sajjad Khan and Paulin Coulibaly. "Bayesian neural network for rainfall-runoff modeling". In: *Water Resources Research* 42.7 (2006). W07409, n/a–n/a. ISSN: 1944-7973. DOI: 10.1029/2005WR003971. URL: http://dx.doi.org/10.1029/2005WR003971.

[36] Greer Bethany Kingston. "Bayesian artificial neural networks in water resources engineering." PhD thesis. The University of Adelaide, 2006.

Annexes

The annexes contain different figures and tables which were not included in the main report to improve its readability.

The first table presented shows the different parameters used for training for the different setups.

Table A.1: Summary of the different setups used for the experiments. "*b*" stands for binary, "*l*" for linear, "*lc*" for linear with weekly cumulative rainfall, "*c*" for classes, "*N,C,No*" for Normal, Causal, respectively No Wavelets, "*S*" for samples, and "*L-steps*" for leapfrog-steps.

Case			S	S HMC	L - Steps	Step-size	Nodes
Output	Wavelet	τ					
b	N	1	85	100	400	0.2	10
l	N	1	121	20	300	0.1	10
lc	N	1	134	30	500	0.2	10
lc	N	7	221	20	300	0.14	10
lc	C	1	241	117	300	0.2	7
lc	C	7	181	20	300	0.15	10
lc	No	1	221	20	300	0.15	10
lc	No	7	421	20	300	0.15	10
c	N	1	122	50	300	0.14	5

All the values were drawn from the results obtained and explained in section 3.3.7.

The following figures all come from the implementation of the weekly cumulative rainfall, with a time forecast τ of one day. The different setups are explained in the caption of each figure.

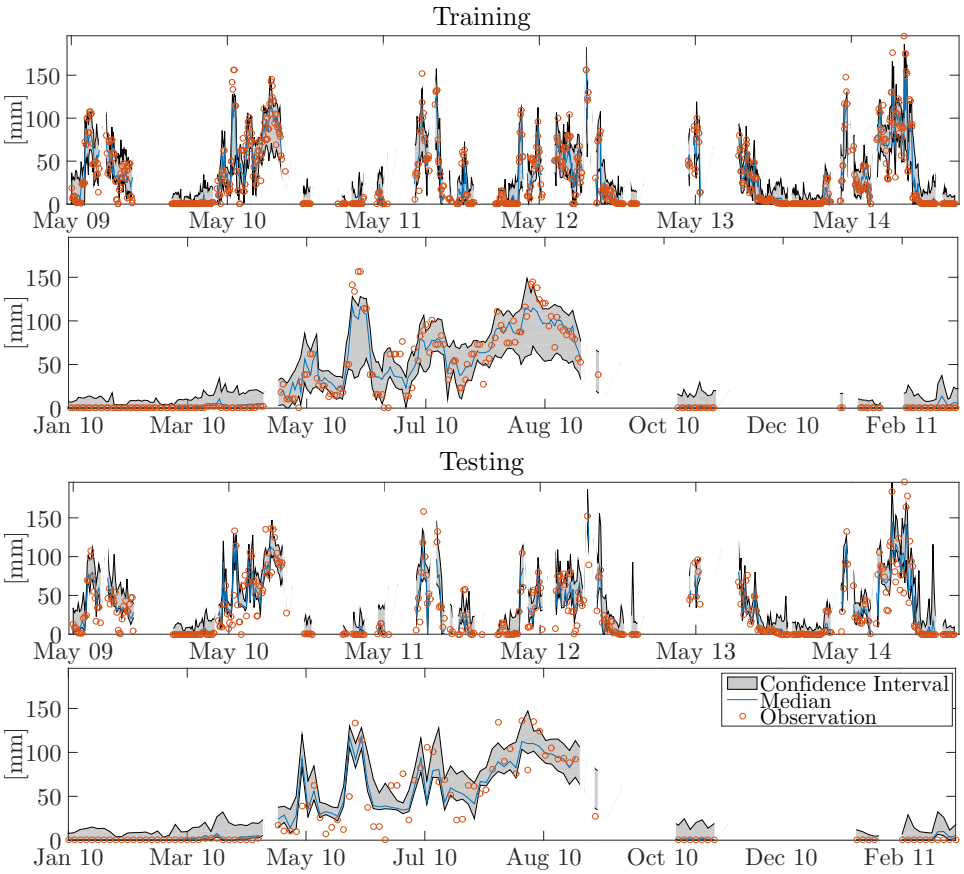

Figure A.1: Results for the normal wavelet decomposition after variable selection.

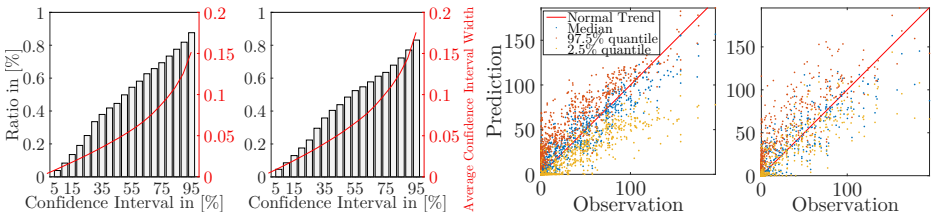

Figure A.2: Error plots for the normal wavelet decomposition after variable selection.

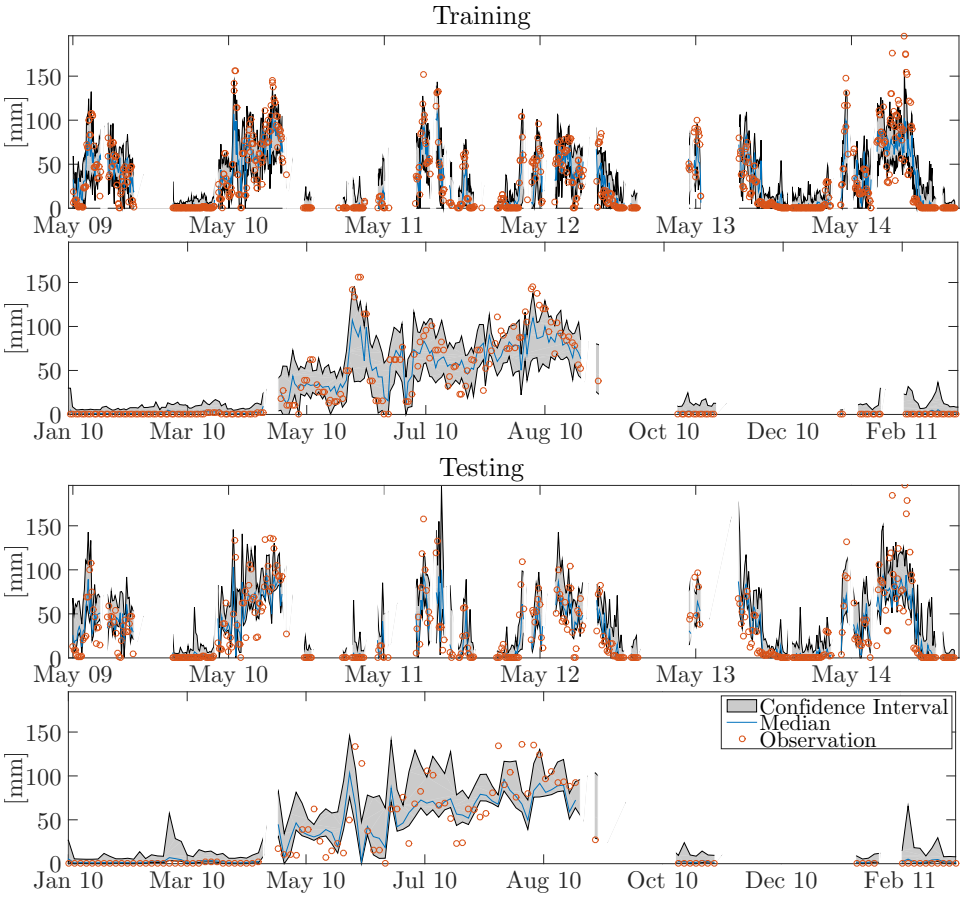

Figure A.3: Results for the causal wavelet decomposition.

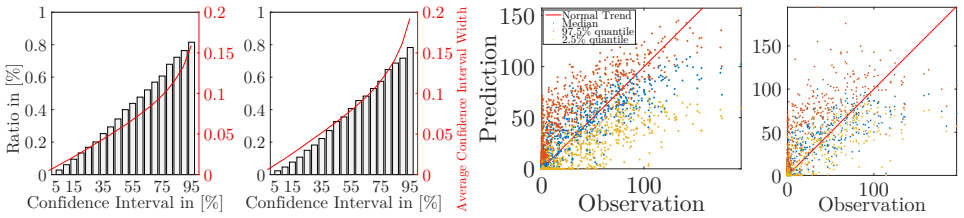

Figure A.4: Error plots for the causal wavelet decomposition.

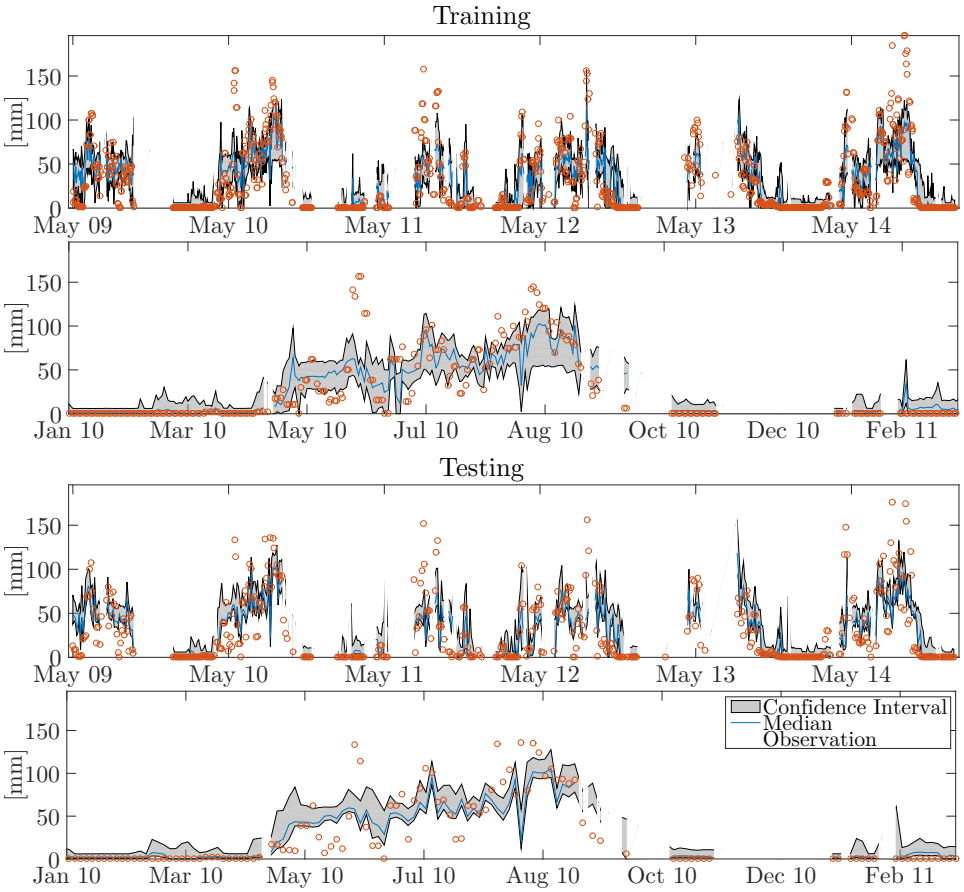

Figure A.5: Results for the causal wavelet decomposition after variable selection.

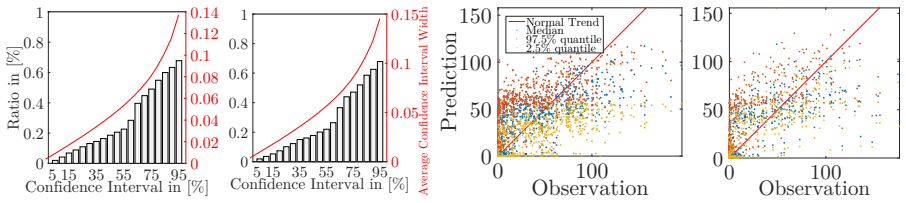

Figure A.6: Error plots for the causal wavelet decomposition after variable selection.

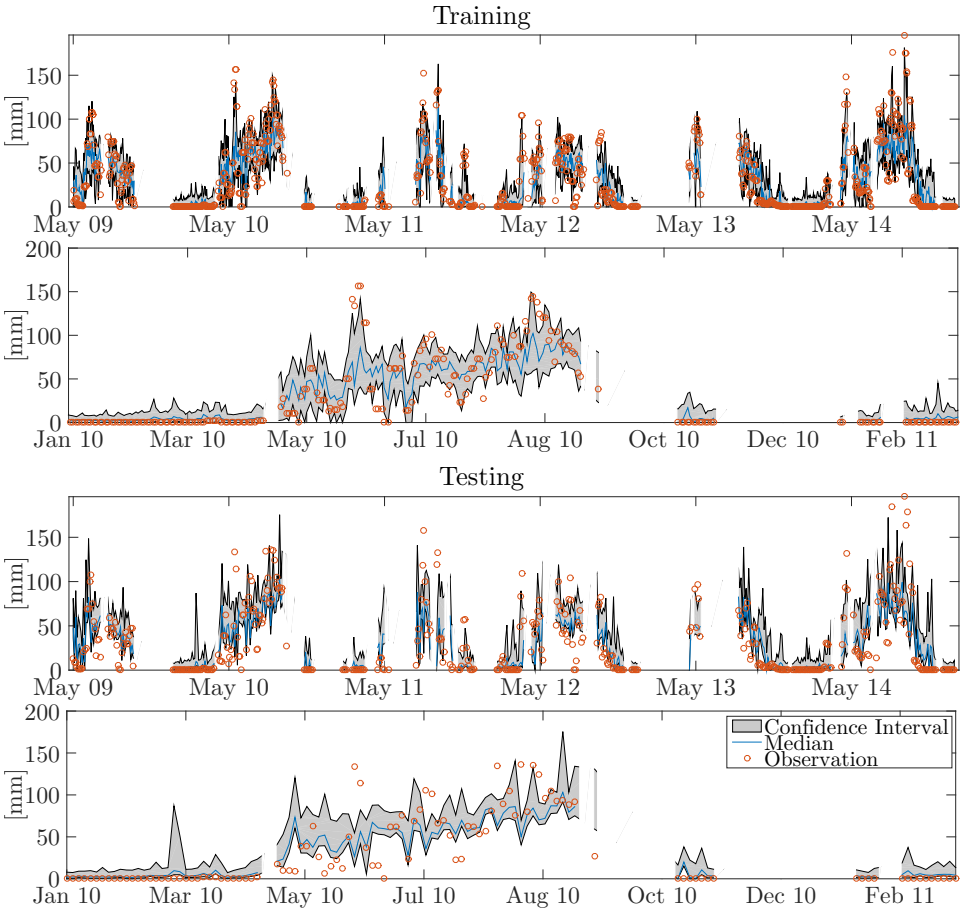

Figure A.7: Results for the case without wavelets.

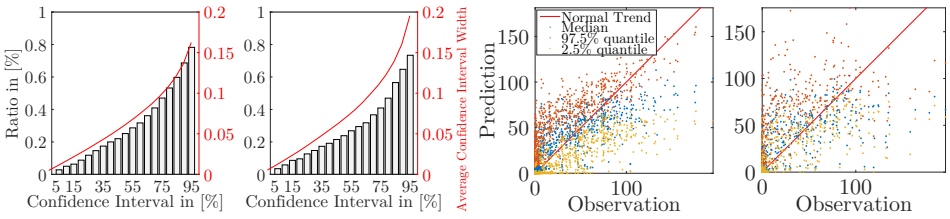

Figure A.8: Error plots for the case without wavelets.