Machine Learning for Short-Term Water Demand Predictions

Submitted by Guoxuan Liu

to the University of Exeter as a thesis for the degree of

Doctor of Philosophy in Engineering

May 2023

I certify that all material in this thesis which is not my own work has been identified and that any material that has previously been submitted and approved for the award of a degree by this or any other University has been acknowledged.

Abstract

Urban water supply is coming under increased pressure due to urbanisation, water scarcity and climate change. Efficient urban water management can help alleviate this pressure by improving service quality and reducing water loss. Accurate demand and consumption forecasting enables expansion planning, financing, and operation of water distribution systems. Current research often focuses on model-centric approaches where the model is improved to drive forecast accuracy; however, more efficient data usage could be realised as an alternative to model-centric approaches, without incurring additional computation costs. This work investigates the potential of data-centric forecasting approaches, focusing on ways to improve the efficiency of data and computation resource usage for short-term water demand forecasting.

To initiate the investigation, several intrinsically different forecasting models are analysed. A total of four different forecasting models, i.e., Prophet, Autoregressive Integrated Moving Average, Neural Network (NN) and Random Forest (RF) are applied to four demand datasets, i.e., one Chinese hourly demand dataset and three UK 15-minute demand datasets. Various aspects of data and model requirements for optimal performance are investigated. Results obtained from the case studies show that prolonging training data may not be necessary, and that accurate sub-daily water demand forecasting is possible with 10 days of past data for model training. In terms of accuracy, neural network and random forest tend to be better suited towards short-term water demand forecasting over statistical models.

The second part of the work aims to unbox the four black-box machine learning methods – NN, Long Short-Term Memory (LSTM), RF, Extreme Gradient Boosting (XGB) and explain their inner workings using SHapley Additive exPlanations and Local Interpretable Model-Agnostic Explanations, Prophet and ARIMA are excluded due to inferior forecasting accuracy. Results have found that feature requirement depends on data resolution, the forecasting model used and the forecast time of day. Network-based models (NN and LSTM) are more temporally dependent and feature intensive, indicating that they require more feature inputs to produce equal accuracy compared to tree-based models (RF and XGB). High-resolution forecasts can maintain a high level of accuracy with only one feature at the previous point.

The final part of the work analyses the possibility of incorporating Transfer Learning (TL) into the context of water demand forecasting. To evaluate the potential of TL, 18 UK DMAs water demand datasets are used. Experiments are designed to predict water demands in one DMA that has limited or unavailable data, with an aim to anaysing the forecasting ability of models built with alternative DMA data. Results have found that four and eight external DMA datasets are respectively suitable for 15-minute and hourly demand and that limited accuracy gains are achieved from samples size larger than 20,000. Finally, TL-incorporated methods can improve machine learning forecasting accuracy when there is limited data availability.

The results obtained in this study prove the usefulness of data-centric approaches' ability to improve forecasting accuracy. The data-centric approaches explored in this thesis can be used to guide the development of machine learning-based short-term demand forecasting models for researchers, operators, and

utilities. Efficient use of forecasting models and demand data holds further potential in improving forecast accuracy, reducing computation cost, and bettering user confidence in the application of machine learning models.

Table of Contents

Acknowledgements

First, I would like to thank my supervisors, Prof Guangtao Fu and Prof Dragan Savic, for their endless patience, support and trust in me to complete my work. Thank you for accepting me in this PhD programme, and for waiting and encouraging me to complete my degree. I am sorry that I have taken as long as I have, thank you for giving me hope on those many occasions when I thought to give up.

I would then like to thank my family. I could not dream of reaching completion without the financial, and emotional support, and abundant free food from home. I have to say, Mother knows best. Thank you, Mum, for that reverse psychology trick eleven years ago, when you tried to dissuade me from attending university. The trick may have worked too well, I would like to stay here forever.

Next, I want to thank the research team at the Dalian University of Technology. I am grateful to Prof Zhang Chi for hosting, to Prof Haixing Liu for tutoring me, and to Mengke Zhao and Chao Zhang for looking after me. The yearlong exchange has been the highlight of my PhD life, shame it could not be longer.

Finally, special thanks go to Adam, Zilin, Xi and Viola. I do not wish to read anyone else's thesis, but Adam has kept to his promise from seven years ago, I am eternally grateful to have you in my life. And to Zilin, Xi and Viola, I have heard many horror stories of the final thesis writing stage, you guys have not only made it bearable, but it was enjoyable. Though as fun as it has been, I do not wish to do it again.

List of Figures

List of Tables

List of Abbreviations

ARIMA	Autoregressive Integrated Moving Average
CHN	China
DMA	District Metered Area
GI	Gini Index
LIME	Local interpretable model-agnostic explanations
LSTM	Long Short-Term Memory
ML	Machine Learning
NN	Neural Network
R^2	Coefficient of determination
RF	Random Forest
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlnantions
TL	Transfer Learning
XGB	Extreme Gradient Boosting

Chapter 1 - Introduction

1.1. Motivation

Water demand management is essential for ensuring water security in urban centres, which are increasingly coming under threat due to urbanisation, water scarcity, ageing infrastructure, and climate change. An effective way to mitigate the increasing threat is to make accurate demand and consumption forecasts, for short, medium, and long forecasting horizons; these different horizons aid utilities with the operation, financing, and planning-related issues (Donkor et al. 2014).

For operational management, short-term water demand forecasting can improve resource allocation, reduce cost and conserve treated water. All the benefits relate to minimising waste, whilst ensuring base customer requirements are met. For demand forecasting to be effectively used, a high level of accuracy is necessary (Wu and Liu 2017).

A great deal of research effort has gone into improving the accuracy of short-term water demand forecasting, though much of the work has focused on using model-centric approaches (Adamowski et al. 2012; Chen et al. 2017; Chen and Boccelli 2018; Gagliardi et al. 2017; Herrera et al. 2010; Lertpalangsunti et al. 1999; Liu et al. 2022; Sardinha-Lourenço et al. 2018). These approaches focus on developing and adapting models to data, through various approaches including parameter optimisation, alterations to model structure and ensemble models.

For machine learning (ML) models with well-defined structures, such as Prophet (Taylor and Letham 2018) and Autoregressive Integrated Moving Average (ARIMA) (Box E. P. G. et al. 2015), parameter optimisation has been the core focus when it comes to model-centric forecasts accuracy improvement (Menculini

et al. 2021; Papacharalampous and Tyralis 2018; Weytjens et al. 2021). Whilst parameter optimisation is also employed in more complex models, such as Neural Networks (NN) (McCulloch and Pitts 1943) and Random Forest (RF) (Breiman 2001), it is often not the focus of the investigation, as these models can be further improved in the model structure. Examples such as changes in the number of hidden layers for NN have shown effective to improve forecasting accuracy, but these improvements are gained at the cost of further computational complexity (Adamowski et al. 2012; Adamowski 2008; Chen et al. 2017; Ghiassi et al. 2008; Toharudin et al. 2023). Alternatively, ensemble modelling combines several forecasting models with either equal or different weights for individual models, and this approach draws out the advantages of individual models, thus achieving higher accuracy compared to individual models (Bata et al. 2020; Grover et al. 2015; Lertpalangsunti et al. 1999).

In contrast to model-centric approaches, data-centric machine learning approaches have received limited attention in the field of time series forecasting including short-term water demand forecasting (Fu et al. 2022). The idea of a data-centric approach has been popularised by Andrew Ng in recent years (DeepLearningAI 2021). The Data-Centric AI Competition (DeepLearning.AI and Landing AI 2021) followed not long after, where participants were asked to improve a dataset using data-centric techniques before it is fed to a fixed model and the level of accuracy improvement that can be achieved is evaluated. In the research communities, the terms data-centric and data-driven were generally used interchangeably before Andrew Ng's definition in 2021. This is evident from many studies that explicitly mentioned the term 'data-centric', yet merely used data for forecasting purposes, the core research focuses still lies on the model or forecasting system (Böse et al. 2017; Faeldon et al. 2014; Grover et al. 2015).

More recently, data-centric approaches were demonstrated more distinctly by Kang et al. (2021), whereby the research has forgone forecasting models. Instead, they used a large pool of real data from multiple sources as reference data. The similarity between the target series (series for forecasting) and the pool of reference series is measured, and a subset of reference series that is most like the target series is chosen for forecasting the target series.

However, a model-less approach is not the sole data-centric method. Guo et al. (2018) have shown that the application of expert knowledge in data type can greatly improve forecasting accuracy. This can also be considered a data-centric approach, as data input is optimised via expert knowledge to improve forecasting efficiency and accuracy. Whilst Guo et al. (2018) understood the data characteristics to optimise input, this is not always the case. To overcome this, ML model explainability has received an increasing amount of effort recently.

A wide range of techniques has been developed for machine learning model explainability. Barredo Arrieta et al. (2020) recognised that some statistical models are explainable due to the simplicity of model structure. More complex ML models require post-hoc approaches to become explainable. The post-hoc approaches are split into model-agnostic and model-specific approaches. As the names suggest, the former works on all machine learning models and the latter works on specific models. Specifically, model-agnostic approaches such as Local Interpretable Model-Agnostic Explanations (LIME) (Garreau and von Luxburg 2020) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017) calculate individual feature contributions to forecasting accuracy. The former does so on a sample-by-sample basis, whilst the latter can produce an overview of all features across all samples. To our knowledge, explainability has not been

considered in the field of water demand forecasting, the closest research that employs an ML model explainer is the work by (Li et al. 2022a), where factors that impact beach water quality are analysed and ranked.

Another approach that optimises the use of available data is Transfer Learning (TL). The idea has seen wide adoption in power/electricity forecasting (Fan et al. 2020; Gao et al. 2020; Le et al. 2020; Ribeiro et al. 2018; Sarmas et al. 2022; Xu and Meng 2020; Zhang and Luo 2015), among other fields (Chen et al. 2020; Karb et al. 2020; Kimura et al. 2019; Liu et al. 2020; Peng et al. 2022; Zheng et al. 2022), though no such efforts are evaluated in the field of water demand forecasting. The intuitive idea of TL is that knowledge from alternative datasets can be leveraged to improve the forecasting accuracy of a target dataset, this can be particularly useful where quality data is scarce. Because water and power demand share similar temporal and spatial relationships, transfer learning could prove equally useful for short-term water demand forecasting.

In this thesis, the data-centric approaches encompass all approaches that seek to maximise information extraction from available data. Three approaches are explored in this thesis, 1) optimise data to models to improve forecasting accuracy; 2) determine feature impact to eliminate non-essential feature inputs; and 3) investigate transfer learning to evaluate the potential of using external data to mitigate the lack of data availability.

The work presented in this thesis uses real-life water demand data from UK and China, to identify more efficient and reliable ways to use past demand data, thus improving the accuracy of short-term water demand forecasting. A range of ML models is tested, using a variety of input scenarios, as well as feature and data

availability scenarios. The results presented could act as a guide to researchers, operators, and utilities.

1.2. Research Questions and Aims

This work explores the potential of a data-centric approach for short-term water demand forecasting. This section starts by outlining several research questions, followed by the aims and objectives that will individually address each question.

1.2.1. Research Questions:

- How do different data input structures impact different forecasting models, and could this be generalised and optimised?
- Could the use of post-hoc Machine Learning model explainers replace human expertise, to improve forecasting accuracy and efficiency?
- Can the use of Transfer Learning mitigate the impact of quality data scarcity, to maintain or improve short-term water demand forecasting accuracy?

1.2.2. Aims and Objectives:

This work aims to develop data-centric approaches for short-term water demand forecasting using historical data. To achieve this, the key objectives are as follows:

- To collect and process water demand data that will be used for model training and testing.
- To better understand different model requirements by altering input features to different forecasting models.

- To review the latest machine learning approaches in the literature that are used for short-term water demand forecasting.

- To improve the understanding of the inner workings and different input requirements between machine learning models, by applying post-hoc model explainers to extract key features.

- To evaluate Transfer Learning techniques in short-term water demand forecasting, to determine its potential for reducing computation cost, improving accuracy, and potentially rectifying missing or error data.

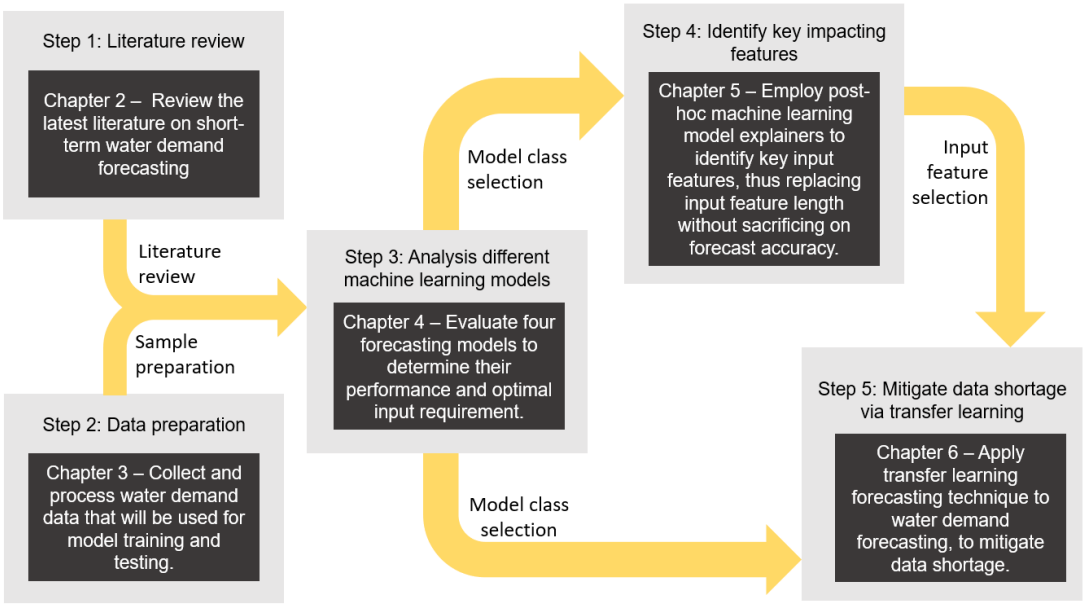

Figure 1.1 Illustration of the relationship between main research topics.

Figure 1.1 illustrates how the methodological chapters are connected, as well as how it relates to the aims and objectives. Steps 1, 2 and 3 respectively correspond with the three research questions asked in subsection 1.2.1. Step 1 evaluates various forecasting models, it determines the suitability of different forecasting models for short-term water demand forecasting, it also evaluates model input and training requirements for univariate water demand forecasting. This lays the groundwork by determining the optimal performance forecasting

class, thus not all forecasting models need to be carried forward for later investigation. Step 2 employs post-hoc ML model explainers to reduce input feature requirements, this could replace expert knowledge in model and data pairing, and the reduced input would improve model efficiency without sacrificing accuracy. Step 3 applies transfer learning forecasting techniques to water demand forecasting, coupled with the reduced feature requirement determined in Step 2, the data requirement could be significantly reduced by the novel technique.

1.3. Thesis Overview

This thesis is divided into seven chapters. Following this chapter, **Chapter 2** provides a comprehensive literature review of research done in the field of short-term demand forecasting. Including all approaches and models used, as well as some alternatives, to shed light on the reasoning for choosing the models evaluated in this thesis. Additionally, the performance indicators and their theories are presented in this section to show what has been used, and the reasoning behind indicator choice.

Chapter 3 provides the details of the data used for model development and testing. The source and statistical characteristics of all datasets are presented. Additionally, data-cleaning techniques performed throughout the thesis are presented and discussed.

Chapter 4 addresses the first objective. By analysing the forecasting accuracy impact of input structures, the optimal data length, temporal resolution, and noise tolerance for four different forecasting models. This is achieved with three experiments the varying input structure, feature length, temporal resolution, and data noise level.

Chapter 5 addresses the second objective. Two ML model explainers are applied to four ML forecasting models, using data of different temporal resolutions. Four experiments are designed to divulge how forecasting models use input features differently; it focuses on the forecasting models' temporal dependence, individual feature dominance and how feature requirements differ for forecasting demands at different times.

Chapter 6 addresses the third objective. Two TL incorporated ML models are evaluated. The impact of external DMA demand data for cases with limited data availability is measured on two ML models across five different temporal resolution and feature pairing combinations. Experiments are designed to reveal external data impact and compare how best to select external data.

Finally, **Chapter 7** provides a summary of the experiments performed, including key results and findings, as well as the limitations of this study and future research recommendations.

1.4. Contributions

The key contributions of this thesis are as follows:

A guide of suitable input structure for different forecasting models. Four forecasting models are evaluated, and the result has shown that they require a different amount of training data and have different levels of noise tolerance. The improved forecasting results act as a guide for forecasters to pair data to models more effectively.

Improved understanding of how ML models use historical water demand data to make accurate forecasts. The application of ML model explainers has proven useful for short-term water demand forecasting, where previously

unknown interactions between data and model have been successfully unveiled. This can assist forecasters to decide on the choice of input data, especially in the absence of expert data knowledge.

An insight into contributions of key input features for short-term water demand forecasting. The list of key input features in terms of their contributions to demand predictions across two temporal resolutions is determined through artificial intelligence explanation methods. The finding can be used as a guide to bypass the need for expert knowledge of data.

A new forecasting technique that improves forecasting accuracy, reduces the impact of limited data availability, and reduces computation cost. The Transfer Learning (TL) technique has proven useful in the field of short-term water demand forecasting. Four and eight external DMA datasets have been shown to respectively suit 15-minute and hourly demand; limited accuracy gain is achieved by samples size beyond 20,000. Correlation-based and quality-based TL incorporation have marginal accuracy differences, though the latter requires far fewer trained models to forecast all DMA.

Further insight into the potential of data-centric and model-centric approaches. The work has shown that whilst data-centric approaches can greatly improve forecasting result, time and effort is still required on the model-centric approaches, though to a lesser extent.

1.5. Published papers

Liu, G., D. Savic, and G. Fu. 2023. "Short-term water demand forecasting using data-centric machine learning approaches." Journal of Hydroinformatics. IWA Publishing. https://doi.org/10.2166/hydro.2023.163.

Liu, G., G. Fu, D. Savic. "Unboxing black-box machine learning models for short-term water demand forecasting." (Submitted)

Liu, G., G. Fu, D. Savic. "Optimising the Usage of Multiple Short-term Water Demand Data via Transfer Learning." (Submitted)

Chapter 2 - Literature Review

This chapter presents the existing literature in the field of short-term water demand forecasting. The chapter is split into five sections. The model-centric and data-centric approaches are first discussed in detail, to show that there is a large research gap around the data-centric forecasting approach, where accuracy can be improved by better use of existing data. The second section looks at the machine learning (ML) models that are available for demand forecasting. The third and fourth sections respectively present Transfer Learning (TL) and ML model explainers, although these methods are yet to be applied to the field of water demand forecasting, their successful application and performance in other fields are presented and discussed. This is followed by popular performance indicators used in the field of water demand forecasting, along with choice and reasoning on selected indicators. Finally, the chapter ends with a summary of research gaps in the field, to elaborate on how this thesis addresses these issues.

2.1. Introduction

Short-term water demand forecasting has received an abundance of research attention. Much of the existing research has taken a model-centric approach, where efforts are spent on altering models to improve forecasting accuracy; there has been limited research effort on data-centric approaches, where forecasting accuracy is improved via more efficient use of data.

2.1.1. Model-centric Approach

Mode-centric approach refers to approaches where forecasting accuracy is improved by developing and adapting models to data. Much of the existing research employs this approach (Adamowski et al. 2012; Chen et al. 2017; Chen

and Boccelli 2018; Gagliardi et al. 2017; Herrera et al. 2010; Lertpalangsunti et al. 1999; Liu et al. 2022; Sardinha-Lourenço et al. 2018). The model-centric approaches are applied in various ways, including parameter optimisation, alterations to model structure and ensemble models.

For models with well-defined structures, such as Prophet (Taylor and Letham 2018) and Autoregressive Integrated Moving Average (ARIMA) (Box E. P. G. et al. 2015), parameter optimisation has been the core focus when it comes to model-centric forecasts accuracy improvement (Menculini et al. 2021; Papacharalampous and Tyralis 2018; Weytjens et al. 2021).

Whilst parameter optimisation is also employed in more complex models, such as Neural Network (NN) and Random Forest (RF), it is often not the focus of the investigation, as these models can be further improved in the model structure. Examples such as changes in the number of hidden layers for NN have shown effective to improve forecasting accuracy, but these improvements are gained at the cost of further computational complexity (Adamowski et al. 2012; Adamowski 2008; Chen et al. 2017; Ghiassi et al. 2008; Toharudin et al. 2023).

Alternatively, ensemble modelling combines several forecasting models with either equal or different weights for individual models, and this approach draws out the advantages of individual models, thus achieving higher accuracy compared to individual models (Bata et al. 2020; Grover et al. 2015; Lertpalangsunti et al. 1999).

2.1.2. Data-centric Approach

In contrast to model-centric approaches, the idea of a data-centric approach has only been popularised by Andrew Ng in recent years (DeepLearningAI 2021). The

Data-Centric AI Competition (DeepLearning.AI and Landing AI 2021) followed not long after the initial discussion, where competition participants were asked to improve dataset classification using data-centric techniques, the result is fed to a fixed model and the level of accuracy improvement achieved is evaluated.

Before Andrew Ng's definition in 2021 (DeepLearningAI 2021), the research communities used the terms data-centric and data-driven interchangeably. This is evident from many studies that explicitly mentioned the term 'data-centric', yet merely used data for forecasting purposes, the core research focuses still lies on the model or forecasting system (Böse et al. 2017; Faeldon et al. 2014; Grover et al. 2015). Andres Ng has recognised that whilst data processing takes up 80% of machine learning researchers' time, very few researchers have spent time trying to improve it.

More recently, however, data-centric approaches were demonstrated more distinctly by Kang et al. (2021), whereby the research has forgone forecasting models. Instead, they use a large pool of real data from multiple sources as references. The similarity between the target series (series for forecasting) and the pool of reference series is measured, and a subset of reference series that is most like the target series is chosen for forecasting the target series.

In addition to improving forecasting accuracy, this thesis also considers more efficient use of data, as another data-centric approach. Data efficiency has been applied in the field of marketing (Zhao et al. 2019) and gene selection (Ding and Peng 2005), through a known as Maximum Relevance and Minimum Redundancy (MRMR). MRMR improves input efficiency by selecting features that are most correlated to the output, whilst least correlated to each other, this reduces the amount of repeated or useless information. Although similar methods

have not been applied to the field of water demand forecasting, Guo et al. (2018) have shown that for univariate short-term water demand forecasting, continuous past demands are not necessary to all be used as features; instead, they have used expert knowledge in data as an alternative to MRMR, to reduce feature inputs. This approach improves forecasting model efficiency without sacrificing accuracy. Though both MRMR and expert knowledge can reduce feature requirements, thus improving feature efficiency; MRMR only offers a guide to feature selection, whilst expert knowledge is very case specific. Fortunately, developments on explainability in ML models have recently become topical (Barredo Arrieta et al. 2020) and would be of great potential to fill in the gap in the feature selection dilemma. Details of existing methods and applications are presented in Section 2.4.

Transfer Learning (TL) can also be considered a data-centric approach. It improves forecast accuracy and efficiency by leveraging the knowledge learnt from external datasets to aid the forecast of a target dataset. Details of its progress and application are presented in Section 2.3. Though it has not been applied in the field of water demand forecasting, its research progress in other fields is presented to shed light on how it could be applied for short-term water demand forecasting.

Whilst this thesis focuses on data-centric approaches, it cannot ignore the potential of model-centric approaches. Most of the research effort is on improving and better-using data, but model and parameter selection are also investigated, albeit not in detail.

2.2. Machine Learning Models

To evaluate the potential of the data-centric approach, several forecasting models are discussed and considered. This section categorises popular forecasting models into four classes – time-series models, network-based models, tree-based models, and ensemble models. Each category is discussed independently in the following subsections, Table 2.1 presents a comparative overview of all models evaluated and provides the reason for the chosen models.

2.2.1. Time-series Models

Amongst time-series forecasting models, Autoregressive Integrated Moving Average (ARIMA) has commonly been employed as a benchmark to evaluate other forecasting models (Adamowski et al. 2012; Chen and Boccelli 2018; Guo et al. 2018; Sardinha-Lourenço et al. 2018; Tiwari and Adamowski 2013). ARIMA model was developed by Box and Jenkins in 1970 (Box E. P. G. et al. 2015), it combines the autoregressive (AR) and moving average (MA) models with a built-in differencing term. Additionally, seasonality factors can be incorporated to make seasonal ARIMA (SARIMA). Though ARIMA rarely outperforms other existing forecasting models, its prevalence makes it the ideal benchmark model to evaluate other forecasting models.

More recently, the Facebook research team have developed a modular regression model – Prophet (Taylor and Letham 2018). The original research paper evaluated events created on Facebook as source data to compare multiple time-series forecasting models, and Prophet has shown superior performance compared to ARIMA, Exponential Smoothing and Random Walk.

Beyond the original paper, other research that compares Prophet to existing forecasting models has shown inconsistent results. Examples of accuracy

comparisons made between Prophet and Linear Regression (LR), Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) have shown mixed results (Gupta et al. 2021; Rahman et al. 2020; Toharudin et al. 2023). However, research that has exclusively used Prophet for forecasting has explored additional features within the model (Aguilera et al. 2019; Xie et al. 2021). As it is a modular additive time-series model, the forecasts are made up of intuitively understandable seasonal components. The seasonal components can be analysed individually, the individual results can offer useful insights, in addition to accurate forecasts.

Overall, time-series forecasting models produce more understandable forecasts compared to other more complex forecasting models, including network-based or tree-based models. Its understandability owes to its simple structures, this allows for the model to produce explainable forecasting results. ARIMA achieves this from the structural set-up, where the number of AR, MA and differencing terms makes clear how the data series relates to its past self; and Prophet achieves this by presenting the modular results, where different seasonality can be individually examined.

2.2.2. Network-based Models

Network-based models have been widely applied to water demand forecasting (Boudhaouia and Wira 2021; Fu et al. 2022; Guo et al. 2018; Tiwari and Adamowski 2013). The Neural Network (NN) forecasting model was introduced by McCulloch and Pitts (1943), prediction can be made by training a model that passes training data forward and backwards to optimise model weighting.

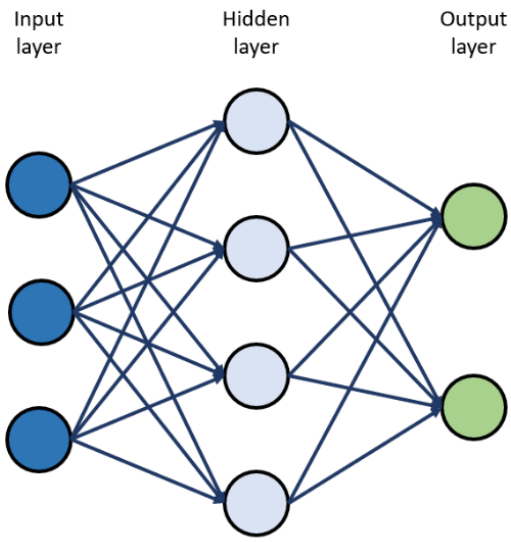

Figure 2.1 Multi-layer perceptron (Fu et al. 2022)

The basic three-layered neural network is known as a Multi-Layer Perceptron (MLP), its structure is presented in Figure 2.1. MLP has been widely applied in the field of short-term water demand forecasting (Fu et al. 2022) and has shown superior forecasting ability compared to time-series forecasting models (Adamowski et al. 2012; Vijai and Bagavathi Sivakumar 2018).

Beyond the basic three-layered structure, a variety of alternative network-based models have been used for prediction and classification. These included Deep Neural Network (DNN) – where the three-layer model is extended to 4 or more, but operates similarly to three-layered networks; Gated Recurrent Unit (GRU) – this model improves upon basic NN model by having an update and reset gates, this allows for information selection, which solves the vanishing gradient problem exhibited by basic NN by retaining long ago information; and Long Short-Term Memory (LSTM), presented in Figure 2.2, structurally, LSTM is similar to GRU, but instead of having an update and reset gates, LSTM utilises forget input and output gates, this structure makes LSTM more complex, but it also allows LSTM to handle larger dataset.

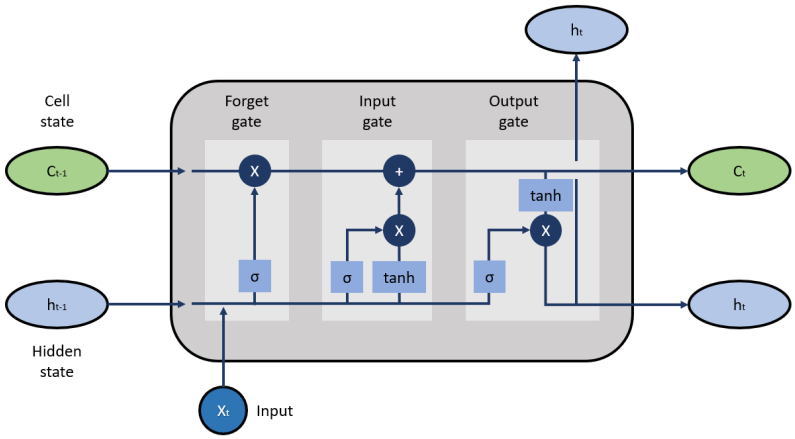

Figure 2.2 Long short-term memory cell (Fu et al. 2022)

Amongst the network-based models, the basic NN has seen the widest adoption (Donkor et al. 2014) in the field of water demand forecasting, owing to its versatility and performance. Whilst NN has shown superior performance compared to other widely tested models (Adamowski et al. 2012; Vijai and Bagavathi Sivakumar 2018), newer models such as GRU and LSTM can produce better forecasts in terms of accuracy (Guo and Liu 2018; Mu et al. 2020; Rahimzad et al. 2021). This thesis evaluates the basic NN and LSTM, the former offers comparability due to its wide adoption, whilst the latter may produce more accurate forecasts, due to its more complex structure.

More specifically, LSTM has proven to be the superior forecasting model compared not only to network-based models. For hourly demand, LSTM has been shown to outperform NN, RF, ARIMA, and Support vector regression (SUV) (Boudhaouia and Wira 2021; Mu et al. 2020; Nasser et al. 2020; Pu et al. 2023), it is only inferior to a hybrid method that incorporates LSTM (Pu et al. 2023). Nasser et al. (2020) have evaluated forecasting models on different aggregated dataset sizes, of two, 10 and 20 households; whilst all results have shown that LSTM is superior to SVR and RF, the writer has tested additional inputs, and each input addition have shown marginal accuracy increase for LSTM. Mu et al. (2020) and Pu et al. (2023) have both used 15-minute demand data, the former

aggravated it up to daily demand, whilst the latter aggravated the data to hourly demand. Mu et al. (2020) have shown that LSTM is superior to other models in all temporal resolutions; whilst Pu et al. (2023) have introduced a superior hybrid forecasting model that incorporates LSTM, but for both temporal resolutions, LSTM still performed better than other non-hybrid models evaluated.

2.2.3. Tree-based Models

Tree-based models have proven their forecast ability in the field of water demand forecasting (Chen et al. 2017; Herrera et al. 2010; Xenochristou and Kapelan 2020), though it has received less attention in the field of water demand forecasting when compared to network-based and time-series models.

Tree-based forecasting and classification models evolve from Decision Trees (DT) (Tso and Yau 2007). A DT is a tree-like model where leaves are considered final decisions, branch splits represent feature grouping that groups sample nearer to leaf nodes. A decision is made when all features or a predetermined number of features are evaluated.

The common tree-based models are Decision Tree (DT) (Tso and Yau 2007), Random Forest (RF) (Herrera et al. 2010), Gradient Boosting Tree (GBT) (Nie et al. 2021), and Extreme Gradient Boosting Decision Tree (XGB) (Xenochristou and Kapelan 2020). The DT is made up of one tree, and the samples and features can be exhaustively or selectively evaluated; though it makes quick and interpretable forecasts, it is very susceptible to noisy data. RF is made up of multiple DTs, it overcomes DT's noise susceptibility by taking an average of the forecasts from multiple trees, where each tree is trained using a subset of samples, making all individual trees unique; however, the improved noise susceptibility does come at the cost of lowered interpretability, as the average

outcome from multiple trees trained on subset samples are not as easily understandable as single DT.

GBT for XGB are similar concepts to RF, they all use multiple DTs to improve accuracy and noise susceptibility. RF make use of multiple DTs by training multiple trees in parallel; whereas GBT and XGB train multiple trees in series, the training is done in an iterative process to focus on the errors made by previous trees. Both GBT and XGB have proven to improve forecasting accuracy.

Since XGB's development by Chen and Guestrin (Chen and Guestrin 2016), it has shown successful applications in various competitions. XGB exhibit better computational efficiency compared to original GBT for its ability in parallel computation, approximate tree matching, effective handling of sparse data and improvement for central processing unit and memory.

XGB has demonstrated strong forecasting ability in the field of energy (Bassi et al. 2021; Lu et al. 2020; Sauer et al. 2022), streamflow and groundwater level forecasting (Ibrahem Ahmed Osman et al. 2021; Nie et al. 2021) as well as market demand forecasting (Gumus and Kiran 2017; Khaidem et al. 2016). Though to date, it has received limited attention in the field of water demand forecasting. Xenochristou and Kapelan (Xenochristou and Kapelan 2020) have shown XGB to have comparable forecasting ability to other individual machine learning forecasting models.

2.2.4. Ensemble Forecasting Models

Beyond forecasting with single models, ensemble forecasting models make use of multiple different models to achieve forecasts with higher accuracy (Sagi and

Rokach 2018). Ensemble models come in four different forms – data decomposition, data bagging, model boosting, and stacking.

Data decomposition is like the modular idea presented by Prophet, where non-linear time-series data is broken down into components, and forecasts are made for all individual components before recombining to give one output. The method has been applied in a variety of engineering fields and has shown promising results (Ali et al. 2020; Chu et al. 2020; Li et al. 2022b; Liu et al. 2019).

Data bagging is how RF relates to DT, and model boosting is how GBT and XGB relate to DT. The former use multiple simple forecasting models are trained using sub-sets of samples, whilst the latter iteratively forecasts prior model errors to improve accuracy. Bagged and boosted ensembles can also be built using ML models other than DT (Khwaja et al. 2020).

Finally, stacking is where multiple different models are used, either in parallel with different weighting options, in series in a boosted manner, or in a mixture of parallel and series formation (Bata et al. 2020; Grover et al. 2015; Lertpalangsunti et al. 1999; Li et al. 2022b; Ribeiro and dos Santos Coelho 2020). All stacking approaches aim to draw out individual model advantages, thus producing more accurate forecasts.

2.2.5. Model Comparison

This subsection compares the models used in this thesis. Whilst many models are discussed in subsections prior, only six are selected, compared, and evaluated in this thesis. The are many models excluded from non-ensemble forecasting classes, to evaluate the data-centric forecasting approach, forecasting models are not exhaustively investigated. For non-ensemble models,

two models are selected from each class, with one widely applied model that can act as a benchmark and another novel model that has shown superior performance. All ensemble models are excluded altogether because these primarily focus on model structure, rather than data efficiency.

In Table 2.1, six models that are used throughout the thesis are compared using four criteria, these include: 1) popularity – how widely applied the method is in the field of forecasting; 2) interpretability – how easily models results are to interpret; 3) speed – how quick the models are to generate forecasts; and 4) parameter – the models' dependence on parameter selection.

The comparisons are noted in terms of good, neutral, and bad, to indicate comparative advantages and disadvantages between models. For a model to have good across all features, it would be a model that is widely adopted, requires little parameter selection, is easily interpretable and has fast forecast output. The relative judgments are made based on the literature reviewed and experiments performed.

Table 2.1 Model feature comparison

Class	Model	Popularity	Interpretability	Speed	Parameter
Time-series models	ARIMA	Good	Good	Neutral	Neutral
	Prophet	Bad	Good	Neutral	Good
Network-based models	NN	Good	Bad	Bad	Bad
	LSTM	Neutral	Bad	Bad	Bad
Tree-based models	RF	Neural	Neutral	Good	Bad
	XGB	Bad	Bad	Good	Bad

2.3. Transfer Learning

Transfer learning (TL) implies techniques where ML knowledge is transferred from source data to target data. In research, the characteristics of the transfer have been thoroughly investigated. It has been shown to help improve data availability, reduce the negative impact of poor data, and reduce training or model set-up requirements.

TL applications can be categorised into three methods – relationship determination, data decomposition and partial model transfer. These methods will be individually discussed in the following sub-sections, along with the latest research that shows cases of each method.

2.3.1. Relationship Determination

The first TL method focuses on investigating the relationship between source and target data. This method makes minimal assumptions around the relationship between available source data and target data; the method calculates the relationship between source and target data and discards source datasets that are deemed too different to the target dataset. A broad spectrum of research fields has used the relationship determination method to improve the forecasts, these include generic time series data (Ye and Dai 2021), power load (Zhang and Luo 2015), energy consumption (Le et al. 2020), sublayer water absorption (Liu et al. 2020) and product sales (Karb et al. 2020).

Ye and Dai (2021) used Dynamic Time Warping and Jenson-Shannon Divergence to measure time series similarities. They have shown TL's ability to improve forecasting accuracy through the inclusion of additional selective external data. Zhang and Luo (2015) used the electric load data from four nearby cities, to investigate the impact TL has on load forecasting. They have applied TL

through source data optimisation, where external data is selected for incorporation based on the correlation coefficient between source and target data. They have emphasised the idea of negative knowledge transfer, the goal of their work is eliminating negative knowledge transfer by excluding source data that are highly dissimilar to target data. Both measures of similarity used by Ye and Dai (2021) and Zhang and Luo (2015) have proven successful in improving accuracy levels.

Le et al. (2020) and Liu et al. (2020) have used Clustering Algorithms to group multiple datasets; the proposed methods have been shown to reduce computation time and improve accuracy. The former used k-Means Clustering Algorithm to group smart building energy consumption, whilst the latter grouped different features within the source data. The grouping acts as a different form of similarity measure, to select groups of external data used in the source data training.

Liu et al. (2020) have used source and target data with known similarities, but different distributions. The source data has more labelled samples than the target. Whilst the source and target data are similar, their differences may result in negative knowledge transfer. Thus, joint distribution adaption is employed to transform source data in aiding sample availability for training.

Chen et al. (2021) have used transfer learning to fix missing data issues in a water quality prediction system. The method employed is like the approach used by Liu et al (Liu et al. 2020), but the employment of the concept of data similarity does not stop at the data pre-processing stage. The overall concept is that a series of weak learners would train using a mixed set of both source and target data. The weighting of samples that aid better target test data prediction

increases over increasing iteration count, and vice versa. The weighted average method is then employed to compute missing data.

2.3.2. Data Decomposition

The second TL method is data decomposition, this method assumes some existing similarity between source and target data. Xu and Meng (2020) have used this method to improve electric load forecasting, and Ribeiro et al. (2018) have used the method for cross-building energy forecasting.

Both Xu and Meng (2020) and Ribeiro et al. (2018) have removed trend and seasonal components from the source data, before transferring the source data knowledge to the target data. The final model then consists of trend and seasonality forecasting trained on target data, and the remaining features trained using both the source and target data. The results from Xu and Meng (2020) have shown improvements in accuracy and reduced the number of negative knowledge transfer occurrences compared to other TL methods. The result from Ribeiro et al. (2018) has improved accuracy in most cases and scenarios evaluated.

2.3.3. Partial Model Transfer

The third TL method is partial model transfer. This method assumes similarities between source and target data, it leverages this similarity to pre-train ML models on source data, and then refine the forecasting model with target data. This would minimise target data requirements and reduce computation costs. This method has shown successful application in the fields of energy prediction (Fan et al. 2020; Gao et al. 2020; Sarmas et al. 2022), ventilation forecasting (Chen et al. 2020), disaster relief (Zheng et al. 2022), flood prediction (Kimura et al. 2019) and long-term water quality prediction (Peng et al. 2022).

The application of the fine-tuning process always employs neural network-based models, as the network layers and training epochs can be easily sliced. Most of the literature reviewed has employed network layer slicing (Chen et al. 2020; Fan et al. 2020; Gao et al. 2020; Kimura et al. 2019; Sarmas et al. 2022; Ye and Dai 2018; Zheng et al. 2022). This method first pre-trains a model on abundant source data, then the model is retrained on target data to refine the parameters. The research varied on where the pre-training ended, and the retraining began. Chen et al. (2020) have illustrated this most distinctly in their research, using two datasets, the researchers have compared four forecasting styles:

- Trained on abundant source data.
- Trained on limited target data.
- Transferred model at the model tail (final layer trained on target).
- Transferred model at the model head (initial layer trained on target).

Other research has frozen pre-trained model parameters at different stages to achieve accuracy improvement.

Peng et al. (2022) have applied the concept of refining more dynamically. Instead of creating a cut-off between pre-training and re-training, the model switches between training using the source and target data after each epoch, and the parameters are continually updated to fit both datasets. The result showed that the TL method has produced more accurate forecasts across all scenarios, but the level of accuracy improvement correlates with the size of the history window size. Since a longer history window would reduce the number of available training samples, this becomes a bigger issue when the target training data is limited. Thus, the inclusion of source data for training would have a greater impact.

2.4. Explainable Machine Learning Approaches

Reducing unnecessary data requirements can help models more efficiently extract useful information. The first step to reducing input data requirements in water demand forecasting is knowing the forecast horizon, as different forecast horizons require different input types (Donkor et al. 2014). The data types used in water demand forecasting can be categorised into three groups, socioeconomic (population and pricing policy), climatic (temperature and rainfall) and past demand.

For short-term demand forecasting, univariate past demand data alone have shown to be sufficient in achieving highly accurate forecasting results (Awad and Zaid-Alkelani 2019; Bakker et al. 2013; Bata et al. 2020; Cutore et al. 2008; Guo et al. 2018; Tiwari et al. 2016). Whilst knowing the demand horizon can reduce the need for alternative inputs other than past demand, Guo et al. (2018) have further reduced input size by showing that continuous and additional past demand is not necessary for aiding high forecasting accuracy; the size of the training input data can be further reduced when expert knowledge is incorporated in both model development and data processing.

Whilst ML models consistently achieve highly accurate forecasts, their black-box nature makes it difficult for users to explain how these forecasts are generated or what features are important in making such forecasts (Fu et al. 2022). This makes it difficult to reduce input size without expert knowledge of both the data and model type, and it offers little explanation to boost confidence when results are presented to decision-makers. The problem is further exacerbated by models becoming increasingly more complex in a bid for higher accuracy. To overcome

this, research on explainability in ML models has recently become topical (Barredo Arrieta et al. 2020).

To our knowledge, explainability has not been considered in the field of water demand forecasting, though applications for some engineering problems such as beach water quality prediction (Li et al. 2022a) and energy and heating (Kim and Cho 2020, 2021; Kuzlu et al. 2020; Zdravković et al. 2022) have been attempted. Explainable machine learning models offer insights into specific forecasting results, thus providing ideas for potential accuracy and efficiency improvement. The result would also give decision-makers a greater degree of confidence when using the forecasting results.

A wide range of techniques has been developed for machine learning model explainability. Barredo Arrieta et al. (2020) recognised that some statistical models are explainable due to the simplicity of model structure. More complex machine learning models require post-hoc approaches to become explainable. The post-hoc approaches are split into model-agnostic and model-specific approaches. As the names suggest, the former works on all machine learning models and the latter works on specific models. Specifically, two popular model-agnostic approaches are evaluated in Chapter 5, these are – Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). Both approaches explain the ML model by estimating feature contributions. LIME does so on a sample-by-sample basis, and SHAP could do individual samples as well as a global overview.

The rest of this section presents the existing research that has employed LIME and SHAP, it also touches on the theories and calculations of how both

approaches estimate feature impact. The process of how both approaches is applied is detailed in Section 5.2.

2.4.1. Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations (LIME) (Garreau and von Luxburg 2020) is a post-hoc ML model explainer that focuses on individual predictions. LIME operates by training the local surrogate models, these are trained with the goal of approximating the prediction of the original black-box model, around the given sample. Using local surrogate models, input feature values are tweaked, and their impact on the output can be measured. The surrogate models only aim for accurate approximation locally, i.e., for a particular sample; but do not have to be accurate globally, i.e., for other samples. This local truthful approximation is called local fidelity.

Amongst the literature that employs LIME, only the works by Parmar et al. (2021) and Zdravković et al. (2022) have used LIME independently, whilst others (Adak et al. 2022; Kuzlu et al. 2020) have used LIME alongside SHAP analysis. The work by Parmar et al. (2021) used LIME to determine the causes of parking time around Delhi, India, it classed different parking periods into groups, making it a classification task. This research had 681 effective samples, and the relatively small sample size makes individual sample review and analysis possible. Though the work by Zdravković et al. (2022) was a regression task on heat demand forecasting, it only validated the application of LIME based on a sample feature impact without concluding any feature, as no cross-sample feature impact comparison was done.

The work by Kuzlu et al. (2020a) and Adak et al. (2022) used both LIME and SHAP, they have applied machine learning to classify the feedback from food

delivery service reviews, and detailed insight in each given sample is important, thus both LIME and SHAP are applied on a sample-to-sample basis, the global classification overview was not taken. Other works (Białek et al. 2022; Kuzlu et al. 2020; Li et al. 2022a) that have incorporated SHAP have all made use of its global contribution overview, whilst Kuzlu et al. (2020) and Białek et al. (2022) also included LIME, both have drawn more results from the global feature impact results from SHAP, with few individual samples compared to the limited SHAP result.

2.4.2. SHapley Additive exPlanations

SHapley Additive exPlanations (SHAP) is a machine learning model explainer that employs a game-theory approach, developed by Lundberg and Lee (2017). It outputs a measure of individual feature contribution in any ML model, both locally and globally. SHAP has received ample research attention. Whilst it has not been applied in water demand forecasting, it has proven useful in a variety of other fields, including energy (Acuña et al. 2018; Kuzlu et al. 2020; Moon et al. 2022; Wang et al. 2019; Zhang et al. 2020), finance (Azzuri et al. 2022; Bussmann et al. 2020; Sajja et al. 2021) and beach water quality (Li et al. 2022a).

For energy-related research, SHAP has been evaluated for forecasting energy (Kuzlu et al. 2020; Moon et al. 2022) – for both demand and production; profit and cost (Acuña et al. 2018; Wang et al. 2019) – for producer, marketer, and end users; as well as for power system emergency control (Zhang et al. 2020). For profit and cost related works, the researchers employed the game theory aspect of SHAP to optimise profit/cost apportioning. The research did not output SHAP's built-in visual analysis but rather used the approach for optimisation.

The application of SHAP for forecasting and emergency control has all presented SHAP visual analysis. For forecasting purposes, the SHAP analysis has identified features that impact electricity production in photovoltaic and buildings' electrical load; the results aim to increase the level of integration of smart technologies. The SHAP analysis for the emergency control system would improve trust and transparency in ML's decision-making, making it possible for ML's adoption when it comes to safety-related decision-making.

In finance, the feature impact comparisons uncovered by SHAP can help identify driven factors for a price change or potential cost savings. Bussmann et al. (2020) have used empirical borrowers' data, grouping them into risk categories based on financial characteristics using SHAP analysis. Sajja et al. (2021) have used SHAP in fashion retail data, to produce intuitively understandable results for all stakeholders, thus bridging competing goals. In addition to determining important features, as shown in other finance work that employs SHAP, Azzuri et al. (2022) have showcased the potential of using SHAP to analyse cooking oil prices, the researcher has exhaustively shown all possible visual results SHAP is equipped with.

The only water-related work that has incorporated SHAP is interpreting features for the prediction of beach water quality (Li et al. 2022a). The work performed an in-depth analysis of features that impact the beach quality of three locations. The resultant forecast aims to determine beach closure, due to potential poor water quality. SHAP has been found to aid the forecast by uncovering previously unexpected impact factors. Giving decision-makers greater confidence in not committing to false closures and evading potential economic loss.

Whilst many researchers have compared both LIME and SHAP, the works that have incorporated and compared the two have all drawn conclusions from SHAP global analysis. The result of local cases may be useful when the sample number is small, or when a certain case is of particular interest. The more common goal of having explainable AI would be to drive clarity for all samples in each model; this means that SHAP global analysis is more useful. However, if computation cost is of concern, LIME could be recognised as a fast-paced local alternative to SHAP. Kuzlu et al. (2020) have explicitly analysed the computation time difference between LIME and SHAP, and LIME is significantly faster (34.3 milliseconds compared to 9.4 minutes). When several local cases are of concern, LIME might be the better option as SHAP would still need to run fully even for one local sample analysis.

2.5. Machine Learning Performance Indicators

There are many measures to determine the performance of forecasting models. Some popular accuracy measures include Root Mean Squared Error (RMSE) (Bata et al. 2020; Chen and Boccelli 2018; Guo et al. 2018; Herrera et al. 2010; Liu et al. 2022; Tiwari and Adamowski 2013), Normalised Root Mean Squared Error (NRMSE) (Bata et al. 2020; Chen et al. 2017), Mean Squared Error (MSE) (Fullerton and Molina 2010), Mean Absolute Error (MAE) (Guo et al. 2018; Herrera et al. 2010; Tiwari and Adamowski 2013), Relative Error (RE) (Bakker et al. 2013) and Mean Absolute Percentage Error (MAPE) (Bakker et al. 2013; Bata et al. 2020; Chen et al. 2017; Chen and Boccelli 2018; Guo et al. 2018; Liu et al. 2022; Sardinha-Lourenço et al. 2018; Wang et al. 2020); some papers have respectively used the names of Relative RMSE (Bakker et al. 2013) and Average Absolute Relative Error (Adamowski 2008; Jain et al. 2000) in place of NRMSE and MAPE, the terms will be jointly discussed due to identical calculation.

Additionally, the Correlation Coefficient (r) (Chen et al. 2017; Chen and Boccelli 2018; Liu et al. 2022), Coefficient of Determination (R^2) (Adamowski 2008; Bakker et al. 2013; Chen and Boccelli 2018; Tiwari and Adamowski 2013; Wong et al. 2010) and Nash-Sutcliffe Efficiency Coefficient (NSE) (Gagliardi et al. 2017; Guo et al. 2018; Herrera et al. 2010) have also been applied in water-related fields, the latter two are identical in calculation, thus, they will be jointly presented and discussed.

The equations for error measures are shown below:

$$RMSE = \sqrt{\sum \frac{(y_o - y_f)^2}{n}} \tag{2.1}$$

$$NRMSE = \frac{RMSE}{y_o} \tag{2.2}$$

$$MSE = RMSE^2 \tag{2.3}$$

$$MAE = \sum \frac{|y_o - y_f|}{n} \tag{2.4}$$

$$RE = \sum \frac{|y_o - y_f|}{n\overline{y_o}} \tag{2.5}$$

$$MAPE = \sum \frac{|y_o - y_f|}{ny_o} \tag{2.6}$$

$$r = \frac{\sum (y_o - \overline{y_o})(y_f - \overline{y_f})}{\sqrt{\sum (y_o - \overline{y_o})^2 \sum (y_f - \overline{y_f})^2}} \tag{2.7}$$

$$R^2 = 1 - \sum \frac{(y_o - y_f)^2}{(y_o - \overline{y_o})^2} \tag{2.8}$$

where n is the number of samples, y_o is the observed values, y_f is the forecasted values, $\overline{y_o}$ and $\overline{y_f}$ respectively represent the mean observed and forecasted values.

Measures in Equations 2.1 to 2.6 all share similar numerators, either the difference between forecasted and observed values squared or the absolute difference between the two. Though the denominators differ, the denominators only use observed value statistics, thus it could be understood as different scaling to the numerators. From these, only one needs to be kept, as all correlate with each other. The values of these depend on the sample size and values, thus, these findings can only be compared with the same sample group.

Equations 2.7 and 2.8 respectively denotes the correlation coefficient and coefficient of determination, the two measures are related, where the coefficient of determination is the square of the correlation coefficient. Unlike previous measures, both *r* and R^2 have bounded outcomes. *r* is bounded between -1 and 1, it measures a negative correlation as well as a positive correlation. Kvalseth (1985) recommends the use of R^2 presented in Equation 2.8, the choice of R^2 has an upper limit of 1 and no lower limit; values closer to 1 indicate a closer fit, and the value of 0 indicates the same level of fit compared to mean observed values and negative values indicate worse fit compared to the mean, which renders any model with negative R^2 values statistically insignificant. As the correlation coefficient and coefficient of determination are correlated, only one is kept.

This work has chosen to use RMSE and R^2 to measure model accuracy. Although the two measures perfectly negatively correlate with each other, they

complement each other in what the values mean. Better RMSE can only be achieved with both high precision and no systematic error; in comparison, R^2 does not reflect systematic errors. Additionally, as RMSE is scaled by the observed values and sample size, it offers an indication for the confidence interval of the predicted values, in contrast, R^2 allows for cross-sample model comparison.

Besides accuracy measures, inequality measures are also used in Chapter 5 of this thesis, to measure how feature contributions spread. The Gini index (GI) (Frank A. Farris 2010) and Theil coefficient (Mookherjee and Shorrocks 1982) are commonly used inequality measures. The equations for the two indices are as below:

$$GI = \frac{\sum\sum\left|x_i - x_j\right|}{2n^2\bar{x}} \tag{2.9}$$

$$Theil = \frac{1}{n}\sum\left(\frac{x_i}{\bar{x}}\ln\frac{x_i}{\bar{x}}\right) \tag{2.10}$$

where x is the feature contribution, n is the total number of features, i and j are counters to iterate the list of available contribution values and $\bar{x}$ is the mean feature contribution.

Both indices measure inequality by how actual inequality is compared to perfect equality. GI is bounded between 0 and 1, where 0 is perfect equality, and 1 is perfect inequality. Theil only has a lower limit of 0, representing perfect equality, but no upper limit. Due to its bounded nature, GI is used in the thesis to determine feature contribution spread.

The chosen accuracy measures are thus R^2 and RMSE, these will be used in all methodological chapters. The additional inequality index GI will be used only for Chapter 5.

2.6. Summary of Research Challenge

This thesis aims to explore the potential of data-centric approaches. Most of the existing research focuses on model-centric approaches, where model alterations are explored to improve forecasting accuracy. As a result, there exists a large research gap surrounding data-centric approaches, where data usage efficiency can be improved to increase forecasting accuracy.

Existing research surrounding water demand forecasting has shown the importance of data collection. However, quality data may not always be readily available. In some research, expert knowledge has been incorporated to reduce input data amount, this would increase the number of usable samples within a given dataset. Chapter 5 in this thesis seeks to replace the expert knowledge with machine learning model explainers, so this level of input reduction can be replicated, without expertise in data and model pairing.

Additionally, the unavailability of prolonged data would also reduce the number of useful samples. Chapter 6 in this thesis uses transfer learning to mitigate this by incorporating external data.

Chapter 3 - Data

3.1. Introduction

This Chapter presents the datasets used throughout the thesis. The data includes real-world measured demand datasets from two sources. The first is a single set of hourly data from university student accommodation buildings in China, and the second is 20 sets of 15-minute data from the UK. The following sections in this chapter provide an overview and visual information on the Chinese (CHN) dataset and UK datasets, and how basic data cleaning is performed. The Chapter ends with Table 3.1 detailing statistical information of all datasets used.

3.2. Chinese Dataset

The first dataset is of hourly demand, collected from university accommodation buildings in China. The data was collected and saved into monthly data files corresponding to its measured month and year. The compiled dataset shows the total demand of around 20,000 individual students, for 10 months, starting from October 2013. The 10 months period incorporates one winter and one summer holiday, due to the annual non-repeating nature of the holidays within the dataset, any knowledge that can be extracted from training cannot be validated via testing, thus those holiday periods are excluded from all experiments. Additionally, the spring term data (the period between winter and summer holidays) is inundated with recording errors, possibly due to equipment deterioration. The overall used data from the Chinese dataset is thus restricted to 10 weeks from term one, between October and December, the cut-off was taken a couple of weeks before the start of the winter holiday in January, to avoid all holiday impacts.

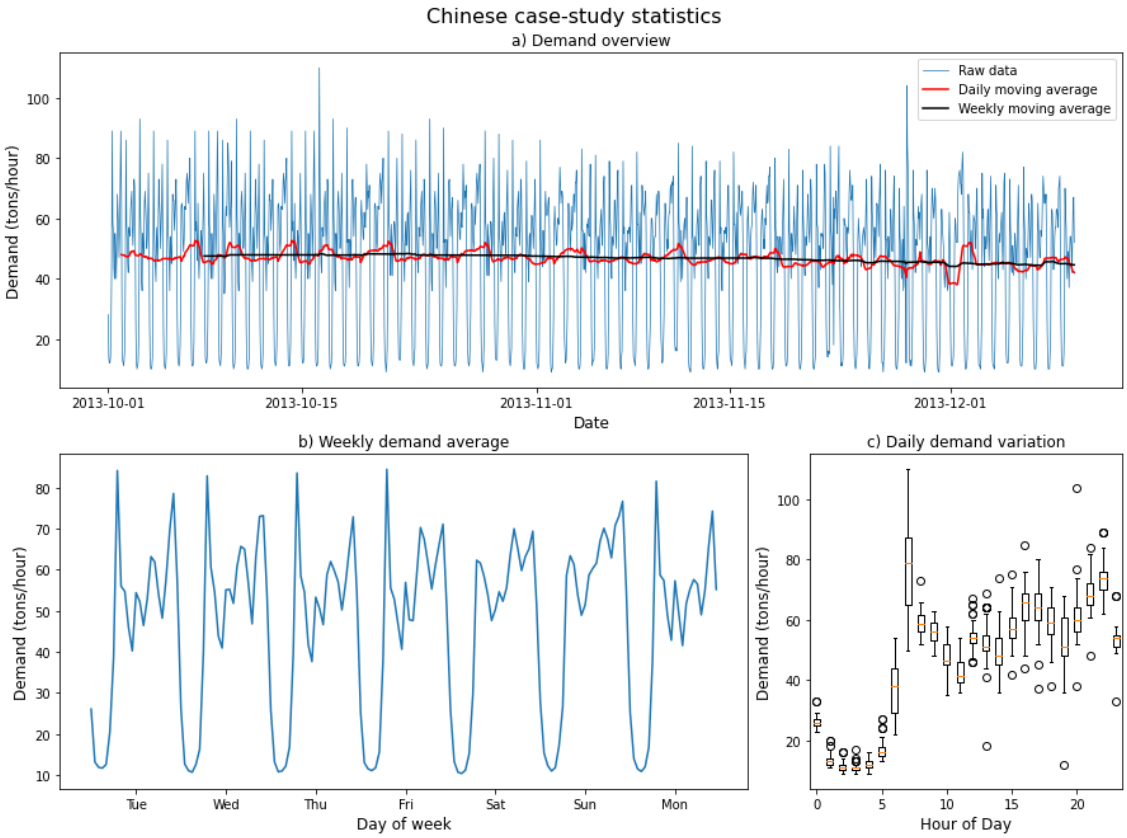

Figure 3.1 Raw data plots of CHN dataset, a) full demand data, with moving average of 24 and 168 hours, b) averaged weekly demand data, c) daily demand data with median and percentiles (25% and 75%)

Figure 3.1 presents an overview of the CHN data used. Figure 3.1a shows the overall used data period, with the addition of daily and weekly moving averages. The moving average lines show a slight downward trend, indicating slightly reduced water usage towards the winter months. Figure 3.1b shows the average day-of-the-week demand. The weekend usage shows a slightly different trend to weekday demand, and Mondays have a slightly smaller morning peak compared to other weekdays. Figure 3.1c shows daily demand variation. Morning peaks show the largest range of varying demand, while daily troughs show that the minimum demand remains consistent.

3.3. UK Dataset

The second dataset is from an unspecified UK water supply area. The entire dataset contains demands for 20 district metered areas (DMA). Of the 20 DMAs, two DMAs are completely excluded from any experiment, due to them having little to no continuous quality data. The remaining 18 DMAs are labelled UKn, where n is numbers 1 to 18.

From the 18 UK datasets, three (highlighted in Table 3.1) are used throughout the experiments for having the longest uncorrupted data recordings. The remaining DMAs from the UK are used for experiment three only, where data quantity mattered more than quality.

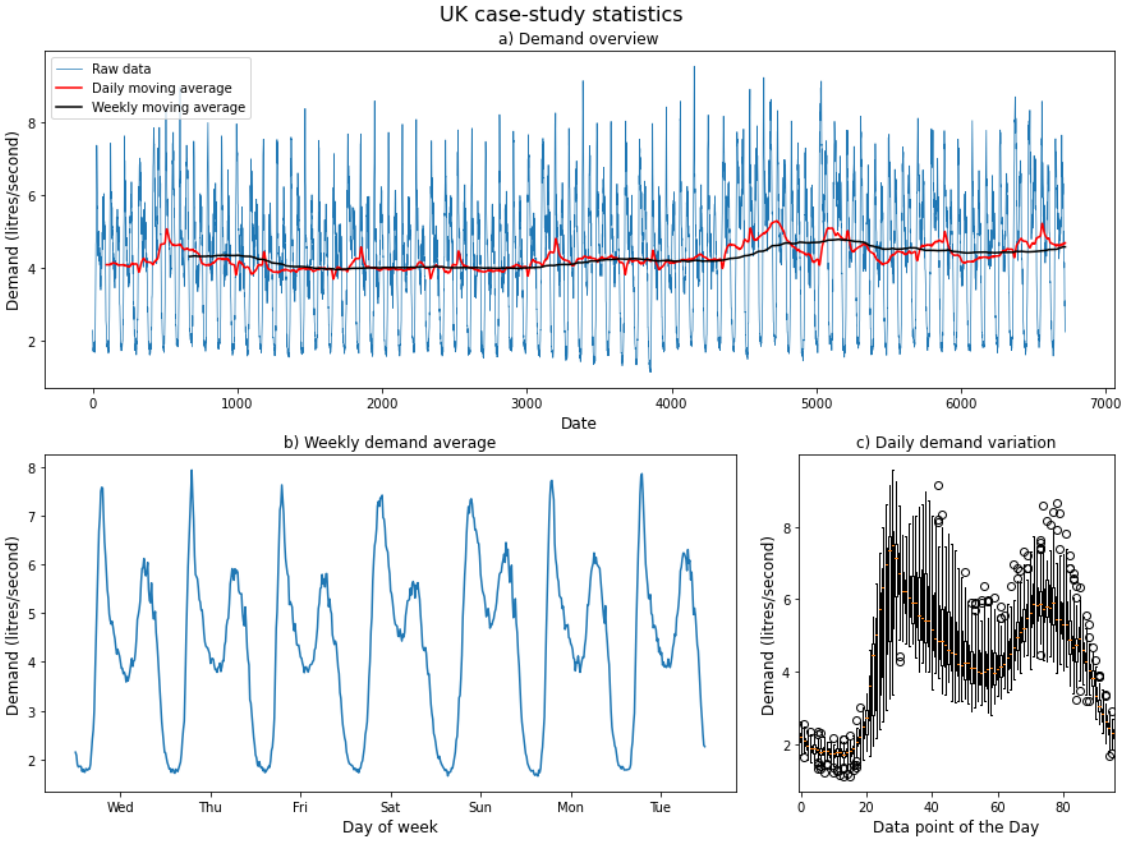

Figure 3.2 Raw data plots of UK dataset (UK11), a) full demand data, with moving average of 24 and 168 hours, b) averaged weekly demand data, c) daily demand data with median and percentiles (25% and 75%)

Figure 3.2 shows the longest uncorrupted period of UK11 demand (second highlighted DMA in Table 3.1). Like the CHN dataset, the UK demand also has consistent daily and weekly moving averages, as evidenced by Figure 3.2a. Figure 3.2b shows the average weekly demand to be more consistent, with weekend peaks appearing to be slightly less pronounced compared to weekdays. In terms of daily demand variation, the range falls into two groups, where the demand range is identically high during the day, and identically low at night.

3.4. Data Cleaning

The datasets used throughout the thesis are cleaned in two ways, 1) quality data extraction, and 2) corrupted data exclusion. The different method is applied in different chapters.

For Chapters 4 and 5, quality data is extracted, as data quality is paramount. For the CHN data used in Chapter 4, the first 10 weeks are extracted, as the remaining contains non-repeating holidays and corrupted recordings. Similarly, the first 30 weeks of data are extracted in UK4, UK11 and UK12, for experiments in Chapters 4 and 5. The periods extracted contain the longest uninterrupted uncorrupted recordings.

For Chapter 6, data quantity is favoured over quality. For this reason, corrupted recordings in the UK datasets are excluded based on visual inspection. Figure 3.3 presents an example, using data from UK7. Figure 3.3a shows the dataset before data cleaning, there is a peak around the 10000^{th} point and a trough at around the 25000^{th} point. The peak and trough appear to be singular based on initial visual inspection, but detailed inspection reveals that the peaks and troughs are continuous. Basic exclusions are performed on all UK datasets, where a maximum and minimum range is set per DMA based on visual inspection, the

minimum is set to zero for all DMAs, and the maximum allowed per DMA is shown in the Cap column in Table 3.1. Any value beyond the allowed range is set to zero. When training samples are formed in later experiments, samples with zeroes are excluded.

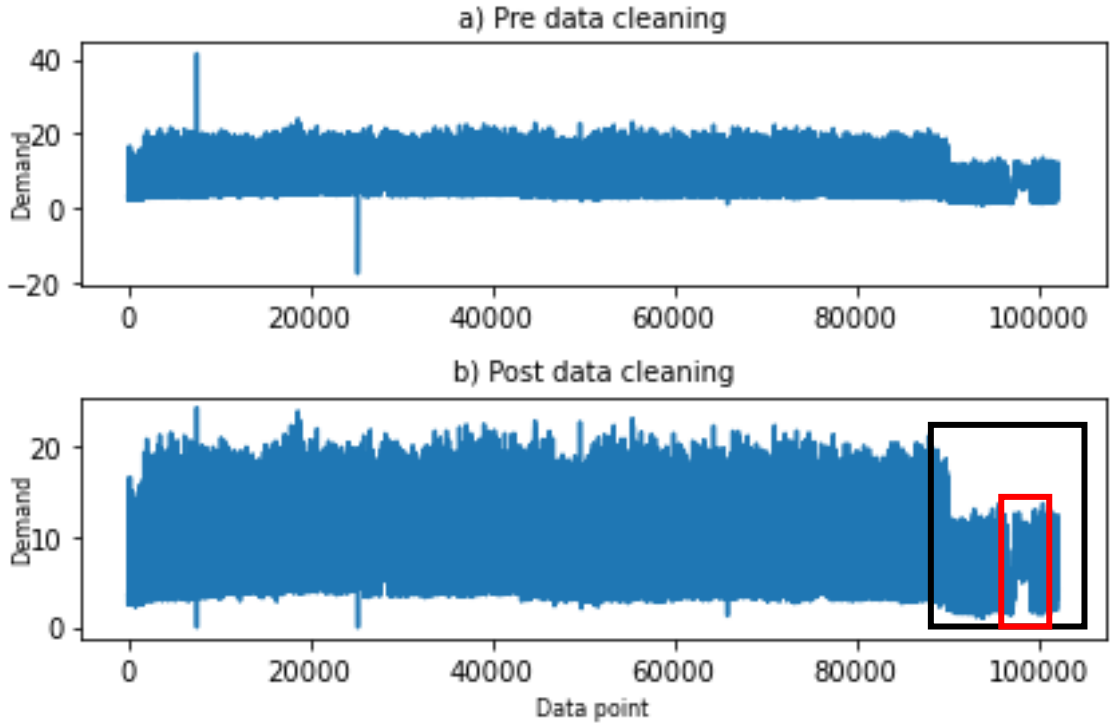

Figure 3.3 Data cleaning example (UK7)

3.5. Data Statistics

Table 3.1 shows basic statistical information regarding the datasets used. The user recording differs for CHN and UK data, where the former shows the total number of estimated individuals of 20,000, and the latter shows the number of properties supplied, between about 300 and nearly 2,000. The highlighted data are of DMA recordings that contain the longest uncorrupted data, and these datasets are used in all experiments. Finally, the maximum and minimum recordings shown are of post-cleaning whilst excluding zeroes. The minimum recordings of zeroes shown in Table 3.1 are from rounding, where all recordings are presented to 2 decimal places.

Table 3.1 Basic statistical information of the DMA data used. (Highlighted datasets contain the longest uncorrupted data, it is used in all experiments)

Data	Users	Start date	Data length (Points/Days)	Cap	Max* (l/s)	Min* (l/s)
CHN	20,000	2013-10-01	7296 / 304	N/A	110	3
UK1	1048	2016-07-01	28512 / 297	25	22.7	0.01
UK2	683	2016-07-01	26496 / 276	15	13.51	0.92
UK3	961	2016-06-01	31056 / 323	20	19.36	2.67
UK4	**461**	**2016-06-01**	**30382 / 316**	**20**	**16.74**	**0.14**
UK5	673	2016-06-01	27840 / 290	20	19.07	1.94
UK6	1243	2016-06-01	27168 / 283	35	34.89	0.7
UK7	965	2014-03-31	102240 / 1065	25	24.43	1.04
UK8	485	2016-07-01	26304 / 274	25	20.08	1.8
UK9	169	2016-06-01	28512 / 297	25	10.12	0.26
UK10	550	2014-03-31	103991 / 1083	14	12.68	0
UK11	**351**	**2016-06-01**	**28512 / 297**	**25**	**9.56**	**1.04**
UK12	**669**	**2016-07-01**	**26304 / 274**	**25**	**12.44**	**1.2**
UK13	1861	2014-03-31	105408 / 1098	25	23.44	3.26
UK14	291	2014-03-31	103296 / 1076	14	12.95	0.03
UK15	948	2014-03-31	103680 / 1080	30	23.85	0
UK16	266	2014-03-31	72335 / 753	14	10.09	0
UK17	972	2014-03-31	73006 / 760	40	39.7	0.01
UK18	313	2014-03-31	74349 / 774	30	13.24	0.94

Chapter 4 - Short-term Water Demand Forecasting Using Data-centric Machine Learning Approaches

4.1. Introduction

Much of the effort aiming to improve the accuracy of water demand forecasting has been using model-centric approaches (Adamowski et al. 2012; Chen et al. 2017; Chen and Boccelli 2018; Gagliardi et al. 2017; Herrera et al. 2010; Lertpalangsunti et al. 1999; Liu et al. 2022; Sardinha-Lourenço et al. 2018). These approaches focus on developing and adapting models to data, through various approaches including parameter optimisation, alterations to model structure and ensemble models.

In contrast, data-centric machine learning approaches have received limited attention in the field of time series forecasting including short-term water demand forecasting (Fu et al. 2022). This Chapter evaluates the potential of data-centric approaches, using four machine learning forecasting models, and compares the result to model-centric approaches. The forecasting models used are Autoregressive Integrated Moving Average (ARIMA) (Box E. P. G. et al. 2015), Neural Network (NN) (McCulloch and Pitts 1943), Random Forest (RF) (Breiman 2001) and Prophet (Taylor and Letham 2018). The background and comparative works used in the models are presented in the Literature Review in Chapter 2. The implementation of the models for this chapter is presented in Section 4.2.

The final sections present the results and summary. The findings have shown that 1) data-centric machine learning approaches offer promise for improving

forecast accuracy of short-term water demands; 2) accurate forecasts are possible with short training data; 3) RF and NN models are superior at forecasting high temporal resolution data; and 4) data quality improvements can achieve a level of accuracy increase comparable to model-centric machine learning approaches.

4.2. Methodology

This section details the implementation of the four forecasting models, and how the experiments are set up.

4.2.1. Autoregressive Integrated Moving Average

Autoregressive Integrated Moving Average (ARIMA) is a statistical model for time series, developed by Box and Jenkins in 1970 (Box E. P. G. et al. 2015). ARIMA combines autoregressive (AR) and moving average (MA) models with a built-in differencing term.

The AR model assumes that the current state of a time series depends linearly on its past states plus error, and the MA model assumes that the current state of a time series depends linearly on its current and past errors. The combination of the two models is known as the ARMA model:

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \ldots - \theta_q \varepsilon_{t-q} \qquad (4.1)$$

where y_t is the information state at different time t, ε is error, ϕ and θ are autoregressive and moving average parameters, p and q are the total number of autoregressive and moving average terms. Equation 4.1 can be simplified as:

$$\phi_p(B^p) y_t = \theta_q(B^q) \varepsilon_t \qquad (4.2)$$

where B is the backshift operator, it shifts y and ε backwards in the temporal space.

ARMA models only work on stationary data, where stationarity is defined by data having constant mean and variance. Non-stationary data can become stationary by differencing data points. This differencing function can be integrated into the ARMA model, and the result is known as an ARIMA model:

$$\phi_p(B^p)\Delta^d y_t = \theta_q(B^q)\varepsilon_t \qquad (4.3)$$

where Δ is the differencing factor, and d is the degree of difference. This integrated component represents the 'I' in ARIMA, is models the series as the difference between the current and previous value, this difference can be modelled multiple times based on the degree of difference required to achieve stationarity.

The basic ARIMA model is presented in the form of ARIMA(*p,d,q*), where p, d and q respectively represent the number of past data points, the order of differencing and the total number of current and past error terms.

For data with a strong sense of seasonality or trend, seasonal ARIMA (SARIMA) can be employed, and the parameters are expanded to include seasonal factors:

$$\Phi_P(B^{sP})\phi_p(B^p)\Delta_s^D\Delta^d y_t = \Theta_Q(B^{sQ})\theta_q(B^q)\varepsilon_t \qquad (4.4)$$

where S is the period of a known seasonality, and P, D, and Q respectively represent the autoregressive, differentiation and moving average terms of the seasonality, like their respective lowercase counterparts. The SARIMA model is presented in the form of SARIMA$(p,d,q)(P,D,Q)_s$.

The seven parameters for SARIMA can be estimated through a series of visual and statistical tests. An initial autocorrelation and partial autocorrelation plot for the case studies confirm the strong daily data periodicity, thus daily seasonal factor is chosen for all case studies. To eliminate the seasonality, the seasonal differencing factor D is chosen to be 1. The visual analysis shows that there is strong stability in the daily moving average in all case studies, suggesting that the seasonal autoregressive and moving average factors P and Q are not necessary, thus both are 0.

Once the seasonal parameters are determined, the lower-case parameters can be estimated. Data stationarity first needs to be confirmed by reviewing the presence of unit roots in data (Gupta et al. 2009). The respective Augmented Dicky-Fuller (ADF) values for the four case studies after seasonal differencing are -11.7, -20.6, -22.2 and -19.4, all values are well below the ADF critical value of -3.4 at 1%. The p-values are 0 for all case studies. The ADF test shows that all four case studies are stationary, and thus d is estimated to be 0. The autoregressive and moving average factors can be estimated by data autocorrelation and partial autocorrelation plots. The plots suggest that the p and q values should be 3 and 0, respectively, for all case studies.

The SARIMA $(3,0,0)(0,1,0)_{24/96}$ model is tested in Python using the 'SARIMAX' from the 'statsmodels' library (Seabold and Perktold 2010), the estimated the p and q terms will be further tested through grid search using the Chinese (CHN) dataset to confirm these are the optimal pair.

4.2.2. Prophet

Prophet is a modular regression model for time series forecasting, developed by the Facebook research team (Taylor and Letham 2018). In their original paper,

the number of events created on Facebook is used as source data to compare the forecast performance of Prophet, ARIMA, Exponential Smoothing and Random Walk. The results have shown that Prophet is superior to all other models tested.

Prophet works by decomposing any given time series into three main components, including trend, seasonality, and holiday effects:

$$y_t = G_t + S_t + H_t + \varepsilon_t \tag{4.5}$$

where y is the demand, G is the trend, S is the seasonality, H is the holiday effect and ε is the error associated with each time step.

The trend can be modelled as a linear function (Equation 4.6) or a non-linear saturating growth (Equation 4.7):

$$G_t = (k + a_t\delta)(t + (m + a_t\gamma)) \tag{4.6}$$

$$G_t = \frac{C_t}{1 + e^{-(k+a_t\delta)(t-(m+a_t\gamma))}} \tag{4.7}$$

where k is the growth rate, a is a binary value indicating the presence of the effect from the change point t, δ is the change rate adjustment, m is the offset parameter and γ is the continuation factor. The nonlinear saturating growth model of trend is an extension of the linear trend with the addition of a carrying capacity C.

The seasonality is modelled using Fourier Series. It is incorporated as an additive component, but can be modified to be a multiplicative component through the log transformation of the original data as below:

$$S_t = \sum_{n+1}^{N} (a_n \cos \frac{2\pi nt}{P} + b_n \sin \frac{2\pi nt}{P}) \quad (4.8)$$

where P is the regular data period and the choice of N for different periods is automatically selected through the built-in selection procedure.

The holiday components are fitted as lists of dates with predictable changes. The dates for recurring events without regular periods are given as lists, and each holiday is given a parameter to signal its effect. The chosen case studies are all without holiday impacts, thus no date is given.

In terms of input parameters, Prophet can make forecasts without any parameter inputs. However, the key parameter - the number of change points and its scale - will be tested through grid search using the CHN dataset to determine its impact and the optimal pairing. The Prophet package is available in both R and Python, and the Python Prophet package (Taylor and Letham 2018) is used in this research.

4.2.3. Neural Networks

Neural Networks (NNs) and their variations have been widely applied to water demand forecasting (Guo et al. 2018; Tiwari and Adamowski 2013). The most common NN consists of three layers and is trained through iterations of feed forward and back propagation processes.

The three layers structure consists of an input layer, a hidden layer, and an output layer; each layer consists of a set number of neurons and each layer is connected to the subsequent layer via a transfer function:

$$h_j = f(\sum_{i=0}^{n_0} w_{ij} x_i + w_{0j}) \quad (4.9)$$

where h_j represent neurons of the middle or output layer, x_i is the neuron of the previous layer, w_{ij} and w_{0j} are connecting weights and bias between h_j and x_i, while f signifies the transfer function between the layers.

The NN model used in this research is a three-layer feed-forward model with backpropagation. The model is applied using 'MLPRegressor' from the 'sklearn' library in Python (Pedregosa et al. 2012). Available numerical parameters are all investigated using the CHN dataset, to determine optimal parameter settings for further experiments.

4.2.4. Random Forests

A Random Forest (RF) model is formed by combining multiple tree predictors and it can perform classification and regression predictions (Breiman 2001). When RF is applied to regression tasks, the result is the mean output amongst all trees in the forest. Individual trees differ from each other because of the bootstrap sampling process. A forest of multiple trees can reduce overfitting, and is less prone to noise, due to the Law of Large Numbers.

Because of the bootstrap sampling process, each tree predictor is trained on a unique subset of data, thus predicting different results from each other. Individual trees are grown through an iterative node splitting process, each node split divides samples (bootstrapped subset) into two regions. The goal of each node split is to minimise the errors (E) in the resultant binary regions, and the error can be calculated as below:

$$E = \sum_{y \in R_1} (y - \hat{y}_{R_1})^2 + \sum_{y \in R_2} (y - \hat{y}_{R_2})^2 \quad (4.10)$$

where R_1 and R_2 correspond to individual binary regions after a node split; y are all present feature values within each corresponding binary region; $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are the mean feature values in the corresponding binary region. The order of features selected for node split is based on the features' impact on E. The node-splitting process is repeated until all features are used or until a pre-determined condition is met.

The RF model used in this research is the 'RandomForestRegressor' from the 'sklearn' library within Python (Pedregosa et al. 2012). Available numerical parameters are first investigated using the CHN dataset, to determine optimal parameter settings for further experiments.

4.2.5. Experimental Set-up

To evaluate the practicality of data-centric machine learning approaches, four experiments are designed to determine different aspects of its performance.

The first experiment aims to establish the effect of basic model-centric approaches on forecast accuracy, which addresses the first research question. This is done by evaluating model parameters available for tuning. Prophet and Seasonal ARIMA have a limited number of parameters available for tuning – Prophet has four parameters, but three overlap with each other, thus, only two parameters warrant investigation. Seasonal ARIMA has seven parameters, but the seasonality, differencing, and seasonal differencing factors are fixed, and the seasonal AR and MA factors are 0 as all case studies have a near-constant moving average, thus only two parameters warrant investigations. In comparison, NN and RF have multiple tuneable numerical parameters. Therefore, all

numerical parameters for NN and RF are first tested individually, using the Chinese dataset; from these results, two crucial parameters are selected for having the most significant effect on forecast accuracy. As this paper focuses on data-centric approaches, only two parameters are selected for each model for sampling analysis to demonstrate the effect of the basic model-centric machine-learning approach.

The two selected parameters for NN and RF are carried forward and further investigated through sampling, along with two parameters each from Prophet and SARIMA, using the case of the CHN dataset and data from UK11, UK11 is chosen as the representative UK dataset due to it having the longest uncorrupted demand recording following a visual inspection.

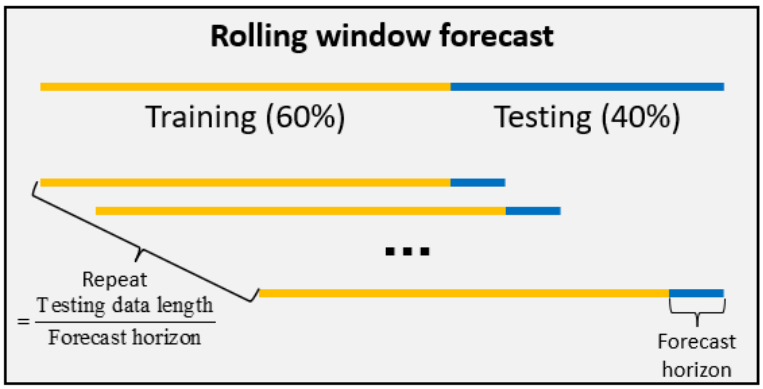

Figure 4.1 Illustration of rolling window forecast

To ensure the same training conditions for all models evaluated, a moving window is employed to move along the training and forecasting data, based on a given forecast horizon. Figure 4.1 demonstrates how the rolling window method is implemented. First, the data is split up into training and testing sections, where the 60% is reserved for training only, and the rest for testing. The reason for the split size is that 60% of data is sufficient for repeated training of sub-hourly data with daily forecast horizon, this allows for more data (40%) to be used to evaluate

model performance. Once the data is split, the 60% training data is used to train models to forecast the established forecast horizon. The training data then moves forward by the length of the forecast horizon to retrain the model multiple times, until the entire testing section is forecasted. The accuracy is calculated by how well the combined forecasted section match the testing section.

The forecast horizon for experiment one is set to one day, that is 24 points for the CHN dataset and 96 for the UK dataset, and the number of repeated trainings is 28 and 84 for CHN and UK dataset, respectively. Whilst Prophet and SARIMA use the continuous 60% training data for each training, NN and RF use the same training data to form samples of input and output pairs, the input of each sample is a full day's demand, that is 24 or 96 features for CHN and UK dataset, respectively.

The second and third experiments aim to establish the effect training data length and data resolution have on the forecast accuracy of different models, which are related to the second research question. This could eliminate the need for potentially large training data sets and define optimal model choice based on data type and accuracy requirement.

For experiments two and three, the setup used is the same as for experiment one. The total data length used is 10 weeks for CHN data and 30 weeks for others; the forecast horizon is one day; the testing period is 40% of the total data; and lastly, a moving window is used to maintain consistent training length for each forecast.

Experiment two will investigate the effect of increasing the training data length. Instead of the 60% of total data for training, the training data is reduced, whilst

other conditions remain the same. Starting at two days, with an increment of one day each time, up to 28 days of training data. This will determine if forecast accuracy correlates with training data length, or if less can be beneficial. Experiment two will be applied to the CHN dataset and the three highlighted (in Table 3.1) UK datasets, the three chosen DMAs have the longest uncorrupted demand recording amongst all UK DMA data.

Experiment three will investigate the effect of data resolution. This experiment is only applied to the highlighted UK datasets, as the CHN dataset involves lower-resolution data. For the case studies tested, the data is aggregated using 30-, 60- and 120-minute-long steps by taking the sum of the raw data, at the required number of steps. Forecasts are made for each new dataset, and the average forecast accuracy is compared with the original data. Though the coefficient of determination is unitless, and can be compared across all resolutions, the RMSE accuracy needs to be divided by number of points summed, as the raw data has unit of litres per second, the summed data would be litres per n seconds, where n is the number of data points summed.

The final experiment aims to determine how well each model tolerates noise; this will address the third research question. A scaled Gaussian noise is added to the training data, the scale is set to be between 0 and 50% of the average data value. The number of forecasts made is significantly higher for this experiment because, 1) each model is repeated for each noise level, and 2) each noise level is repeated ten times, due to the random nature of the added noise. To reduce the computation time, the forecast horizon is increased to seven days, and the noise scale increment is 5% for the case studies. The accuracy reduction resulting from prolonging the forecast horizon can be overlooked, as the focus of this

experiment is to compare how each model tolerates noise, the forecast with no noise therefore can be used as a reference point for each model.

Table 4.1 Overview of the experimental set-up

Experiment	Data	Forecast horizon	Variable factor
1	CHN and UK11	1 day	Parameters
2	CHN and UK4, 11, 12	1 day	Training length
3	UK4, 11, 12	1 day	Data resolution
4	CHN and UK4, 11, 12	7 days	Noise

4.3. Results and Discussion

4.3.1. Parameter Analysis

Using the Chinese dataset, the numerical parameters for RF and NN are considered for evaluation. The choice of evaluated numerical parameter range is designed to go up to or around the default values, to determine various parameters impact on accuracy. Based on this result, the two parameters with the most significant effect on accuracy will be selected for detailed sampling analysis.

As Prophet and ARIMA have only two core parameters each, they can be subject to sampling without initial parameter analysis.

Table 4.2 Random Forest initial parameters test parameters

Parameter	Default	Min	Interval	Max
n_estimators	100	1	200	1001
max_depth	24	2	20	102
min_samples_split	2	2	40	202

min_samples_leaf	1	1	20	101
min_weight_fraction_leaf	0	0	0.002	0.01
min_impurity_decrease	0	0	20	100
max_leaf_nodes	Unlimited	2	40	202
ccp_alpha	0	0	20	100
max_features	24	2	4	22
max_samples	984	84	180	984

Figure 4.2 shows the effect different numerical parameters have on forecast accuracy for NN and RF. In all figures, the y-axis shows the forecast accuracy, and the x-axis shows the selected parameter orders, detailed selection of parameter scale and values is provided in Tables 4.2 and 4.3.

Table 4.3 Neural Network initial parameters test parameters

Parameter	Default	Min	Interval	Max
hidden_layer_sizes	100	1	20	101
alpha	0.0001	0.0005	0.0005	0.003
batch_size	200	10	100	510
learning_rate_init	0.001	0.0005	0.0005	0.003
power_t	0.5	0.1	0.2	1.1
max_iter	200	10	100	510
beta_1	0.9	0.74	0.05	0.99
beta_2	0.999	0.974	0.05	1.224
epsilon	1e-08	(1e-8)/4	(1e-8)/4	(3e-8)/2

From Figure 4.2, the top panels show accuracy results for NN, and the bottom for RF; the left is R^2 accuracy, and the right is RMSE accuracy levels. The y-axis of the R^2 accuracy result is limited between 0 and 1, as beyond these limits is either impossible or insignificant. The examinable feature results from the right (R^2) and comparable result from the left (RMSE) are all in agreement, where the R^2 and RMSE result negatively correlate with each other. This suggests that poor-performing results are both less accurate in terms of correlation and bias and residual.

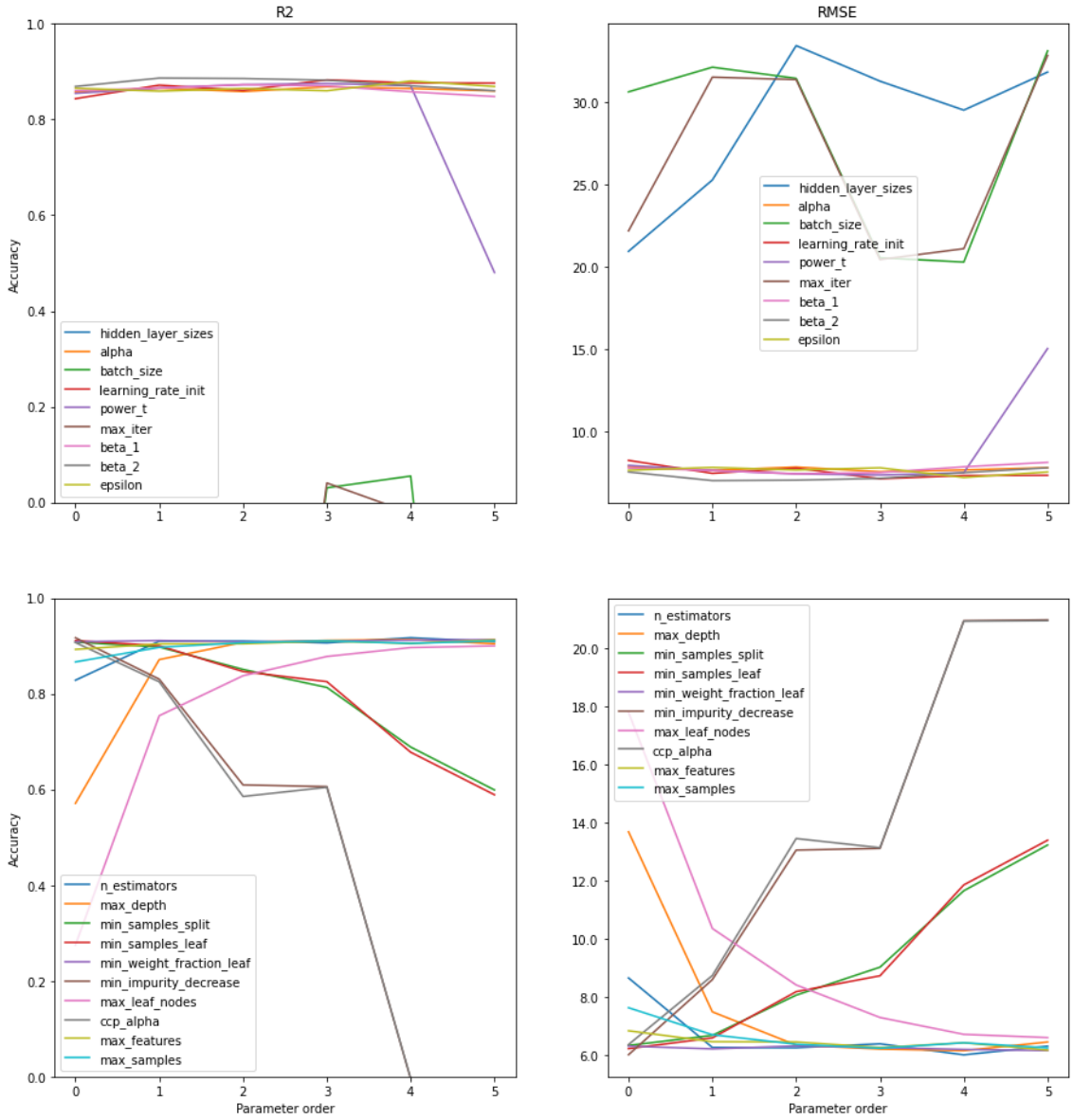

Figure 4.2 Initial parameter analysis of all available numerical parameters for NN and RF (parameter orders given in Tables 4.2 and 4.3)

The list of numerical parameters analysed in Figure 4.2 can be separated into two categories – model complexity and early stopping mechanism. Most parameters produce the highest accuracy with default parameter values, only the parameter that relates to model complexity (maximum feature for RF and hidden layer size for NN) varies significantly around default values, suggesting a need for further investigation. Along with this, the maximum depth for RF and maximum iteration for NN are also selected as the representative early stopping mechanisms. These two parameters are chosen because 1) these two parameters did not peak at the default value, as seen in Figure 4.2; and 2) compared to other early stopping mechanisms, these two parameters are comprehensible.

Table 4.4 Parameter choice and analysed values for different models

Model	Parameter	Chinese dataset	UK dataset
Prophet	Number of change points	1,2,3,6,42	1,2,3,18,126
	Changepoint prior scale	0.0005, 0.005, 0.05, 0.5, 5.0	0.0005, 0.005, 0.05, 0.5, 5.0
ARIMA	p	0, 1, 2, 3, 4	0, 1, 2, 3, 4
	q	0, 1, 2, 3, 4	0, 1, 2, 3, 4
RF	Maximum feature	1, 6, 12, 18, 24	1, 24, 48, 72, 96
	Minimum sample split	2, 4, 6, 8, 10	4, 13, 22, 31, 40
NN	Hidden layer node count	12, 24, 48, 96, 192	24, 48, 96, 192, 384

Maximum iteration	50, 100, 200, 400, 800	50, 100, 200, 400, 800

From the results of the initial parameter analysis, the selected parameters for sampling analysis for NN are the hidden layer size and maximum iteration, and for RF the maximum feature count and maximum depth. The main parameters for sampling analysis for ARIMA are p and q coefficients relating to the moving average parameter and autoregressive parameter, and for Prophet are the number of change points and the change point scale.

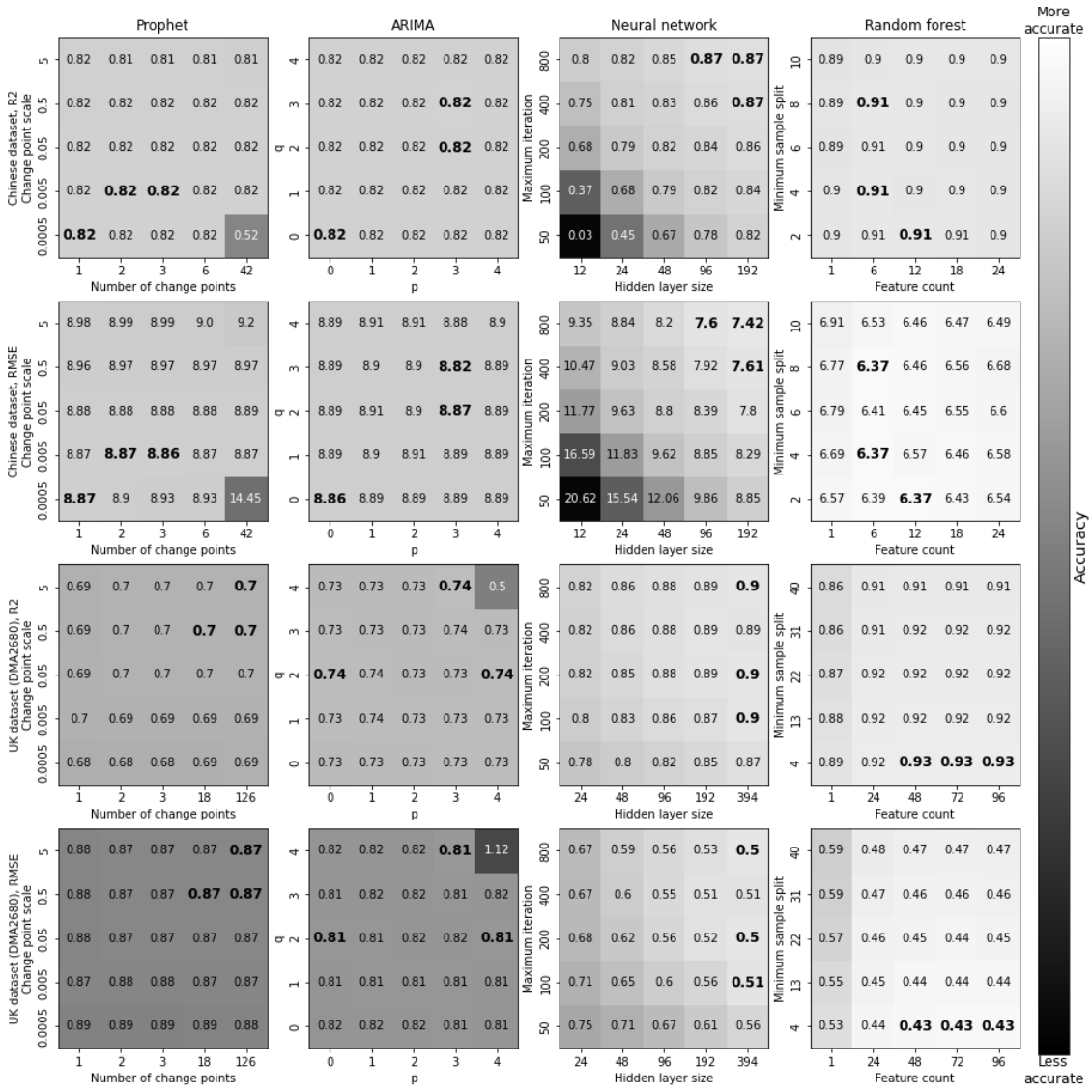

Figure 4.3 Sampling analysis for CHN data and UK11, the shading corresponds to forecast accuracy, the squares with bold texts are the most accurate forecasts

Table 4.4 shows the selected sampling parameters and the chosen investigation range for the two case studies (as UK datasets are similar in resolution and data length, thus investigation on parameter choice is only performed on UK11).

There are several differences between the ranges of selected parameters for the two case studies: 1) the number of change points for Prophet is lower for the CHN dataset because the data record is shorter; the two maximum numbers of change points are set to be the number of days and weeks within the training data, setting the change points to be intuitively understood; 2) both parameters for RF and the hidden layer size for NN are lower for the Chinese dataset because the size of these parameters correlates with the input of each training model, whilst both models consider a full day's data as input, the number of points in a day in the Chinese dataset is 24, compared to 96 in UK datasets.

Figure 4.3 shows the accuracy results for parameter sampling for the CHN dataset and UK11 dataset. The heading above each figure in the top panels indicates the model used across the column, and the left headings for each figure in the left panels indicate the case study and accuracy measure across the row. The axis headings and values for each panel show the feature and values sampled, as detailed in Table 4.4. The shading corresponds with accuracy levels, where lighter colour corresponds with higher accuracy and vice versa. The text in each panel shows the accuracy values, rounded to two decimal places. Each panel also has three highlighted (bold) values, these correspond with the three most accurate forecasts within each panel. Though there appears to be more than three of the same high accuracy value, this is the result of rounding to two decimal places, all accuracy values differ from each other if not rounded.

The sampling analysis finds that R^2 and RMSE show agreeable findings, where the parameter pair that produces higher R^2 accuracy also produced lower RMSE accuracy, this suggests that forecasts are accurate both in terms of correlation and bias.

The panels in the first and second columns show that for short-term water demand data, Prophet and ARIMA produce consistent forecasts, independent of parameter pairings. Only extreme parameter choices have a slightly negative impact on accuracy for these models. RF too is not overly dependent on parameter choice, though a feature count equal to half of the features available, and a small minimum sample split tend to produce slightly better results. The parameters in NN play a more significant role, where the higher computational complexity results in more accurate results, but the accuracy plateaus for all case studies when the maximum iteration is 800 and the hidden layer size is more than double the number of features.

The parameter analysis results from the two case studies show that the model-centric approach of optimising parameters for datasets has a limited effect, or when it does, the optimal values can be generalised, this holds for stable data such as short-term water demand.

4.3.2. Training Data Length Analysis

Whilst there are sufficient quality data available in all case studies analysed, this may not be the case for all real-life forecasting situations. Experiment two aims to establish a baseline for data required for each model to make a sufficiently accurate forecast.

The parameter values used are selected from experiment one. The effect of training data length can be visualised by varying the amount of training data used, starting at two days with an increment of one day each time, up until 28 days. This is applied to the CHN dataset and three UK datasets.

The experiments are repeated 10 times each for RF and NN, as these models are initialised with random weights. The repeats aim to identify and exclude outliers. With the 10 repeated results, a boxplot is drawn for RF and NN to show both the accuracy increase and variance decrease in response to the increased training length. Prophet and ARIMA achieve the same forecasts with the same parameters, thus they often overlap for each training set.

Figure 4.4 shows how the forecast accuracy reacts to reduced training data length for the CHN and three UK datasets. The x-axis represents the length of data used for training, measured in days, and the y-axis represents the forecast accuracies (left panels for R^2 and right for RMSE). The grey line in all figures corresponds to the accuracy level achieved by a naïve method, where all demands in the forecast period are assumed to be equal to the demand from the same time on the previous day. It needs to be noted that ARIMA has similar forecast accuracy to the naïve method, in most cases, thus the lines overlap once ARIMA plateaus.

Like experiment one, the R^2 and RMSE negatively correlate with each other, suggesting that forecasts with low accuracy are underperforming in both correlation and bias. Because of this correlation, all accuracy discussions followed will not distinguish R^2 and RMSE, unless specific accuracy values required discussion.

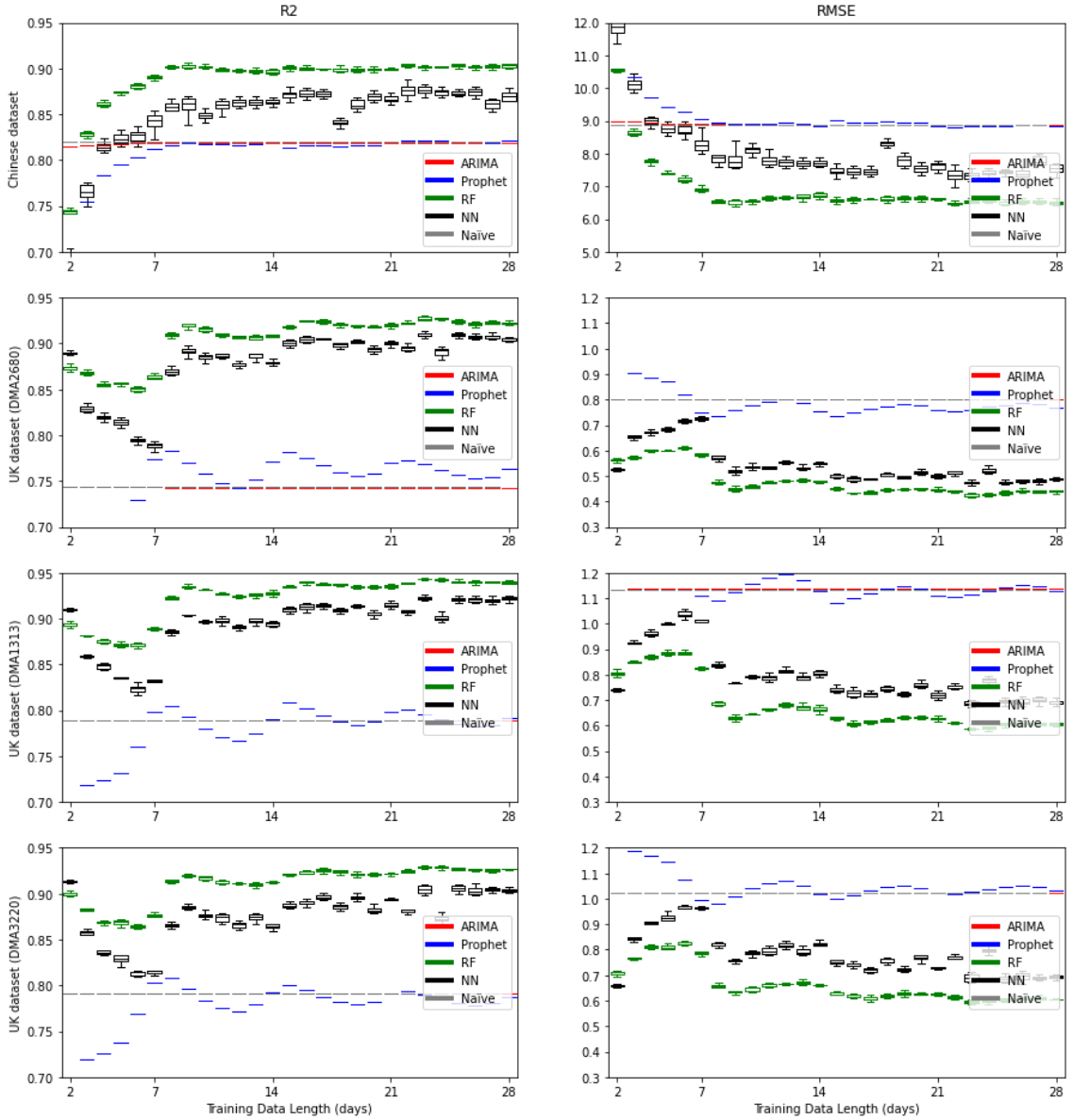

Figure 4.4 Training data length analysis

From Figure 4.4, the results from the top panels for the CHN dataset show that all models approach their optimal accuracy level with 10 days of training data, with only NN showing a significant further improvement, both in terms of accuracy and model stability (variance decrease). Prophet and ARIMA produce similar forecast accuracy when plateaus are reached.

Panels from rows 2, 3 and 4 show that there is a periodicity in how result accuracy changes with increased training length in all models for UK datasets. The period identified is seven days, and the first peak appears on day eight or nine depending on the forecasting model.

RF and NN have reached their first local peak on day nine, then each subsequent local peak is reached seven days after the previous peak. The improved results for two days beyond *n* whole weeks could be explained by the importance of weekend information. As Saturday and Sunday have slightly different demand patterns, two additional days of training data can improve the weekend forecast, especially when the training data is short. The accuracy oscillation effect diminishes with longer training data. Overall, RF has shown to be more accurate and stable compared to NN, and it has reached a stable peak at nine days compared to 25 days for NN.

In contrast, Prophet has reached global optimum at first accuracy peak at eight days. The accuracy then oscillates around the naïve method level, peaking every seven days after day eight; however, the average accuracy slowly decreases with increased training data length. This effect can be explained by reviewing the Prophet model structure. Due to Prophet's additive nature, the seasonal trends remain consistent. The overall trend change in the testing period follows the trend change frequency detected in the training period. Since short-term water demand does not experience significant overall trend change, prolonged training data would introduce unnecessary change points and could cause overfitting in the testing data. Additionally, a shorter training record means that the training data more closely relates to the testing data in the temporal space. The first local peak in forecast accuracy for Prophet should be taken as the global peak.

For ARIMA, all results lie closely to the naïve method, suggesting its seasonal factor played the most significant role in the forecast model, and the remaining parameters had little effect.

The oscillation effect in Prophet, NN and RF suggests that when facing limited training data availability, more data is not always conducive to higher forecasting accuracy. Although more training data usually benefit model performance, a cut-off should be recognised, that splits data availability into sufficient and insufficient groups. The former would benefit from ever more training data, and the latter may benefit from using strategic training length based on availability. The result here shows that strategic training length choice would benefit models that are trained using less than three weeks of training data. This analysis suggests that weekly or less frequent data features have a minimal impact on model forecast accuracy. Models that consider these features such as ARIMA and Prophet show no advantage over models that do not.

4.3.3. Temporal Resolution Analysis

Another point of interest is to review how different models react to decreased data temporal resolution. Since decreasing data resolution is done by taking the moving average of the original data, the new low-resolution demand record is a smoother version of the original demand series. The results shed light on how each model would react to extreme points in data. As the Chinese dataset already has lower data resolution and a shorter total data length, it is excluded from this experiment. The other three case studies are analysed here by aggregating every n demand value (n values of 1, 2, 4 and 8 correspond to 15-, 30-, 60-, and 120-minute sample rates). The lowered resolution data series are forecasted and compared to the original data.

Figure 5 shows how the forecasting models react to reduced data resolutions for case studies 2, 3 and 4. The R^2 measure is unit free, but the RMSE does have a unit, which correlates with the size of the measured demands. Since the lower-

resolution datasets are generated by aggregating high-resolution data, the RMSE values at different resolutions cannot be directly compared. As a result, each RMSE result shown in the right panels of Figure 4 is divided by the number of aggregating points, namely 1, 2, 4 and 8, for 15-, 30-, 60-, and 120-minute sample rates, respectively.

Like previous experiments, the R^2 and RMSE results negatively correlate with each other, thus, the R^2 and RMSE results in Figure 4.5 will be jointly discussed. The results all show that the forecast accuracy increased with decreased data resolution for Prophet, ARIMA and naïve method forecasting (accuracy overlaps with ARIMA results); and the opposite is true for RF and NN.

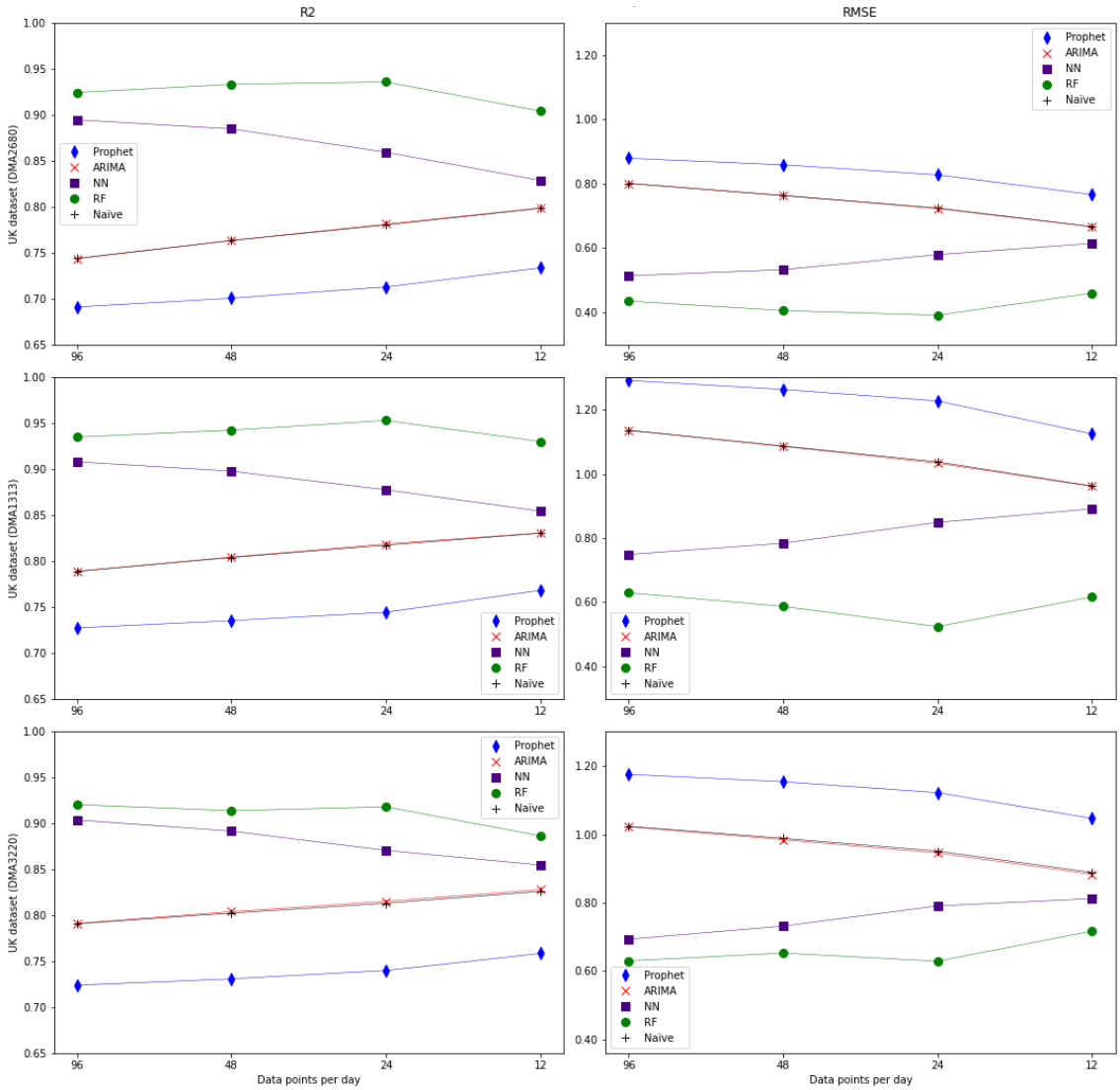

Figure 4.5 Data resolution analysis

The model reaction difference to resolution can be explained by reviewing the model structural differences. Prophet and ARIMA can both be viewed as holistic forecasting models, where an overview of the data is drawn and used, whereas RF and NN build models by reviewing data relationships modularly, without any overview. Lower resolution demand is generated by taking the moving average of the original demand; thus, the peaks and throughs are less pronounced. Modular forecasting models such as RF and NN allow more flexibility in forecasting data peaks and throughs. As a result, RF and NN are better at forecasting high-resolution data compared to Prophet and ARIMA.

It is worth noting that whilst the forecast accuracy improved for Prophet and ARIMA when the resolution is decreased, it is at best on par with the naïve method, still far worse than RF and NN. This analysis indicates that the holistic models (ARIMA and Prophet) are inferior for short-term water demand forecasting. Combining this with the results from the previous experiment, it can be generalised that for sub-daily water demand forecasting, the daily data feature plays a key role, while features that are weekly or less frequent have minimal impact. However, the impact of less frequent seasonal factors increases with the decrease in data temporal resolution.

4.3.4. Data Uncertainty Analysis

The final experiment aims to determine the impact of data uncertainty. This is done by forecasting using noisy data for training. The noise is added by generating Gaussian noise to the training data, with the mean noise zero and a varied scale (between 0 and 50% of average demand).

Figure 4.6 shows the impact of noisy data on all case studies. The left panel shows R^2 accuracy whilst the right panel shows RMSE accuracy. As it has been with R^2 and RMSE comparisons in previous experiments, the R^2 and RMSE accuracy negatively correlate with each other, suggesting that better forecasting results are superior in both correlation and residuals. Therefore, subsequent discussions of accuracy will be done in terms of more and less accuracy when comparing methods or case studies.

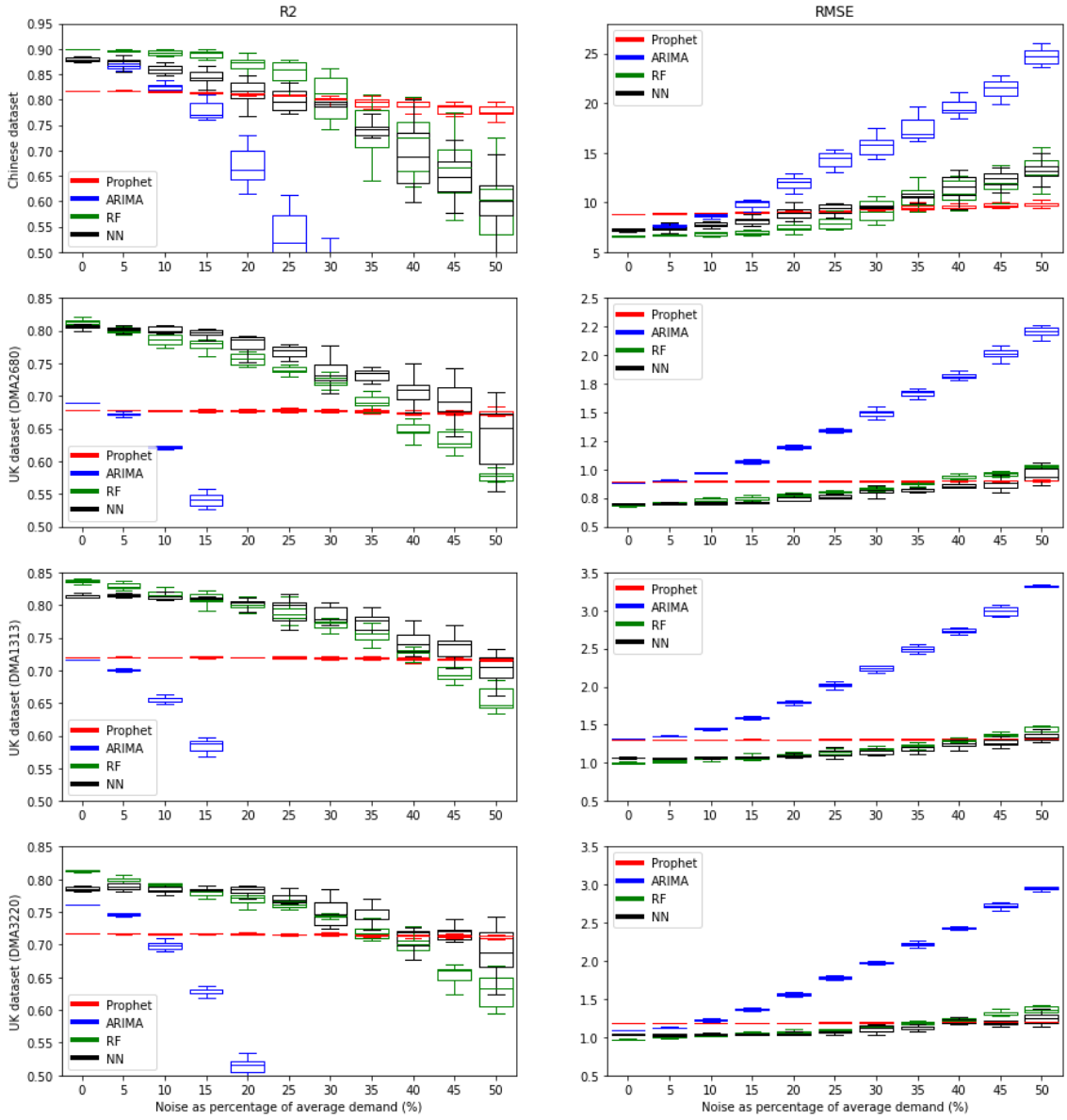

Figure 4.6 Uncertainty analysis

The uncertainty results from Figure 4.6 shows that the impact of data noise differs greatly between models, with the most significantly affected being ARIMA and

the least being Prophet. All figures show that Prophet can maintain the same level of accuracy regardless of noise, albeit the accuracy variance slightly increases towards a higher noise level. Whilst Prophet made inferior accuracy using training data without noise, its robustness allows it to eventually outperform all other models.

Whilst RF and NN eventually fall below Prophet, the rate of accuracy reduction differs for RF and NN. RF models show consistent accuracy decrease regardless of the noise level. In contrast, NN models' performance drops slowly at low noise levels, and then the rate of drop accelerates rapidly when the noise level is high than 20%. Both models show significant forecast variance at high noise levels.

The findings show that data quality is of great importance to most forecasting models. For NN and RF models, a 10% data quality improvement would raise the R^2 accuracy level by 0.05. This shows that superior forecasting models are sensitive to data quality.

4.4. Summary

Short-term demand forecasting is particularly useful for operation management and, for example, could be used for leak detection. In this work, four models including three often used models – ARIMA, RF and NN, and one relatively new model – Prophet, are compared to determine the advantages of data-centric approaches in the field of short-term water demand forecasting.

The results show that all models can make highly accurate forecasts both in terms of R2 and RMSE. Whilst all models have proven their ability in their application in the field of short-term water demand forecasting, the performance of different models varies with the same data set, with RF consistently producing forecasts

with the highest R2 and lowest RMSE. This implies that whilst data-centric approaches deserve more research attention, model-centric approaches cannot be omitted, as the appropriate model choice is an important first step in ensuring accurate forecasts.

The parameter analysis has shown that most models are insensitive to parameters in most cases. This is especially true for Prophet, ARIMA and RF; for ARIMA and RF, knowing the data seasonality is more important than searching for optimal parameter values. Whilst NN is shown to be significantly affected by the number of neurons in the hidden layer, its choice can be generalised to twice the number of inputs. These results imply that efforts in model calibration can be minimised for short-term water demand forecasting. The high accuracy and lack of parameter effects confirm that data-centric approaches warrant more investigation than model-centric approaches.

The training data length analysis has shown that more data does not necessarily provide better forecasts. This is especially true when using Prophet to forecast short-term water demands. The accuracy oscillations in Prophet, RF and NN suggest that when using shorter training data, high accuracy can still be achieved when using the right amount of training data. This study found that when using small training datasets, the optimal training data length is one day more than n whole weeks for Prophet and two days more than n whole weeks for RF and NN. Prophet performs better with shorter training data, for cases where data has little long-term value shift, such as short-term water demands.

When considering data temporal resolution and forecasting model pairing, RF and NN are better for high-resolution data, whilst ARIMA and Prophet are better for low-resolution. Due to the data used in this research, the resolution effect is

present but varies for different models. This implies the significance of analysing data temporal resolution in the development of machine learning. This finding needs to be further confirmed by doing similar tests on short- to medium-term water demand predictions.

The findings from training data length analysis and data temporal resolution analysis show that daily data feature plays the most significant role in short-term water demand forecasting with data features that are present with a weekly frequency having minimal impact. Therefore, the models such as Prophet and ARIMA that consider longer-term seasonal factors have no advantage over other models.

Lastly, data quality is shown to have a significant impact on forecast accuracy in most models. Although Prophet has shown that it is immune to data noise, it has produced lower accuracy forecasts compared to other models with uncorrupted data. RF and NN data uncertainty test has shown that 10% data quality improvement can improve R2 by 5% This shows that higher accuracy forecasting models are sensitive to data quality, and data quality improvement can offer similar accuracy improvement to that of complex model-centric approaches.

Overall, data-centric machine learning approaches hold great potential in improving the accuracy of short-term water demand forecasting. In addition to improving data quality, a data-centric approach also considers how to make the best use of data. In this research, training data length, data resolution and data uncertainty are analysed under the data-centric approach framework. The results have shown that these aspects have a greater impact compared to model tuning which is an aspect of the model-centric approach. Further research could investigate other aspects of data-centric machine learning approaches to improve

forecast accuracy and reduce computation costs. Whilst all forecasting models have proven capable of forecasting short-term water demand, further research should focus on machine learning forecasting models, as Prophet and ARIMA have shown inferior performance capability,

Chapter 5 - Unboxing Black-box Machine Learning Models For Short-term Water Demand Forecasting

5.1. Introduction

Machine learning (ML) models have long been used for water demand forecasting with a generally high accuracy achieved. The forecasting models investigated in the previous Chapter have shown that ML models are superior in terms of forecasting accuracy, compared to statistical models. However, the black-box nature of ML models produces forecasts with unexplainable results; unlike statistical models such as Prophet and ARIMA, where the forecast components are intuitively comprehensible and readily presentable. It would be ideal to produce forecasts with ML models, whilst retaining the explainable components of statistical models.

Recently, explainable AI has received ample interest to overcome the black-box nature of ML models. Techniques such as Local Interpretable Model-Agnostic Explanations (LIME) (Garreau and von Luxburg 2020) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017) offer a post-hoc add-on to ML models, to determine how much each input feature contributes to forecasts. The goal would be to maintain the forecasting accuracy achieved by ML models, whilst making the results more explainable.

This Chapter aims to investigate the impact of input features on machine learning (ML) models, for short-term water demand forecasting. To achieve this, explainer models LIME and SHAP are applied to four different ML models – NN, RF, XGB

and LSTM. The following sections review the background and application of the explainers and ML models. The final sections present the results and summary.

5.2. Methodology

This section starts with an overview of LIME and SHAP. The two explainer models are selected as they work as post-hoc methods that can be applied to existing ML models, and they have been applied in other forecasting fields and have produced useful findings. Whilst both are useful in determining the feature contribution, SHAP can offer both individual sample analysis and overall feature impact overview, though it is slower. Whilst LIME calculates feature impact much more quickly, it can only produce results for one sample at a time.

After an overview of the explainer models, the section presents a brief explanation of the four forecasting models and the choices of accuracy and performance indicators. Finally, this section ends with the experimental setup.

5.2.1. Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations (LIME) (Garreau and von Luxburg 2020) is a post-hoc machine learning model explainer that focuses on individual predictions. LIME operates by training the local surrogate models, these are trained with the goal of approximating the prediction of the original black-box model, around the given sample. Using local surrogate models, input feature values can be tweaked, and their impact on the output can be measured. The surrogate models only aim for accurate approximation locally, i.e., for the given sample; but do not have to be accurate globally, i.e., for other samples. This local truthful approximation is called local fidelity.

The mathematical equation for the LIME explanation is the following:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (5.1)$$

where x is the local sample interested; f is the original black-box model (e.g., RF or NN); g is the explainable model used (e.g., linear regression model); G are all potential explainable models, i.e., if g is a linear regression model, G would be all possible linear regression models; π_x measures the size of the neighbourhood around sample x that is considered for an explanation; L is the local fidelity, it can be calculated by how accurate surrogate models g approximate original model f, for the given sample; and lastly, $\Omega(g)$ is model complexity.

When applied, model complexity $\Omega(g)$ is determined by users. It correlates to the number of features required for consideration by the surrogate model g. Thus, LIME focuses on minimising the local fidelity L from Equation 5.1.

The application of LIME starts with choosing the sample of interest. The sample's feature values are perturbed, around a neighbourhood of size π_x. Black-box predictions are made for the perturbed samples using the original model f. The new samples are weighed based on their proximity to the sample of interest. An interpretable surrogate model can be formed by training the weighted model using the perturbed dataset. Finally, the surrogate model can be explained, though only accurately reflecting the forecasting result for the initially selected sample.

LIME is implemented in Python, using the 'lime.lime_tabular' package (Ribeiro 2021), for all forecasting models. Within this, 'LimeTabularExplainer' is applied to NN, RF and XGB, and 'RecurrentTabularExplainer' is applied to LSTM. The difference between LIME explainer choices is based on the expected model

inputs. LSTM expects an input of shape (n_samples, n_timesteps, n_features), whilst the other models expect inputs of shape (n_samples, n_features). The LIME explainer, therefore, builds surrogate models differently for LSTM compared to other forecasting models.

5.2.2. SHapley Additive exPlanations

SHAP is a machine learning model explainer that employs a game-theory approach, developed by Lundberg and Lee (Lundberg and Lee 2017). It outputs a measure of individual feature contribution in any ML model, both locally and globally.

The SHAP method determines feature impact contribution by finding features' Shapley values. Shapley values represent a distributed contribution among the features, and they are computed using coalitional game theory. Like LIME, the goal of making a black-box model explainable can be achieved by determining individual feature impact; but unlike LIME, SHAP could determine feature impact globally, and then individual feature impacts are weighted to be comparable.

SHAP determines individual feature impact by calculating the average marginal change in prediction for feature combinations with and without the given feature. The idea can intuitively be understood by considering the average predictions with all possible feature combinations excluding feature i, and all combinations including feature i. The Impact of the feature i is the difference between the two prediction classes. The mathematical definition of Shapley value is as below:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}\left[f_x(z') - f_x(z' \setminus i)\right] \qquad (5.2)$$

where $\phi_i(f,x)$ is the Shapley value for feature i of model f built via features x. M is the total number of input features, x' is all possible feature combinations that include feature i, $|z'|$ is the number of features in combination z', $f(z')$ and $f(z' \setminus i)$ are unique models trained on z' and $z' \setminus i$ (z' without feature i).

The application of SHAP can be further clarified by considering an ML model with three features, named 'A', 'B' and 'C'. To determine the Shapley value of feature 'A', the following feature combinations must be considered:

- B
- C
- B & C

The above feature combinations are considered both with and without feature 'A', the difference in the average of feature 'A' inclusion and exclusion is the individual impact of feature 'A'.

SHAP is implemented in Python (Lundberg and Lee 2017), using the 'shap' package. The 'shap' package has different explainers for different forecasting models. The basic 'KernalExplainer' works on all models, though it is slower and only offers an approximation of the SHAP value. The explainers used in this study are 'TreeExplainer' and 'DeepExplainer', as the name suggests, the former focuses on tree-based models and the latter on deep learning models.

5.2.3. Neural Network

The theory behind Neural Network is discussed in Chapter 4.2.3, the following will discuss the implementation of NN that are specific to the experiments designed in this chapter.

The NN model used in this work is a four-layer feed-forward model with backpropagation. The choice of the four-layer NN model is used instead of the basic three-layer model is due to the nature of different SHAP packages. Whilst SHAP is model agnostic, only the 'KernalExplainer' can be applied to all models, the 'KernalExplainer' operates like LIME, using regression models to approximate predicted outcomes. But as SHAP must calculate feature contributions on a global scale, the rate of contribution calculation is very slow. In comparison, 'DeepExplainer' aggregate many background sample data to speed up the approximation process, and this explainer works on both LSTM and DNN.

The model is built using the 'Keras' package in 'TensorFlow' in Python (Goodfellow et al. 2016). The model consists of an input layer, which varies in size based on the experiments; this is followed by two hidden layers of size 12 and 8, both using Rectified Linear (relu) activation function, this activation function is used here due to its simple and effective nature, it outputs zero when input is negative, and outputs linear values when input is positive, this is suitable as all inputs are normalised between 0 and 1. The parameter choices are all standard choice taken from other examples, the training accuracy is saved to show that the models are well trained, thus, the parameter choices are not altered. The output layer consists of one node and uses the sigmoid function as an activation function (Goodfellow et al. 2016). The model is compiled using the 'adam' optimiser, as it is faster and requires fewer parameters for tuning, compared to other optimisers. Each layer is densely connected to the adjacent layers, meaning all nodes from one layer connects to all nodes from the adjacent layers. The choice of parameters is taken from example forecast models in the reference and has been shown to produce a comparable accuracy level to other models. As this

work focuses on feature contribution analysis, the parameter choice is not exhaustively studied.

5.2.4. Long Short-Term Memory

LSTM is a specific type of Recurrent Neural Network (RNN), which is a category of NN. LSTM has been applied (Mu et al. 2020; Nasser et al. 2020) and compared to NN (Boudhaouia and Wira 2021). Recurrent Neural Networks (RNNs) are designed to improve upon NN by having the ability to learn long-term data dependencies, where the current event depends on successive past events. The model uses memories to learn long-term events and, as a result, deeper RNNs are better as they can remember more past information. However, due to its architecture, RNN suffers from the vanishing gradient problem (Mu et al. 2020), where knowledge from long-term dependencies fails to register an impact on (Fu et al. 2022)current forecasts. LSTM is designed to overcome this limitation of RNN by having the ability to retain longer periods of information.

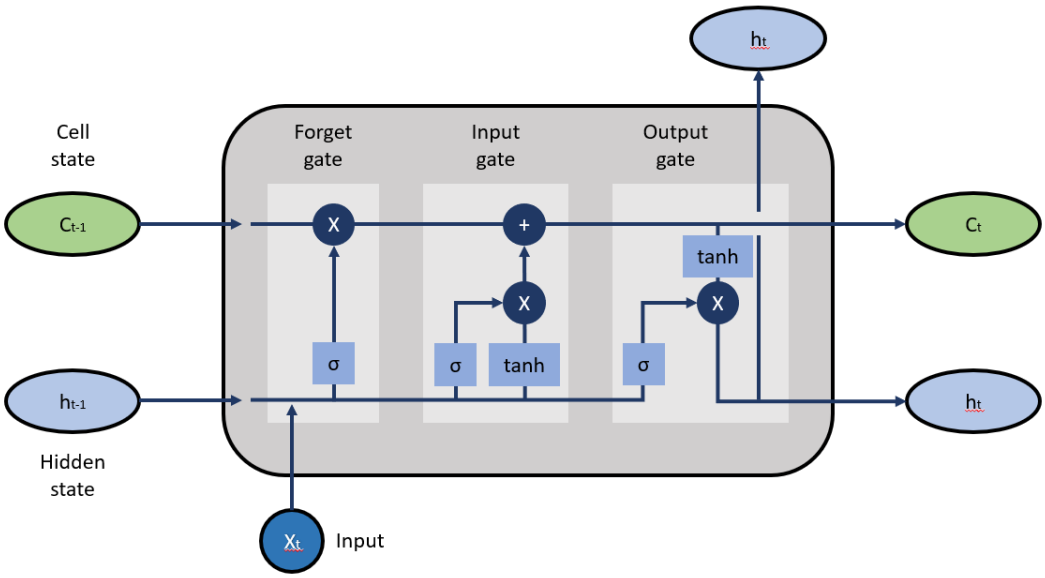

Figure 5.1 LSTM structure (Fu et al. 2022)

Figure 5.1 illustrates the structure of LSTM. The model contains three parts, which are the input gate, forget gate and output gate, each processed by a

sigmoid neural network layer (σ) and a multiplicative unit ($\otimes$). The equations for how LSTM connect input and output are shown below, and further details can be found in (Gers et al. 2000):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5.3}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5.4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5.5}$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{5.6}$$

$$s_t = g_t \otimes i_t + s_{t-1} \otimes f_t \tag{5.7}$$

$$h_t = \tanh(s_t) \otimes o_t \tag{5.8}$$

where $\otimes$ denotes the multiplication between two vectors; σ and $\tanh$ are sigmoid and $\tanh$ activation functions, respectively, the sigmoid activation function outputs values between 0 and 1, whilst the tanh activation function outputs values between -1 and 1; t is the current time; W and U are weights; b are biases at different processes; i, o and f correspond to the input, forget and output thresholds, respectively; g is the candidate cell state generated by the neural network layer; s is the cell state; and h is the output vector.

The LSTM model used in this work is built using the 'Keras' package within the 'TensorFlow' library in Python (Goodfellow et al. 2016). All LSTM models used have been initialised with the relu activation function and have employed 'adam' and mean squared error (MSE) respectively as the optimizer and loss function (Goodfellow et al. 2016). The reason for using relu and 'adam' for LSTM is due to their suitability, simplicity, and performance, the same as it is for NN. No further

analysis is done on the subject as these choices produced comparable forecast accuracy to other models.

5.2.5. Random Forests

The theory behind Random Forest is discussed in Chapter 4.2.4, the following will discuss the implementation of RF that are specific to the experiments designed in this chapter.

The RF model used in this research is the 'RandomForestRegressor' from the 'sklearn' library within Python (Géron 2017). Most parameters take default values since alternative parameter values offered no gain in forecast accuracy. All available features are considered, as the goal of this work is to determine the impact of available features.

5.2.6. Extreme Gradient Boost

Extreme Gradient Boost is a scalable tree-boosting system, which uses Gradient Boosting (GB) methods to improve the decision tree model's speed and accuracy. The method was developed by Chen and Guestrin (Chen and Guestrin 2016) and has shown successful applications in various competitions. XGB exhibit better computational efficiency compared to original GB methods for its ability in parallel computation, approximate tree matching, effective handling of sparse data and improvement for central processing unit and memory. The goal of XGB optimisation is to minimise the objective functions (L). The objective function for XGB is shown in Equation 5.9:

$$L = \sum_{i=1}^{n} l(y_i^o, y_i^f) + \sum_{k=1}^{K} \Omega(f_k) \qquad (5.9)$$

$$y_i^f = \sum_{k=1}^{K} f_k(x_i) \tag{5.10}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda w^2 \tag{5.11}$$

where y_i^f is the forecasted output; y_i^o is the observed output; x is the input vector; f_k represent the kth regression tree; K is the total number of available trees; Ω is the regularisation term used to penalise model complexity to avoid overfitting, where γ, T, λ and w respectively represent the complexity of each leaf, the total number of leaves, a trade-off parameter to scale the penalty and the vector of scores on leaves.

The XGB model used in this work is the 'xgboost' package within Python (xgboost developers 2022). All parameters take default values as alternative parameter values offering no gain in forecast accuracy. The sole inputs to the XGB model are the features (past demand) with the expected outputs.

5.2.7. Experimental Set-up

To evaluate feature contribution and requirements, four experiments are designed to evaluate explainer models for different forecasting scenarios. The data used for the experiments in this Chapter are of the three highlighted UK datasets from Table 3.1, further mentions of the data will reference their individual DMA numbers.

The first experiment aims to create an overview of the four models' performance, and then use LIME and SHAP to determine feature contribution rankings. The forecast horizon is set to 15 minutes or one hour for high- and low-resolution data, respectively. Resolutions lower than hourly demand is ignored in this chapter to preserve high sample availability during training, the focus in this chapter is to

determine key feature requirements in short-term water demand data, to replace expert data knowledge. Similar effort was only shown to have been made towards sub-daily demand, thus, the resolutions investigated are of 15-minute and hourly demand. The total data length is 10 weeks and 80% of the data is selected for training and the remaining 20% for testing. Compared to Chapter 4, the ratio of training data is increased, this is done because the focus in this chapter is on feature contribution, as opposed to forecasting accuracy, and feature contribution is viewed using the training data as samples, thus, longer training set means more samples to review. The R^2 and RMSE accuracy are recorded for both the training and testing periods, and the LIME and SHAP explainer models are applied to the training data. The accuracy results will demonstrate model forecasting ability, and the LIME and SHAP results will show an overview of feature contributions. The contribution ranking comparison between LIME and SHAP will reveal the differences between models. The overall feature contribution results from SHAP will be further analysed via Gini Index (GI) to determine the contribution spread between all features and the difference in contribution spread between varying data resolutions. The number of features used in each model is the number of demands at previous time steps up to a whole day. This experiment is done on all three DMAs for both high- and low-resolutions. The bi-resolution analysis aims to identify different feature requirements due to resolution differences.

The second experiment aims to determine the effect of having longer forecast lead times. As some models are more reliant on forecasting features being temporally close to the point of the forecast, this experiment will demonstrate the scale of models' temporal dependency. The experiment will be done by using the same forecast horizon as experiment one (one point) but adding a temporal gap

between the final feature and the point of the forecast. The gaps analysed are between 0 and 23, where a temporal gap of 23 is equal to forecasting one day ahead. The results are expected to show how the GI of feature contributions and forecast accuracies react to the increasing temporal gap between features and the point of the forecast.

The third experiment aims to show the accuracy impact of primary essential features on different models, where the primary essential features are those with the highest SHAP rankings. This experiment would numerically show the amount of information different models can extract from fewer input features. The experiment is done by training forecasting models with reduced features, starting from one feature, with an increment of one feature at each step, up to two days of features. The features reintroduced in each feature addition are selected from SHAP analysis, where the highest contributing features are reintroduced to training each time. The result should show an accuracy increase with the increasing number of features, but the rate of increase could differ for models due to differing feature dominance.

The second and third experiments will use only low-resolution data, but the temporal length of features is extended to two days from one. The reason that the high-resolution data is not investigated here is a result of experiment one, where high-resolution data all showed identical primary dominant features between DMAs.

The final experiment aims to discover how feature impacts compare based on the resulting daily peaks and troughs. This is done by first locating the morning peaks, afternoon peaks and daily troughs; the contribution plot is then shown for each

point of interest, alongside the all-sample contribution plot. The result could reveal varying feature impacts for different times of the day.

5.3. Results and Discussion

This section presents the results relating to the application of LIME and SHAP. The two explainer models are applied to NN, RF, LSTM and XGB. The data used are the non-corrupted sections of three UK DMAs – UK4, UK11 and UK12.

The four experiments respectively look at 1) an overview of the feature impact; 2) the impact on accuracy when forecasting n-hours ahead; 3) the impact on accuracy when the features are reduced; and 4) the feature requirements for different times of the day.

5.3.1. Model Performance and Feature Importance Analysis

The first experiment aims to provide an overview of feature requirements and the forecasting ability of different models for all DMAs across two resolutions. The forecast accuracy can be examined from Table 5.1, while feature impact can be visually analysed from Figures 5.2 and 5.3. The feature impact spread can be seen in Table 5.2.

Table 5.1 shows the forecast accuracy for both training and testing data, for all datasets across the four models. The accuracy measures of R2 and RMSE display a negative correlation, suggesting that superior forecasting models achieve both better correlation and lower residual. As there is a negative correlation between the two accuracy measures, further discussions of accuracy will be done without referencing specific accuracy indicators, and the term ‘accuracy’ will encapsulate both R2 and RMSE.

Table 5.1 Model forecast accuracy

			UK4		UK11		UK12	
			High res	Low res	High res	Low res	High res	Low res
NN	Train	R2	0.970	0.964	0.966	0.956	0.973	0.969
		RMSE	0.040	0.048	0.037	0.047	0.037	0.041
	Test	R2	0.966	0.956	0.959	0.952	0.971	0.964
		RMSE	0.043	0.051	0.041	0.051	0.037	0.043
LSTM	Train	R2	0.968	0.951	0.966	0.957	0.977	0.958
		RMSE	0.042	0.056	0.037	0.047	0.034	0.047
	Test	R2	0.967	0.949	0.960	0.966	0.977	0.962
		RMSE	0.043	0.056	0.039	0.045	0.034	0.045
RF	Train	R2	0.996	0.994	0.995	0.993	0.997	0.994
		RMSE	0.015	0.020	0.015	0.019	0.013	0.018
	Test	R2	0.968	0.953	0.964	0.951	0.977	0.969
		RMSE	0.042	0.053	0.039	0.052	0.035	0.040
XGB	Train	R2	0.997	1.000	0.997	1.000	0.998	1.000
		RMSE	0.012	0.004	0.012	0.004	0.010	0.003
	Test	R2	0.965	0.957	0.960	0.949	0.975	0.970
		RMSE	0.044	0.051	0.041	0.053	0.036	0.040

All forecasting models have produced results of similar accuracy in the testing data. The tree-based models (RF and XGB) tend to be superior with training data forecasting compared to network-based models (NN and LSTM). They achieved exceptionally high accuracy during training periods, though this superiority fades

when facing unknown (testing) data and approaching those of network-based models.

The comparison between training and testing data accuracy shows that most of the training accuracy is higher than testing accuracy, this is intuitively understandable as the forecasting models are fitted to the training data. However, the training and testing accuracy difference is far greater for tree-based models compared to network-based models. This suggests that the tree-based models tend to overfit training data, and their forecasting accuracy for testing data drops to the same level as network-based models. Additionally, the four models used forecast high-resolution data slightly better than low-resolution data, which holds for most forecasts (all but for RF in training data forecast).

Feature impact is first analysed via LIME, and sample ranking examples are shown in Figure 5.2. Due to the nature of LIME analysis, only sample rankings can be extracted and viewed, thus example results are taken from UK11, and three examples are presented for each model and resolution combination. The top three panels in Figure 5.2 show the high-resolution results and the bottom three show the low-resolution results. The order of models from the left panel to the right is – NN, LSTM, RF, and XGB. For each resolution, the three results are of the top three samples. The sample selection is randomised, but the example rankings can be compared.

Figure 5.2 LIME feature impact analysis for DMA 1 between two resolutions (hourly and 15-minute)

Within each sample result in Figure 5.2, the y-axis corresponds with the feature and its value range that generates the shown impact; the x-axis represents the relative feature impact size; the coloured bars show whether the given feature positively (green) or negatively (red) correlates to the point of the forecast. The y-axis translates directly between NN, RF and XGB results, where -n represents the past n point's demand; however, the y-axis labels for LSTM differ from others due to its 3-dimensional input structure. For LSTM, the y-axis labels of '-1_t-n' can be understood as follows: the initial '-1_t' is the past point behind the point of

the forecast, and the following '-n' is the number of points behind '-1_t'. Thus, the label '-1_t-n' represent –(n+1) point's demand.

Feature impacts are further analysed via SHAP, Figure 5.3 shows the SHAP results for feature impact ranking on different models, DMAs, and resolutions. Figure 5.3 is split into six panels, each containing the SHAP feature results for a particular DMA with a particular resolution. Each panel contains SHAP results of four forecast models shown in the order of – NN, LSTM, RF and XGB.

Within each result, the blue/red lines are made up of multiple dots, with each dot representing a particular scaled feature contribution. The colour of the dot corresponds to the specific feature value (blue - low, red - high). The horizontal location of each dot indicates the feature contributions (x-axis, bottom) corresponding to a respective ranked feature (y-axis, left). It is important to distinguish between the feature value (colour) and the feature contributions (horizontal location, x-axis). The feature values are scaled per feature, using the UK11 high-resolution NN result as an example, the top two features are demands from the previous one point (-1) and the previous one hour (-4), respectively; whilst both lines contain the same red and blue, the same colour does not mean the same feature value, as the feature values are scaled per feature. However, the x-axis locations are uniform for all features, where the middle black line corresponds to zero, indicating no contributions and the left and right of the line correspond to negative and positive impacts, respectively.

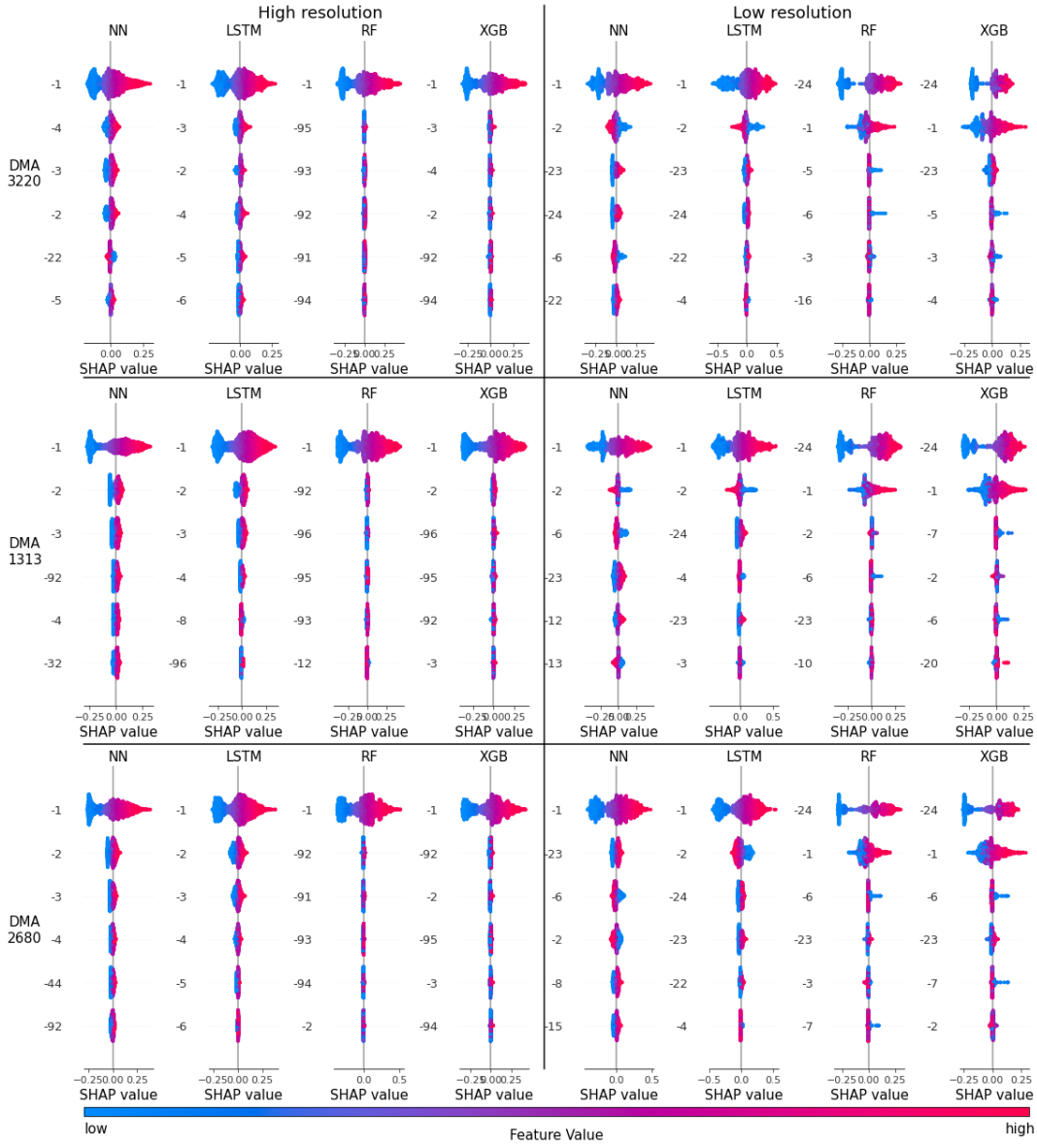

Figure 5.3 SHAP value analysis for all DMAs between two resolutions (hourly and 15-minute)

The comparison of impact analysis between LIME and SHAP shows general agreement between the impact rankings. For high-resolution results, the primary dominant feature shows a significantly larger impact size compared to others, the degree of this size difference is more prominent for tree-based models (RF and XGB) compared to network-based models (NN and LSTM), and this holds for both explainer models. For low-resolution data, there appear to be two dominant impacting features, the impact contribution is for the primary two features are more evenly split for tree-based models, compared to network-based models. For tree-based models, the two primary features are the previous one hour and the

previous day's demand; whilst network-based models show preference towards previous demands that are closer to the point of forecast in the temporal space.

One day of features is used for both high- and low- resolution datasets, which corresponds to 24 points for low-resolution data and 96 points for high-resolution data. The high-resolution results from Figure 5.3 (left panels) show that the primary dominant feature is consistent between all four forecasting models - the dominant feature is the previous one point. All remaining features have a relatively minor impact compared to the primary, the visual results from Figure 5.3 suggest that feature inclusion beyond one point may not be necessary for high-resolution data.

Whilst the visual result shows agreement of one dominant feature for high-resolution results in all models, the impact spread can be further analysed via the GI. The theory and calculation of the GI are presented in Chapter 2.4, it is commonly used to measure wealth inequality, here it is used to measure feature contribution difference. The GI result from Table 5.2 shows that the NNs have achieved a comparatively low GI compared to other forecasting models. This suggests that NN requires more input data compared to other models to achieve optimal accuracy.

Table 5.2 Gini index result

Model	Data type	UK4	UK11	UK12
NN	High res	0.551	0.535	0.603
	Low res	0.540	0.555	0.510

	Change	0.011	-0.021*	0.093
LSTM	High res	0.846	0.782	0.807
	Low res	0.690	0.770	0.778
	Change	0.156	0.012	0.029
RF	High res	0.888	0.884	0.902
	Low res	0.807	0.778	0.803
	Change	0.081	0.106	0.099
XGB	High res	0.728	0.712	0.741
	Low res	0.705	0.676	0.717
	Change	0.022	0.036	0.024

Results from low-resolution SHAP (Figure 5.3 – right panel) show feature ranking disparities between models. Tree-based models have two similarly important features, demand from the previous 24 hours (-24) and demand from the previous one hour (-1); whilst network-based models have the previous one hour as the primary dominant feature, the subsequent ranked features vary, though -2, -3, -23 and -24 repeatedly appears in the top five of the network-model SHAP rankings. The results suggest that the network models tend to rely on a longer temporal dependency for demand predictions. This dependency will be further investigated in experiment two, where a temporal gap is created between the point of forecast and the features used as input.

Table 5.3 High-res forecast accuracy comparison between 96 and 1 feature

	UK4	UK11	UK12

			One day	One point	One day	One point	One day	One point
NN	Train	R2	0.970	0.948	0.966	0.940	0.973	0.941
		RMSE	0.040	0.053	0.037	0.050	0.037	0.054
	Test	R2	0.966	0.945	0.959	0.934	0.971	0.935
		RMSE	0.043	0.054	0.041	0.051	0.037	0.056
LSTM	Train	R2	0.968	0.950	0.966	0.940	0.977	0.959
		RMSE	0.042	0.052	0.037	0.050	0.034	0.045
	Test	R2	0.967	0.947	0.964	0.935	0.977	0.957
		RMSE	0.043	0.053	0.039	0.051	0.034	0.046
RF	Train	R2	0.996	0.959	0.995	0.949	0.997	0.966
		RMSE	0.015	0.047	0.015	0.046	0.013	0.0410
	Test	R2	0.968	0.936	0.964	0.924	0.977	0.949
		RMSE	0.042	0.058	0.039	0.055	0.035	0.050
XGB	Train	R2	0.997	0.958	0.997	0.948	0.998	0.965
		RMSE	0.012	0.048	0.012	0.047	0.010	0.041
	Test	R2	0.965	0.940	0.960	0.927	0.975	0.953
		RMSE	0.044	0.056	0.041	0.054	0.036	0.048

From Table 5.2, the feature impacts' GIs have shown to decrease from high- to low-resolution cases, which is true for all cases but one (starred). The high-resolution results in Figure 5.2 show the previous data point to be overwhelmingly dominant for all models. The GI results from Table 5.2 show that the feature contributions are more concentrated in a few features in high-resolution forecasting, as shown by higher GI values in high-resolution forecasting

compared to low-resolution. Based on these observations, it can be concluded that high-resolution data can be forecasted to a high degree of accuracy using solely the demand from the previous data point. The accuracy established using only the previous data point as a feature can be examined in Table 5.3, where each left column under the corresponding DMA shows forecasts made using one day (96 points) of data as input features and the right shows forecasts made using only the previous data point. The results show that for high-resolution data, reducing the features used from one day to one point only has a marginal impact on forecasting accuracy.

5.3.2. Forecasting Feature Analysis for n-hours Ahead

This experiment shows how different forecasting models react to a varying temporal gap between the point of forecast and the feature data. The R^2 and RMSE accuracy is measured for both the training and testing for low-resolution data, along with the GI to show feature impact spread. For this experiment, a varying temporal gap is generated between the point of forecast and the final feature, whilst the forecast horizon is kept as one. The result reveals the temporal dependency differences between forecasting models.

Figure 5.4 shows the forecasting accuracies and GI. Each column corresponds to one DMA data (labelled above the top panel), and each row corresponds to an index type (labelled to the left). Within each plot in Figure 5.4, the x-axis corresponds to the size of the temporal gap, and the y-axis corresponds with each measure of interest. Like the previous experiment, the R2 and RMSE accuracies negatively correlate with each other and thus will be discussed jointly in this section.

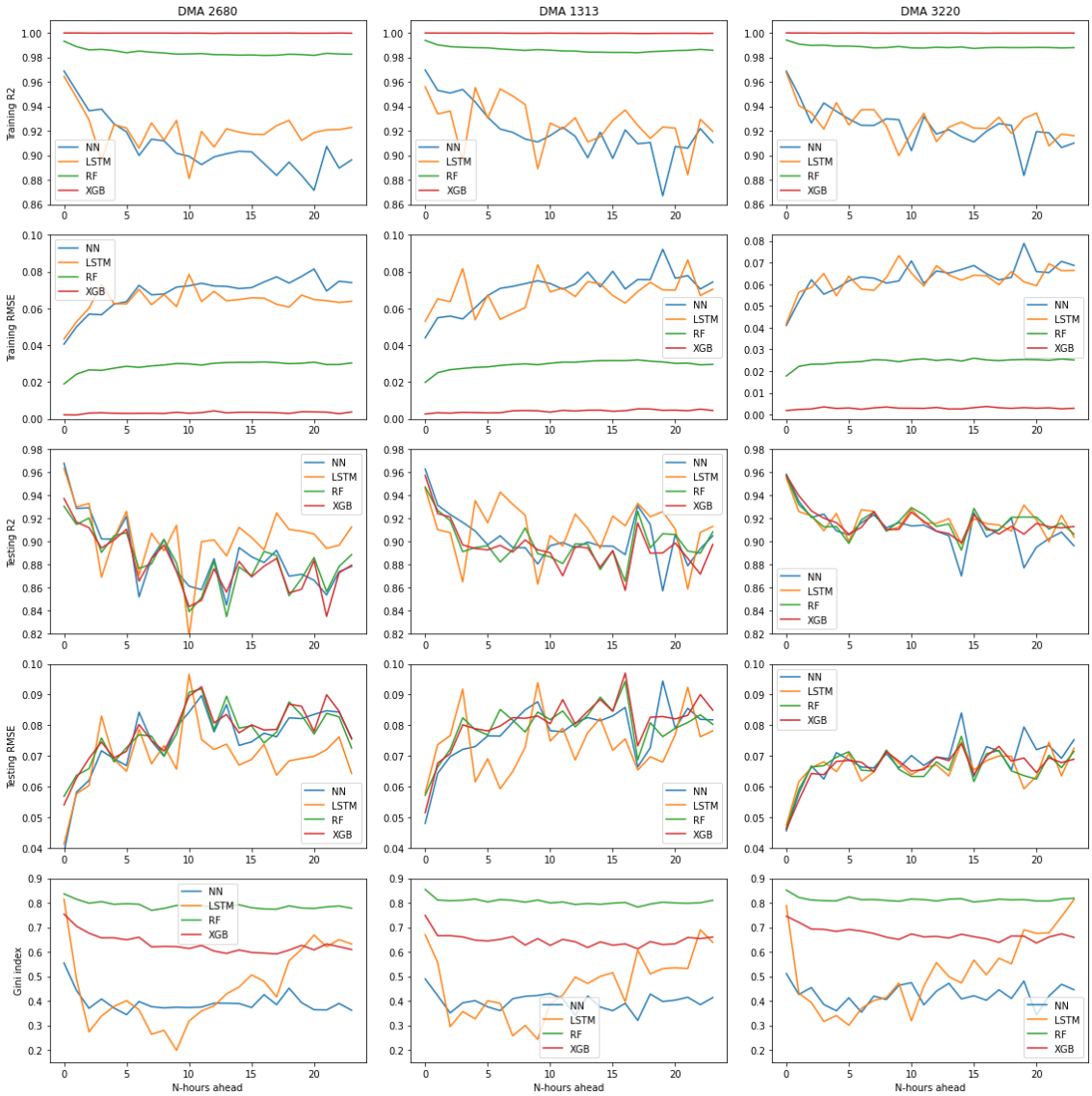

Figure 5.4 N-hour ahead forecasting for temporal dependency analysis

In terms of forecasting accuracy, all models experience a slight accuracy decline when forecasting with a larger temporal gap, for both training and testing results. But interestingly, tree-based models are more accurate when forecasting training data, compared to network-based models. This superiority fades when facing new data (testing data). Additionally, network-based models show a more erratic change in forecasting accuracy on training data when facing a longer temporal gap.

When comparing the GI for different models, RF models have produced the highest GI values, which is consistent throughout, suggesting that the feature

impact spread is not affected by the temporal change. XGB models show similar results, though their GI values do show a slight decline. Network-based models, however, experienced more significant changes in feature impact spread. Even though NNs have produced GI values all in a similar range, their values varied throughout changing temporal gaps; LSTMs have shown to be the most affected model, with GI values dropping significantly after a temporal gap of one, then slowly recovering after a temporal gap of 10. The GI results suggest that network-based models are more temporally dependent, and LSTM is the most affected model by the presence of a temporal gap.

5.3.3. Optimal Feature Inclusion

This experiment looks at how features dependent the forecasting models are. Conversely, the results show models' ability to extract the maximum amount of information from the minimum number of input features, whilst maintaining high forecast accuracy.

Figure 5.5 shows how forecasting accuracy changes for training and testing datasets when the number of features is increased. The features reintroduced in each forecasting model are the highest-ranked features based on SHAP values. This experiment imagines having the knowledge of feature contribution rankings for short-term water demand forecasting, and the degree of each additional feature contributions can be visualised its impact of accuracy. Figure 5.5 is arranged as Figure 5.4 where each column corresponds with a DMA (labelled above the top panel) and each row corresponds to an index result (labelled on the left). Within each plot, the x-axis represents the number of top features used for forecasting, and the y-axis represents the corresponding index value. The SHAP feature ranking is performed once for each model, but the training and

forecasting are repeated 10 times each to show how accuracy results spread. The R^2 and RMSE indices show a negative correlation again, thus discussions of accuracies will encapsulate both measures in this section.

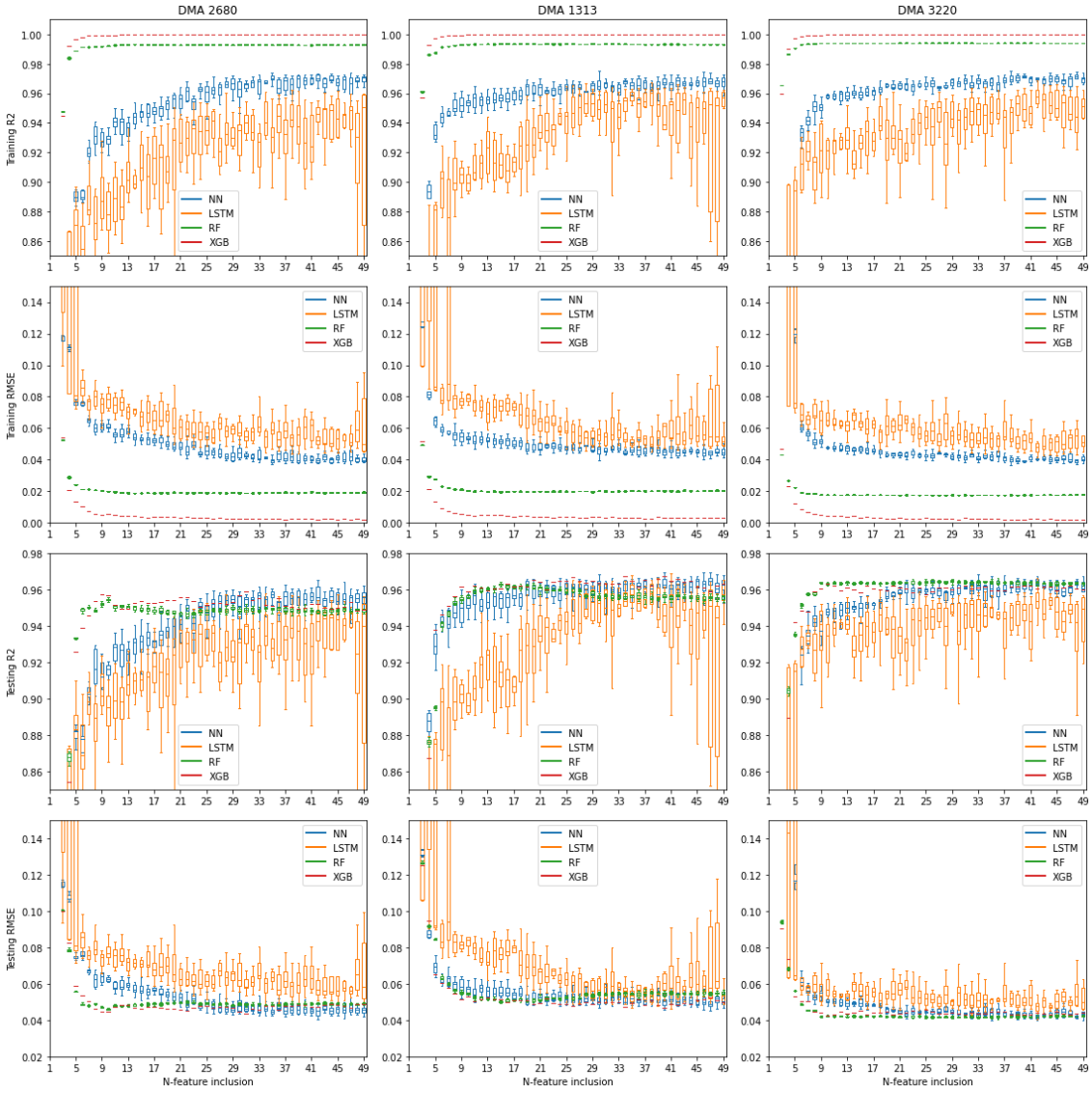

Figure 5.5 N-feature inclusion forecast for information extraction analysis

In Figure 5.5, the training data accuracy confirms that of previous experiments, where tree-based models produce near-perfect forecasts judging by result accuracy, though this superiority is not repeated in testing data. The testing data results show that RF and XGB can produce forecasts with near-optimal accuracy using around 10 features. Adding more tends to have minimal or even negative impact, suggesting that full features may cause overfitting in these models. NN

can reach near-optimal accuracy with 10 features (for UK4), but this result is not repeated in all DMAs, nor is this level of accuracy consistent. Therefore, NN can be said to reach consistently accurate near-optimal forecasts with around 25 features. LSTM has shown to be the worst performer among the four models, as it requires around 30 features for the median forecast to be of near-optimal accuracy. Its forecast accuracies are significantly more varied compared to other models.

5.3.4. Peaks and Trough Feature Impact Analysis

The final experiment looks at the feature requirements for specific times of the day. The times of interest are the morning and afternoon peaks and daily troughs. These three are selected as knowing the demand at these times is more important for short-term water management than for other times.

The feature impact of a specific time of day is investigated by extracting the time of the morning peaks (highest demand between mid-night and noon), afternoon peaks (highest demand between noon and midnight) and daily troughs (lowest demand of the day), then using the time to locate and plot the SHAP values (same as Figure 5.3). The SHAP value plot for all times of day is shown next to individual SHAP value plots for the times of interest. All DMAs have shown to agree on dominant feature rankings for high- and low-resolution data, thus, the SHAP values plots are combined for all DMAs. The combined results are shown in Figure 5.6, where the middle line splits the results into high- (left) and low- (right) resolution panels, each row corresponds to a given forecasting model (labelled on the far left), and each column corresponds to a time of interest (labelled above the first row).

For high-resolution data, where the primary dominant feature (previous point) was shown to overshadow all other features in terms of impact, the scale of this impact correlates positively with the size of the feature. While the previous data point is the dominant feature for all models, network-based models show weaker primary feature dominance, particularly for daily troughs. Additionally, the subsequently ranked features in network-based models show a gradient decline in SHAP value compared to tree-based models, further revealing the models' temporal dependency.

Figure 5.6 Daily peaks and trough feature impact contribution analysis

For low-resolution data, the feature impact differs for morning and afternoon peaks. Whilst the previous data point is important for both the morning and afternoon peaks, its scale and direction of impact differ based on time and forecasting model.

For morning peaks, extreme demands from the previous data point have a high impact on the current point for all models, with high values having a positive impact and low values having a negative impact. In contrast, tree-based models have found the demand from the previous day to be more impactful. The impact of the previous day's demand is strongly positive regardless of the feature value.

For afternoon peaks, the previous day's feature impact for tree-based models is the same as the previous day's impact for morning peaks. However, the previous data point impact differs from morning peaks for all models. The impact of demand from the previous data point correlates well with the actual demand size, where low demand from the previous data point has a smaller impact (in contrast to a negative impact for morning peaks).

The disparity of the previous data point impact on morning and afternoon peaks suggests that morning peak demand varies more greatly and is more closely correlated with the previous data point demand, whilst afternoon demand has a more stable minimum usage and can only be driven up by higher previous data point demand.

For daily troughs, most high-ranking features have a high negative impact regardless of feature value. However, like its high-resolution counterpart, the dominant features (top two) for tree-based models show that the top two ranked

features are the sole impacting features. This contrasts with network-based models where each feature in a long list has an impact on the forecast result.

5.4. Summary

This Chapter investigated the varying feature contributions to short-term water demand forecasting, across different forecasting models and data resolutions. Four forecasting models are applied to 15-minute and hourly data, across three DMAs. SHAP is then applied to each model to determine the feature contribution rankings. The key conclusions are:

The previous data point is the dominant feature for all models when forecasting high-resolution data. Further testing where only the previous data point is used for forecasting high-resolution data shows that extending to the previous day (96 data points) of features only achieved a marginal gain in predictive accuracy compared to the previous data point forecast.

Network-based models are more temporally dependent and feature intensive compared to tree-based models. They have stronger adverse reactions to temporal gaps between features and the point of the forecast, both in terms of accuracy and GI. Network-based models have also shown a significant accuracy drop from having a reduced number of features compared to tree-based models.

Forecasting morning peaks, afternoon peaks and daily troughs depend on different past features. Whilst the previous data point is important for both peak demands, the previous data point influences morning and afternoon peaks differently. The previous data point can influence morning peak demand in both directions, suggesting a lack of minimum in the morning peak; in contrast, the

previous data point can only cause the afternoon peak to go higher, suggesting a more stable minimum for afternoon demand.

The application of SHAP in the field of water demand forecasting could be developed further to evaluate its use in medium- to long-term demand forecasting where more input variables (such as climate and social-environmental input variables) are often included. SHAP can be used to determine key contributing features to water demand forecasting across various forecast horizons. Additional investigation on RF and XGB could also be done using longer training datasets, to see if additional training could improve model performance when facing testing data; the result would determine whether the models are overfitting training data.

Chapter 6 - Optimising the Usage of Multiple Short-term Water Demand Data via Transfer Learning

6.1. Introduction

This chapter looks at the application of transfer learning (TL). The experiments are designed to explore TL's impact across different temporal resolutions, different data availability scenarios, different feature inclusions and different models. A literature review on transfer learning is presented in Chapter 2. The rest of this Chapter is laid out as follows: the following subsection reveals how TL is applied and the details of the experiment design; then results and discussions are presented; finally, conclusions are drawn to complete the Chapter.

6.2. Methodology

This section gives an overview of how TL is applied in this thesis, via the form of a flowchart with a detailed description; then it presents the layout and parameters of the experiments that are designed to evaluate the performance of transfer learning across different scenarios.

6.2.1. Transfer Learning

In this section, target data refers to the data from the DMA to be forecasted, whilst source data refers to data from DMAs other than the target data DMA. The assumed possibility of cross-DMA data knowledge transfer is due to the nature of the data used. The data used in this thesis is 18 DMA datasets of short-term water demand data, collected by the same supply company, all at 15-minute temporal resolution.

Figure 6.1 is a flow chart that illustrates how TL-incorporated ML is applied and compared to traditional ML forecasting in this thesis. The TL approach employed starts with source and target data. The datasets are first transformed into samples of input and output pairs that can be used for ML training. The data is then pre-processed to eliminate all zero and negative values, as well as extremely high values based on visual inspection (detailed in Chapter 3, Section 4). The data is then aggregated into a different temporal resolution, i.e., original 15-minute demand and hourly demand.

After both source and target samples are temporally aggregated, training samples for traditional and TL-incorporated ML forecasting models are formed. An example of the training sample difference is given using DMA UK1 as target data, and the remaining DMAs as source data. First, UK1 data is split into training and testing samples (80% training and 20% testing); the 80% training samples from DMA UK1 are used to train the traditional ML model. The training samples from all other DMAs (source data) are added to the 80% samples from DMA UK1, to form a far larger training sample for the TL-incorporated ML model. The goal is to uncover positive knowledge that could be leveraged from DMAs UK2 to UK18, to assist with forecasting DMA UK1.

6.2.2. Machine Learning Models

The ML models employed for TL evaluation are XGB and LSTM. The reason for using these models are as follows: LSTM and NN are both network-based models, LSTM have shown superior forecasting accuracy; XGB and RF are both tree-based models, though both achieved similar results in terms of accuracy, XGB can be trained much faster. The theory and implementation of both models are

detailed in Chapter 5, under Sections 5.2.4 and 5.2.6. The implementation of both models remains the same in this chapter as they are in the previous chapter.

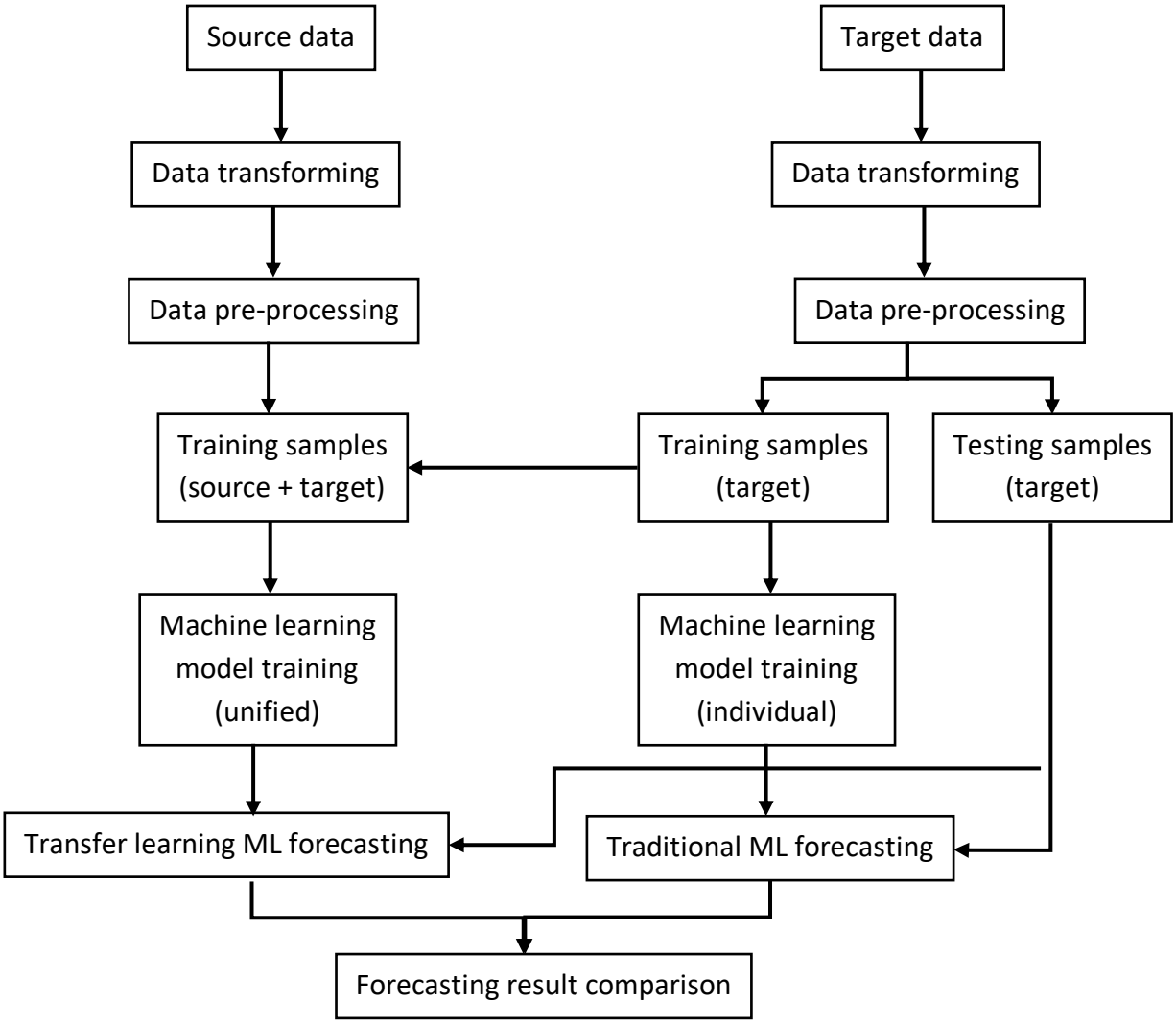

Figure 6.1 Flowchart of how Transfer Learning is applied

6.2.3. Experiment Set-up

To evaluate the impact TL has on ML forecasting, three experiments are designed to test the TL approach under different scenarios. All experiments are done across two temporal resolutions with different feature pairings. Table 6.1 details the temporal resolution and feature pairings. All experiment results are split into four performance categories, these are split between training and testing

data, between R^2 and RMSE. From here on, individual and unified forecasting will be used to respectively denote traditional ML models and TL-incorporated ML forecasting models.

Table 6.1 Temporal Resolution and feature pairings

Temporal resolution	Feature count	Label	Features
15 minutes	1	15-minute 1 feature	Past 1 point (continuous)
	12	15-minute 12 features	Past 12 points (continuous)
1 hour	2	Hourly 2 features	Past 2 points (continuous)
	2	Hourly 2* features	Past 1 point and past 24th point (discrete)
	24	Hourly 24 features	Past 24 points (continuous)

The first experiment will use XGB to exhaustively determine the impact of different amounts of source data inclusion on source forecasting. Source data are selected for each target DMA from the 17 other DMAs, and the number of source data included increases from 0 (represented by the individual model) up to 17 (where all other DMA data are included). The order of source DMA inclusion is based on the correlation between source and target data, with the most correlated DMAs included first. 80% of the target data will be used for individual model training, with the remaining 20% used for testing for all models. The unified model will be trained in two ways – with or without target training data (80%). The unified model trained with target data represents the impact source data has on target DMA forecasting when target data are abundant, and the unified model trained without target data represents how TL can be used in extreme cases where there

is no target data available. The training accuracies measure how well each model fits the samples supplied for model training, and the testing accuracy measures how well each model fits unknown target data. Finally, each model is retrained 10 times to show the result variance. The first experiment is done using only XGB because of the large number of model training and sample count in unified models, LSTM is far slower to train thus it is excluded from this experiment.

The second and third experiments will investigate the impact of source data selection. Both experiments will simulate scarce target data availability by limiting the target data training sample count to 100, 500 and 1000. The two experiments differ in the choice of source data. The second experiment will select source data in the same manner as the first experiment, where the source data is selected based on its correlation to target data, and the number of source DMAs kept is based on the result from the first experiment. The third experiment will select source data based on data quality, the uncorrupted periods (30 weeks) from the three DMAs (UK4, UK11, UK12) used throughout Chapters 4 and 5 will be used as ideal source data.

The individual models in the second and third experiments will be trained using limited target data samples, whilst the unified models will add source data samples to target training samples. Like the first experiment, the training accuracy will measure how well each model fits its given samples, and the testing accuracy will measure how well the models fit 20% of unknown target data. Both experiments will be done on LSTM and XGB.

6.3. Results and Discussion

This section presents the results of the TL application under different scenarios. The TL approach is applied to XGB and LSTM, using 18 UK DMA demand

datasets. Three experiments are designed to reveal 1) the ideal amount of external data; 2) how correlated source data impact target forecasting; 3) how high-quality source data impact target data forecasting; and 4) the best way to select source data to aid target data scarcity.

6.3.1. Source Data Class Determination

The first experiment uses XGB to draw an overview of the impact increasing source data amount has on target forecasting. The conclusions are drawn from two scenarios – 1) abundant target data; and 2) zero target data. Figure 6.1 shows the forecasts in different resolutions and feature pairings. There are 20 plots in Figure 6.1, and each plot contains two lines. The blue is the unified model with abundant target data, and the orange is the unified model with zero target data. Each point on the line is made of the average accuracy of 10 repeated forecasts of all 18 DMAs, under the two scenarios, with varying amounts of source data inclusion. The x-axis represents the number of source datasets incorporated, starting from 0, this represents the forecast accuracy of target data without any source data; the x-axis values increase up to 17, showing the impact increasing the number of source data has on target data forecasting accuracy. The x-axis value for the orange line starts at 1, as no forecast can be done without both source and target data training samples. The y-axis differs in each row, where each row corresponds to different accuracy measures, starting from the top, each row shows the result for training R^2, testing R^2, training RMSE and testing RMSE. The training result indicates how the models fit the training data samples, which differs for each group; and the testing result shows how the models forecast 20% of the unseen target data. The columns correspond to results from different resolutions and feature pairings, the column orders are the same as the labels shown in Table 6.1, starting from the left column.

The results in Figure 6.2 show that R^2 and RMSE are inversely related, thus they will be jointly discussed in the following. From the training results, the accuracy comparison between the two scenarios shows that training sample forecasting accuracy is dependent on the number of DMAs included as the accuracy from the orange line (zero target data) is one DMA behind the blue line (full target data). The training accuracy lines overlap beyond 4 to 8 DMA inclusions.

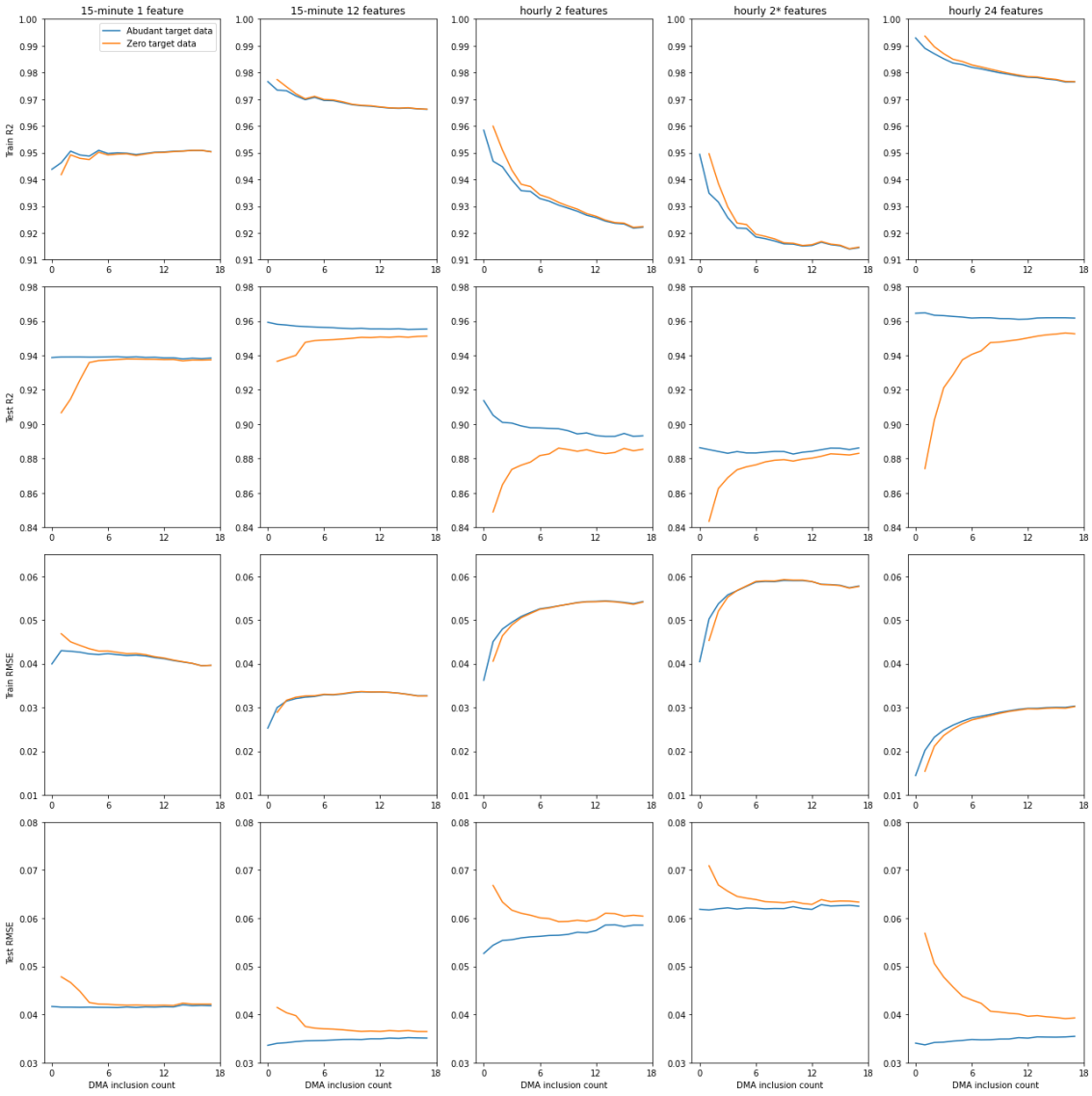

Figure 6.2 The impact of source data class on forecasting accuracy, using the XGB forecasting model

For scenario one, the testing forecast accuracy decreases with an increased number of source data. This is to be expected as the target forecasting should achieve the highest accuracy if trained using target data samples only. Additional

source data reduces the ratio of target data samples in the overall training set. However, the level of accuracy drop is minuscule (0.02 drop in R^2 for the largest drop), suggesting that additional source data inclusion has a minimal negative impact.

For scenario two, the testing forecast accuracy increases with an increased number of source data. However, the rate of increase and point of plateau differs for different temporal resolutions of forecasting. The 15-minute forecasts plateaus at four source datasets, whilst the hourly data plateaus at eight. The difference in the amount of source data requirement suggests that additional source data does positively compensate for reduced training sample size, as hourly demand has less than four times the sample count compared to the 15-minute demand. Additionally, the overall increase in accuracy when increasing source data amount has shown that poorly correlated source data can still improve target data forecasting accuracy, this emphasises the significance of source data quantity, when forecasting with minimal target data samples.

Further feature analysis from Figure 6.2 affirms the findings from Chapter 5, Section 1. The Section has shown that hourly demand forecasts require more features, compared to 15-minute demand forecasts, as the SHAP analysis of low-resolution data has achieved a lower Gini index, compared to the high-resolution result. Figure 6.2 shows that whilst reducing feature count to dominant features only has a limited impact on accuracy, the impact on hourly demand is more significant.

6.3.2. Source Data Length Determination

The second experiment aims to determine the minimum source data length required. Using the ideal number of source datasets, determined by the previous

experiment – four datasets for 15-minute demand and eight for hourly. The resultant plots for this experiment are presented in Figure 6.3 like experiment one. The y-axis in all plots is of accuracy measures, with each row corresponding to training R^2, testing R^2, training RMSE and testing RMSE; the columns represent the temporal resolution and feature pairing; the blue and orange lines respectively represent accuracy with and without target training samples.

The only difference is the x-axis, as the number of source datasets is set, the x-axis now represents the number of source data samples used. The number of samples evaluated ranges between 5,000 and 75,000, with 5,000 increments. In cases where the evaluated training sample count is lower than the available sample count, the number of samples is capped at the maximum number of available samples.

Like experiment one, experiment two is only performed on XGB, due to the slow model fitting exhibited by LSTM. The goal of the first two experiments is to optimise the sample class and size, for later experiments that use LSTM. Thus, reducing the need for repeated forecasts, and significantly reducing computation costs.

From Figure 6.3, the training result of forecasts with and without target data closely align with each other, with decreasing training accuracy (for both R^2 and RMSE) as the number of samples increased. This is to be expected as ML models can better fit smaller training samples compared to larger samples. For testing accuracies, however, the accuracy increases with increased training samples, but accuracy levels plateau at about 20,000 samples. Further accuracy gains can be achieved beyond this sample count, but further gains beyond 20,000 training

samples are insignificant. Thus, 20,000 samples will be used as a training sample size cap for the next experiment.

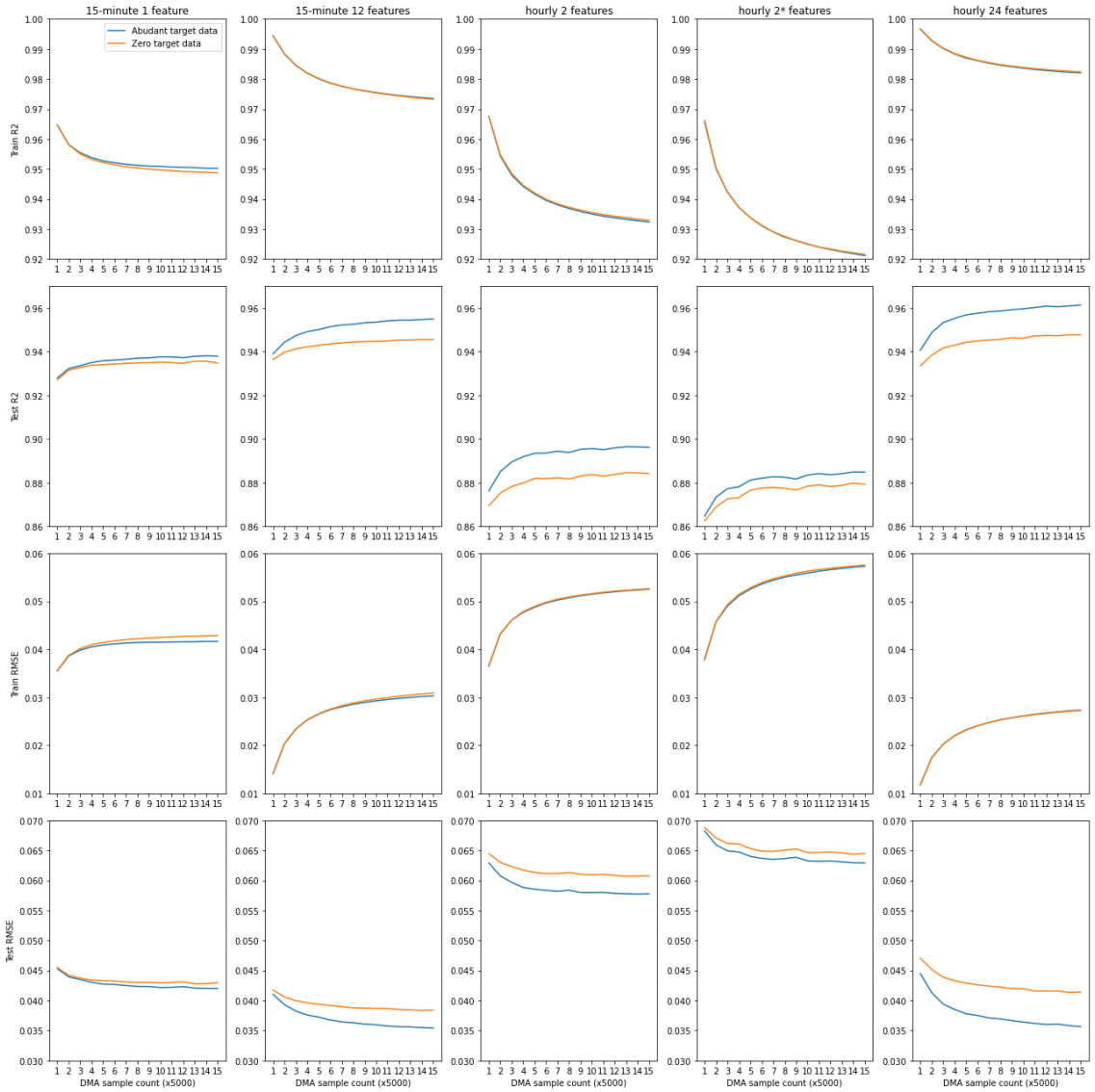

Figure 6.3 The impact of source data length on forecasting accuracy, using the XGB forecasting model

6.3.3. Correlation-based Source Data Inclusion

The first two experiments have determined the ideal training class and size – the ideal class has shown to be four and eight source datasets for 15-minute and hourly demand, respectively; and the source sample size larger than 20,000 have shown to have an insignificant impact on accuracy.

The third experiment employs the previous findings, to determine TL's performance when there is a limited number of target samples, using both XGB

and LSTM. The target sample availability is set to 1000, 500 and 100, to simulate diminished sample availability. The forecasts are made using both XGB and LSTM.

As previous experiments have proven models' ability to fit training data, the results for experiments three and four will focus on testing accuracy. Figures 6.4 and 6.5 show the testing accuracy for correlation-based source data inclusion, the figures are split by the testing accuracy indicators; Figure 6.4 shows the R^2 result for both XGB and LSTM, and Figure 6.5 shows RMSE. Within each figure, the columns correspond to resolution and feature pairing; the top three rows are for XGB forecast accuracies under three different data availability scenarios, and the bottom three rows are for LSTM forecast accuracies under three scenarios. All y-axis correspond to accuracy values, and the x-axis corresponds to different DMA representing target data.

The results from Figures 6.4 and 6.5 show that the R^2 and RMSE result agrees with each other, where higher R^2 correlates with lower RMSE. Thus, the two accuracy indicators will be jointly discussed. For both figures, the y-limits are the same for all plots, for ease of comparison between models and resolutions.

Within each plot, the box plots show the spread of forecast accuracies across 10 repeats, and the lines represent the mean accuracy across all DMAs. The blue results show TL-incorporated forecasts, and the red show traditional ML forecasts.

For forecasts with 100 target training samples (rows 1 and 4), the individual ML forecast repeats have shown that the forecasts are unstable. Although this is expected as the 100 samples are randomly selected from all available samples,

the difference in accuracy range between XGB and LSTM forecasts indicates that LSTM requires more training samples for consistent forecasts.

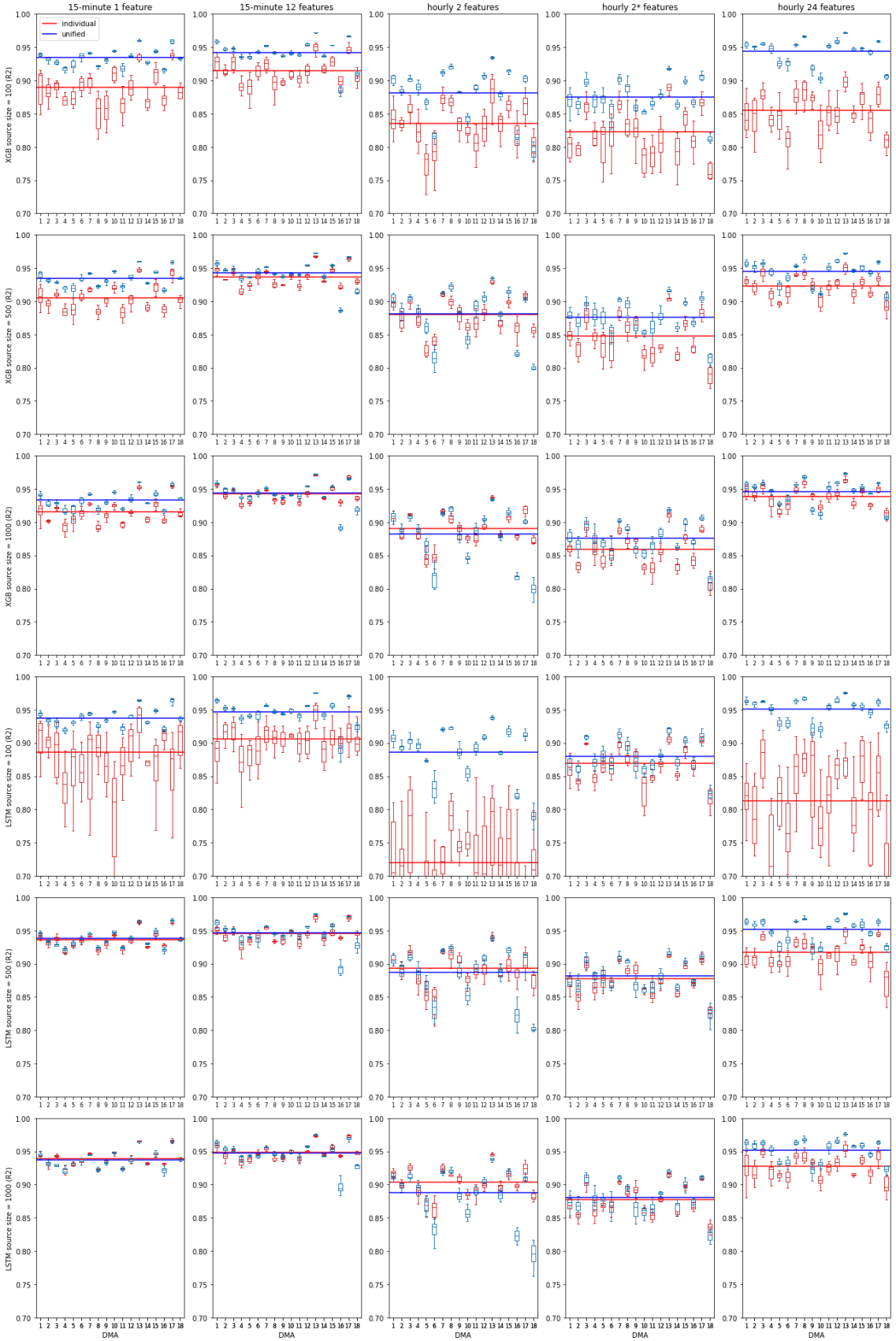

Figure 6.4 Transfer learning analysis using correlation-based source data inclusion (R2)

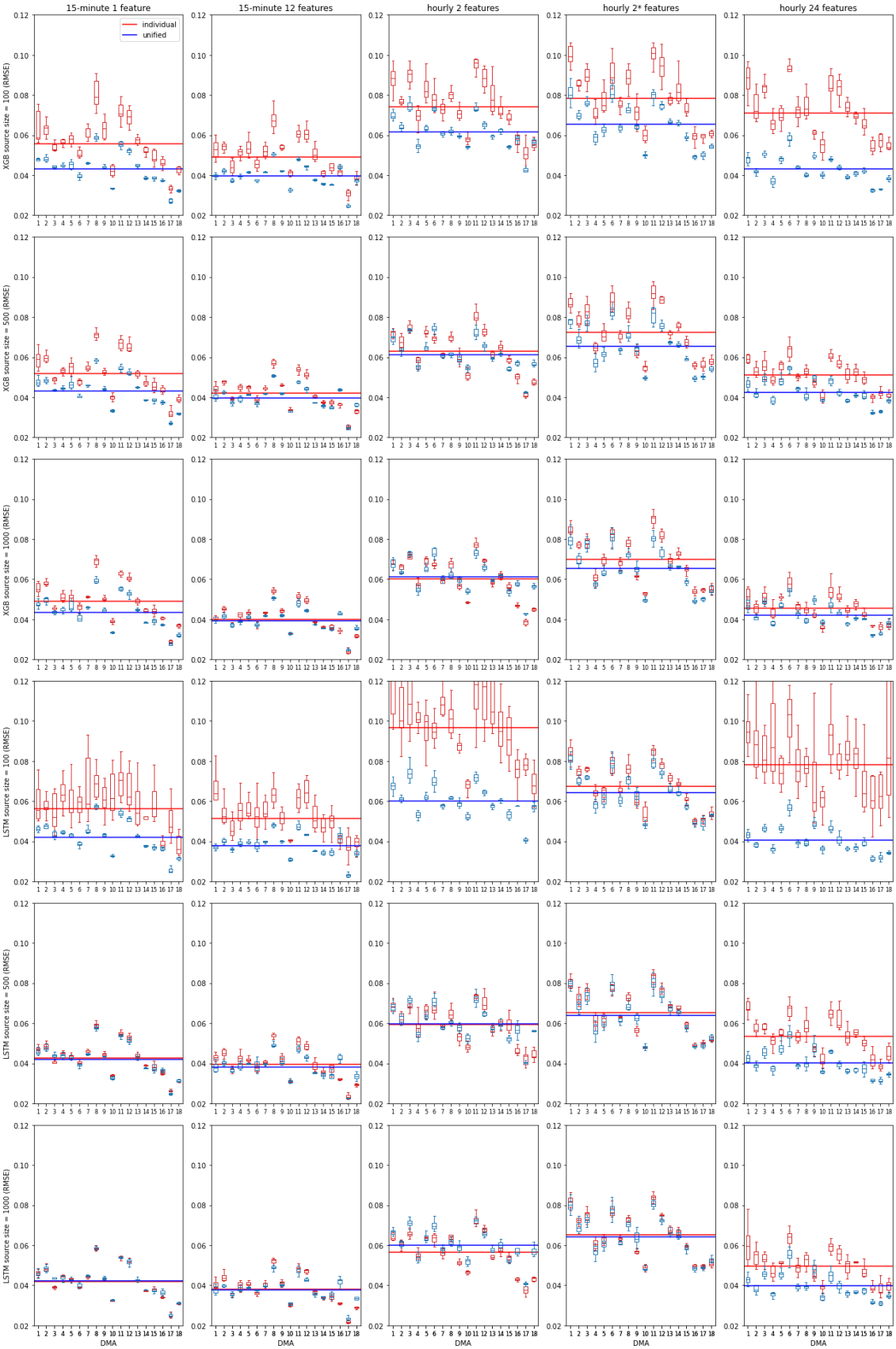

Figure 6.5 Transfer learning analysis using correlation-based source data inclusion (RMSE)

The correlation-based source data inclusion shows that TL-incorporated ML forecasts achieve comparable or higher forecasting accuracy compared to traditional ML models. The accuracy improvement is most evident for the 100 samples scenario and for the hourly demand forecast with 24 features. The improvement for the 100 samples scenario is intuitively understood as traditional ML samples cannot successfully train models with too little data. But for hourly demand forecasting with 24 features, the large accuracy improvement with abundant data with numerous features suggest that there is ample positive knowledge stored in external DMAs.

The comparison of feature impact for 15-minute demand forecasting shows that additional feature inclusion has limited impact on TL-incorporated ML models but has improved forecasting accuracy for traditional ML. This suggests that for 15-minute demand resolution, abundant features or training samples (even external samples) can both improve forecasting accuracy.

The impact of feature increase is more significant for hourly demand forecasting. The 24 feature forecasts have shown to be superior for both TL-incorporated and traditional ML models. The choice of which two features to include (continuous past two demand points or discrete past point and past 24^{th} point) have little impact on TL incorporated ML model, but traditional ML model have been shown to favour past two continuous points. For hourly demand, feature amount and training sample amount are both shown to have an impact on forecasting accuracy.

The comparison of target training sample availability shows that both models can make consistent forecasts with 500+ training samples. The increased amount of

target samples have zero impact on all TL-incorporated forecasting cases. The only exception is shown for hourly demand with 24 features.

For all TL-incorporated forecasting cases, the target training samples are first combined with all source training samples, and then 20,000 samples are selected for training from the large sample pool. As a result, the number of target samples that are selected for training varies, but the probability of target sample inclusion increases with the increase of available target samples. Though the increase of target samples for training should train models that fit the target group more closely, the result has shown target data increase has a limited impact on TL-incorporated forecast accuracy for most resolution cases. The only resolution where this impact is evident is forecasting hourly demand with 24 features. This suggests increasing target data is only useful when ample features are used. This affirms the usefulness of TL incorporation, as increasing features increase both computation cost and data quality requirement, in the event where both are limited, ample source data would be useful to improve forecasting target data.

6.3.4. Quality-based Source Data Inclusion

The previous experiment has selected source datasets based on their correlation to target data, whilst this has proven to be useful in improving accuracy, the selected source datasets have varying amounts of corrupted data. As discussed in Chapter 3, Section 4, only basic data cleaning is performed on all datasets, to only exclude zeroes, negatives, and extreme measures.

Amongst the DMAs, there exists a varying length of corrupted data that falls between zero and post-cleaning maximum. Examples of this are shown in Figure 3.3, where the black and red boxes pinpoint the periods. The black-boxed section shows the demand dropping significantly compared to the remaining sections,

whilst there may be a cause for this uniform drop, its variation from prior measurements nevertheless makes the section useless, once the measures are normalised. The red-boxed section shows an extended period where the demand remains consistent for several continuous points, and the whole corrupted section looks like a prolonged daily demand; the only explanation is that the measurement date for this period is off (i.e. the month and date values are switched).

Instead of delving into each DMA and fixing issues on a case-to-case basis, the alternative to correlation-based source data inclusion would be quality-based inclusion, where source data with no visible corruption are kept. The UK DMAs sections used in Chapters 4 and 5, namely – UK4, 11 and 12 are used for this experiment, and results will be compared to the previous experiment.

The final experiment is designed and presented the same as experiment three. Three limited target data scenarios are analysed (100, 500 and 1,000 samples). XGB and LSTM are both used for forecasting. Instead of picking top correlated source datasets, the first 30 weeks of demand from UK4, 11 and 12 are used as source training samples, in addition to the limited target samples. Like experiment three, a cap of 20,000 samples is placed on the unified training samples, to obtain comparable results between the two experiments. Figures 6.6 and 6.7 presents the results of the three limited target data scenarios. The layouts are identical to that of Figures 6.4 and 6.5.

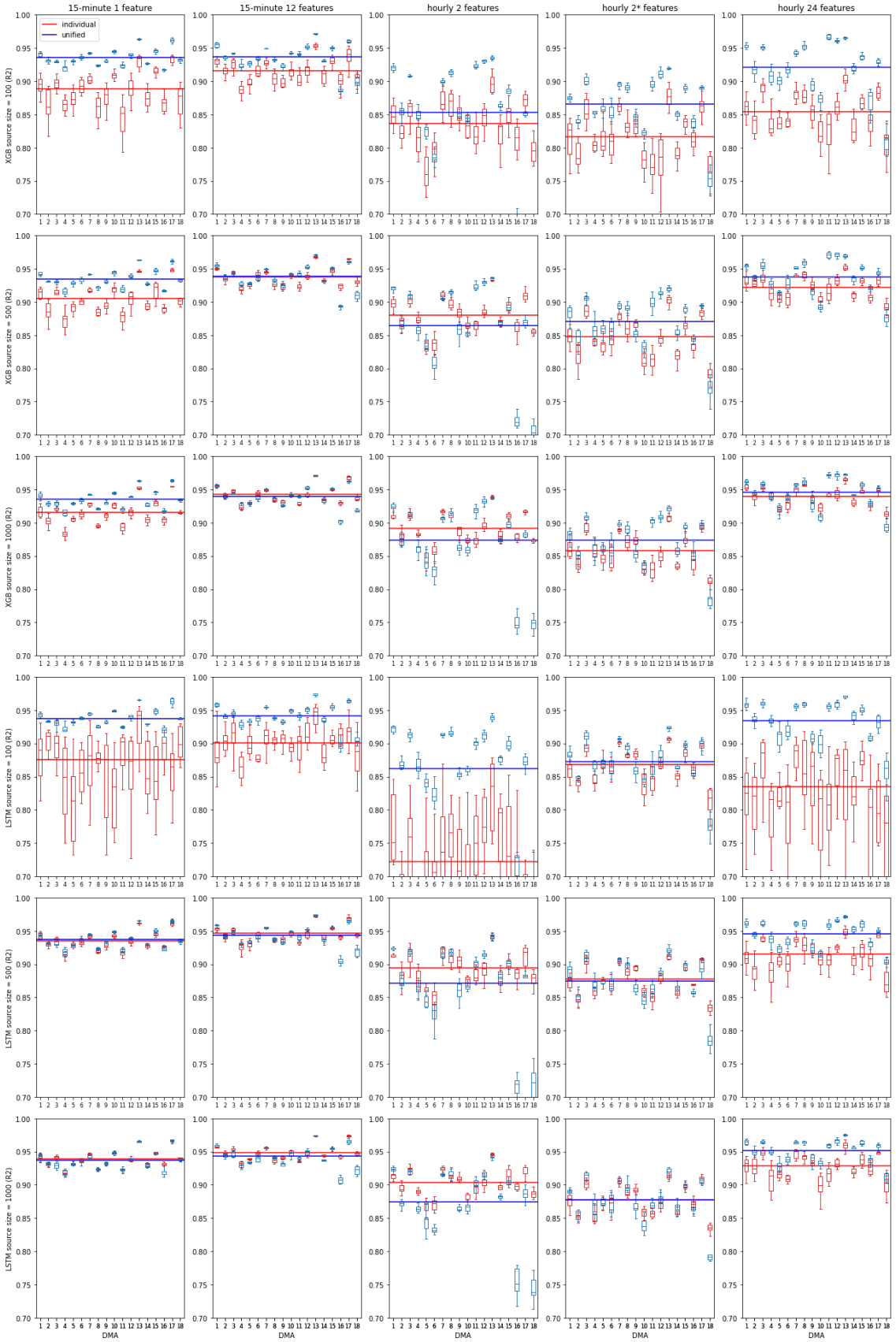

Figure 6.6 Transfer learning analysis using quality-based source data inclusion (R2)

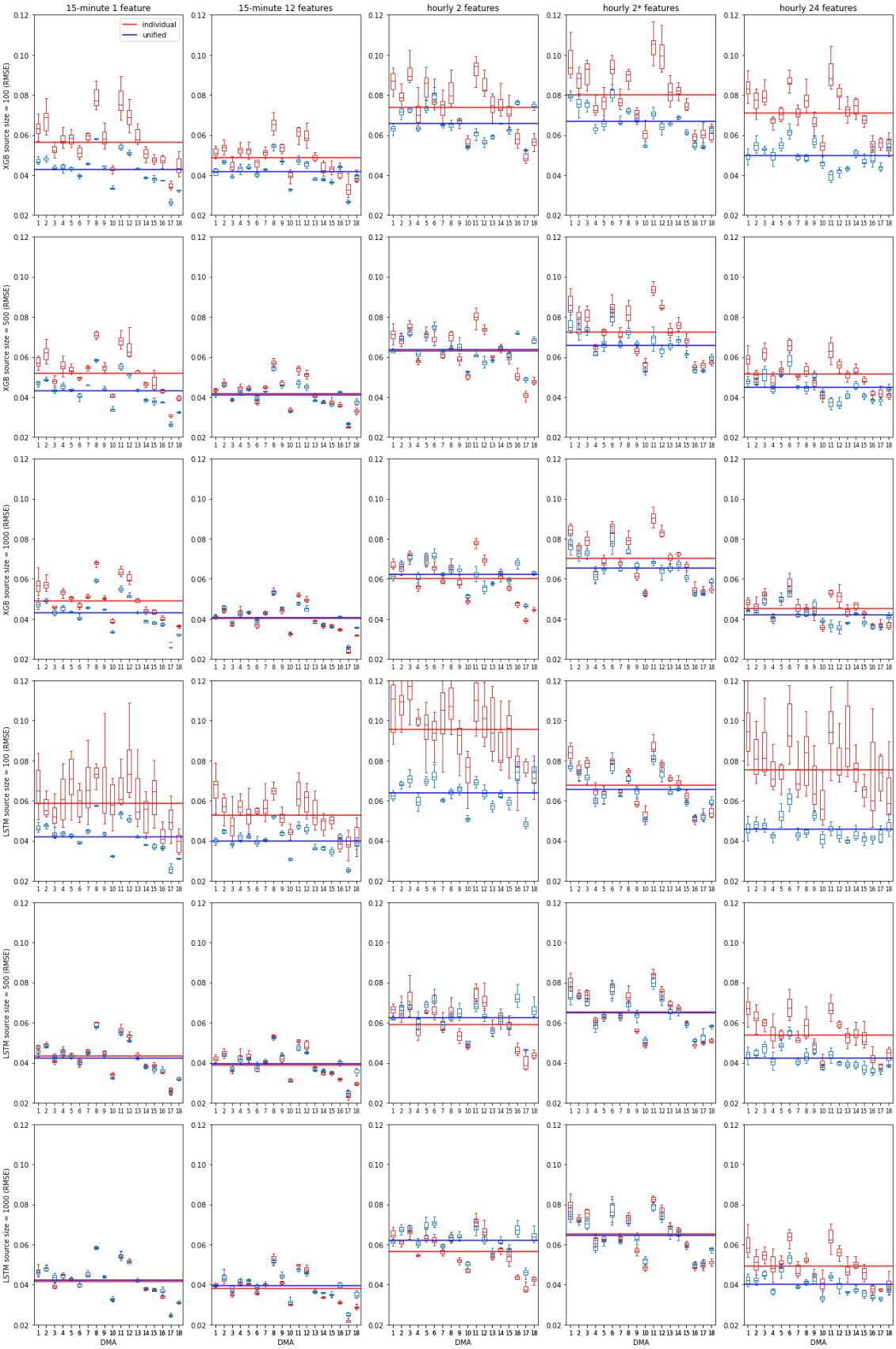

Figure 6.7 Transfer learning analysis using quality-based source data inclusion (RMSE)

The quality-based source data inclusion has shown similar results to that of correlation-based results from the previous sub-section. The only significant

difference is in the forecasting accuracy of hourly demand with the past two continuous features. The average accuracy is significantly lower for TL-incorporated ML models for quality-based source data inclusion compared to correlation-based inclusion. This is mainly lowered by DMA 16 and 18. Further visual analysis of demand patterns and correlation shows no discernible features for these two DMAs; the lowered accuracy could only be concluded as two outliers.

The average forecasting accuracy from quality-based source data inclusion is marginally lower compared to correlation-based selection. Though this slight accuracy drop means the difference between having 18 individual models and only having one model for all 18 DMAs. The former would prove useful when computation cost is of concern.

6.4. Summary

This chapter investigates the impact Transfer Learning has on short-term water demand forecasting. Tests are done on different models, temporal resolutions, and feature pairings. Different target data availability and source data choice are evaluated in detail. The key findings are:

External DMA past demand data can be used as training samples to successfully forecast demands at a specific DMA with no available training data. The accuracy of forecasts made with external training samples increases with an increasing number of external DMA datasets included. The level of accuracy improvements achieved with further DMA inclusion diminishes beyond four DMAs for 15-minute demand forecasts, and eight DMAs for hourly demand forecasts.

When forecasting with external DMA training samples, more data can increase accuracy, but the level of accuracy gained becomes insignificant when the training sample size increases beyond 20,000 samples.

TL-incorporated ML forecasting can help mitigate data availability issues. The performance of correlation-based and quality-based source data inclusion have both been shown to improve forecasting target data accuracy under limited data availability scenarios.

Accuracy comparison between correlation-based and quality-based source data inclusion shows that correlation-based source data choice achieves marginally better forecasting accuracy. Additionally, an increased number of target training samples has achieved a limited impact on accuracy for TL-incorporated ML forecasting models. Whilst quality-based source data inclusion is inferior, the reduction in forecasting accuracy is only marginal, and it can be used as a forecasting model across all DMAs. The resultant forecasting model could significantly reduce both data requirements and computation costs, as the quality-based TL incorporation would train one model for all DMAs, whereas correlation-based TL incorporation would train one model per DMA.

Chapter 7 - Conclusions

This work investigates the potential of data-centric forecasting approaches for short-term demand forecasting. The approach is evaluated against the commonly employed model-centric approach, and it has been tested from a variety of different aspects. This study has found that data-centric holds further potential, as there is untapped knowledge within available data. Two new techniques are also presented. The first employs machine learning (ML) model explainers to identify dominant features, thus reducing feature requirements, and subsequently computation costs. The second technique shows the possibility of applying Transfer Learning (TL) across different DMA forecasting, thus drastically reducing the model and data requirements from individual DMAs.

7.1. Summary

This section summarises all main experiments and analyses carried out during this work, and it outlines a brief overview, including the aim, method, and findings.

7.1.1. Data-centric water demand forecasting

Short-term water demand forecasting is vital to ensure urban supply quality and has the potential of improving leakage detection. In this work, four forecasting methods are tested to forecasting real short-term demand data, three well-established methods and one novel method. The application of these models and their optimal pairing with training data is considered. The results found that for short-term demand forecasting, near-term past demand played a more significant role, and long-term trends and seasonality had little impact. Consequently, models that consider longer-term seasonality, such as Prophet and ARIMA, have

shown inferior performance compared to models that focused on near-term demand impacts, such as NN and RF.

7.1.2. Machine learning explainability and feature importance

Data and computation requirements for demand forecasting can be significantly reduced by the presence of expert knowledge, in both the forecasting model and the data. However, whilst model employment can be generalised based on data type, the data input choices cannot be generalised so easily. The work carried out in Chapter 5 looks at the application of ML model explainers – SHAP and LIME. The explainers unveiled the inner workings of machine learning models, by presenting key features that are impacting results. The findings show that high-resolution data forecasting depends heavily on the past-point demand, for all models. Whilst for low-resolution data forecasting, tree-based models (RF and XGB) favour the demand from the past-point and past-day nearly equally, as evident from Figure 5.3, where the SHAP values for the past-point (-1) and past-day (-24) are near equal; network-based models (NN and LSTM) favour the past-point demand the most, though this favouritism is not as dominant as it is for tree-based models, and other near-term demand also have a significant impact on demand forecasting.

7.1.3. Transfer learning to tackle data scarcity

Data quality and availability can create unnecessary difficulties when forecasting univariate short-term water demand. The incorporation of external datasets has proven useful in the field of energy forecasting, its application in forecasting water demand is evaluated in Chapter 6. The result has shown that TL incorporation can help improve forecasting accuracy, in cases where there is zero or little target training data. Correlation-based and quality-based source data inclusion have

shown comparable accuracy improvements, though correlation-based inclusion is marginally superior in accuracy improvement, quality-based inclusion can result in significant computation cost savings.

7.2. Research Limitations and Recommendations

This research reveals the potential of optimising data usage for short-term water demand forecasting. Whilst useful findings and conclusions are drawn, the work can be extended in future research projects.

First, a limited number of forecasting models are reviewed in this thesis, and the methods employed focus on more efficient use of existing data. Further work could extend the range of forecasting models, particularly to evaluate alternative deep learning models. The impact of explainer modules and Transfer Learning can be tested on deep learning models that are optimised by hyperparameter tuning.

Second, the forecasting models in this thesis are trained solely using past demand data. Whilst the results in this thesis and from other research have shown this to be sufficient for short-term water demand forecasting, the impact of alternative data (non-demand data) on accuracy can be numerically determined via machine learning model explainers.

Third, whilst the models have shown to make accurate forecasts, the work focuses on the theoretical potential of univariate short-term demand forecasting, without considering other types of demand forecasting which are faced by water utilities in practice. One major example for this is how one past point is sufficient feature for high-resolution demand forecasting. Whilst the results have shown good accuracy levels, common sense was overlooked in the face of high

accuracy. In some cases, one past point might not be suitable since it gives ML models no indication of the direction of travel following the one point, regardless of the accuracy level.

Fourth, the findings from the TL forecasting approaches show promise in the technique's employment in water demand forecasting. However, the method can be further validated with more data. Whilst 18 DMA demand data are used, they belonged to the same area. Further testing can see the same method applied to separate DMA cases, or if there is an abundance of DMA datasets, clustering techniques can be used to separate DMA into groups.

Finally, the usefulness of ML explainers and ML can be evaluated by applying both methods jointly for particular use cases, e.g., operation planning and leakage detection cases. The goal of making accurate forecasting for short-term water demand data is to improve supply operation or improve leakage detection. The improved forecasting methods can be jointly used as a first step to operation improvement purposes, the resultant accuracy gain can be compared to cost reduction, to determine the significance of forecasting accuracy improvement.

Overall, the work in this thesis can be expanded and improved with more data, finer tuned models, and better consideration on practical implication of result. Future work needs to exercise a balance between data-centric and model-centric approaches.

7.3. Research implementation

The work presented in this thesis can help current industry in various ways. The UK Water Industry Research have listed 11 big questions that are challenging the

current water industry, better use of data and improved forecasting accuracy of water demand can help with reducing leakage and interruptions.

The recognition of the essential nature of data in this thesis should increase its industry awareness, and thus, dedicate more resources to data capture, storage, and utilisation. To make demand forecasts accurately and efficiently, data monitoring is the first step. In the UK, smart water meter role out has been done by most major water utilities. The storage of mass consumer data needs to be done correctly, to both be secure for privacy reasons, and be accessible by people that needs it; most water utilities hold data in dated systems, guarded by IT departments, where accessibility is difficult and inefficient. More up to date methods of storage are being implemented in recent years, and with the recognition of data importance, extensive data utilisation across the industry would improve water supply, wastewater treatment, and asset management, for a sustainable water future.

References

Acuña, L. G., D. R. Ríos, C. P. Arboleda, and E. G. Ponzón. 2018. "Cooperation model in the electricity energy market using bi-level optimization and Shapley value." *Operations Research Perspectives*, 5: 161–168. Elsevier Ltd. https://doi.org/10.1016/j.orp.2018.07.003.

Adak, A., B. Pradhan, N. Shukla, and A. Alamri. 2022. "Unboxing Deep Learning Model of Food Delivery Service Reviews Using Explainable Artificial Intelligence (XAI) Technique." *Foods*, 11 (14): 2019. https://doi.org/10.3390/foods11142019.

Adamowski, J. F. 2008. "Peak Daily Water Demand Forecast Modeling Using Artificial Neural Networks." *J Water Resour Plan Manag*, 134 (2): 119–128. https://doi.org/10.1061/(ASCE)0733-9496(2008)134:2(119).

Adamowski, J., H. Fung Chan, S. O. Prasher, B. Ozga-Zielinski, and A. Sliusarieva. 2012. "Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada." *Water Resour Res*, 48 (1). https://doi.org/10.1029/2010WR009945.

Aguilera, H., C. Guardiola-Albert, N. Naranjo-Fernández, and C. Kohfahl. 2019. "Towards flexible groundwater-level prediction for adaptive water management: using Facebook's Prophet forecasting approach." *Hydrological Sciences Journal*, 64 (12): 1504–1518. Taylor and Francis Ltd. https://doi.org/10.1080/02626667.2019.1651933.

Ali, M., R. Prasad, Y. Xiang, and Z. M. Yaseen. 2020. "Complete ensemble empirical mode decomposition hybridized with random forest and kernel ridge regression model for monthly rainfall forecasts." *J Hydrol (Amst)*, 584: 124647. Elsevier B.V. https://doi.org/10.1016/j.jhydrol.2020.124647.

Awad, M., and M. Zaid-Alkelani. 2019. "Prediction of Water Demand Using Artificial Neural Networks Models and Statistical Model." *International Journal of Intelligent Systems and Applications*, 11 (9): 40–55. https://doi.org/10.5815/ijisa.2019.09.05.

Azzuri, F., L. S. Darfiansa, R. S. Grananta, A. Permatasari, and N. Yudistira. 2022. "Explainable AI Prediction of Cooking Oil Prices Over Time." *7th International Conference on Sustainable Information Engineering and Technology 2022*, 47–56. New York, NY, USA: ACM.

Bakker, M., J. H. G. Vreeburg, K. M. van Schagen, and L. C. Rietveld. 2013. "A fully adaptive forecasting model for short-term drinking water demand." *Environmental Modelling & Software*, 48: 141–151. https://doi.org/10.1016/j.envsoft.2013.06.012.

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. 2020. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." *Information Fusion*, 58: 82–115. Elsevier B.V. https://doi.org/10.1016/j.inffus.2019.12.012.

Bassi, A., A. Shenoy, A. Sharma, H. Sigurdson, C. Glossop, and J. H. Chan. 2021. "Building Energy Consumption Forecasting: A Comparison of Gradient Boosting Models." *The 12th*

International Conference on Advances in Information Technology, 1–9. New York, NY, USA: ACM.

Bata, M., R. Carriveau, and D. S.-K. Ting. 2020. "Short-term water demand forecasting using hybrid supervised and unsupervised machine learning model." *Smart Water*, 5 (1): 2. Springer Science and Business Media LLC. https://doi.org/10.1186/s40713-020-00020-y.

Białek, J., W. Bujalski, K. Wojdan, M. Guzek, and T. Kurek. 2022. "Dataset level explanation of heat demand forecasting ANN with SHAP." *Energy*, 261: 125075. Elsevier Ltd. https://doi.org/10.1016/j.energy.2022.125075.

Böse, J.-H., V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, and Y. Wang. 2017. "Probabilistic demand forecasting at scale." *Proceedings of the VLDB Endowment*, 10 (12): 1694–1705. https://doi.org/10.14778/3137765.3137775.

Boudhaouia, A., and P. Wira. 2021. "A Real-Time Data Analysis Platform for Short-Term Water Consumption Forecasting with Machine Learning." *Forecasting*, 3 (4): 682–694. MDPI. https://doi.org/10.3390/forecast3040042.

Box E. P. G., Jenkins G. M., Reinsel G. C., and Ljung G. M. 2015. *Time Series Analysis: Forecasting and Control*. Hoboken, New Jersey: Wiley Blackwell.

Breiman, L. 2001. "Random Forests." *Mach Learn*, 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Bussmann, N., P. Giudici, D. Marinelli, and J. Papenbrock. 2020. "Explainable AI in Fintech Risk Management." *Front Artif Intell*, 3. Frontiers Media S.A. https://doi.org/10.3389/frai.2020.00026.

Chen, G., T. Long, J. Xiong, and Y. Bai. 2017. "Multiple Random Forests Modelling for Urban Water Consumption Forecasting." *Water Resources Management*, 31 (15): 4715–4729. Springer Netherlands. https://doi.org/10.1007/s11269-017-1774-7.

Chen, J., and D. L. Boccelli. 2018. "Forecasting Hourly Water Demands With Seasonal Autoregressive Models for Real-Time Application." *Water Resour Res*, 54 (2): 879–894. https://doi.org/10.1002/2017WR022007.

Chen, T., and C. Guestrin. 2016. "XGBoost." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. New York, NY, USA: ACM.

Chen, Y., Z. Tong, Y. Zheng, H. Samuelson, and L. Norford. 2020. "Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings." *J Clean Prod*, 254: 119866. Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2019.119866.

Chen, Z., H. Xu, P. Jiang, S. Yu, G. Lin, I. Bychkov, A. Hmelnov, G. Ruzhnikov, N. Zhu, and Z. Liu. 2021. "A transfer Learning-Based LSTM strategy for imputing Large-Scale consecutive missing data and its application in a water quality prediction system." *J Hydrol (Amst)*, 602: 126573. Elsevier B.V. https://doi.org/10.1016/j.jhydrol.2021.126573.

Chu, Y., P. Xu, M. Li, Z. Chen, Z. Chen, Y. Chen, and W. Li. 2020. "Short-term metropolitan-scale electric load forecasting based on load decomposition and ensemble algorithms." *Energy Build*, 225: 110343. Elsevier Ltd. https://doi.org/10.1016/j.enbuild.2020.110343.

Cutore, P., A. Campisano, Z. Kapelan, C. Modica, and D. Savic. 2008. “Probabilistic prediction of urban water consumption using the SCEM-UA algorithm.” *Urban Water J*, 5 (2): 125–132. https://doi.org/10.1080/15730620701754434.

DeepLearningAI. 2021. “A Chat with Andrew on MLOps: From Model-centric to Data-centric AI.” YouTube.

DeepLearning.AI and Landing AI. 2021. “Data-Centric AI Competition.”

Ding, C., and H. Peng. 2005. “MINIMUM REDUNDANCY FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA.” *J Bioinform Comput Biol*, 03 (02): 185–205. https://doi.org/10.1142/S0219720005001004.

Donkor, E. A., T. A. Mazzuchi, R. Soyer, and J. Alan Roberson. 2014. “Urban Water Demand Forecasting: Review of Methods and Models.” *J Water Resour Plan Manag*, 140 (2): 146–159. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000314.

Faeldon, J., K. Espana, and D. J. Sabido. 2014. “Data-centric HPC for Numerical Weather Forecasting.” *2014 43rd International Conference on Parallel Processing Workshops*, 79–84. IEEE.

Fan, C., Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, and Y. C. Tseng. 2020. “Statistical investigations of transfer learning-based methodology for short-term building energy predictions.” *Appl Energy*, 262: 114499. Elsevier Ltd. https://doi.org/10.1016/j.apenergy.2020.114499.

Frank A. Farris. 2010. “The Gini Index and Measures of Inequality.” *The American Mathematical Monthly*, 117 (10): 851. https://doi.org/10.4169/000298910x523344.

Fu, G., Y. Jin, S. Sun, Z. Yuan, and D. Butler. 2022. “The role of deep learning in urban water management: A critical review.” *Water Res*, 223: 118973. Elsevier Ltd. https://doi.org/10.1016/j.watres.2022.118973.

Fullerton, T. M., and A. L. Molina. 2010. “Municipal water consumption forecast accuracy.” *Water Resour Res*, 46 (6). https://doi.org/10.1029/2009WR008450.

Gagliardi, F., S. Alvisi, Z. Kapelan, and M. Franchini. 2017. “A Probabilistic Short-Term Water Demand Forecasting Model Based on the Markov Chain.” *Water (Basel)*, 9 (7): 507. MDPI AG. https://doi.org/10.3390/w9070507.

Gao, Y., Y. Ruan, C. Fang, and S. Yin. 2020. “Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data.” *Energy Build*, 223: 110156. Elsevier Ltd. https://doi.org/10.1016/j.enbuild.2020.110156.

Garreau, D., and U. von Luxburg. 2020. “Explaining the Explainer: A First Theoretical Analysis of LIME.”

Géron, A. 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. *O’Reilly Media*.

Gers, F. A., J. Schmidhuber, and F. Cummins. 2000. “Learning to Forget: Continual Prediction with LSTM.” *Neural Comput*, 12 (10): 2451–2471. https://doi.org/10.1162/089976600300015015.

Ghiassi, M., D. K. Zimbra, and H. Saidane. 2008. “Urban Water Demand Forecasting with a Dynamic Artificial Neural Network Model.” *J Water Resour Plan Manag*, 134 (2): 138–146. https://doi.org/10.1061/(ASCE)0733-9496(2008)134:2(138).

Goodfellow, I., Y. Benigo, and Courville Aaron. 2016. “Deep Learning (Adaptive Computation and Machine Learning series): Ian Goodfellow, Yoshua Bengio, Aaron Courville: 9780262035613: Amazon.com: Books.” *MIT Press*.

Grover, A., A. Kapoor, and E. Horvitz. 2015. “A Deep Hybrid Model for Weather Forecasting.” *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 379–386. New York, NY, USA: ACM.

Gumus, M., and M. S. Kiran. 2017. “Crude oil price forecasting using XGBoost.” *2017 International Conference on Computer Science and Engineering (UBMK)*, 1100–1103. IEEE.

Guo, G., and S. Liu. 2018. *Short-term water demand forecast based on deep neural network*.

Guo, G., S. Liu, Y. Wu, J. Li, R. Zhou, and X. Zhu. 2018. “Short-Term Water Demand Forecast Based on Deep Learning Method.” *J Water Resour Plan Manag*, 144 (12): 04018076. American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000992.

Gupta, A. K., V. Singh, P. Mathur, and C. M. Travieso-Gonzalez. 2021. “Prediction of COVID-19 pandemic measuring criteria using support vector machine, prophet and linear regression models in Indian scenario.” *Journal of Interdisciplinary Mathematics*, 24 (1): 89–108. Taru Publications. https://doi.org/10.1080/09720502.2020.1833458.

Gupta, H. V., H. Kling, K. K. Yilmaz, and G. F. Martinez. 2009. “Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling.” *J Hydrol (Amst)*, 377 (1–2): 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003.

Herrera, M., L. Torgo, J. Izquierdo, and R. Pérez-García. 2010. “Predictive models for forecasting hourly urban water demand.” *J Hydrol (Amst)*, 387 (1–2): 141–150. https://doi.org/10.1016/j.jhydrol.2010.04.005.

Ibrahem Ahmed Osman, A., A. Najah Ahmed, M. F. Chow, Y. Feng Huang, and A. El-Shafie. 2021. “Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia.” *Ain Shams Engineering Journal*, 12 (2): 1545–1556. Ain Shams University. https://doi.org/10.1016/j.asej.2020.11.011.

Jain, D. A., U. C. Joshi, and A. K. Varshney. 2000. “Short-term water demand forecasting using artificial neural networks: IIT Kanpur experience.” *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 459–462. IEEE Comput. Soc.

Kang, Y., E. Spiliotis, F. Petropoulos, N. Athiniotis, F. Li, and V. Assimakopoulos. 2021. “Déjà vu: A data-centric forecasting approach through time series cross-similarity.” *J Bus Res*, 132: 719–731. Elsevier Inc. https://doi.org/10.1016/j.jbusres.2020.10.051.

Karb, T., N. Kühl, R. Hirt, and V. Glivici-Cotruta. 2020. “A network-based transfer learning approach to improve sales forecasting of new products.”

Khaidem, L., S. Saha, and S. R. Dey. 2016. “Predicting the direction of stock market prices using random forest.”

Khwaja, A. S., A. Anpalagan, M. Naeem, and B. Venkatesh. 2020. “Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting.” *Electric Power Systems Research*, 179: 106080. Elsevier Ltd. https://doi.org/10.1016/j.epsr.2019.106080.

Kim, J.-Y., and S.-B. Cho. 2020. “Electric Energy Demand Forecasting with Explainable Time-series Modeling.” *2020 International Conference on Data Mining Workshops (ICDMW)*, 711–716. IEEE.

Kim, J.-Y., and S.-B. Cho. 2021. “Explainable prediction of electric energy demand using a deep autoencoder with interpretable latent space.” *Expert Syst Appl*, 186: 115842. Elsevier Ltd. https://doi.org/10.1016/j.eswa.2021.115842.

Kimura, N., I. Yoshinaga, K. Sekijima, I. Azechi, and D. Baba. 2019. “Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions.” *Water (Basel)*, 12 (1): 96. MDPI AG. https://doi.org/10.3390/w12010096.

Kuzlu, M., U. Cali, V. Sharma, and O. Guler. 2020. “Gaining Insight Into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools.” *IEEE Access*, 8: 187814–187823. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2020.3031477.

Kvalseth, T. O. 1985. “Cautionary Note about R 2.” *Am Stat*, 39 (4): 279. https://doi.org/10.2307/2683704.

Le, T., M. T. Vo, T. Kieu, E. Hwang, S. Rho, and S. W. Baik. 2020. “Multiple Electric Energy Consumption Forecasting Using a Cluster-Based Strategy for Transfer Learning in Smart Building.” *Sensors*, 20 (9): 2668. MDPI AG. https://doi.org/10.3390/s20092668.

Lertpalangsunti, N., C. W. Chan, R. Mason, and P. Tontiwachwuthikul. 1999. “A toolset for construction of hybrid intelligent forecasting systems: application for water demand prediction.” *Artificial Intelligence in Engineering*, 13 (1): 21–42. https://doi.org/10.1016/S0954-1810(98)00008-9.

Li, L., J. Qiao, G. Yu, L. Wang, H.-Y. Li, C. Liao, and Z. Zhu. 2022a. “Interpretable tree-based ensemble model for predicting beach water quality.” *Water Res*, 211: 118078. Elsevier Ltd. https://doi.org/10.1016/j.watres.2022.118078.

Li, Z., C. Zhang, H. Liu, C. Zhang, M. Zhao, Q. Gong, and G. Fu. 2022b. “Developing stacking ensemble models for multivariate contamination detection in water distribution systems.” *Science of The Total Environment*, 828: 154284. Elsevier B.V. https://doi.org/10.1016/j.scitotenv.2022.154284.

Liu, W., W. D. Liu, and J. Gu. 2020. “Predictive model for water absorption in sublayers using a Joint Distribution Adaption based XGBoost transfer learning method.” *J Pet Sci Eng*, 188: 106937. Elsevier B.V. https://doi.org/10.1016/j.petrol.2020.106937.

Liu, X., Y. Zhang, and Q. Zhang. 2022. “Comparison of EEMD-ARIMA, EEMD-BP and EEMD-SVM algorithms for predicting the hourly urban water consumption.” *Journal of Hydroinformatics*, 24 (3): 535–558. https://doi.org/10.2166/hydro.2022.146.

Liu, Z., X. Wang, Q. Zhang, and C. Huang. 2019. “Empirical mode decomposition based hybrid ensemble model for electrical energy consumption forecasting of the cement grinding process.” *Measurement*, 138: 314–324. Elsevier B.V. https://doi.org/10.1016/j.measurement.2019.02.062.

Lu, H., F. Cheng, X. Ma, and G. Hu. 2020. “Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower.” *Energy*, 203: 117756. Elsevier Ltd. https://doi.org/10.1016/j.energy.2020.117756.

Lundberg, S., and S.-I. Lee. 2017. “A Unified Approach to Interpreting Model Predictions.”

McCulloch, W. S., and W. Pitts. 1943. “A logical calculus of the ideas immanent in nervous activity.” *Bull Math Biophys*, 5 (4): 115–133. https://doi.org/10.1007/BF02478259.

Menculini, L., A. Marini, M. Proietti, A. Garinei, A. Bozza, C. Moretti, and M. Marconi. 2021. “Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices.” *Forecasting*, 3 (3): 644–662. https://doi.org/10.3390/forecast3030040.

Mookherjee, D., and A. Shorrocks. 1982. “A Decomposition Analysis of the Trend in UK Income Inequality.” *The Economic Journal*, 92 (368): 886. https://doi.org/10.2307/2232673.

Moon, J., S. Rho, and S. W. Baik. 2022. “Toward explainable electrical load forecasting of buildings: A comparative study of tree-based ensemble methods with Shapley values.” *Sustainable Energy Technologies and Assessments*, 54: 102888. Elsevier Ltd. https://doi.org/10.1016/j.seta.2022.102888.

Mu, L., F. Zheng, R. Tao, Q. Zhang, and Z. Kapelan. 2020. “Hourly and Daily Urban Water Demand Predictions Using a Long Short-Term Memory Based Model.” *J Water Resour Plan Manag*, 146 (9). American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(ASCE)WR.1943-5452.0001276.

Nasser, A. A., M. Z. Rashad, and S. E. Hussein. 2020. “A Two-Layer Water Demand Prediction System in Urban Areas Based on Micro-Services and LSTM Neural Networks.” *IEEE Access*, 8: 147647–147661. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2020.3015655.

Nie, P., M. Roccotelli, M. P. Fanti, Z. Ming, and Z. Li. 2021. “Prediction of home energy consumption based on gradient boosting regression tree.” *Energy Reports*, 7: 1246–1255. Elsevier Ltd. https://doi.org/10.1016/j.egyr.2021.02.006.

Papacharalampous, G. A., and H. Tyralis. 2018. “Evaluation of random forests and Prophet for daily streamflow forecasting.” *Advances in Geosciences*, 45: 201–208. Copernicus GmbH. https://doi.org/10.5194/adgeo-45-201-2018.

Parmar, J., P. Das, and S. M. Dave. 2021. “A machine learning approach for modelling parking duration in urban land-use.” *Physica A: Statistical Mechanics and its Applications*, 572: 125873. Elsevier B.V. https://doi.org/10.1016/j.physa.2021.125873.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. 2012. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 12: 2825–2830.

Peng, L., H. Wu, M. Gao, H. Yi, Q. Xiong, L. Yang, and S. Cheng. 2022. “TLT: Recurrent fine-tuning transfer learning for water quality long-term prediction.” *Water Res*, 225: 119171. Elsevier Ltd. https://doi.org/10.1016/j.watres.2022.119171.

Pu, Z., J. Yan, L. Chen, Z. Li, W. Tian, T. Tao, and K. Xin. 2023. “A hybrid Wavelet-CNN-LSTM deep learning model for short-term urban water demand forecasting.” *Front Environ Sci Eng*, 17 (2): 22. Higher Education Press Limited Company. https://doi.org/10.1007/s11783-023-1622-3.

Rahimzad, M., A. Moghaddam Nia, H. Zolfonoon, J. Soltani, A. Danandeh Mehr, and H.-H. Kwon. 2021. “Performance Comparison of an LSTM-based Deep Learning Model versus

Conventional Machine Learning Algorithms for Streamflow Forecasting." *Water Resources Management*, 35 (12): 4167–4187. Springer Science and Business Media B.V. https://doi.org/10.1007/s11269-021-02937-w.

Rahman, A. T. M. S., T. Hosono, O. Kisi, B. Dennis, and A. H. M. R. Imon. 2020. "A minimalistic approach for evapotranspiration estimation using the Prophet model." *Hydrological Sciences Journal*, 65 (12): 1994–2006. Taylor and Francis Ltd. https://doi.org/10.1080/02626667.2020.1787416.

Ribeiro, M., K. Grolinger, H. F. ElYamany, W. A. Higashino, and M. A. M. Capretz. 2018. "Transfer learning with seasonal and trend adjustment for cross-building energy forecasting." *Energy Build*, 165: 352–363. Elsevier Ltd. https://doi.org/10.1016/j.enbuild.2018.01.034.

Ribeiro, M. H. D. M., and L. dos Santos Coelho. 2020. "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series." *Appl Soft Comput*, 86: 105837. Elsevier Ltd. https://doi.org/10.1016/j.asoc.2019.105837.

Ribeiro, M. T. 2021. *lime Documentation Release 0.1*.

Sagi, O., and L. Rokach. 2018. "Ensemble learning: A survey." *WIREs Data Mining and Knowledge Discovery*, 8 (4). Wiley-Blackwell. https://doi.org/10.1002/widm.1249.

Sajja, S., N. Aggarwal, S. Mukherjee, K. Manglik, S. Dwivedi, and V. Raykar. 2021. "Explainable AI based Interventions for Pre-season Decision Making in Fashion Retail." *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, 281–289. New York, NY, USA: ACM.

Sardinha-Lourenço, A., A. Andrade-Campos, A. Antunes, and M. S. Oliveira. 2018. "Increased performance in the short-term water demand forecasting through the use of a parallel adaptive weighting strategy." *J Hydrol (Amst)*, 558: 392–404. Elsevier B.V. https://doi.org/10.1016/j.jhydrol.2018.01.047.

Sarmas, E., N. Dimitropoulos, V. Marinakis, Z. Mylona, and H. Doukas. 2022. "Transfer learning strategies for solar power forecasting under data scarcity." *Sci Rep*, 12 (1): 14643. Nature Research. https://doi.org/10.1038/s41598-022-18516-x.

Sauer, J., V. C. Mariani, L. dos Santos Coelho, M. H. D. M. Ribeiro, and M. Rampazzo. 2022. "Extreme gradient boosting model based on improved Jaya optimizer applied to forecasting energy consumption in residential buildings." *Evolving Systems*, 13 (4): 577–588. Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/s12530-021-09404-2.

Seabold, S., and J. Perktold. 2010. "Statsmodels: Econometric and statistical modeling with python." *Proceedings of the 9th Python in Science Conference*, 10–25080. Austin, TX.

Taylor, S. J., and B. Letham. 2018. "Forecasting at Scale." *Am Stat*, 72 (1): 37–45. https://doi.org/10.1080/00031305.2017.1380080.

Tiwari, M., J. Adamowski, and K. Adamowski. 2016. "Water demand forecasting using extreme learning machines." *Journal of Water and Land Development*, 28 (1): 37–52. Institute for Land Reclamation and Grassland Farming. https://doi.org/10.1515/jwld-2016-0004.

Tiwari, M. K., and J. Adamowski. 2013. “Urban water demand forecasting and uncertainty assessment using ensemble wavelet-bootstrap-neural network models.” *Water Resour Res*, 49 (10): 6486–6507. https://doi.org/10.1002/wrcr.20517.

Toharudin, T., R. S. Pontoh, R. E. Caraka, S. Zahroh, Y. Lee, and R. C. Chen. 2023. “Employing long short-term memory and Facebook prophet model in air temperature forecasting.” *Commun Stat Simul Comput*, 52 (2): 279–290. Taylor and Francis Ltd. https://doi.org/10.1080/03610918.2020.1854302.

Tso, G. K. F., and K. K. W. Yau. 2007. “Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks.” *Energy*, 32 (9): 1761–1768. Elsevier Ltd. https://doi.org/10.1016/j.energy.2006.11.010.

Vijai, P., and P. Bagavathi Sivakumar. 2018. “Performance comparison of techniques for water demand forecasting.” *Procedia Comput Sci*, 143: 258–266. Elsevier B.V. https://doi.org/10.1016/j.procs.2018.10.394.

Wang, J., Q. Huang, W. Hu, J. Li, Z. Zhang, D. Cai, X. Zhang, and N. Liu. 2019. “Ensuring profitability of retailers via Shapley Value based demand response.” *International Journal of Electrical Power & Energy Systems*, 108: 72–85. Elsevier Ltd. https://doi.org/10.1016/j.ijepes.2018.12.031.

Wang, X., G. Guo, S. Liu, Y. Wu, X. Xu, and K. Smith. 2020. “Burst Detection in District Metering Areas Using Deep Learning Method.” *J Water Resour Plan Manag*, 146 (6): 04020031. American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(ASCE)WR.1943-5452.0001223.

Weytjens, H., E. Lohmann, and M. Kleinsteuber. 2021. “Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet.” *Electronic Commerce Research*, 21 (2): 371–391. Springer New York LLC. https://doi.org/10.1007/s10660-019-09362-7.

Wong, J. S., Q. Zhang, and Y. D. Chen. 2010. “Statistical modeling of daily urban water consumption in Hong Kong: Trend, changing patterns, and forecast.” *Water Resour Res*, 46 (3). https://doi.org/10.1029/2009WR008147.

Wu, Y., and S. Liu. 2017. “A review of data-driven approaches for burst detection in water distribution systems.” *Urban Water J*, 14 (9): 972–983. Taylor and Francis Ltd. https://doi.org/10.1080/1573062X.2017.1279191.

Xenochristou, M., and Z. Kapelan. 2020. “An ensemble stacked model with bias correction for improved water demand forecasting.” *Urban Water J*, 17 (3): 212–223. Taylor and Francis Ltd. https://doi.org/10.1080/1573062X.2020.1758164.

xgboost developers. 2022. “XGBoost Documentation.” *dmlc XGBoost*. Accessed October 30, 2022. https://xgboost.readthedocs.io/en/latest/.

Xie, C., H. Wen, W. Yang, J. Cai, P. Zhang, R. Wu, M. Li, and S. Huang. 2021. “Trend analysis and forecast of daily reported incidence of hand, foot and mouth disease in Hubei, China by Prophet model.” *Sci Rep*, 11 (1): 1445. Nature Research. https://doi.org/10.1038/s41598-021-81100-2.

Xu, X., and Z. Meng. 2020. “A hybrid transfer learning model for short-term electric load forecasting.” *Electrical Engineering*, 102 (3): 1371–1381. Springer. https://doi.org/10.1007/s00202-020-00930-x.

Ye, R., and Q. Dai. 2018. “A novel transfer learning framework for time series forecasting.” *Knowl Based Syst*, 156: 74–99. Elsevier B.V. https://doi.org/10.1016/j.knosys.2018.05.021.

Ye, R., and Q. Dai. 2021. “Implementing transfer learning across different datasets for time series forecasting.” *Pattern Recognit*, 109: 107617. Elsevier Ltd. https://doi.org/10.1016/j.patcog.2020.107617.

Zdravković, M., I. Ćirić, and M. Ignjatović. 2022. “Explainable heat demand forecasting for the novel control strategies of district heating systems.” *Annu Rev Control*, 53: 405–413. Elsevier Ltd. https://doi.org/10.1016/j.arcontrol.2022.03.009.

Zhang, K., P. Xu, and J. Zhang. 2020. “Explainable AI in Deep Reinforcement Learning Models: A SHAP Method Applied in Power System Emergency Control.” *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, 711–716. IEEE.

Zhang, Y., and G. Luo. 2015. “Short term power load prediction with knowledge transfer.” *Inf Syst*, 53: 161–169. Elsevier Ltd. https://doi.org/10.1016/j.is.2015.01.005.

Zhao, Z., R. Anand, and M. Wang. 2019. “Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform.”

Zheng, Y.-J., S.-L. Yu, Q. Song, Y.-J. Huang, W.-G. Sheng, and S.-Y. Chen. 2022. “Co-Evolutionary Fuzzy Deep Transfer Learning for Disaster Relief Demand Forecasting.” *IEEE Trans Emerg Top Comput*, 10 (3): 1361–1373. IEEE Computer Society. https://doi.org/10.1109/TETC.2021.3085337.