Universidade de Aveiro Departamento de Eletrónica,
Telecomunicações e Informática
2019

Pedro Guilherme
Silva Matos

Previsão do comportamento de redes de distribuição de água com métodos de aprendizagem automática

Predicting the behaviour of water distribution networks with machine learning methods

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2019

Pedro Guilherme Silva Matos

Previsão do comportamento de redes de distribuição de água com métodos de aprendizagem automática

Predicting the behaviour of water distribution networks with machine learning methods

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor Sérgio Guilherme Aleixo Matos, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor António Gil D'Orey Andrade Campos, Professor auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Dedico este trabalho à minha família pelo incansável apoio ao longo de todos estes anos.

o júri / the jury

presidente / president	Prof. Doutora Ana Maria Perfeito Tomé Professora Associada da Universidade de Aveiro
vogais / examiners committee	Doutora Bernardete da Costa Coelho Investigadora da Universidade de Aveiro - Departamento de Engenharia Mecânica
	Professor Doutor Sérgio Guilherme Aleixo de Matos Professor Auxiliar em Regime Laboral da Universidade de Aveiro

agradecimentos / **acknowledgements**

Um sincero agradecimento ao meu orientador Sérgio Matos, ao coorientador Gil Campos e ao meu amigo André Antunes, sem o vosso apoio, dedicação e paciência, a realização deste trabalho não teria sido possível.

Palavras Chave

Sistemas de distribuição de água, aprendizagem automática, previsão de consumos de água, modelação, simulação.

Resumo

Os sistemas de abastecimento de água são infraestruturas indispensáveis em qualquer civilização moderna. Qualquer casa moderna tem sempre água corrente. As pessoas estão de tal maneira dependentes deste bem essencial que hoje em dia é impensável qualquer meio social viver sem abastecimento de água. Os sistemas de abastecimento são responsáveis por manter o constante fornecimento de água a casas, hospitais, indústrias, etc., e, consequentemente, também são responsáveis por manter o funcionamento da sociedade. Como são sistemas indispensáveis no quotidiano não se tem tanto em consideração os custos associados com o seu funcionamento. Estes sistemas têm de bombear água para satisfazer os consumos dos seus clientes e enfrentam grandes problemas de custos energéticos relacionados com as operações de bombeamento.

Este trabalho apresenta e analisa uma possível solução para este problema utilizando um sistema de apoio à decisão que tira partido da variação da tarifa energética ao longo do dia para fazer a otimização dos custos energéticos. A solução apresentada utiliza métodos de aprendizagem automática para prever consumos de água e simular o consequente comportamento das redes possibilitando assim o agendamento das operações de bombeamento para que coincidam com o período em que a tarifa é mais barata. Os resultados indicam que Redes Neuronais, Máquinas de Aprendizagem Extrema e Redes Neuronais Recurrentes com *Gated Recurrent Units* são capazes de alcançar um bom desempenho na previsão de consumos de água. Também foi possível criar um modelo que reproduz com precisão o comportamento de uma rede de abastecimento de água de médio tamanho usando Redes Neuronais.

Keywords

Water distribution systems, machine learning, water demand forecasting, modelling, simulation.

Abstract

Water supply systems are indispensable infrastructures in any modern civilisation. Any modern house has running water at all time. People are so dependent on this essential good that today it is unthinkable for any social environment to live without water supply. Supply systems are responsible for maintaining the constant supply of water to homes, hospitals, industries, etc., and, consequently, are also responsible for maintaining the functioning of society. Since these systems are so indispensable in daily life, the costs associated with their operation are not taken into account. These systems have to pump water to meet their customers' demands and face major energy cost-efficiency issues related to pumping operations.
This work presents and analyses a possible solution to this problem using a decision support system that takes advantage of variations in the energy tariff throughout the day to optimise energy costs. It uses machine learning methods to predict future water demands and simulate the consequent behaviour of the networks, thus allowing the scheduling of pumping operations to coincide with the period when the tariff is cheaper. The results indicate that Artificial Neural Networks, Extreme Learning Machines and Recurrent Neural Networks with Gated Recurrent Units are capable of achieving a good performance forecasting water demands. It was also possible to create a model that accurately reproduces the behaviour of a water supply network of reasonable size using Artificial Neural Networks.

Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
BiGRU	Bidirectional Gated Recurrent Unit
BiGRUN	Bidirectional Gated Recurrent Unit Network
BFGS	Broyden-Fletcher-Goldfarb-Shanno
DANN	Dynamic Artificial Neural Network
DBN	Deep Belief Networks
dd	Daily Step Demand Vector
DRAGAN-ANN	Dynamic, Real-time, Adaptive Genetic Algorithm – Artificial Neural Network
ELM	Extreme Learning Machine
GA	Genetic Algorithm
GRU	Gated Recurrent Unit
GRUN	Gated Recurrent Unit Network
KNN	k-nearest neighbours
LBFGS	Limited Memory Broyden-Fletcher-Goldfarb-Shanno
MAE	Mean Absolute Error
pd	Prior Demand Vector
R^2	Coefficient of Determination
RBM	Restricted Boltzmann Machines
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
SLFN	Single Layer Feed-Forward Neural Network
SVR	Support Vector Regression
WCP	Water Consumption Point
WDN	Water Distribution Network
WSS	Water Supply System

CHAPTER 1

Introduction

This chapter presents the overall theme of this work. It starts with a small contextualization, followed by an explanation on why this topic is relevant nowadays, and then the two main objectives of dissertation. At the end of the chapter there are some brief guidelines on how to read this document.

1.1 Water Supply Systems

Water Supply System (WSS) are infrastructures that collect, transport, treat, store and distribute water for homes, industries, irrigation, firefighting, etc., [1]. They provide an essential service to modern society since people are dependent on water for the great majority of everyday tasks. From simple tasks such as drinking to commercial and industrial activities water is needed in both quantity and quality, and it is up to water supply systems to fulfil these requirements. To accomplish this, water supply systems are composed of pumping stations that pump water from a variety of sources, aquifers, springs, glaciers, or others. This water is then moved through a network of pipes that is usually extended over many square kilometres, traversing all kinds of terrain before reaching its destination.

Tanks (reservoirs) are essential for meeting water demands, storing water that allows WSSs to function in an efficient manner [2]. The stored water can be used for either operating storage or emergency storage. Commonly, reservoirs are located near the clients of the WSS, such as villages or industrial areas. This allows clients to have immediate access to water as soon as needed. While eliminating the need of constantly pumping water to fulfil the demands and maintain pipe pressure. A reservoir in a good location can even supply clients by action of gravity alone without requiring additional pumping operations. Despite all these benefits, reservoirs have some limitations related to their capacity, and therefore amount of stored water must be managed.

Pumping water from one place to another consumes a lot of energy. In fact, between 90% and 95% of the electricity that its purchased by the water supply system is used for pumping [3]. The water sector was, in 2014, responsible for 4% of the total global electricity consumption [4]. WSSs must transport water through terrain that is, usually, irregular, which may require several pumping stations between the water source and its destination. All these pumping operations spend a great amount of energy that translates to a big electricity bill. This cost is also associated with the way the pumping operations are managed. One popular method to control the water level is activating the pumping operations to refill the tanks when their level is near a predetermined minimum. The opposite is done when they reach a predetermined maximum. This process only takes into account the immediate water demand of the network. The cost of the energy that it spent in order to perform the pumping operations is not considered. This process keeps the clients satisfied but is energetically inefficient.

It is possible to improve the efficiency of this process and reduce the cost associated with electrical consumption for pumping by 30% or more [5], [6], [7]. One possible solution to the previously mentioned problem is to take advantage of the daily variation in the electric tariff offered by the electric company responsible for supplying the WSS [8]. Elaborately, depending on the electric tariff, there are time intervals when the electricity cost is lower. If a WSS is capable of scheduling the pumping operations in such a way that they coincide with the cheaper tariff, it is possible to reduce the energy costs while guaranteeing that the tanks contain sufficient water to suit the consumers' needs.

Finding the optimal pumping schedule is not an easy task. Foremost, in order to plan the pumping operations, it is necessary to know beforehand how much water will be consumed. Then, by combining the forecasted water demands with the current state of the water network, a simulation of the behaviour of the network must be performed, which will output the energy that will be spent to meet the customers' demands along with the consequent state of the network. With this information, it is then possible to calculate the cost of the future pumping operations and optimise the energy costs.

1.2 A conceptual decision support system

A decision support system that can forecast the water demands, simulate the behaviour of the network, calculate the system energy use for a given pumping schedule, and eventually find the optimal pumping schedule is a useful tool to help water supply companies achieve the goal of reducing the energy consumption costs. Such system would also allow the companies to examine the future state of the network which in itself is a very useful feature. Taking this into consideration, it is easy to conceptualise a decision support system that helps the company responsible for managing the water distribution network. Figure 1.1 illustrates a possible architecture for such a system, composed of three modules:

- **Water demand prediction module** - The pumping schedule will depend on the water demand, *i. e.*, the amount of water that is going to be needed at each moment of the day will directly affect the number of pumping operations and the time of those operations. This module must predict, for a given observation window with a certain time-step, the water demand during the day.

- **Network simulation module** - Based on the current state of the network and on the pumping operations that are expected to be executed to satisfy the predicted water demands, this module predicts and simulates the network behaviour and calculates the energy consumption to execute those operations.
- **Optimisation module** - The main goal of the module is to minimise the cost of energy consumption by matching the schedule of the pumping operations with the best electric tariff structure while satisfying the water demands. This module is not the scope of this project.

In order to work, the system must be supplied with water demands that were observed in the past, the water demand prediction module will then create a prediction for a certain time window in the future (for example the next 24 h). With the predicted water demands it is then possible to know how much water must be supplied and/or pumped to keep the clients satisfied and, consequently, the amount of water that is going to be drained from the network. A second set of information must also be supplied, namely the current state of the network, tank levels, pump operations that must be performed in order to satisfy the water demands, etc. This information, combined with the predicted water demands, is passed to the water simulation module. The module outputs the consequent state of the network and how much energy would be spent in order to achieve it. Taking this into consideration the optimisation model tries to find a pumping schedule that minimises the energy costs. With this new pumping schedule, a new simulation is formed, this process is repeated until the optimal schedule is found. When that happens, the optimal schedule and the consequent network simulated state can be sent to the management company, allowing them to take an informed decision on how to perform the pumping operations.

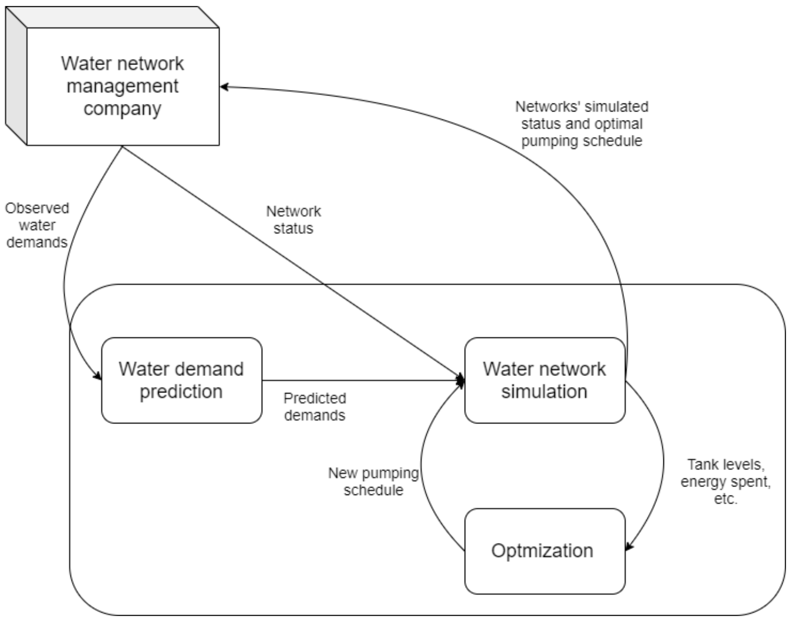


Figure 1.1: Illustration of the conceptual decision support system that finds a pumping schedule that minimises the pumping energy costs.

1.3 Objectives

A decision support system like the one mentioned in Section 1.2 requires a very precise water demand prediction module. All the processes of simulating the behaviour of the network and optimising the pumping schedule are based on the capability of predicting water demands. The current literature related to this topic demonstrates several solutions capable of achieving very good results. Since water demand forecasting is such an important part of the system, it is essential that the best possible results are achieved. Therefore, the first objective of this work is to develop a state-of-the-art solution for this problem.

Concerning the water simulation module, the most common solution is to an hydraulic simulation software, such as EPANET [9]. As mentioned in Chapter 2, the EPANET software has some problems concerning the computational effort required to both run and calibrate the software. Additionally, the calibration process requires that an operator, or an additional software, adjusts several characteristics of the water network elements, *e.g*, the roughness of the pipes, water leaks, etc. until the model is able to correspond to the real-life scenario. This makes its use somewhat impractical. A solution that does not require a calibration process and that is more computationally efficient is required. The literature points out that it is possible to use machine learning to meta-model a Water Distribution Network (WDN), eliminating the need to use a simulation model like EPANET. Thus, the second and main objective of this work is to implement and analyse a machine learning technique to model a water distribution network.

1.4 Contributions

The work developed during this thesis project resulted in the following publication:
Pedro Matos, Sérgio Matos, António Andrade-Campos, "Predicting the behaviour of water distribution networks with machine learning models", INForum 2019, Setembro 2019, Guimarães, Portugal.

1.5 Structure of the document

This work is composed by a total of 5 chapters.

Chapter 1 presents and introduces the overall theme that is discussed in this document by explaining a problem that WSS face, why it's a relevant topic nowadays, a possible solution, and the objectives of this work.

On chapter 2 a bibliographic analysis regarding the state of water demand forecasting and WDN simulation is presented. It's divided in two sections, one for each topic, and gives an insight on the challenges that are faced when tackling problems of this nature. It also presents several state-of-the-art solutions, some of which were used as a reference for the methods that have been developed in this dissertation.

Chapter 3 is also divided in two sections each with their respective subsections. The first section is related with water demand forecasting and presents various methods that were developed for that purpose, the data that was used, and how that data was processed. As for the second section, the method that was adopted to simulate WDNs is presented along with its inputs and outputs. Additionally, the procedure adopted to generate data to train, test and validate the solution is explained.

Chapter 4 is composed of a section where the architecture of the various techniques used in water demand forecasting and their respective results are shown. Then, another section shows two case studies used to perform a series of tests on the method used to simulate water networks. Such tests include a verification of the performance of the method on a 24 h scenario and analysis on the method's tolerance to noise and on the amount of training data required to achieve a good performance.

Finally, chapter 5 discusses the previous results and presents some conclusions from what has been demonstrated on previous chapters.

Attached to this document there is an appendix that contains some extra results generated with the water network simulation method.

CHAPTER 2

State-of-the-art

This chapter presents a bibliographical analysis of the state-of-the-art regarding water demand forecasting and water networks modelling. As such, it is divided in two main sections. The first one is related with water demand forecasting and starts by presenting some works that use mathematical methods used to forecast water demands. Then it shifts its focus to more recent works that use machine learning. The second section presents works related with simulation of WDNs, starting with some works that model networks based on a system of equations and in some more recent works that use deep learning.

2.1 Forecasting water demand

Forecasting systems traditionally use mathematical models based on statistical regression, harmonic analysis, and time series analysis[10], [11], [12]. This type of models are still in use today [10] and have been proven to be effective to some degree. However, the popularity of machine learning techniques has increased exponentially in the past decade and with it also the number of forecasting systems using machine learning to forecast short- and long-term water demands. This shift in methodology seems to be paying off since several studies have shown that machine learning outperforms the traditional forecasting methods [13], [14], [6]. Within these techniques, Artificial Neural Networks (ANNs) are the ones that stand out for being the most used ones [14], [6]. However, these are not the only techniques being used [6].

Given their diversity, forecasting models can be classified according to different criteria. One of them is linear and nonlinear models. Due to their simplicity and practical use, linear models are widely used. These models include univariate time series analysis and Autoregressive Integrated Moving Average (ARIMA) models [6],[15]. These are solely based on chronological sequences of observations from the past. ARIMA models reflect static and dynamic properties of stationary series. Certain non-stationary series are interpreted as white noise passing through a discrete, finite-dimensional linear filter. The accuracy of their results is oftentimes unsatisfactory. Therefore, these can become more accurate by transforming the raw data with a Fourier transform. Nonlinear methods for water forecast include: nonlinear regression models, bilinear models, threshold autoregressive models, ANN models, fuzzy logic, extended Kalman filters and genetic programming [16], [17], [18].

Kozłowski et al. [10] presents a method in which a variant of time series analysis to forecast hourly water consumptions 24 h ahead is used. This method, in [10], is refereed to as trend analysis but it is also known as seasonality. It consists on identifying the periodic variation (all the phases of a cycle) occurring in a time series while considering their respective daily temperature. The hourly water consumptions are analysed based on the assumption that a day/cycle contains 24 h. Therefore, the parametric identification is done by dividing the time series with N elements, $\{X_i\}_{1\leq i\leq N}$, into 24 sub-series:

$$\begin{aligned}
&\{X_1^k\}_{1\leq i\leq T} = \{x_1, x_{25}, ..., x_{24(T-1)+1}\},\\
&\{X_2^k\}_{1\leq i\leq T} = \{x_2, x_{26}, ..., x_{24(T-1)+2}\},\\
&...\\
&\{X_{24}^k\}_{1\leq i\leq T} = \{x_{24}, x_{48}, ..., x_{24T}\},
\end{aligned} \tag{2.1}$$

Where X_j^k denotes water consumption from hour $j-1$ until the j^{th} hour of the k^{th} day for $1 \leq j \leq 24$ and $1 \leq k \leq T$, assuming that $(24T = N)$. A monthly period is considered, and given the above, it is assumed that the elements of the series $\{X_j^k\}_{1\leq k\leq 31}$ for the phase $1 \leq j \leq 24$ satisfy the equation:

$$X_j^k = \alpha_0^j + \alpha_1^j t_k + \epsilon_j^k, \tag{2.2}$$

Where α_0^j and α_1^j denote the free term and the scope in expression in the j^{th} phase, t_k is the mean ambient temperature for the k^{th} day, and $\{\epsilon_j^k\}_{1\leq k\leq 24}$ is a series of independent and identically distributed random variables normal distribution $N(0, \sigma_j^2)$ for $1 \leq j \leq 24$.

The values of α_0^j and α_1^j were determined with the least squares method. The values of the series $\{\bar{X}_j^k\}_{1\leq k\leq 24}$ are the theoretical values (forecasts) of water consumption for the k^{th} day:

$$\bar{X}_j^k = \hat{\alpha}_0^j + \hat{\alpha}_1^j t_k, \tag{2.3}$$

$\hat{\alpha}_0^j$ and $\hat{\alpha}_1^j$, with $1 \leq j \leq 24$, denote the estimated structural parameters in equation 2.2. A linear dependence between water consumption and time factor is investigated in the model 2.2, when it is assumed that $t_k = k$. As a result, the hourly water consumption is forecasted using 2.3 and linear trends.

Harmonic analysis assumes that a time series consists of sine and cosine waves with different frequencies also called harmonics, and elements of the series are linear combinations of these harmonics. The Fast Fourier Transformation is used to identify periodic variations in the time series $\{x_s\}_{1\leq s\leq N}$. For every day $1 \leq k \leq T$, where $24T = N$, hourly water consumption is considered, and an analysis is conducted on the behaviour of the elements of the series $\{X_n^k\}_{1\leq n\leq 24} \overset{\text{def}}{=} \{X_s\}_{24(k-1)\leq s\leq 24k} \subset \{X_s\}_{1\leq s\leq N}$. Therefore, for the series $\{X_s^k\}_{1\leq s\leq 24}$, with $1 \leq k \leq T$, 24 harmonics are considered while the series are expressed as:

$$\{X_n^K\} = \sum_{j=1}^{24} \alpha_j^k exp(-\frac{2\pi i}{24}(j-1)(n-1)), \tag{2.4}$$

Where α_j^k represents the coefficient for the j^{th} harmonic of k^{th} day, $1 \leq k \leq T, 1 \leq j \leq 24$. The values of the coefficients are determined by:

$$\alpha_j^k = \frac{1}{24}\sum_{n=1}^{24} X_n^k exp(\frac{2\pi i}{24}(j-1)(n-1)), \tag{2.5}$$

$1 \leq k \leq T$, $1 \leq j \leq 24$. For each day, the 24 factors of Fourier transform are evaluated and then the influence of the temperature of the day is analysed. By this point, by predicting the behaviour of the harmonic series $\{\hat{\alpha}\}_{j\,1\leq j\leq 24}^k$ for $k > T$, it is possible to forecast the water consumption $\{\hat{X}\}_{n1\leq n\leq 24}^k$. Kozłowski et al.[10] investigate the dependence of each harmonic on time factor:

$$\alpha_j^k = \beta_j^0 + \beta_j^1 k + \upsilon_j^k, \tag{2.6}$$

and temperature:

$$\alpha_j^k = c_j^0 + c_j^1 t_k + \omega_j^k, \tag{2.7}$$

t_k denotes the mean daily temperature of k^{th} day, $\upsilon_{j\,1\leq k\leq T}^k$ and $\omega_{j\,1\leq k\leq T}^k$ represent series of independent and identically distributed random variables with normal distributions $N(0,\sigma_j^2)$ and $N(0,\gamma_j^2)$, respectively. To estimate the forecasts of hourly water consumption for k^{th} day, $k > T$ the series $\{\hat{X}_n^k\}_{1\leq n\leq 24}$ is calculated with the following expansion:

$$\hat{X}_n^k = \sum_{j=1}^{24} \hat{\alpha}_j^k exp(-\frac{2\pi i}{24}(j-1)(n-1)), \tag{2.8}$$

Where $\hat{\alpha}_j^k$ is the estimation of j^{th} of this harmonic, which can be calculated using either:

$$\hat{\alpha}_j^k = \hat{\beta}_j^0 + \hat{\beta}_j^1 k \tag{2.9}$$

or

$$\hat{\alpha}_j^k = \hat{c}_j^0 + \hat{c}_j^1 t_k, \tag{2.10}$$

$\hat{\beta}_j^0$, $\hat{\beta}_j^1$, $\hat{c}_j^0$, $\hat{c}_j^1$ are values of structural parameter estimators for equations 2.6 and 2.7, respectively. The study shows that the previous models can effectively be used for water forecasting.

The work developed in [11] resorts to mathematical models to forecast water demands. This work is relevant due to the fact that it was able to achieve very good results using a 48h time window with 15 min time steps using measured water demand and static calendar data differentiating itself from the usual 24 h time window with 1 h steps observed in the majority of the literature. It demonstrates that 1 h time steps are too large to describe variations in water demands in some hours of the day. This can affect the optimisation of the pumping scheduling where it is essential to know the exact time to switch the pumps. In addition to that, it is also explained in [11] that different patterns can be found between weekdays, weekends, holiday season, etc. Based on observations in the Netherlands, this work shows that more deviant water demand patterns can be discerned, identifying four types of days with deviant water demand patterns:

- On national holidays (Easter, Christmas, etc.) people and industries behave as on Sundays.
- Weekdays in primary school holiday periods are different from normal weekdays. The most significant difference being the lower peaks in the morning that smoothed out compared to non-holiday days. Different patterns can also be discerned for each holiday (Summer holiday, Christmas holiday, Spring holiday, etc.).
- Individual annual occurring days with a deviant pattern like New Year's Days, Ascension Day, etc.
- Days with deviant water demand patterns that are related with weather conditions. An example of this are dry and sunny days where the water demand resembles the normal pattern in the first part of the day but higher than normal in the late afternoon, presumably caused by people sprinkling their gardens.

This approach differs from literature, where most models tend to have limited number of patterns. Previous works like [15] and [19] use a model with demand patterns for each individual day of the week and only two patterns, weekdays and weekends, respectively. In all three papers it was observed that patterns change according to the season and therefore the models use different patterns for each season.

2.2 Machine learning in forecasting water demands

Camarinha-Matos et al. [20] demonstrates early work in this area by the turn of the 21^{th} century. It uses data from SMAS-Sintra, a Portuguese water distribution company. An important aspect is the opinion and knowledge of the domain experts. Camarinha-Matos et al. [20] show that domain experts have heuristic knowledge about the water network that is useful for several tasks like identifying relevant data variables to be used in the training phase of the machine learning algorithms. Domain experts are also useful to evaluate the significance of the obtained results. Despite using machine learning techniques that are not so commonly used today, this work is useful because it sheds some light on some important aspects of WDN.

Bougadis et al. [13] presents a study performed on data from the city of Ottawa, Ontario, Canada. The goal of this study was to determine the best peak water demand forecasting methods by comparing their performance. The ANN models outperformed regression and time-series models. The authors in [13] even go as far as stating "ANN models are far superior than regression and time-series models . . . ”.

Unlike [13], the work presented in [14] had the sole purpose of comparing the performance of “conventional methods” with machine learning techniques. Eleven years of data, from 1982 to 1992, from Lexington, Kentucky were considered, six for model training and 5 for model evaluation. To represent the “conventional methods” the authors chose Time Series and Regression models. Their results were compared against ANNs and Rule-Based [21] models which are methods that can manually or automatically identify and evolve “rules” that are capable of collectively represent knowledge. Both machine learning techniques offer a better mechanism to forecast short-term water demand than the other methods, with ANNs showing slightly better performance than Ruled-Based models.

As machine learning techniques became the standard to forecast water demands, instead of comparing the performance of these techniques with the more “traditional” mathematical models, studies began to compare the performance between different machine learning techniques. This tendency is encouraged by the fact that recent developments in machine learning has led to the creation of several new methods.

The study presented in [22] compared three machine learning models with the goal of forecasting daily, weekly and monthly demands with several configurations of input climate data. The models chosen for this purpose were Dynamic Artificial Neural Network (DANN) [23], a variant of the typical ANN where the structures of the network can change during the training phase; Focused Time Delay Neural Networks, a network that is dynamic at the input layer and k-nearest neighbours (KNN), where the k nearest neighbour are selected in terms of weighted Euclidean distance. The methods used water consumptions and water production data from March 2003 to April 2010. Two approaches for the daily forecasts were considered: one where the weekdays were distinguished from the weekend days and another where not such distinction was made. The best results were achieved with no partitioning of the days of the week. The three models showed good results with the DANN obtaining the best performance.

Following the trend of comparing machine learning techniques, Tiwari and Adamowski [24] combine ELMs models with wavelet-analysis and bootstrap methods with the goal of assessing the capacity of these modelling approaches in forecasting daily water demands and comparing their performance to that of the more typical ANNs models. The data used in this study was composed by the average daily water demand, maximum temperature and total precipitation, obtained from the city of Calgary, Canada from March 2003 to December 2006. The data was divided into three sets, training, cross-validation and performance testing. The results show that both models with wavelet-analysis, ELM and ANN have the best results out of the rest of the models, ELM being the best model and ANN the second best. This work represents a pioneer study regarding the application of ELM with wavelet-analysis to forecast urban water demands. Its results demonstrate the potential of this modelling technique.

Continuing the study of Extreme Learning Machines, [25] explores the ability of several machine learning methods in order to improve the accuracy of water demand forecasting in urban areas at lead times of 1 and 3 days. The models developed in this study were ANNs, Support Vector Regression (SVR) and Extreme Learning Machines ELM. Mouatadid et al. [25] used data from the city of Montreal, Canada from February 1999 to August 2010, consisting of average water demand, maximum temperature, total precipitation and occurrence of precipitation. The performance deteriorated as the lead time increased, however this deterioration was not considered drastic by the authors. The results indicated that ELMs had a superior performance compared with the other models in both lead times.

One hot topic in the field of machine learning is deep learning and up until now it has rarely been applied to forecast water demand. Guancheng et al. [26] tackle this issue. Their objective was to explore the potential of deep learning in water demand forecasting, compare its performance with conventional ANN models, and investigate a method to correctly predict errors with the purpose of further improving the performance of the models. In their study, a Gated Recurrent Unit Network (GRUN) was developed, along with an ANN. Both models were used for 15 min online forecasts where the model received data updates making each prediction on real observed data, and for 24 h forecasts with 15 min timesteps where the output of the previous moment forecast was used as input for the next moment until 96 values were forecasted. A correction model was implemented with the goal of decreasing the accumulated error that occurs between forecasts. This model takes the output of 96 values of the 24 h forecast and approximates them to the real values. Two more model variants can then be considered: ANN-correction and GRUN-correction. The correction model consists of an ANN model with one dense layer trained with the predicted values obtained from the GRUN model or ANN model, and sample labels are the observed values obtained from the field. This model is only applied in the 24 h forecast. This study uses data collected between February 2016 and January 2017 from the city of Changzhou, China, divided in two sites, a mainly residential area and an industrial area. The results show that as the forecasting horizon increases, the performance of the models deteriorates. The GRUN model demonstrates better performances in both 15 min forecast and 24 h. The application of the correction model improved the prediction accuracy of the models, making the GRUN-correction model the one with the best overall performance. In [26] it is concluded that:

- The proposed deep learning-based method achieves accurate and reliable water demand predictions for 15 min and 24 h. The GRUN predicts more accurately and is more stable than the conventional ANN model;
- The correction module can enhance the performance of both ANN model and GRUN model.

Special attention must be given to the work developed by Antunes et al. [6]. This study was developed in the same context of this work and uses similar data. Analogous to other previously mentioned studies, Antunes et al. [6] explore and analyse several machine learning techniques for short-term water demand forecast such as KNN, SVR, Random Forest Regression and ANNs. The results of these techniques are compared to traditional mathematical models. Several model benchmarks are presented on different datasets with the intention of analysing their forecasting performance. The data is provided by two water companies, one from the northern region of Portugal and the other from the central region, dating between September 2012 to July 2013 and September 2015 to December 2016, respectively. Data from the two regions has both quality and quantity errors, so only selected points of the networks were considered. Even so, it was possible to identify some problems regarding outliers, here defined as values that were not in the range between the average and margin of 3 times the standard deviation, and absence of data. The absent values were replaced with the global average. A second iteration of this process was done to reduce the impact of the errors of the first iteration, after the normalization of values. This study concluded that a "small" ANN with the Limited Memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) learning algorithm and the Rectified Linear Unit (ReLU) activation function produced the best results respecting the criteria of 24 h forecast window and using approximately 2 weeks of previous water demand observations.

2.3 Climatological Data

The inclusion of weather data in the prediction models is not a topic with a unanimous opinion within the literature. Generally, studies recommend its use, reporting improvements in their forecasts, with some exceptions. However, the choice of climatological variables is a controversial topic, with no apparent agreement in which variables should be considered. The use of climatological variables has been reported in statistical as well as in machine learning models. There are even some that do not recommend their use at all.

Howe and Linaweaver [27], published in 1967 a study in which regression models use climatological variables such as summer precipitation, summer potential and maximum day potential evapotranspiration, average summer and irrigable area per household. Some variables have proven to be more effective than others depending on the model.

An exploratory analysis was developed in [13] to quantify the relationship between climatological variables and peak water demand. The developed ANN models were hypothesized using weekly maximum temperature, weekly rainfall amounts and occurrence of rainfall in the week. The results of the exploratory analyses showed that there is a link between temperature at a certain time and the peak weekly water demand. Regarding rainfall, the strongest link was found between peak demand and rainfall amount. The study concluded that the ANN models obtain better results when employing previous weekly demand along with the week's temperature and rainfall amount.

A similar study, [28] also implements several ANN models with different climatological input variables in order to determine which variables were more relevant to improve the peak weekly water demand forecasts. Contrary to what was stated in the previously mentioned study, in this work it is shown that it is possible to better describe weekly peak water demand with the occurrence/nonconcurrence of rain instead of rainfall amount.

Bakker et al. [11] explains that there is potential drawback in using weather information since it is an extra input for the model and, typically, weather information has low availability and reliability. Therefore, the implementation of a water forecasting models with weather information is more difficult and less reliable. The study states that despite existing a relation between weather conditions and water demands, there is an advantage in not using weather data from an implementation standpoint. This study presents a method that manages to take into consideration weather condition despite not using weather data as input, as explained in 2.1.

Antunes et al. [6] also analyse the use of weather features in order to improve the water demand forecasts accuracy and tests models with and without weather data. This study considers that water demand forecasts depend on the previous water consumption observations paired with the predicted weather conditions for the next day. For example, to predict the water demand for tomorrow at noon, the forecast depends on the water consumptions from previous days at that time and the predicted weather conditions for tomorrow at noon. It was concluded that the use of weather features did not bring any improvements to the performance of the models since its best results are obtained with models that did not consider weather features as input. These results may be correlated with the sets of data that are being used.

2.4 Modelling water distribution networks

Much like the forecasting techniques discussed in the previous section, the methods used to model WDNs started with mathematical hydraulic models and then shifted to machine learning techniques. Previous work that has been done on this topic usually involves three main methods that come with their own advantages and short comings.

It is worth noting that there are not as many studies regarding water distribution network modelling as there are studies about forecasting water demands. Some works from the late 1970's show early efforts of simulating the behaviour of WDN, such as [29]. But until this date the number of studies that tackle this topic are scarce, especially those that use machine learning.

2.4.1 Empirical Models

Since the behaviour of WDNs is time-variant, spatially distributed and non-linear,the networks have traditionally been represented by empirical models like mass-balance models which consist of functional relationships between some characteristics of the network such as storage tank levels, pump-discharge and flows. The weights of the functional relationships can be calculated with linear regression or other methods. The work of Sterling and Coulbeck [30] is an example of an incorporation of these methods in a dynamic programming solution with the goal of optimising pumping stations of the Doncaster District, United Kingdom. Mass-balance models allow a rapid system response since these do not require a lot of computational capabilities. However, these are more suited for regional water supply schemes rather than WDN.

2.4.2 Simplified network-hydraulic models

Simplified network-hydraulic models can be considered as an intermediate step between empirical models and full hydraulic network simulation. As their name suggests, these models reproduce the network from a simplified perspective taking into consideration few network characteristics. Some cases approximate the network using a macroscopic model. Others analyse the network using a series of linearised hydraulic equations. In certain cases it might be possible to represent the network by a simple linear model. Jowitt and Germanopoulos [31] present a solution for pump scheduling optimization on Buckinghamshire, England. It uses a linear model that is based on some assumptions such as the internal network pressures are expected to remain within certain bounds, etc. These assumptions may not always be true. It is worthwhile noticing that such model is site-specific and must be judged on that basis.

2.4.3 Network Simulation Models

Empirical and simplified network-hydraulic models require less computational effort but their accuracy and robustness are questionable. Network simulation models have the capacity to model the nonlinear dynamics of water networks by solving hydraulic equations based on the quasi-steady state assumption. In the case of water supply and distribution systems, these equations include conservation of energy and mass. Network simulation models are very adaptable to physical change and more robust than either empirical or simplified network-hydraulic models. However, they require a considerable amount of data to formulate and a significant amount of effort to calibrate. A calibration process consists of several adjustments of many characteristics of the elements of the network, such as, pipe roughness, water leaks, etc. Any minor change to the model requires a full calibration. In addition, more calibration requires extra computational effort. Since 1990, software packages for generic hydraulic simulation have been developed such as EPANET [9], WATNET [32], WESNET [33], etc. These softwares have been widely used for operational development of control strategies, operational analyses and real-time near-optimal control. However, their use is somewhat impractical in some cases. Tavares [34] describes in a very detailed way the use of the software EPANET to model the regional water distribution system of Carvoeiro, Albergaria-a-Velha, Portugal. Despite some efforts, this study was not able to achieve a totally calibrated simulation stating a high sensitivity between water losses coefficients, pump and valve control routines and the results that were obtained for simulations of flow and pressures variation.

2.5 Machine learning in hydraulic simulation

Rao and Alvarruiz [35] present an entirely new approach for using machine learning in the water distribution systems focusing on operational control rather than on planning or design exercises as all other applications known until that date. This study describes the development of an ANN that takes as input:

- Control variables representing pump settings;
- Valves settings;
- Water demands for a certain time period;
- Storage tank levels;

And outputs:

- Pump power consumption for a certain time period;
- Pressures in specific network nodes;
- Flow in specific network pipes;
- Storage tank levels.

The use of separate models, one for each output variable, was considered but it was found that the accuracy of the individual predictions did not improve over the use of one composite network for predicting all the outputs. The model was applied to a hypothetical network, the 'Any Town' network [36], since it is simple, well-documented and extensively modelled previously. The results are compared to those of the EPANET model on the same network. A high degree of accuracy and a 10-fold reduction in the computational time required by conventional simulation models was reported in [35].

Based on what is presented in [35], Salomons et al. [7] later applied the same technique to real world data from Haifa-A, Israel. The goal of this study was to reduce the operational costs of water distribution system by finding the near-optimal control process, that is the one that matches with the best energy tariff structure. The authors explain that the typical operating regime of the storage tanks depends solely on their water levels and no special attention is given to the energy tariff structure. When the level goes below a certain margin the pump is switched on and vice-versa. The problem is formulated by an objective function that minimises the overall cost of delivering the required amount of water in a given period. A 24 h operating horizon was adopted since water distribution systems operate on a daily cycle, and a time step of 1 h was chosen considering the impact of the computational burden. Since the objective function has the purpose of minimising pumping costs, the optimisation process will avoid scheduling pumping operations. This will eventually draw the water thereby emptying the tanks towards the end of the operating period causing the water level of the tanks to be too low to fulfil the demands of the next operating period. To prevent this situation an end-state constraint is applied. This consists on leaving each tank with a prescribed water level at the end of the day. Failure to impose this constraint will result in not being able to refill the tanks in time to suffice the needs of the clients on the next day. To achieve this, a method that is referred as a Dynamic, Real-time, Adaptive Genetic Algorithm – Artificial Neural Network (DRAGAN-ANN), an ANN combined with a Genetic Algorithm (GA)

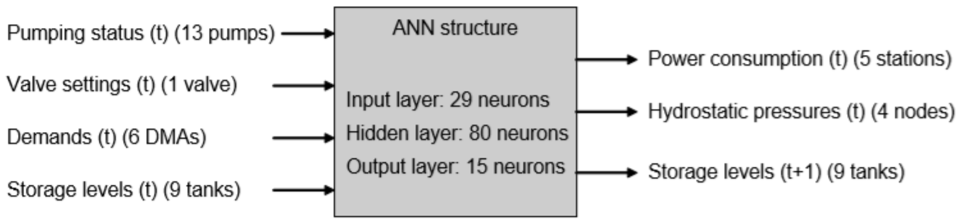

Pumping status (t) (13 pumps)
Valve settings (t) (1 valve)
Demands (t) (6 DMAs)
Storage levels (t) (9 tanks)
ANN structure
Input layer: 29 neurons
Hidden layer: 80 neurons
Output layer: 15 neurons
Power consumption (t) (5 stations)
Hydrostatic pressures (t) (4 nodes)
Storage levels (t+1) (9 tanks)

Figure 2.1: Structure of the ANN used in [7].

was developed. The ANN replicates a conventional hydraulic simulation model, which is significantly more computationally efficient than using a simulation model directly. The cost of each potential solution proposed by the GA is estimated by the ANN. In order to capture the domain knowledge of a conventional hydraulic simulation model, the ANN is used to map a multivariate space (inputs) to another (outputs). It can be regarded as a input/output model. That said, to train the ANN, a set of input/output pairs was generated by EPANET. Figure 2.1 shows the structure of the ANN model along with its inputs and outputs. To forecast the water demands a method, which consists of a combination of Fourier series and time-series analyses is used. Salomons et al. [7] reports a potential cost reduction of about 25%.

In an attempt to model a water distribution network of greater dimensions than the ones previously mentioned, Behandish and Wu [37] tackle a problem in which the accuracy of a larger ANN cannot be compromised. Due to its size, the ANN requires a larger training set and, consequently, a fair amount of computational effort. The case study in question is the Oldham network located in Greater Manchester, England, which is constituted by 3500+ pipes, 3500+ junctions, 420 valves, 5 reservoirs, 12 storage tanks and 19 constant speed pumps. An important aspect that is worth noticing is that, in addition to the inputs that are considered in [7], Behandish and Wu also consider the aggregated demand and the demand differential (see Figure 2.2). As explained in [37], a single ANN model to predict all tank levels and pumping energy creates some problems such as an inevitable decline of the accuracy and overall RMSE increase. On the other hand, there are advantages in using multiple independent sub-ANNs for certain outputs. These sub-ANNs can be independently trained, tested and modified, the degree of dependence of a particular output on different inputs can be investigated by monitoring the RMSE. This allows the sub-ANNs to only have inputs that are relevant to their outputs. Training experiments show that the total pump energy can be predicted with high accuracy solely from the pump statuses showing negligible dependence on tank levels and water demands. However, there is a strong dependency on some pumps, tanks and consumer demands in tank level prediction. To effectively identify the inputs that immediately affect a given tank, 12 sub-ANNs, like the one depicted in Figure 2.2, were trained, one for each tank. The input space of the networks is composed of all the variables regarding pump statuses, tanks levels and water demands, a total of 24 inputs. Each out of those 24 outputs of the sub-ANN is then analysed individually for its sensitivity to the specific input of that given network. Each input is set to its minimum, maximum and

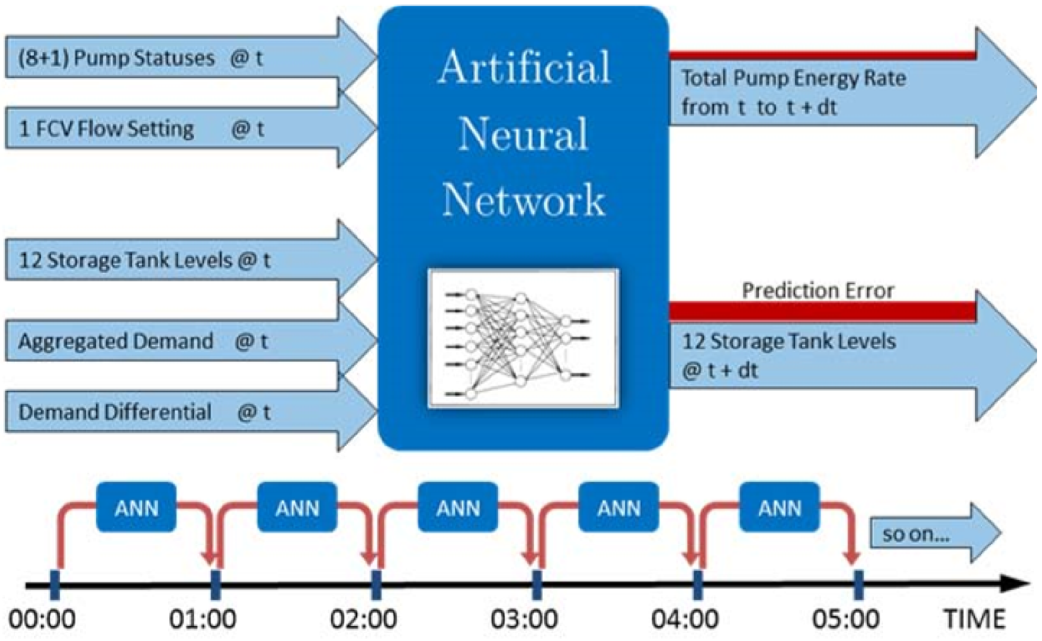


Figure 2.2: Input/Output mapping on of the sub-ANN. Adapted from [37].

average; and a noise of 0%, 2%, 5% and 10% is introduced in order to observe the effects on the RMSE. For this sensitivity analyses, minor changes are ignored. The results of this analysis allowed the development of sub-ANNs of reduced topologies and a structure that the authors call Sensitive-Input Separate-Output. These sub-ANNs took significantly less time to reach the same level of accuracy of other ANN topologies that required much more computational effort.

The previously explained sensitivity analysis is described in [38] as tedious and time consuming. Wu et al. [38] show a solution that eliminates the need for it as well as using multiple ANNs to represent the whole water distribution system. They proposes the use of deep learning. The greatest advantage of deep learning is its capability of automatically extract features by training the system with unlabelled data. Its architecture consists of several layers that can be individually trained to extract a higher level of abstractions from the input. Deep Belief Networks (DBN) [39] are a deep learning architecture that are composed of stacked Restricted Boltzmann Machines (RBM), being the methodology adopted in [38]. The results are compared to conventional ANN and the sub-ANNs presented in [37]. The DBN achieves slightly better accuracy without requiring sensitivity analysis in its construction.

CHAPTER 3

Methodology

This chapter, regarding water demand forecasting, explains how the used data is processed and arranged. Also, the various forecasting methods that were developed are presented. Then, the methodology adopted to simulate water networks is described along with its inputs/outputs. Finally, it is explained how the data used by the simulation method was generated.

3.1 Developing the water forecasting module

As explained in Chapter 2, a literature analysis concerning this topic indicates that ANNs are currently very popular and are capable of obtaining very good results [6]. More and more studies are being developed where the performance of ANNs is compared with other, more recent, machine learning techniques. Such is the case of [25] and [24], both works show that ELMs are capable of outperforming ANNs in short-term water demand forecasting. Another example of this is [26], where an implementation of a Gated Recurrent Network was able to obtain better results than ANNs.

Since it is not possible to draw a definitive conclusion on the best technique, several methods were implemented in order to assess their performance on short-term forecasting using the data mentioned in 3.1.2.

3.1.1 Data Processing

Processing the data and extracting the features necessary to train the models is an essential step in developing any machine learning solution which can influence the model's structure and its performance. With that in mind, searching for the models that have the best performance in forecasting water demands implies also studying which set of data provides the best results for that model.

All the data is processed beforehand, adopting the methodology of [6]. Outliers are identified by singling out values that are above or below 3 times the standard deviation.

Values that correspond to the upper and lower limit of the allowed range values are then assigned to those outliers. A second iteration of this process is executed to mitigate the impact of the errors that were previously found.

After the pre-processing procedure that was previously described the data must be rearranged, matching labels to their respective features. Some data arrangements were analysed in order to find those with the best performance. The following sections explain the data arrangements that were considered.

3.1.2 Data sources

In order to study water demand forecasting, this study uses three sets of data provided by two Portuguese companies that manage water supply systems. The data is stored by saving the cumulative amount of water that passes in certain points of the network, which means that the amount of water that is demanded at a given time period is the difference between the amount of water that has passed up to that point and the amount water that was registered in the previous time period.

The first two sets of data were provided by a water utility in the northern region of Portugal, WD2 and WD4. They represent data collected from two different points of a network. The observed water demands date between September 2012 and July 2013. A more detailed description of this consumption points can be consulted in the work of B. Costa [40].

The third and fourth sets of data come from the central region of Portugal. The sets represent two consumption points of the network, and they are referred to as WDC and WDV. The data was recorded between January of 2018 and December of the same year.

Since each dataset represents a different water consumption point of their respective network, the amount of water that is delivered to each point at any time of the day is very distinct. This means that the time series that represents water demands at given times may widely vary from a point to another. Therefore, when forecasting a point's water demands the structure and hyperparameters of a model should be optimised.

3.1.3 Artificial Neural Networks

ANNs [41] are the most common and popular tools used in machine learning. Neural Networks consist of several layers that are divided in three categories: the input is the first layer that comes into contact with the inputs that are feed to the network, its size depends number of input features; the output layer, as its name suggests, it is the last layer of the network and its size depends on the size of the output space; the hidden layers are in-between the previous layers, hence their name. ANN's can have countless topologies depending on the number of neurons on each layer. Regardless of their size or complexity, they all share similar trades:

- Neurons (perceptrons) are small scale linear classifiers that map an input x, a real value, to a value between 0 and 1. Its resulting output can be interpreted as the probability of an input belonging to a certain category.
- Connections between neurons of each layer, also called synapses, associate inputs to the neuron.

In all layers of the network, all inputs must be connected to all neurons. Inside each neuron the sum of the product between the weights associated with each connection, w, and their respective input, x, is calculated, by:

$$\sum_{i=0}^{m} w_i x_i + b_i,\ 0\ \leq\ i\ \leq\ m, \tag{3.1}$$

where m is the number of inputs, b represents a bias, a parameter that determines the location of the decision surface. A function, more commonly known as an activation function, is then used to map $\sum_{i=0}^{m} w_i x_i + b$ into a real value between 0 and 1, $\sigma\left(\sum_{i=0}^{m} w_i x_i + b\right) = \{y \in \mathbb{R} \vee 0 \leq y \leq 1\}$. A common function that is used for this purpose is the sigmoid function:

$$\sigma\left(x\right) = \frac{1}{1 + e^{-x}}. \tag{3.2}$$

The output of a neuron, $a_n^{(1)}$, can be expressed as:

$$a_n^{(1)}\ =\ \sigma(\sum_{i=0}^{m} w_i x_i + b). \tag{3.3}$$

Where (1) indicates that the corresponding parameters are in the first layer and n is the number of neurons of that layer.

For the network to learn, a training algorithm must be applied. Commonly, the training algorithms consist in an iterative procedure that minimise a cost function. This is accomplished with adjustments to the weights and biases of the several neurons of each layer from one end of the network to the other. This process is known as error backpropagation [41].

Implementation

As mentioned before, the data used in this work is similar to the data used in [6], it is expected that the topology of the ANN that is used in that study can achieve the same results using this work's data. With that in mind, the implementation of the ANN was based in what was described by Antunes et al. [6] which respected the following criteria:

- Use either ReLU or Identity activation function. The ReLU activation function is defined by $f(y) = max(0, y)$, and the Identity function, also known as linear, simply by $f(y) = y$.
- The network must have a "small" dimension. Which it will be assumed, judging by the results, that it refers to networks that do not exceed 2 hidden layers nor 75 nodes in each layer.

- During the training procedure use Limited Memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm as the optimisation function. This is a computational efficient implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, a second order optimisation technique [42].

The best results presented in [6] were generated with ANNs that followed the previous guidelines and had slight variations between them regarding the size of their hidden layers. For example, the two models that obtained the best performance one had 2 and 25 hidden nodes, and the other had 8 and 25 hidden nodes.

Two separate implementations of the models were conceived using two distinct libraries, Scikit-learn [43] and Keras [44], which will be explained with more detail further on. This Scikit-learn implementation was developed as a sanity check for the work developed in [6] and as a performance baseline for future improvements.
The Keras implementation was developed with the goal of trying to improve the performance of ANN models. Since Keras allows for greater model customization, new alterations were made to the previous mentioned ANNs models. One feature that Keras offers that is not available in the Scikit-learn library is the possibility to add dropout, a technique to reduce overfitting. Dropout makes the model trade training performance for generalization, making the training loss error somewhat worse. This is achieved by deactivating a percentage of a layers' neurons during training which forces the layer to learn the same "concept" with different neurons [45]. Keras also allows the implementation of distinct activation functions on distinct layers. A parameter search was required to find the parameters that best suit the model, taking in consideration that dropout affects the training performance.

Data arrangement

In order to train the models, it is necessary to create a dataset of input features associated with a given output, in this case, a water demand at time t, denoted as D_t. This dataset must be constructed from a stationary time series of water demands $[D_{t-\gamma}, ..., D_{t-2}, D_{t-1}]$, where γ is the number of previous recorded water demands. This can be accomplished by rearranging the values of the time series in several ways. Kozłowski et al. [10] present a method where a time series of water demands is divided in 24 sub series (one for each hour of a day). Each demand that is being foretasted is represented by the demands that took place beforehand on a 24 h periodic step. This method is explained in more detail in Section 2.1. Also taking in consideration the procedure described in [6], which results in a good model performance, the data was also arranged so that to predict a water demand at time t, D_t, the model is given the values of the demands that took place a day prior to t, which means that D_t is predicted using $\left[D_{t-\ 24(n-1)}, ..., D_{t-24\ *\ 3}, D_{t-24\ *\ 2},\ D_{t-24}\right]$, where n is the number of demands of prior days that are being considered, and, consequently, the number of features that are being used to predict D_t. For the sake of simplicity, this data arrangement is referred to as dd.

3.1.4 Extreme Learning Machines

Extreme Learning Machines were proposed in 2006 [46] and are based on a specific architecture of ANN: Single Layer Feed-Forward Neural Network (SLFN). As it name suggests, they are composed of an input layer, a single hidden layer and an output layer. This method was proposed with the goal of contrasting the slow learning speed, the frequent convergence on local minima, and the requirement of many learning steps to obtain a good performance of feedforward neural networks. Its input weights and hidden layer biases are randomly chosen and the output weights are determined through the inverse operation of the hidden layer output matrices. Huang et al. [46] states that Extreme Learning Machines' learning speed can be thousands of times faster than feedforward network learning algorithms like backpropagation, while achieving a better generalization performance.

Implementation

Since the previous studies that use ELMs as their methodology of choice utilize data that is not related in any way to the data being used for this study, there is no guarantee that the parameters used to tune the models presented in the studies will have a good performance with this data. For that reason, ELM models were initially conceived with no consideration for any specific parameters. An extension to the Scikit-learn library that provides ELMs implementations can be added [47], it provides wider range of options for activation functions.

Data arrangement

The ELMs were tested using 2 distinct data arrangements. The first was created in similar fashion of what is explained in 3.1.3, denoted as dd. The second data arrangement uses as features the demands that took place immediately prior to the demand that is being predicted, meaning that, to predict a demand at time t, D_t, the model is given $[D_{t-n},\ ..., D_{t-2},\ D_{t-1}]$, where n is the amount of hours before t, referred to as Prior Demand Vector (pd).

3.1.5 Recurrent Neural Network

As Chung et al. [48] explains, Recurrent Neural Networks are an extension of conventional feedforward neural network, which are able to handle variable-length input by having a hidden state whose activation at each time is dependent on that of the previous time.
Formally, given a sequence $x = \ (x_1, x_2,\ ...,\ x_T\ )$, the Recurrent Neural Networks (RNN) update its hidden state h_t by

$$h_t = \begin{cases} 0, & t{=}0 \\ \emptyset(h_{t-1}, x_t), & otherwise \end{cases} \tag{3.4}$$

where $\emptyset$, denotes a nonlinear function. The update of the recurrent hidden state in 3.4 is implemented as

$$h_t = g(W_{x_t} + Uh_{t-1}), \tag{3.5}$$

where g is a bounded function such as a logistic sigmoid function, W_{x_t} is the weight matrix at the step x_t and U is the hidden state matrix.

Given the current state h_t the RNN outputs a probability distribution over the next element of the sequence.

It has been shown that it is difficult for RNNs to capture long-term dependencies due to the vanishing and exploding gradient problem [49]. Gradient-based optimisation struggles due to variations in gradient magnitudes and the effect of long-term dependencies being hidden by effect of short-term dependencies. One approach to deal with this problem is to use gating units.

3.1.6 Gated Recurrent Units

Gated Recurrent Unit (GRU)s were proposed by Cho et al. [50] to make each recurrent unit adaptively capture dependencies of different time scales. As explained in [48], GRUs have two recurrent gates that control the flow of information inside the unit, a reset gate and update gate. The activation h_t^j at time t is a linear interpolation of the previous activation h_{t-1}^j and the candidate activation $\widetilde{h}_t^j$:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \widetilde{h}_t^j, \tag{3.6}$$

The update gate, z_t^j, decides how much of the previous state should be kept. The gate is computed by

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j. \tag{3.7}$$

The candidate activation $\widetilde{h}_t^j$ is computed by:

$$\widetilde{h}_t^j = tanh(W x_t + U(r_t \odot h_{t-1}))^j, \tag{3.8}$$

where r_t is a set of reset gates and $\odot$ is an element-wise multiplication. When r_t is close to 0 the reset gate makes the unit act as if it is reading the first symbol of an input sequence, allowing it to discard the previously computed state.

The reset gate determines how to combine new input to previous memory, in similarity with the update gate, the reset gate is computed by:

$$r_t^i = \sigma(W_r x_t + U_r h_{t-1})^j. \tag{3.9}$$

3.1.7 Gated Recurrent Unit Network

A recurrent neural network with gated recurrent units or gated recurrent unit network model was developed based on the architecture presented in [26]. This work was previously analysed in Chapter 2. Rather than forecasting water demands with a 15min timestep, the model was adapted for 1 h timesteps. Figure 3.1 shows a representation of the structure of the GRUN model. It consists of three GRU layers that process data at different time periods. The GRU layers put the water demand data through an nonlinear transformation that produces a memory state for past water demand, establishing dependencies among demands at different time periods. The output of the GRU layers is then concatenated in a merge layer which is connected to a set of regular fully connected layers, all layers except for the output layers which uses a linear activation function, use the ReLU activation function.

Data arrangement

For the GRUN model two distinct data arrangements were experimented with.

The first is based on the one presented in [26]. The recent time period data refers to water demands that are immediately before the desired prediction at time t (D_t), the recent sequence can be represented with the vector $[D_{t-i}, ..., D_{t-2}, D_{t-1}]$, where i is the number of 1h timesteps of water demand data. The near time period represents the temporal dependence of the previous day (D_{t-24}). Let j, be the number of timesteps of data that were close to time t of the previous day, the near time period can be represented by $[D_{t-24-j}, ..., D_{t-24}, ..., D_{t-24+j}]$. The distant time period represents the temporal dependence of two days prior to time t (D_{t-48}), and it can be expressed in similar fashion of the near time period, $[D_{t-48-j}, ..., D_{t-48}, ..., D_{t-48+j}]$.

The second data arrangement is similar to dd, explained in 3.1.3. However, the feature input vector is divided into three sub-vectors: recent, near and distant. So, for example, if the feature vector is $[D_{t-24*14}, ..., D_{t-24-1}, D_{t-24}]$, the recent sub-vector will be $[D_{t-24*4}, ..., D_{t-24}]$, the near sub-vector $[D_{t-24*9}, ..., D_{t-24*5}]$, and the distant sub-vector $[D_{t-24*14}, ..., D_{t-24*10}]$. Like their name suggest, the recent sub-vector corresponds to the demands that are the closest to the value that is being predicted, D_t, unlike the distant sub-vector which corresponds to the features that are the furthest way from D_t. One restriction is imposed by this arrangement, the original feature vector must be divisible by three.

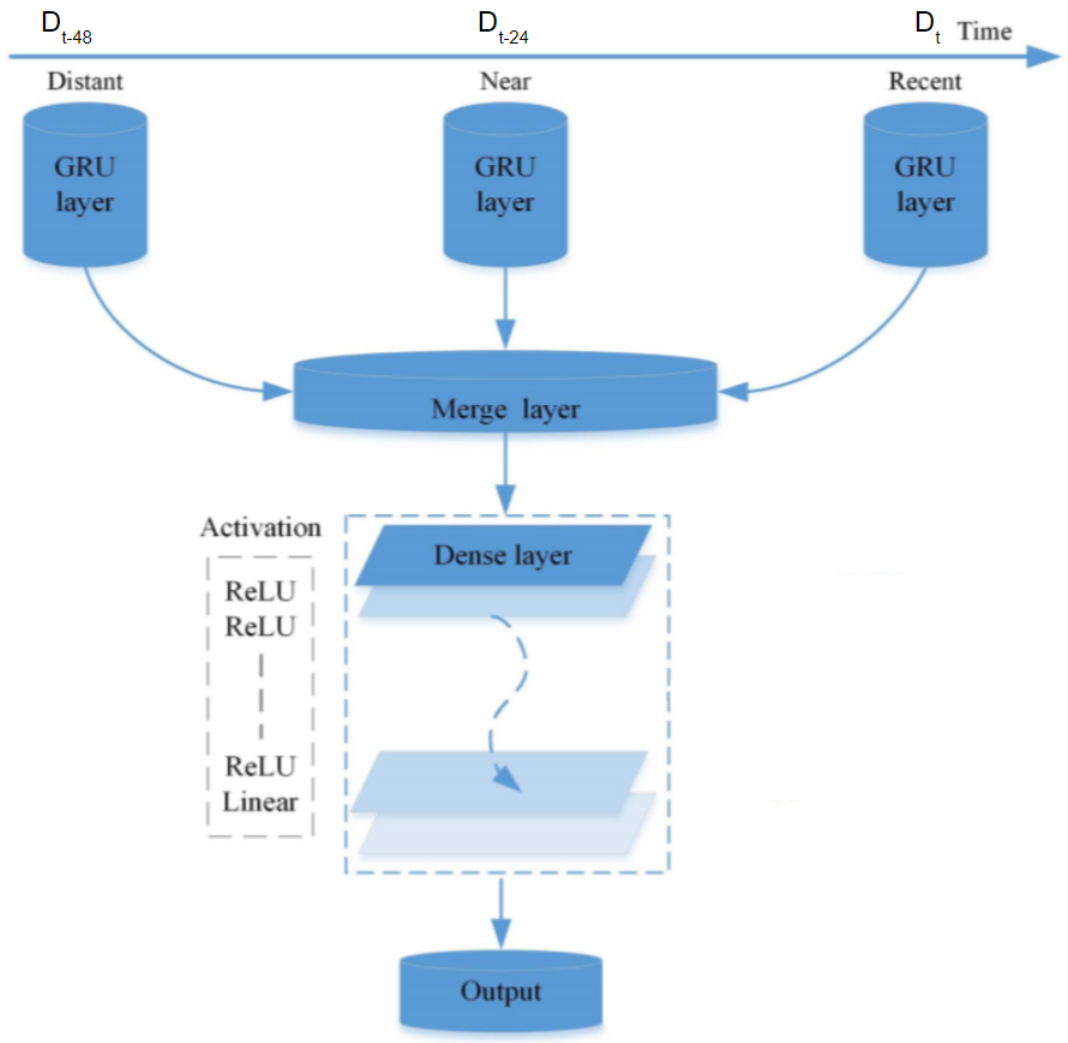


Figure 3.1: GRUN model structure. Adapted from [26].

3.1.8 Bidirectional Gated Recurrent Unit Network

This experimental model was based on the design of the previous model. The two models are very similar but instead of having a GRU layer for each sub-vector of data, it has a BiGRU. These layers can be described as a wrapper that averages the outputs of two conventional GRUs. Figure 3.2 illustrates a possible representation of such layer. One of the GRU receives data in chronological order and other receives inverted, in order words, data that is ordered in opposition to the chronological order. Techniques like this one are commonly-used in Natural Language Processing where a certain word on a sentence can be influence by the words that both precede and succeed that word. In this context, the assumption is that a certain water demand that also be influenced by the demands that happen before and after its occurrence. For obvious reasons, it is impossible to know the water demands that occurred after a demand that did not happen. The previous assumption is not possible if the data that is being considered to the forecast is only water demands that immediately precede the demand that is being predicted. Therefore, some constraints had to be applied: the

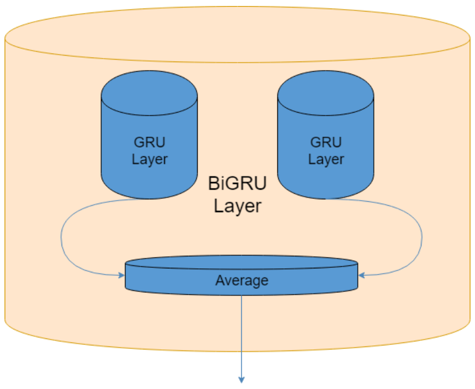


Figure 3.2: Structure of a BiGRU.

bidirectional data is only available for the distant and near layers; D_{t-24} is the maximum value taken in consideration for the bidirectional data. Since each BiGRU layer requires 2 inputs, BiGRUN model needs double the inputs of the previous model.

Data arrangement

Given the similarities between the GRUN and BiGRUN, the data arrangements from 3.1.7 were adopted with small alterations. There are now 3 more inputs that need to be considered, the chronological inverted recent, near and distant inputs, denoted as $recent_{rev}$, $near_{rev}$, and $distant_{rev}$, respectively. In both data arrangements the 3 original inputs, explained in 3.1.7, are not altered.

In the case of the first data arrangement, to forecast a demand at time t, D_t, the reserved inputs can be defined as:

$$distant_{rev} = [D_{t-48+j+|distant|-1}, ..., D_{t-48+j+1}], \tag{3.10}$$

$$near_{rev} = [D_{t-24+j+|near|-1}, ..., D_{t-24+j+1}], \tag{3.11}$$

$$recent_{rev} = [D_{t-1}, D_{t-2}, ..., D_{t-i}], \tag{3.12}$$

as was explained in the in 3.1.7, i is the number of water demands that occurred immediately before the demand that is being forecasted, and j the number of timesteps close to t. Since the procedure adopted to create the $near_{rev}$, and $distant_{rev}$ sets it is not applicable to $recent_{rev}$, the set is just the recent set in reverse order.

As for the second data arrangement, considering n as the total number of features, $\left[D_{t-24*(n-1)}, ..., D_{t-24*2, D_{t-24}}\right]$, the reversed inputs are:

$$distant_{rev} = \left[D_{t-24*9}, ..., D_{t-24*5}\right], \tag{3.13}$$

$$near_{rev} = \left[D_{t-24*4}, ..., D_{t-24}\right], \tag{3.14}$$

$$recent_{rev} = [D_{t-5}, ..., D_{t-2}, D_{t-1}]. \tag{3.15}$$

The $distant_{rev}$ and $near_{rev}$ contain the elements of the near and recent set in reversed order, respectively. Like the first data arrangement, the $recent_{rev}$ contains the water demands, in reversed order, that happen some timesteps before the demand that is being forecasted

3.2 Developing the simulation model

WDNs are complex systems with many distinct attributes that can be useful to be tracked. Since the goal is to predict the behaviour of the network in order to optimize the pumping energy cost, two features are needed: the water level of the tanks and the energy consumed at the end of each time step.

An important aspect to keep in mind when choosing a solution is the complexity of the water distribution network from which its behaviour has to be replicated. The works that were previously analysed demonstrate solutions of varying complexity. [37] and [38] use a GPU-optimized ANN and Deep Belief Networks, respectively, to meta-model a water distribution system of great complexity. In both cases, the computational effort is an issue taken into consideration. [7] tackles a smaller water distribution network of considerably less complexity using a simple ANN. The next sections will explain the networks that will be considered for this study. Given their complexity and comparing with the networks used in [7], [38], a ANN seems a reliable solution.

Nothing in the literature indicates that an 1 h timestep is not enough to faithfully simulate the behaviour of the network. As such, a conservative 1 h timestep was adopted. Despite this, some results are generated with a 15 min timestep with the purpose of briefly examining the performance of the model with that timestep.

3.2.1 Data Generation

ANNs are capable of accurately representing non-linear relationships. However, in order to achieve this, they require a fair amount of training data. A set of inputs/outputs that map the behaviour of the network for several diverse input scenarios has to be created. In this work, the hydraulic models are used to replace real observations. This is due to the difficulty of obtaining a large amount of reliable data from the WSSs. This choice has also the advantage of leaving a reference solution for validation and cross-validation. Therefore, virtual observations are used instead of real ones. The networks can be accurately emulated using the EPANET software or mathematical models that use differential calculus.

Regarding the inputs related with water demands, Behandish and Wu [37] only used as input the differential demand and aggregated demand. Although this has been proven to work, it was considered that having the individual demands of each consumption point as model inputs may result in a more precise simulation. This is due to several factors like the layout of the network, the elevation of each element, etc. A simple example where individual demands for each point are preferable over having an aggregated demands is depicted in Figure 4.5. In that case, a Water Consumption Point (WCP) is only supplied by the tanks that it is directly attached to, the network would behave on a completely different manner if theWCP weas located between the tank and the pump since its demands could be sufficed with either water directly from the pump or water from tank. The water level of the tank can vary in two very distinct ways in both situations. The hydraulic simulation model can be described as an input-output model like the one depicted in Figure 3.3, where:

- $[P_t^1,\ P_t^2,\ ...,\ P_t^n]$ – input vectors of values in $[0, 1]$ range representing a fraction of the given timestep from t to Δt, where a certain pump was on at time t. Given by $P_t^i = \frac{t_{on}^i}{\Delta t}$, where P_t^i and t_{on}^i are the status of pump i and the time that pump i was active in a certain timestep Δt, respectively. $P = 1$ means the pump was always active between t and Δt. n represents the total number of pumps.
- $[S_t^1,\ S_t^2,\ ...,\ S_t^y]$ – input vectors of values representing the water level, in meters, of the certain tanks at time t, where y represents the total number of storage tanks.
- $[D_t^1,\ D_t^2,\ ...,\ D_t^j]$ – input vectors of values representing the water demands, in cubic meters, of each consumption point of the network between time t and $t + \Delta t$, where j represents the total number of consumption point on the network.
- A_t - an input vector of values containing the aggregated demand of every operational window (*e. g.* 24 h) at time t in cubic meters. Given by $A_t = \begin{cases} \sum_{i=1}^{j} D_t^i,\ t=0 \\ A_{t-1}+\sum_{i=1}^{j} D_t^i,\ t>0 \end{cases}$
- $[S_{t+\Delta t}^1, S_{t+\Delta t}^2, ..., S_{t+\Delta t}^y]$ – output vectors of values representing the water level, in meters, of the certain tanks at time $t + \Delta t$, where y represents the total number of storage tanks.
- $E_{t+\Delta t}$ – output vector of values representing the energy consumption, in kWh, of the pumps between time t and $t + \Delta t$.

Generating the dataset of input-output pairs that represent the consequent behaviour of the network (output) when faced with certain circumstances (inputs) means that, firstly, it is necessary to create a set of parameters that represents a wide variety of input combinations. This provides a richer training process for the model improving its generalization capabilities. The pump status and the storage tank levels were generated using the Monte Carlo method, generated by random sampling. The water demands were mathematically simulated and randomly sampled from a dataset; the aggregated demand is then calculated. After the creation of the inputs, these are passed as arguments to a previously mentioned system that emulates the network.

Just like in the case of the timestep that was previously mentioned, at the time the decision concerning the amount of data that would be generated was made, it was not known how much data was require for the model to precisely simulate a WDN. As such, it was considered that a dataset of 24000 samples was enough. No consideration was given to the computational effort required to create the dataset as it would only be created once. A thousand batches of

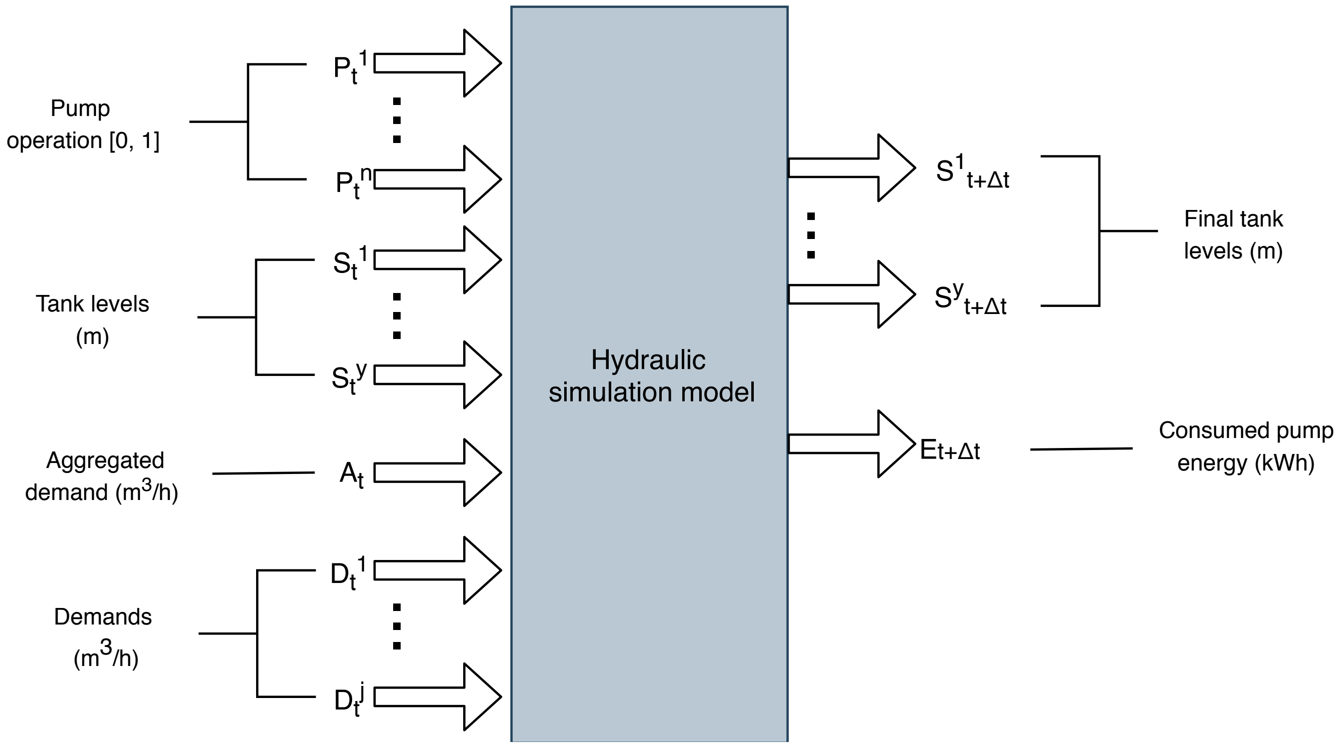


Figure 3.3: Representation of the input - output variables of the model

24 h scenarios were generated, resulting in 24000 samples. The time that this process took varied widely depending on distinct dataset, explained in the next Chapter, that were created.

3.2.2 Model creation and training

Since the solution consists of a simple ANN, the Scikit-learn library, previously mentioned in Section 3.1, offers an easy to use implementation along with a big variety of tools to optimize and analyse the performance of the models.

Before starting the training procedure, the data is normalized and split into 3 sets: train, test and validation. The training set is composed of 80% of the total data, 19200 inputs, the test set has 20% minus 24 samples. Those remaining 24 samples that compose the validation set. The starting indices of each split are randomly permutated.

CHAPTER 4

Results and discussion

The chapter is divided in two sections. The first demonstrates the results of the water demand forecasting methods on a 24 h scenario along with evaluation scores. The second section is sub-divided in two case studies, one for each network that is used to test the simulation model. The networks are presented along with the results of several analysis that were performed with the purpose of studying the capacities of the network simulation method.

4.1 Water demand forecasting

ANNs already show that they can obtain good results in water demand forecasting. However, some methods, explained in Section 3.1, seem to also be capable of precisely predicting water demands and may outperform ANNs. Therefore, with the intention of identifying those methods, the performance of ANNs is used as a benchmark to which the results of other models are compared to. The benchmark ANN uses the hyperparameters described in Section 3.1.3. A 24 h scenario from one of the datasets, demonstrated in Section 3.1.2, was used as a reference in which the models are tested. The structure and hyperparameters of models were optimized for that same dataset and their optimization results are demonstrated. The selection of the set of optimal architectural and training hyperparameters for the simulation model was performed using a grid search method which exhaustively tests combinations of parameters to determine which obtains the best performance. Several searches were performed, the first of which tested a wide range of values for the parameters that became more specific with each search as the values that produced the best score stood out.

4.1.1 Artificial Neural Network implementations

Table 4.1 shows the optimization results for both ANN implementations, the Keras implementation and the benchmark ANN, in this section referred to as the scikit-learn implementation. There are some differences in some parameters, mainly on the activation function, optimizer, dropout, etc. The Keras implementation uses ReLU activation function in all layer, except the last where the identity function is used instead. Keras also does not support the LBFGS

	ANN - keras	ANN - scikit-learn (benchmark)
Number of hidden layers	2	
Number of nodes	50, 25	8, 25
Learning rate	adaptive	
Initial learning rate	0.001	
Batch size	20	200
Activation	ReLU, identity (last layer)	identity
Optimizer	Adam	LBFGS
Dropout	9%	n/a

Table 4.1: Parameter optimisation results for the keras and scikit-learn models.

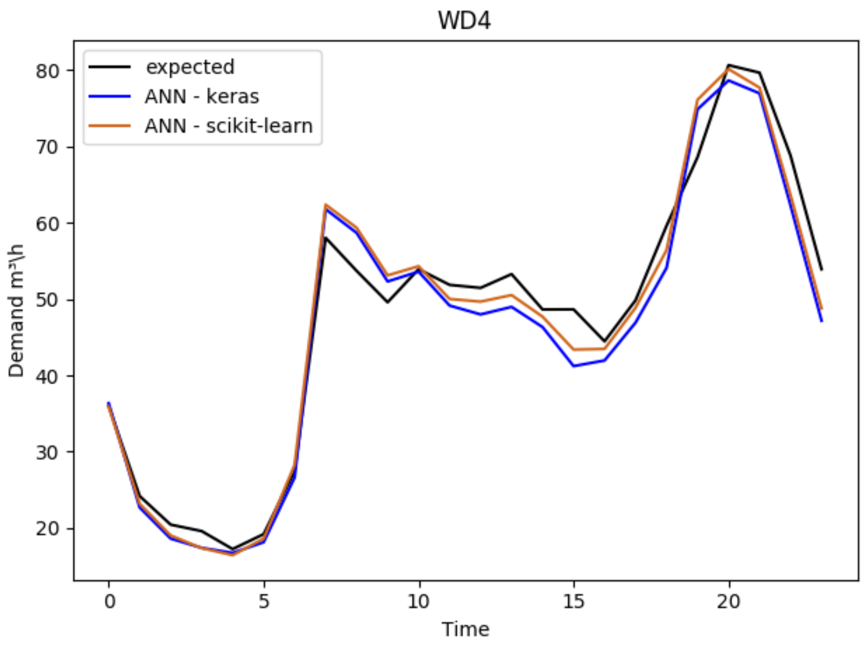


Figure 4.1: Forecasting results for a 24 h scenario of the WD4 dataset.

	RMSE	MAE	R^2
ANN - keras	3,7703	3,1725	0,9562
ANN - scikit-learn	3,1434	2,454639	0,9690

Table 4.2: Evaluation results of ANNs implementations for a 24 h scenario of the WD4 dataset.

optimizer, so Adam was used instead. Approximately 9% of dropout on each layer seems an adequate amount that does not harm the performance. Dropout is not available in the scikit-learn library. Both models are trained with data that is arranged as described in 3.1.3, where n, the total number of features is 50.

Figure 4.1 shows the forecasting results for the 24 h scenario, both models can cope with the demand variations at any hour of the day. For that same scenario, the results of the evaluation metrics, RMSE, MAE and R^2, of each implementation are listed on Table 4.2. Despite the results being very similar, the benchmark implementation has slightly better performance in every metric.

	RMSE	**MAE**	R^2
ANN - keras	4,2594	3,3088	0,9088
ANN - scikit-learn	4,3934	3,4198	0,8874

Table 4.3: Evaluation metric average results for five 24 h scenarios of different datasets.

The advantage of using dropout is to be able to improve the model's generalization. In order to test this, the models were tested with 10 other scenarios of the WD4 dataset. The results of this experiment can be consulted in Table 4.3. In this context, the performance of the keras implementation has a modest edge over the benchmark.

4.1.2 Extreme Learning Machines data arrangements

The ELM models, each with the data arrangements mentioned in 3.1.3 and 3.1.4, were also optimized. The number of total features for the models that use the *pd* and *dd* data arrangements are 48 and 50, respectively. The parameter optimization results can be consulted of Table 4.4. The optimization procedure yield the same results for both models. A hundred nodes for the only hidden layer, using the multiquadric activation function. The parameter alpha represents regularization term, also known as penalty, its purpose is to prevent overfitting by restraining the size of the weights. The results show that the optimum value for alpha is 0.5.

Once again, the performance of the models with the distinct data arrangements was compared with the benchmark ANN. Figure 4.2 shows a 24 h scenario of water demands and their respective forecasted values for each model. All models demonstrate reasonable performance with minor variations from the expected value. Table 4.5 shows that the ELM models that uses the *pd* data arrangement has a slightly better RMSE and R^2 results compared to those of the benchmark.

What ELMs lack in the capability to significantly improve the results of the benchmark ANN, they gain in computational efficiency during their training procedure. Table 4.6 shows the average of 10 training and prediction times for each model. The results corroborate what is stated in the literature, ELMs are far more efficient than ANNs.

	ELM - pd	**ELM - dd**
Number of hidden nodes	100	
Activation	multiquadric	
Alpha	0.5	

Table 4.4: Parameter optimisation results for the ELM models with different data arrangements.

	RMSE	**MAE**	R^2
ANN	3,5988	2,7538	0,96014
ELM - pd	3,4170	2,8825	0,96407
ELM - dd	4,0006	2,8708	0,95075

Table 4.5: Evaluation metric results of the ELMs models with distinct data arrangements for a 24 h scenario of the WD4 dataset.

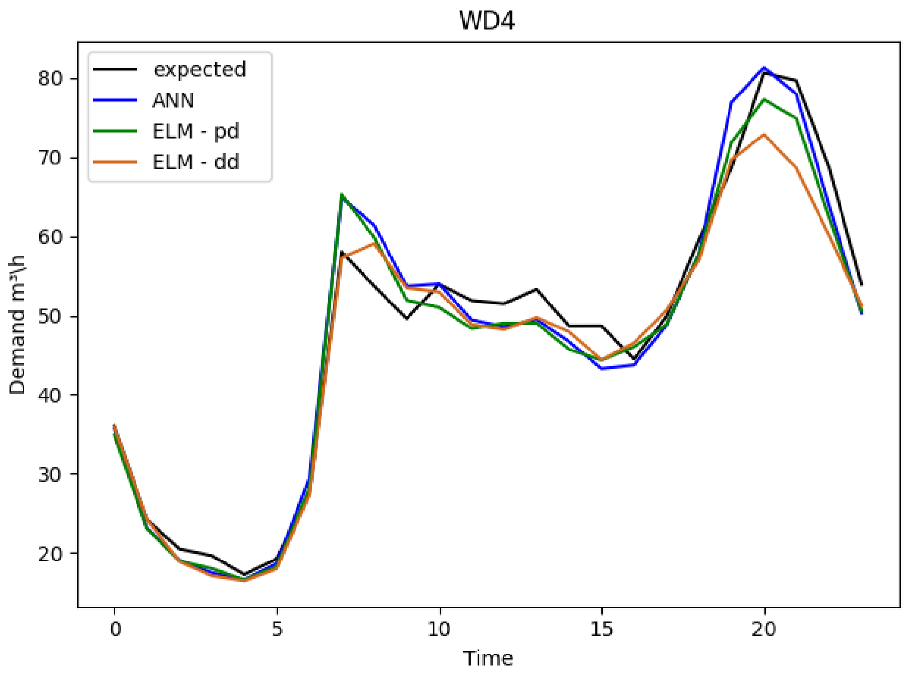


Figure 4.2: Forecasting results of the ELM models with different data arrangements for a 24 h scenario of the WD4 dataset.

	ANN	ELM - pd	ELM - dd
Average training time (seconds)	1,142	0,128	0,202

Table 4.6: Average time of 10 training procedures.

4.1.3 Gated Recurrent Networks and Bidirectional Gated Recurrent Networks

The first set of GRUNs and BiGRUNs that were tested used the first data arrangement described in 3.1.7 and 3.1.8, respectively. As such, the number of features that were used to calculate a water demand can also be optimized. In other words, the optimum values for i and j must be found. Table 4.7 shows that both models achieve their best performance when $i = 3$ and $j = 2$. Due to the complexity of the models that are being considered in this section, in comparison to those of previous sections, there are a lot more parameters that can be adjusted. Parameters such the output space of the GRU layers that are referred to according the respectively sub-input vector of inputs that are assigned to them. The activation function used on the recurrent units was the identity and the candidate activation used was the ReLU. The optimal number of dense layers is 3, each with 20, 10 and 1 nodes. The first dense layer uses the ReLU activation function and the layer that follows uses the identity. The models were optimized with the Adam algorithm with an adaptive learning rate that start at 0.002, using batches with a size of 20 samples, during 25 epochs.

Figure 4.3 and Table 4.8 show the performance of the models in comparison to the benchmark ANN in the form of evaluation metrics and forecasted values for a scenario of 24 h. The models' performance resembles that of the benchmark ANN but is somewhat worst, and due to the complexity of the models, comes with a great computational cost. The stability of

	GRUN	BiGRUN
Data arrangement (i and j)	3,2	
Number of GRUs' output nodes (distant, near, recent)	16, 48, 32	
GRUs' activation and recurrent activation	identity, ReLU	
Number of dense layers	3	
Number of dense layer nodes	20, 10, 1	
Dense Layer activation	ReLU, identity (last layer)	
Optimizer	Adam	
Learning rate	"adaptive"	
Initial learning rate	0.0017	
Batch size	20	
Epochs	25	

Table 4.7: Parameter optimization results for the GRUN and BiGRUN models.

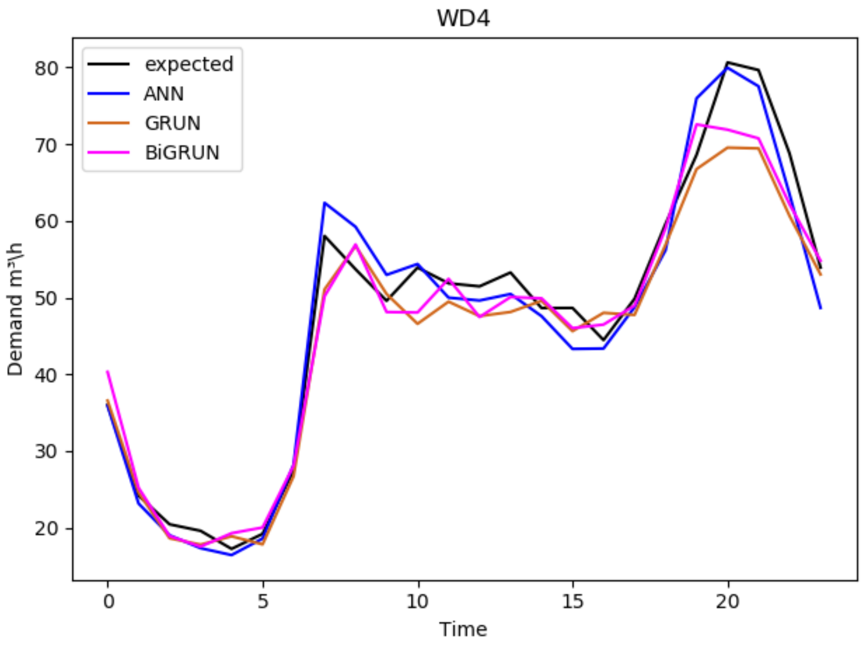


Figure 4.3: Forecasting results of the GRUN and BiGRUN models for a 24 h scenario of the WD4 dataset.

	RMSE	MAE	R^2
ANN	3,2067	2,5068	0,9683
GRUN	4,5813	3,4370	0,9354
BiGRUN	4,0643	3,1289	0,9491

Table 4.8: Evaluation metric results of the GRUN and BiGRUN models for a 24 h scenario of the WD4 dataset.

	GRUN - *dd*	**BiGRUN - *dd***
Number of features	50	
Number of GRUs' output nodes (distant, near, recent)	7, 20,20	7, 10, 20
GRUs' activation and recurrent activation	ReLU, identity	
Number of dense layers	3	
Number of dense layer nodes	12, 6, 1	8, 4, 1
Dense Layer activation	ReLU, identity	
Learning rate	"adaptive"	
Initial learning rate	0.001	0.002
Batch size	50	100
Optimizer	Adam	
Epochs	100	

Table 4.9: Parameter optimisation results for the GRUN and BiGRUN models that use the *dd* data arrangement.

	RMSE	**MAE**	R^2
ANN	3,1737	2,4735	0,9690
GRUN - *dd*	5,6707	3,8223	0,9010
BiGRUN- *dd*	5,0326	3,5376	0,9220

Table 4.10: Evaluation metric results of the GRUN and BiGRUN models with the *dd* data arrangement for a 24 h scenario of the WD4 dataset.

the models is also questionable, one training procedure might yield good results but, on rare occasions, one can result poor model performance.

Table 4.9 shows the parameter optimization results of models with *dd* data arrangement. The number of features used by the models with the *dd* data arrangement was also optimized. The models were able to achieve their best performance with 50 features, $n = 50$. In the case of these models, the number of output nodes of the GRU layers differentiates from one another, the layer of the GRUN and BiGRUN that takes the near sub-vector as input has 20 and 10 output nodes respectively. In the GRUN model, the dense layers start with 12 nodes, a middle layer with 6 nodes, and another with 1 , it is trained with an initial learning rate of 0.001 using batches with a size of 50 samples. As for the BiGRUN its dense layer structure is 8, 4 and 1, its training starts with a learning rate of 0.002 and uses batches of 100 samples.

Figure 4.4 and Table 4.10 show the results of these models are noticeably worst. In addition, the models have become more unstable as their performance varies mildly from one training procedure to another.

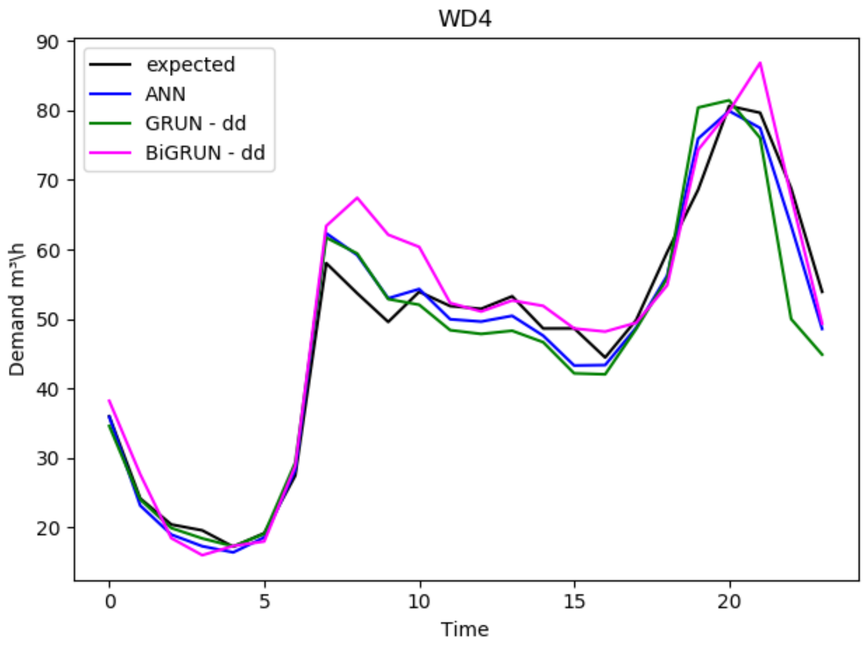


Figure 4.4: Forecasting results of the GRUN and BiGRUN models using the *dd* data arrangement for a 24 h scenario of the WD4 dataset.

4.1.4 Conclusion

The benchmark ANN set a high standard for the models to follow. Even so, some performance improvements were possible. The ANN implementation with Keras achieved very similar performance to that of the benchmark ANN and was able to have slightly better results when tested on multiple scenarios due to its generalization capabilities. This can be useful depending on the contexts were the water demand pattern is more unpredictable.

ELMs were the only models that truly rivalled the performance of the benchmark ANN. The model that used the *pd* was even able to obtain better results in some evaluation metrics. It would be interesting to further study this model and determine if further improvements are possible. In terms of computational efficiency during the training procedure the ELM models have a clear advantage over ANNs. This might be convenient in a system that forecasts water demands of a network with many consumption points and uses a regularly trained model for each point. An important aspect that is worth point out is that the *pd* data arrangements is limited in terms of the number of values that it can predict ahead of the current time. In other words, in a context were forecasts are made in a 24 h operational window, which is common on systems that try to optimize the pumping schedules, the model using de *pd* data arrangement cannot forecast demands for the whole day. This happens because it needs the last demand that took place right before the demand that is going to be forecasted. Meaning that, to forecast a 24 h scenario at time t, $[D_{t+1}, D_{t+2}, ..., D_{t+24}]$ the model requires $[D_t, D_{t+1}, ..., D_{t+23}]$. A possible solution to this problem is to use the value that was forecasted by the model, $\widetilde{D}_{t+1}$, as an input to forecast D_{t+2}. This solution is not ideal since each forecasted value carries an error that is going to be accumulated along the forecasting scenario.

The GRUN and BiGRUN models, despite their complexity, in comparison with ANNs, were not able to improve the results achieved by the benchmark with both data arrangements that were used considered. They require far more computational effort to train, and furthermore, their stability is questionable. Just like the previous ELM model that used the *pd* data arrangement, the first GRUN and BiGRUN models that were tested are also affected by the same problem of not being able to forecast 24 h without proper data.

4.2 Water network simulation

With the purpose of thoroughly test and analyse the simulation model, two water distribution networks were taken into consideration. These networks vary in size, number of elements and complexity. Thus, the following sections present two case studies containing an analysis of the networks. It is important to know the complexity of the networks to understand what kind of challenges the simulation model must face, as such, the case studies present a brief description of the networks and their elements. The hydraulic behaviour of each network was modelled with different techniques and its meta-model (simulation model) has different number of inputs/outputs. With that in mind, it is explained the amount of data used to represent the behaviour of the network and how was generated, along with the architecture and the parameters used to optimize the simulation model. Companies that manage water supply system usually optimize the pumping operations on a daily basis. So, it is important to know the performance of the model on a 24 h time window. The case studies present the results of the models simulating a 24 h scenario with a timestep of 1 h. It is common for the companies to receive faulty data due to some sensor malfunction and some error on the data transfer. The sensors themselves operate on an error margin. Some outliers are easily handled, but even so, noisy data remains a problem. Hence, an analysis on how the models behave with noisy data is conducted. Since the process to add noise to the data relies on a random generator with a uniform distribution, the results of the metrics are an average of 10 runs. It has been shown that adding artificial noise in small amounts to the training set can improve the models generalization and its tolerance to noise itself. This process is called jittering [51] and its effects on the models results tested with noisy data are presented in the next case studies. In addition to the noise, sometimes the amount of data that is available from the WSSs is scarce due to companies that just recently adopted an infrastructure capable of monitoring the water distribution network. The case studies also show an examination of the performance of the simulation model in function to the amount of training data. To avoid the model catching a favourable scenario where its performance is not affected by the varying amount of training data, the result metrics of this analysis are also an average of 10 runs.

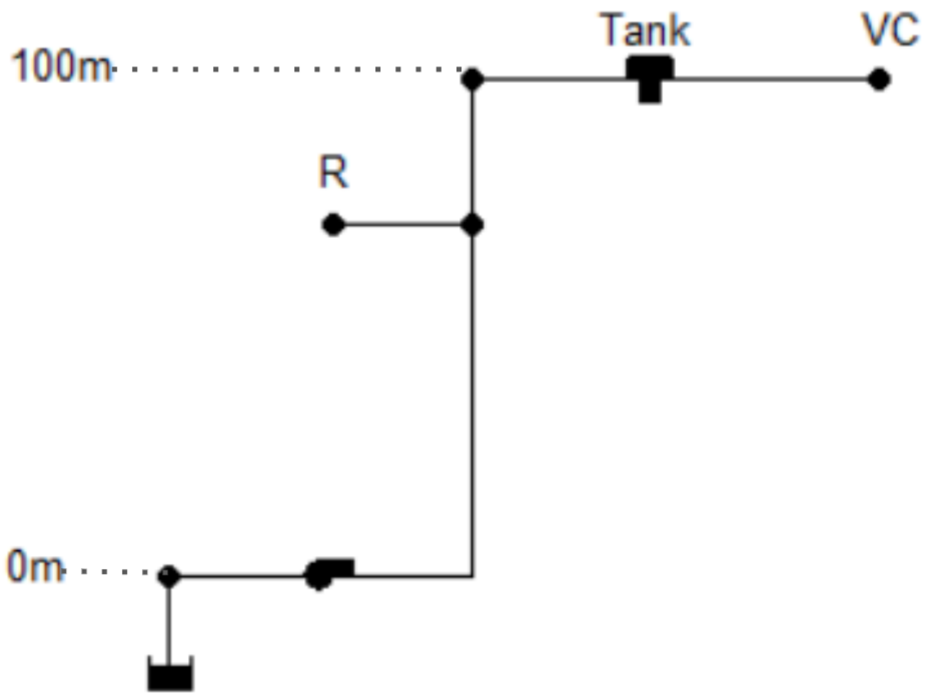


Figure 4.5: Schematical representation of Fontinha's network.

4.2.1 Fontinha

This water distribution network is part of a larger network located is central Portugal. Figure 4.5 shows a representation of this network. It is composed of a single pump, which is assumed to be at a height of 0 m, pumping water to a consumption point designated as R, and to a tank located at a relative height of 100 m. The tank supports a maximum water level of 7 m. Attached to this tank, there is another consumption point, VC. The water consumption point VC is only supplied by the tank while the consumption point R can be supplied by the tank or directly from the pump. Water discharges from the tank take place by gravity, therefore no energy is spent. Despite having a simple structure, Fontinha network is more complex than appears to be. It is modelled in a way that the water that enters the tank does it so from bellow instead of being poured from the top. This implies that the pump's power depends on the height of the water in the storage tank.

Data generation

As mentioned in 3.2.1, it is necessary, in order to train, test and validate the simulation model, to generate a dataset of input-output pairs. On the specific case of Fontinha network the model that was developed has as input:

- P_t - the status of the only pump on the network,
- S_t - the level of the storage tank,

- $[D_t^1,\ D_t^2]$ - the water demands of the two water consumption points,
- A_t - the daily aggregated demand.

And outputs:

- $S_{t+\Delta t}$ - the storage tank level at the end of the time step,
- $E_{t+\Delta t}$ - the energy that was spent.

These inputs/outputs are explained in more detail in 3.2.1. In total, the Fontinha's network model has an input-output space of 5-2. The data was generated using a hydraulic model implemented with the sole purpose of simulating this network.

ANN design and optimisation

The optimisation was performed using the same method that was described in Section 4.1, grid search.

Table 4.11 shows the Fontinha's model parameter optimization results. A single hidden layer with 15 neurons using the ReLU activation function. The learning rate parameter is set to "constant" with a value of 0.002. The batch size is set to 50. The training optimisation function selected was Adam.

Figure 4.6 shows the normalised validation scores for different parameters of the model of Fontinha's network. The mean absolute error is used as a metric which means that the lower the score the better. It is possible to observe that the resulting parameters from the grid search produce indeed the best scores as they stand out with the lowest error score.

Validation

After the optimization and training, the ANN can be validated. A 24 h scenario is chosen from the sample dataset to examine the performance of the model. Figure 4.7 demonstrates the evolution of the model inputs during the day. Figure 4.7a represents the water level of the tank of the beginning of each time-step. Figure 4.7b, 4.7c and 4.7d show the water demands and the aggregated water demand along the 24 h scenario. Figure 4.7e shows the status of the pump at each time step.

Figure 4.8 demonstrates both the expected and the predicted energy and tank level values. The red and blue line represent the expected and simulated values, respectively. The inputs from Figure 4.7 are in correspondence with the outputs from Figure 4.8. These results demonstrate that the model was able to faithfully recreate the behaviour of the network. In addition, Table 4.12 shows the scores of the metrics RMSE, MAE and R^2 for the same 24 h scenario. The error metrics are similarly low and the R^2 is almost 1, this applies to both tank level and energy.

Number of hidden layers	Number of nodes	Batch size	Learning rate	Initial learning rate	Activation	Optimizer
1	15	50	"constant"	0.002	ReLU	Adam

Table 4.11: ANN best configuration for Fontinha's simulation model.

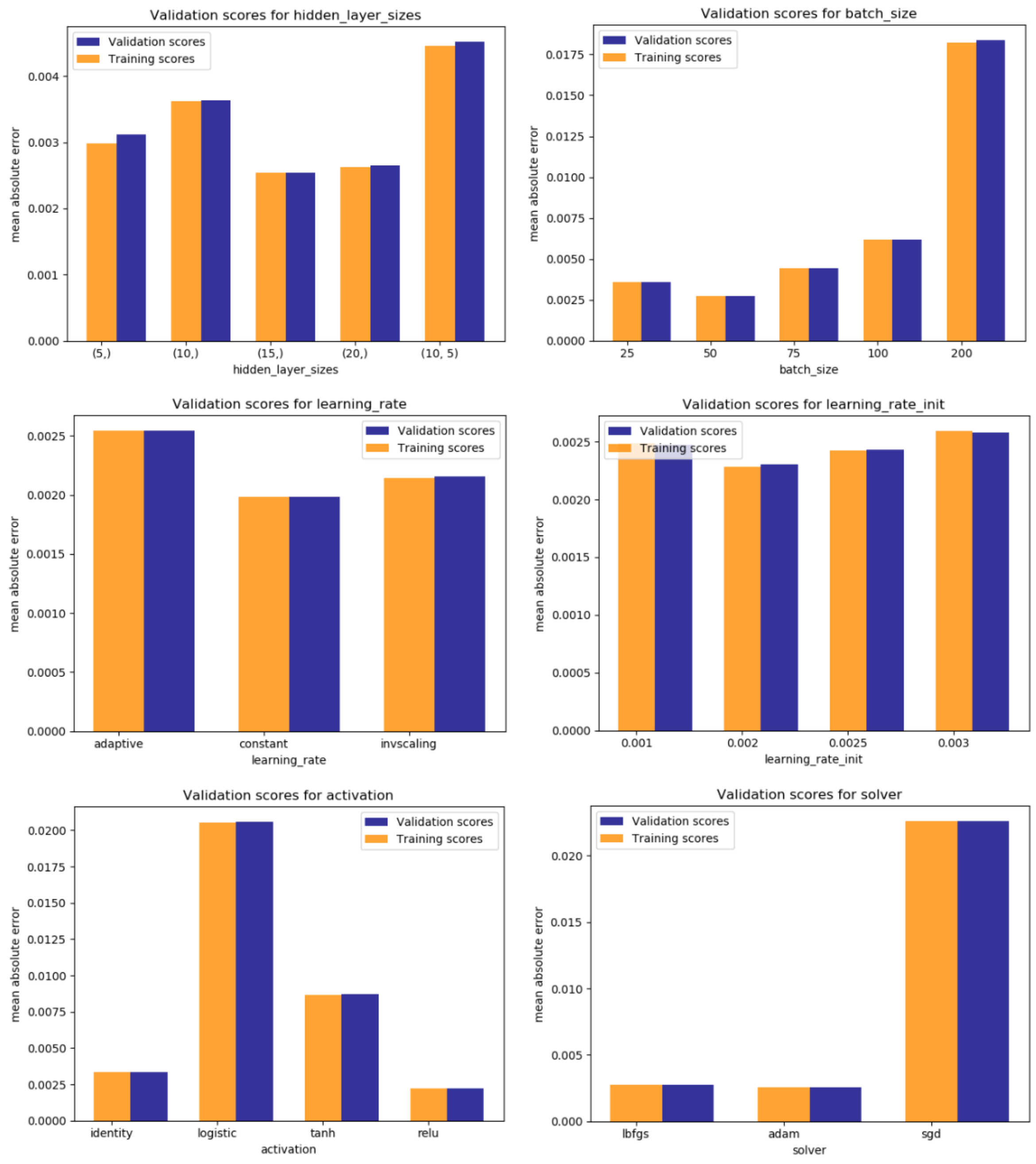
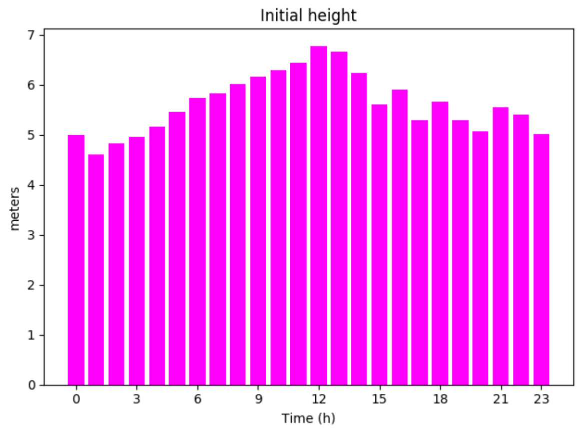
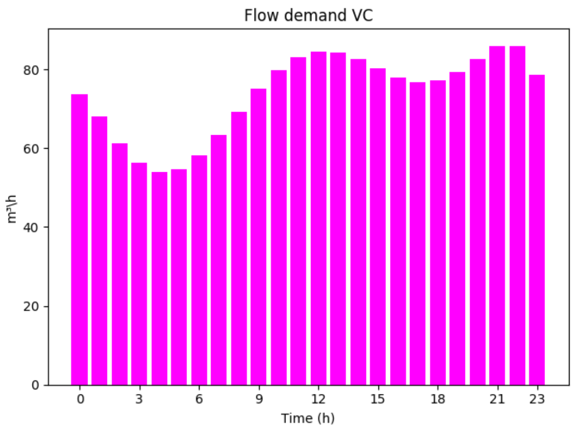


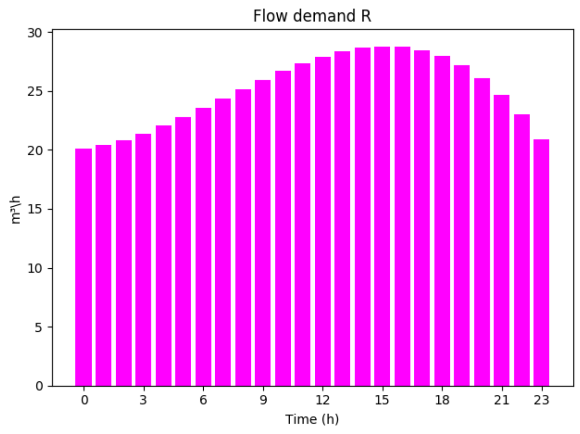
Figure 4.6: Validation scores for different model parameters using mean absolute error as scoring method on normalised values of Fontinha's network.



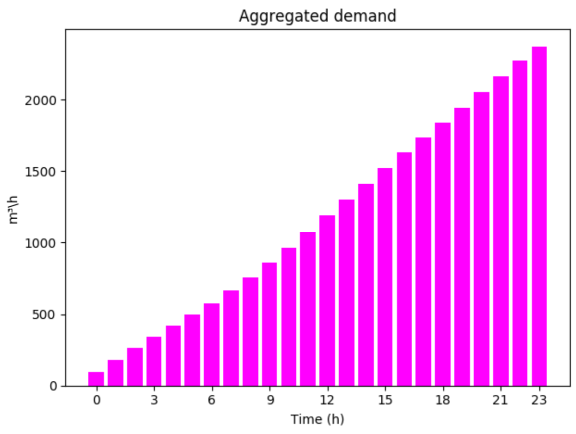
(a)



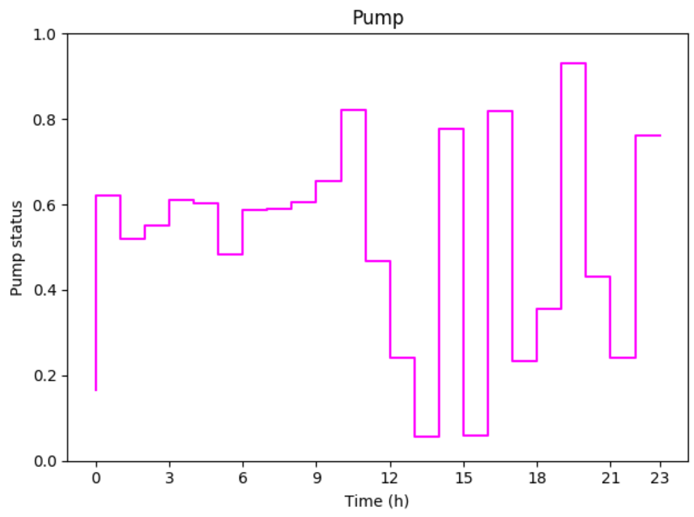
(b)



(c)

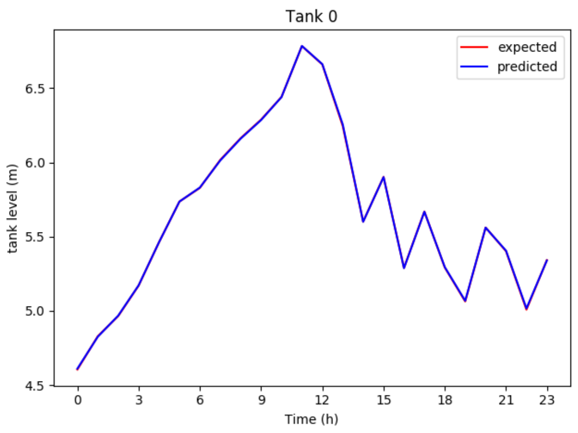


(d)

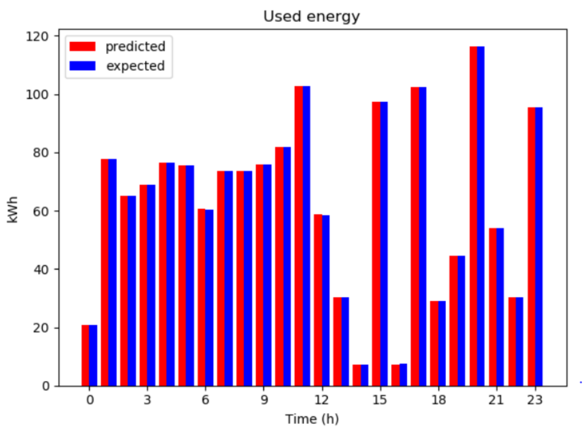


(e)

Figure 4.7: Fontinha's simulation model inputs on a 24 h scenario.



(a)



(b)

Figure 4.8: Fontinha's simulation model outputs on a 24 h scenario.

	RMSE	**MAE**	R^2
Tank level (m)	0,0036	0,0027	0,9999
Energy (kW)	0,0607	0,0570	0,9999

Table 4.12: Fontinha's simulation models RMSE, MAE and R^2 metric scores for the tank level and energy on a 24 h scenario.

Noise analysis

The metrics are calculated taking into consideration a 24 h scenario. Figure 4.9 and 4.10 demonstrate the evolution of the metrics RMSE, MAE, and R^2 as the noise increases on the tank level and energy, respectively. The values of both error metrics are in meters. Figure 4.9 shows that the simulation model produces reasonable results up until 20% of noise, where the MAE and RMSE are approximately 0.5 m. From that value forward the performance of the model starts to decrease very rapidly. As for the Figure 4.10 indicates that the performance deteriorates at 15% of noise. To put this into perspective, it is possible to assume that on a normal scenario the pump is active for approximately 10 h a day which, in the case of this pump that spends 75 kW an hour, means that at the end of the day used 750 kWh. Thus, an approximated 8 kW error corresponds to 1,06% of error.

Noise tends to decrease the performance of the model but in some cases it is possible to use noise to improve the generalization results. If the model is trained with data which contains a certain amount of noise, it may be possible to improve the model's capacity to deal with noisy data. An experiment was conducted where it was added up to 30% of noise in the training set (5760 samples). Figure 4.11 and 4.12 show the results of the experiment on the level of the tank and the energy spent, respectively. As expected, the quality of the results decreases as the percentage of noise increases on the testing set. However, in the case of the tank level, it's possible to observe performance improvements as the testing noise percentage increases with 5% to 10% of noise on the training set, (see Figure 4.11b and 4.11c). Figure 4.11d shows that these improvements are not noticeable when the percentage of training noise increases to 15%. It is important to notice that Figure 4.11c, despite having a higher noise percentage on the training set, shows better overall results than Figure 4.11b, and even beats the performance of the model when its training noise is 0% and the testing noise is 15%, shown in Figure 4.11a. This indicates that a noise percentage on the training set of 5% to 10% may improve the model's "resistance" to noisy data. The same cannot be said about the results of the spent energy. Figures 4.12a to 4.12d show that the results get progressively worst as the amount of training noise increases.

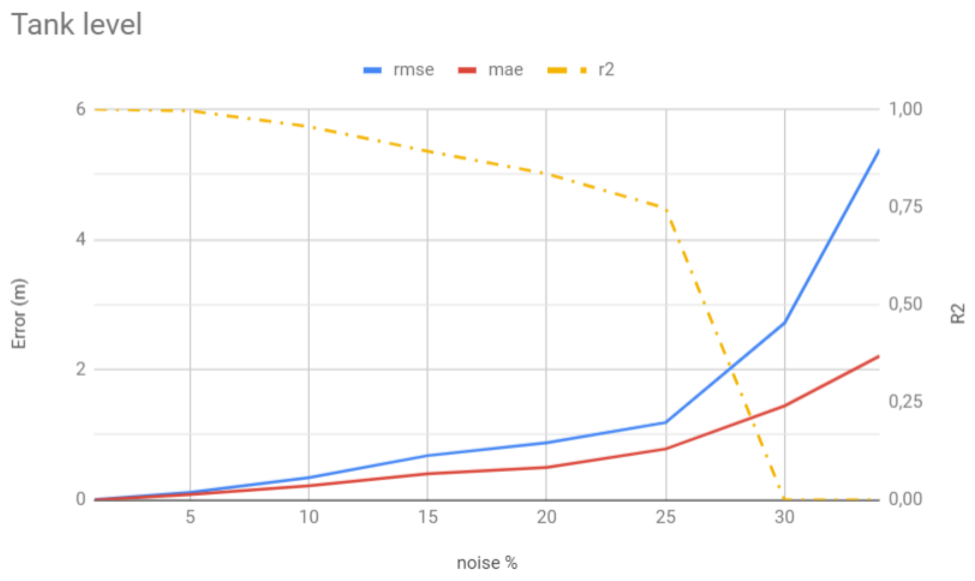


Figure 4.9: The RMSE, MAE and R^2 values for the tank level with different levels of noise for the Fontinha case study.

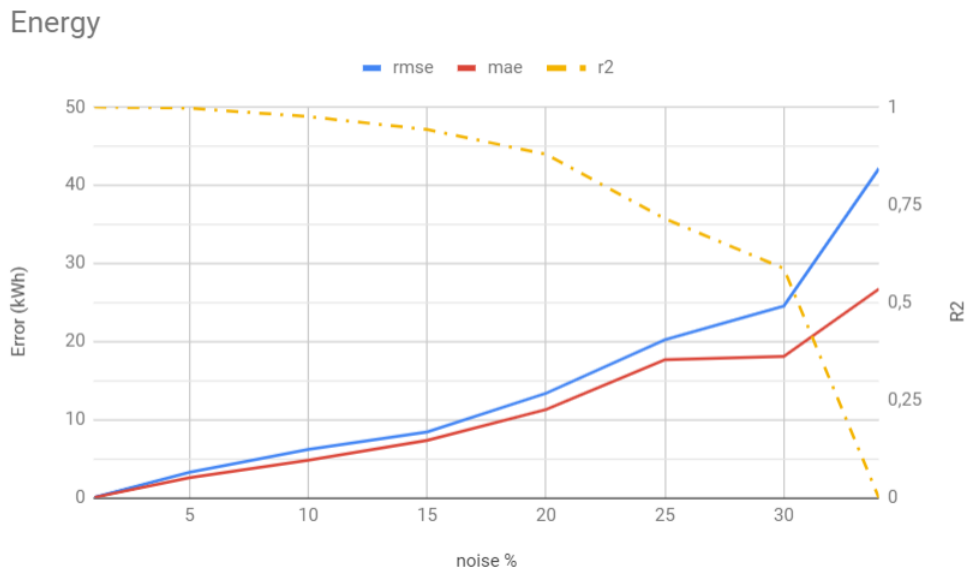
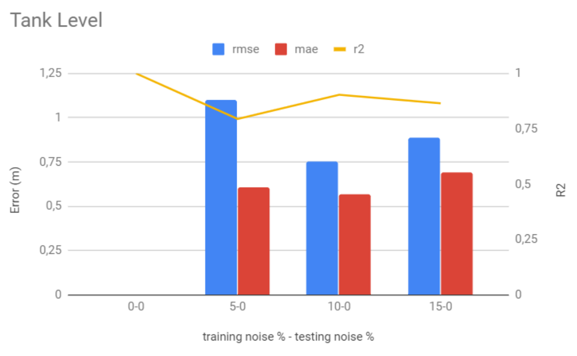
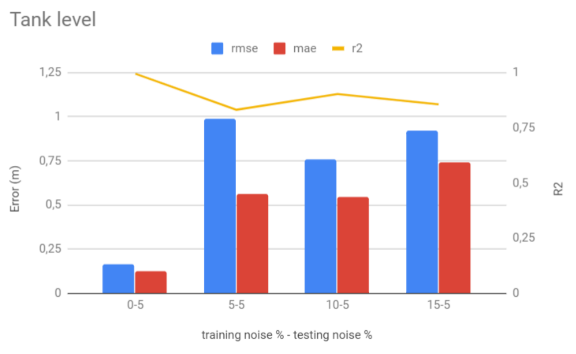


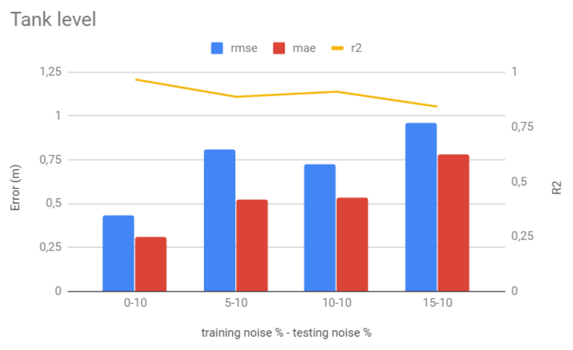
Figure 4.10: The RMSE, MAE and R^2 values for the used energy with different levels of noise for the Fontinha case study.



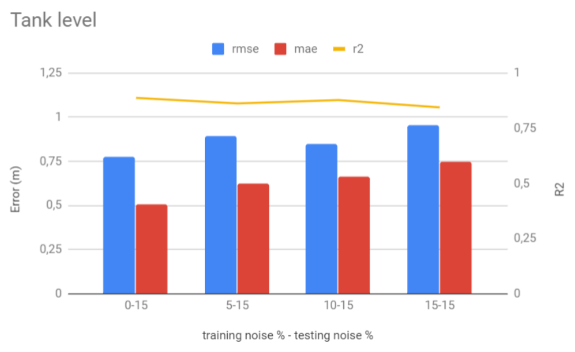
(a) 0% test noise



(b) 5% test noise

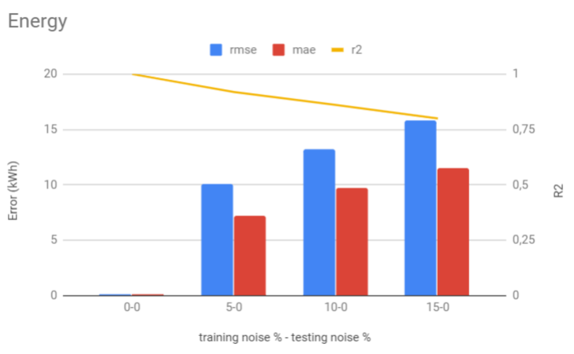


(c) 10% test noise

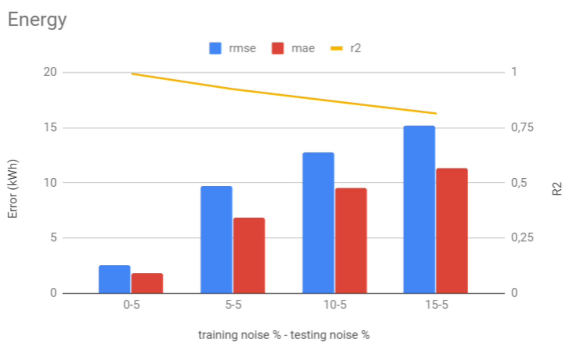


(d) 15% test noise

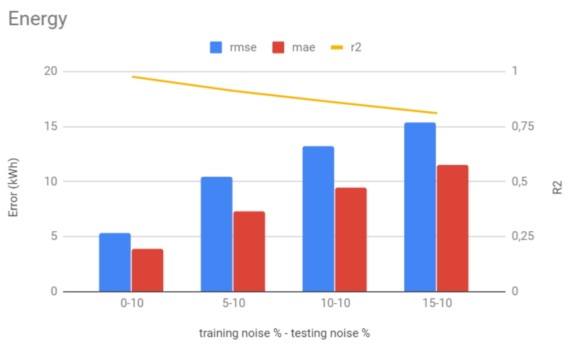
Figure 4.11: Fontinha's simulation model results of the RMSE, MAE and R^2 metrics on the tank level for several amounts of noise to both the training and testing set.



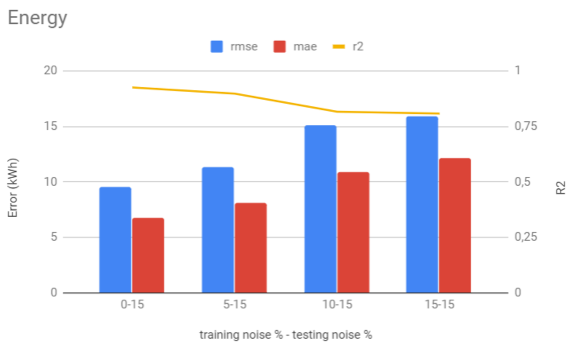
(a) 0% test noise



(b) 5% test noise



(c) 10% test noise



(d) 15% test noise

Figure 4.12: Fontinha's simulation model results of the RMSE, MAE and R^2 metrics on the spent energy for several amounts of noise to both the training and testing set.

Training data amount analysis

In similarity to the previous section, Figure 4.13 and 4.14 display the evolution of the RMSE, MAE and R^2 for different amounts of training data samples on a 24 h scenarios. The minimum amount of training samples is 24. Despite Figure 4.13 and 4.14 appearing to have a big increase on both error metrics, they are not significant for the worst scenario. With the minimum amount of data possible, the error was approximately 0.6 m and 5 kW, respectively, which, considering the scenario described in the previous section, corresponds to 0.67% of error on the case of the spent energy.

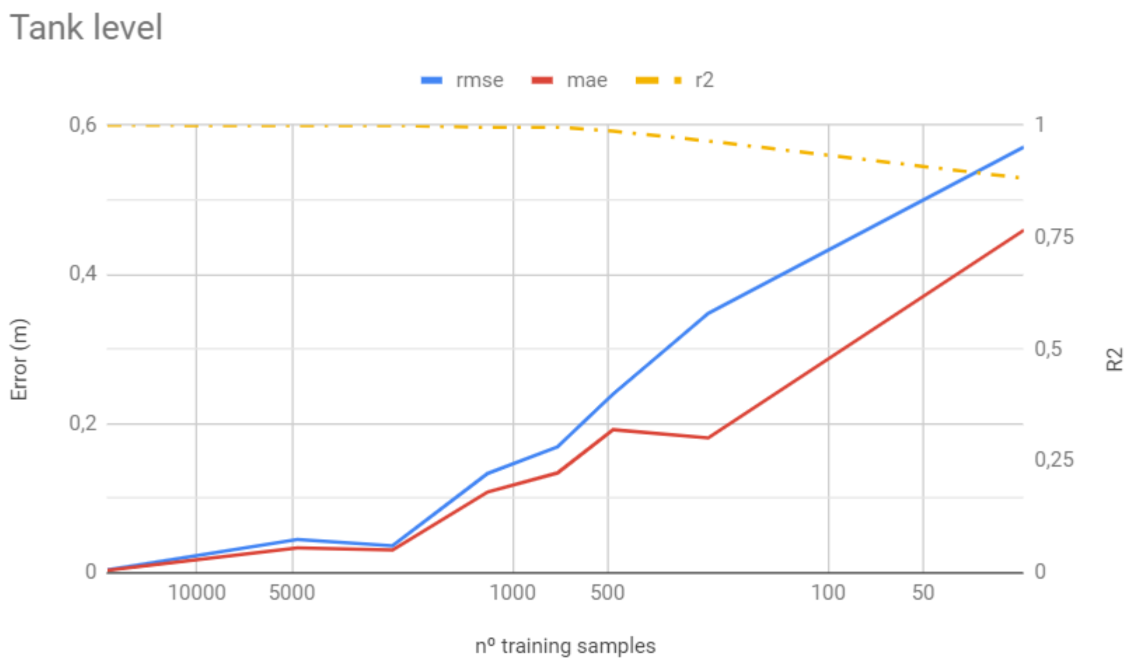


Figure 4.13: Fontinha's simulation models results of RMSE, MAE and R^2 as the amount of training data decreases on the level of the tank.

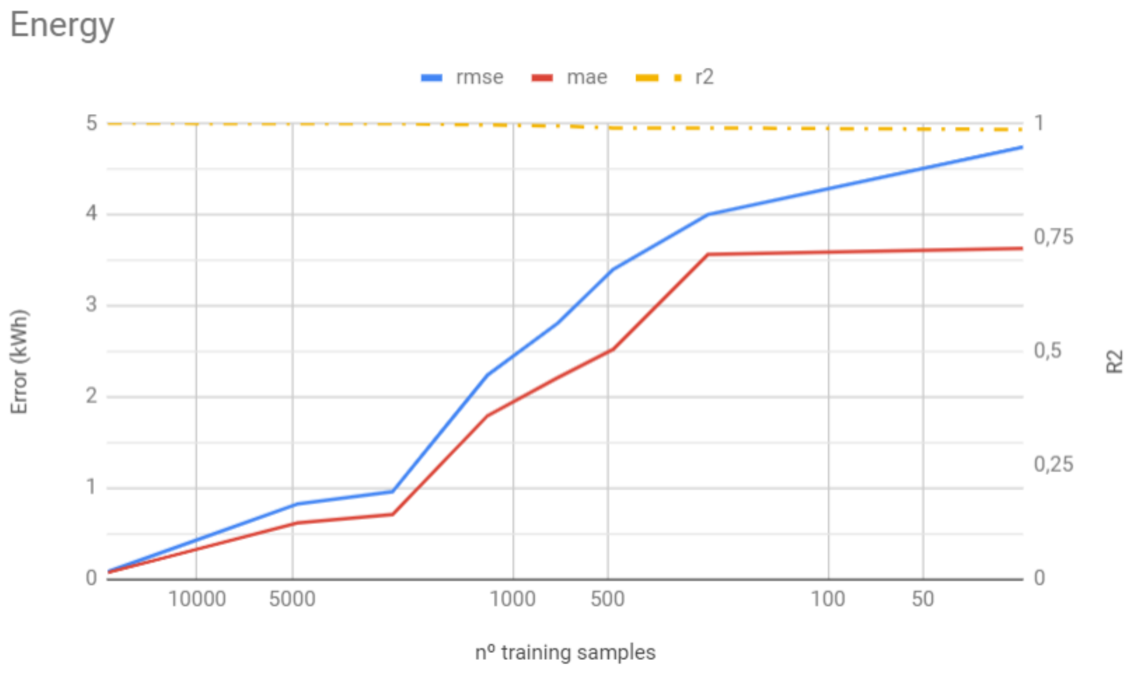


Figure 4.14: Fontinha's simulation models the results of RMSE, MAE and R^2 as the amount of training data decreases on the energy that is being spent.

Conclusion

Relatively to the networks that are analysed on other works, Fontinha is not a very complex network. So, a good performance from the simulation model was expected and was obtained. The simulation model was able to minutely mimic the water network.

The model, regarding the tank level output, was able to tolerate up to 20% of noise in the data. As for the used energy simulation, the model tolerated a maximum 15% of noise. Given these observations, it is possible to claim that Fontinha's simulation model is able to have reasonably acceptable results with 15% or less noise in the data. If the noise surpasses 15%, the simulation of the spent energy is no longer credible. The same happens in the simulation of the tank level at 20% of noise. It may be possible to improve the simulation model's tolerance to noise by also adding some noise, up to 10%, to the training data. This comes at the expense of the reliability of the energy output, as demonstrated that adding any amount of training noise only worsens the energy simulation.

Regarding the amount of training data the results from the analysis show that the model is capable of a good performance even when the amount of training data is very low.

4.2.2 Richmond

The Richmond network represents a description of the Richmond water distribution system, owned by Yorkshire Water. It was used in a PhD research on operational optimization of water distribution systems by Jakobus van Zyl [52] and later by Coelho B. [40]. Figure 4.15 demonstrates a skeletonized representation of the network which is composed of 6 tanks located at a height ranging approximately between 187 and 260 meters, 7 pumps, 1 reservoir, 41 consumption points, and other components. The water is allowed to return towards the direction where it was pumped from. The Richmond is a well-documented and analysed network allowing a comparison of results with other studies, as well as a challenge of greater complexity for the simulation model.

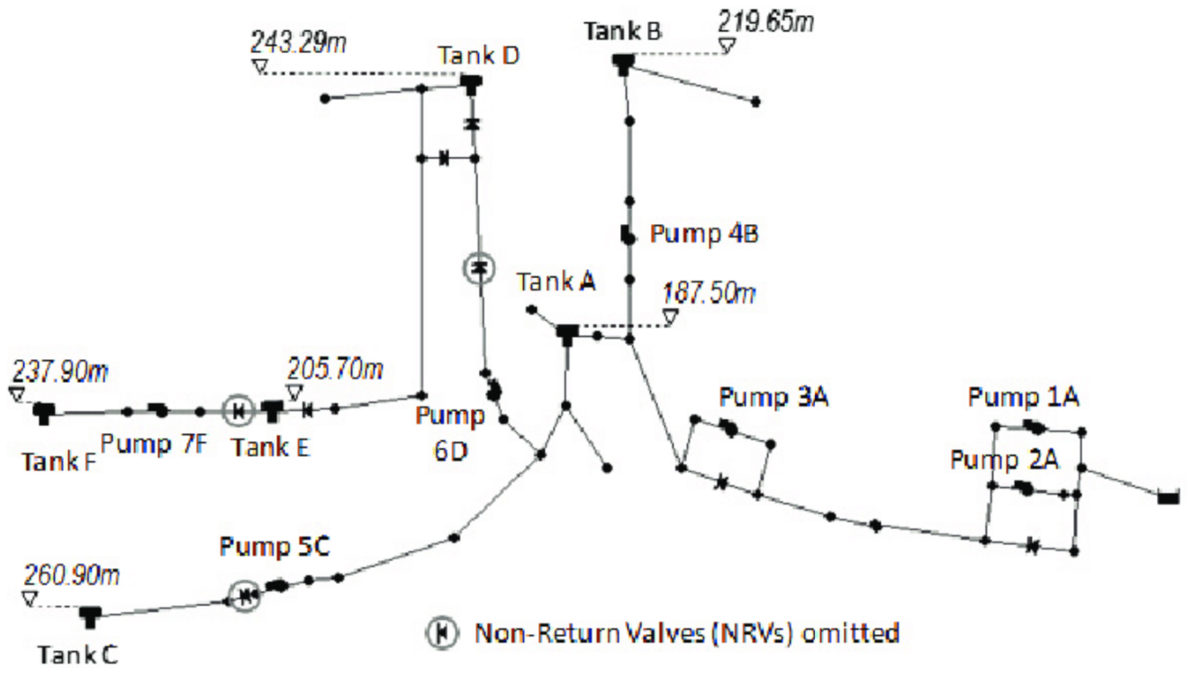


Figure 4.15: Schematic representation of Richmond's network. Adapted from [52].

Data generation

In this case study the model has the following inputs:

- $[P_t^1,\ P_t^2,\ ...,\ P_t^7]$ - the status of the 7 pumps,
- $[S_t^1,\ S_t^2,\ ...,\ S_t^6]$ - the 6 storage tank levels,
- $[D_t^1,\ D_t^2,\ ...,\ D_t^{41}]$ – 41 consumption points water demands,
- A_t - the aggregated water demand,

and outputs:

- $[S_{t+\Delta t}^1, S_{t+\Delta t}^2, ..., S_{t+\Delta t}^6]$ – the consequent level of the 6 storage tanks,
- $E_{t+\Delta t}$ – the energy spent.

More detailed explanation of these inputs/outputs is present in 3.2.1. The model has an input-output space of 55-7. Richmond's network has been extensively modelled using EPANET, some examples are publicly available for download [53].

ANN design and optimization

To select the set of optimal architectural and training hyperparameters, the same procedure presented in 4.2.1 was followed.

In similarity with the previous model, the best performance was achieved with a single hidden layer with the rectified linear unit as the activation function, along with the Adam optimization function and a "constant" learning rate. As for the batch size, in this case, is set to 100. The learning rate is 0.0025, and the hidden layer has 22 neurons. This discrepancy between the two case-study networks on the values of these parameters is expected due to the difference in size of the input-output space of the models. The previous parameters are listed in Table 4.13.

The validation scores of Richmond's network model are illustrated in Figure 4.16. Unlike the scores from the previous model, the scores for each values of each parameter of this model are very similar, which means that despite the value that is attributed to some parameters the score may not be significantly altered. Even so, a close observation of the results reveals that the parameters shown in Table 4.13 have slightly better scores than others.

Number of hidden layers	**Number of nodes**	**Batch size**	**Learning rate**	**Initial learning rate**	**Activation**	**Optimizer**
1	22	100	"constant"	0.0025	ReLU	Adam

Table 4.13: ANN design parameters for Richmond's simulation model.

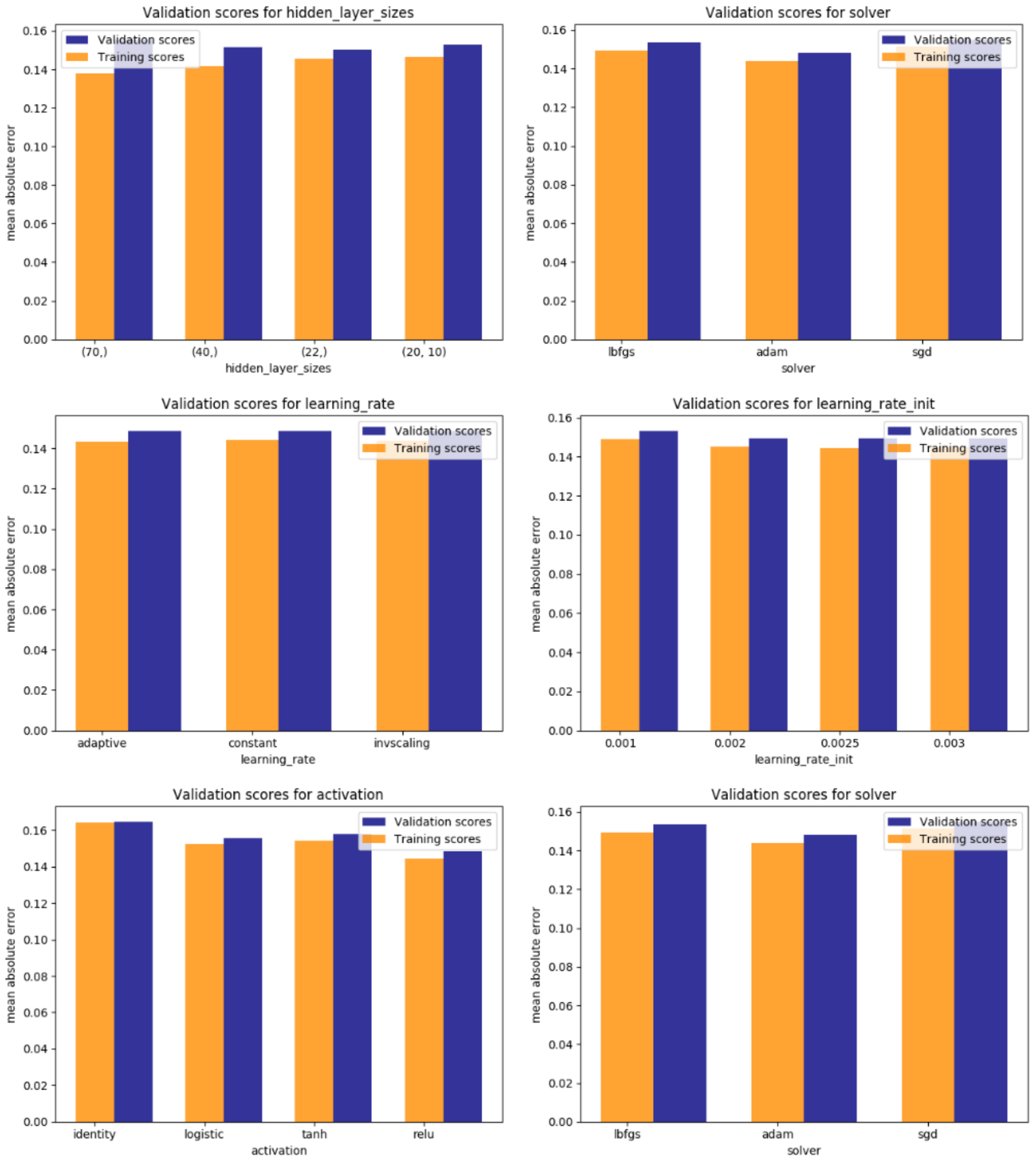


Figure 4.16: Validation scores for different model parameters using mean absolute error as scoring method on normalized values of Richmond's network.

Validation

Unlike the network of the previous section, Richmond has too many inputs to be depicted. As such, Figure 4.17 only displays a portion of the model's inputs for a 24 h scenario. Compared with Fontinha, the tanks on this network have a much lower water capacity, and, consequently, the variations on their levels are measured between centimetres, in some cases, and a few meters maximum. Figure 4.17 also shows a case where on the water demands appears to have a negative value, this means that, in the particular case of that consumption point, the water is flowing in instead of being drawn. Several situations like this one do not happen on Fontinha's network and it is important to assess whether or not the model is capable of dealing with them.

Figure 4.18 shows the resulting outputs from the input values of Figure 4.17. There is a noticeable difference between the expected values and the predicted values. However, the values differ by a maximum 13%, and only in the case of tank 5. Table 4.14 demonstrates the model evaluation metric scores for the same scenario. The scores are slightly worse than Fontinha's but still indicate a quite good model performance. The errors for both the tank level and the energy are expected to be of the order of centimetres and tenths of a kilowatts, respectively.

	RMSE	**MAE**	R^2
Tank level (m)	0,0554	0,0201	0,9820
Energy (kW)	0,6043	0,4701	0,9983

Table 4.14: Richmond's models RMSE, MAE and R^2 scores for the tank level and energy on a 24 h scenario.

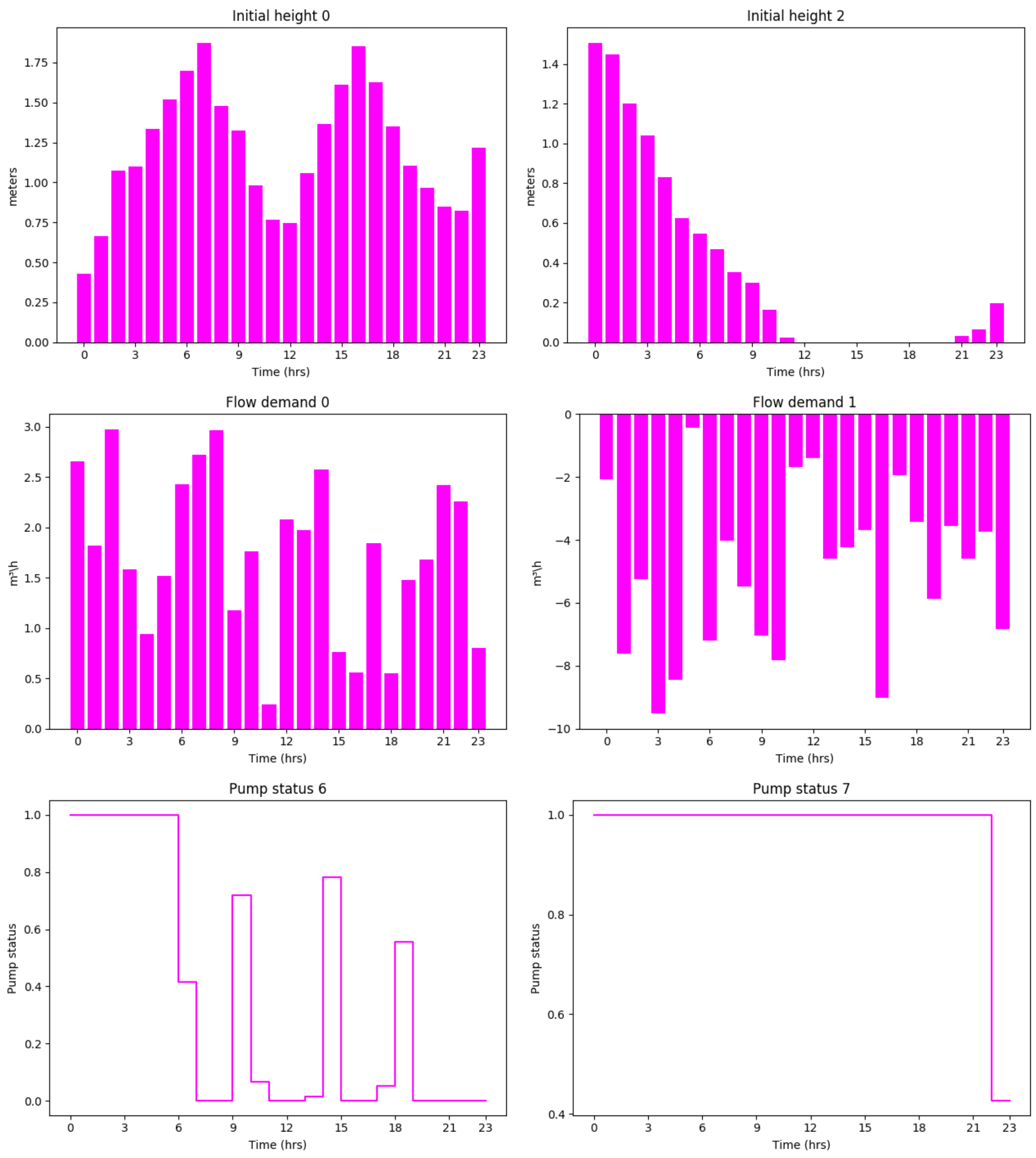
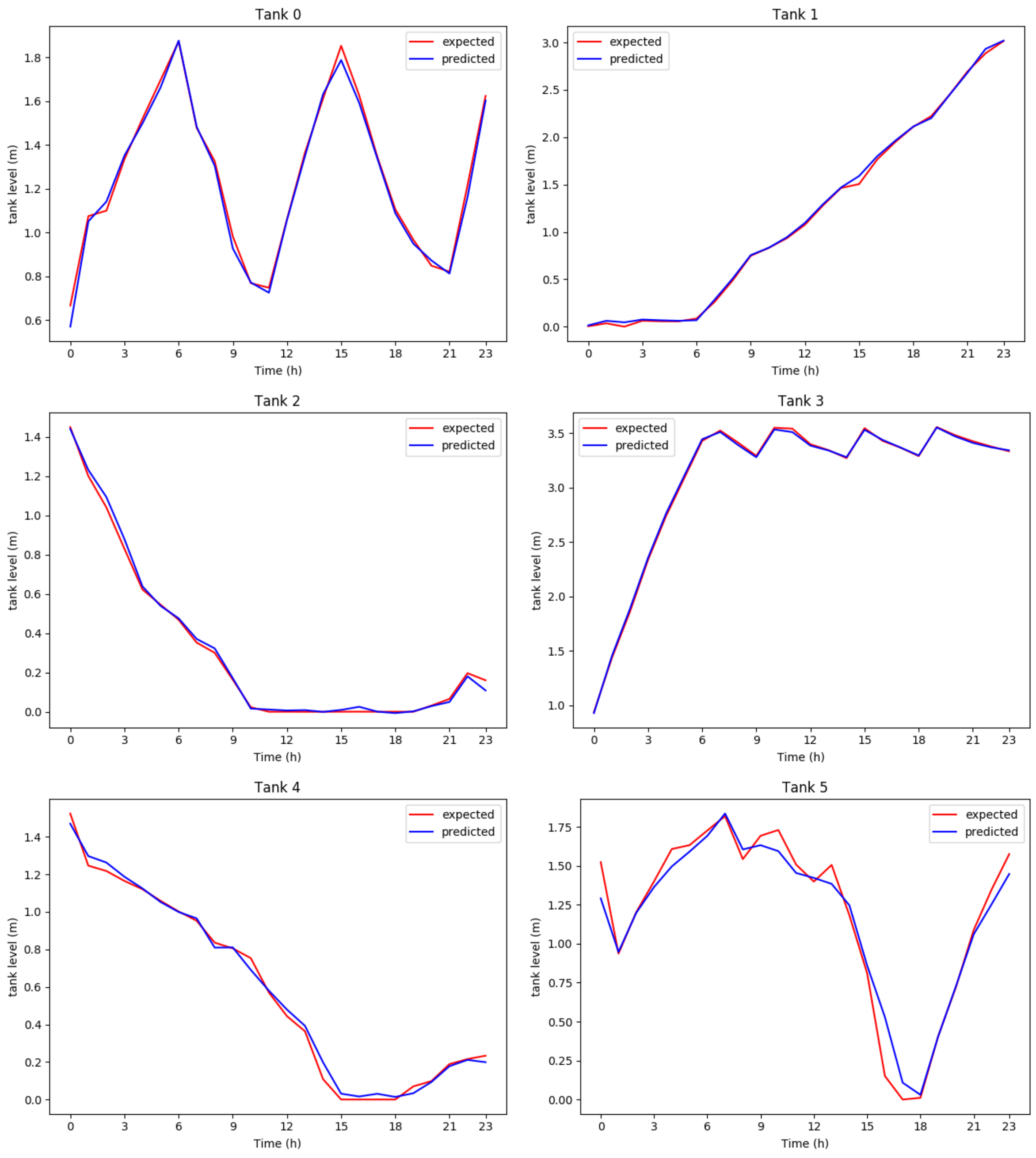


Figure 4.17: Portion of Richmond's simulation model inputs on a 24 h scenario.



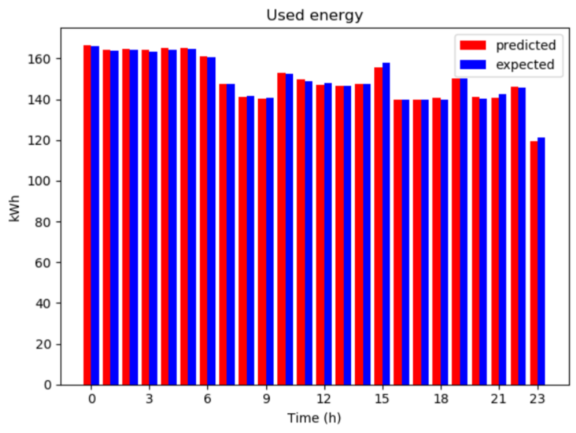


Figure 4.18: Richmond's simulation model outputs on a 24 h scenario.

Noise analysis

This analysis is similar to the one conducted in 4.2.1. Several levels of noise were tested. Figure 4.19 and 4.10 show the results as the level of noise increases. It is possible to observe that the simulation model shows reasonable results even with high levels of noise. In the case of the tank level, represented in Figure 4.19, with 40% of noise the MAE, RMSE and R^2 are approximately 10 cm, 14 cm, and 0.80, respectively. With a percentage higher than 40% the performance of the model steadily decreases but it does not plummet. As for the energy, Figure 4.20, the model can withstand high percentages of noise without a significant impact on its performance. With a maximum percentage of noise of 60%, the energy is predicted with and approximately MAE, RMSE and R^2 scores of 7 kW, 8 kW and 0.90, respectively.

Regarding the results from the analysis of different amounts of training noise, Figures 4.21 and 4.22 show the effects, on the simulation of the tank level and energy, respectively, of different amounts of training and testing noise on the metric scores. Like the previous analysis the noise was added to 30% of the training data. It is noticeable that both the tank level and energy predictions, when facing noise on the testing data, can be improved by adding noise to the training data. Compared to the other testing scenarios related with the tank levels, 50% of training noise has the best results for any amount of noise on the testing data, Figure 4.21d. As for the energy prediction, the best results are achieved with 40% of training noise, Figure 4.22c.

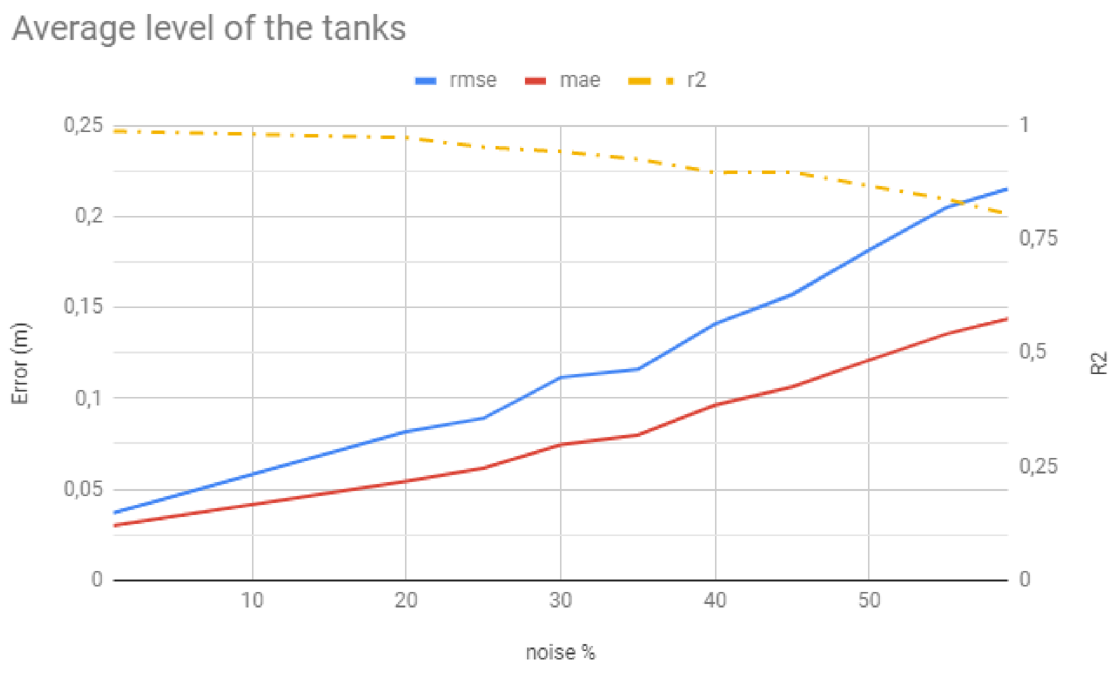


Figure 4.19: Richmond's of the results of RMSE, MAE and R^2 as the level of noise in the data increases on the average level of the tanks.

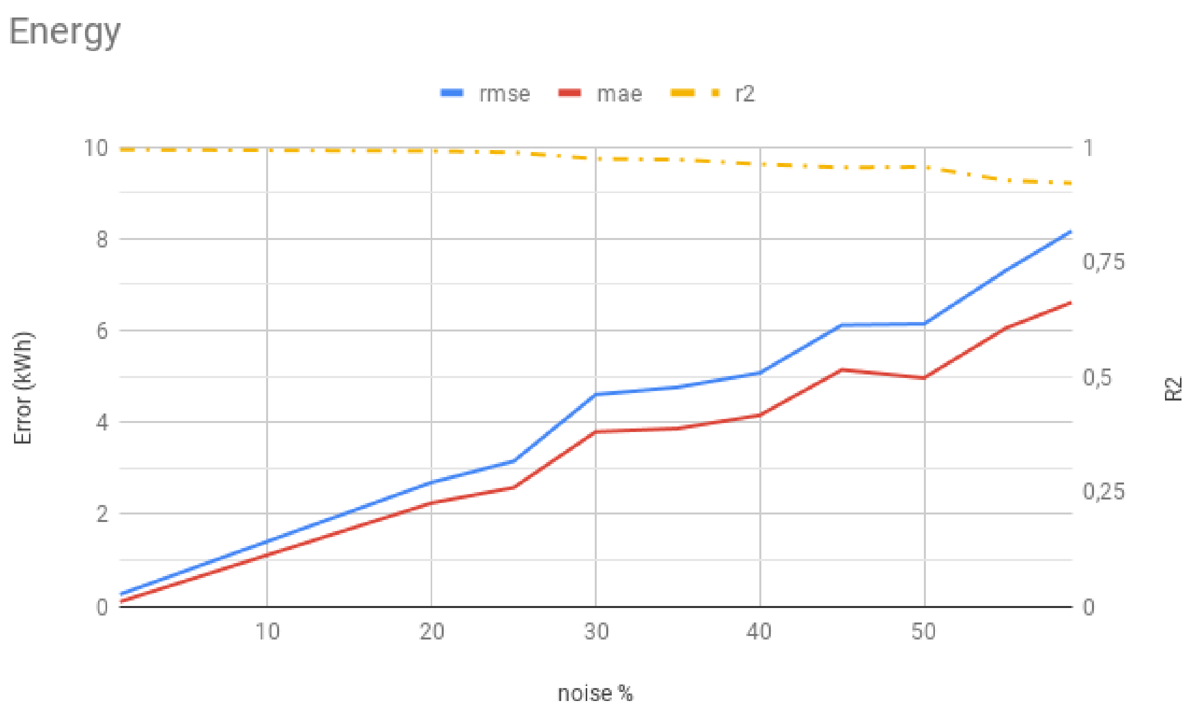
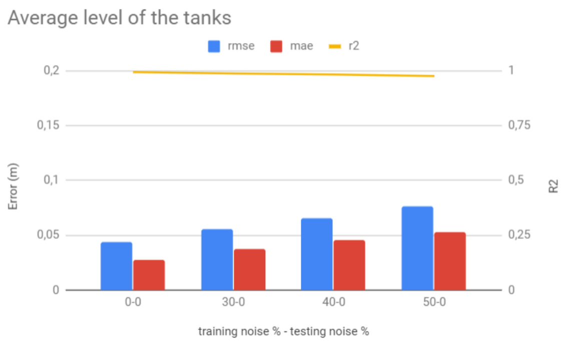
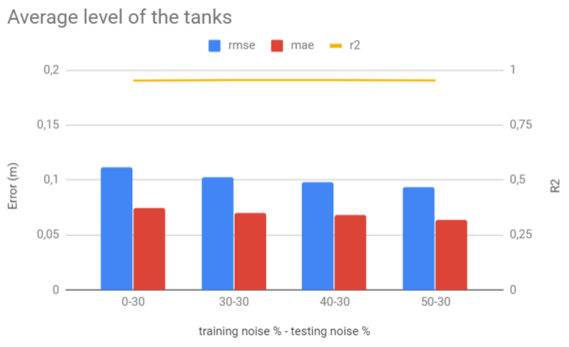


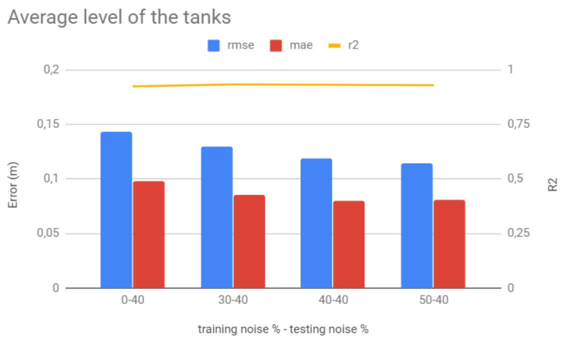
Figure 4.20: Richmond's of the results of RMSE, MAE and R^2 as the level of noise in the data increases on the energy that is being spent in kilowatts.



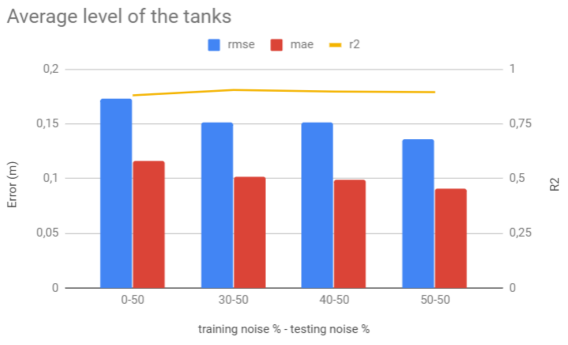
(a) 0% testing noise



(b) 30% testing noise

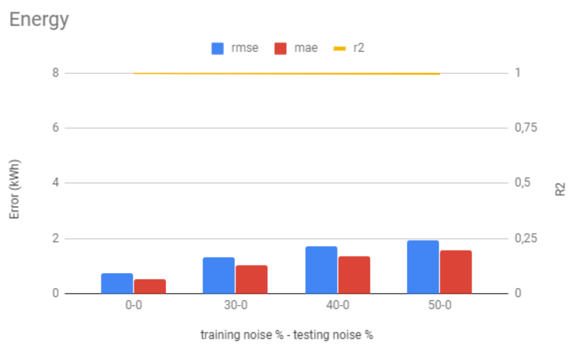


(c) 40% testing noise

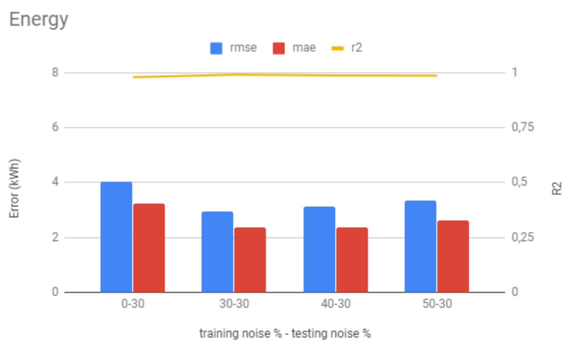


(d) 50% testing noise

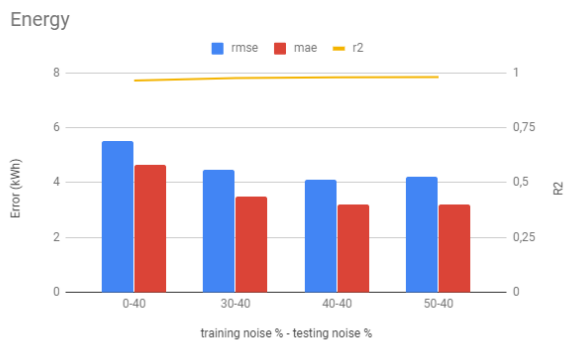
Figure 4.21: Richmond's simulation model results of the RMSE, MAE and R^2 metrics on the average level of the tanks for several amounts of noise to both the training and testing set.



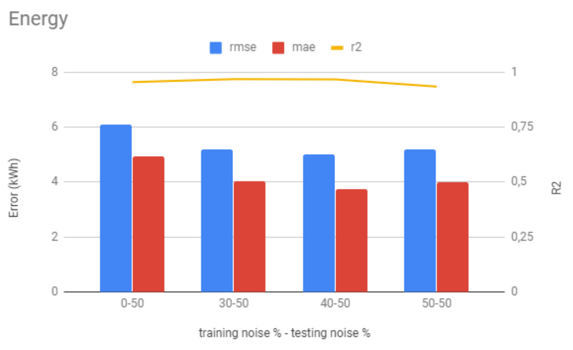
(a) 0% testing noise



(b) 30% testing noise



(c) 40% testing noise



(d) 50% testing noise

Figure 4.22: Richmond's simulation model results of the RMSE, MAE and R^2 metrics on the spent energy for several amounts of noise to both the training and testing set.

Training data amount analysis

The results on the performance impact of different amounts of training data are depicted in Figures 4.23 and 4.24 for the simulation of the tank levels and energy, respectively. Compared to Fontinha, this model requires more data to achieve a good performance, which is expected due to the size of the WDN that is being modelled. A noticeable decrease in performance is observed when the number of training samples becomes less than 720.

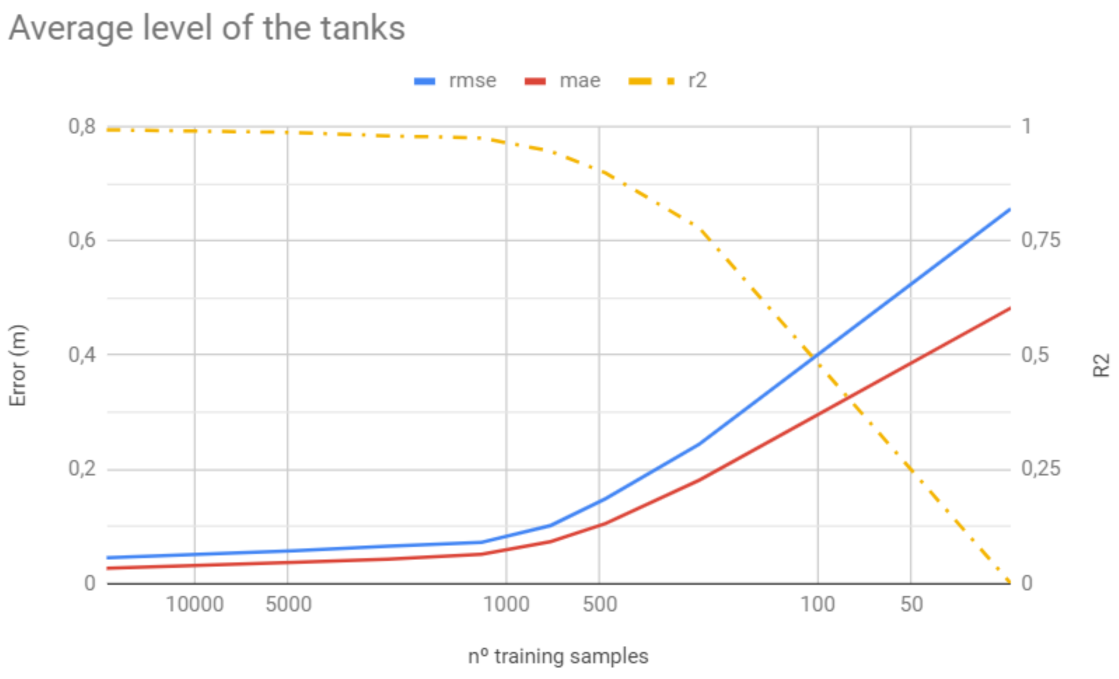


Figure 4.23: Richmond's simulation models results of RMSE, MAE and R^2 as the amount of training data decreases on the level of the tank.

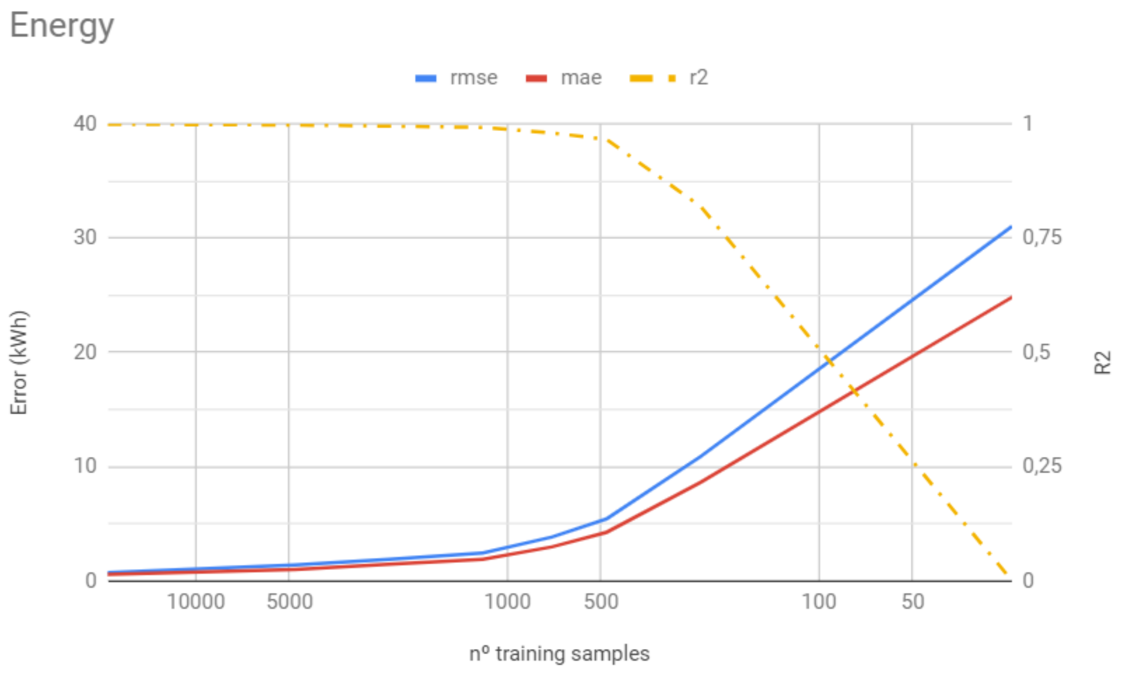


Figure 4.24: Richmond's simulation models the results of RMSE, MAE and R^2 as the amount of training data decreases on the energy that is being spent.

Conclusion

The case study of Richmond's network showed that the ANN model can precisely mimic the behaviour of a reasonable size network. In comparison with the previous model, Richmond's simulation model had slight changes regarding the number of neurons in its layers, the size of each training batch, and its learning rate.

The noise analysis revealed that the model can tolerate high percentages of noise with minor impact on its performance. Even with 60% of noise in the data, which is an unlikely amount of noise, the model behaved reasonably well, especially in the case of the energy simulation. However, it is not advisable to trust the models results if the data has a noise percentage higher than 40%. As for the results regarding the noise on the training data, it is possible to conclude that adding noise to a portion of the training set may improve the model's tolerance to noise. 40% of training noise seems to be the right amount to improve the model's tolerance without harming its performance in simulating both the tank levels and energy.

In a context where the amount of training data is limited, it can be expected for the model to perform well if the number of training sample is not less than 720.

The results from Richmond's simulation model demonstrates that the adopted methodology is able to create a model that replicates the behaviour of water networks of higher complexity.

CHAPTER 5

Conclusion

Forecasting water demands has proven be a challenge more complex than it first appeared. Given the results presented in this work, none of the considered methods were capable of significantly improving the performance of ANNs. The only model capable of producing results that rivalled those of the ANN were ELMs. In addition, they can also offer a solution that carries less computational burden. In this work, a dataset was used as a main reference with which the models were tested. The good performance displayed by the ANNs is due in part to the previously described *dd* data arrangement. Each demand that took place $24n$ hours prior to a certain timestep is considered as a feature. This, not only solves the problem of forecasting a whole operational horizon (*e.g* 24 h), but it also captures the seasonality of each hour of the day. Even if a specific model is the one that gets the best results with the data of a certain network consumption point, it does not necessarily mean that it would also be the one to obtain the best results with data from another point. It would be interesting to test these models with data from other consumption points that belong to other networks and offer different variations in their water demand patterns.

Regarding the simulation of water networks, the results show that ANNs are capable of thoroughly mimicking the behaviour of water distribution networks of various sizes and complexities. In this work it is assumed that the models are trained on data with a timestep of one hour, however, the methodology adopted allows a model to be trained with data that contains timesteps of varying lengths. This is beneficial since it is very common for companies to store data with irregular timesteps. Thus, avoiding the need to extrapolate or interpolate data in order to adapted it to a predetermine timestep. The results of a small experiment where the model is used to simulate a 24 h scenario with timesteps of varying sizes during training, the smallest being 15 min, can be consulted on Appendix A. Nonetheless, a more detailed analysis of this issue would be interesting. ANNs provide a computational efficient solution with the main advantage of being capable of learning by themselves the behaviour of the network without requiring a tedious and prolonged calibration process associated with hydraulic simulators.

References

[1] J. Nathanson, *Water supply system.* [Online]. Available: https://www.britannica.com/technology/water-supply-system (visited on 03/14/2019).

[2] J. Cuarta, *The Importance of Water Storage in Distribution Systems.* [Online]. Available: https://kyocp.wordpress.com/2012/10/02/the-importance-of-water-storage-in-distribution-systems/ (visited on 03/15/2019).

[3] S. M. Bunn and L. Reynolds, "The energy-efficiency benefits of pump-scheduling optimization for potable water supplies", *IBM Journal of Research and Development*, vol. 53, no. 3, 5:1–5:13, 2009, ISSN: 0018-8646. DOI: 10.1147/JRD.2009.5429018. [Online]. Available: http://ieeexplore.ieee.org/document/5429018/.

[4] Energy Agency International, *World Energy Outlook 2016*, ser. World Energy Outlook. OECD, Nov. 2016, p. 684, ISBN: 9789264264946. DOI: 10.1787/weo-2016-en. [Online]. Available: https://www.oecd-ilibrary.org/energy/world-energy-outlook-2016%7B%5C_%7Dweo-2016-en.

[5] B. Coelho, "Energy resources optimization in water supply networks", PhD thesis, Universidade de Aveiro, 2011, p. 89.

[6] A. Antunes, A. Andrade-Campos, A. Sardinha-Lourenço, and M. Oliveira, "Short-term water demand forecasting using machine learning techniques", *Journal of Hydroinformatics*, vol. 20, no. 6, 2018, ISSN: 1464-7141. DOI: 10.2166/hydro.2018.163.

[7] E. Salomons, A. Goryashko, U. Shamir, Z. Rao, and S. Alvisi, "Optimizing the operation of the Haifa-A water-distribution network", *Journal of Hydroinformatics*, vol. 9, no. 1, p. 65, 2007, ISSN: 14647141. DOI: 10.2166/hydro.2006.018. [Online]. Available: http://jh.iwaponline.com/cgi/doi/10.2166/hydro.2006.018.

[8] O. Lindeil E. and L. Kevin E., "Optimal control of water supply pumping systems", *J. Water Resour. Plann. Manage*, vol. 2, no. 10-11, pp. 237–252, 1994, ISSN: 0042207X. DOI: 10.1016/0042-207X(85)90371-9.

[9] U. S. E. P. Agency, *Epanet.* [Online]. Available: https://www.epa.gov/water-research/epanet (visited on 05/31/2019).

[10] E. Kozłowski, B. Kowalska, D. Kowalski, and D. Mazurkiewicz, "Water demand forecasting by trend and harmonic analysis", *Archives of Civil and Mechanical Engineering*, vol. 18, no. 1, pp. 140–148, 2018, ISSN: 16449665. DOI: 10.1016/j.acme.2017.05.006.

[11] M. Bakker, J. H. Vreeburg, K. M. van Schagen, and L. C. Rietveld, "A fully adaptive forecasting model for short-term drinking water demand", *Environmental Modelling and Software*, vol. 48, pp. 141–151, 2013, ISSN: 13648152. DOI: 10.1016/j.envsoft.2013.06.012. [Online]. Available: http://dx.doi.org/10.1016/j.envsoft.2013.06.012.

[12] S. L. Zhou, T. A. McMahon, A. Walton, and J. Lewis, "Forecasting daily urban water demand: A case study of Melbourne", *Journal of Hydrology*, vol. 236, no. 3-4, pp. 153–164, Sep. 2000, ISSN: 00221694. DOI: 10.1016/S0022-1694(00)00287-0.

[13] J. Bougadis, K. Adamowski, and R. Diduch, "Short-term municipal water demand forecasting", *Hydrological Processes*, vol. 19, no. 1, pp. 137–148, 2005, ISSN: 08856087. DOI: 10.1002/hyp.5763.

[14] A. Jain and L. E. Ormsbee, “Short-term water demand forecast modeling techniques - Conventional methods versus AI”, *Journal / American Water Works Association*, vol. 94, no. 7, pp. 64–72, 2002, ISSN: 0003150X. DOI: 10.1002/j.1551-8833.2002.tb09507.x.

[15] S. Alvisi, M. Franchini, and A. Marinelli, “A short-term, pattern-based model for water-demand forecasting”, *Journal of Hydroinformatics*, vol. 9, no. 1, p. 39, 2007, ISSN: 14647141. DOI: 10.2166/hydro.2006.016. [Online]. Available: http://jh.iwaponline.com/cgi/doi/10.2166/hydro.2006.016.

[16] Z. Y. Wu and X. Yan, “Applying genetic programming approaches to shortterm water demand forecast for district water system”, no. 2, pp. 1498–1506, 2010.

[17] E. Arandia, A. Ba, B. Eck, M. Asce, and S. Mckenna, “Tailoring seasonal time series models to forecast short-term water demand”, *Journal of Water Resources Planning and Management*, vol. 142, no. 3, pp. 1–10, 2016. DOI: 10.1061/(ASCE)WR.1943-5452.0000591..

[18] P. Vijai and P. B. Sivakumar, “Performance comparison of techniques for water demand forecasting”, *Procedia Computer Science*, vol. 143, pp. 258–266, 2018, ISSN: 1877-0509. DOI: 10.1016/j.procs.2018.10.394. [Online]. Available: https://doi.org/10.1016/j.procs.2018.10.394.

[19] S. L. Zhou, T. A. McMahon, A. Walton, and J. Lewis, “Forecasting operational demand for an urban water supply zone”, *Journal of Hydrology*, vol. 259, no. 1-4, pp. 189–202, 2002, ISSN: 00221694. DOI: 10.1016/S0022-1694(01)00582-0.

[20] L. M. Camarinha-matos, F. J. Martinelli, Q. Torre, and M. Caparica, “Application of machine learning techniques in water distribution networks assisted by domain experts”, 1998, ISSN: 978-3-642-33217-3. DOI: 10.1007/978-3-642-33218-0. arXiv: arXiv:1011.1669v3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33218-0.

[21] S. M. Weiss and N. Indurkhya, “Rule-based Machine Learning Methods for Functional Prediction”, *Journal of Arti cial Intelligence Research*, vol. 3, pp. 383–403, 1995.

[22] M. Ghiassi, F. Fa’al, and A. Abrishamchi, “Large metropolitan water demand forecasting using DAN2, FTDNN, and KNN models: A case study of the city of Tehran, Iran”, *Urban Water Journal*, vol. 14, no. 6, pp. 655–659, 2017, ISSN: 17449006. DOI: 10.1080/1573062X.2016.1223858.

[23] M. Ghiassi, H. Saidane, and D. Zimbra, “A dynamic artificial neural network model for forecasting time series events”, *International Journal of Forecasting*, vol. 21, no. 2, pp. 341–362, Apr. 2005, ISSN: 01692070. DOI: 10.1016/j.ijforecast.2004.10.008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169207004001116.

[24] A. K. Tiwari M., Adamowski J., “Water demand forecasting using extreme learning machines”, *Journal of Water and Land Development*, vol. 28, no. 14, pp. 37–52, 2016, ISSN: 20834535. DOI: 10.1515/jwld-2016-0004.

[25] S. Mouatadid and J. Adamowski, “Using extreme learning machines for short-term urban water demand forecasting”, *Urban Water Journal*, vol. 14, no. 6, pp. 630–638, 2017, ISSN: 17449006. DOI: 10.1080/1573062X.2016.1236133. [Online]. Available: http://dx.doi.org/10.1080/1573062X.2016.1236133.

[26] G. Guancheng, L. Shuming, W. Yipeng, L. Junyu, Z. Ren, and Z. Xiaoyun, “Short-Term Water Demand Forecast Based on Deep Learning Method”, *Journal of Water Resources Planning and Management*, vol. 144, no. 12, p. 4 018 076, 2018. DOI: 10.1061/(ASCE)WR.1943-5452.0000992. [Online]. Available: https://doi.org/10.1061/(ASCE)WR.1943-5452.0000992.

[27] C. W. Howe and F. P. Linaweaver, “The impact of price on residential water demand and its relation to system design and price structure”, *Water Resources Research*, vol. 3, no. 1, pp. 13–32, 1967, ISSN: 19447973. DOI: 10.1029/WR003i001p00013.

[28] J. Adamowski and C. Karapataki, “Comparison of Multivariate Regression and Artificial Neural Networks for Peak Urban Water-Demand Forecasting: Evaluation of Different ANN Learning Algorithms”, *Journal of Hydrologic Engineering*, vol. 15, no. 10, pp. 729–743, Oct. 2010, ISSN: 1084-0699. DOI: 10.1061/(ASCE)HE.1943-5584.0000245. [Online]. Available: http://ascelibrary.org/doi/10.1061/%7B%5C%%7D28ASCE%7B%5C%%7D29HE.1943-5584.0000245.

[29] B. Coulbeck and M. J. H. Sterling, “Modelling Techniques in Dynamic Control of Water Distribution Systems”, *Measurement and Control*, vol. 11, no. 10, pp. 385–389, Oct. 1978. DOI: 10 .

1177 / 002029407801101003. [Online]. Available: http : / / journals . sagepub . com / doi / 10 . 1177 / 002029407801101003.

[30] M. Sterling and B. Coulbeck, "A dynamic programming solution to optimization of pumping costs", *Proc. tnsfw Civ. Engrs*, vol. 118, pp. 813–818, 1975.

[31] P. W. Jowitt and G. Germanopoulos, *Optimal pump scheduling in water-supply networks*, 1992. DOI: 10.1061/(ASCE)0733-9496(1992)118:4(406).

[32] I. Nouiri, *Watnet software*. [Online]. Available: https://sites.google.com/site/drissamnouiri/watnet-software (visited on 05/31/2019).

[33] R. Huntington, "Uprating and Development of a Regional Telemetry Scheme", *Water Science and Technology*, vol. 28, no. 11-12, pp. 123–131, Dec. 1993. DOI: 10.2166/wst.1993.0652. [Online]. Available: https://iwaponline.com/wst/article/28/11-12/123-131/3989.

[34] M. Tavares, "Aplicação do EPANET ao Sistema Regional do Carvoeiro e análise do comportamento de indicadores de eficiência", PhD thesis, Universidade de Aveiro, 2018.

[35] Z. Rao and F. Alvarruiz, "Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model", *Journal of Hydroinformatics*, vol. 9, no. 1, p. 15, 2007, ISSN: 14647141. DOI: 10.2166/hydro.2006.014. [Online]. Available: http://jh.iwaponline.com/cgi/doi/10.2166/hydro.2006.014.

[36] T. M. Walski, E. D. Brill, J. Gessler, I. C. Goulter, A. M. Asce, R. M. Jeppson, M. Asce, K. Lansey, H.-l. Lee, S. Members, J. C. Liebman, and L. Mays, "Battle of the network models: Epilogue", vol. 113, no. 2, pp. 191–203, 1987.

[37] M. Behandish and Z. Wu, "GPU-based Artificial Neural Network Configuration and Training for Water Distribution System Analysis", *World Environmental and Water Resources Congress 2012: Crossing Boundaries*, pp. 3108–3121, 2012, ISSN: 00221112. DOI: 10.1111/j.1095-8649.2005.00822.x.

[38] Z. Y. Wu, M. El-Maghraby, and S. Pathak, "Applications of deep learning for smart water networks", *Procedia Engineering*, vol. 119, no. 1, pp. 479–485, 2015, ISSN: 18777058. DOI: 10.1016/j.proeng.2015.08.870. [Online]. Available: http://dx.doi.org/10.1016/j.proeng.2015.08.870.

[39] G. E. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets", *Neural computation*, vol. 18, no. 7, pp. 1527–54, 2006, ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. arXiv: 1111.6189v1. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16764513.

[40] B. Coelho, "Energy efficiency of water supply systems using optimisation techniques and micro-hydroturbines", PhD thesis, Universidade de Aveiro, 2016, p. 268.

[41] M. Jordan, J. Kleinberg, and B. Scho, *Pattern Recognition And Machine Learning*. Springer, 2006, p. 758, ISBN: 9780387310732.

[42] R. Byrd, L. Peihuang, and J. Nocedal, "A limited-memory algorithm for bound-constrained optimization", Argonne National Laboratory (ANL), Argonne, IL, Tech. Rep., Mar. 1996. DOI: 10.2172/204262. [Online]. Available: http://www.osti.gov/servlets/purl/204262-FVeKR4/webviewable/.

[43] *scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation*. [Online]. Available: https://scikit-learn.org/stable/ (visited on 05/26/2019).

[44] *Home - Keras Documentation*. [Online]. Available: https://keras.io/ (visited on 03/26/2019).

[45] *Dropout Layer · Artificial Inteligence*. [Online]. Available: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/dropout%7B%5C_%7Dlayer.html (visited on 03/31/2019).

[46] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, Dec. 2006, ISSN: 0925-2312. DOI: 10 . 1016 / J . NEUCOM . 2005 . 12 . 126. [Online]. Available: https : / / www . sciencedirect . com / science / article / pii / S0925231206000385.

[47] W. McGinnis, *Extreme Learning Machines — sklearn-extensions*, 2015. [Online]. Available: http://wdm0006.github.io/sklearn-extensions/extreme%7B%5C_%7Dlearning%7B%5C_%7Dmachines.html (visited on 04/11/2019).

[48] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", pp. 1–9, 2014, ISSN: 19457901. DOI: 10.1109/ICORR.2015.7281186. arXiv: 1412.3555. [Online]. Available: http://arxiv.org/abs/1412.3555.

[49] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, no. 2, 1994, ISSN: 19410093. DOI: 10.1109/72.279181. arXiv: arXiv:1211.5063v2.

[50] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", in *Empirical Methods in Natural Language Processing*, vol. 28, 2014, pp. 1724–1734, ISBN: 0021-9258 (Print)\r0021-9258 (Linking). DOI: 10.3115/v1/D14-1179. arXiv: 1406.1078.

[51] G. An, "The effects of adding noise during backpropagation training on a generalization performance", *Neural Computation*, vol. 8, no. 3, pp. 643–674, Apr. 1996, ISSN: 0899-7667. DOI: 10.1162/neco.1996.8.3.643.

[52] J. E. V. Zyl, D. a. Savic, and G. a. Walters, "Operational optimization of Water Distribution Systems using a hybrid Genectic Algorithm", *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 160–170, 2004.

[53] *Richmond*. [Online]. Available: http://emps.exeter.ac.uk/engineering/research/cws/research/distribution/benchmarks/operation/richmond.html (visited on 06/02/2019).

Appendix A

Validation with varying timestep sizes

Figure 1 demonstrates the inputs of a network simulation of a 24 h scenario. The data is displayed with 15 min intervals but the simulation model was trained with timesteps of various sizes, the smallest being 15 min. In this context, the values of the pump statuses can take values greater than one which means that it remained active for more than 15 min. The consequent forecasting and evaluation metric results are displayed on Figure 2 and on Table 1. This results show that the adopted methodology can in fact produce good results even when the model is trained in data of various time intervals.

	RMSE	**MAE**	R^2
Tank level (m)	0,0352	0,0180	0,9922
Energy (kW)	0,7036	0,4090	0,9995

Table 1: Fontinha's simulation models RMSE, MAE and R^2 metric scores for the tank level and energy on a 24 h scenario.

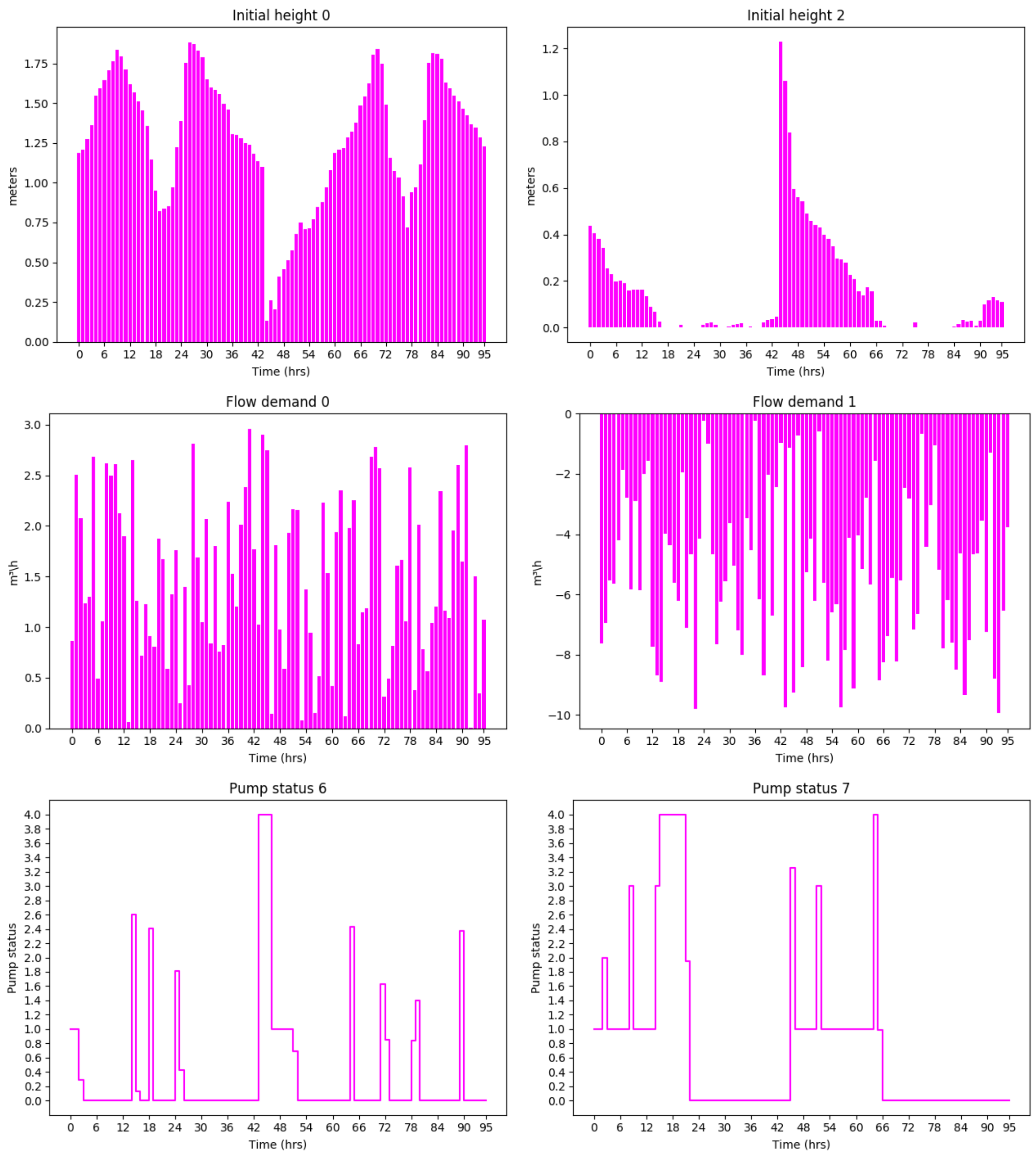


Figure 1: Portion of Richmond's simulation model inputs on a 24 h scenario with timesteps of varying size.

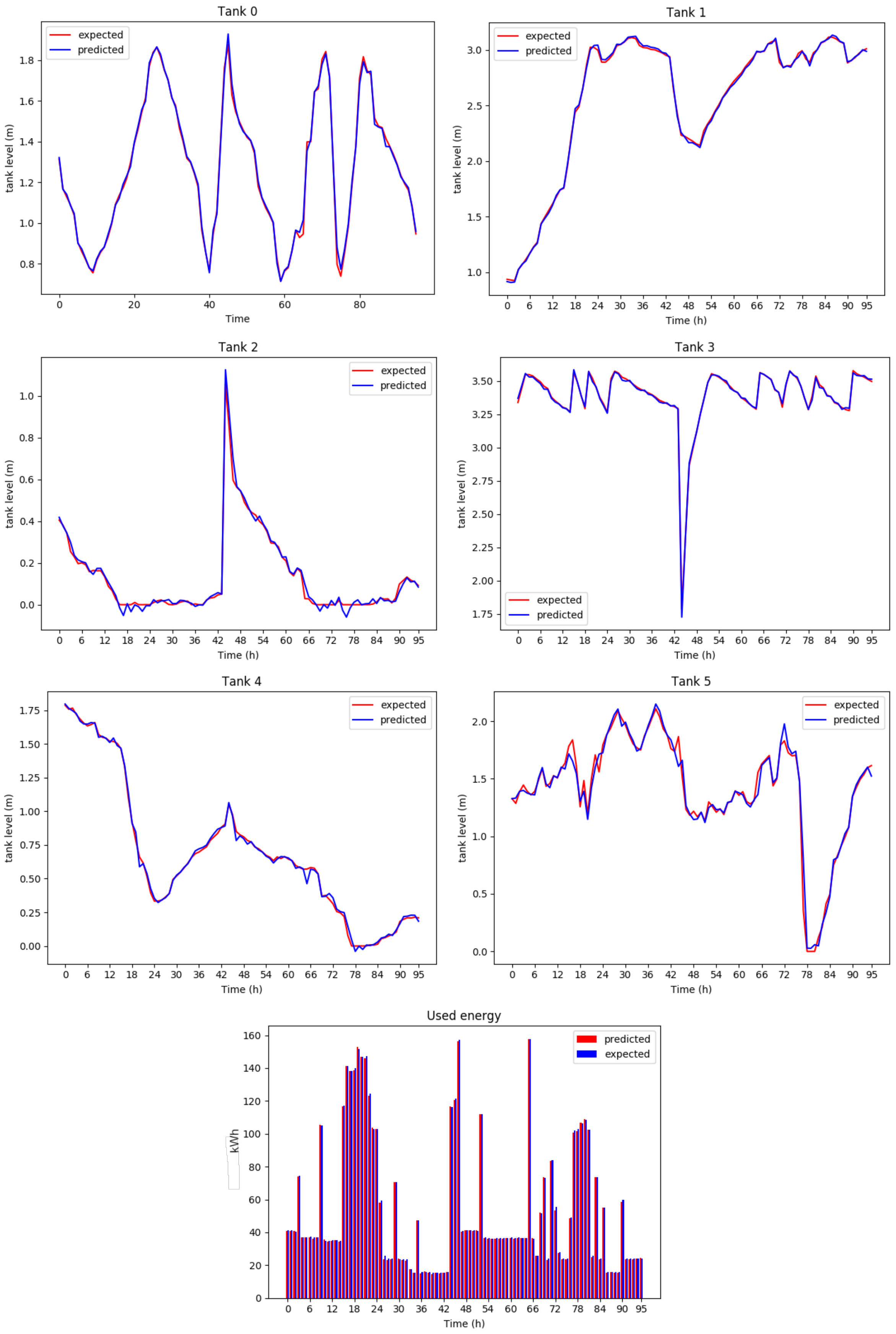


Figure 2: Richmond's simulation model outputs on a 24 h scenario with timesteps of varying size.