Data Driven Modelling of a Dam System Discharge

Paulo Henrique Pinto Rocha

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Dr. Lígia Laximi Machado de Amorim Pinto
Prof. Susana Margarida da Silva Vieira

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Dr. Lígia Laximi Machado de Amorim Pinto
Member of the Committee: Prof. Miguel Afonso Dias de Ayala Botto

January 2021

To all who made me arrive here...

Acknowledgments

First, I would like to express my gratitude to my supervisors Lígia Pinto and Susana Vieira for all the advice and guidance in this work, that made it possible.

I would also like to acknowledge Augas de Galicia for the data used for this study and project HazRunoff (UCPM-783208) supported by the Directorate-General for European Civil Protection and Humanitarian Aid Operations.

I would like to thank my parents and grandparents who gave me all the support to make this dream a reality, they gave everything I needed and more. To my friends for all the moments that made these years memorable.

Last but not least, a special thanks to Bárbara Ribeiro for all the support and patience.

Resumo

No estudo de bacias hidrográficas é frequente a utilização de modelos computacionais físicos. Contudo estes modelos não têm em consideração elementos não naturais (e.g. barragens) que podem alterar o curso de água natural. A previsão do caudal de saída de uma barragem requer a utilização de modelos que tem por base dados. Neste estudo são propostas duas abordagens para prever o caudal de saída de um sistema de barragens localizado no rio ulla (Espanha). Um abordagem linear com base na modelação de series temporais e outra não linear com uso de redes neuronais.

Neste estudo são utilizados dados diários, entre 2013 e 2018, de um sistema constituido por três barragens para produção elétrica. O primeiro modelo tem como objetivo prever a entrada de agua na primeira barragem com base em variáveis metereológicas e de valores de caudal obtidos por estações localizadas no rio. Cada barragem foi simulada com um modelo para a descarga e outro para o armazenamento.

Para as redes neuronais, dois algoritmos foram testados tendo a sua estrutura sido otimizada através do ajuste de hiperparâmetros com recurso a otimização bayseana. As entradas para cada modelo foram selecionadas através de testes com os modelos não lineares.

Os modelos foram comparados, e os melhores modelos foram testados em serie e validados em 330 dias. A comparação dos resultados dos vários modelos propostos mostra que o melhor resultado foi obtido para redes neuronais. Este resultado demonstra que este tipo de modelos pode ser aplicado com sucesso a sistemas de barragens para a previsão de caudal de saida.

Palavras-chave: Previsão de descarga, Simulação de reservatório, Aprendizagem automática, Previsão armazenamento de barragens

Abstract

In the study of river basins conceptual computational models are ubiquitous, however they cannot simulate unnatural elements which change the flow of water. Dams introduce human containments which require data driven modelling to forecast. In this study two types of modelling are proposed to predict the flow at a dam system. Linear solutions using time series analyses modelling and non-linear using neural networks techniques.

The daily time series data set was at the river Ulla, Galicia, with three gravity dams for hydropower production. The data set used, ranged from 2013 to 2018. The first model was concerned with the forecast of the inflow to the first dam by using weather variables and hydrometric station data. For each dam, one model was used to predict the discharge and another to predict the storage.

In the feedforward neural network two algorithms were tested and the structure optimized using an hyperparameter tuning with Bayesian optimization. The inputs for each model were selected through subset testing with the non-linear models.

The models were compared, and the best performing ones were tested by connecting them in series and validating through 330 days. The best results were obtained for the neural networks and this results shows that these models can be successfully applied to the forecast of a dam system discharge.

Keywords: Outflow Prediction, Reservoir simulation, Machine Learning, Dam storage forecast

Contents

List of Tables

List of Figures

Nomenclature

ACF	Autocorrelation Function
ADAM	Adaptive moment estimation
AIC	Akaike information criteria
ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive integrated moving average model
ARMA	Autoregressive moving average model
ARMAX	Autoregressive moving average with exogenous inputs model
BP	Back Propagation of Error
BPPT	Backpropagation through time algorithm
CCF	Cross Correlation Function
GBHM	Geomorphology-Based hydrological model
GD	Gradient Descent
IDE	Integrated Development Environment
LM	Levenberg–Marquardt algorithm
LSTM	Long Short Term Memory
MA	Moving Average
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Percepton
MLR	Multiple Linear Regression
MSE	Mean Square Error

NARX	Nonlinear autoregressive exogenous model
NSE	Nash–Sutcliffe efficiency
PACF	Partial Autocorrelation Function
PFC	Peak Flow Criterion
r	Pearson correlation Coefficient
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SCG	Scaled Conjugate Gradient
STA	Early Stopped training approach

Chapter 1

Introduction

1.1 Motivation

For a variety of reasons people have built structures to retain water, disrupting the natural process of water flow. Reservoirs have serve humanity's needs, allowing us to control the natural flow of water. The need to study these man made structures is important to understand its impact on the region and those down stream.

Often, conceptual models are used to tackle watershed studies, allowing researches to simulate under various conditions the mechanism in play within a river. These models are complex and interdependent simulating the behaviour of complex physical phenomena. However what they lack is the ability to incorporate the unknown management of a man made dam, which is connected to a variety of different factors, dependent on the use of such a structure.

To solve this problem, data driven models are needed, to extract from data the human reasoning behind the water management. This was precisely, the problem faced by MARETEC researchers (www.maretec.org) during one of the research projects the HazRunoff project (www.hazrunoff.eu). The aim of this thesis is to develop models to simulate the outflow of a dam system and test the application of such technique. In the future the models developed in this thesis will be applied together with watershed physical model.

1.2 Problem Description

The problem to be tackled in this thesis is the modelling of a dam reservoir system composed of three dams, through data driven models. The dams are located in the Ulla River, Galicia, Northwestern Spain. The three dams are:

- Portodemouros
- Brandariz
- Touro

The problem of predicting the reservoir outflow implies an understanding of the inflow at every given time. As it will be discussed in section 1.3.1 the time series given does contain the inflow. However this value was not measure but in fact estimated from the knowledge of the reservoir geometry and the height of the water level. To avoid work with an estimated value, first the inflow needs to be determined. Then, two different types of model will be used one for the prediction of the reservoir inflow (Model 1) and another to predict the reservoir outflow (Models 2, 3 and 4).

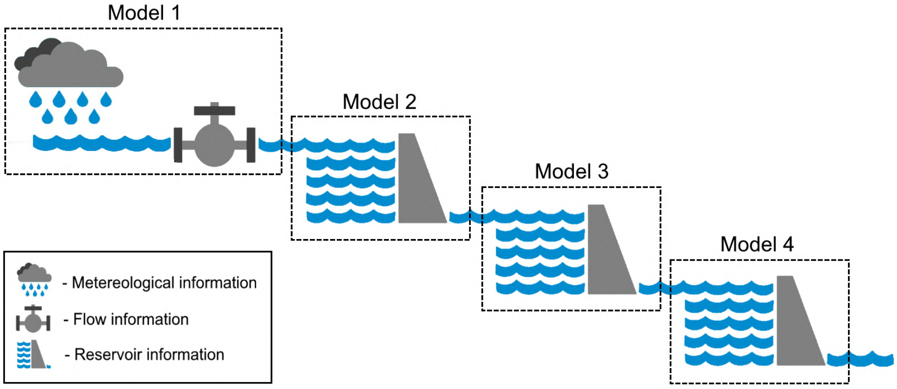


Figure 1.1: Overall Scheme of the models

The figure 1.1 shows a simplified schema of the problem and the general segmentation approach used. Model 1 inputs are the meteorological data and flow data upstream from the first dam system. The model 2 takes the inflow as one of its input in order to predict the outflow. As it will be further explained, in section 3.2.1, the difference between the outflow of the first dam to the inflow of the second dam is negligible. The same happens from the second to the third dam.

1.3 Data Characterization

The information required was obtained from three different sources:

- Dam reservoir
- Meteorological stations
- Hydrometric stations

The data used is a daily time series. The image 1.2 presents the geographical locations of the multiple elements considered.

The only additional data used outside of this group, is information related to the national and regional holidays as well as the week information (Monday, Tuesday, Wednesday etc). This information is used in the context of trying, in a quite simplistic way, impart some of the information related to the power demand which shifts the prices of electricity and the use of the reservoir. The normal behavior of the dam is mentioned to be related to the electricity prices at each instance, yet no further explanation is given in [1]. The electrical demand is connected to multiple factors mainly the weather information,

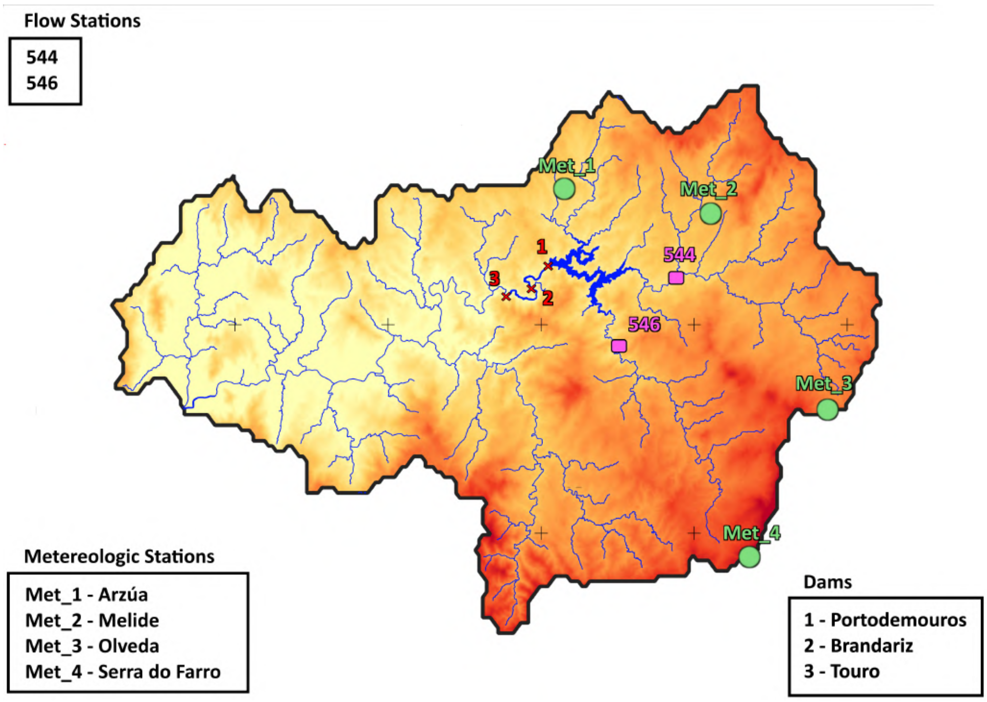


Figure 1.2: Geographical locations of dams, meteorological stations and flow stations

specially temperature [2, 3]. However it is important to account the social and economical factors which also shape the demand.

Valor et al. [3] observed two seasonal effects, when analyzing the data for the entire country of Spain in daily consumption from all sectors. The first seasonal effects is from month to month as the change in seasons and work holiday shifts the use of heating or cooling systems throughout the year. The other seasonal factor is the difference in day to day demand. During a week without holidays the electricity demand is low on Saturday and even lower on Sunday. There is also an effect of a lower value on Monday than other week days this is due to the pause of economic activity in the weekend. The same reason affects days of holiday during the week and the day after. The electricity prices themselves display the same seasonality behaviour as the electricity consumption [4]. For the reasons presented above, the national and Galicia regional holidays were gathered.

1.3.1 Dam Reservoir Characterization

Ulla river drainage basin contains a three dam system, all of them are gravity dams with similar structures but with highly different volumes. All dams have three types of output flows. One related to the power production through a system of hydroelectric turbine. The second is related to the flood gates responsible for the outflow of excess water on the reservoir resulting from flood events. The third one being the bottom gate responsible for maintenance of the dam, as well as water renovation on the reservoir when needed.

Information relating to the fixed parameters of the dams are present in the document for the norms of exploration for each dam,[1, 5, 6]. The documents focused on the maintenance aspects of the dam as well as how to behave in events of extreme weather. To better understand the differences between the three dam system some of the parameters relating to capacity are presented in table 1.1.

	Eletrical Power (MW)	Reservoir Capacity(Hm^3)	Max flow rate turbines (m^3/s)
Portodemouros	85.6	297	126.5
Brandariz	18	2.74	90
Touro	12.68	3.78	60

Table 1.1: Dams structure characteristics

The time series for each reservoir contains values of water level, volume, inflow, total outflow, hydropower outflow, flood gates outflow, bottom gates outflow. All the values are presented in SI units except the volume which is given in cubic hectometers. All the flow rates are given as the average flow rate for each day. The water level is given in meters compared to the sea level. The volume is a quantity derived directly from the water level from tables presented in the norms of exploration ([1, 5, 6]) which contain detailed information regarding the structure of the dams.

Given that the dams were build in different year the data available had different lengths. Regarding the older dam, Portodemouros, the time series started at 1990, its construction dates back to the year 1967. The quality of the data is presented in the figure 1.3, where the percentage of existing data can be seen (100% corresponding to a complete time series with no missing parameter). Over all the data is quite complete, although something cleared changed in 2007 hitting a low point in 2010, but recovering quite well after 2012.

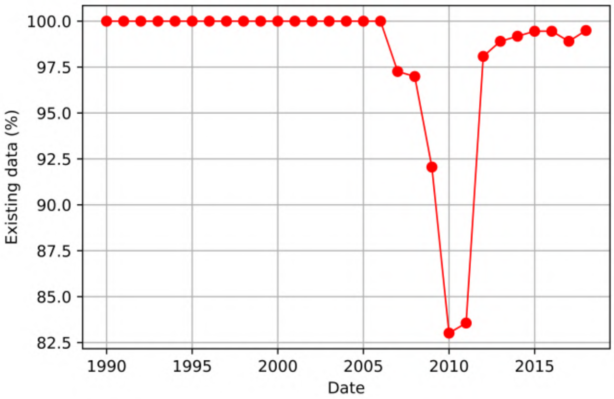


Figure 1.3: Percentage of existing data, Portodemouros Dam

Because the dams is relatively old, when the data is carefully analysed there is a clear difference in management of the reservoir throughout the years. This can be easily seen through the minimum amount of water that was released in order to guarantee the conditions downstream, values in table 1.2. Figure 1.4 compares the monthly minimum flow described in the dams exploration norms [1, 5, 6] to the

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
flow (m^3/s)	15	15	11,9	9,6	8,6	7,9	5	5	5	8,9	11,9	15

Table 1.2: Ecological flow - monthly minimum required

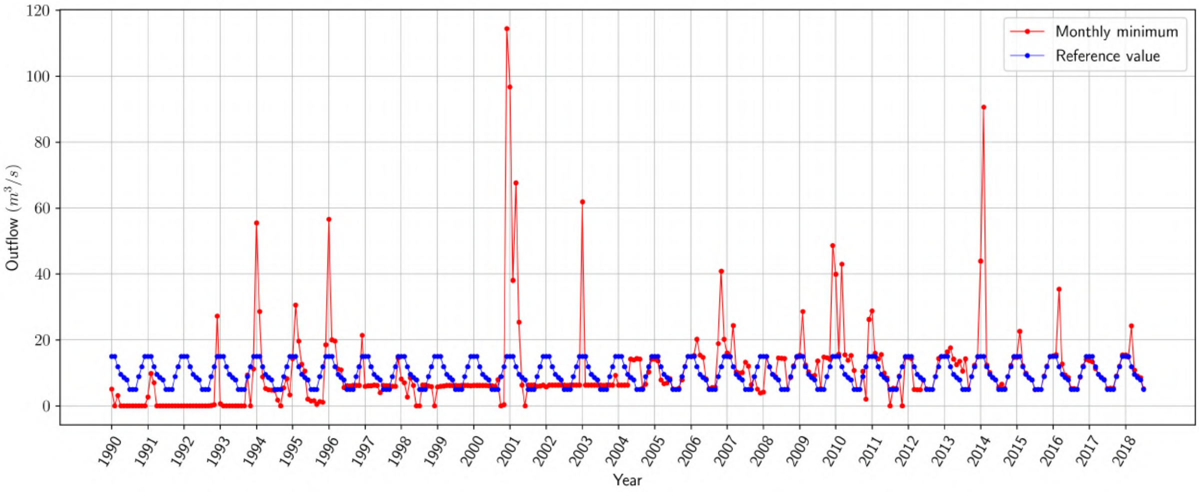


Figure 1.4: Minimum Monthly Outflow Comparison

actual minimum verified in the Portodemouros dam. As it can be seen, this rule is followed only from the middle of 2012 foward.

Both Brandariz and Touro Dams started activity in 2008. The data available from both reservoirs started at the very end of 2009. The few days of December 2009 were first ignored either in Brandariz or Touro, after which the same quantification of existing data was done and presented in figure 1.5. From that information we can also conclude that not every year had useful information. Information from 2014 and forward was chosen as the segment to use for application into the model.

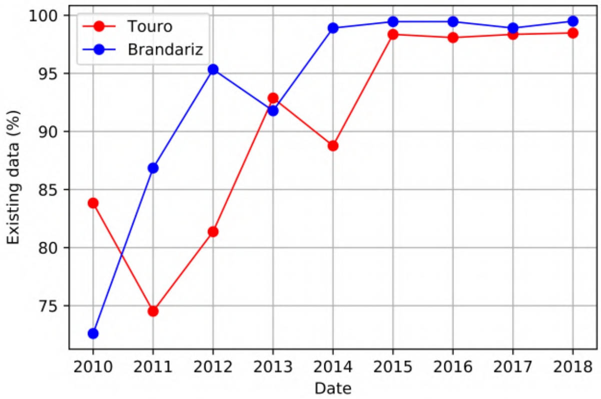


Figure 1.5: Percentage of existing data, Touro and Brandariz Dams

The time series for the three dams ends in 16 July of 2018, therefore all the models and subsequent information will go only as far as that in time.

1.3.2 Hydrometric station characterization

Galicia regional government allows for the consultation of any flow and height sensor along their rivers, through the website [7]. These stations measure the flow rate as well as the height of the water level in 10 minute intervals. Because it is not the aim of this thesis to get finer resolution the information was retrieve in the daily average format.

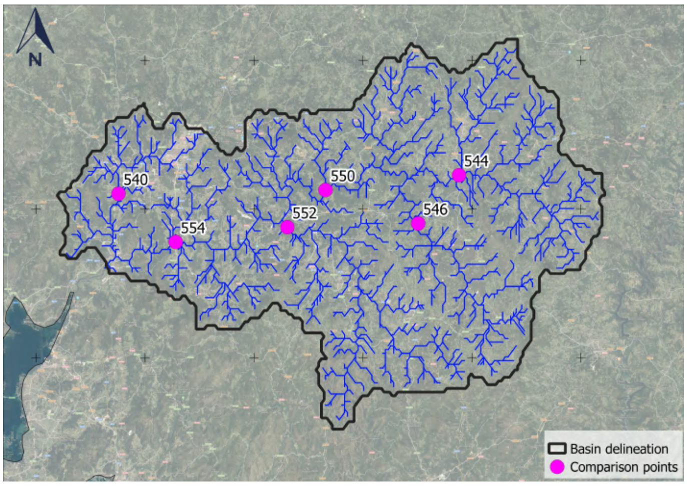


Figure 1.6: Locations of flow stations within hydrological basin

There is 6 flow stations within the hydrological basin as presented in figure 1.6. Only the stations 544 and 546 were upstream from our area of interest, as can be seen in figure 1.2. Both have data starting at the year 2009. However station 544 only from a small fraction of the year, as can be concluded from the analyses of the existing data, figure 1.7. The decision was made to work with data only from 2012 and forward to reduce the amount of missing data. It is important to note that all the information on these existing data graphs is calculated by taking the raw amount of data. For instance, if for a given day there is one variable but not the other, the existing one is still counted. Although, once this data is taken into account to use in a model the full day information will be ignored, in order to avoid methods of completing missing data. This justify why years, like 2011, with percentages of 82,19% for station 544 and 91,23% is still disregarded given that in actual applications the value of the information would be even lower.

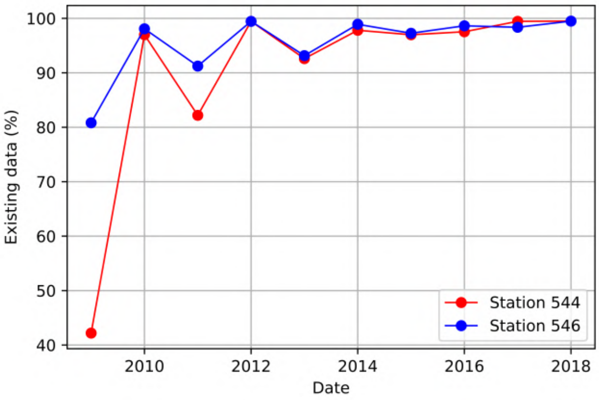


Figure 1.7: Percentage of existing data, Flow stations

1.3.3 Meteorological Stations

Similarly the data relating to the meteorological stations records is publically available by the same organization, Meteogalicia (from the regional government structure, Xunta Galicia), website [8]. In figure 1.8 all the meteorological stations in the surrounding area are shown. To define which stations are of interest two steps were taken. First those outside of the hydrological basin were eliminated. Secondly those that mainly influences flow downstream of the dams were also not considered, for being too far. Given these restrictions only 4 stations were used, as can be seen in figure 1.2 related to the other elements. The four selected stations are near tributaries of the first reservoir.

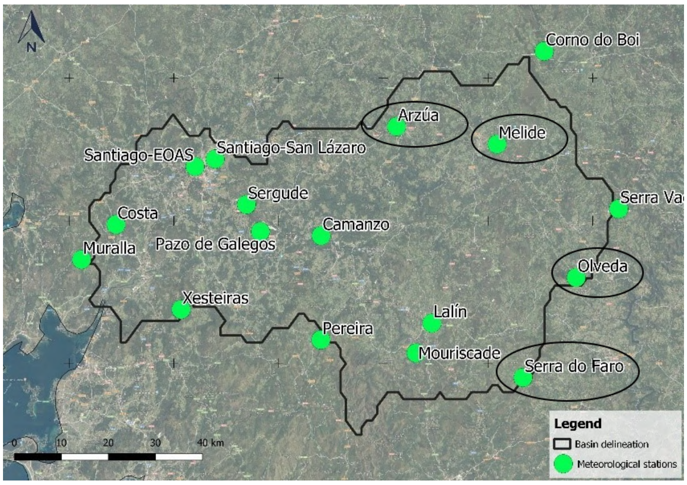


Figure 1.8: Meteorological Stations

The information is recorded with 10 minute intervals, for this application the data was extracted in the daily format. Five parameters were considered from each stations:

- Temperature - daily average temperature measured at $1.5m$ in Celsius
- Humidity - Average relative humidity expressed in %
- Precipitation - global amount of rain a day per area measured in L/m^2
- Irradiation - The power of solar radiation per area measured in $kj/m^2.day$
- Pressure - Atmospheric pressure measured in hPa

The same existing data quantification per year was performed to understand what amount of data is missing, figure 1.9. In this case the bulk of the missing information is easily justify. Olveda does not contain any information related to pressure therefore its possible maximum is 80%. Arzúa only starts to have information of the solar irradiation in 2014. For this reason data was only considered from 2014 forward.

Table 1.3 summarizes the temporal windows chosen for each model.

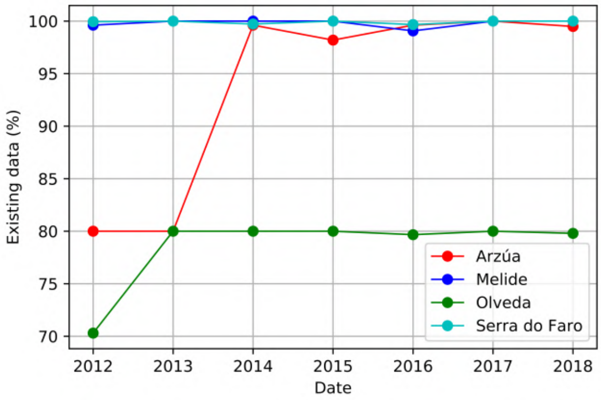


Figure 1.9: Percentage of existing data, Meteorological stations

	Info needed	Usefull time window	Overall model
Model 1	Metereological	2014 - 2018	2014 - 2018
	Hydrometric	2012 - 2018	
	Portodemouros	2013 - 2018	
Model 2	Portodemouros	2013 - 2018	2013 - 2018
Model 3	Brandariz	2014 - 2018	2014 - 2018
Model 4	Touro	2014 - 2018	2014 - 2018

Table 1.3: Data time window for each model

1.4 State of the Art

Considering that the problem tackled was decomposed into two types of prediction problems as explained in section 1.2 to understand what as been done in the area it was subdivided into:

- Inflow forecast problem
- Reservoir outflow forecast problem

1.4.1 Inflow Forecast Problem

This is a fundamental issue when studying a given reservoir, quite important for the prediction of extreme weather conditions such as floods. Several types of models have been used with different available information these range from conceptual model which take into account the physical phenomena, to those data driven. The later type of model, range from models based on time series analyses, such as an autoregressive integrated moving average (ARIMA) models to others based on Artificial Neural Networks, (ANN).

Physical models

Yang et al. [9] took on the task of determining the inflow values trough a physical model, a Geomorphology-Based hydrological model (GBHM). This model can be divided into a two component calculation using: hydrological hill slope simulation and the river network routing. To understand the complexity of such models in the hill slope part of the model it is included: canopy interception, evapotranspiration, infiltration, surface flow, unsaturated flow and ground water flow. These models use grid calculation to determine the river discharge into the reservoirs (more detail [10]).

Another physical model importance to take note is MOHID-Land. MOHID-Land is based around three mediums and their interactions: atmosphere, soil and water. The model works with a spatial grid with variable resolution where the calculations are made for each cell. A multitude of parameters are used to calculate the processes associated with the physical phenomena of a watershed. This type of model has complex calculation which are extremely useful to understand the multiple hydrological process which can be used to extract a variety of values [11].

The clear benefit of such models is the wide range of variables which could be extracted because of the necessary calculation. This allows, when analyzing, the study of multiple steps of the process and a more detailed understanding of the system. Data driven models are very general in scope but once a model is trained it offers either none or limited perspective on what is happening between the input and output.

Data Driven Model

Several authors have compared the use of typical regression models to predict inflow against the more modern and adaptable ANN.[12–14]. Regression models have been widely used in hydrological predictions given the ability to predict variables with a stochastic nature as well as their readability that makes them better at interpretation.

Jain et al [12] used a Multilayer Percepton (MLP) strucuture and an ARIMA model to predict monthly inflow. The study used 32 years of data. A very high seasonal effect was detected as expected, with a 12 month cycle. This was observed through the time series models, as well as the Autocorrelation function (ACF). A negative linear trend was detected, which justify the need of an ARIMA instead of an autoregressive moving average model (ARMA). The models were tested for the minimum Akaike information criterion (AIC) as well as Q-statistic. The ANN used, was a quite standard feedfoward, back propagation (BP) network with three layers. The sigmoid function was chosen, and the loss function used the mean squared error criteria, (MSE]). There is inherent problems to the simple BP algortihm, such as: function trapped in local minima and slow convergence. To counter such problems momentum and noise were added. Which produced better results even when compared with one that scaled variables into a range of [0,1]. To speed convergence, a phased training schedule was used. The results were overall better with the ANN model when compared with the ARIMA model. Notably, higher flows were better mapped by the ANN where lower flows were better mapped by the ARIMA model.

Mohammadi et al.[14] had a very similar problem predicting the monthly inflow to a reservoir, the main difference being the context. In this problem the spring snow melt was a major contribution to the flow. Three models were compared: a regression analysis, a ARIMA and a ANN. The multiple linear regression (MLR) and ANN used as inputs: the rainfall in the previous time step, the snow melt watershed at previous time step, temperature at the previous month, and the inflow of the previous time step. As in the study discussed previously ([12]) a ARIMA model was used with a 12 month cycle and the final parameters of the ARIMA were (1,0,1).(0,1,1). The ANN used a simple MLP with three layers. The results were better for the ANN than any other method.

Budu [13] compared MLR to ANN with pre-processing techniques such as wavelet and moving average(MA). Budu used daily information instead of monthly. The information used were: daily rainfall, inflow and stream flow at upstream flow stations. The relevance of this study falls in the use of wavelet transform pre-processing. Wavelet analysis consists of changing the signal to a different signal (shifted and scaled versions). The wavelet shows its particular use when localized high frequencies or large scale variations are presented in the same signal. In hydrology, shifts of scale coincide with the seasons. The wavelet decomposes the signal into multiple other signal this can help with extracting relevant information. Two types of ANN were used: BP with LM and another with radial basis function. Several wavelets were tested and a MLR was also tested. The results showed that the wavelet approach showed better results than the unprocessed inputs alone. Surprisingly when coupled with the wavelet the MLR had the best results.

Ahmad et al. [15] attempted to model the inflow of a reservoir with the use of ANN. The goal was to improve flood peak management. As expected the need for such methods of machine learning steam from their innate ability to adapt to different circumstances. As well as its ease of processing when compared to model that use physical equations. The main difference here lays on the fact that this study used data from 23 reservoirs. A feed forward ANN with BP algorithm optimized by LM was used here. The early stopped training approach STA was also implemented along the LM algorithm. The STA consisted in using the validation data as a stooping criteria. The rule being applied was that if the validation data started to have increasing error values then the model was likely over fitting to the noise in the data. In this study, to stop the training a value of 6 iterations of increasing validation error is necessary and the minimum error model is recovered. To improve generalization, another method was implemented called regularization. The forecast was made for a 7 day period. To determine the input variables a cross correlation function (CCF) was implemented, getting the correlation of the signal with a lagged versions of another. The results were quite satisfactory, even at 7 day prediction for most dams. Although it is clear that different dams in size and conditions showed quite different results. Yet, the approach of using ANN coupled with a robust way of determining what are the relevant variables, produced a strong result for inflow forecast.

Coulibaly et al. [16] attempted to forecast daily inflow through the use of a feedfoward back progation ANN with again the LM algorithm. Similarly, the [14] a STA is implemented as the innovative factor. Predictor variables are the previous water inflow value $(t-1)$, temperature (maximum, minimum, mean), precipitation $(t$,$t-1$,..,$t-4)$ and snowmelt $(t$,$t-1$,..,$t-4)$. The ANN consisted of the typical three layer where the number of hidden neurons were obtained through trial-and-error. The stoping criteria is based on a generalization loss at epoch t ($GR(t)$).

$$GR(t) = 100 * (\frac{E_{val}(t)}{E_{low}(t)} - 1) \tag{1.1}$$

where E_{val} was the validation error at time t, E_{low} was the minimum validation error at a previous time value. To compare the model an Autoregressive–moving-average with exogenous inputs model (ARMAX) was considered, as well as one conceptual model. As for the criteria to evaluate the models, the Nash–Sutcliffe efficiency coefficient (NSE), the RSME, Pearson Correlation and a peak flow criteria (PFC). The later is used as the name implies to evaluate the peak flow performance as it can be an important factor of the flow prediction. It is defined as:

$$PFC = \frac{(\sum_{p=1}^{n_p}(y_p - \hat{y}_p)^2 * y_p^2)^{\frac{1}{4}}}{(\sum_{p=1}^{n_p}(y_p^2))^{\frac{1}{4}}} \tag{1.2}$$

where n_p represented the number of peak flows, y_p was the real value and $\hat{y}_p$ was the estimated value. The results of the ANN with or without the STA were better than the ARMAX approach, as expected. The performance of the ANN with STA were in fact better, although not by a strong margin. This could indicate that overfiting when the number of neuron is tested is not such a extensive problem in this data set. The main advantages were the faster training time when compared to not implementing the STA.

1.4.2 Outflow Forecast Problem

Forecasting the discharge of a reservoir does imply either an understanding of the rules and demands of water associated with a given reservoir or modelling the history of releases.

Some reservoir are managed according to rule curves related to the desired storage and the outflow targets. Naturally, the operation of reservoirs systems is a compromise either to minimize spillage or to avoid any kind of water shortage downstream. [17]

Several authors have presented data driven models to estimate the discharge of water when the exact rules of operations are unknown. [9, 18]

One of the models assumed a rather simplistic approach, despite modelling a reservoir with $920Hm^3$ of capacity (Manwan dam). These models used only previous data points of flow to predict the next one. As input, assumed four values of outflow from the current time, up to three days prior and as output the forecast for the next day. It used a feedfoward ANN with three layers and it focused on understanding the best optimization algorithm. Three optimization algorithms were tested: LM, Scaled Conjugate Gradient (SCG) and gradient descent (GD). In both daily and monthly models the SCG algorithm presented stronger results. [18]

Yang et al. [9] took into account a more complete understanding of the basin in a more comprehensive approach, using a combination of physical based models and (ANN). The implementation of data driven models on reservoir outflow forecast, comes from the inability of conceptual or physical models to predict operations outside of the defined set of reference curves or rules for management. Even when the rules of operation are understood in actual reservoir management differences occur. Either by people imparting their own judgment based on experience or in extreme situations like floods or droughts.

Yang et al. [9] considered as input to the models the delayed outflow $(t-1)$, inflow both delayed and forecast $(t-d_{\alpha}, ..., t, ..., t+d_{\beta})$, as well as the storage value at the previous time$(t-1)$. Where d_{α} refereed to the days of delays considered and d_{β} to the days of forecast. The output on the models was only the outflow at time t. To determine the next storage value, the following equation was taken into account:

$$S_{t+1} = S_t + \frac{\Delta t}{2} * (Q_{in,t} + Q_{in,t+1}) - \frac{\Delta t}{2} * (Q_{out,t} + Q_{out,t+1}) \tag{1.3}$$

where S represented the storage, Q_{in} was the inflow to the reservoir, Q_{out} the outflow and Δt the time interval.

The problem was formulated as a Nonlinear autoregressive exogenous model (NARX). Three models where implemented an MLP, an MLP with a genetic algorithm and a long short term memory neural network (LSTM). The MLP model studied used a three layer configuration and a LM method for optimization. The applied hyperparameter tuning was used for the determination of days of delay and the number of neuron in the hidden layer. Genetic algorithm modified MLP, uses this optimization method in an attempt to minimize problems, such as high sensibility to initial conditions of the weights and biases of the network. Hyperparameter tuning considered the same parameters. The LSTM model contained three layers and used to train the adaptive moment estimation (ADAM) algorithm. All three models considered the MSE as the loss function and the data was split the same way. Training data was 70%, validation 15%, with the remaining 15% to test. The results were satisfactory for all the models and all displayed a behaviour of underestimating high flow situation and overestimating low flow situation. This effect was said to potentially be related to the sigmoid activation function used. [9]

Jain et al. [12] optimized reservoir operation using an MLP and compared it to other simpler methods, such as dynamic programming, linear regression among others. Chaves et al. [19] worked on a similar optimization problem of the reservoir operation. The difference being the use of a genetic algorithm to calculate the weights of the ANN. It showed similar results to dynamic programming solution on single decision variable problems. However, when multiple decision variables were tested, the ease on adding more output nodes made it a better solutions than more conventional approaches.

1.5 Thesis Outline

The objective of this study is to apply data driven techniques to achieve good forecast models to predict outflow of the dams reservoirs. For that this work will be divided into: data driven modelling, model implementation and results.

In the next chapter, data driven modelling, the concepts over the decided approaches will be presented. First the explanation of the linear modelling approach, ARMAX and the respective method of selecting the order. Then the machine learning (ML) techniques to be implemented regarding an MLP network and its training. The chapter finishes by discussing the performance criteria to be used as well as a summary of the procedure to be taken in the following study.

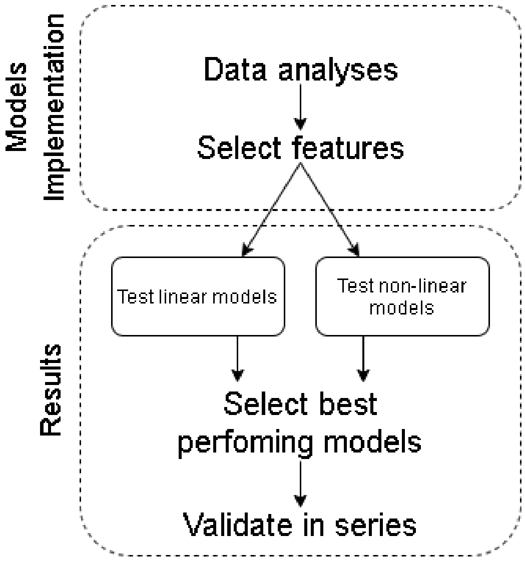


Figure 1.10: Study methodology

The methodology for this work can be summarize with the graph in figure 1.10.

For the model implementation, an analyses of data is presented, considering data distribution and relationships between variables. Then, the two ideas for selecting input variables are present and the selected features explained. The structure of the full dam system is also displayed.

In the results chapter the two modelling approaches will be compared and the best performing one selected for the full system validation. In this section, the models will be connected in series to validate the work, with the outputs of the previous model being the inputs of subsequent models.

Chapter 2

Data Driven Modelling

In the previous chapter, in section 1.4, several solutions used to tackled forecasting problems related to the study of water reservoirs were discussed. The types of solution can be split into three types:

- Physical or conceptual models
- Time series analyses models
- Artificial neural networks

Only models of the later two types are going to be implemented in this work. Such models are highly adaptable to a wide variety of problems, with a greater computational efficiency. Despite such major advantages these models generally are a 'black box' approach, were the exact mechanism of the system remains unknown. On the other hand, physical models have the advantage of being focus on modelling the phenomena associated to the system. However, when the rules are not known, data driven models became a necessity. In the present case the objective is to model a reservoir which is controlled by people, despite knowing some of the story (the capacity, the quantities that can be discharge and even the minimum required flow per day [1]) there is still plenty of variables that are unobservable or impossible to relate without recurring to models which 'learn' from previous data. Machine learning techniques are increasingly being applied to all fields of science and engineering, generating a multitude of approaches from black box models to even mixed model approach which combine physical knowledge with data driven methods [20].

In this chapter, time series analysis (ARMA type) and a NARX model using an ANN, will be presented. The evaluation criteria of such models will also be explained, as well as hyperparameter tuning.

2.1 ARMAX

The analyses of time series data can be made through several application, Box et al.[21] defines to different methods: the forecasting of a time series (univariate model) and a transfer function approach (multivariate model).

Forecasting of a time series consists of predicting values only considering the previous values of the time series. For stationary processes this is achieved through a combination of autoregressive (AR) and moving average (MA) processes. A stationary process can be defined by its mean, variance and autocorrelation function(ACF), the series varies around a fixed mean. For some non stationary behaviours other processes can be combined, either to remove trends or seasonal patterns, ARIMA models [21]. A transfer function approach assumes that instead of using only the signal itself for the forecasting it takes other variables as explanatory inputs to predict the desired output.

The problem is a multivariate problem, so a transfer function type model will be required as well as the delayed versions of the inputs and the output. This problem can be described as such:

$$\hat{Y}(t+1) = f(Y(t), Y(t-1), ...Y(t-m), X(t), X(t-1), ..., X(t-n)) \tag{2.1}$$

where $\hat{Y}_{t+1}$ is the estimated variable, Y_t is the value at time t of the study variable, X_t is the explanatory variable which affects the results of the study variable. Different types of transfer function models can be implemented depending on the problem. One standard approach used to model systems into polynomial models is the auto regressive moving average with exogenous variables model,ARMAX.

An ARMAX model [22], can be described by the following equations:

$$Y_t + a_1 Y(t-1) + ... + a_{n_a} Y(t-n_a) = b_1 X(t-1) + ... + b_{n_b} X(t-n_b) + e(t) + c_1 e(t-1) + ... + c_{n_c} e(t-n_c) \tag{2.2}$$

where the $e(t)$ is assumed to be white noise, with

$$A(z^{-1}) = 1 - a_1 z^{-1} + ... + a_{n_a} z^{-n_a} \tag{2.3}$$

$$B(z^{-1}) = b_1 z^{-1} + ... + b_{n_b} z^{-n_b} \tag{2.4}$$

$$C(z^{-1}) = 1 - c_1 z^{-1} + ... + c_{n_b} z^{-n_b} \tag{2.5}$$

A, B and C are represented in the z plane. The model can be rewritten as

$$A(z^{-1})Y(t) = B(z^{-1})X(t-1) + C(z^{-1})e(t) \tag{2.6}$$

The values n_a, n_b and n_c are the orders of the corresponding components of the model. The model is therefore represented by ARMAX(n_a, n_b, n_c), a forth term can also be added, n_k, which correspond to the pure input-output delay, in the formulation is assumed to be one.

In the representation above it can be understood that the A component is the autoregressive part. B is the modelling of the extra input X dynamics on Y. C is the modelling of white noise with a moving average process. A representation of this model in block diagram style is presented in the figure 2.1. In the diagram, it can be seen that the noise component, is not only modelled by a moving average process, but also by the dynamics of the system. Other models have this dynamic completely decoupled, for example the Box-Jenkins models [22].

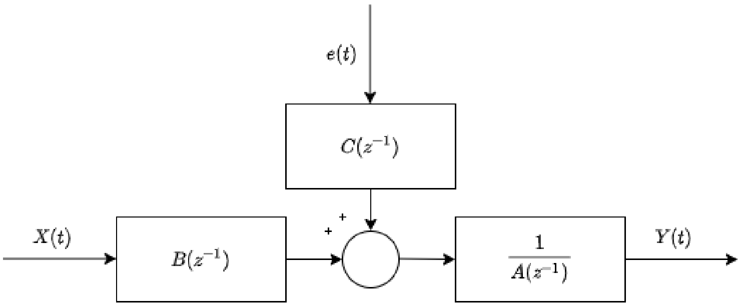


Figure 2.1: Block Diagram of ARMAX

For multiple predictor variables the reformulation consists in simply adding more B functions to the system for as many inputs as needed.

Several methods can be used to calculate the parameters of a ARMAX such as:

- Gaussian-Newton Method
- Levenberg-Marquardt
- Gradient Descent Search

These methods minimize the sum of the square values of the model's residuals, although some algorithms incorporate into this calculation a derivative of the function to determine in which direction to go.

This type of modelling assumes that the behaviour of the system is linear and time invariant [21].

When testing ARMAX models with different orders the Akaike information criteria (AIC) [23] can be used as a decision criteria for which model order to select. This criteria uses the number of parameters used to build the model, as well as the maximum value of the likelihood function. The likelihood function relates to the quality of the fit to a set of observed data. The minimum value of the AIC represents the minimum information loss while penalizing overfitting.

2.2 Artificial Neural Network

The natural processes all around us have been a constant source of inspiration in the world of engineering. With advances in computational power, it was a matter of time before, trying to simulate a learning process.

In 1943, McCulloch and Pitts [24] published an article mathematically characterizing the theory, at the time, of how neurons interact with each other. It described a mathematical formulation of a net of neurons where the neurons have an "all-or-none" behavior. In 1956, Rochester et al. [25] simulated a net of neurons in an IBM Type 704 Electronic Calculatorwith with the objective of testing proposed mathematical models to study how the brains work. In 1958, Rosenblatt [26], presented a model for a single percepton still to learn how our brains works. In this study, the neuron (percepton) activation was based on a threshold. In 1960, Widrow et al. [27] applied artificial intelligence to a real world problem, with the machine called ADELINE (adaptive linear switches), capable of patterns recognition using a single layer network. Eventually the performance of neural network, were limited by the computational power of the time, and by the impossibility of a single percepton to solve non linearly separable problems (the exclusive or problem). Which introduced a slow down in research later to be picked up in the 1980s. Now neural networks are used for the ability to solve problems efficiently and learn complex patterns from data. Making them ideal tools for problems where the patterns are unknown, like the problem at hands.

To better explain the neural network to be applied to this problem, first the structure and calculus of a feedfoward network will be explained and then the back propagation and learning algorithms.

2.2.1 Multilayer Percepton Structure

The feedfoward MLP is a network capable of solving non linearly separable problems unlike the single percepton machine. As the name suggests this type of network is defined by having three or more layers, one related to the inputs and another for outputs. The intermediate layer(s) are called hidden layer. A general illustration is presented in figure 2.2. Important to note, that the number of neurons in the input and output layers match the number of respective signal of input and output [28].

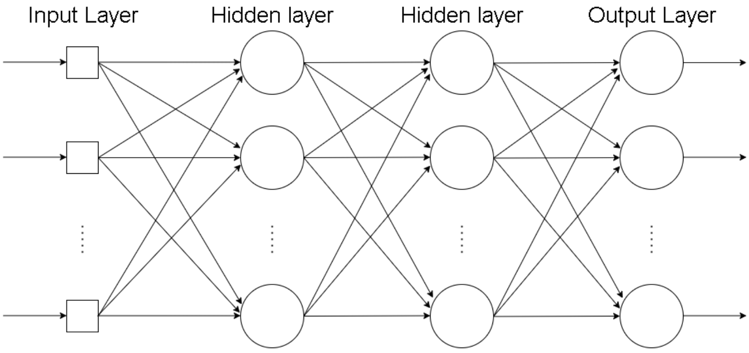


Figure 2.2: Architecture graph of an MLP

Figure 2.2 shows a fully connected graph, meaning that every neuron connects to every neuron in the next layer. The densely connected is the most common approach, however neural network can be connected in different ways. The input layer is composed of sensory units, merely acting as a source node and no calculation is made. The network start with the input layer then for every neuron of the first hidden layer a value is calculated, the input of the next layer. This operation is repeated until the output signal is reached, the signal propagates forward[28].

At each neuron the value is calculated through the use of an activation function that takes the multiple inputs and calculates the output of that neuron as illustrated in figure 2.3.

Figure 2.3 can be described mathematically as:

$$v = \sum_{i=1}^{m} w_i x_i + b \tag{2.7}$$

$$y = f(v) \tag{2.8}$$

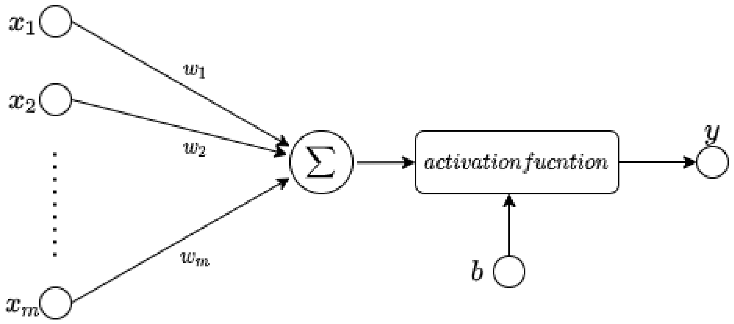


Figure 2.3: Signal flow graph of a perceton

where v is the induce local field, x are the inputs to the neuron, y is the output of the neuron, w is the weights of the connection, b is the bias of the neuron and f represents the activation function. The weights are different for every connection between neurons and the bias is characteristic of every neuron.

For the activation function a variety of functions can be used. The Rosenblatt's perceptron [26] used a hard limiter, a threshold, the output was either: +1 or -1 for classification problems. However, in the MLP case, one commonly used function is the sigmoidal function defined by a logistic function [28]:

$$y = \frac{1}{1+e^{-v}} \tag{2.9}$$

Other activation functions can be used, for example:

- tanh function

$$y = \frac{e^{v}-e^{-v}}{e^{v}+e^{-v}} \tag{2.10}$$

- Rectified linear unit (ReLU) [29]

$$y = \begin{cases} 0 & \text{if } v \leq 0 \\ v & \text{if } v > 0 \end{cases} \tag{2.11}$$

The most important difference between these functions are the computational efficiency and the types of derivatives which will be relevant for the training of the network.

2.2.2 Network Training

One central idea in neural network is the error back propagation (BP) algorithm, which enables the optimization of weights and bias associated with multiple layers of a network. The following explanation was mainly based on the book, Box et al. [28].

Unlike the signal function, where the flow starts at the input and moves to the output, in the BP it starts with the error of the calculation and calculates the parameter correction backwards. First in the output layer connections, moving then, for the hidden layer that precedes it, and so on for the other layers.

To understand the algorithm two particular cases, need to be understood. The case of an output neuron

(o) and one hidden neuron (h).

Neuron o - output neuron

For the case of a regression problem, lets assume the error signal is the difference between the measured value (y_o^M) and the estimated by the model (y_o) defined as:

$$e_o(n) = y_o^m(n) - y_o(n) \tag{2.12}$$

where n represents a given value of the training data set. The average error,used as the objective is then defined as:

$$E_(av) = \frac{1}{N}\sum_{n=1}^{N} E(n) \qquad \text{where} \qquad E(n) = \frac{1}{2}\sum_{o \in O} e_o^2(n) \tag{2.13}$$

where N is the size of the data set, O is the set of output neurons and $E(n)$ is defined as to be the instantaneous error energy

To minimize the average error for every instance of data the objective is then to minimize the $E(n)$. To group every free parameter (weights and bias) in the same way it is useful to consider the bias as a weight that has always input 1. Now the equation of the induced local filed (eq. 2.9) can be rewritten as

$$v = \sum_{h=0}^{m} w_{oh} y_h \tag{2.14}$$

where y_h is the output of hidden neurons that serves as input to the output neuron.

The value $E(n)$ is a function of all the weights. To minimize, it is useful to know the value of the partial derivative for every weight, $\frac{\partial E(n)}{\partial w_{oh}}$.

$$\frac{\partial E(n)}{\partial w_{oh}} = \frac{\partial E(n)}{\partial e_o(n)} \frac{\partial e_o(n)}{\partial y_o(n)} \frac{\partial (y_o(n))}{\partial v_o(n)} \frac{\partial v_o(n)}{\partial w_{oh}} \tag{2.15}$$

Differentiating equations 2.13, 2.12, 2.8, 2.14 results in

$$\frac{\partial E(n)}{\partial w_{oh}} = -e_o(n) f_o'(v(n)) y_h(n) \tag{2.16}$$

Knowing this value the weight correction can be made through the use of the following equation (delta rule):

$$\Delta w_o h = -\eta \frac{\partial E(n)}{\partial w_{oh}} \tag{2.17}$$

where the minus sign exists in order to move the correction in the direction that minimizes $E(n)$ and η corresponds to the learning rate. Defining local gradient as

$$\delta_o(n) = \frac{\partial E(n)}{\partial v_o} = -e_o(n) f_o'(v(n)) \tag{2.18}$$

the delta rule can be rewritten as

$$\Delta w_{oh} = \eta \delta_o(n) y_o(n) \tag{2.19}$$

The calculation in the case of the output nodes is quite straight forward and can be summarized has a multiplication between the error value, the derivative, the input and a learning parameter.

Neuron h - hidden neuron

For the output node the reasoning is quite direct however it is also essential to update the value of the hidden layers. Despite the fact that these layers are not immediately accessible, part of the error is a results from these parameters.

In order to correct these parameter, first is important to define the local gradient of a hidden neuron.

$$\delta_h = -\frac{\partial E(n)}{\partial y_h(n)} \frac{\partial y_h(n)}{\partial v_h(n)} = -\frac{\partial E(n)}{\partial y_h(n)} f'_h(v_h(n)) \tag{2.20}$$

To continue exploring these calculation we move to define the error energy (equation 2.13) in relation to the output of an hidden neuron.

$$\frac{\partial E(n)}{\partial y_h} = \sum_O e_o(n) \frac{\partial e_o(n)}{\partial y_h(n)} \tag{2.21}$$

Now, to understand the derivative of the error on the output space, it is necessary to decompose it in relation to the induce local field of neuron o.

$$\frac{\partial e_o(n)}{\partial y_h(n)} = \frac{\partial e_o(n)}{\partial v_o(n)} \frac{\partial v_o(n)}{\partial y_h(n)} \tag{2.22}$$

In equation 2.12 if considered that the value of y_o is the activation function in terms of the induce local field, results in

$$\frac{\partial e_o(n)}{\partial v_o(n)} = -f'_o(v_o(n)) \tag{2.23}$$

Now differentiating equation 2.7 it results in

$$\frac{\partial v_o(n)}{\partial y_h(n)} = w_{oh}(n) \tag{2.24}$$

Rewriting the local gradient of the hidden neuron, with equations 2.21, 2.22,2.23 and 2.24 results in

$$\delta_h(n) = f'_h(v_h(n)) \sum_O -e_o(n) f'_o(v_o(n)) w_{oh}(n) \tag{2.25}$$

The equation above makes intuitive sense, since it takes into account all the error of the output layer and the weight of the connection, as well as the function of the neuron itself. Taking into account the equation2.18 the local gradient of a hidden node can finally be written in terms of the subsequent local gradients, resulting in

$$\delta_h(n) = f'_h(v_h(n)) \sum_O -\delta_o(n) w_{oh}(n) \tag{2.26}$$

Figure2.4 illustrates this equation. It is now clear that the back propagation derives its name from the nature of the optimization process, which starts with the output error and goes backwards updating the values of the weights.

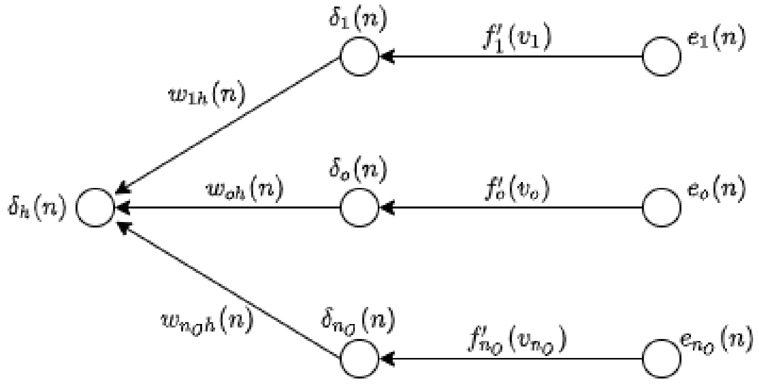


Figure 2.4: Graph of a local gradient on a hidden neuron

To summarize, the correction of the weights is given by the delta rule which is the multiplication of the local gradient of the neuron with the learning rate and the input of the neuron corresponding to that connection. The local gradient can either be for an output neuron or hidden neuron.

Other considerations and variations on the algorithm

The process described above is for one instance of data (one day worth of data if the data set is daily time series). Going over the entirety of the data set once is described as an epoch. Several epochs are used for the training of every network.

One effect the learning rate as over the training process is if it is too small it can be slow and easily trapped in local minima. However if it is too big, it can easily create instability, with oscillatory behaviour for example. So, one common practice is to add a momentum term, α which takes into account the previous correction of the weights illustrated in the following equation. [28]

$$\Delta w_{oh}(n) = \eta\delta_o(n)y_o(n) + \alpha\Delta w_{oh}(n-1) \tag{2.27}$$

The momentum varies in the range $[0, 1[$. The inclusion of this variable has a stabilizing effect on the training process.

Several variations exist on this algorithm in this thesis only two will be compared:

- Mini Batch Gradient Descent
- RMSProp

The batch gradient descent unlike the sequential and online approach above, which updates the values for every step of the training data, it only updates the value after an epoch [28].

The mini batch variation takes a compromise between both approaches it performs an update in a mini batch manner, from n to n values of the training data. It varies typically between 50 to 256 [30].

The RMSProp is notorious for not being formally published despite the widespread use especially for deep neural network, it was introduced by Geoff Hinton in Lecture 6e of his Coursera Class. The idea was to solve a problem with the Adrag [31] algorithm which adapts the learning rate over time, it works by keeping record of the previous local gradients.

$$\text{new learning rate} = \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \tag{2.28}$$

where G_T is a diagonal matrix with the sum square of the previous local gradients up to that point in time, ϵ is a smoothing term which avoids the division by zero. The problem in this algorithm lies in the fact that it keeps adding more and more positive value. This makes the learning rate progressively smaller, resulting eventually in a value of the learning rate so small that the model does not learn anymore. [30]
To solve this problem the RMSProp algorithm introduces a term similar to the momentum on the sum square of the gradient, $E[\delta^2]_t$.

$$E[\delta^2]_t = \rho E[\delta^2]_{t-1} + (1-\rho)\delta_t^2 \tag{2.29}$$

where ρ is recommended to be set at 0.9. Making the new learning rate become:

$$\text{new learning rate} = \frac{\eta}{\sqrt{E[\delta^2]_t + \epsilon}} \tag{2.30}$$

Hinton also suggests that a normal momentum term can be used but it does not pack the same impact as in the other methods discussed.

2.2.3 Hyperparameter tuning

As was seen in the previous subsection, the calculation of a neural network is a problem of optimization of weights and biases. However the network is define by several other parameters:

- Number of hidden neuron
- Number of hidden layers
- Type of activation function
- Learning rate
- Momentum
- etc.

The optimization of these parameters, becomes in it self an optimization problem were the relationship with the performance is not directly obtain. These parameters are called hyperparameters and two types of optimization can be implemented. One is the trial an error manual approach which can easily became extremely time consuming. The other is an automatic approach were several methods can be implemented. One immediate thought is to test all possibilities which guarantees an optimal solution (grid search). However this approach suffer from the same problem, the time it could take to test several parameters. One approach which as shown good results is the Bayesian optimization [32].

The Bayesian optimization is a strategy, design to optimize problems of computationally expensive functions. This approach takes into account the prior tested data. It also takes into account the idea of exploration and exploitation, which is common in optimization algorithms. It is based on a Gaussian process (GP) which is a function, where the variable is a Gaussian distribution not a scalar.

$$f(x) \backsim GP(m(x), k(x, x')) \tag{2.31}$$

where $m(x)$ is the mean vector, k is covariance function and x is a data point of the search space.

So the function k usually the exponential square function measures the degree of approximation between two tested points x. The idea is that, the closer two points are, the less uncertainty there is and if two points are further away the more uncertainty there is. So, with enough points of data GP can give an overall idea of the desired function to optimize. [32]

The other component, is the acquisition function that determines where to search next. It can either chose to move in the direction of lower value of the mean (for a minimization problem) which corresponds to exploitation of the known data. Or it could move in the direction of the greater variance, which corresponds to exploration of high uncertainty of the function space. [32]

The upper confidence bound is the acquisition function implemented in the following work.

2.3 Performance criteria

To evaluate the performance of the model 4 criteria will be used, in order to have a more complete view of the performance of the model:

- Root mean square error
- Mean absolute error
- Pearson coefficient
- Nash Sutcliffe Efficiency

Root Mean Square Error (RMSE)

This parameter is a common measure of error and most models leading with regression problems use it as the objective function and is defined as

$$RSME = \sqrt{\frac{1}{N}\sum_{n=1}^{N} e^2} \tag{2.32}$$

where N is the length of the training set and e is the error define as the difference between the measured value and the estimated from the model, equation 2.12. The use of the RMSE assumes the error follows a normal distribution for other distribution other metrics might be more usefull [33].

Mean Absolute Error (MAE)

To complete our view of the error of the model a metric such as the MAE will also be used. It is defined as

$$MAE = \frac{1}{N}\sum_{n=1}^{N} e^2 \tag{2.33}$$

The main difference between RMSE and MAE is the fact that the RMSE penalizes variance while the MAE gives the same importance to every point. [33]

Pearson coefficient (r)

Another widely used metric to compare two sets of data. This metric is of importance because it uses the covariance between the data, focusing on the way the data varies along its mean an not the absolute value. It is defined as

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{2.34}$$

where X and Y are the variables to be compared (measured against estimated), cov is the covariance and σ_X represents the deviation of a variable X. This metric varies between 0 and 1, 1 being perfect linear correlation.

Nash Sutcliffe Efficiency Index (NSE)

This criteria is widely used in hydrological models and it can be written as

$$NSE = 1 - \frac{\sum_{n=1}^{N}(\hat{y}(n) - y(n))^2}{\sum_{n=1}^{N}(y(n) - \mu_y(n))^2} \tag{2.35}$$

where $\hat{y}(n)$ is the estimated value, $y(n)$ is the measured data and $\mu_y(n)$ is the mean of the data. This metrics is widely used although it presents some limitation and should not be used solely to evaluate the model. [34]

2.4 Summary

To summarize the concepts presented in this chapter, it is important to then explain how they will be applied in this study. Both ARMAX and MLP models will be compared for the same data set and objective. The selection of the important variables will be made using the MLP models given that it is capable of both linear and non linear modelling. For the MLP, model parameters will be optimized by the use of hyperparameter tunning. For the ARMAX, the AIC will be the decision criteria used to

choose the order. After selecting the models with the best performance, the complete system will then be validated with all models in series.
In this study, two programming languages will be used: Pyhton (3.7.3 and 3.7.4) and Matlab (R2020a). Python will be used for the data processing, ANN modelling, hyperparameter tunning and final validation with all models. Two integrated development environments, (IDE) were used, Spyder and Visual Studio Code. The more relevant libraries used in pyhton are:

- tensorflow - version 2.4.0 [35]
- keras-tuner - version 1.0.1

Matlab will be used for the ARMAX modelling, with the System Identification toolbox. [36]

Chapter 3

Models Implementation

Upon understanding the literature regarding the problem at hands in chapter 1 and the main theoretical concepts in chapter 2, this chapter is concerned with the preparation and analyses of the data for model implementation. The chapter is divided into three main sections. The first is concerned with the pre-processing of the data from the initial available format to a model ready data set. Secondly, a brief statistical analyses of the data is presented. In the third section the feature analyses will be presented which relates to the process of selecting which information, is more relevant to consider and which is redundant. Regarding this final section, two approaches will be used: one comparison using paired t-test and Pearson coefficients and another of trial and error. The discussion of the models used and its results will be reserved for the following chapter. For this chapter only the Spyder IDE was used with python version 3.7.3 and 3.7.4.

3.1 Pre-processing of Data

In modelling systems, the majority of the work consists in the data processing and selection of the important data to consider. In this section the pre-processing of the data will be described.

Reservoir Data

The data was already in the daily time series format, and the non existing data was simply skipped.
In terms of clear outliers, only the dams of Brandariz and Touro had impossible values in the time series. Fortunately, those values in the case of the Brandariz Dam were outside the selected window and corresponded to values of zero in the height of the dam which is not possible since it is presented in relation to sea level. In the lowest point, the reservoir has a height of 143 meters above sea-level,[5].
A similar problem was observed in the Touro dam, however, this was within the selected range. An example, was the height in the year 2015, figure 3.1.
As can be seen in the figure 3.1, there are three sudden one day long spikes in the graph to a value of $154,75m$. This value corresponds to the height of the crest of the dam therefore this could not be the height of the water level. These values were therefore removed.

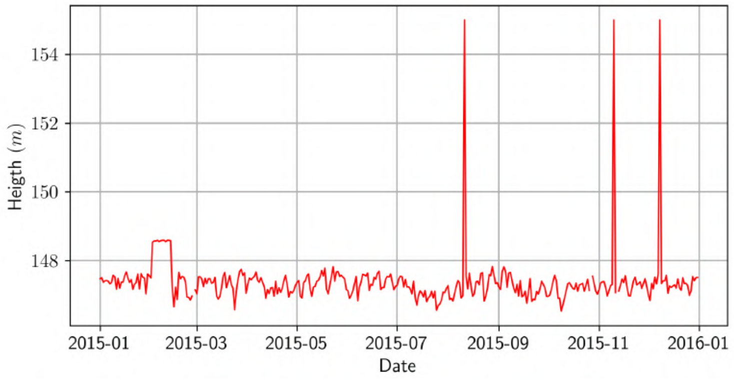


Figure 3.1: Touro dam height measure in 2015

The data that can be used is summarized below.

- Storage information
  - Height in m above sea level
  - Volume in hm^3
- Inflow approximation in m^3/s
- Outflow information m^3/s - Total flow
  - Global outflow
  - Separate flow (turbine flow, flood flow and bottom gate flow)

The dams are all restricted by the ecological flow, as discussed in section 1.3.1. Because this information is an imposed rule there is no point in trying to model such a consistent model dynamic. For this reason, the ecological flow was subtracted from the outflow.

The data was also normalized between 0 and 1 as presented in the table 3.1. In the table, the min and max value correspond to the physical limitations. The 0 and 1 columns are the values used to normalize looking at the minimum and maximum values observed in the data as well as the physical limitation of the dams.

Table 3.1: Normalization rules, dam data

	Portodemouros				Brandariz				Touro			
	Min value	Max value	0	1	Min value	Max value	0	1	Min value	Max value	0	1
Height (m)	220	254.69	225	260	157,4	170	158	170	143	153.4	140	150
Volume(hm^3)	54.5	297	80	300	0.327	2.74	0.4	3	2.21	6.037	2	6
Inflow (m^3/s)	0	-	0	300	5	-	5	230	5	-	5	230
Outflow - Eco flow (m^3/s)	0	-	0	250	0	-	0	230	0	-	0	230
Outflow Turbine - Ecof low (m^3/s)	0	121.5	0	130	0	85	0	90	0	55	0	60
Flood gate flow (m^3/s)	0	-	0	130	0	-	0	180	0	-	0	200
Bottom gate flow (m^3)	0	155	0	10	0	24.20	0	7	0	19.67	0	7

Meteorological Data

From the website [8], the data was extracted one variable at a time in comma separated-values file. The value -9999 indicated the missing data. The data presented no clear outliers. In the same away as before, the data was normalized taking into account physical limitation and maximum (max) and minimum (min) values present in the data.

Table 3.2: Normalization rules, meteorological data

	Min Value	Max Value	0	1
Temperature (C)	0,6	26,80	0	30
Humidity ($\%$)	38	100	35	100
Precipitation (mm)	0	106	0	110
Solar Irradiation ($kJ/m^2.day$)	33	3340	30	3350
Pressure (hPA)	927,70	1031,20	920	1040

Flow stations

The information of the hydrometric stations, unlike the previous data set, contained a column of data quality. That column contained information such as: data with no quality control, correct data, probably correct data, incorrect data susceptible to be corrected, incorrect data and not extracted

Only the data points described as correct were considered. To normalize, the same process was used, here being only based around the minimum and maximum values presented in the time series for both stations. In the table 3.3 the information used for normalization is presented.

Table 3.3: Normalization rule, flow station data

	Min Value	Max Value	0	1
Flow (m^3/s)	0,5	349,8	0	350

3.2 Data Analyses

3.2.1 Reservoir and hydrometric data

The first step consists on visualizing the distribution of the data. For the case of flow data the histogram tended to be very similar between inflow, outflow or hydrometric station. An example of Portodemouros reservoir is presented in figure 3.2, which displays a pronounced left skew.

When observing the histogram of the precipitation it displays the same left skew distribution, figure 3.14, naturally one could assume this is a major influence for the flow rate.

For the storage information, both height and volume are expected to have similar distribution for the same reservoir, figure 3.3 and 3.4. However, there is a clear non linear relation between both variables which is given by the geography of the reservoir. This function of volume is presented in the document for exploration and displayed in figure 3.5.

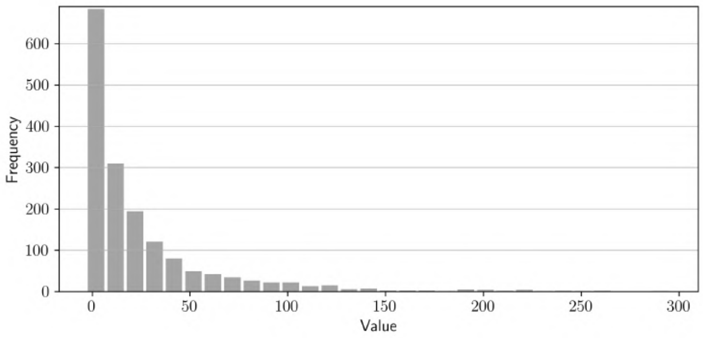


Figure 3.2: Histogram of inflow Portodemouros

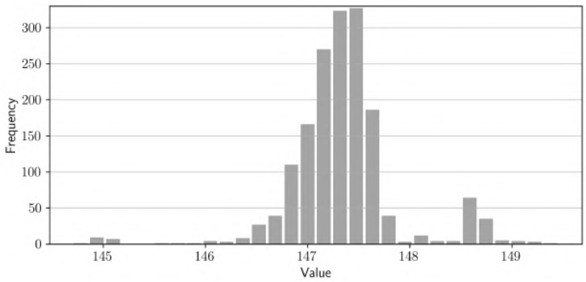


Figure 3.3: Histogram of heigth Touro

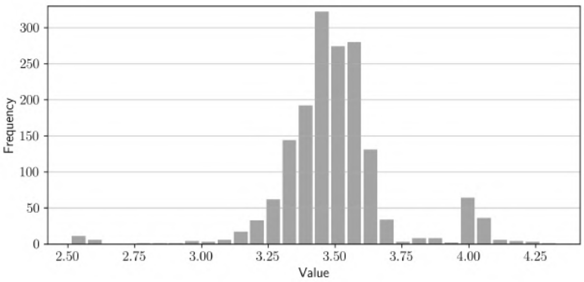


Figure 3.4: Histogram of volume Touro

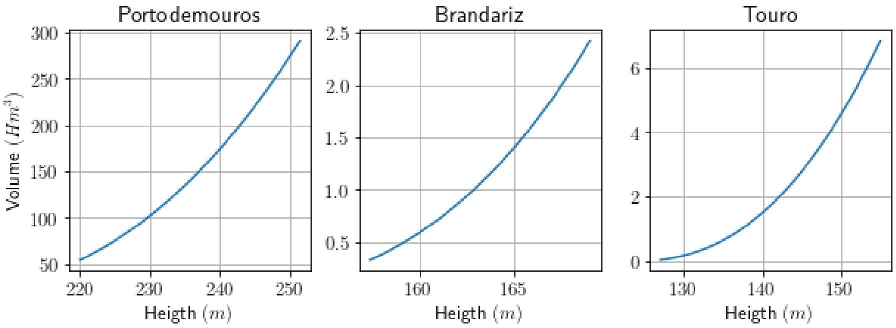


Figure 3.5: Function of volume in terms of height

As it can be seen, all three of the dam follow a 'funnel' like geometry, for increasing height there is more surface area resulting in more volume for every meter. In figure 3.5 it can also be seen the major differences in capacity, resulting in different storage variable distribution between dams. Storage information can be considered an important managing variable which is kept in check. Touro and Brandariz dams display similar histogram (figure 3.3) with an unimodal fairly symmetric distribution. In a time series analyses of these variables it can be seen that the values is maintained around an average value, figure 3.6.

Unlike the behaviour described for Brandariz and Touro, Portodemouros had a much slower response dynamics, figure 3.8. The histogram of height , figure 3.7, displays a greater variety of values. These observation are consistent with the thought that in the dam system, Portodemouros dam is the main reservoir, responsible for storage and the others are used mainly to extract additional energy.

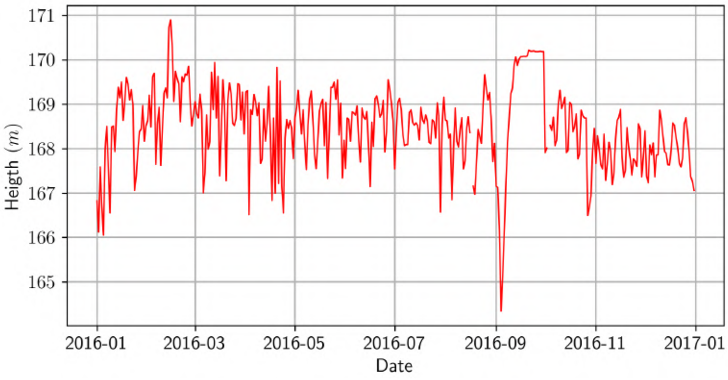
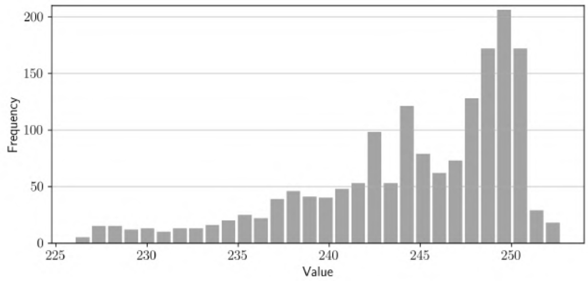


Figure 3.6: Brandariz height timeseries year 2016



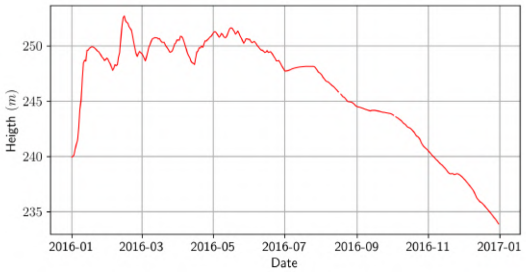
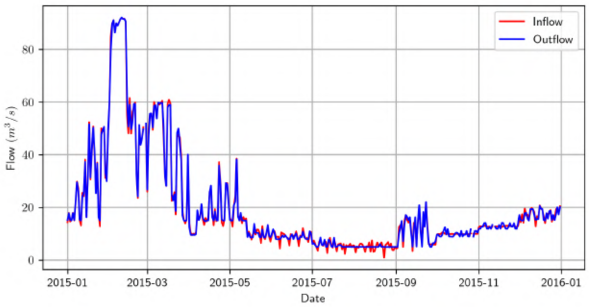


Figure 3.7: Histogram of height Portodemouros

Figure 3.8: Portodemouros height time series year 2016

To reinforced this ideia, figures 3.9, 3.10 and 3.11 represent the inflow time series overlapped with the outflow. For Brandariz and Touro the overlap presented very similar values, unlike Portodemouros where the values are quite distinct.



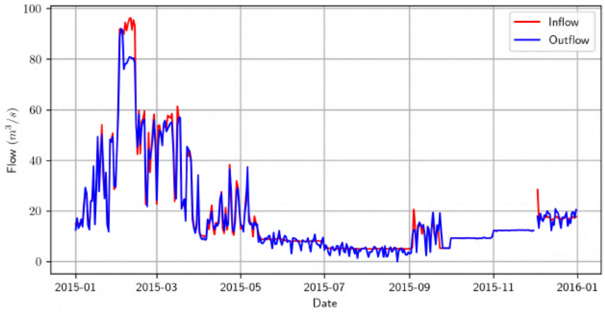


Figure 3.9: Inflow and Outflow, year 2015 - Touro

Figure 3.10: Inflow and Outflow, year 2015 - Brandariz

It is also important to understand the difference between the outflow of one dam an the inflow of another, given that the dams are close to each other. Figures 3.12 and 3.13 show the time series overlapped for both graph the Pearson coefficient and RMSE were calculated to decide if modelling was required or not. These values are presented in table 3.4 where can be seen that the strong similarities can exclude the need for modelling the difference. This is also reinforced by the fact, that no direct explanatory variables

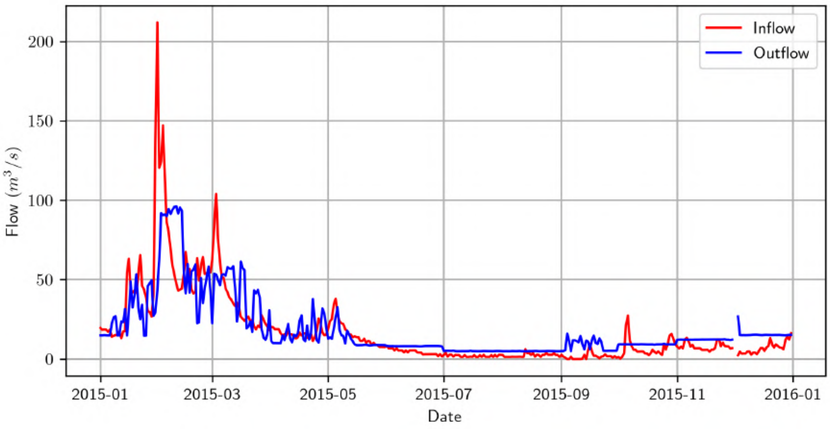
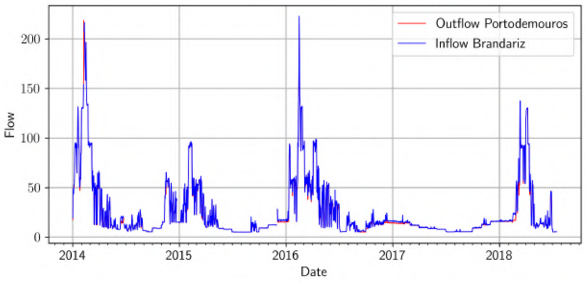


Figure 3.11: Inflow and Outflow, year 2015 - Portodemouros

could be obtained since no significant tributary exists between the outflow of one an inflow of another.



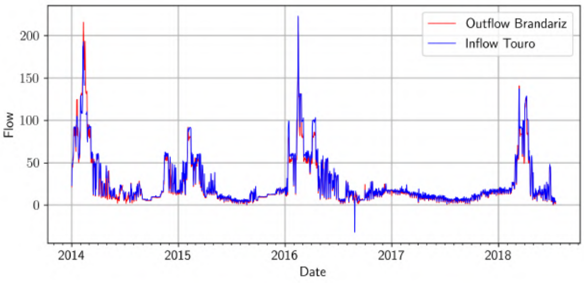


Figure 3.12: Outflow Portodemouros vs Inflow Brandariz

Figure 3.13: Outflow Brandariz vs Inflow Touro

Table 3.4: Differences between outflow and inflow

	Portodemouros - Brandariz	Brandariz - Touro
r	0.99969	0.98945
RMSE	1.41735	4.57817

After this analyses, it is expected that Portodemouros dams would be the harde to model, while the others were expected to be more simple.

3.2.2 Meteorological Data

Regarding the meteorological data, which will be used to model the inflow of Portodemouros a variety of interesting dynamics can be observed.

The precipitation, as it was mentioned in the previous subsection, has a left skewed distribution, with behaviour of zero in most days and a clear seasonal behaviour with the amount of rain more substantial in the winter months, figure 3.14 and 3.15.

The temperature displays an histogram with a somewhat bimoidal behaviour, suggesting the two mean temperature of the dry and the flood season, figure 3.16.

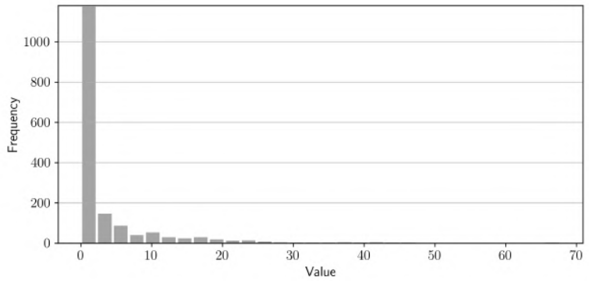


Figure 3.14: Histogram of precipitation, Arzúa Station

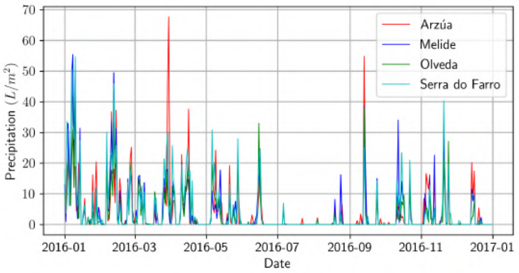


Figure 3.15: Time series precipitation, year 2016

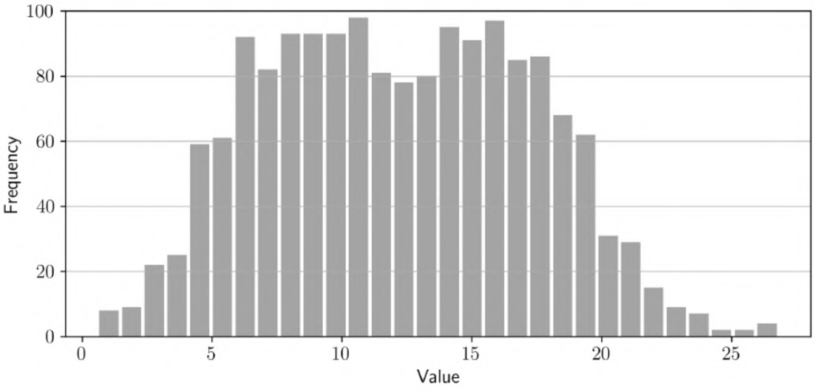


Figure 3.16: Histogram of temperature, Melvide Station

The seasonality continues to be a theme, again when analysing the solar irradiation values with a peak on summer and a low point on winter, figure 3.17. Figure 3.18 shows for 2015 a roiling average with 40 days to clean the time series and to better show the behaviour of the mean.

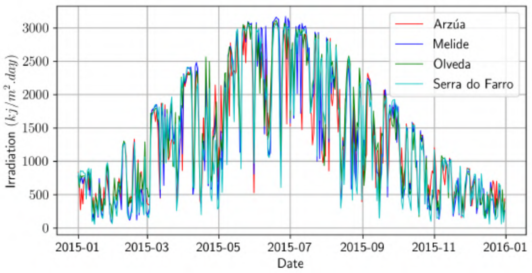


Figure 3.17: Times series solar irradidation, year 2015

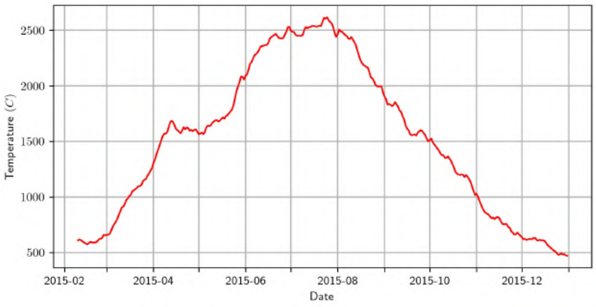


Figure 3.18: Rolling average graph solar irradiation Olveda Station

The remaining two variable to analyse is humidity and atmospheric pressure. Relative humidity displays a right skew distribution, figure 3.19. Atmospheric pressure shows a unimoidal distribution around a mean value,figure 3.20. In the time series plot of the pressure two observation can be made. Firstly as suggested by the histogram, the variable varies around a mean value without the strong seasonal effect observed in all other weather values. Secondly one of the factors that influence the value of pressure is altitude, so much so, that altitude is estimated in planes using a barometric pressure measure.

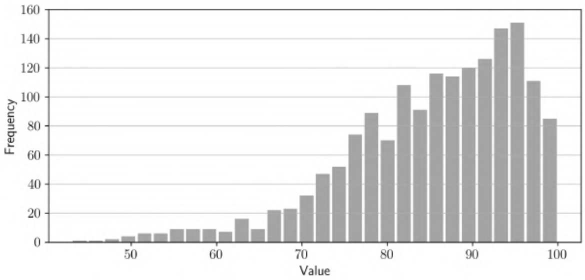


Figure 3.19: Histogram of humidity, Olveda station

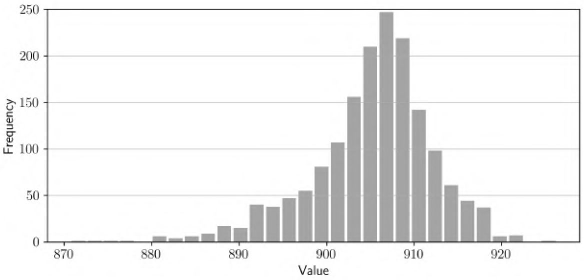


Figure 3.20: Histogram of pressure, Serra do Farro station

When observing the time series pressure values of multiple stations, figure 3.21, the values are very similar, only shifted up and down along the y axis with a direct correlation to the height of the meteorological stations, displayed in table 3.5.

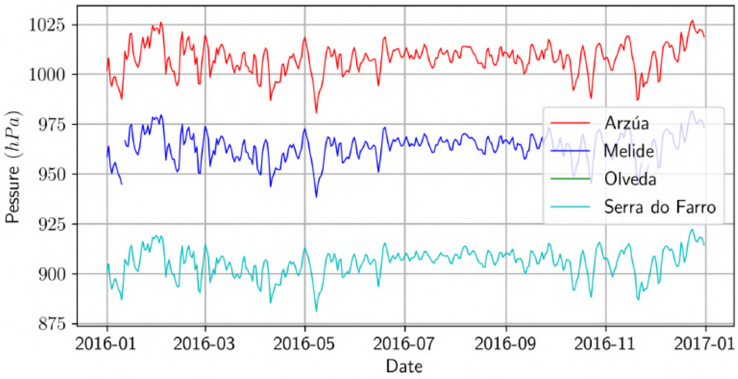


Figure 3.21: Time series pressure, year 2016

Table 3.5: Altitude of meteorological Stations

	Altitude (m)
Arzúa	362
Melide	477
Serra do Faro	780

All these variables are relevant for the modelling of the weather effects such as rain, evaporation or even factors such as soil infiltration. A meteorological systems is complex with a multitude of non linear relationship. It is then, expectable, for the model whose objective is to forecast inflow, that the ANN approach will produce better results than the ARMAX.

3.3 Feature Analyses

Feature selection methods consist of selecting a subset of variables from the available explanatory data. The idea is to avoid the use of redundant or irrelevant data. The rise of machine learning application was accompanied by bigger and bigger data sets meaning that feature selections algorithms became more and more important. In this study no specific algorithm was followed but ideas from both filter and wrapper methods were used.

3.3.1 Inflow Forecast

The inflow forecast problem is the one with more explanatory variables: all weather variables, hydrometric stations flow rate and previous values of inflow. The objective of this first model is to determine the inflow to Portodemouros. To determine how many lags to consider, the partial autocorrelation function (PACF) was used. This function obtains the value of the correlation of a variable with a lagged version of itself. This is useful in auto regressive modelling since it gives a criteria on how many lagged signals to use [21]. Based on the analyze of Portodemouros inflow PACF (Figure 3.22) it was decided to use a single lag.

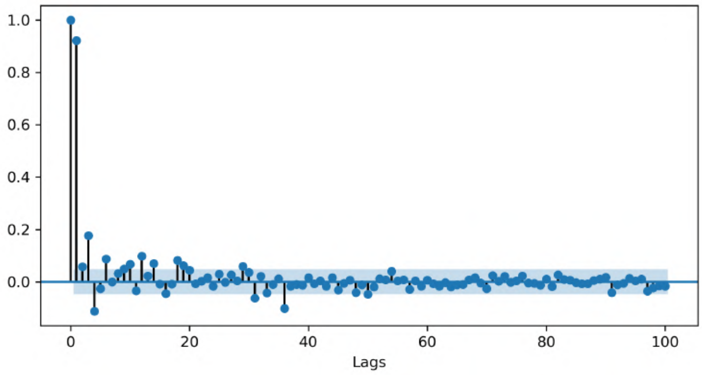


Figure 3.22: PACF of Portodemouros inflow

Statistical tests

Filter methods of feature selection consists of using statistical tests to rank variables against the target variable. Then, the obtained values is compared to a standard, below which features are removed. The name filter comes from the fact that this method is applied before the use of the model. [37] On this study, an idea was implemented to compare the data between the meteorological stations. Aligned with the idea of filtering but with the goal of reducing the number of weather variables by comparing them with each other, between the four stations. The idea was to compare variables of the same type, for instance compare temperature from all stations to see if the information is significantly different to justify the use of four values of temperature. The aim was to understand, if some data between stations was similar enough to consider the average or eliminate. To define significant similarity both the Pearson Correlation and a paired t-test were conducted. The paired t-test consist on a test to understand if the mean difference between two sets of observation is zero. From this test a p-value is obtain, which determines the probability of the observed result under the hypothesis of equal mean. The lower the value means that the observed data is rare under the hypothesis. Generally, the minimum value is defined at 0.05, meaning that p-values lower than that have statistically significant differences. [38] For the Pearson correlation a value of 0.95 or higher was considered to be similar variables.

For every year and type of data two table were constructed. One with the p-value and the other with the Pearson coefficient between all stations. An example for the temperature, year 2015 is presented in the table 3.6 and 3.7. In those tables, green values are considered acceptable and red value unacceptable, to the goal of finding similarity. The Met x cells correspond to the 4 meteorological stations.

Table 3.6: p-value, temperature measure 2014

	Met_1	Met_2	Met_3	Met_4
Met_1	-	$2.34x10^{-7}$	$2.16x10^{-13}$	$5.66x10^{-27}$
Met_2	$2.34x10^{-07}$	-	0.023437	$1.50x10^{-09}$
Met_3	$2.16x10^{-13}$	0.023437	-	0.000107
Met_4	$5.66x10^{-27}$	$1.50x10^{-9}$	0.000107	-

Table 3.7: Pearson Coefficient, temperature measure 2014

	Met_1	Met_2	Met_3	Met_4
Met_1	1	0.961523	0.954134	0.886567
Met_2	0.961523	1	0.992205	0.967535
Met_3	0.954134	0.992205	1	0.97038
Met_4	0.886567	0.967535	0.97038	1

An interesting result are the table for the atmospheric pressure. In the previous section, it was concluded that the pressure within this area only varies greatly with the altitude. The Pearson Coefficient was high all across the board as expected, given that it is essentially the same signal shifted up or down. However, because the paired t-test is a test of the difference in means, the results were quite close to zero meaning that, in order to observe those results the hypothesis is almost certainly untrue.

Variables which pass both tests were average, the reaming were not, except for the barometric pressure which were so similar in dynamics that were averaged anyway. Unfortunately, when tested with an MLP model this selection preformed worst on all metrics when compared to a control test, containing all possible variables, figure 3.23.

A number of factor could be attributed to this result:

- Pearson coefficient considers linear correlation, others may be important
- Paired t-test assumes a normal distribution of the variables which is only verified in the case of pressure
- Other important relationships between variables were not considered

Given the results, another procedure using a test model to evaluate subsets of features was considered.

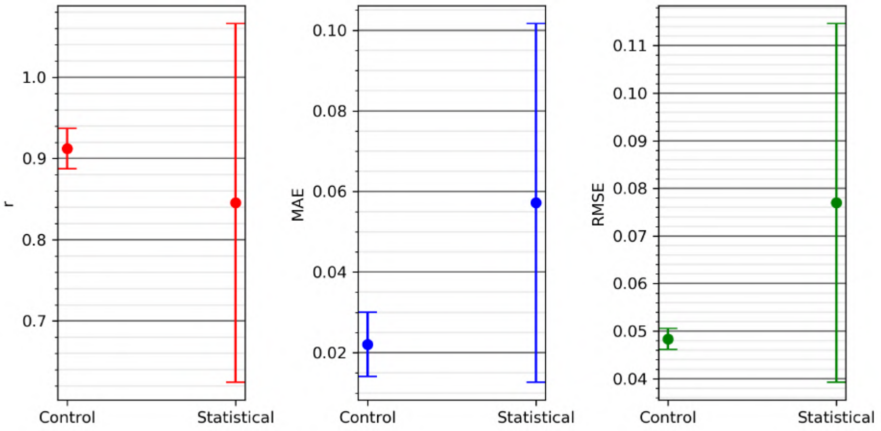


Figure 3.23: Control features vs Statistical tests conclusions

Wrapper style method

Wrapper methods consists in evaluating subset of variables with a model and identify the best performing model. With more and more features, it becomes harder and harder to perform an extensive search. To accommodate for this, either sequential selection methods are use or heuristic search algorithms. Sequential selections consists in adding or removing one feature at a time in order to obtain the better performing model. Heuristic search algorithms evaluate different subsets, through the use of optimization algorithms such as Genetic Algorithm, Particle Swarm Optimization. These methods, as expected, obtain local optima. [37]. For this study, no optimization algorithms was followed but some analysis of selected subsets were considered to identify a better set than the full data set.
First important task is to determine our search space, figure 3.24 shows a schema of the inputs to be considered.

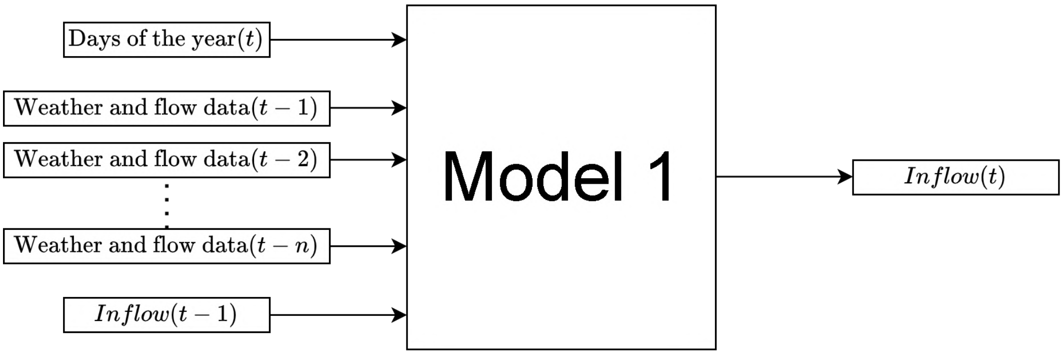

Days of the year(t)
Weather and flow data$(t-1)$
Weather and flow data$(t-2)$
Weather and flow data$(t-n)$
$Inflow(t-1)$
Model 1
$Inflow(t)$

Figure 3.24: Model 1 Schema

In the figure 3.24, n corresponds to the days of delays on the weather and flow variables. The weather and flow data is:

- Temperature - Arzúa, Melide, Olveda and Santo do Farro Stations
- Humidity - Arzúa, Melide, Olveda and Santo do Farro Stations
- Humidity - Arzúa, Melide, Olveda and Santo do Farro Stations

- Precipitation - Arzúa, Melide, Olveda and Santo do Farro Stations
- Pressure - Arzúa, Melide and Santo do Farro Stations
- Flow rate - Stations 544 and 546

The weather variables can either be discarded, average or all considered. The flow stations can either be considered or not. The subsets will all be tested with an MLP given the ability to model non linear functions. For this implementation, the structure of the MLP was as follow:

- Two hidden layers
- First hidden layer contained 50 neurons[1]
- Second layer contained 20 neurons
- RMSProp algorithm
- Learining rate = 0.0005
- Momentum = 0.2
- Epochs = 450
- Batch size = 50

For every tested features subset, 20 trial models were tested, to minimize the effects of the random initial weights of the network. The data set was divided into:

- First 70% data for training
- 10 % for validation
- The last 20 % of data for testing

To start the most relevant variables in the search space is the number of days delayed, n and for this analyses the first hidden layer is going to be set the same amount of neurons as the corresponding input. Delays from 1 to 10 were tested, considering the full data set, this meant a input sizes from 23 up to 212. Figure 3.25[2] shows the results in which the best value over all the metrics was 3 days of delay.

All the subsequent tests, consider 3 days of delayed values. Various other hypothesis were tested, the full results are presented in the appendix A. However the two most relevant operation were: to average types of weather variables one at a time (figure 3.26) and removing one type of variable at a time(figure 3.27).

In figure 3.26, it is clear that the removal of variables which improved the results were the temperature data and Station 546. Figure 3.27 shows less clear results, averaging all variables results in a quite better pearson coefficient but it produces higher values of MAE or RMSE. The variable which cannot be

[1] Except the delay testing

[2] The dot represents the mean an the interval shows the standard deviation

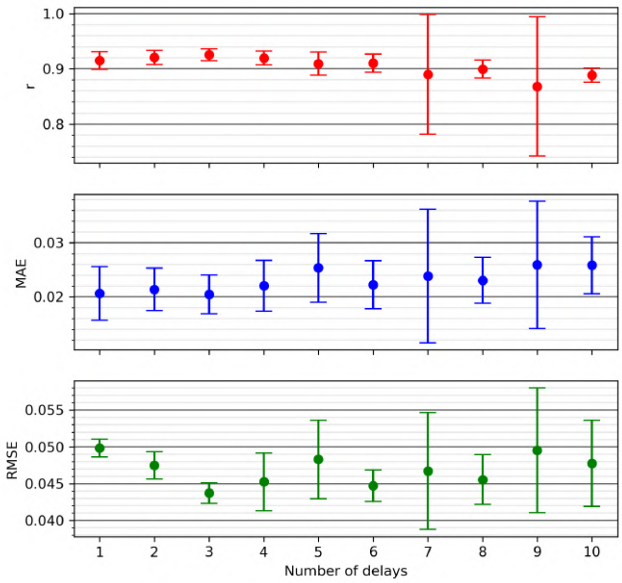


Figure 3.25: Model 1, number of delayed variables

average is clearly the precipitation values.

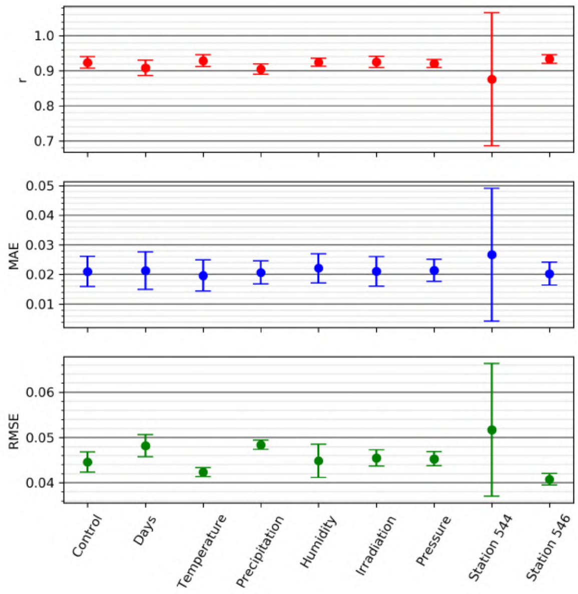


Figure 3.26: Resutls from removing one type of variable at a time

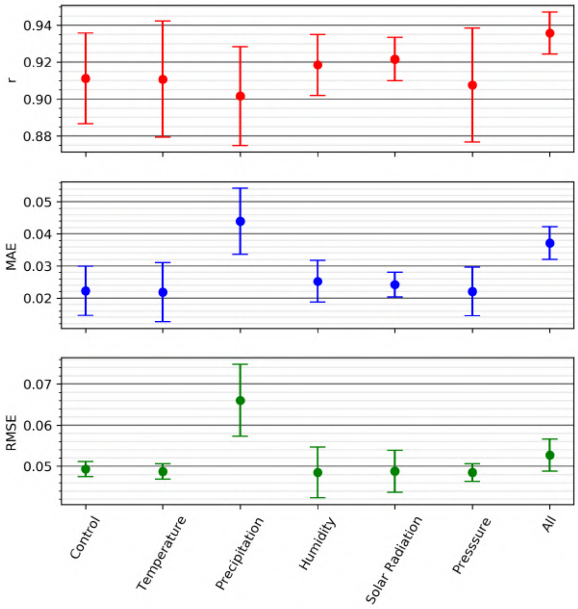


Figure 3.27: Resutls from averaging one type of variable at a time

From those conclusions, a final test of subset hypothesis were made, figure 3.28:

- Control, considering all features
- Averaging very type of weather variables, except precipitation (**AV-Precip**)
- Averaging very type of weather variables, except precipitation and temperature (**AV-Precip**/**Temp**)
- Removing both temperature and station 546 (**RM-Temp**/**St546**)

- Removing only temperature (**RM-Temp**)
- Removing only station 546 (**RM-St546**)
- Removing temperature and averaging the rest except the precipitation (**RM-AV**)

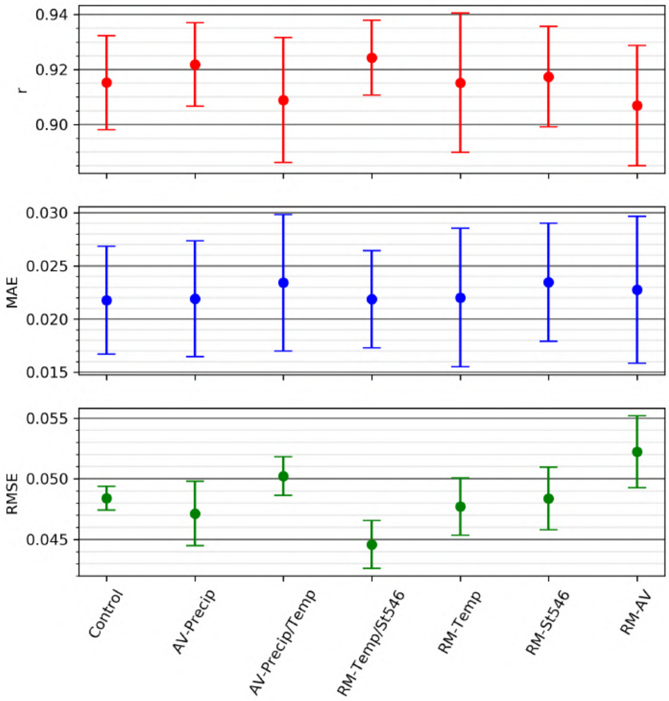


Figure 3.28: Final tested feature subset, model

It is then clear that the best subset is the full data set minus the temperature variables and the flow rate from station 546. Station 546 is in fact located on a tributary, unlike station 544 located at the Ulla river. This difference in location could explain the lack of important information.

3.3.2 Outflow Forecast

For the outflow forecast, the first step was to observe the PACF of the outflow, which were very similar to the results of the inflow. Obtaining, approximately the same results, resulting in the choice of considering one day delayed outflow as an input for all 3 dams (plots in appendix A).

For the models related to the managing of the dam systems the process was similar to the one taken by the model to predict inflow (Model 1). To help define the search space, figure 3.29 shows a schema of the outflow models (Model 2,3 and 4).

To model each dam behaviour a second smaller model will be implemented to calculate the remaining storage levels, this will be described as a secondary model.

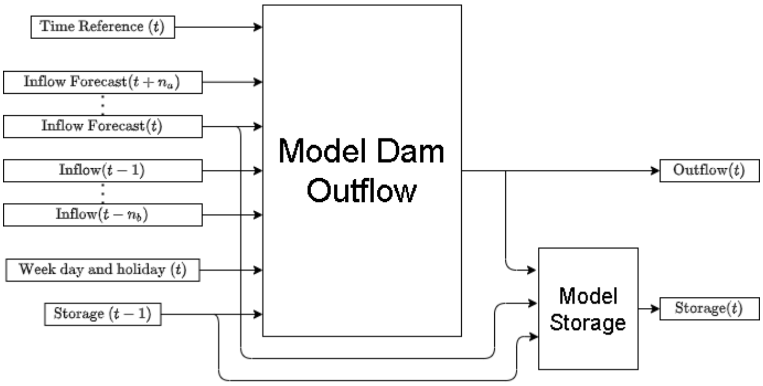


Figure 3.29: Outflow Models Schema

Main model

Lets then define the possible variables for the main model. The most extensive feature is in fact, the inflow either in 'prediction'[3] or in delayed version. The choice is then how many days of delayed version and how many ahead. The remaining variables choice can be summarize in the following way:

- Reservoir Storage
  - Height (m)
  - Volume (Hm^3)
- Time reference
  - None
  - Month
  - Day
- Week day and holiday information
  - Binary variable with 0 for normal working day and 1 weekend or holiday
  - Separate Binary variables for weekday/weekend and holiday
  - Full description of week day from Monday(0) to Sunday(1) with 1/7 increments
- Outflow
  - Total
  - Separate (Flood, Power and Bottom)

Considering both that, the bottom gate is seldom in use, and that the power and flood gate are related

[3] The real values were use in selection and tuning of the models

by the limitation of the turbine, a decision was taken to only use in this section the global outflow.

The MLP structure and training variables used were the same as the in previous model, except the number of neurons in the hidden layers was reduced to 10 [4].

The first test was to change the multiple combination of inflow variables. For the same remaining features, 5 days ahead $(t, t+1, t+2, t+3, t+4)$ and 5 days of delay were tested $(t-1, t-2, t-3, t-4, t-5)$. The results similarly to the previous section were in three metric (r,RMSE and MAE). For the Portodemouros dam (model 2), the results are presented in figure 3.30[5].

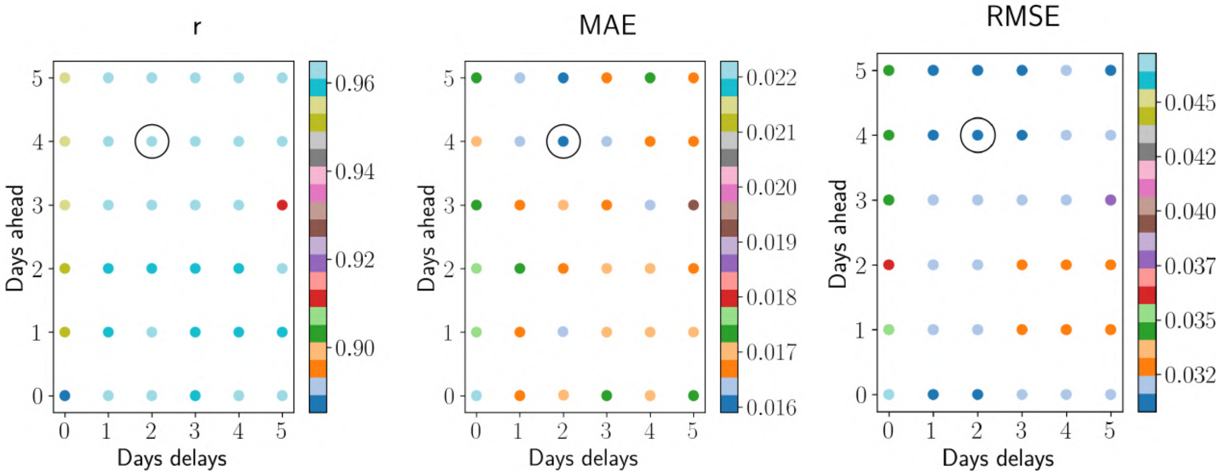


Figure 3.30: Test of different days for inflow Portodemouros

The black circles, in figure 3.30, represent the best possible solution for each metrics. Despite the results, pointing to 2 days of the delay and 4 ahead the values of 2 days before but only one ahead were chosen, in order to minimize the number of inflow value necessary to predict from the previous model. The results were still quite satisfying with the chosen values.

For Touro and Brandariz the process was exactly the same and the plotted results are present in appendix A, the results are as follows:

- Brandariz - 1 day delay, 1 day ahead
- Touro - 2 days delay, 1 day ahead

Moving now, towards the remaining possible subsets, apart from the outflow the full list of combination were tested for all three dams. The full plot of the results is presented in appendix A

After analysing the results the best features were selected, table 3.8.

Secondary model

This secondary model is necessary because of the phenomena like , leakage and evapotranspiration. Taking the equation 1.3 presented by Yang et al. [9], it might be necessary to consider not only the inflow

[4] except for the calculation of the inflow days in which the first layer was set the same as the number of inputs

[5] The values are the mean of the tests for each combination

Table 3.8: Results of the feature selection, Outflow models

	Portodemouros Model 2	Brandariz Model 3	Touro Model 4
Time reference	Month	Month	Day
Storage	Heigth (m)	Volume (hm^3)	Heigth (m)
Week day	Full description	Binary	Combined
Holiday Info	Binary	Binary	Combined

and outflow at time t but also at a previous time step $t-1$. A comparison between the two options is presented in figure 3.31 and 3.31.

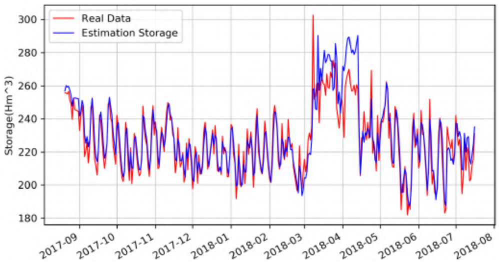


Figure 3.31: Test response of secondary model 3 with 1 time delay

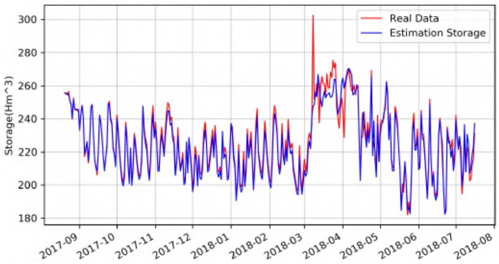


Figure 3.32: Test response of secondary model 3 with 2 time delay

In terms of the storage units it makes logical sense that the volume, which considers the shape of the reservoir, would present better results than the height. This is true for the Touro and Brandariz (figure 3.33 and figure3.32) due to their faster dynamics. For the slow response of the Portodemouros both metrics were comparable in results.

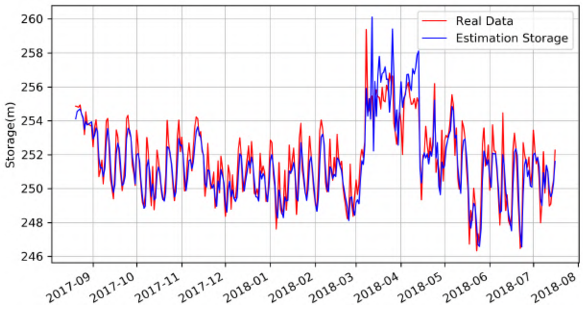
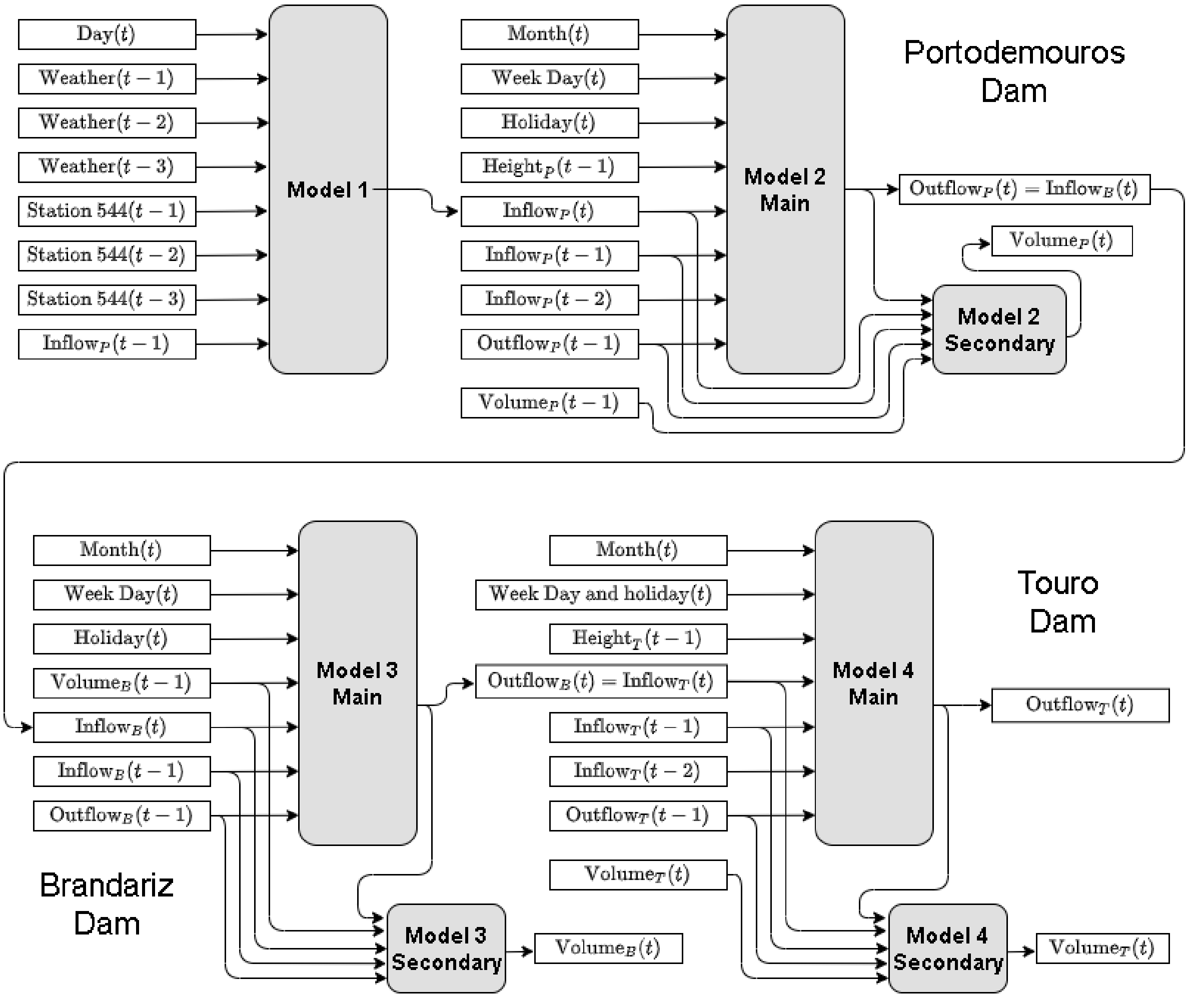


Figure 3.33: Test of model 3 with volume

To summarize the global results, an extensive diagram shows the entire system and variables selected, figure 3.34.



*For the ARMAX implementation the delays are different

Figure 3.34: General Graph

Chapter 4

Results

In this chapter the implementation and results of the various models is presented. Firstly, the results of the inflow model (Model 1) will be described followed by the reservoir models (Model 2,3 and 4). After presenting the modelling results a comparison between the two modelling approaches will be performed and subsequently a discussion between the models themselves. The final part of the chapter is concerned with the full system implementation and the propagation of the error across the entire system.

4.1 Inflow Forecast Model

4.1.1 ARMAX

For the ARMAX implementation the system identification toolbox of MATLAB was used. For the training 70% of the data set were used, and 30 % for validation. The variables used were the same as described in section 3.3.1, with the difference being the delays of inputs which will be calculated with the selection of the best order for the model.
To select the best order of the ARMAX model the following procedure was performed for all models.

1. Implement an ARX model with the order for all three parameter (autoregressive , input, pure delay) ranging from 1 to 10.
2. Select the optimal order for the ARX with the AIC criteria.
3. After the ARX model selected, the order for the moving average of the ARMAX model will be tested, with a range from 1 to 3.
4. The best performing model is then selected.

To complete the missing data a zero order hold was implemented, given the data format . Both a normalized version and a originally scaled data set were tested. The originally scaled version outperformed the normalized data set for the inflow problem, the results for the non normalized version are presented in appendix B. The difference could be related to the way the normalization was done or some relevant information could be lost on normalization.

The results for the order selection are presented in figure 4.1. The best model, accordingly to the AIC criteria, as a pure delay of 2 and a regression order on the output of 3 and 1 for the inputs.

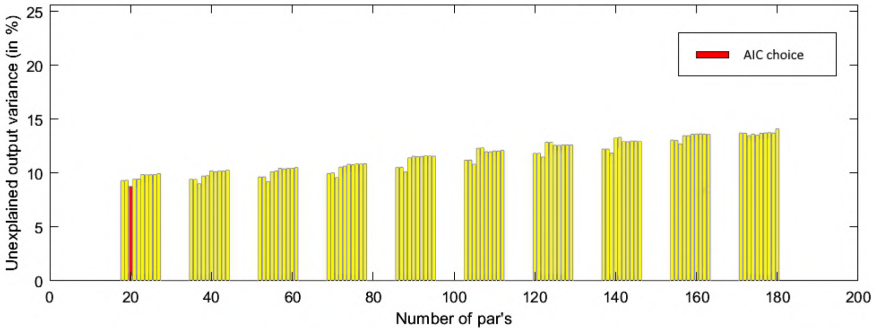


Figure 4.1: Misfit for order selection

Trying to model the residuals information by a moving average process (order form 1 to 3) only made the performance considerably worst and unstable, figure 4.2, values refer to the normalized RMSE.

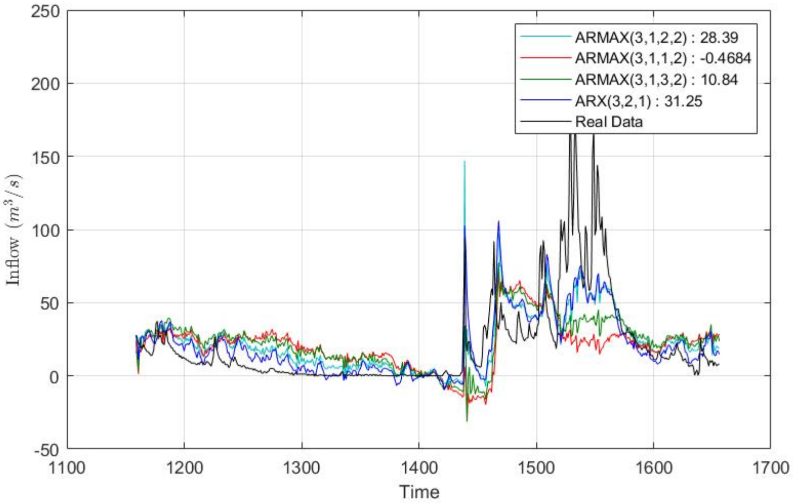


Figure 4.2: Tests with MA, model 1

This could be due to the fact that the residuals cannot be modelled as white noise, which indicates unmodelled behaviour beyond noise. The results of the best model were then exported and performance criteria such as RMSE, MAE and Pearson Coefficient were calculated [1]:

- RMSE - 22.127
- MAE - 12.935
- r - 0.757

[1] These were only studied for the same period as the ANN models so for 20 % of the data set. This is important because periods of flood or drought change the results greatly.

4.1.2 MLP

For the MLP modelling the features and delays values are exactly those described in section 3.2.1. The data was divided into 70% training, 10% validation and 20% testing. A feed forward, densely connected neural network with three or four layers was the choice for the ANN. The exact MLP structure was optimized with a hyperparameter tunnig, keras.tuneR version 1.0.1. The search space considered was:

- Learning rate - Options : 0.1, 0.01, 0.001, 0.0001 and 0.00001
- Activation function - Options: relu, tanh and sigmoid
- Number of neurons of first hidden layer - range from 10 to 50 with intervals of 2.
- Existence of second hidden layer
- If needed, number of neurons of second hidden layer - range from 0 to 50 with intervals of 2.

More parameters could have been optimized, however the convergence of the results became harder and harder with more parameters. Several hyperparameter runs were made with two different algorithms: Mini batch gradient descent and the RMSprop. The optimization used a Baysien optimizer with the MSE as the objective function. For each attempt 400 trials were performed with every trial being executed three times to minimize the effects of the random weights initialization. The first 50 trials were random to allowing enough point for the Gaussian distribution first approximation. The results of the best preforming models for each optimization algorithm are presented in table 4.1.

Table 4.1: Results Hyperparameter tuning, Model 1

Algorithm	Learning rate	Activation function	Nº neurons 1st layer	Nº neurons 2nd layer
RMSprop	0.001	Relu	46	22
Mini batch GD	0.1	Relu	50	24

For both algorithm the relu activation function was the best performing, however the slower mini batch gradient descent algorithm required a much higher learning rate, than the more adaptive RMSprop algorithm. The number of the neurons was the component of the search space which the tuning algorithm had the most difficulties to optimize, driving the number of maximum trials up.

To ensure the best model was saved for each algorithm 30 models were tested and the one with the best performance was saved[2]. For the training 500 epochs were used with a batch size of 50 data points. The fixed algorithm parameter[3] are:

- RMSprop
  - Momentum = 0
  - $\rho = 0.9$

[2] MSE criteria used

[3] Used for all subsequent models

    - $\epsilon = 1e - 7$

- Mini batch gradient descent

    - Momentum = 0

The results can be observed in figure 4.3 and the values in table 4.2. Despite, being a slim margin the best performing model was the one using the RMSprop algorithm.

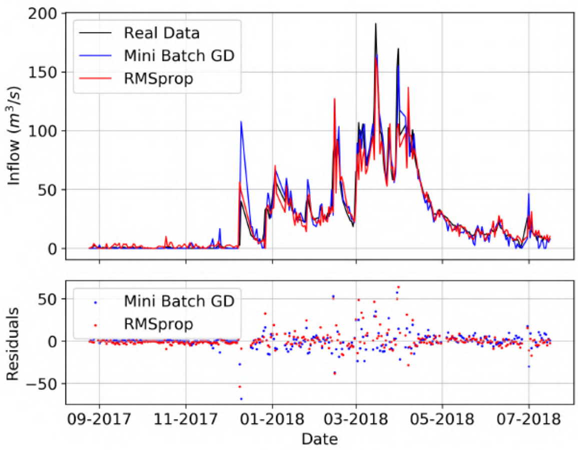


Figure 4.3: Model 1 testing with different algorithms

Table 4.2: Results for the different algorithms, Model 1

Algorithm	RMSE	MAE	r
RMSprop	9.777	5.188	0.957
Mini batch GD	10.404	5.187	0.952

4.2 Outflow Forecast Models

The outflow forecast models of the three reservoir followed the same produce as described above, for both the ARMAX as well as the MLP.

4.2.1 ARMAX

For Model 2 (Portodemouros) the data set was a year longer so it was split 75% training and 25% testing, with models 3 and 4 retaining the 70/30 split. Again, both a normalized version and a non normalized data set were used. A key detail however, was the inflow variable which in section 3.3.3 considered one day ahead for all three dams, to give the ARMAX the best possible chance, unlike the height variables set at $(t - 1)$ this one was set a time t.

Main model 2 had slightly better results in the originally scaled data set. The best performing model for Portodemouros was the ARMAX(5,1,1,4), figure 4.4.

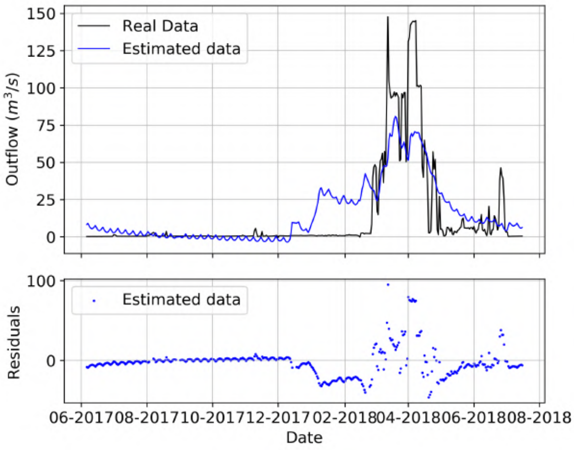


Figure 4.4: Testing results ARMAX, Model 2

The secondary model 2 unlike the main model performed slightly better with the normalized version. However for both data sets, the best model behaved with a clear problem of scaling, 4.5. For the secondary models of both Brandariz and Touro the model behaved quite poorly this might be due to the faster dynamics of these dams, results can be found in appendix B.

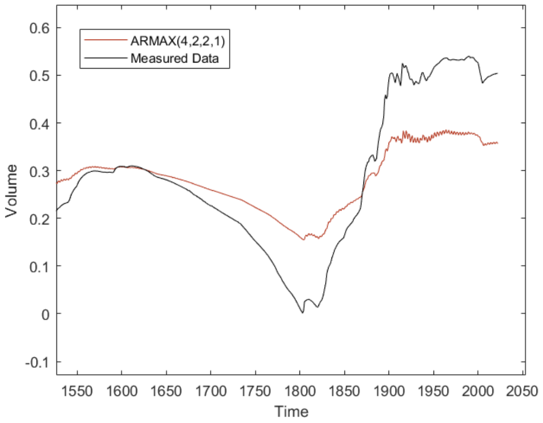


Figure 4.5: Testing results ARMAX, Secondary model 2

For the main models of Brandariz and Touro dams the results were quite similar. In both cases, the ARX model outperformed the ARMAX, both the normalized and non normalized models performed similarly. The results of the best performing models are presented in figures 4.6 and 4.7.

The dams with smaller reservoirs greatly outperformed model 2, since these are much more correlated with the inflow and less with the storage. However about the secondary models the comparison was the opposite, secondary model 2 performed somewhat decently, and for model 3 and 4 the results were unusable. Table 4.3 shows the results of the best main models[4].

[4] Error calculation for the normalized models are done after denormalization

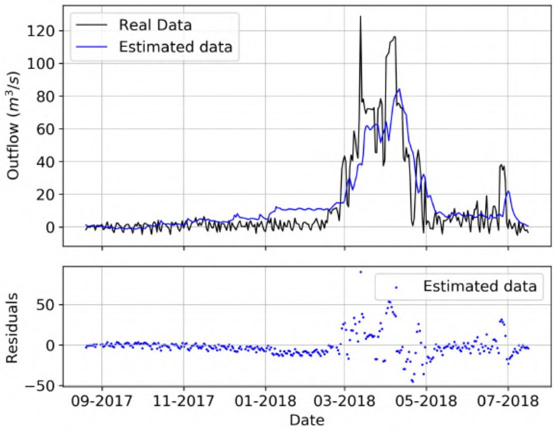


Figure 4.6: Results ARMAX, model 3

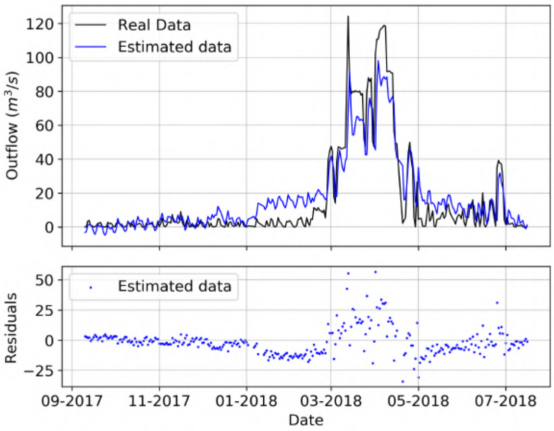


Figure 4.7: Results ARMAX, model 4

Table 4.3: Results best dam models with ARMAX

	RMSE	MAE	r
Model 2	18.478	11.154	0.817
Model 3	13.900	8.691	0.856
Model 4	11.372	7.846	0.936

4.2.2 MLP

For models 2, 3 and 4 a procedure was taken to determine a good implementation of the MLP models. Starting with the features selected in section 3.3.3, a hyperparameter tuning was performed with the following search space:

- Learning rate - Options : 0.1, 0.01, 0.001, 0.0001 and 0.00001
- Activation function - Options: relu, tanh and sigmoid
- Number of neurons of first hidden layer - range from 1 to 10 with intervals of 1.
- Existence of second hidden layer
- If needed, number of neurons of second hidden layer - range from 0 to 10 with intervals of 1.

The same two algorithms were tested and only the main models were considered, the results for the parameters of the best models are presented in table 4.4.

Interestingly, a pattern emerged between the algorithms, while the RMSprop benefited from the sigmoid function, the less efficient Mini batch gradient descent required the more efficient relu activation function. In the case of model 3, when both performed better with relu the learning rate is lower for the RMSprop algorithm than for the Mini Batch GD.

To compare the results between algorithms, the best performing model out of 30 training attempts[5] was stored, table 4.5.

[5] with the parameters described in table 4.4

Table 4.4: Results from hyperparameter tuning, models 2, 3 and 4

	Algorithm	Learning rate	Activation function	Nº neurons 1st layer	Nº neurons 2nd layer
Model 2	RMSprop	0.1	Sigmoid	10	-
	Mini Batch GD	0.1	Relu	10	-
Model 3	RMSprop	0.01	Relu	10	5
	Mini Batch GD	0.1	Relu	10	-
Model 4	RMSprop	0.1	Sigmoid	7	-
	Mini Batch GD	0.1	Relu	10	-

Table 4.5: Results for best performing MLP models 2, 3 and 4

	Algorithm	RMSE	MAE	r
Model 2	RMSprop	8.008	3.781	0.966
	Mini Batch GD	8.130	3.697	0.965
Model 3	RMSprop	2.769	2.055	0.994
	Mini Batch GD	2.239	1.746	0.996
Model 4	RMSprop	1.674	1.222	0.999
	Mini Batch GD	5.136	3.397	0.991

As expected the better performing models were those related to smaller dams with less focus on storage. Two models out of the three performed better with the RMSprop algorithm, however both models achieved similar results. Both algorithms are effective at obtaining similar performing solutions. The main difference across algorithms though, was the rate at which each one learned. All models used 500 epochs to train, however the RMSprop displayed better results with less training epochs. Figure 4.8 and 4.9 shows the training and validation as a function of epochs (Model 3 best performing for each).

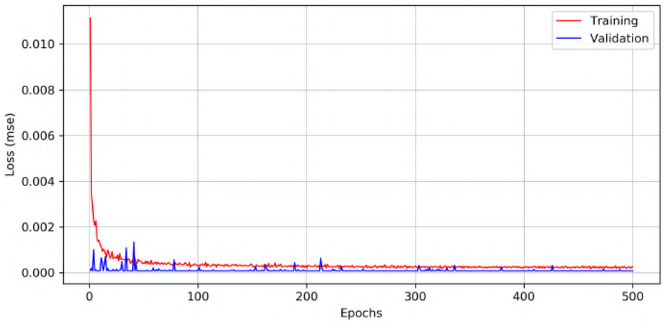


Figure 4.8: Training loss, Model 3, RMSprop

Despite the difference in y-axis scale, it can be seen that the RMSprop as a steeper descent in the error, figure 4.8. RMSprop at epoch 100 displays a very similar value to the final epoch, on the other side the mini batch GD model at epoch 100 is further away from the final value, figure 4.9.

Unlike the main models, the secondary models were not optimized by hyper parameter, given that the number of inputs were smaller and that increases in structure did not equate to much difference in performance.

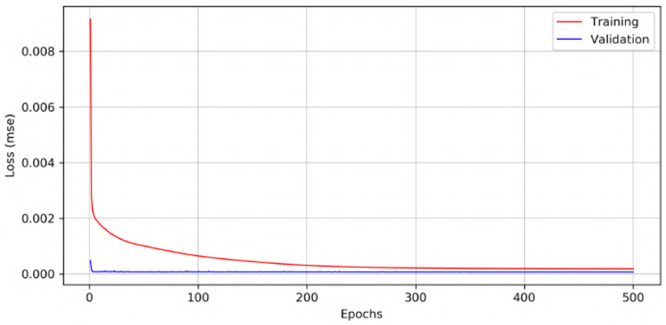


Figure 4.9: Training loss, Model 3, Mini Batch GD

The features used as inputs are as described in section 3.3.2 and the structure was the same for all three dams:

- 3 layer MLP
- 4 neurons on the hidden layer
- Learning rate = 0.001
- Training algorithm - RMSprop
- Batch size = 50
- Epochs = 500

As one could expect, the difference in behaviour discussed previously are also shown in the modelling of dam storage. For the slower model 2 the modelling is near perfect, figure 4.10.

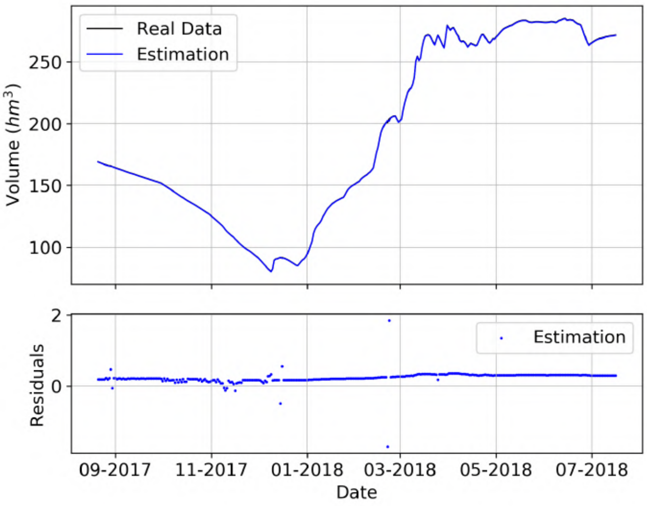


Figure 4.10: Secondary model 2 testing

For Touro and Brandariz the results for the secondary model were not as good as for the Portodemouros dam. Touro had better results, although both were satisfactory, figure 4.11 and figure 4.11.

The results of the secondary models best performance are expressed in table 4.6, it is important to note

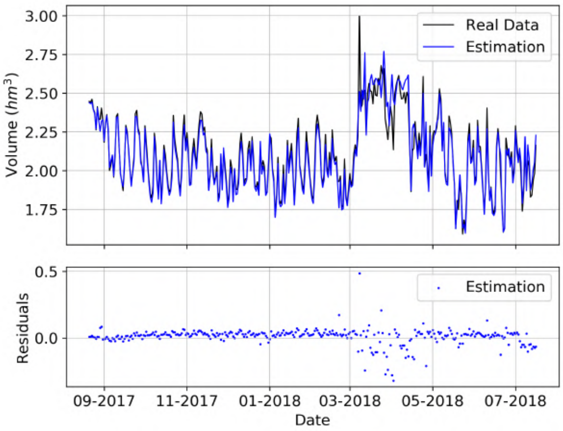


Figure 4.11: Secondary model 3 testing

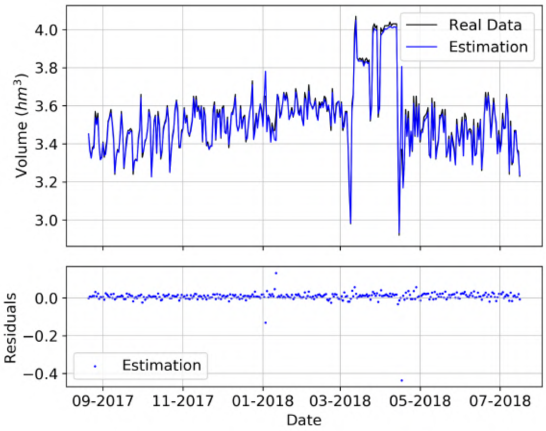


Figure 4.12: Secondary model 4 testing

Table 4.6: Secondary models, MLP, results

	RMSE	MAE	r
Model 2 Secondary	0.0011	0.0011	1
Model 3 Secondary	0.0125	0.0237	0.9655
Model 4 Secondary	0.0030	0.0039	0.9964

that the error measure here are of the normalized data.

4.3 Comparison between ARMAX and MLP

From section 4.1 and section 4.2 it can be seen that for all models the MLP performed better than the ARMAX counterpart. The ARMAX could have been further optimized with a more selective feature analyses for linear models, as well as an order selection personalized for every input. However, given the performance difference it did not seem justifiable to continue developing linear models into a system with non linear dynamics. Figures 4.13, 4.14, 4.15 and 4.16 shows the difference between the models.

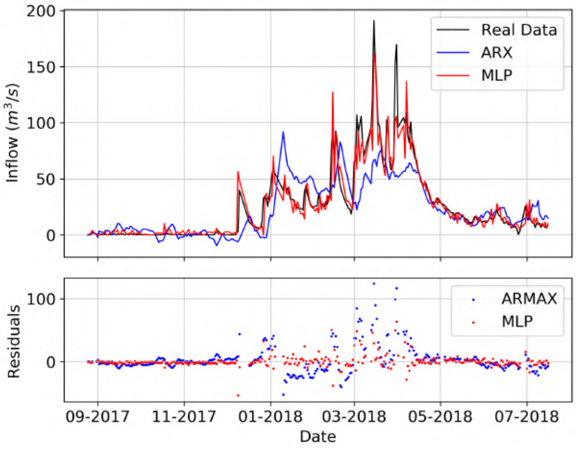


Figure 4.13: Model 1 comparison

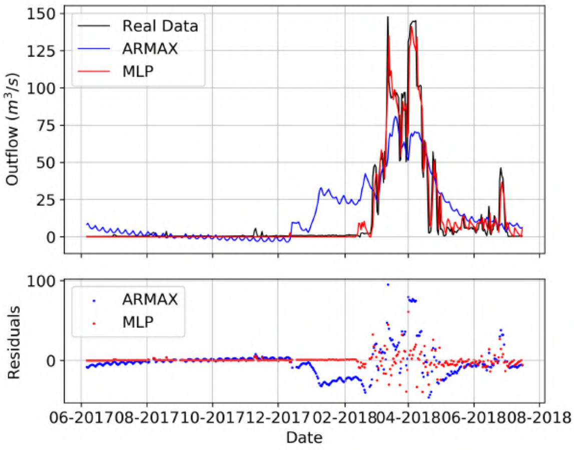


Figure 4.14: Model 2 comparison

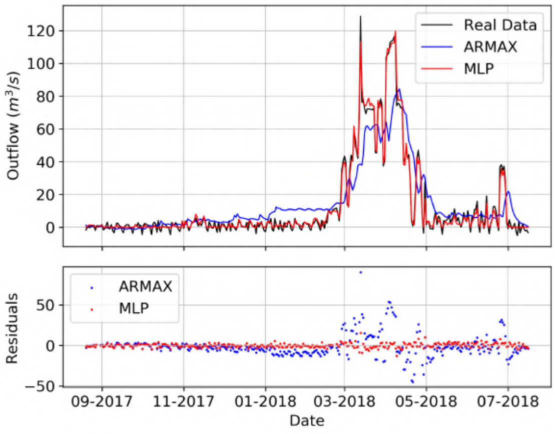


Figure 4.15: Model 3 comparison

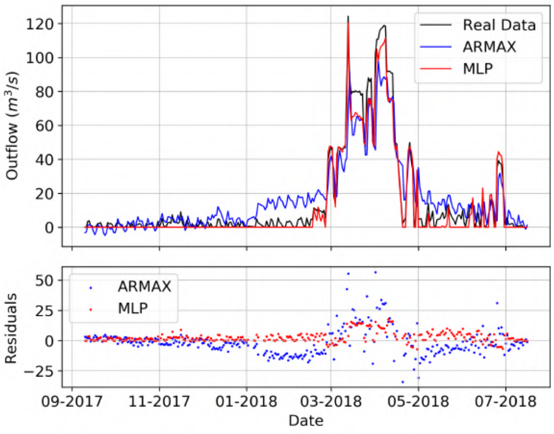


Figure 4.16: Model 4 comparison

The secondary models despite being relatively simple are clearly non linear in nature with much better results obtained for the MLP and poor results for the ARMAX attempts.

Although ARMAX models are simpler to train and more computationally efficient, for data set with sizes of this order (4/5 years daily time series) the training times are quite low. The MLP models were trained and simulated in a windows 10 machine with only CPU usage (Intel i7-10510u). Using Spyder IDE and keras models , with 500 epochs and batch size of 50, the computational times were:

- Model 1 - 10/11 s
- Model 2 - 12/13 s
- Model 2 sec - 12/13 s
- Model 3 - 10/11 s
- Model 3 sec - 9/11 s
- Model 4 - 10/13 s
- Model 4 sec - 10/12 s

For the full system integration it took just under a minute to simulate 330 data points, so time was not an issue, much more complex models with bigger sets of variables could easily be implemented with still reasonable time of training.

4.4 Consideration on the models

As expected, the model with worst results was model 1, which required a more complex structure. Models 2, 3 and 4 most of the time did not even require a second hidden layer.

To improve model 1 inflow prediction, more weather information could be useful as only 4 locations were available. When analysing several hypothesis, it was clear the importance of the precipitation on the model performance. This variable is subject to a variety of errors in measurement [39]. Rain gauges used in ground measure can have problems of saturation in heavy rainfall or evaporation on very light rainfall. More significant errors can occur when gauges are not properly maintained. Despite this, in actual simulation the model would have to work with predicted data if the goal is to predict several days ahead of time. Model 1 also uses data from an hydrometric station located in the river which would have to be estimated to predict values further ahead in time.

Model 2 did not perform as well as 3 and 4 given the nature of the reservoir. Two aspects were identify that could improve the model 2 results. One, being the lack of weather variable on the reservoir, like rainfall which increase the amount of water or the temperature, solar irradiation and humidity which affect the rate of evaporation. Second, being the details of electric prices which constrains the power production. The most important patterns are considered with the introduction of week day and holiday information. However the electric market works both on a day ahead pricing and a current pricing and updates during the day, meaning that a hourly time series could be better at understanding the variation of the outflow and the relationship with electric price variables.

Model 3 and 4 performed notably well, attempts to improve do not seem particularly necessary. The secondary models could possibly benefit from weather variables since again evaporation and rain is not accounted for.

All the isolated models successfully modelled each of the objective dynamics. In terms of the reservoirs the degree of success varies with the purpose of the storage.

4.5 Full Model Integration

The last tests to perform was to connect all the models in series and analyse the accumulated errors throughout the models.

To perform this simulation several assumptions were taken:

- The actual weather information and the flow rate from station 544 were used, instead of predicted values.
- The test data had the same size for all model, except model 2 which had more data. To test the system the smaller size was used on all models.
- When training, the models incomplete data was ignored but for testing the entire system a zero order hold was performed to complete the missing data.

Other details had to be taken into account to make the system function correctly:

- Because the secondary models used volume and the main models used height a converter was created using the data presented in the [1, 5, 6]. The document contained information for every possible centimetre of height and the corresponding volume.[6]

[6] Touro only had for every meter. Values between were estimation through a linear transformation between the two points

- Different normalization values were used and some data had to be denormalized and normalized with the correct range.
- Outflow data on the model refers always to the outflow without the ecological flow. This had to be added when using the outflow data as inflow.

The results for 330 days are presented in figures 4.17, 4.18, 4.19 and 4.20. Overall, as can be seen the results are not as good as the standalone model, but still acceptable. The full system seems to have more difficulty to reproduce the peaks but it does register most peak flows. Naturally, results from Model 1 (figure 4.17) were very comparable to the standalone model, since the difference was only the previous inflow information.

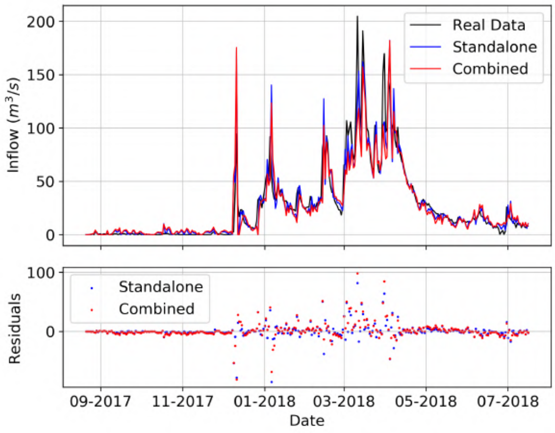


Figure 4.17: Model 1 full system results

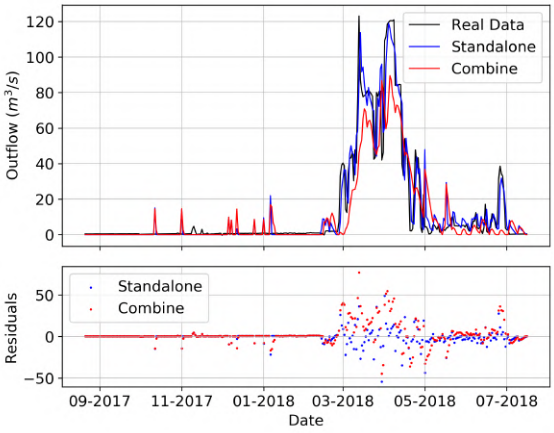


Figure 4.18: Model 2 full system results

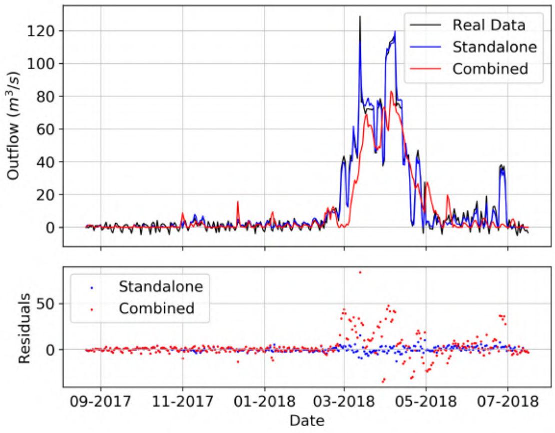


Figure 4.19: Model 3 full system results

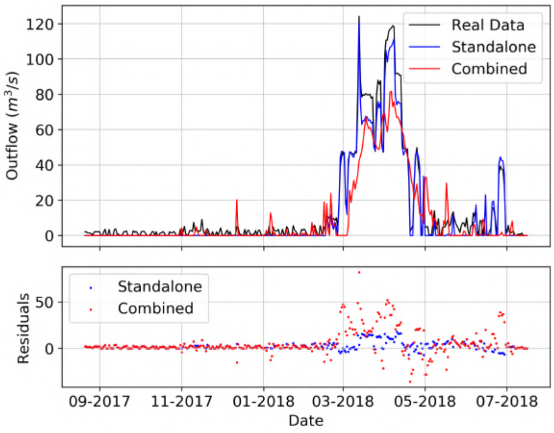


Figure 4.20: Model 4 full system results

Results from the reservoir models were similarly reduced from the standalone counterparts. Where results varied widely was in terms of the secondary models where smaller reservoir could not even remotely keep up with the results of model 2, figure 4.21.

Resutls form secondary models 3 (figure 4.22) and 4 (figure 4.23) are quite poor, this could be due to the much faster and more sensitive dynamics of these models. The resutls, however, did not impacted

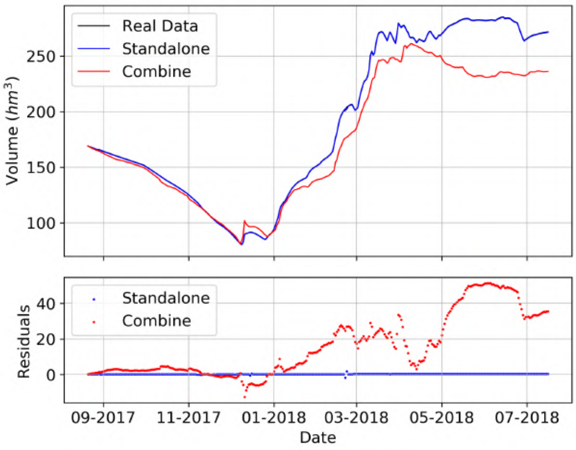
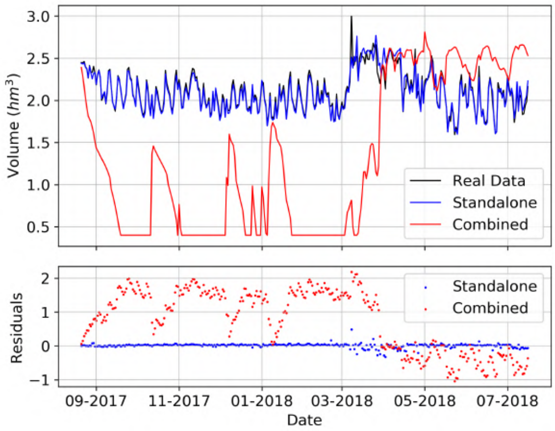


Figure 4.21: Model 2 secondary full system results

the main models 3 and 4, which means the network places very small weigths on these inputs. This reinforces the idea that these reservoir are not used for the storage capacity, but simply to extract additional energy from the flow of the river.



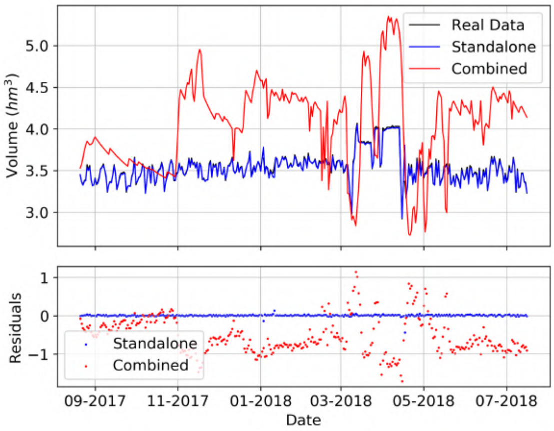


Figure 4.22: Model 3 secondary full system results

Figure 4.23: Model 4 secondary full system results

Secondary model 2 (figure 4.21) displays small errors that accumulate over time, to counter this phenomena and improve results over all the system, an update period was introduced. This meant that from time to time the real values of the storage were given as inputs to the secondary models. To asses this idea, tests were made with 7 days and 30 days intervals. Figure 4.24 and 4.25 show the differences in performance of the secondary model 2.

The updates of the previous storage inputs improves substantially the accumulation of errors. These better results improved the main model 2 (figure 4.26) which directly impacted model 3 and 4, figures 4.27 and 4.28. Peak flow prediction was much better across the board and the models were able to predict the scale of the flow much better. It also improved results for secondary models although the results were still quite poor, figures 4.29 and 4.30.

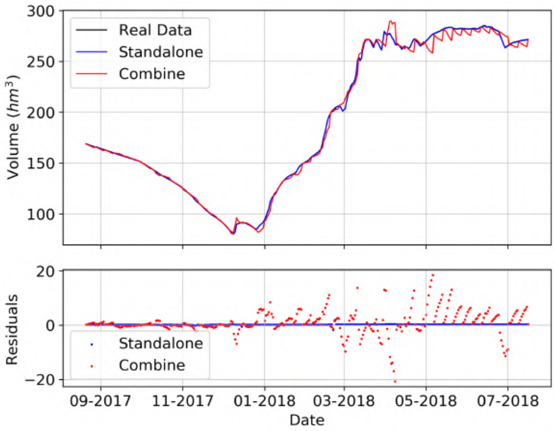


Figure 4.24: Secondary Model 2 - update time 7 days

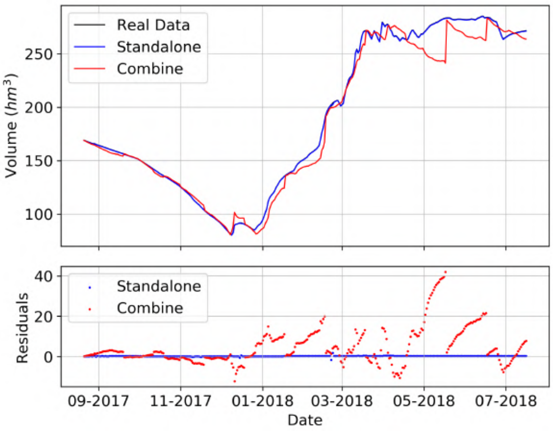


Figure 4.25: Secondary Model 2 - update time 30 days

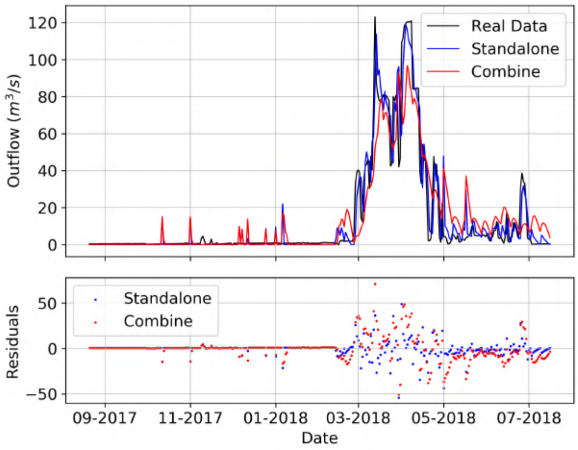


Figure 4.26: Model 2 results - update time 7 days

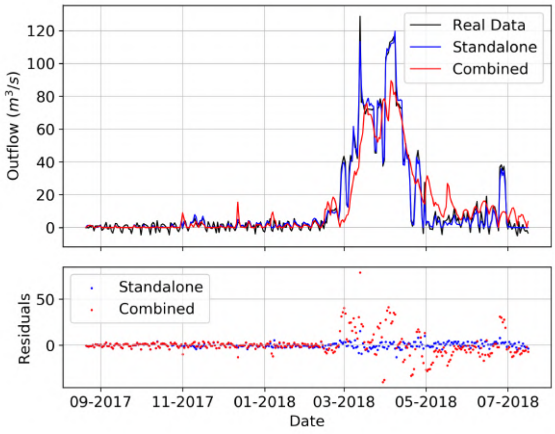


Figure 4.27: Model 3 results - update time 7 days

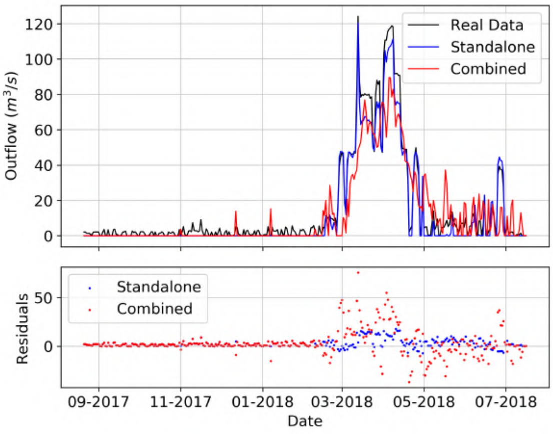


Figure 4.28: Model 4 results - update time 7 days

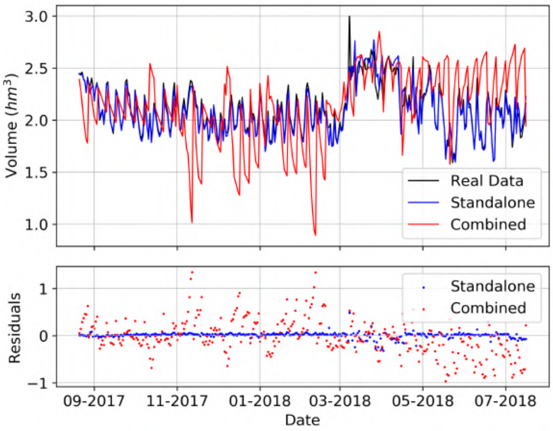


Figure 4.29: Secondary Model 3 - update time 7 days

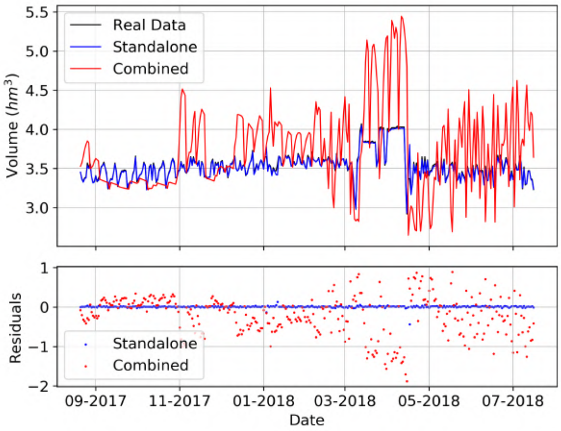


Figure 4.30: Secondary Model 4 - update time 7 days

The full results are presented in table 4.7[7]. The remaining plotted results are shown in appendix B.

Table 4.7: Full system results

	Storage Update Time											
	None				30 days				7 days			
	RMSE	MAE	r	NSE	RMSE	MAE	r	NSE	RMSE	MAE	r	NSE
Model 1	14.405	7.129	0.919	0.842	14.405	7.129	0.919	0.842	14.405	7.129	0.919	0.842
Model 2	13.623	7.083	0.886	0.756	12.824	7.024	0.890	0.784	13.005	7.390	0.885	0.777
Model 3	12.994	6.946	0.886	0.752	12.012	6.577	0.895	0.788	12.042	6.847	0.895	0.787
Model 4	14.547	7.953	0.894	0.727	13.432	7.511	0.898	0.767	13.643	7.716	0.889	0.759
Model 2 Secondary	23.479	16.620	0.987	0.893	10.392	6.607	0.992	0.979	4.532	2.800	0.998	0.996
Model 3 Secondary	1.187	1.020	0.078	-26.129	0.729	0.569	0.099	-9.216	0.382	0.289	0.209	-1.812
Model 4 Secondary	0.714	0.624	0.387	-15.404	0.645	0.517	0.439	-12.384	0.539	0.401	0.470	-8.339

The results shows competent prediction even with no storage update. Moriasi et al. [40] defines a very good fit value of the NSE index as bigger than 0.75[8]. All flow models displays values of NSE bigger than 0.75, except model 4 with no update which has a NSE equal to 0.727, still in the good range. Storage for model 2 also has very good results, but models 3 and 4 have negative values of NSE meaning that an average values is better that the results of the models. Other simpler models, less concerned with the small variations could be useful for the secondary models of small reservoirs.

Overall the models present very good results, the objective was successfully achieved. The techniques shown in this study can successfully predict discharge of multiple dams, however it was clear that the size of the reservoir does play a role on the success rate and ease of modelling.

[7] Values of error for denormalize data

[8] for monthly time step, watershed simulation. Useful despite not the same time step

Chapter 5

Conclusions

In this work, data driven modelling was applied to a system of dam reservoirs in a watershed. Computational models are regularly applied in this filed of study, although generally these are conceptual model. Dams create a human barrier that changes the behaviour of the flow, for those effects data driven model became important to help the study of a watershed.

The goal was to implement models capable of learning with prior data and forecast outflow at the reservoirs. The problem consisted of three gravity dams of different sizes.

Not existing an abundance of work developed for this specific objective the approach taken was to try with a linear models and with a non linear models to understand if a linear model would suffice or if the non linear would be imperative. Similarly to some of the prior work developed, an MLP was the choice for the non linear models. Because the formulation consisted of several delayed inputs for all the models a regressive model approach seemed logical for the linear model, an ARMAX was then chosen.

For each dam, two models were needed: one for to predict discharge and another for the remaining storage. Also, a model to predict the first inflow of water was required. Because the dams were close to each other with no relevant tributary in between, the outflow from one was the inflow of the other. This meant that the full system could only use initial condition, weather variable and the flow of a hydrometric station.

The features of the models were selected with tests on the data to understand what were the best subset of inputs (similar to a wrapper method). An the structure parameters of the ANN models were determined by a hyperparameter tuning with bayesian optimizer. The ARMAX models on the other hand, were selected based on the AIC.

The best results were obtained for the MLP implementation on all objectives, in the full system the results were similarly successful across all dams.

5.1 Final Remarks

The non linear nature of the problem proved that the ANN was across the board, a better alternative to models this kind of problem. Results using these ML techniques were very successful at modelling the outflow of all dams.

To model the inflow of the first dam the structure had to be more complex to accommodate for the higher number of inputs and its relationship since it takes data from 4 meteorological station. Interestingly, the removal of the temperature information produce better results. The precipitation was naturally the weather variable with the most influence on the inflow.

Portodemouros dam is the main dam in this system with the bigger capacity and the results were quite influenced by the storage levels. This was clear when the results with periodic update of storage where compared with the system without update. The update time allowed the results of the storage to not accumulate much errors in this reservoir.

For the smaller dams the poor results of the storage models when combined in the full system simulation, did not influence much the results of the reservoir outflow. This showed that these smaller dams are less concerned with storage serve more to extract extra energy from the river.

Over all the approach taken in this study proved very successful at forecasting outflow of reservoir. However it is important that each dam and watershed is its own unique system which may require other explanatory variables. Given that the behaviour can be widely different in the same area(because of the size) it is clear that other cases may have different success levels, if a similar implementation are taken. Machine learning techniques can be used to solve problems like this one, that other computational models struggle with, enhancing the view over the problem. These approaches if working in tandem with physical models, can potentially produce rich 'grey' models.

5.2 Future Work

A number of different ML techniques could be implemented in search of better results. However those obtained suggest that more complex models may not be of great use. Instead the study of slightly different conditions could be interesting.

In this work the electrical demand was only considered as far as the seasonality effects of the week day and holiday. These are in fact the main influence in the electrical demand which naturally influences the opening of the dam gates. Electric prices are, however, adjusted on a much smaller time step than a day, a study which works with an hour step could use direct electric prices to improve results, as well as, a finer view of a dam behaviour.

For full system integration on the storage models of small reservoir other more simplistic approach, much less sensitive to small changes in its inputs, could produce more consistent results and should be attempted.

Integrating these models with a physical model, like MOHID land, would be interesting to understand the full benefits of having good reservoir models on a broader watershed simulation.

Bibliography

[1] Normas de explotación. Presa de Portodemours, Gas Natural Fenosa Generación. Noviembre 2015.

[2] B. Psiloglou, C. Giannakopoulos, S. Majithia, and M. Petrakis. Factors affecting electricity demand in athens, greece and london, uk: A comparative assessment. *Energy*, 34(11):1855–1863, 2009.

[3] A. Pardo, V. Meneu, and E. Valor. Temperature and seasonality influences on spanish electricity load. *Energy Economics*, 24(1):55–70, 2002.

[4] F. J. Nogales, J. Contreras, A. J. Conejo, and R. Espínola. Forecasting next-day electricity prices by time series models. *IEEE Transactions on power systems*, 17(2):342–348, 2002.

[5] Normas de explotación de la presa de Brandariz., Enel Union Fenosa Renovables. Septiembre 2007.

[6] Normas de explotación de la presa de salto de Touro, Patrimonio Hidroeletrico De Galicia. Eneiro 2008.

[7] X. Galicia. Meteogalicia: Rede de aforos, 2020. URL `http://www2.meteogalicia.gal/servizos/AugasdeGalicia/estacions.asp`.

[8] X. Galicia. Meteogalicia: Rede meteorolóxica, 2020. URL `https://www.meteogalicia.gal/observacion/estacions/estacions.action`.

[9] S. Yang, D. Yang, J. Chen, and B. Zhao. Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model. *Journal of Hydrology*, 579: 124229, 2019.

[10] D. Yang, S. Herath, and K. Musiake. A hillslope-based hydrological model using catchment area and width functions. *Hydrological Sciences Journal*, 47(1):49–65, 2002.

[11] A. R. Oliveira, T. B. Ramos, L. Simionesei, L. Pinto, and R. Neves. Sensitivity analysis of the mohid-land hydrological model: A case study of the ulla river basin. *Water*, 12(11):3258, 2020.

[12] S. Jain, A. Das, and D. Srivastava. Application of ann for reservoir inflow prediction and operation. *Journal of water resources planning and management*, 125(5):263–271, 1999.

[13] K. Budu. Comparison of wavelet-based ann and regression models for reservoir inflow forecasting. *Journal of Hydrologic Engineering*, 19(7):1385–1400, 2014.

[14] K. Mohammadi, H. Eslami, and D. S. DAYANI. Comparison of regression, arima and ann models for reservoir inflow forecasting using snowmelt equivalent (a case study of karaj). *JOURNAL OF AGRICULTURAL SCIENCE AND TECHNOLOGY (JAST)*, 2005.

[15] S. K. Ahmad and F. Hossain. A generic data-driven technique for forecasting of reservoir inflow: Application for hydropower maximization. *Environmental Modelling & Software*, 119:147–165, 2019.

[16] P. Coulibaly, F. Anctil, and B. Bobée. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology*, 230(3-4):244–257, 2000.

[17] S. A. Johnson, J. R. Stedinger, and K. Staschus. Heuristic operating policies for reservoir system simulation. *Water Resources Research*, 27(5):673–685, 1991.

[18] C. Cheng, K. Chau, Y. Sun, and J. Lin. Long-term prediction of discharges in manwan reservoir using artificial neural network models. In *International Symposium on Neural Networks*, pages 1040–1045. Springer, 2005.

[19] P. Chaves and F.-J. Chang. Intelligent reservoir operation system based on evolving artificial neural networks. *Advances in Water Resources*, 31(6):926–936, 2008.

[20] F. J. Montáns, F. Chinesta, R. Gómez-Bombarelli, and J. N. Kutz. Data-driven modeling and learning in science and engineering. *Comptes Rendus Mécanique*, 347(11):845–855, 2019.

[21] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[22] L. Lennart. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, pages 1–14, 1999.

[23] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

[24] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[25] N. Rochester, J. Holland, L. Haibt, and W. Duda. Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on information Theory*, 2(3):80–93, 1956.

[26] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[27] B. Widrow and M. E. Hoff. Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs, 1960.

[28] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice-Hall, Inc., 2007.

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[30] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[31] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[32] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[33] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?– arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[34] S. K. Jain and K. Sudheer. Fitting of hydrologic models: a close look at the nash–sutcliffe index. *Journal of hydrologic engineering*, 13(10):981–986, 2008.

[35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

[36] L. Ljung. System identification toolbox: for use with matlab. 1988.

[37] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[38] H. Hsu and P. A. Lachenbruch. Paired t test. *Encyclopedia of Biostatistics*, 6, 2005.

[39] F. J. Tapiador, F. J. Turk, W. Petersen, A. Y. Hou, E. García-Ortega, L. A. Machado, C. F. Angelis, P. Salio, C. Kidd, G. J. Huffman, et al. Global precipitation measurement: Methods, datasets and applications. *Atmospheric Research*, 104:70–97, 2012.

[40] D. N. Moriasi, J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900, 2007.

Appendix A

Feature subset test results

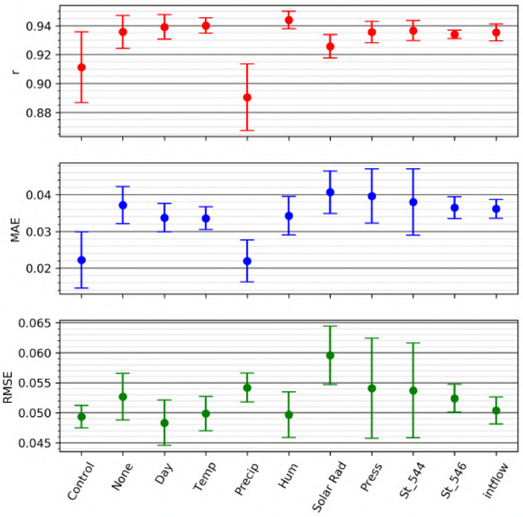


Figure A.1: Removing one type of variable at a time, the rest averaged, Model 1

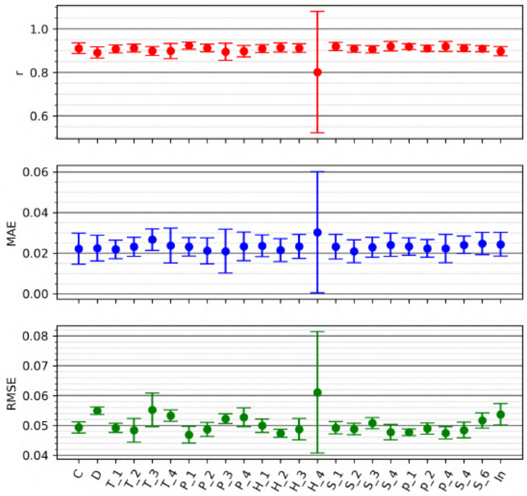


Figure A.2: Removing one variable at a time, Model 1

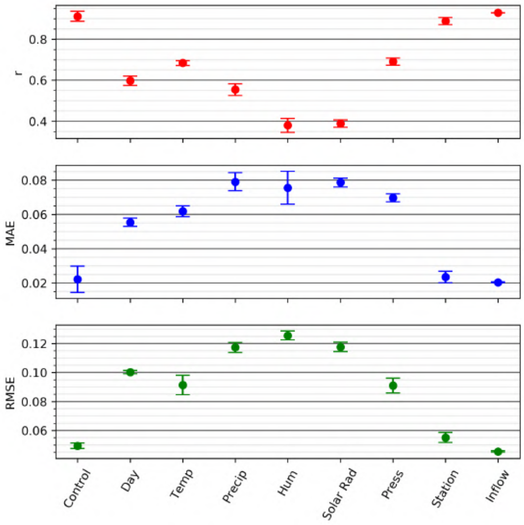


Figure A.3: One type of variable at a time, Model 1

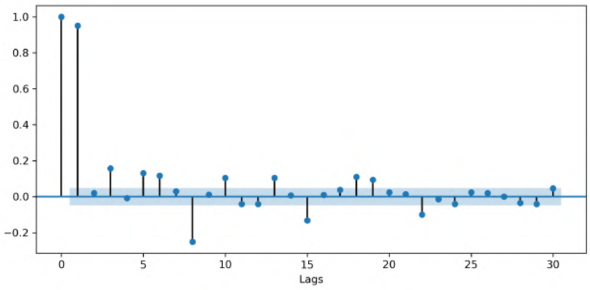


Figure A.4: PACF outflow Portodemouros

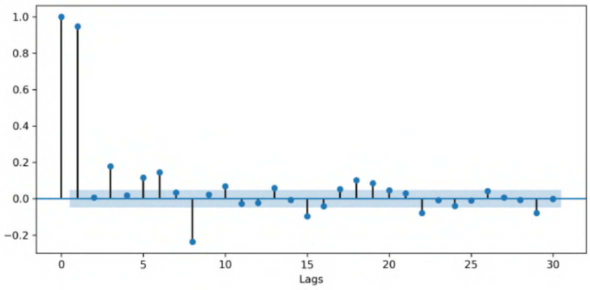


Figure A.5: PACF outflow Brandariz

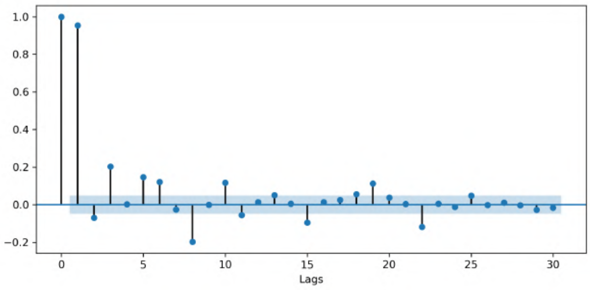


Figure A.6: PACF outflow Touro

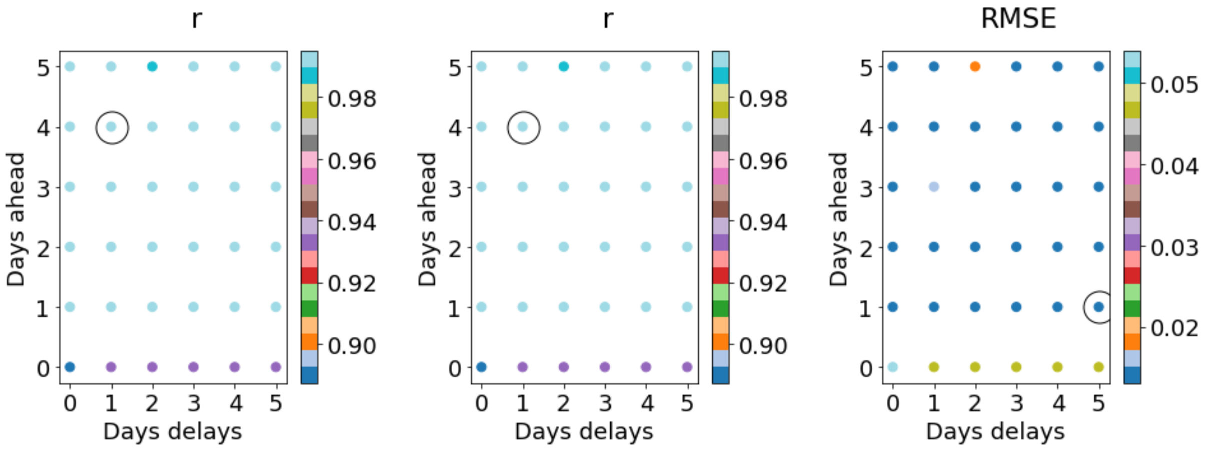


Figure A.7: Test of different days for the inflow Brandariz

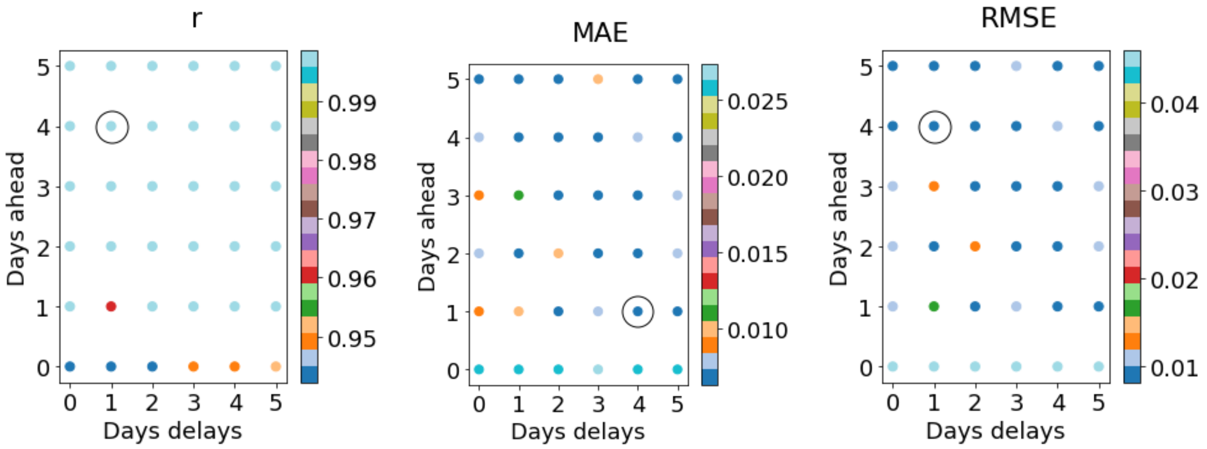


Figure A.8: Test of different days for the inflow Touro

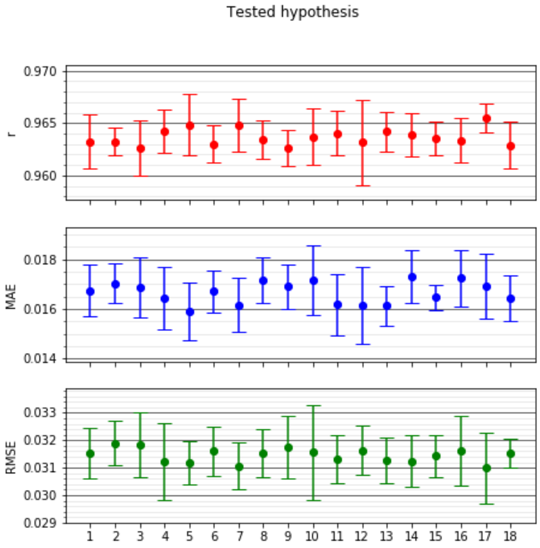


Figure A.9: Tested hypothesis for Model 2

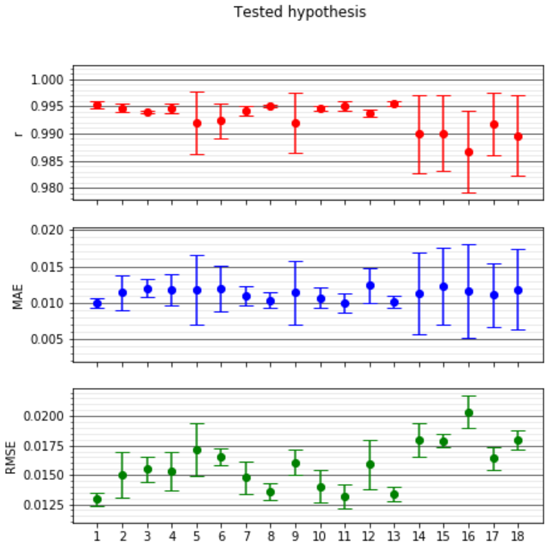


Figure A.10: Tested hypothesis for Model 3

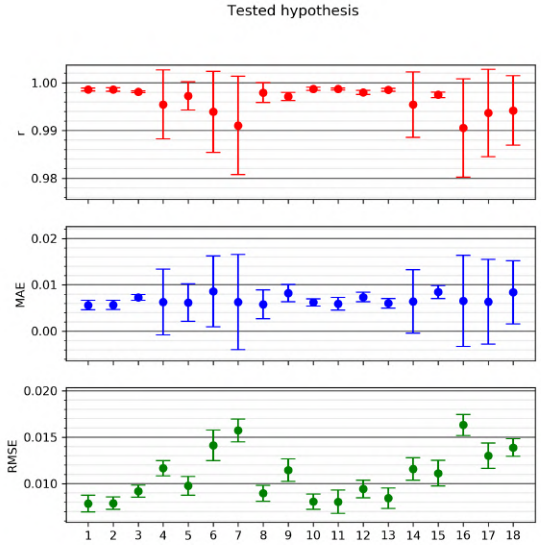


Figure A.11: Tested hypothesis for Model 4

Appendix B

Complementary Results

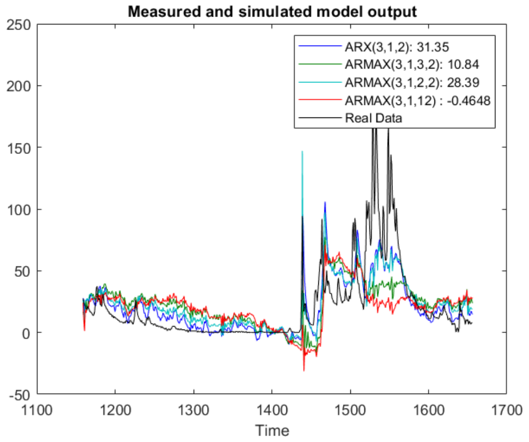


Figure B.1: ARMAX resutls for the originally scaled Model 1

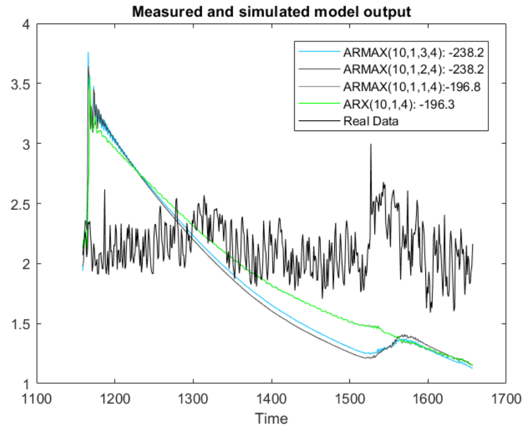


Figure B.2: ARMAX resutls for the originally scaled secondary Model 3

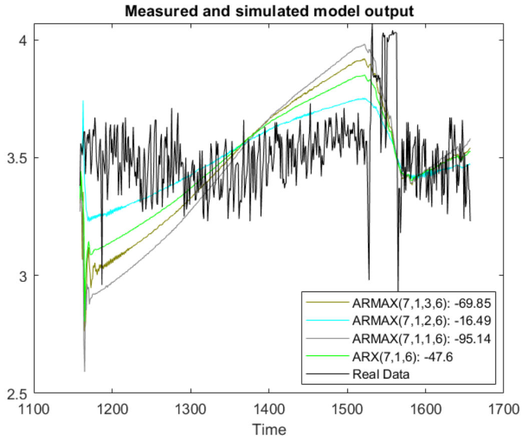


Figure B.3: ARMAX resutls for the originally scaled secondary Model 4

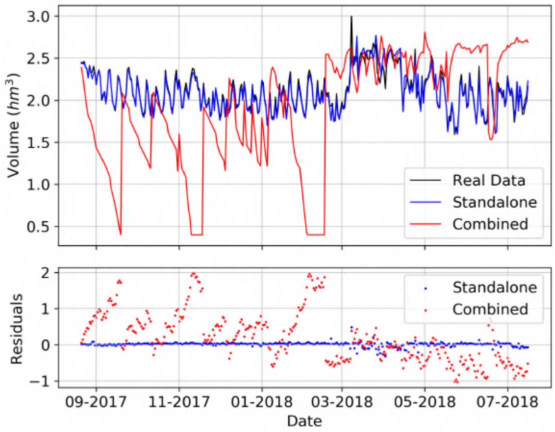


Figure B.4: Model 3 secondary full system results, update time 30 days

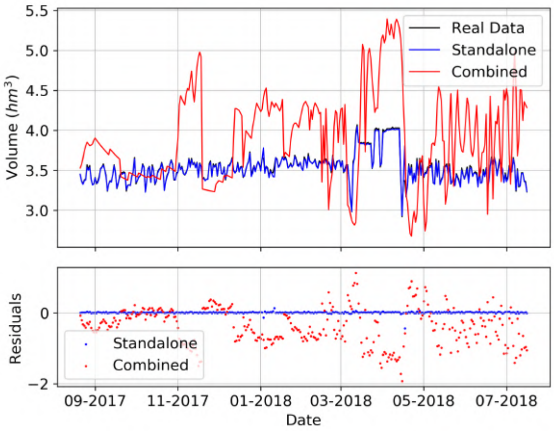


Figure B.5: Model 3 secondary full system results, update time 30 days

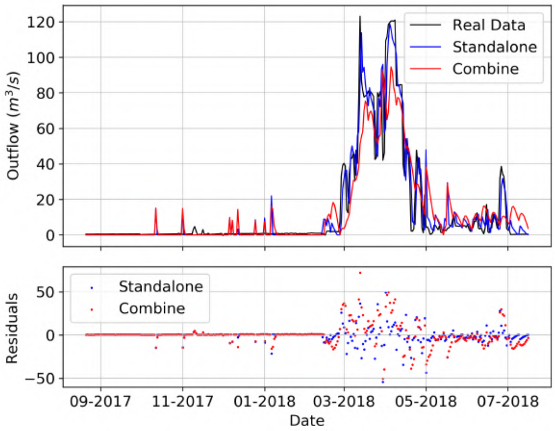


Figure B.6: Model 2 full system results, update time 30 days

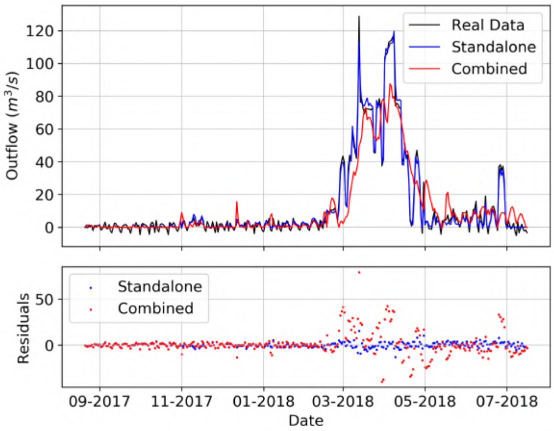
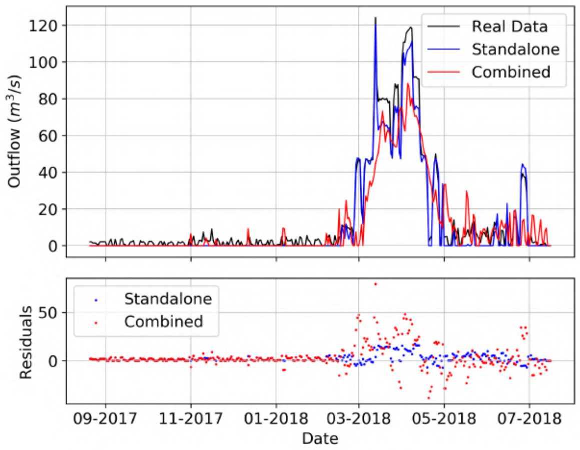


Figure B.7: Model 3 full system results, update time 30 days



(h)

Figure B.8: Model 4 full system results, update time 30 days