Hybrid Neural Network Techniques for Short Term Water Demand Forecasting

Oscar Wooding

Department of Civil and Environmental Engineering
Imperial College London

Abstract

This study investigates the prediction of hourly water consumption within water distribution networks over a period of 24 hours using past demand data. Short-term demand predictions empower water managers to make informed decisions regarding water supply to meet residential and industrial needs. Furthermore, predictions enable utilities to identify potential network failures such as water leaks and pipe bursts quickly. The forefront of short-term water demand forecasting consists of increasingly vanguard machine learning techniques, notably neural networks in combination with novel preprocessing methods. Unlike traditional statistical models (such as SARIMA), neural networks operate with fewer assumptions and excel at modelling highly complex, non-linear relationships between explanatory variables and past water consumption data. There is a lack of rigorous comparison across pre-processing techniques applied to modern neural networks, especially for different types of water networks. This research first proposes a novel data imputation technique, proven to positively impact later forecasting, using seasonal decomposition and missForest in combination with anomaly removal using Isolation Forests. A collection of pre-processing techniques from recent literature are then compared; first on simple multilayer perceptron (MLP) networks. An MLP-SARIMA hybrid model and novel diurnal flow residual MLP have the most promising results, marginally outperforming forecasts from SARIMA. Both techniques are then applied to a 2 layer GRU network and 1D CNN. 1D CNNs produce the best performance and are tested alongside SARIMA on new data for various training window sizes and exogenous variables. The results show that a 1D CNN returns the lowest MSE of all models across residential, industrial, and hospital districts and achieves peak performance when combined with fragmenting and either SARIMA residuals or diurnal demand.

1. Introduction and Literature Review

Water demand forecasting has become indispensable for planning new developments, assessing system expansion, determining reservoir sizes, optimizing pumping stations, and managing pipe capacities. It also aids in addressing urban water management concerns such as pricing strategies and usage limitations. In particular, short-term demand predictions empower water managers to make informed decisions regarding water supply, residential and industrial needs, and enables utilities to identify potential network failures such as water leaks and pipe bursts quickly. Water demand forecasting periods are usually divided into three categories; short-term (approximately 15min to daily resolution), medium-term (daily to monthly resolution), and long-term (monthly to annual resolution). The objective of this study is to develop short-term water demand forecasts of 1 hour resolution over the next day by evaluating models from the literature in combination with modern artificial neural network (ANN) architectures and novel preprocessing techniques.

Methods for water demand forecasting are typically split into linear and non-linear techniques [1]. Linear methods analyse univariate (single variable) time series using techniques such as exponential smoothing, auto regressive integrated moving average (ARIMA), and multivariate time series using linear regression (MLR) [2]. Although linear methods have recently been surpassed, they often achieve a high level of accuracy and are still used as a benchmark in the literature and in this study. In particular, SARIMAX models appear as the superior linear method and therefore feature heavily in recent publications [3] [4] [5] [6] [1] [7] [8] [9]. However, classic statistical models assume a normal distribution of input data which leads to limitations when handling complex and non-linear time-series (which hourly water demand data is, as evidenced by 4).

Herrera et al. [10] concludes an evaluation of water demand forecasting techniques by stating that the most common and successful models are those based on variations of ANNs and regression methods including support vector and random forest regression methods. Donkor et al [11] arrives at a similar conclusion for papers reviewed between 2010-2015. This review highlights the increased performance of neural network models over time-series regression or univariate time series models.

1.1. Neural Network Forecasting

The persistent challenges of dealing with non-linear data, and the inherent stochastic nature of demand remain significant hurdles in this field. Niknam et al., 2022 [12], discusses how artificial neural networks (ANN) gained traction in the early 2000s, notably using the back-propagation algorithm for various civil engineering applications. Unlike traditional statistical models, ANNs operate with fewer assumptions. ANNs often excel at generalizing insights, enabling predictions for new, unseen data. They also possess the ability to effectively model highly complex, non- linear relationships within data and accurately estimate non-linear functions. Further details on ANNs can be found in the appendix Appendix C. Neural networks tend to be favoured for short-term predictions, whereas econometric models and simulations are commonly employed for long-term forecasting purposes. The attributes of alternate forecasting methodologies compared to AI-based ANN methods can

be seen in Table 1 below from [11].

Table 1: Comparison of techniques commonly used in literature.

Method	Data Needs	Accuracy	Interpret-ability	Efficiency
ANN-like	High	High	Low	Low
SVR-like	High	High	Medium	Low
ARIMAX	Low	Low	High	High
Metaheuristics	High	Medium	Low	Low
Regression	Low	Medium	High	High
Hybrid	High	High	Medium	Medium

Finding the best architecture for an ANN is challenging with many variants used for different objectives. Additionally, ANNs potentially lack explain-ability and therefore generalisation, and often require large amounts of data for training and validation. Across the literature, neural networks face a few primary shortfalls including high time and space complexity, accumulative error when using iterative multi-step forecasting, and extreme point prediction. Enhancing the accuracy of neural networks is a focus point in modern research, the most promising findings are summarised below.

Deep Learning

Although conventional ANNs repeatedly produce improved results over traditional statistical methods, deep learning methods can give even more accurate results when predicting water demand for short horizons (Guo et al., 2018 [13]). Deep learning is a subset of machine learning which extracts features through many hidden layers - enabling the capture of highly complex patterns. Types of deep neural networks that can learn from labelled data (supervised) include Deep Forward Neural Networks (DFNN), simple Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), and Gated Recurrence Units (GRU) Appendix C. Simple ANNs have just 1 or 2 hidden layers while deep networks may contain considerably more. It is observed that shallow networks are incapable of learning patterns from more complex input structures. Although deep networks are more computationally demanding, several techniques are adopted which can speed up computation.

GRU

A 2021 study from Salloom et al. [4] uses the newer Gated Recurrent Unit (developed in 2014) to yield more accurate and less computationally demanding results than either SVM or ANNs. GRUs behave similarly to LSTMs, however they utilise fewer gates and subsequently less parameters are tuned, leading to a lower complexity than LSTM. The success of GRU networks is also seen in the following studies: [1] [13].

CNN

1D convolutional neural networks are the latest NN variant to show success in recent water demand forecasting literature. While prevalent in image and video analysis historically, CNNs have only recently been applied to water demand forecasting by utilising the new 1D CNN design proposed by Kiranyaz et al. [14]. Several recent studies show the strength of CNNs to maintain the high levels of accuracy observed in GRU and LSTM methods, while drastically reducing the computational time [15] [9] [1].

Hybrid Models

In 2011, Herrera et al. [16] found that the combination of SARIMA and ANN methodologies was able to improve accuracy over each individual model on hourly water demand data. By modelling the linear part of the time series using SARIMA, explain-ability was maintained, while a simple ANN was used to describe the erratic nature of residuals – playing on the strength of neural networks to predict highly complex, non-linear patterns. A similar conclusion was also found by Kofinas et al. [17] in 2014 however this study was based on monthly demand. A gap exists in combining modern neural networks such as 1D CNNs and GRUs with SARIMA to see if results can be improved from those observed in the literature. It is also noted that although hybrid models are not widely applied, they typically perform better than any single part. Therefore, this study will focus on developing individual SARIMA and ANNs as a baseline before combining models with novel pre-processing techniques and evaluating the hybridisation of SARIMA and ANNs.

1.2. Literature Review

Topics of difference between available studies include, water demand data sources and availability, data pre-processing techniques, predictive and training methods and models used, goodness of fit metrics, and conclusions developed. A comparison of the models implemented will help guide exploration by mimicking the results obtained in promising studies over multiple DMAs. Subsequently, combining the best features of each study to develop a high accuracy model for day-ahead forecasting. A selection of key forecasting techniques which appear across the literature are grouped in Table 2 from [11].

Another important difference between methods applied is the exogenous factors fed into models. The following exogenous factors are often seen in past studies:

- *Climactic or weather conditions*: Temperature, Humidity, Precipitation, Dew point. Atmospheric pressure
- *Economic / Demographic*
- *Calendar features*: Weekend, Holiday, Event Days

Results between reports also differ based on the fitting techniques used and fit criteria. Across all time series forecasting domains, the main measures of fit are not indifferent to those used for water demand forecasting seen in Table 3 below from [10].

Table 2: Frequency of techniques used in water demand literature between 2015-2022.

Technique	Frequency of use	Sources
Artificial Neural Networks	18.2%	[18], [19], [4], [3], [13],[1], [8], [9],
Traditional Time Series Analysis	20.2%	[11], [20], [21], [10], [22], [23], [24]
Regression	8.9%	
Support Vector Machines	9.4%	[10], [22], [3]
Hybrid Models	8.9%	[16],
Metaheuristic Algorithms	6.9%	[3]
Machine Learning	2.0%	[10], [25]
System Dynamics	1.5%	[3]
Other	19.7%	

Table 3: Frequency of error measures used in water demand literature between 2015-2022.

Technique	Frequency of use
Mean Squared Error	23.1%
Mean Absolute Percentage Error	16.6%
R^2 Error	13.0%
Mean Absolute Error	15.4%
NSE	13.0%
Other	25.4%

Based on the above conclusion, initial exploration should start by understanding basic time series analysis models such as SARIMA, shown to be highly accurate until recently. Then, more complex neural network methodologies will be implemented in line with a selection of key studies. The main contributors to neural network model performance are detailed below.

Parameter Optimisation

Optimisation methods are varied in the literature and commonly include, Grid Search [26], Genetic Algorithms [27], and Particle Swarm Optimisation [28]. Bayesian parameter optimisation based on Gaussian processes was selected due to similar performance to grid search, but typically requiring fewer trials by leveraging prior knowledge and feedback to guide the search.

Data Preparation

Raw data is almost always unsuitable for direct use in machine learning algorithms and often requires both missing data imputation and outlier correction. Studies often neglect to mention their imputation methodology, or use simple techniques such as filling data gaps with the mean demand on a specific hour and day. Accurate imputation is required such that missing data can be filled with sensible estimates mimicking the structure of

real data. For datasets with missing data ranging from 5 - 20%, several studies find that both missForest and K Nearest Neighbours (KNN) imputation algorithms are suitable for multivariate water demand data and require relatively low computation costs [29]. A comparison between various data reconstruction methods can be seen in work by [30], their results indicate the highest degree of accuracy when using seasonal decomposition methods, missForest and KNN. Especially for multivariate data, KNN and missForest produced very similar results, with KNN outperforming for data missing not at random and missForest for data missing completely at random. These results are echoed in similar studies where KNN is found to outperform methods including mean, median, predictive mean matching, Bayesian linear regression, and random methods [31]. Additionally, water demand data typically contains anomalous points which can hinder model training if data points are not realistic. Outlier mitigation is not used widely in the literature, but when utilised superior results are obtained [18]. [32] finds that the combination of imputation and Isolation Forest outlier correction can improve GRU model accuracy by upto 30%.

Pre-processing

A strong set of explanatory features is critical in improving neural network accuracy. This is reflected in several studies which utilise prior classification and decomposition techniques alongside neural networks to improve forecast accuracy. [4] develops a GRU deep learning model using the k-means unsupervised classification method as a prior step. Prior classification is beneficial as it both creates new features to reduce accumulative error and creates a relationship between different days – improving the space complexity of models. The inaccuracy due to large non-linearity at extreme points is also reduced with the inclusion of virtual data between actual data as a pre-processing step and it is found that one virtual value can reduce the error at extreme points by 30% of its original value. [18] utilises wavelet transforms, [32] proposes Complete Ensemble Empirical Mode Decomposition with Adaptive Noise, and [18] implements Principal Component Analysis to transform original variables into a new dataset of uncorrelated derived variables.

Research Gaps and Report Outline

Three main gaps in the literature were identified,

- A lack of systematic comparison between different neural network data preprocessing techniques applied to the latest ANN architectures, specifically gated recurrence units (GRUs) and 1 dimensional convolutional neural network (1D CNNs).
- Few comparisons of modern forecasting models on water networks with differing characteristics, i.e. residential or industrial.
- No evidence of hybridisation of data imputation and anomaly removal methods alongside indications of the impact on forecasting results.

Figure 1 indicates the stages explored during the study, including the techniques selected for initial data preparation, traditional time-series model and neural network development, and the most effective pre-processing techniques.

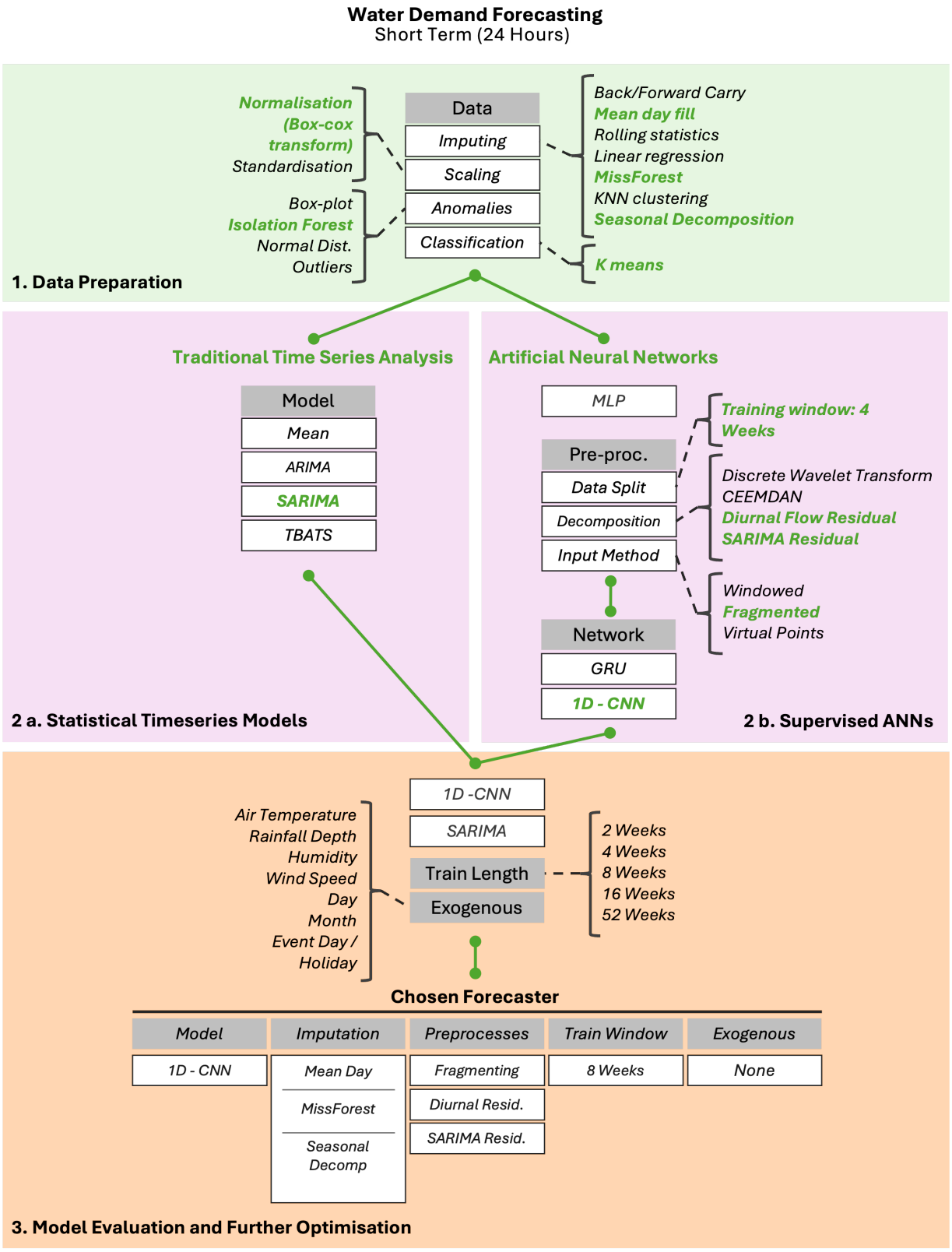

Figure 1: Progress Map with selected research avenues highlighted with green stems and most successful methods chosen in green text

This research compares the most promising data preparation and pre-processing techniques from recent water demand literature by first using MLPs and then more complex 1D CNNs and GRUs. Hourly short-term water demand data for various DMA types (e.g. residential, city, countryside, and industrial) in North-East Italy is utilised. Inspired by the successful hybrid strategies in foregoing research, newer neural networks will also be combined with SARIMA. A new diurnal demand residual method is proposed. This method leverages the repeatable intra-day nature of water demand to enable neural networks to focus only on variations from the mean demand at any given day and hour. Variations from the mean are typically non-linear as they rely on a combination of influencing factors and are not regular; playing on the strengths of ANNs with complex data. Finally, the best combination of techniques can be applied in confidence that each element has been isolated and proven to progress results. Advantages of the method developed will by bench-marked against the traditional time-series model SARIMA, using MSE and MAPE as evaluation metrics, to verify the effectiveness of the proposed neural network.

Section 2 outlines the properties of the dataset and the different techniques used for imputation and anomaly removal. Section 3 then details the development of traditional timeseries and ANN models along with the various pre-processing techniques for neural networks. The results of the fitted models are explored in Section 4 alongside an evaluation of the preprocessing techniques tested. The best three models are then selected and tweaked to find suitable values of training set size, exogenous variables used, and the impact of initial data preparation techniques. Finally, a judgement is made on the best type of model for each DMA. Section 5 presents conclusions and further research opportunities. The following key contributions are discussed,

1. Rigorous evaluation of different preprocessing techniques for neural networks across DMAs with differing characteristics, demonstrating the wide range of results that can be obtained based on DMA type and simple data manipulation methods.

2. A novel 1-Dimensional Convolutional Neural Network model capable of delivering superior forecasting ability over techniques examined from modern literature and traditional methods.

3. Imputation and anomaly removal techniques surpassing other methods studied and significantly improving model forecasting ability.

2. Data Preparation

2.1. Data Overview

2.1.1. Past Consumption

The data used in this analysis covers a case-study water distribution network located in the North-East of Italy. 10 District Metered Areas (DMAs) are considered covering a period between January 2021 and March 2023. DMAs represent a subset of pipe networks connected to the main water distribution system by a single inlet featuring a pressure

control valve. Generally, multiple DMAs connect to one central reservoir and rely on specific storages and pumping to meet customer needs each day. At the entry valve to each DMA, sensors operate to monitor and record the water consumed (demand) at varying resolutions, e.g. hourly.

The DMAs considered vary considerably in characteristics, size, and statistical properties. 13,679 Hourly data points are provided per DMA with the aim to optimise system operation for the upcoming day. Water demand for each DMA represents the net inflow and therefore includes all water consumption and leakage types of the DMA. Furthermore, hourly exogenous data is available concerning air temperature, rainfall depth, air humidity, and wind speed. A summary of each DMAs demand traits and average flows during specific days can be seen in Table ?? and Figure 2 below. Forecasting models and data preparation methods often rely on the underlying timeseries following set statistical distributions, Figure 4 shows the demand distribution for each DMA. It is clear that particularly in DMA D, E, G, H, and J, the distribution is far from normal.

Table 4: DMA descriptive statistics.

DMA	Description	Mean [L/s]	Quartile lower	Std
A	Hospital District	8.40	6.39	0.65
B	Residential district in countryside	9.59	8.32	1.87
C	Residential district in countryside	4.50	3.52	1.50
D	Residential/commercial district	33.58	27.90	6.84
E	Residential district near city centre	77.56	64.35	14.97
F	Suburban district including offices	8.05	6.57	1.99
G	Residential district close to the city centre	24.30	20.22	5.75
H	City centre district	20.27	15.13	5.75
I	Commercial/industrial district near port	19.62	17.22	3.15
J	Commercial/industrial district near port	26.31	22.61	4.46

DMA	Quartile upper	Skewness	Users supplied	Min. demand	Max. demand
A	10.27	2.80	162	1.83	29.75
B	10.56	1.08	531	6.27	21.02
C	5.23	0.92	607	1.77	11.68
D	38.6	-0.22	2094	12.66	55.96
E	88.04	-0.31	7955	48.68	113.64
F	9.30	0.29	1135	2.62	19.11
G	28.19	-0.20	3180	8.25	42.10
H	24.77	-0.09	2901	9.36	83.13
I	21.70	0.84	425	13.48	41.75
J	29.88	0.27	776	13.74	43.97

The amount of past demand data utilised for short term water demand forecasting

varies between studies, for example [33] utilised 168 hours of past demand data, while [34] uses just 72 hours to predict the next 24 hours of water consumption. These studies use a *windowed* approach where the feature given is a chronological array of adjacent data points. However, this method of data input can be excessive as many demand values in e.g. the past week have very little impact on 1 step predictions, evidenced by Appendix F.21. Instead, only the most important data can be considered with a technique called fragmenting.

2.1.2. Other Features

Features typically included in water demand forecasting studies can be classified into calendar features, climactic features, and water consumption itself. The univariate models in this study will use only the past water demand data as an input using either windowed data (i.e. a specified number of past data points from the point of forecasting), or fragmenting in neural networks. Multivariate models will be tested with a selection from time and weather features.

Calendar information can benefit forecasts by adding explanatory variables for certain demand behaviours, for example weekends may have peak demand later in the day. Time features tested include, day of week, weekend indicator, holiday indicator, and hour in day. As a single-step forecasting technique is used, future time indicators do not provide much value as e.g. if the most recent data point is a weekend or holiday, then it is highly likely that the following point also will be. Therefore, future time features are not included.

Day of the week is represented by 7 equidistant values between 0 and 1, and the hour as 24 values between 0 and 1 (i.e. 1 to 24 normalised). A boolean variable is used as the indicator for holidays and weekends, 0 for a normal day and 1 for a holiday or weekend. The two features above are inputted as arrays with one entry corresponding to each past demand value provided.

Air temperature, rainfall depth, air humidity, and wind speed are all provided as exogenous variables with the dataset. It is unrealistic to assume that a perfect forecast can be obtained for weather over the upcoming day and so these exogenous variables are used with caution as results may not be realistic. Air temperature has the highest correlation with demand data across all DMAs, with an average correlation of 0.21, particularly in DMAs A, B, C, and G. The next most impactful factor is air humidity with an average correlation of -0.10. Therefore, it is expected that air temperature and humidity will have the largest impact on predictive ability.

2.2. Data Transformation

2.2.1. Normalization

Data is scaled prior to model training to prevent large-scale data from biasing results. Two normalisation methods are commonly adopted, Min-Max normalisation and standardisation. Min-Max normalisation rescales data points to be within the range 0 and 1, and normalisation rescales data based on a normal distribution using the following formula,

$$Z_i = \frac{X_i - \mu}{\sigma_i}$$

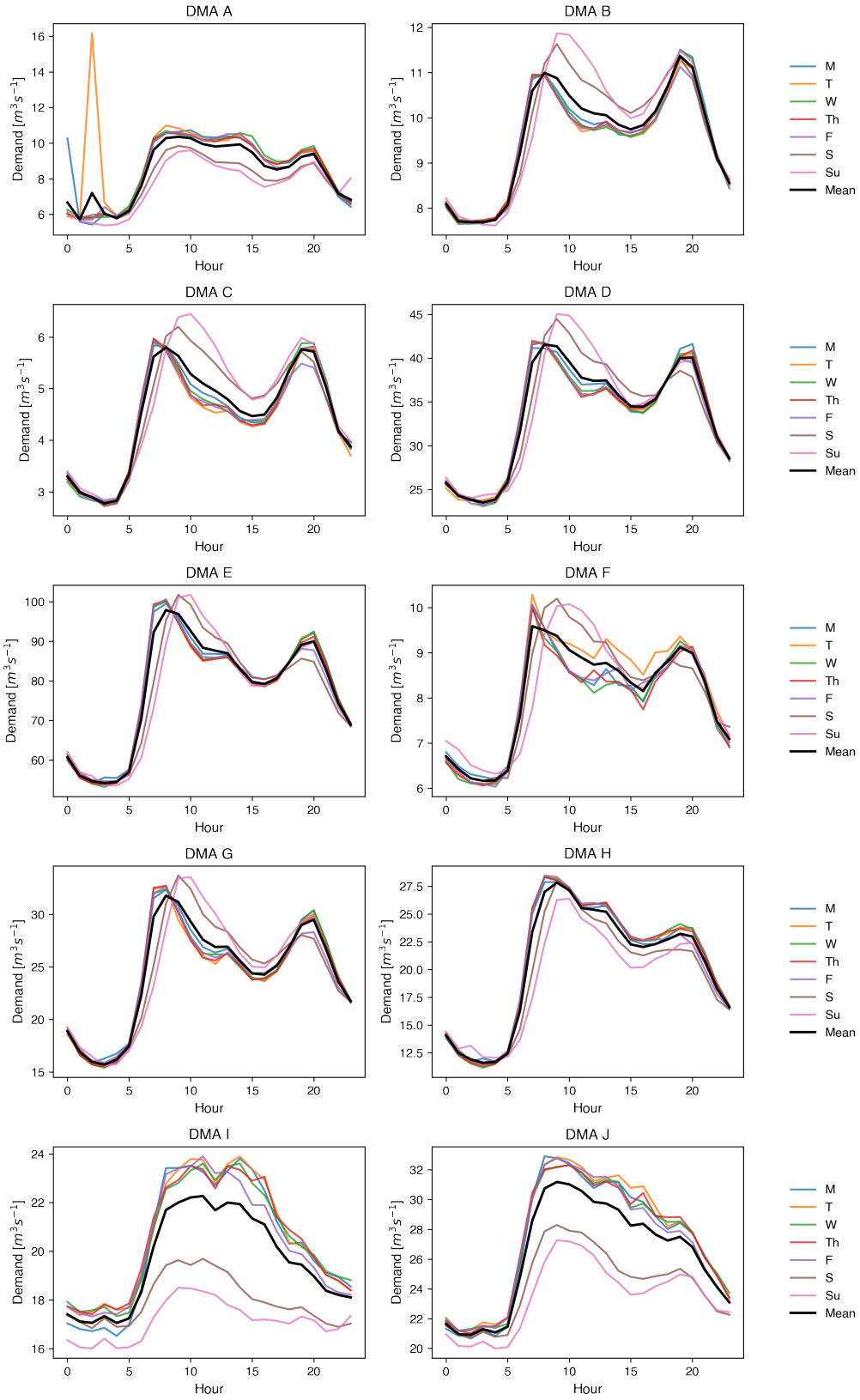

Figure 2: Average daily demand fluctuations for different days of the week.

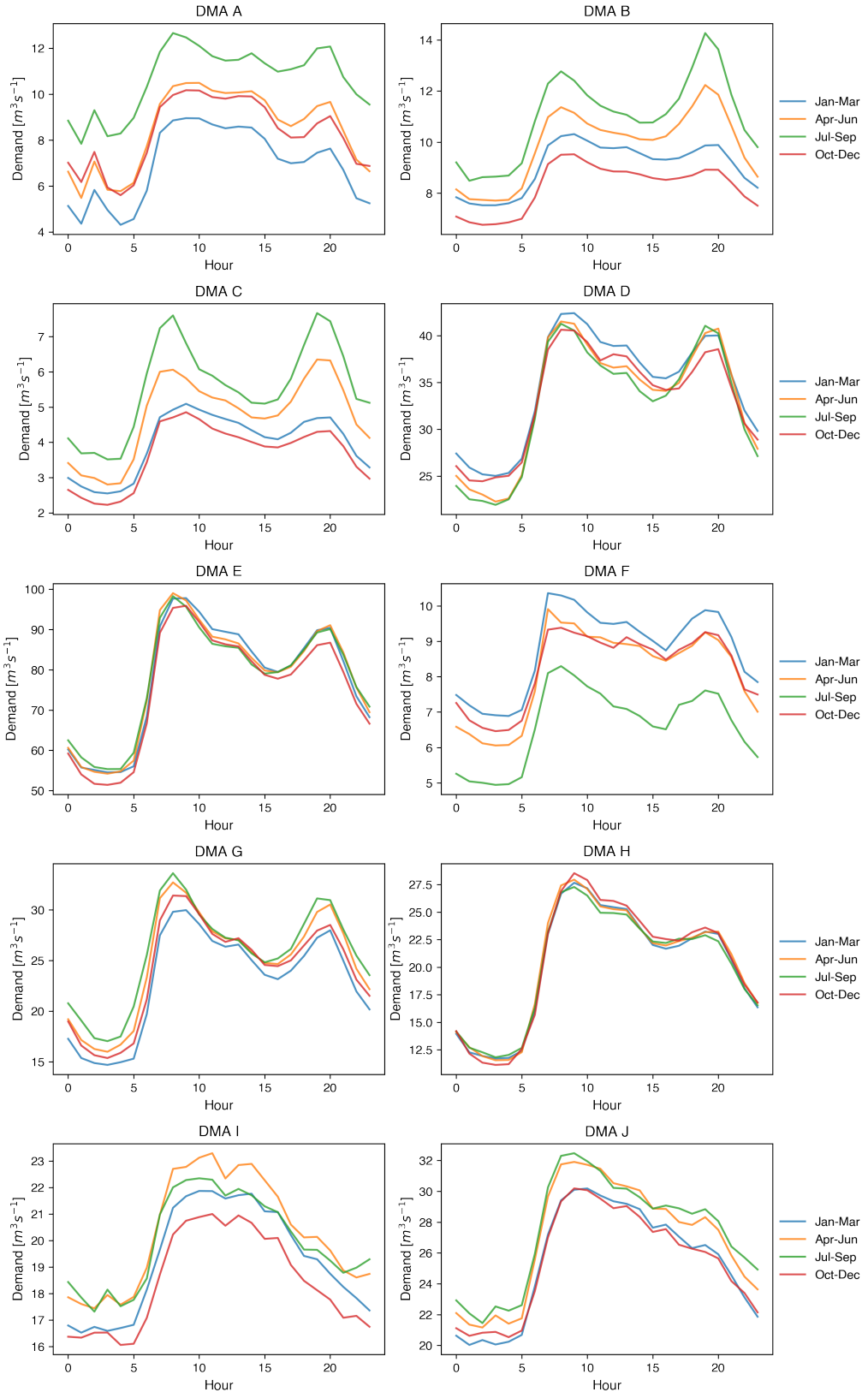

Figure 3: Average daily demand fluctuations for different periods of the year.

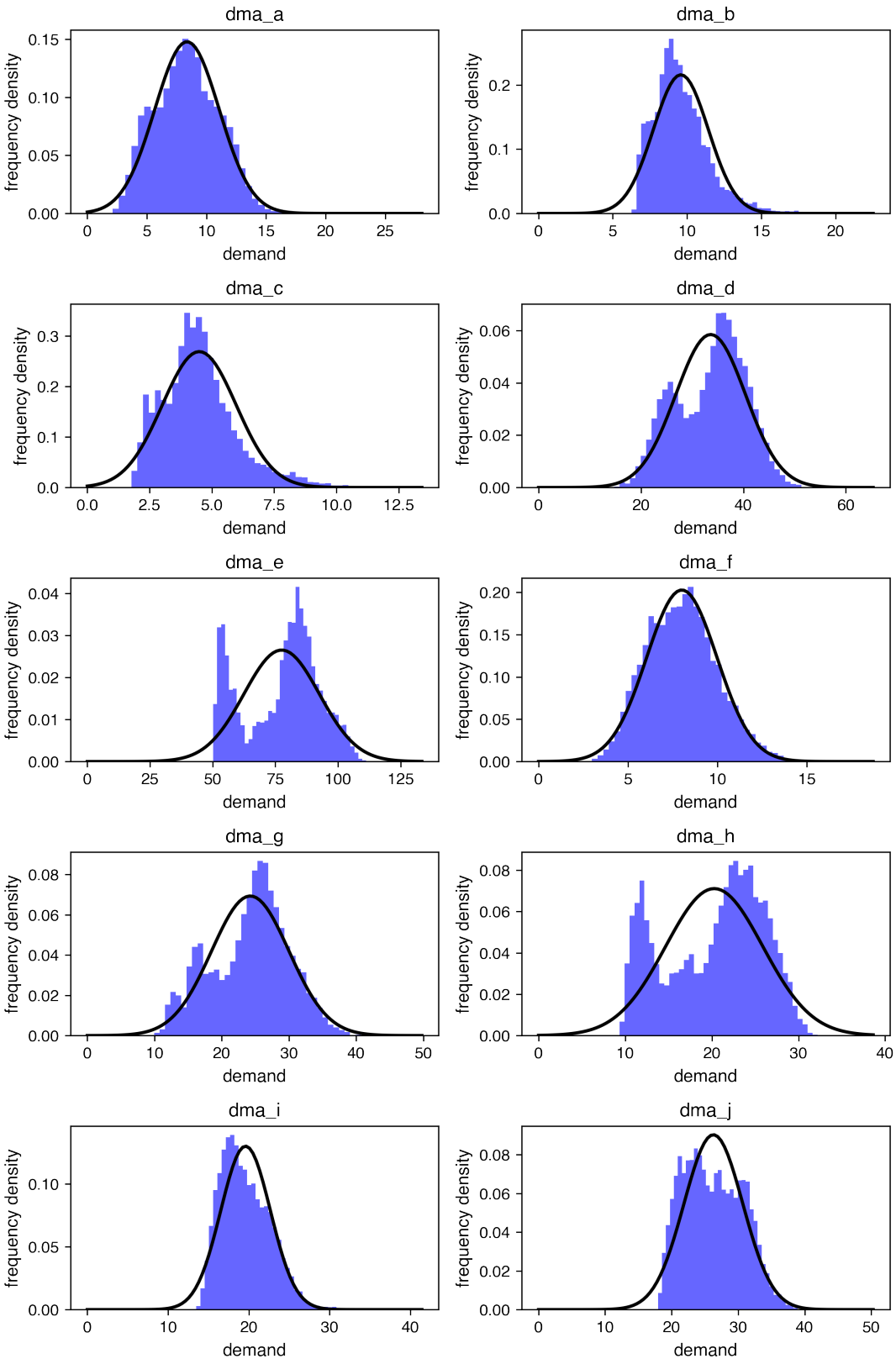

Figure 4: Histogram showing demand value distributions in all 10 DMAs with a fitted normal distribution shown with a black line.[13]

Where Z_i is the standardised value, X_i is the intial demand value, μ is the mean, and σ the standard deviation. Standardisation is used as it employs the standard deviation for scaling which mitigates the impact of outliers.

2.2.2. Data Imputing

Demand data given is not post-processed so there are gaps related to SCADA systems malfunctioning and other data collection or transmission issues. For model training, missing data gaps must be filled and so a selection of imputation methods are considered. The imputation methods observed in current research are either univariate (single variable) or multivariate. A selection of imputation methods are tested including; deletion, interpolation, carrying data forward or backwards, mean day fill, rolling statistics, linear regression, missTree, KNN clustering, and a hybrid approach. A description of each methodology can be found in Appendix Appendix B.

The data given contains multiple DMAs, therefore KNN and missForest are considered using data from correlated DMAs where available to fill missing data gaps rather than using solely weather or recent demand data. Additionally, a novel hybrid seasonal imputation approach is considered; applying the most appropriate imputing technique where possible and falling back on the next best when required if e.g. all correlated DMAs are also missing data and univariate imputation is required.

The hybrid seasonal imputation method is a hybrid between missForest, seasonal decomposition, and a technique from the literature using mean demand and a residual correction component. Where there is no data for correlated DMAs, a mean value with residual correction is employed as shown in Appendix E.1. Where data from correlated DMAs is available, missForest imputation is then used. The result of this initial imputation is then divided into a seasonal and residual component (using statsmodels *seasonal_decompose* function). The sum of these two components forms the total water demand. The seasonal component of the missForest imputation result is then subtracted from the original demand data (with missing gaps) to return a residual. This residual contains missing data gaps and is consequently imputed using linear interpolation. The imputed residuals and the missForest seasonal component are then summed to give a final imputed dataset, a summary of this method can be seen in 5 below.

Datasets from DMA H, E and I are used for testing as they include the longest uninterrupted stretches of complete data and enable tests between E and H, with many highly correlated partners, and I with low correlations to other DMAs. Gaps will be introduced artificially as a mapping of missing data gaps from other DMAs onto a sample of DMA E, H, and I as seen in 6. This allows the best imputation method with respect to each DMA's actual missing data patterns to be sought. The accuracy of each imputation method is then evaluated using MAPE.

For each of the complete data samples taken from DMA E, H, and I, 60 different missing data patterns are applied from incomplete stretches in all other DMAs. Subsequently, each imputation method is tested on 180 test cases spanning missing data patterns from all DMAs and underlying data characteristics from DMA E, H, and I. The results for each imputation method are detailed in Table 5.

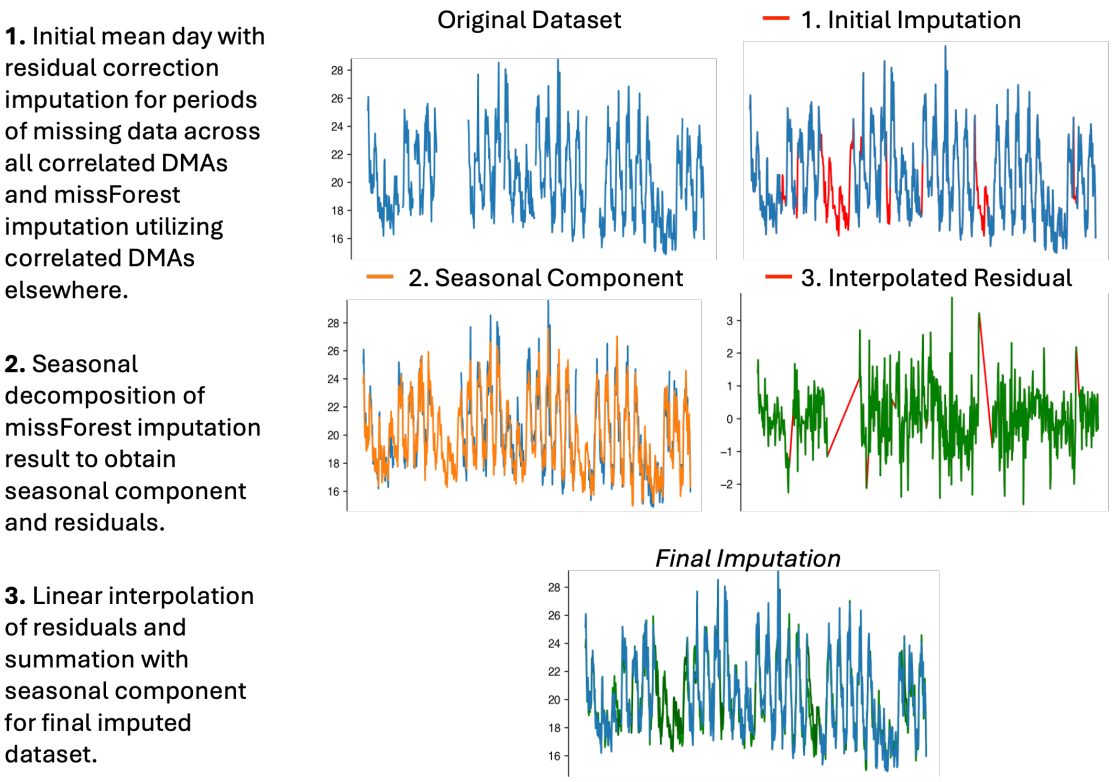

Figure 5: Explanation of final imputation method selected which combined, mean day with residual correction, missForest imputation, and seasonal decomposition.

Table 5: Mean Absolute Percentage Error (%) for different imputation methods applied using the missing data patterns from all 10 DMAs artificially applied to a test set of DMA E, H, and I.

DMA	knn	spline	rolling	nocf	mean	median	hybrid	seasonal	missForest
E	2.1	51.7	7.2	21.8	2.2	2.0	2.1	**1.6**	1.8
H	4.8	87.3	8.9	28.7	3.5	3.5	4.3	**3.2**	4.0
I	5.9	21.8	6.7	11.0	6.5	6.6	5.3	5.1	**4.6**

Across both DMA E and H, the seasonal imputation algorithm produces the most accurate results and has a high accuracy within 0.5% of the best method in DMA I. Although this method generates results superior to just a missForest imputation, further improvements could be made by also imputing the seasonal residual using missForest rather than linear interpolation. As the results are already highly accurate, and the missForest algorithm is computationally demanding, there is little need to continue improving the method. The combination of missForest and seasonal decomposition surpasses the accuracy of methods found in literature and will therefore be utilised for the remainder of analysis.

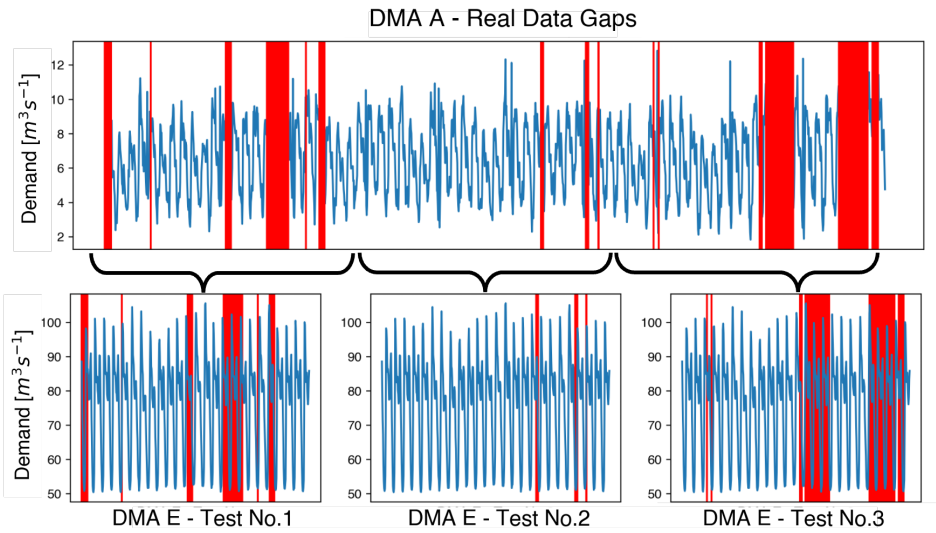

Figure 6: Example of missing data test set generation with real data gaps observed in DMA A artificially applied to a complete set of data in DMA E.

2.2.3. Anomaly Correction

Outliers in the data due to different kinds of measurement errors lead to inconsistencies with the rest of the dataset. These anomalies may add confusion to data decompositions [32] or model training and should be corrected to be consistent with the rest of the dataset.

Following the methodology presented in the literature [32], first the water demand data is grouped into 24 hourly clusters, i.e. all data for the 1st hour of the day in one group and all data for the 2nd hour in another, etc. Then, 24 Isolation Forest models are built, one for each group (using the sklearn library in Python), and outliers are identified. Each outlier is then removed and filled using the most effective above imputation method, a combination of missForest and seasonal decomposition. An example of outlier points and their correction can be seen in the Figure below 7.

It is observed that clear outliers are corrected appropriately with new values close to either the upper or lower bound of the corresponding hourly group.

2.3. Data Selection

2.3.1. DMA Selection

Due to the large similarities between DMAs and long computational times for more complex models, it is beneficial to utilise fewer DMA series during hyperparameter selection and model development phases. Therefore, the DMA time-series are normalised using a min-max scaler and clustered using K-means (with the tslearn Python package) into 4 clusters based on demand values alone. 4 clusters were selected based on trial and error until the timeseries' present in each cluster had visibly similar characteristics. From each cluster, the DMA with the fewest data gaps over the training period is utilised for the rest of the analysis. Figure 8 indicates the clusters selected. DMA I was chosen in cluster 0,

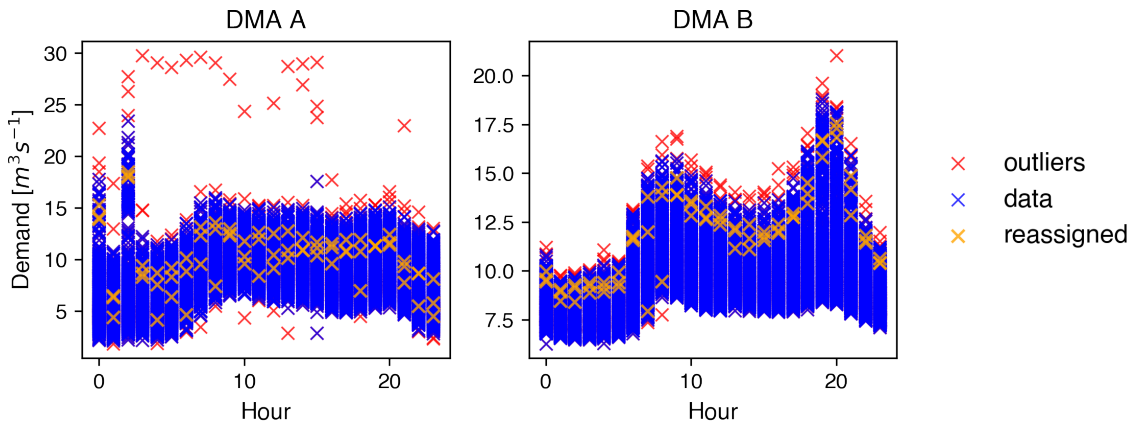

Figure 7: Example of outliers detected in DMA A and B and their corrected values in orange.

DMA G in cluster 1, DMA B in cluster 2 and DMA A for cluster 3. Given the visible similarity of demand patterns in each cluster, it is highly likely that the results of one DMA from each group will be representative of the DMAs in the rest of the cluster.

DMA D, E, and G are the only large residential DMAs of the 10 and so their grouping aligns with expectations. The characteristics of a hospital district with very few users likely has much higher variance and different demand patterns to other DMAs so it's separation from the rest is sensible. DMA B and C serve a relatively low number of users and are both situated in the countryside - a higher percentage of residential gardens could explain the notable difference in demand patterns identified. This leaves DMA F, H, I, and J clustered together; all of which have prominent industrial uses. It follows that DMA characteristics can generally be grouped based on features such as use - residential, industrial, or other, and location - city or countryside.

2.3.2. Train and Test Data

Initially, the final two weeks of data are withheld during all traditional timeseries (TTS) and ANN parameter tuning. The penultimate week is then revealed for testing during TTS and ANN model comparison. The top 3 models produced in Section 3 are then compared using the final week of data available. The final week of data is also used to assess the impact of varying training set lengths and exogenous variables. This allows validation of different model types to be done separately from an evaluation based on three selected models; confirming the strong performance of the models selected is not anomalous. During the validation stage where different preprocessing techniques are compared, 4 weeks of past data is used for all model training. During the final evaluation stage, training set length is varied to find the optimal.

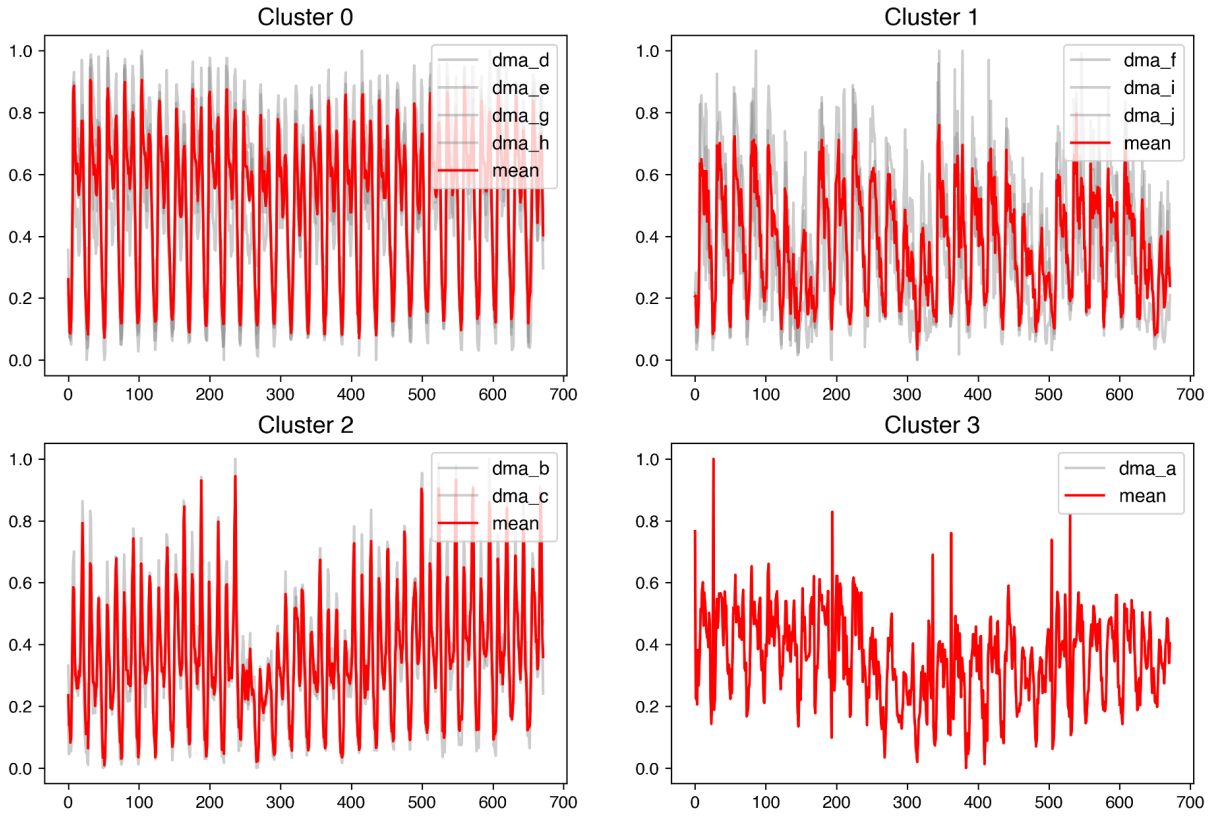

Figure 8: Clustering of 10 DMA demand datasets provided based on k-means with 4 initial clusters.

3. Statistical Time Series Models

This section gives a description of traditional timeseries models developed including SARIMA and Mean day with residual correction as a benchmark for more complex neural network performance. During the majority of model development, univariate models are used. As a final stage, the best models are then modified and tested for multivariate inputs. All models are trained and tested using a 2019 M1 Macbook Air.

3.1. Mean Day model with residual correction

First, a simple mean value model is utilised with the same methodology as seen in 2.2.3. The average demand value for each hour of each individual day in the training set provides a base estimate. The actual training data is then subtracted from this mean value to give a set of residuals. The mean of the 48 (2 days) residuals leading up to the test period is then used to correct the mean hour and day values for the test period. A correction using the average residual for each *specific hour of* ***any*** *day*, as well as a correction of the *average residual of each hour* ***and*** *specific day* was tested, but yielded worse results. 2 days for residual correction was chosen based on trial and error between 24, 48, and 168 (1 week)

past hours. It is expected that this method will be most effective in DMAs where demand has a very repeatable nature, e.g. in large residential areas with many different consumers such as DMA D, E, G, and H.

3.2. SARIMA

As the timeseries shows clear seasonality with repeating patterns recurring each day, week, and year, it is appropriate to utilise the seasonal auto regressive integrated moving average model (SARIMA). Statistical time series models typically require that data is stationary. Therefore the following conditions must be met: constant mean, constant variance, constant auto correlation structure, and no periodic component.

The auto-correlation and partial auto-correlation plots of each DMA are used in tandem with the augmented Dicky-Fuller test to assess the stationarity of each DMA. The ACF and PACF plot for each DMA can be seen in the Appendix F.21. These plots consider just the past 1 year of data as studies repeatedly show that this amount of data is sufficient for accurate SARIMA prediction [5]. The augmented DF p-value for all DMAs is above the threshold of 0.05, and additionally there is clear seasonality in all ACF plots. Therefore to make the data stationary, a single step and week differencing is applied. The ACF and PACF plot for the differenced data is shown in the Appendix F.22. Following differencing, the data sets all have a p-value below 0.05 and are visually stationary as seen in the ACF and PACF plots.

SARIMA models are represented by the following formula,

$$SARIMA(p,d,q)x(P,D,Q,s)$$

the parameters of the model are as follows:

- p and seasonal P: indicate the number of autoregressive terms (lags of the stationarised series).
- d and seasonal D: indicate differencing that must be done to stationarise series.
- q and seasonal Q: indicate the number of moving average terms (lags of the forecast error).
- s: indicates seasonal length in the data.

Two approaches are used to find and validate the SARIMA model parameters; observations of the ACF and PACF plots to narrow down a few potential parameters, and a grid search using pdmaria's *auto_arima* function. For example, considering DMA A: In the ACF plot, there are significant negative spikes at a lag of 1, and lag of 168 (weekly seasonality). In the PACF plot, there are significant negative spikes at lags of, 1, and 2. Therefore the following initial parameters are selected:

- p and P set to 0 as there are no significant positive spikes in either plot post differencing (other than at a lag of 0).

- d and D set to 1 for the first and seasonal difference mentioned earlier.
- s set to 24 to cover the daily seasonality.

It is important to note that although s is set to 24 to cover the daily seasonal component, it is also evident that there is a weekly seasonality component to consider. This would require $s = 168$ which is highly computationally demanding when training the SARIMA model. Therefore, an alternate method of de-seasonalising the data prior to training is used here. The data is de-seasonalised using the diurnal water demand. Diurnal demand is the averaged demand corresponding to each individual weekday and hour. The diurnal demand derived for each day is shown in Figure 2. Diurnal demand is first subtracted from the original data to get a residual. The residual is then modelled using SARIMA and summed with the diurnal demand to form a forecast.

Table 6: Description of different traditional time series models applied.

	Model	**Train Window**	**Description**
1.	Mean	4 Weeks	Mean day and hour with residual correction.
2.	Base SARIMA	4 Weeks	Daily seasonality with processed data.
4.	Diurnal SARIMA	4 Weeks	Modelling diurnal flow residual with daily seasonality.
5.	TBATS	4 Weeks	Daily and weekly seasonality on diurnal flow.

SARIMA models were tested with training datasets of length ranging between 1, 4, 8, 24, and 52 weeks. Model accuracy was consistently higher when using shorter training periods of four and one weeks across all DMAs, it is possible that this is due to a single seasonality of 24 being used. Without other seasonalities such as monthly or annually being considered, it is unlikely that the SARIMA model gains much useful information from larger training sets where monthly and annual variations not measured may interfere with predicting short term daily variations. Four weeks of training data were therefore selected for all initial models (including neural networks), rather than 1, as neural networks would also be trained on this volume of data and typically prefer longer training sets.

4. Supervised Neural Networks

This section gives a description of the models developed including Multilayer Perceptrons (MLPs), Gated Recurrence Units (GRUs), and 1-Dimensional Convolutional Neural Networks (1D CNNs). The technique used for forecasting demand values a day ahead, recursive multi-step forecasting, is then explained alongside fragmenting and virtual data methods used in ANN feature development. Multiple ANN pre-processing techniques

are discussed including, discrete wavelet transforms (DWT), complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), and diurnal and SARIMA residual methods. Networks are implemented using the *Keras* library in Python (Keras is the high-level API of the TensorFlow platform) and hyperparameter tuning is applied using the *scikit-optimize* package with Bayesian parameter optimisation. We first consider the development of univariate models. Multivariate models are later evaluated in Section 4.

4.1. Network Structure and Training

Three variations of neural networks are applied including an MLP, a GRU Network, and a 1D CNN. Initially, manual hyperparameter tuning was conducted concerning the number of hidden layers, activation functions, epochs, and over-fitting methods. Features of the network were varied iteratively, while following best practices from water demand literature [1], to find models producing the lowest MSE in DMA A. After deducing appropriate initial models, further hyperparameter tuning was conducted on a new validation set using Bayesian Optimisation for each DMA. Figure 9 indicates the structure of the initial models prior to Bayesian hyperparameter tuning.

All models are tested using the same equipment and datasets. Parameters are initialised using the Xavier uniform initialiser (provided by default by Keras) where weights are initialised using random numbers selected form a uniform distribution. The weights of the ANNs are modified using the iterative Mini-batch Gradient Descent and MSE loss function. This method calculates prediction error for a small batch of prediction steps and the average error is backpropagated to adjust models' weights and biases each epoch. Best weights are selected using callbacks to get the best weights across all epochs, rather than the final epoch. Adam's algorithm is used to adjust learning rate due to it's out-performance of other adaptive learning rates, evidenced by [1].

Each model tested is capable of accepting both univariate and multivariate inputs. The input of univariate models using windowed features is an array of size (168, 1), and with fragmented features (45, 1) including 45 past water demand values. In the case of multivariate models with fragmenting, the single input array is expanded to include new features in an array of shape (45, n+1, 1) where n is the number of features included. During training, multiple inputs are used as part of a *batch*, giving an input array of size (45, n+1, b) where b is the batch size.

4.2. Parameter Optimisation

Hyperparameters are variables used to define how a model operates. Traditional time-series models were optimised using ACF and PACF plots to narrow down a few potential parameters, and a grid search using pdmaria's *auto_arima* function. In the case of neural networks, *hyperparameters* include batch size, learning rate, drop-out rate, or number of neurons in each layer. A validation set is used to first select the optimal parameters of each model before testing their performance on a new test set (the second to last week of available data). Each model is given 60 iterations for the optimisation, with a 7-fold cross

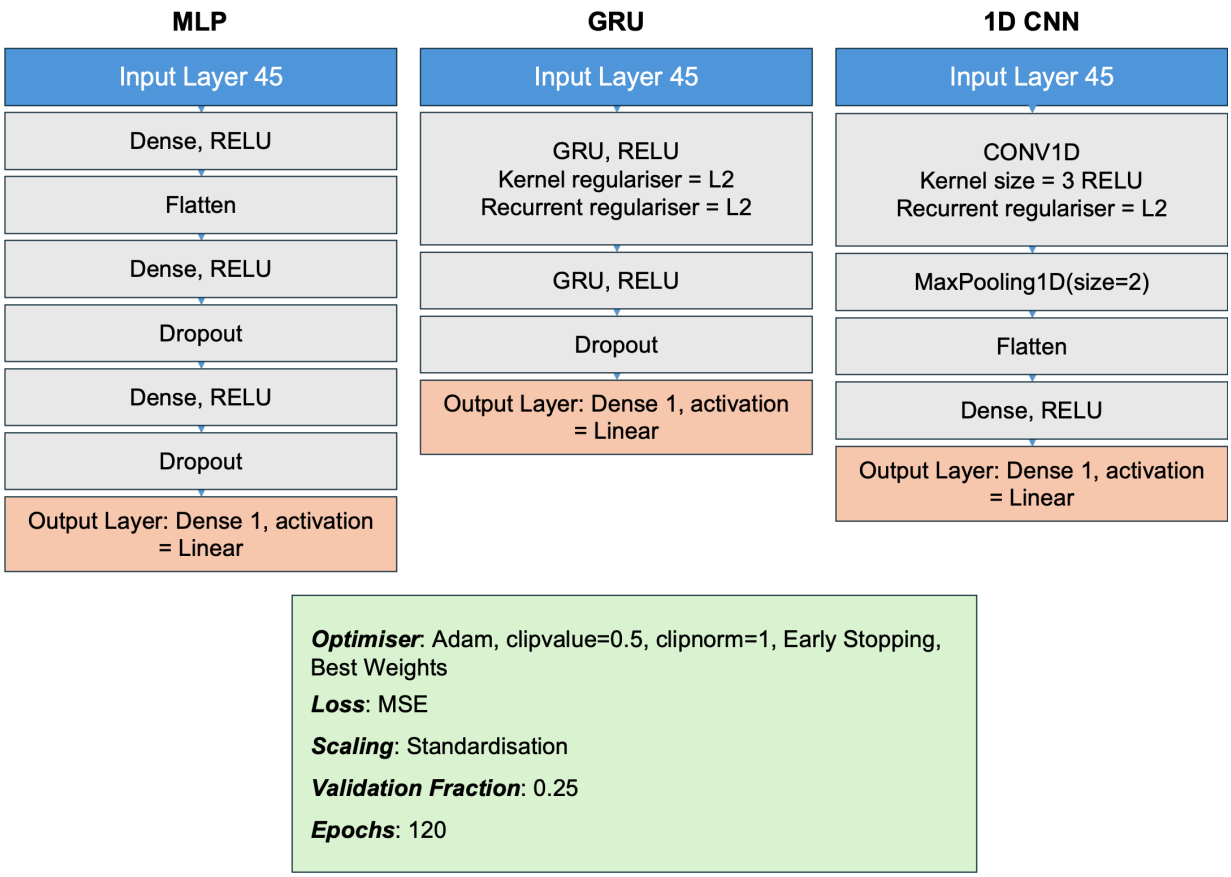

Figure 9: Architecture of deep learning models

validation conducted each iteration as detailed in 4.3. The mean squared error (MSE) averaged across each cross-validation on the test set was used to compare the combinations of hyperparameters from different iterations. The variables varied during optimisation include:

- *Number of units per hidden layer* Ranging between values of 32, 64, 96, and 160
- *Dropout Rate* Ranging between 0.0 to 0.5
- *Learning Rate* Ranging between 0.0001 and 0.01
- *Early-Stopping Patience* Ranging between 5 and 15
- *Batch Size* Ranging between 16, 24, and 48

Partial dependence plots (using the scikit-optimize package for Python) were also observed to validate results, an example of which can be seen in 10. The details of parameters selected for each different model tested are presented in D.13 for the traditional timeseries models, D.14 for the simple MLP variations, and D.15 for GRU and 1D CNNs.

60 iterations is typically considered low for an exhaustive parameter optimisation, however due to the large volume of models to be fitted, assumptions were made about certain hyperparameters to reduce the set of variables to optimise. For the models using windowed data, 168 past data points (1 week) were used, epochs were set as 120 following [13], validation fraction was set to 15%, and the loss function constrained to MSE.

Many of the preprocessing techniques applied also required specific parameter choices. This study uses the best parameters found in other reports with slight alterations made where a manual trial and error reveals clearly better configurations.

4.3. Forecasting Technique

Predicting multiple successive hourly demands is important for operation strategy concerning water production and pumping. The focus of this study is a multi-step forecast of 24 hourly water demand values, starting from the beginning of each day (i.e. 12am). Recursive multi-step forecasting is chosen, amongst the other methods often observed in the literature Appendix F.1. This method uses a single model which is trained once and then used to predict a single timestep ahead. This prediction can then be added as a model input to iteratively predict another timestep, and so on for a chosen number of points ahead, i.e.

$$prediction(t+1) = model(obs(t), obs(t-1), ..., obs(t-n))$$
$$prediction(t+2) = model(prediction(t+1), obs(t), obs(t-1), ..., obs(t-n))$$

Recursive multi-step forecasting has a relatively low computational requirement, while retaining dependencies between predictions as shown by [35]. The method is outlined in 11 alongside a depiction of fragmenting. The process is then repeated for each day of the week using a method called cross-validation 13.

4.4. Pre-processing Techniques

This study focuses on the impact of a selection of data pre-processing techniques including: fragmenting, and diurnal demand and SARIMA residual modelling. Virtual data Appendix E.5, DWT Appendix E.6 and CEEMDAN Appendix E.4 decomposition methods are discussed further in the appendix. This section will describe the different methods tested. An overview of the different models used in initial preprocessing method comparison are outlined in 7

4.4.1. Fragmenting

Other than windowed data, some recent studies also include a technique called *fragmenting* which uses only the most important passages of past demand data as input via fragments [36], [35]. Instead of a simple *windowed* array of the past demand values in order, e.g. $(D_t, D_{t-1}, D_{t-2}, ..., D_{t-n+1}, D_{t-n}$, where D_t is the demand at time t) a fragmented feature includes samples of past data at different lags from the current time step to give present, past, and distant samples. For example, present data could include the past 3 time-steps (D_t, D_{t-1}, D_{t-2}), past data could include the 3 time-steps around the

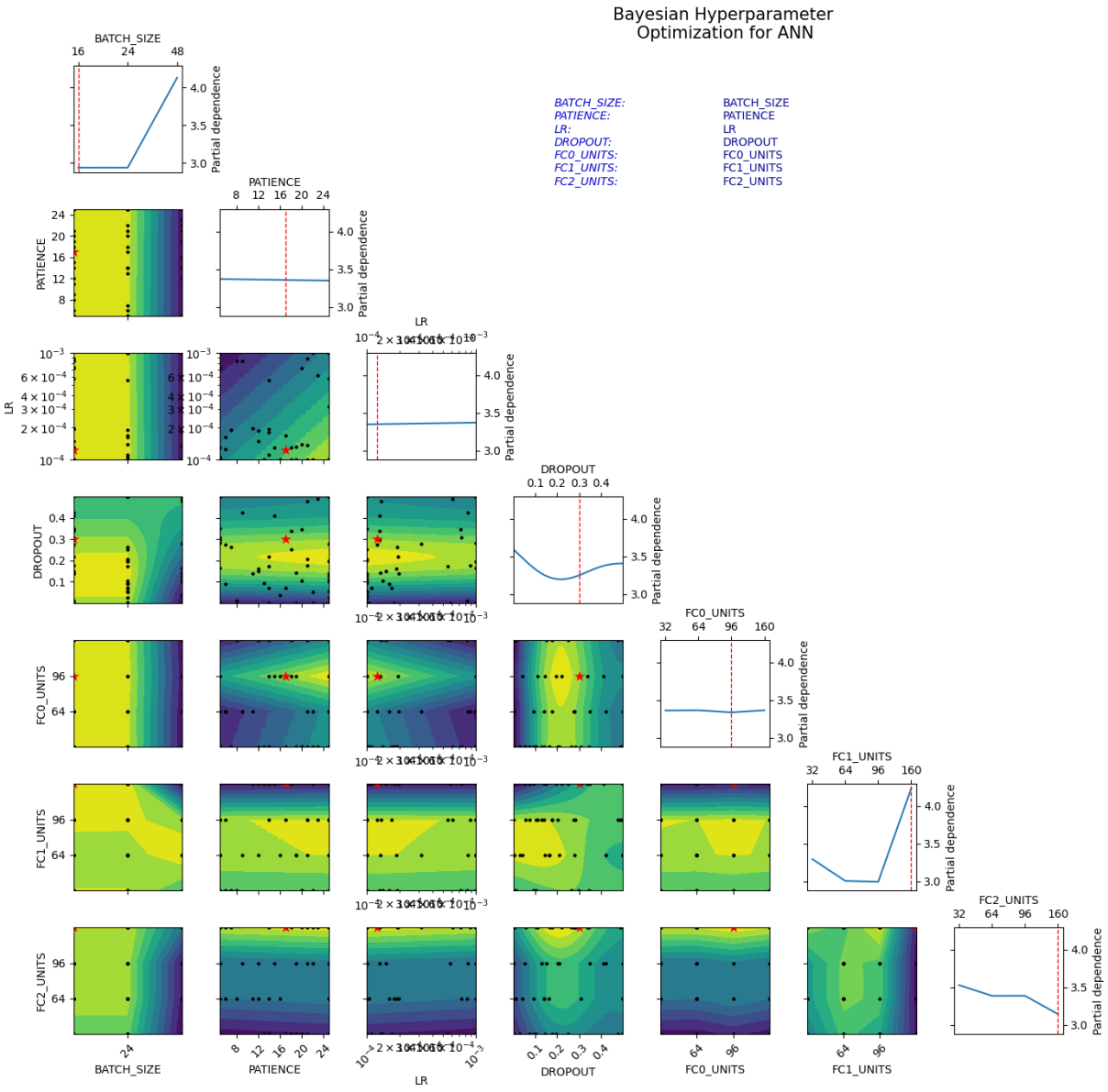

Figure 10: Partial dependence plots for average test set MSE during cross validation (objective function) for a simple ANN with two hidden layers created from the results of a bayesian parameter optimisation. Coloured plots are 2D PDPs with brighter regions corresponding to lower values for the objective. Black dots correspond to values of the hyperparameter for each iteration of the optimisation and the red star indicated the best iteration. Graphs on the diagonal are one-dimensional PDPs for each hyperparameter considered. The red dotted line indicates the best iteration.

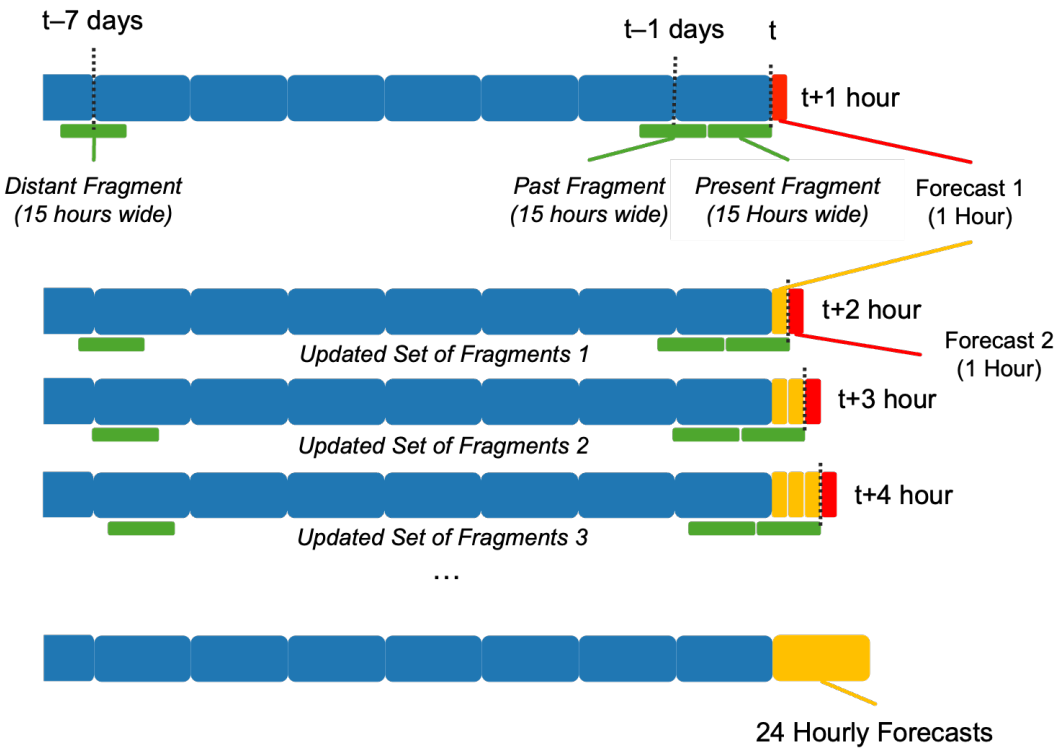

Figure 11: Example of multi-step forecasting for 24 hours ahead using recursive multi-step forecasting with fragments as input. Ranges of data used in each fragment are shown by green bars, blue bars indicate all data available in the past week, red represents a forecast of 1 step ahead, and yellow is a 1 step prediction which has been added to the training set to make another 1 step prediction.

same hour on the prior day $(D_{t+1}^{day-1}, D_t^{day-1}, D_{t-1}^{day-1})$, and the distant data could include 3 time-steps around the same hour on the same day last week $(D_{t+1}^{day-7}, D_t^{day-7}, D_{t-1}^{day-7})$. Figure 11 demonstrates this.

[35] uses a grid search procedure to select the best fragments for hourly data as 3 time steps wide at a distance of 0, 1, and 2 days for the present, past, and distant fragments. This setup was used as a baseline, however as the auto correlation plots also indicated a large dependence on demand values a week in the past (as seen in [1]), a distant fragment with a distance of a week was also tested. A further exploration by [37] uses grid search to test the fragment widths and finds that a width of 15 time-steps is optimal for 15-minute resolution when compared to [35]. During an initial trial and error, it was found that the results from [37] were reflected in DMA A, B, G, and I with a fragment width of 12-16 offering the highest accuracy with an insignificant increase in computation time. However, as this study uses single-step forecasting rather than multi-step, the distant fragment distance a 7 days offered superior results to the 4 days proposed.

Table 7: Description of different pre-processing method comparison models applied.

	Model	Methods	Description
1.	3 layer ANN	window	Sequential list of past demand values used as input.
2.	3 layer ANN	fragments	Set of past demand values condensed into fragments.
4.	3 layer ANN	virtual points, window	Virtual data points introduced between past demand values.
5.	3 layer ANN	CEEMDAN, fragments	Separate models trained for each CEEMDAN decomposition using fragmenting of past decomposition values.
5.	3 layer ANN	DWT, fragments	Separate models trained for approximate and detail component of discrete wavelet transform with fragmenting used.
5.	3 layer ANN	Diurnal, fragments	Model trained on residual of actual flow subtracting diurnal flow then summed back to diurnal flow, using fragmenting.
5.	3 layer ANN	SARIMA residual, fragments	Model trained on residual of SARIMA prediction subtracting the actual demand, using fragmenting.

4.4.2. SARIMA Residual

Firstly, the SARIMA model is deployed using cross-validation 5.1.2 to produce 7 separate 24 hour forecasts covering the next week. Residuals are obtained by subtracting the SAIRMA model fit over the *training set* from the actual training set demand values. The SARIMA model is then used to predict 24 hours ahead in the first fold, meanwhile the training data residuals from SARIMA are used to train the neural network. The neural network is then used to predict 24 hours ahead of SARIMA residuals. This is repeated for all 7 days in the week of test data. Once the residuals have been forecasted over the same week-ahead period, they will be combined with the SARIMA forecast to give a final hybrid forecast. The methodology of this is depicted in 12.

4.4.3. Diurnal Demand Residual

This method is similar to the SARIMA residual method described above, however the residual to be modelled is obtained by subtracting actual water demand from the average demand on each specific day and hour. A model is trained on this residual with the resulting forecast being added back to the diurnal demand to give a final prediction.

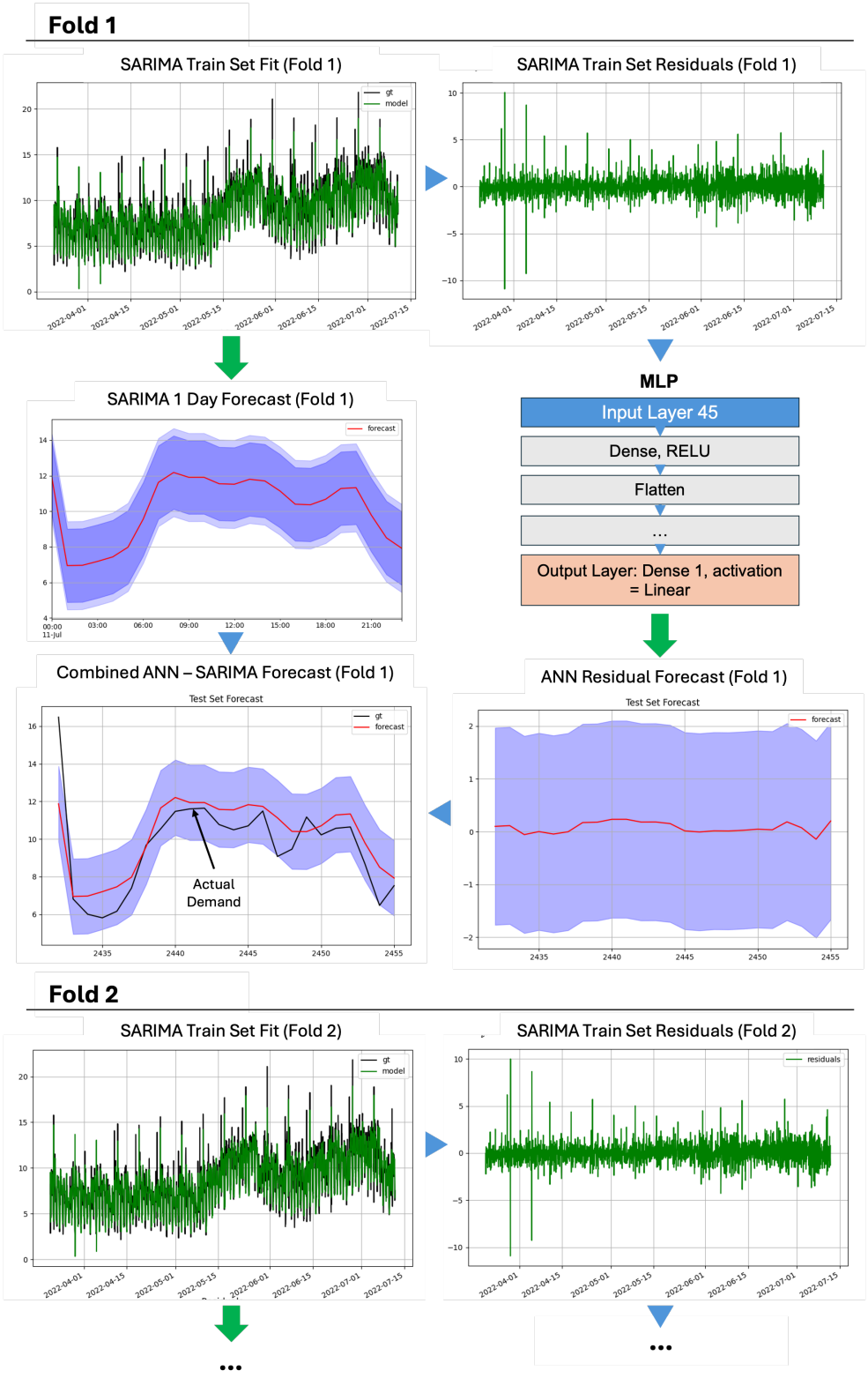

Figure 12: Methodology used for ANN - SARIMA hybrid model for each cross-validation fold.

4.4.4. Over fitting Mitigation

Complex models trained with small datasets may be prone to overfitting. While models may show good performance on a training dataset, the prediction of unseen data includes large errors as models are mimicking exact patterns in the training set, rather than the over-arching characteristics of the underlying data. Dropout layers, L2 regularisation on network weights, and early stopping are employed to reduce the impact of over-fitting. A decaying learning rate is another popular method, however results were hindered when using this technique in testing.

5. Results and Discussion

5.1. Evaluation Methods

The following subsections describe the methods used in the evaluation of both traditional timeseries models and ANNs.

5.1.1. Performance Metrics

Root mean squared error (RMSE), mean squared error (MSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) are used due to their prevalence in recent studies 3. As mean squared error (MSE) serves as the optimisation objective during hyperparameter tuning, it is considered the primary evaluation metric. The formulation of each is given below.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

n represents the number of observations or data points. y_i stands for the actual value of the dependent variable for each observation i, while $\hat{y}_i$ represents the corresponding predicted or estimated value. RMSE and the MSE provide varying measures of the spread of errors. Mean Absolute Percentage Error (MAPE) allows comparison across DMAs. Mean Absolute Error (MAE) calculates the average of the absolute differences between actual and predicted values, representing the average magnitude of errors.

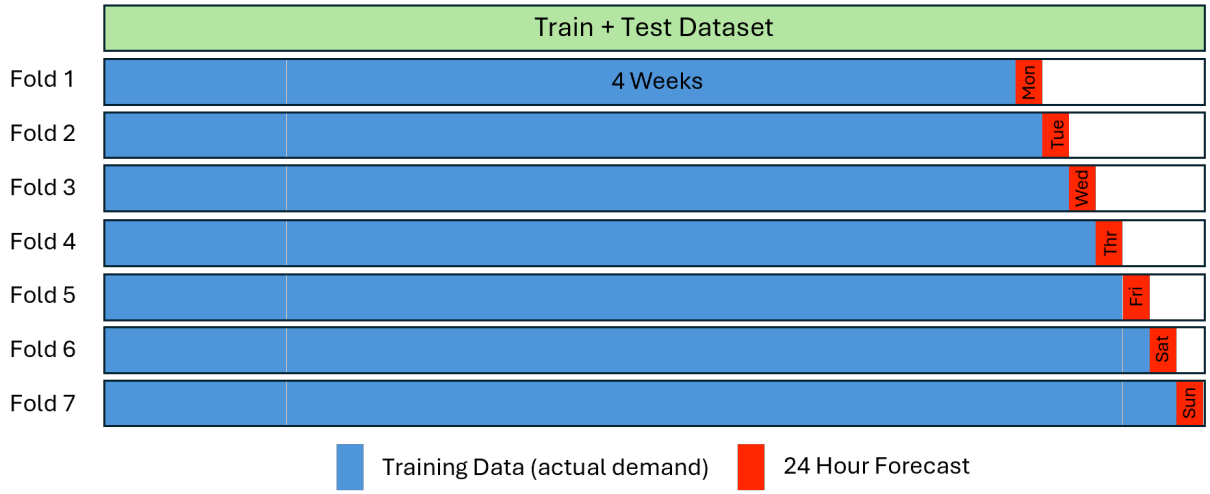

Figure 13: Depiction of cross validation technique.

5.1.2. 7-Fold Walk-Forward Cross Validation

A walk-forward validation is utilised for day-ahead forecasting covering a period of one week. During the first stage of model building, four weeks of data (672 hours) comprise the training dataset, with one week of data (168 hours) for testing. In this case, cross validation consists of first training the model with four weeks of past water demand data, split into features and labels (e.g. 144 sets of windows of 24 past values, with 144 values of the demand straight after the windows). Then, the trained model uses recursive multi-step forecasting to produce a forecast for the next day. This forecast is then evaluated and its scores saved. The next day of *real* data is then added to the initial four weeks and the model is re-trained on four weeks and one day of actual demand data. Again, the next day is forecasted and the process repeats for an entire week as shown by 13. Finally, the evaluation scores are averaged over all seven days of iterative prediction.

A test set of 7 days allows one 24 hour forecast for each day of the week, however this test set size is perhaps slightly too small as the week may be anomalous. This is recognised as a flaw in the evaluation methodology, however due to computational constraints, a single week is deemed sufficient for selecting the best preprocessing techniques before utilising a further week of test data for selected model comparison.

5.1.3. Testing Pipeline

Models are built up iteratively, slowly introducing new elements in a pursuit of the lowest model MSE and MAPE. By introducing complexity step-wise, the direct impact of different model types, features, pre-processing techniques, and network architectures can be viewed in isolation. It is therefore easy to judge how certain techniques will impact results and give a robust evaluation of different methods seen in papers, as well as selecting the elements from each paper which have the biggest positive impact. The flowchart of model development is given by Figure 1.

Traditional time-series models are first experimented with and fitted, including SARIA,

SARIMAX, TBATS, and a mean residual method to provide a baseline for forecast performance given that these methods are well established in timeseries forecasting literature. Four weeks of data for training is chosen based on SARIMA experimentation with a week of testing data. This data split is also applied to all MLP models. ANN models were first tuned using Bayesian parameter optimisation on a validation dataset using 7-fold walk-forward cross validation. Exogenous variables relating to time, or weather are not included until toward the end of model development so that the direct impact of these factors can be seen in isolation and models can be trained more quickly before introducing further variables. The focus of the study is the impact of pre-processing techniques and therefore simple ANNs are initially used alongside pre-processing techniques relating to feature selection and time-series decompositions. With past water demand data as the only variable, this input is tested as a windowed feature, fragments, and with the inclusion of virtual points. After selecting the most accurate ANN combined with these techniques, decompositions are then compared. Two successful decompositions from the literature are tested (CEEMDAN and DWT) alongside diurnal demand residual decomposition, and SARIMA residual modelling.

After selecting the most accurate data input method, and then decomposition, the final combination of techniques is applied to networks shown to have strong results in recent literature; GRUs and CNNs. After selecting the most accurate model from GRUs, CNN, and ANN networks with the selected pre-processing techniques, a new test set is introduced (the final week of data available). This test set will be used to compare the top 3 models selected from method evaluation. Additionally, the impact of exogenous time and weather variables, and train set length is evaluated. The effect of initial imputation and anomaly removal is also assessed. The best model and accompanying exogenous variables (if any) will be used with the proposed imputation and anomaly removal method, and then again with a simple mean imputation technique observed in the literature (without anomaly removal) to determine if complex imputation can improve forecasting results. At each stage, evaluation metrics were calculated for each cross validation fold and averaged as shown in 9, 10, 11, 12 for each DMA. Due to the random aspect of neural networks tested, each model was run 10 times and the metrics averaged to give a robust estimate. The full code deployed for testing can be found at [38].

5.2. Traditional Timeseries Model Results

We first discuss the results of traditional timeseries models which act as a baseline for the ANNs developed. A summary of all model results for each DMA can be found in Tables 9, 10, 11, and 12 for DMA A, B, G, and I respectively. Given the highly repeatable nature of water demand, the strong performance of the mean method is expected. Particularly in DMAs with low variability in consumption week-to-week, such as DMA G and I, the results of the mean (4.7% and 4.3% in DMA G and I respectively) are very close to the MAPE of the best performing ANN methods (4.1% and 4.2% in DMA G and I respectively). With the addition of residual correction, this method is also able to incorporate short term changes in consumption over the past couple of days. Although the reason for this change may not be clear, a shift across all demand values may occur due to generally higher

temperatures, or leaks which mean demand patterns shift above the average. However, this model is unable to capture non-linear intra-day variations from the average flow as a single correction value (the average mean residual over the past 48 hours) is used for the entire upcoming day forecast.

Basic SARIMA models without exogenous variables, or diurnal flow, struggled to beat predictions made with the Mean model, with a forecasting MAPE of approximately 15% in all DMAs tested. One reason for this was the inability to distinguish between different flow characteristics on different days such as weekends and weekdays as only daily seasonality was considered. A SARIMA model in combination with diurnal flow showed the highest accuracy across all traditional timeseries models with the lowest MSE and MAPE across DMA A, B, and G. When using diurnal demand with SARIMA, the MSE was reduced by over 80% in most of the DMAs, highlighting the large improvement in accuracy achievable by modelling the mean flow residuals with SARIMA rather than the past demand data alone. The highly routine demand in DMA I meant that diurnal SARIMA was outperformed by the simple mean forecasting model. The results of the SARIMA model can be seen in 14. In this plot, *residuals* in the legend refers to train set errors, and *errors* refers to test set (forecasted) errors. Results for each fold are concatenated to show a 7-step week of forecasting results. The *Training Set Predictions* plot reflects the predictions of the fitted model on the data used for model training. The *Residuals* plot shows in green the residuals of the training set prediction next to the errors of each 24 hour forecast in red. *Residuals Distribution* presents the average distribution of test set errors (*errors*) for each day.

Observing the residual plots of both training and testing shows very similar distributions in all DMAs. This mitigates concerns for over fitting as performance on both the training and test set was very similar. As a result of this, the confidence bounds identified (e.g. in blue in 14) are accurate and actual data rarely breaches the 95% forecasting confidence level. DMA B and G have particularly tight confidence bounds, showing the high interval forecasting ability of models, especially for residential DMAs both near and far from cities. DMA B does however present noticeable issues for SARIMA which struggles particularly to predict daily peaks. As this is a countryside residential area, this is likely linked to the longer-term seasonality as garden related water consumption is typically much more seasonal. This explains why DMA B is the only DMA to benefit from a full 52 week training set in the later testing stage. Remaining errors are likely linked to temperatures spurring garden water use, or proximity to planting seasons.

There seems to be no clear day on which forecast errors have the same distribution for a specific day, and the mean distribution of errors for all DMAs is very close to normal, centered around zero. The remainder of residuals are likely either from unobserved variables which cannot be considered, anomalies, and more complex, non-linear relationships which neural networks are suitable for forecasting.

Although the TBATS model included several seasonalities for periods of 6, 12, 24, and 168 hours, the results were worse than SARIMA with a MAPE of 11.8%, 7.0%, 4.6%, and 4.6% in DMAs A, B, G, and I respectively, compared to 11.0%, 6.1%, 4.5%, and 4.2% with the Diurnal SARIMA model.

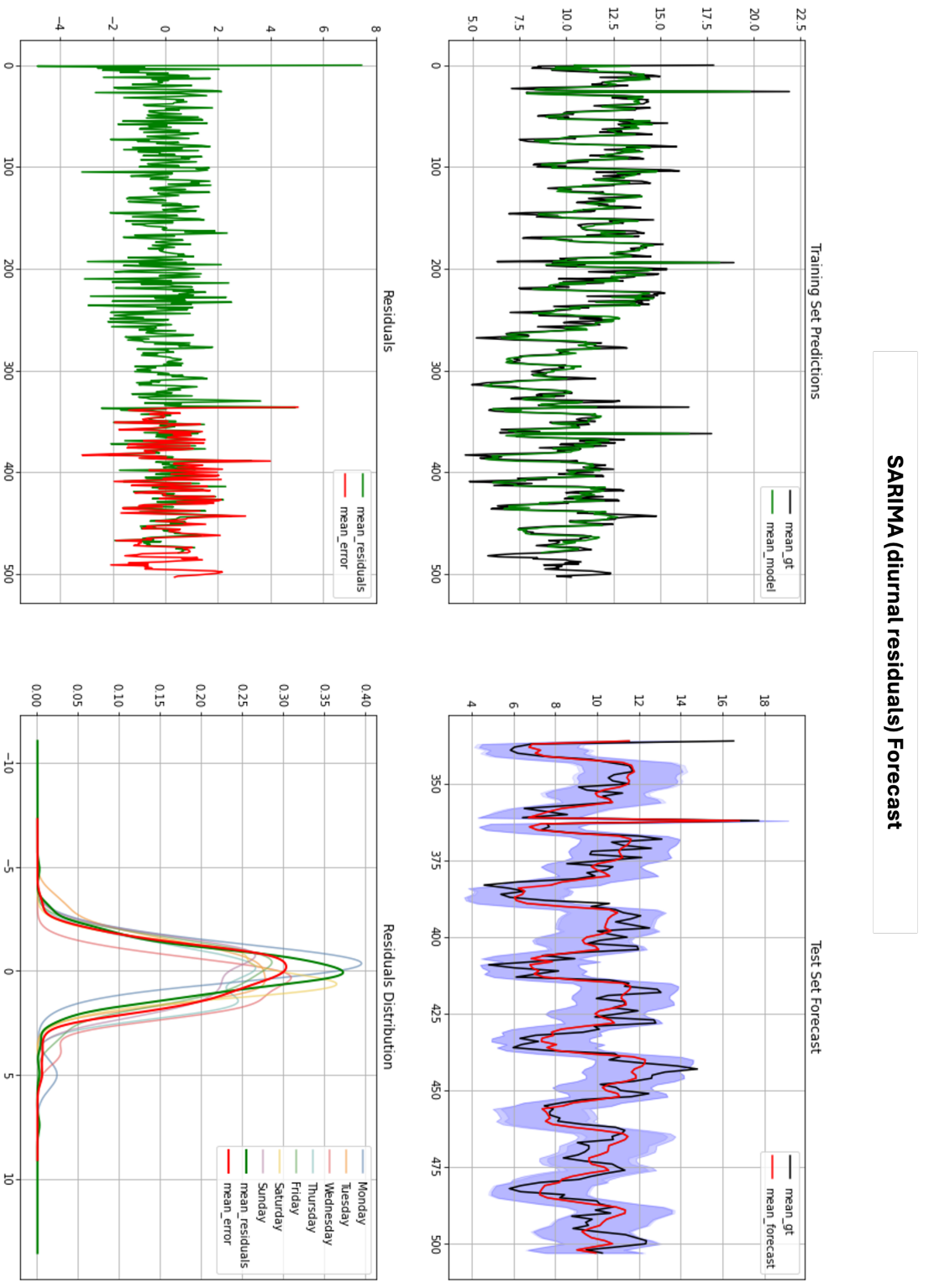

Figure 14: Plot showing forecasting ability of the most accurate SARIMA model using diurnal flow and time features where useful.

5.3. MLP with Preprocessing Techniques

Simple MLP networks were used to test multiple pre-processing techniques due to their low train times. It is anticipated that techniques which excel on simple ANNs will also show enhanced results on the most recent, more complex networks used. First testing was done to determine the best method to input data; via a sequential, windowed series, fragments, or using virtual points.

Contrasting others' findings, virtual data points do not appear to aid the forecasting of large changes in demand more accurately, as shown by Figure 15. As twice as many data points need to be forecasted, the likelihood of errors accumulating with recursive multi-step forecasting is much larger than with the other methods which explains the poor performance of virtual data points in general. Across DMA A and B, fragmenting induces a clear reduction of 6.1% and 10.9% respectively in the MSE compared to windowed data. DMA G and I instead show windowed data to be the most accurate with MSE reductions of 10.8% and 4.7% respectively when using windowed data over fragments. During further testing of decompositions, fragments produced higher accuracy in DMAs G and I, as well as much lower computation times and therefore fragments were used across all DMAs during further testing. 15 shows that both windowed and virtual methods generally over predict water demand, whereas the distribution of fragmenting residuals is centred at zero with no clear skew. Green circles in the Figure outline forecasts where fragmenting has an especially large improvement over the other two methods. At these points, the actual demand pattern is abnormal compared to the surrounding days, hinting that fragments are more likely to pick up on day-specific demand characteristics.

Decompositions were tested next including a 5 part CEEMDAN decomposition, 1 level DWT decomposition, diurnal flow removal, and SARIMA residual modelling. CEEMDAN reduced the MSE by 14% and 2% over using just fragmenting in DMA A and B respectively, beating DWT decomposition in DMA A. Alternatively, CEEMDAN with fragments increased the MSE by 25% and 31% against the base ANN with windowed data in DMA G and I respectively. DWT reduced the MSE by 5.4%, 7% and 4% over the best non-decomposed ANN method in DMA B, G, and I respectively, beating CEEMDAN decomposition in DMA B, G, and I based on the MSE. DWT improved results over the next best non-decomposed ANN method in 3 out of 4 DMAs with an average MSE reduction of 3.5%, while CEEMDAN only improved forecasts in 2 with an average detriment of 41% largely due to very poor performance in DMA G and I. Therefore, it can be concluded that DWT is the more effective decomposition of the two tested from recent literature. However, the two residual modelling methods suggested outperformed both decomposition methods with a large margin across all DMAs.

Two types of residual were tested using the simple MLP model, the water demand variance from the mean value, and the residual based on SARIMA forecasts. Diurnal residual forecasting with fragmenting reduced the MSE by 45%, 21%, and 34% in DMA A, B, and I respectively for an average improvement of 24% compared to ANNs without decompositions; outperforming SARIMA residuals in DMA I. SARIMA residual forecasting also significantly improved results, with a MSE reduction of 53.2%, 33.3%, 11.1%, and 30% in DMA A, B, G, and I respectively for an average MSE reduction of 32% over the best

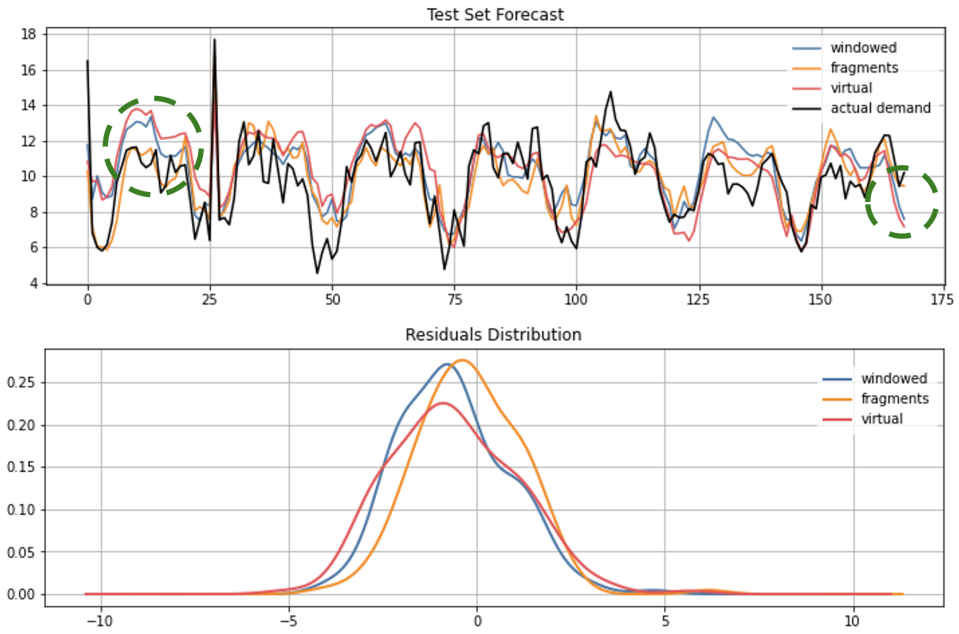

Figure 15: Comparison of forecasting performance on DMA A between windowed, fragmented, and virtual input data preprocessing. With notable differences circled in green, and forecast error distributions plotted below.

ANN without decomposition. Additionally, in all DMAs other than A, the combination of an ANN and SARIMA improved results against SARIMA alone by an average of 6.4% of MSE. Very few ANNs were able to give better results than the best SARIMA models, and when they were able to it was because of their combination with SAIRMA, or use of diurnal flow which has not been seen in recent literature. Given the similar high performance of both residual modelling methods, they were both applied during further testing.

5.4. Complex Neural Networks

Given that both SARIMA residual and Diurnal flow residual modelling showed the highest success amongst simple ANNs (MLPs), the complex networks were implemented for both of these preprocessing techniques. GRU networks were most accurate with the use of both kernel and recurrent regularisers, whereas 1D CNNs benefited most from just recurrent regularisation. Both network's forecasts and computational times improved when the number of dense layers post the CNN and GRU units was reduced from 2 to 0.

The use of complex networks to either diurnal, or SARIMA residuals generally agreed with the ANN, with the most accurate model for DMA A and B being applied to sarima residuals. However, for DMA G the best GRU and CNN models used diurnal flow, rather than SARIMA residuals. DMA I used diurnal residuals for the best GRU and CNN models. In all cases, the GRU network under-performed it's MLP counterpart. The 1D CNN showed improved results over the ANN in 50% of comparable cases with a negligible impact to MSE on average and a slight reduction in computation times. Additionally, the 1D CNN

produced improved results over the GRU network in almost all cases but only by a marginal amount with an average reduction in MSE of 5.2%. The MAPE is reduced by an average of 1% relatively when using 1D CNN over ANN. Due to the reduced computation time and slight improvement in MAPE of the 1D CNN over MLP, it is selected for the final model comparisons on the last week of available data.

The most accurate method varies largely between DMAs, with SARIMA (diurnal flow) producing the lowest MSE in DMA A, SARIMA residual MLP in DMA B, the 1D CNN in DMA G, and the mean with residual correction in DMA I. In DMA A, the percentage difference in the MSE between SARIMA and the 1D CNN is just 0.5%. In DMA B, the 1D CNN is also the best performing method with a negligible MSE difference. The only DMA with a notable difference between the top methods is DMA I where there is a 2% difference in MSE between the mean forecast and MLP. The 1D CNN also has the highest accuracy in all DMAs based on RMSE and the majority in MSE, indicating its superior ability in point forecasting and low spread of errors. A 1D CNN utilises *MaxPooling* to cut down input data while maintaining the essential features necessary for accurate prediction. This is similar to fragmenting which contains only the most relevant past data and reinforces the idea that forecasting can be improved by providing models with only the most relevant data.

The average training times of each model tested during this stage were recorded and presented in 16. The SARIMAX model tested in the next stage is also included in the comparison. The base sarima model has the lowest cross-validation time, closely followed by the 1D CNN model which took 28 seconds. It is also interesting to note the large decrease in computation times when using fragmented data over windowed, or especially virtual input techniques. Given the higher accuracy of fragmenting, the motivation for this methodology is reinforced. As the training times for both the 1D CNN and SARIMA are both very low relative to all other models, the case for prioritising SARIMA over 1D CNNs based on lower computational cost is weak.

Testing on the final week of data will be used to compare the best traditional time-series models and neural networks, as well as assessing the impact of anomaly removal, training window size, and exogenous variables. For all DMAs, SARIMA and the 1D CNN with SARIMA residuals and fragmenting is used. The 1D CNN with diurnal flow is also applied. Although, the 1D CNN was not the best ANN in DMA B and I, its results were very close to the MLP so the decreased computational time justifies its further use.

5.5. Comparison of Exogenous Variables and Train Window on Selected Models

The best model during the second to last week of data, and the final week is inconsistent in DMA A and I. Where a 1D CNN - SARIMA residual model was the most accurate before, the 1D CNN - diurnal forecasts more accurately on the following week of data. Therefore, it is neccesary to consider both methods despite one showing worse result on a particular week.

Further testing on the final week of data again proved the high accuracy of the diurnal SARIMA model, closely following the performance of the modern networks developed. The highest performing SARIMA models use 2 weeks of past data in DMA A and G, 52

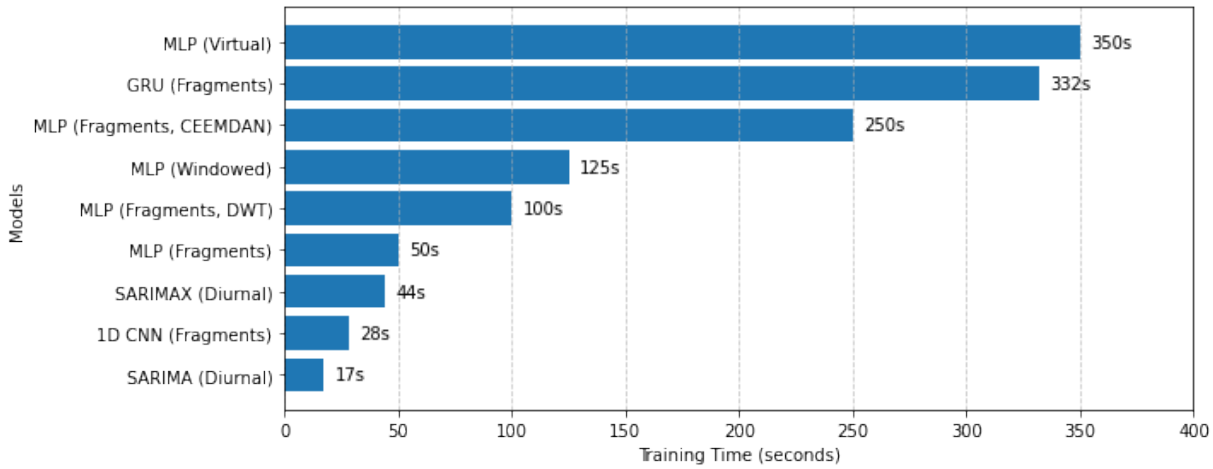

Figure 16: Comparison of training times for models tested using a single 7-fold cross validation and 8 weeks of training data.

weeks in B, and 8 weeks in DMA I according to MSE. The highest performing ANNs instead required 8 weeks of past data in all DMAs. This is expected as ANNs generally have higher data requirements. As SARIMA only uses daily seasonality, a shorter training dataset is sensible as longer-term seasonalities such as changing seasons, rising temperatures, or evolving populations do not need to be observed and may confuse short term forecasts.

After looking at the training performance of DMA B, it is evident that the water demand characteristics differ drastically across the majority of the year. The demand patterns observed just before the forecast window are only seen in past demand data roughly a year ago. This would explain the longer, 52 week training window required in DMA B; there is a recent shift to a new demand regime only observed approximately a year ago from the forecasting date.

As expected, the MAPE in DMA A is much higher than the other DMAs due to the less predictable nature of hospital water demand. Across all other DMAs, a similar level of accuracy is achievable with a lowest MAPE of 4.5%, 3.4%, and 4% in DMA B, G, and I respectively.

The inclusion of time variables with SARIMA induced no increase in forecasting accuracy in any DMAs other than B, this is likely because no holiday was observed during the test week, and the use of diurnal flow already includes variability due to weekends. The inclusion of these variables should be beneficial in the case that the flow strays from the mean by a predictable amount on different days of the week, however these results indicate that more complex patterns occur. Air temperature and humidity were most beneficial to SARIMAX results, leading to improved forecasting in DMA A and B for a reduction in MSE of 4.5% and 29% over the base SARIMA model to give an average MSE improvement of 0.4% across all DMAs. The large improvement in DMA B may reflect the increased variability in water consumption due to weather in residential areas outside the city. With more temperature based water demand from gardens or e.g. pools, it is expected that

weather has a larger impact on DMA B. In other DMAs, weather adds unrelated noise which confuses forecasting, causing large accuracy declines in DMA G and I. Despite the slight improvement in forecasting ability gained by SARIMAX with weather variables, the accuracy was still below that achieved by ANNs, other than DMA B where the MSE was reduced by 23% over the next best ANN. It is therefore concluded that SARIMAX models should be prioritised in DMAs where there is a large influence from weather factors, such as non-city residential districts, and otherwise 1D CNNs should be used.

Neither time nor weather features provide any improvement in ANN model predictions. This is consistent with other studies [36], however there are some cases where the inclusion of time variables has been beneficial. The impact of time variables may have already been incorporated in the diurnal demand, or SARIMA forecast so no result is seen in modelling the residuals. Alternatively, exogenous variables may appear useless due to their method of input. This study feeds additional exogenous features into the model using a single 2D array where each column represents a new feature. This may confuse results as the neural network cannot adequately determine the patterns in each separate series and is instead confused by the simultaneous combination of multiple variables. [32] feeds the water demand and other features into separate ANN units which are then concatenated such that each timeseries can be first considered in isolation.

The difference in computational time between the SARIMAX and 1D CNN is low, as evidenced by 16, for the same training period. Therefore the main differences in computational complexity are based on the length of training period used. The biggest difference in training times between the top models is observed in DMA B. Although the SARIMAX model provides a large accuracy improvement, it required a training period of 529 seconds - 18 times larger than the 1D CNN.

The best model is finally used to assess the importance of robust imputation and anomaly removal methods. Two datasets are provided, one with the proposed imputation and anomaly removal methods, and one with a simple mean hour and day imputation without anomaly removal. Compared to the proposed methodology, the simple imputation without anomaly removal increases the MSE by 19%, 16%, 20%, and 13% in DMA A, B, G, and I respectively. Consequently, it is concluded that without the use of a robust imputation and anomaly removal stage, results can appear significantly worse, despite perhaps more effective models in a cross-literature comparison. Without a full set of realistic data, model training is not as effective and although forecasts may be accurate, they are compared to poorly imputed data which can make good forecasts appear inaccurate.

A comparison of the forecast from each of the most successful models is displayed in 17 below with test errors compared in 18. Confidence intervals for all models are not shown as they appear almost indistinguishable between forecasts, differing only where predictions diverge. Therefore, no model has a clear advantage in interval forecasting.

5.6. Summary

A top down approach to model design was used to forecast water demand for 24 hours ahead as seen in Figure 1. Imputation methods were compared prior to any modelling and the proposed method combining seasonal decomposition, residual corrected mean, and

missForest yielded the highest accuracy with a 1.6%, 3.2%, and 5.1% error over stretches of missing data. Isolation Forest anomaly removal was then applied and removed outliers sufficiently based on data observations.

Traditional timeseries models were then developed as a benchmark for the more complex neural networks. A SARIMA model using diurnal demand residuals had the highest accuracy, and was unimproved by the inclusion of time features in a SARIMAX model. The optimal training time varied amongst DMAs, but generally short training windows of 2 weeks were sufficient. SARIMA showed results comparable with even the best ANNs developed, achieving a MAPE within 0.3% of the most accurate model for all DMAs during the final week of testing 14.

The ANN models were built up in stages, each stage seeking to determine the most effective pre processing technique from a selection. Each network built was first tuned using Bayesian hyperparameter tuning for each individual DMA and pre processing technique. Windowed, fragmented, and virtual data input methods were then tested. Fragmented data yielded the highest forecast accuracy 15 and significantly cut down computation times. CEEMDAN and DWT decomposition techniques were then tested (with fragmenting) but the training of separate models for each decomposition did not yield accurate results and had very high computation times. Instead, a SARIMA - MLP hybrid and a MLP model for diurnal demand residuals had the lowest MSE of all MLP models and was able to surpass SARIMA in all DMAs except A.

The success of the two residual modelling methods was then replicated using modern networks; GRU and 1D CNNs 9. The recurrent nature of 1D CNNs meant that these networks were reliably able to beat MLP counterparts in all DMAs. The GRU network instead hindered results and had high computation costs. GRUs were therefore not tested further. SARIMA models had the lowest computation time of 17 seconds for each cross-validation, closely followed by the 1D CNN with 28 seconds, and SARIMAX model with 44 seconds.

Finally, the last week of data available was used for testing varying training window lengths, exogenous variable use, and the impacts of anomaly and imputation methods initially applied. 1D CNNs were applied using both residual modelling methods, and although the most accurate method varied between DMAs, 8 weeks was consistently the optimal training set length. The training window for SARIMA models varied between 2 weeks in DMA A and G, and 8 to 52 weeks in DMA I and G respectively. In all DMAs, using exogenous variables with the 1D CNN was detrimental to performance. Time features hindered results across both SARIMA and 1D CNN networks (other than DMA B), however air temperature and humidity lead to improved forecasting in DMA A and B for a reduction in MSE of 4.5% and 29% over the base SARIMA model. Despite this, the SARMIA model was unable to outperform 1D CNNs, other than in DMA B where both time and weather features decreased MSE by 23% over the 1D CNN.

Simplistic data imputation and lack of anomaly removal was detrimental to results of the best performing model across all DMAs. Due to confusion during training, and mis-measurement of forecasts using poorly imputed data, the MSE across all DMAs increased by an average of 17%, showing the clear advantages of accurate imputation 5 and anomaly

removal.

The most accurate model for each DMA based on MSE and MAPE is summarised in 8 below. Additionally, the forecast made by each model and the associated errors can be observed in 17, and 18 respectively.

Table 8: Best forecasting model from each DMA and the methodology used.

DMA	Network	Data Input	Preprocess	Exog.	MAPE	MSE
A	1D CNN	Fragments	Diurnal residual	None	10.0%	1.38
B	SARIMA	Window	Diurnal residual	Time feats.	4.3%	0.46
G	1D CNN	Fragments	Diurnal residual	None	3.4%	1.60
I	1D CNN	Fragments	SARIMA residual	None	4.0%	1.56

The 1D CNN for either Diurnal or SARIMA residuals had the most accurate forecast in the majority of DMAs, with SARIMA achieving a similar level of accuracy (and the second best results) in DMA A and I. The SARIMAX model did however provide the best results in DMA B with a margin of 0.1% MAPE likely due to water demand having a higher reliance on weather variables in non-city residential DMAs. As the best 1D CNN model varied even within the same DMA for different test weeks, and SARIMA was consistently able to compete in accuracy, it is recommended that SARIMA models are still considered alongside the more complex ANN developed. Furthermore, there is inconclusive evidence to determine if modelling residuals of SARIMA or diurnal demand has better results generally. Intuitively, the diurnal demand residuals has a more repeatable pattern given that diurnal demand is unchanged week-to-week so ANNs should have a higher chance of inferring repeatable patterns.

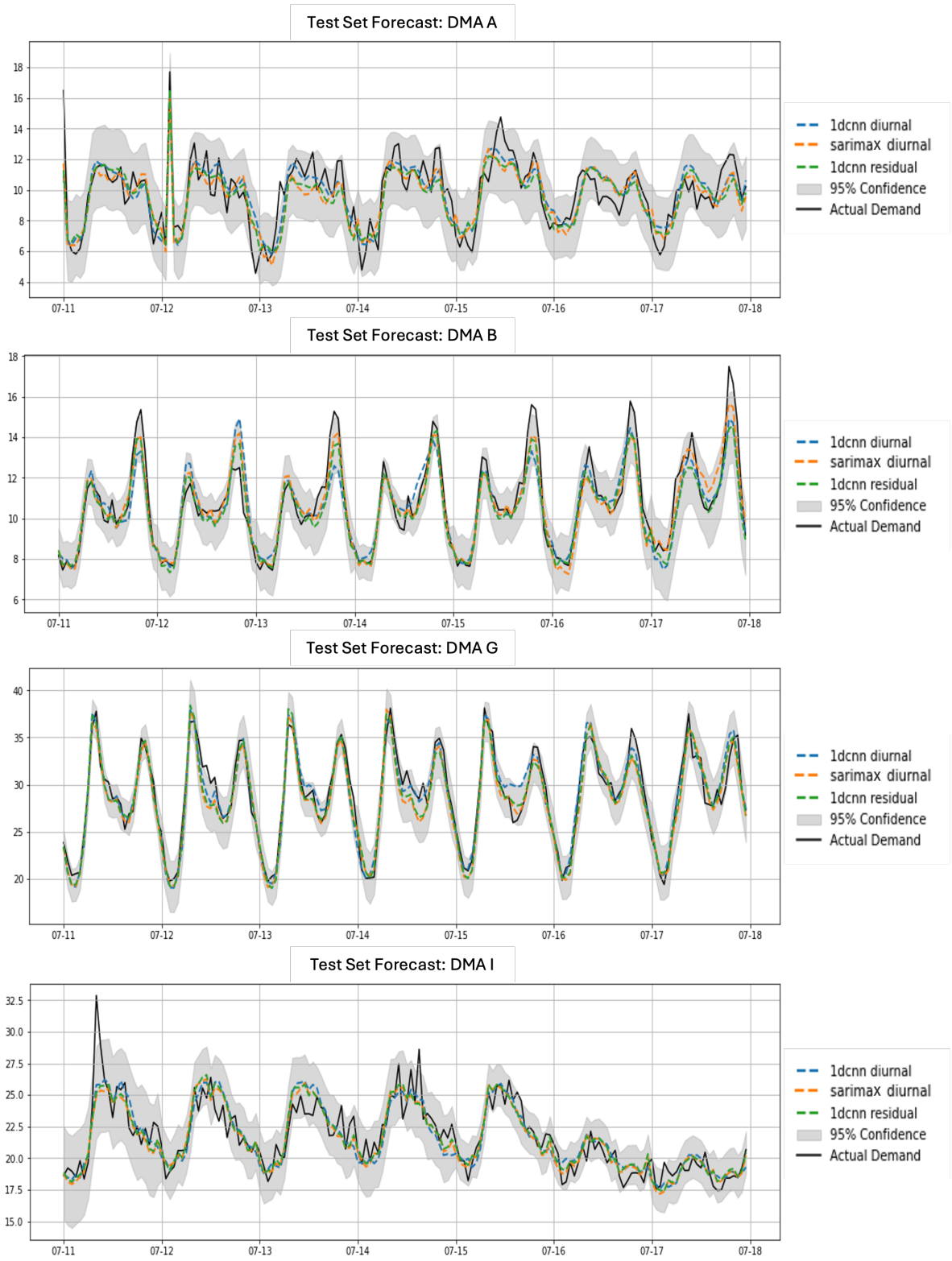

Figure 17: Top three forecasting model performance plotted against actual demand for each DMA with forecasts shown by dashed lines.

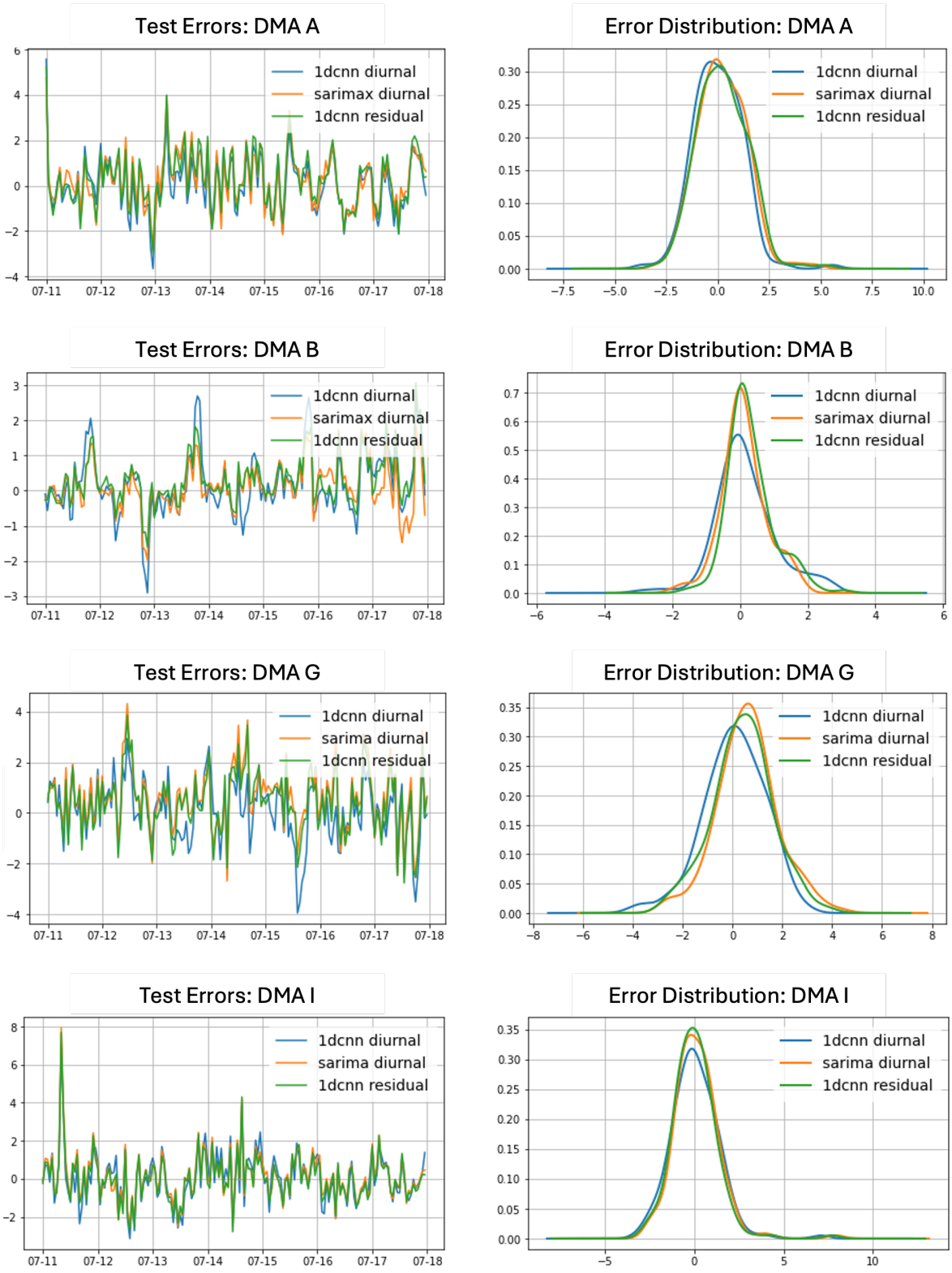

Figure 18: Forecasting errors from each of the best three models plotted both in a timeseries and as a distribution over the 168 hour forecast.

Table 9: Statistics of model prediction errors for 7-fold cross validation, 24 hour ahead forecasting on DMA A.

Model	MAE	RMSE	MAPE (%)	MSE
Stage 1				
TTS Models				
Mean	1.289	1.625	12.98	2.645
SARIMA	1.529	1.857	15.45	3.670
SARIMA (Diurnal)	*1.052*	*1.306*	*10.95*	*1.899*
TBATS (Diurnal)	1.129	1.445	11.76	2.088
Basic Neural Networks				
MLP (window)	1.773	2.070	19.47	4.345
MLP (fragments)	1.644	1.971	17.63	4.079
MLP (virtual points, window)	1.898	2.199	20.80	4.870
MLP (CEEMDAN, fragments)	1.510	1.819	15.86	3.509
MLP (DWT, fragments)	1.676	1.992	18.38	4.181
MLP (Diurnal, fragments)	1.142	1.427	11.76	2.247
MLP (SARIMA residual, fragments)	*1.055*	*1.306*	*10.92*	*1.911*
Complex Neural Networks				
GRU (SARIMA residual, fragments)	1.065	1.321	11.10	1.963
1D CNN (SARIMA residual, fragments)	*1.056*	*1.306*	*10.79*	*1.909*
GRU (diurnal residual, fragments)	1.373	1.566	14.33	2.537
1D CNN (diurnal residual, fragments)	1.134	1.390	11.89	2.212
Stage 2				
Selected Networks (Training Times)				
1D CNN (SARIMA, fragments, 2 Weeks)	0.9697	1.228	10.30	1.527
SARIMA (Diurnal, 2 Weeks)	0.9584	1.211	10.31	1.482
1D CNN (Diurnal, fragments, 8 Weeks)	**0.9190**	**1.165**	**10.03**	**1.380**
Selected Networks (Exogenous Variables)				
1D CNN (Holidays, Weekends)	0.9558	1.211	10.59	1.487
1D CNN (Air Temp.)	0.9391	1.190	10.14	1.442
SARIMA (Holidays, Weekends)	0.9627	1.212	10.36	1.484
SARIMA (Air Temp, Air Humidity)	0.9353	1.178	10.09	1.415
Selected Networks (Impute Comparison)				
1D CNN (Seasonal Impute, No Anomaly)	0.9190	1.165	10.03	1.380
1D CNN (Mean Impute, Anomaly included)	1.010	1.234	10.97	1.643

Table 10: Statistics of model prediction errors for 7-fold cross validation, 24 hour ahead forecasting on DMA B.

Model	MAE	RMSE	MAPE (%)	MSE
Stage 1				
TTS Models				
Mean	0.7419	1.081	6.838	1.169
SARIMA	1.687	2.062	14.92	4.628
SARIMA (Diurnal)	*0.6882*	*0.9618*	*6.136*	*1.039*
TBATS (Diurnal)	0.7432	1.031	6.96	1.063
Neural Networks				
MLP (window)	0.9432	1.221	9.106	1.647
MLP (fragments)	0.8545	1.140	8.020	1.469
MLP (virtual points, window)	1.049	1.324	10.29	1.817
MLP (CEEMDAN, fragments)	0.8380	1.116	7.937	1.440
MLP (DWT, fragments)	0.8440	1.132	7.811	1.390
MLP (Diurnal, fragments)	0.7205	1.008	6.567	1.164
MLP (SARIMA residual, fragments)	*0.6804*	*0.9347*	*6.161*	*0.9802*
Complex Neural Networks				
GRU (SARIMA residual, fragments)	0.6820	0.9477	6.176	0.9913
1D CNN (SARIMA residual, fragments)	*0.6753*	*0.9343*	*6.119*	*0.9808*
GRU (diurnal residual, fragments)	0.7491	1.023	6.944	1.188
1D CNN (diurnal residual, fragments)	0.7122	0.9844	6.672	1.276
Stage 2				
Selected Networks (Training Times)				
1D CNN (Diurnal, fragments, 4 Weeks)	0.6606	0.9087	5.747	0.8569
1D CNN (SARIMA, fragments, 8 Weeks)	**0.5263**	**0.7245**	**4.483**	**0.5707**
SARIMA (Diurnal, 52 Weeks)	0.5594	0.7757	4.707	0.6503
Selected Networks (Exogenous Variables)				
1D CNN (Holidays, Weekends)	0.5428	0.7490	4.626	0.5970
1D CNN (Air Temp.)	0.5443	0.7500	4.649	0.6179
SARIMA (Holidays, Weekends)	0.4937	0.6645	4.292	0.4620
SARIMA (Air Temp, Air Humidity)	0.4969	0.6626	4.328	0.4617
Selected Networks (Impute Comparison)				
1D CNN (Seasonal Impute, No Anomaly)	0.5263	0.7245	4.483	0.5707
1D CNN (Mean Impute, Anomaly included)	0.5690	0.7817	4.836	0.6634

Table 11: Statistics of model prediction errors for 7-fold cross validation, 24 hour ahead forecasting on DMA G.

Linear Models	**MAE**	**RMSE**	**MAPE (%)**	**MSE**
Stage 1				
TTS Models				
Mean	1.308	1.762	4.705	3.106
SARIMA	4.26	5.035	14.79	26.84
SARIMA (Diurnal)	*1.271*	*1.631*	*4.455*	*2.718*
TBATS (Diurnal)	1.290	1.777	4.578	3.159
Neural Networks				
MLP (window)	1.224	1.592	4.330	2.730
MLP (fragments)	1.263	1.632	4.449	3.043
MLP (virtual points, window)	1.687	2.067	6.242	4.393
MLP (CEEMDAN, fragments)	1.423	1.765	5.089	3.408
MLP (DWT, fragments)	1.135	1.513	3.981	2.538
MLP (Diurnal, fragments)	1.279	1.587	4.510	2.836
MLP (SARIMA residual, fragments)	*1.179*	*1.520*	*4.113*	*2.427*
Complex Neural Networks				
GRU (SARIMA residual, fragments)	1.199	1.532	4.205	2.449
1D CNN (SARIMA residual, fragments)	1.176	1.517	4.118	2.409
GRU (diurnal residual, fragments)	1.425	1.737	5.104	3.238
1D CNN (diurnal residual, fragments)	*1.167*	*1.465*	*4.156*	*2.369*
Stage 2				
Selected Networks (Training Times)				
SARIMA (Diurnal, 2 Weeks)	1.037	1.305	3.542	1.794
1D CNN (SARIMA, fragments, 4 Weeks)	1.001	1.249	3.434	1.622
1D CNN (Diurnal, fragments, 8 Weeks)	**0.9915**	**1.242**	**3.437**	**1.596**
Selected Networks (Exogenous Variables)				
1D CNN (Holidays, Weekends)	1.048	1.292	3.623	1.711
1D CNN (Air Temp.)	1.134	1.402	3.924	2.046
SARIMA (Holidays, Weekends)	1.144	1.417	4.012	2.066
SARIMA (Air Temp, Air Humidity)	1.096	1.381	3.839	1.971
Selected Networks (Impute Comparison)				
1D CNN (Seasonal Impute, No Anomaly)	0.9915	1.242	3.437	1.596
1D CNN (Mean Impute, Anomaly included)	1.078	1.362	3.653	1.919

Table 12: Statistics of model prediction errors for 7-fold cross validation, 24 hour ahead forecasting on DMA I.

Linear Models	**MAE**	**RMSE**	**MAPE (%)**	**MSE**
Stage 1				
TTS Models				
Mean	*0.9549*	*1.290*	*4.287*	*1.664*
SARIMA	2.415	2.903	10.43	9.760
SARIMA (Diurnal)	0.9736	1.306	4.221	1.979
TBATS (Diurnal)	1.013	1.390	4.579	1.933
Neural Networks				
MLP (window)	1.216	1.526	5.511	2.568
MLP (fragments)	1.239	1.562	5.624	2.693
MLP (virtual points, window)	1.434	1.753	6.568	3.208
MLP (CEEMDAN, fragments)	1.387	1.717	6.246	3.365
MLP (DWT, fragments)	1.163	1.507	5.239	2.465
MLP (Diurnal, fragments)	*0.9539*	*1.250*	*4.212*	*1.698*
MLP (SARIMA residual, fragments)	0.9468	1.259	4.148	1.803
Complex Neural Networks				
GRU (SARIMA residual, fragments)	0.9570	1.284	4.158	1.907
1D CNN (SARIMA residual, fragments)	0.9666	1.302	4.208	1.930
GRU (diurnal residual, fragments)	0.9349	1.261	4.067	1.834
1D CNN (diurnal residual, fragments)	*0.9245*	*1.245*	*4.035*	*1.744*
Stage 2				
Selected Networks (Training Times)				
1D CNN (SARIMA, fragments, 8 Weeks)	**0.8925**	**1.192**	**4.006**	**1.556**
1D CNN (Diurnal, fragments, 8 Weeks)	0.9884	1.277	4.479	1.746
SARIMA (Diurnal, 8 Weeks)	0.9054	1.201	4.054	1.597
Selected Networks (Exogenous Variables)				
1D CNN (Holidays, Weekends)	0.9028	1.202	4.044	1.588
1D CNN (Air Temp.)	0.9151	1.222	4.099	1.663
SARIMA (Holidays, Weekends)	1.009	1.314	4.554	1.851
SARIMA (Air Temp, Air Humidity)	1.036	1.346	4.682	1.949
Selected Networks (Impute Comparison)				
1D CNN (Seasonal Impute, No Anomaly)	0.8925	1.192	4.006	1.556
1D CNN (Mean Impute, Anomaly included)	0.9555	1.250	4.270	1.754

6. Conclusion

The objective of this study was therefore to forecast hourly water demand over the next 24 hours in a selection of DMAs by evaluating techniques from other studies with a focus on artificial neural networks. Three main gaps in the literature were identified including, a broad comparison of different neural network data preprocessing techniques for ANNs, a comparison of forecasting models on multiple DMA types, and hybridisation of data imputation and anomaly removal methods. Several key contributions are made,

1. Rigorous evaluation of different preprocessing techniques for neural networks across DMAs with differing characteristics, demonstrating the wide range of results that can be obtained based on DMA type and simple data manipulation methods.

2. A novel 1-Dimensional Convolutional Neural Network model capable of delivering superior forecasting ability over techniques examined from modern literature and traditional methods.

3. Imputation and anomaly removal techniques surpassing other methods studied and significantly improving model forecasting ability.

An array of models variants were developed, ranging from traditional predictors such as SARIMA to 1D CNNs and GRUs at the vanguard of current research. Several preprocessing techniques were combined with each neural network looking to optimise the way that data is fed into a model, the way in which data can be broken down into separate forecasts, and finally the impact of training window length and exogenous variables. The tested methods showed inconsistent performance across DMAs, with certain techniques showing promise in specific DMA types (such as a hospital district), but not others (such as residential districts). Neural networks yielded the best results in the majority of DMAs, however the performance of SARIMA was undefeated for DMAs where weather variables are particularly impacting (e.g. residential DMAs with a high presence of gardens). This highlights the need for studies to test models across a selection of DMAs and determine which DMA types can be accurately forecasted. This exploration also outlines the simple methods that can be used with almost any neural network to easily boost performance, notably fragmenting and diurnal demand.

Two promising model configurations emerged, both of which were 1D CNNs using fragmented data and residual modelling. The benefit of recurrent networks was reinforced, however this was only the case for 1D CNNs which outperformed their MLP couterparts. Results varied between DMAs, but 1D CNN predictions based on SARIMAX and diurnal demand residuals were consistently the most accurate based on RMSE, MAPE, and MSE. Although their results were closely followed by the novel SARIMA variant developed, in the majority of tests 1D CNNs produced superior accuracy with comparable computation times. Neither of these residual modelling techniques have been seen in the literature and therefore contribute to current research. 8 weeks was identified as the optimal training data length for ANNs, and exogenous variables were shown to hinder results. The strong

performance of SARIMA and its combination with different exogenous variables should also be noted due to its consistently accurate predictions, particularly in residential DMAs in the countryside. Therefore, it is recommended to apply all three models for a robust forecast. Although, 1D CNNs had the highest accuracy in most cases, it was unclear which residual modelling technique yields the best results and SARIMAX often had the second best forecast of the three models.

Before model development, a number of data imputation methods were tested and a novel method involving missForest and seasonal decomposition filled missing data points with the highest accuracy. Additionally, an Isolation Forest was applied to correct anomalous data. The impacts of this were assessed on the best models by comparing simple imputation methods from other studies with the proposed imputation and anomaly removal procedure. It is evident that the proposed method significantly reduces forecast error, reducing the MSE by an average of 17% over all DMAs.

This study lays the groundwork for future research exploring the interaction of different pre-processing techniques outside the scope of this report, and further experimentation with the optimal exogenous variable inclusion methodology. The following points were also identified which may lead to higher model accuracy in future studies:

- [9] presents promising resutls by using a graph CNN based on the correlation between different DMAs. While this report leverages the similarity between DMAs to impute data, it does not explore the use of multiple DMA demands to aid forecasting.

- To fully conclude the superiority of 1D CNNs against SAIMAX, further testing should be applied for higher seasonalities such as weekly or annually rather than solely daily.

- The SARIMA - 1D CNN model consistently gave highly accurate forecasts, however the combination of SARIMAX and 1D CNN was not evaluated despite SARIMAX typically outperforming SARIMA. Future works could also look at the hybridisation of the 1D CNN with other models as very few reports have covered this area.

Acknowledgements

Guidance from Dr Aly-Joy Ulusoy, Bradley Jenks, and Carlos Jara Arriagada was crucial during the development of forecasting models and heavily inspired the direction of this project. Their unparalleled supervision and persistent support make me hugely thankful that I switched to the Wastewater module and met the team. Many thanks also to Dr Ivan Stoianov who created this project and has provided key insights.

References

[1] H. Namdari, A. Haghighi, and S. M. Ashrafi, "Short-term urban water demand forecasting; application of 1d convolutional neural network (1d cnn) in comparison with different deep learning schemes," *Stochastic Environmental Research and Risk Assessment*, Sep 2023.

[2] D. C. M. Ristow, E. Henning, A. Kalbusch, and C. E. Petersen, "Models for forecasting water demand using time series analysis: a case study in southern brazil," *Journal of Water, Sanitation and Hygiene for Development*, vol. 11, p. 231–240, Jan 2021.

[3] I. Ghalehkhondabi, E. Ardjmand, W. A. Young, and G. R. Weckman, "Water demand forecasting: review of soft computing methods," *Environmental Monitoring and Assessment*, vol. 189, Jun 2017.

[4] T. Salloom, O. Kaynak, and W. He, "A novel deep neural network architecture for real-time water demand forecasting," *Journal of Hydrology*, vol. 599, p. 126353, 2021. ID: 271842.

[5] M. Bata, R. Carriveau, and D. S.-K. Ting, "Short-term water demand forecasting using hybrid supervised and unsupervised machine learning model," *Smart Water*, vol. 5, May 2020.

[6] A.-B. Al-Ghamdi, S. Kamel, and M. Khayyat, "A hybrid neural network-based approach for forecasting water demand," *Computers, Materials & Continua*, vol. 73, 05 2022.

[7] A. Anele, Y. Hamam, A. Abu-Mahfouz, and E. Todini, "Overview, comparative assessment and recommendations of forecasting models for short-term water demand prediction," *Water*, vol. 9, p. 887, Nov 2017.

[8] M. G. Shirkoohi, M. Doghri, and S. Duchesne, "Short-term water demand predictions coupling an artificial neural network model and a genetic algorithm," *Water Supply*, vol. 21, p. 2374–2386, Feb 2021.

[9] A. Zanfei, B. M. Brentan, A. Menapace, M. Righetti, and M. Herrera, "Graph convolutional recurrent neural networks for water demand forecasting," *Water Resources Research*, vol. 58, Jul 2022.

[10] I. J. Herrera M., Torgo L. and P.-G. R., "Predictive models for forecasting hourly urban water demand," *Journal of Hydrology*, vol. 387, pp. 141–150, 2010.

[11] E. Donkor, T. Mazzuchi, R. Soyer, and A. Roberson, "Urban water demand forecasting: Review of methods and models," *Journal of Water Resources Planning and Management*, vol. 140, pp. 146–159, 02 2014.

[12] A. Niknam, H. K. Zare, H. Hosseininasab, A. Mostafaeipour, and M. Herrera, "A critical review of short-term water demand forecasting tools—what method should i use?," *Sustainability*, vol. 14, p. 5412, Apr 2022.

[13] G. Guo, S. Liu, Y. Wu, J. Li, R. Zhou, and X. Zhu, "Short-term water demand forecast based on deep learning method," *Journal of Water Resources Planning and Management*, vol. 144, no. 12, p. 04018076, 2018.

[14] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, 2021. ID: 272413.

[15] S. Zhou, S. Guo, B. Du, S. Huang, and J. Guo, "A hybrid framework for multivariate time series forecasting of daily urban water demand using attention-based convolutional neural network and long short-term memory network," vol. 14, p. 11086–11086, Sep 2022.

[16] M. Herrera, J. C. García-Díaz, J. Izquierdo, and R. Pérez-García, "Municipal water demand forecasting: Tools for intervention time series," *Stochastic Analysis and Applications*, vol. 29, p. 998–1007, Nov 2011.

[17] D. Kofinas, N. Mellios, E. Papageorgiou, and C. Laspidou, "Urban water demand forecasting for the island of skiathos," *Procedia Engineering*, vol. 89, pp. 1023–1030, 2014. ID: 278653.

[18] B. Du, Q. Zhou, J. Guo, S. Guo, and L. Wang, "Deep learning with long short-term memory neural networks combining wavelet transform and principal component analysis for daily urban water demand forecasting," *Expert Systems with Applications*, vol. 171, p. 114571, 2021. ID: 271506.

[19] K.-M. Koo, K.-H. Han, K.-S. Jun, G. Lee, J.-S. Kim, and K.-T. Yum, "Performance assessment for short-term water demand forecasting models on distinctive water uses in korea," *Sustainability*, vol. 13, no. 11, p. 6056, 2021.

[20] J. Chen and D. Boccelli, "Demand forecasting for water distribution systems," *Procedia Engineering*, vol. 70, pp. 339–342, 2014.

[21] E. Arandia, A. Ba, B. Eck, and S. McKenna, "Tailoring seasonal time series models to forecast short-term water demand," *Journal of Water Resources Planning and Management*, vol. 142, no. 3, p. 04015067, 2016.

[22] M. Braun, T. Bernard, O. Piller, and F. Sedehizade, "24-hours demand forecasting based on sarima and support vector machines," *Procedia Engineering*, vol. 89, pp. 926–933, 2014. ID: 278653.

[23] I. Okeya, Z. Kapelan, C. Hutton, and D. Naga, "Online modelling of water distribution system using data assimilation," *Procedia Engineering*, vol. 70, pp. 1261–1270, 2014.

[24] M. E. Banihabib and P. Mousavi-Mirkalaei, "Extended linear and non-linear autoregressive models for forecasting the urban water consumption of a fast-growing city in an arid region," *Sustainable Cities and Society*, vol. 48, p. 101585, 2019.

[25] M. Xenochristou, C. Hutton, J. Hofman, and Z. Kapelan, "Short-term forecasting of household water demand in the uk using an interpretable machine learning approach," *Journal of Water Resources Planning and Management*, vol. 147, p. 04021004, 04 2021.

[26] A. Carriero, G. Kapetanios, and M. Marcellino, "Forecasting large datasets with bayesian reduced rank multivariate models," *Journal of Applied Econometrics*, vol. 26, p. 735–761, Jan 2010.

[27] Z. Zhou, F. Xiong, B. Huang, C. Xu, R. Jiao, B. Liao, Z. Yin, and J. Li, "Game-theoretical energy management for energy internet with big data-based renewable power forecasting," *IEEE Access*, vol. 5, pp. 5731–5746, 2017.

[28] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "Demand forecast of pv integrated bioclimatic buildings using ensemble framework," *Applied Energy*, vol. 208, pp. 1626–1638, 2017.

[29] A. Zanfei, A. Menapace, B. M. Brentan, and M. Righetti, "How does missing data imputation affect the forecasting of urban water demand?," *Journal of Water Resources Planning and Management*, vol. 148, Nov 2022.

[30] N. Umar and A. Gray, "Comparing single and multiple imputation approaches for missing values in univariate and multivariate water level data," *Water*, vol. 15, 2023.

[31] P. D. Jadhav A. and R. K., "Comparison of performance of data imputation methods for numeric dataset.," *Applied Artificial Intelligence*, vol. 33, p. 913–933, 2019.

[32] S. Hu, J. Gao, D. Zhong, L. Deng, C. Ou, and P. Xin, "An innovative hourly water demand forecasting preprocessing framework with local outlier correction and adaptive decomposition techniques," *Water*, vol. 13, p. 582–582, Feb 2021.

[33] A. Zanfei, B. M. Brentan, A. Menapace, and M. Righetti, "A short-term water demand forecasting model using multivariate long short-term memory with meteorological data," *Journal of Hydroinformatics*, vol. 24, p. 1053–1065, Sep 2022.

[34] K. Wang, Z. Ye, Z. Wang, B. Liu, and T. Feng, "Macla-lstm: A novel approach for forecasting water demand," *Sustainability*, vol. 15, p. 3628–3628, Feb 2023.

[35] G. Guo, S. Liu, Y. Wu, J. Li, R. Zhou, and X. Zhu, "Short-term water demand forecast based on deep learning method," *Journal of Water Resources Planning and Management*, vol. 144, p. 04018076, Dec 2018.

[36] L. Mu, F. Zheng, R. Tao, Z. Qingzhou, and Z. Kapelan, "Hourly and daily urban water demand predictions using a long short-term memory based model," *Journal of Water Resources Planning and Management*, vol. 146, p. 05020017, 09 2020.

[37] D. Cheong Kok Weng, "Comparison of statistical and machine learning methods for multi-step short term water demand interval forecasting," 2023.

[38] LaminarBlock, "oscarwoodi/water-demand-forecasting-neural-networks," Jun 2024.

Appendix

Appendix A. Acronym Glossary

DMA District Metering Area
ARIMA Autoregressive Integrated Moving Average
SARIMA Seasonal Autoregressive Integrated Moving Average
ANN Artificial Neural Network
MLP Multilayer Perceptron
LSTM Long Short-Term Memory
GRU Gated Recurrence Unit
1D CNN 1 Dimensional Convolutional Neural Network
KNN K Nearest Neighbours
DWT Discrete Wavelet Transform
CEEMDAN Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
MAE Mean Absolute Error
MAPE Mean Absolute Percentage Error
MSE Mean Squared Error
RMSE Root Mean Squared Error

Appendix B. Imputation Methods

Deletion

Deletion is appropriate when the amount of missing data is very small, such that it represents a negligible portion of the total dataset. It is also used when the missing data is Missing Completely at Random (MCAR), meaning that there are no patterns in the missing data.

Last Observation Carried Forward / Next Observation Carried Backwards

LOCF can be useful in time-series data where there is a trend or pattern that continues from one time point to the next. It's also useful when the data is collected at regular intervals, and the values are relatively stable or change slowly over time.

Mean / Median Day Fill

Mean/Median/Mode imputation can be used when the data is missing completely at random (MCAR). If the missing data is not random, this method can lead to biased and inefficient estimates. Mean/Median/Mode imputation can lead to an underestimate of the variance, which can lead to biased estimates of the covariances and correlations. This can affect the model's ability to accurately capture the relationships between variables. Therefore, it's important to carefully consider the nature of the missing data before using this method.

Rolling Statistics

Rolling Statistics Imputation is particularly effective for time series data with temporal dependencies. It can handle situations where the missing data is Missing At Random (MAR) or Not Missing At Random (NMAR). This method assumes that nearby points in time are more similar to each other, which often holds true for time series data.

Linear Interpolation

Linear interpolation is suitable when the data shows a linear trend between the observations. It's not suitable for data that shows a nonlinear trend or seasonal pattern as it will almost distort the seasonal pattern at all.

Spline Interpolation

Spline interpolation is used when the data is numeric and continuous, and you want to avoid the problem of over fitting that can occur with polynomial interpolation. Provides a smoother fit than linear interpolation.

Regression Imputation

Regression imputation can be used when the data is numeric and there is a strong correlation between the variable with missing values and other variables. Makes use of the relationship between variables. Regression imputation can lead to more accurate models by making use of the relationships between variables. However, it can also lead to an overestimate of the correlation between variables, which can affect the model's performance.

KNN

The KNN imputation method works by finding the K-nearest neighbors to the observation with missing data and then imputing them based on the non-missing values in the neighbors. KNN can be used when the data are missing at random, and the variables are almost equally important. KNN is better when the data are uniformly distributed, and it's not suitable for high dimensional data. KNN imputation can introduce bias into the data if the missing data mechanism isn't random. This is because the KNN algorithm relies on the assumption that similar observations exist in the dataset. If this assumption is violated, the imputed values may be far from the true values, leading to biased estimates in the model-building process.

Appendix C. Model Descriptions

ANN

An artificial neural network (ANN) is a network of interconnected artificial neurons that individually perform nonlinear computations based on input values, transmitting these results to other neurons. These neurons are structured in layers, forming a complex web of connections. ANNs function through a sequence of steps:

- **Hidden Layers:** Intermediate layers between the input and output layers where complex computations occur. These layers identify patterns and relationships within the input data.
- **Input Layer:** Receives data, which is then transmitted to the neurons in the next layer.
- **Hidden Layers:** Intermediate layers between the input and output layers where complex computations occur. These layers identify patterns and relationships within the input data.
- **Output Layer:** Neurons here provide the result of prediction based on the processed information from the hidden layers.

After sufficient training via back propagation, this network can reveal hidden and complex correlations and mechanics in the training data which can help us to understand short term demand patterns more intuitively.

GRU

The Gated Recurrence Unit, GRU, can consider previous information (similarly to a LSTM) and is therefore appropriate for time-series data analysis. Reset and update gate phases avoid the vanishing and exploding gradient problems. Additionally, this structure enables model complexity to be reduced significantly while maintaining highly accurate results.

- **Update gate:** Each iteration, new information enters the GRU alongside the information in the memory cell related to the past. The update gate decides what proportion of information is transferred into future iterations.
- **Reset gate:** This gate determines what information should be deleted based on the new input and the information in the memory cell of the past. A candidate memory cell is then generated using the remaining data and information of the memory cell's previous time step. The update cell gate then dictates how much of the candidate memory cells information is carried forward to obtain the memory cell at time t. This memory is then used as a hidden state, transformed with an activation function, and used as the output at t+1.

CNN

Two-dimensional convolutional neural networks are a variant of deep neural networks which are capable of analysing images and videos for image and video recognition, image classification, medical image analysis, and natural language processing. The networks are part of feed forward ANN which utilise convolution and pooling layers. The inputs important features are extracted in the pooling layer which enables a reduced computational burden. Finally, the output of the convolution and pooling layers is converted into a vector and imported as an input into a fully connected MLP network to form the prediction of

the desired variable. 1D convolutional neural networks are a modified version of the typical 2D CNNs. 1D arrays are used to replace the 2D matrices in convolution and pooling layers to prepare the feature map. This enables a computational complexity of O(NK) rather than $O(N^2K^2)$ for 2D CNNs. Additionally, past studies have demonstrated that 1D CNN models have a shallow architecture, but in 1D CNN, a deep architecture is needed to achieve reasonable accuracy.

Appendix D. Fitted Hyperparameter Values

Table D.13: Ranges and final values of hyperparameters for each model using DMA A / B / G / I.

Model	**Hyperparameter**	**Range**	**Value (A/B/G/I)**
Mean	Residual window length	[1, 12, 24, 48, 168]	48
SARIMA			
	p	0 - 5	1 / 4 / 3 / 1
	d	0 - 5	0 / 0 / 0 / 0
	q	0 - 5	1 / 0 / 0 / 0
	P	0 - 4	0 / 0 / 0 / 0
	D	0 - 4	0 / 0 / 0 / 0
	Q	0 - 4	2 / 1 / 1 / 1
	seasonality	[24, 168]	24
SARIMA (Dirunal)			
	p	0 - 5	2 / 3 / 4 / 1
	d	0 - 5	1 / 0 / 0 / 0
	q	0 - 5	1 / 0 / 0 / 1
	P	0 - 4	4 / 0 / 0 / 0
	D	0 - 4	0 / 0 / 0 / 0
	Q	0 - 4	0 / 3 / 1 / 1
	seasonality	[24, 168]	24

Table D.14: Ranges and final values of hyperparameters for each model using DMA A / B / G / I.

Model	Hyperparameter	Range	Value (A/B/G/I)
ANN (window)			
	Batch size	[16, 24, 48]	16 / 16 / 24 / 16
	Patience	5 - 15	11 / 5 / 13 / 14
	Learning rate	0.0001 - 0.01	0.0005 / 2e-5 / 3e-5 / 0.0005
	Dropout	0.0 - 0.5	0.5 / 0.21 / 0.18 / 0.33
	Layer 1 units	[32, 64, 96, 160]	160 / 96 / 160 / 96
	Layer 2 units	[32, 64, 96, 160]	160 / 96 / 160 / 160
	Layer 3 units	[32, 64, 96, 160]	96 / 32 / 32 / 32
ANN (fragments)			
	Batch size	[16, 24, 48]	16 / 24 / 16 / 24
	Patience	5 - 15	11 / 11 / 5 / 11
	Learning rate	0.0001 - 0.01	0.0007 / 0.0005 / 5e-5 / 2e-5
	Dropout	0.0 - 0.5	0.5 / 0.15 / 0.05 / 0.25
	Layer 1 units	[32, 64, 96, 160]	64 / 32 / 64 / 160
	Layer 2 units	[32, 64, 96, 160]	64 / 64 / 96 / 64
	Layer 3 units	[32, 64, 96, 160]	96 / 64 / 64 / 32
ANN (Diurnal, fragments)			
	Batch size	[16, 24, 48]	16 / 48 / 16 / 24
	Patience	5 - 15	7 / 11 / 9 / 5
	Learning rate	0.0001 - 0.01	0.001 / 0.0005 / 0.0001 / 0.0001
	Dropout	0.0 - 0.5	0.27 / 0.20 / 0.35 / 0.5
	Layer 1 units	[32, 64, 96, 160]	160 / 160 / 64 / 64
	Layer 2 units	[32, 64, 96, 160]	160 / 160 / 96 / 96
	Layer 3 units	[32, 64, 96, 160]	160 / 64 / 96 / 160
ANN (SARIMA residual, fragments)			
	Batch size	[16, 24, 48]	16 / 48 / 16 / 24
	Patience	5 - 15	5 / 15 / 7 / 11
	Learning rate	0.0001 - 0.01	0.0002 / 1e-5 / 0.0002 / 0.001
	Dropout	0.0 - 0.5	0.0 / 0.4 / 0.15 / 0.5
	Layer 1 units	[32, 64, 96, 160]	96 / 64 / 64 / 160
	Layer 2 units	[32, 64, 96, 160]	160 / 160 / 64 / 96
	Layer 3 units	[32, 64, 96, 160]	64 / 96 / 96 / 64

Table D.15: Ranges and final values of hyperparameters for each model using DMA A / B / G / I.

Model	Hyperparameter	Range	Value (A/B/G/I)
GRU (SARIMA residual, fragments)			
	Batch size	[16, 24, 48]	16 / 24 / 16 / 24
	Patience	5 - 15	13 / 6 / 8 / 7
	Learning rate	0.0001 - 0.01	0.0001 / 0.0001 / 0.001 / 0.001
	Dropout	0.0 - 0.5	0.12 / 0.5 / 0.2 / 0.4
	Layer 1 GRU units	[4, 8, 16, 32, 64]	5 / 32 / 16 / 64
	Layer 2 GRU units	[4, 8, 16, 32, 64]	4 / 8 / 64 / 64
1D CNN (SARIMA residual, fragments)			
	Batch size	[16, 32, 64]	32
	Patience	5 - 15	15
	Learning rate	0.0001 - 0.01	0.0003
	Filters	[4, 8, 16]	16
	Kernel Size	3 - 5	3
	Stride	1 - 2	1
	No. Conv. Layers	1 - 3	2
	Dropout	0.0 - 0.5	0.0
	Dense Layer Units	[16, 32, 64]	64
GRU (DIURNAL residual, fragments)			
	Batch size	[16, 24, 48]	16 / 24 / 16 / 24
	Patience	5 - 15	13 / 6 / 8 / 7
	Learning rate	0.0001 - 0.01	0.0001 / 0.0001 / 0.001 / 0.001
	Dropout	0.0 - 0.5	0.12 / 0.5 / 0.2 / 0.4
	Layer 1 GRU units	[4, 8, 16, 32, 64]	5 / 32 / 16 / 64
	Layer 2 GRU units	[4, 8, 16, 32, 64]	4 / 8 / 64 / 64
1D CNN (DIURNAL residual, fragments)			
	Batch size	[16, 32, 64]	32
	Patience	5 - 15	15
	Learning rate	0.0001 - 0.01	0.0003
	Filters	[4, 8, 16]	16
	Kernel Size	3 - 5	3
	Stride	1 - 2	1
	No. Conv. Layers	1 - 3	2
	Dropout	0.0 - 0.5	0.0
	Dense Layer Units	[16, 32, 64]	64

Appendix E. Alternate Methods Tested

Appendix E.1. Mean Day with Residual Correction Imputation

A further iteration on the typical imputation techniques viewed in the literature is developed by combining average day statistics with a rolling window and residual correction. The first component of the model includes the average, over a set window, of the water demand for the hour and day of week for which the imputed data point covers. A second term is then summed to this which takes into account under or over estimations seen in the recent data points, based on an application of the first component alone. This adjustment is again an exponentially weighted average of the errors committed by the first component in the most recent hours. E.g. if water demand in the past few hours is persistently above its weighted mean for each hour and day, a positive residual correction will shift the imputed values up. This is written mathematically as,

$$W_{i+1} = f_1((W_{i+1-k\times 24\times 7})_{k=0}^{p} + f_2((e_{i-l})_{l=0}^{q})$$

where p and q are memory parameters, and $f_1()$ and $f_2()$ are averaging functions. The term $f_1((W_{i+1-k\times 24\times 7})_{k=0}^{p}$ represents the average demand values for hour i (out of all 168 hours in a week), calculated using the past p weeks of data. The error term e_i is calculated as, $e_i = f_1((W_{i-k\times 24\times 7})_{k=0}^{p}) - W_i$ - an average over the past q hours of the calculated average demand value for hour i minus the actual demand value at hour i. This formula can be iteratively applied for multiple missing values, however this assumes that there are enough available lagged data points available to give a mean value of the demand at a set day and hour. This method is named *Mean Day with Residual Correction Imputation*.

Appendix E.2. TBATS

Multiple seasonality can also be modelled using the PyPi *TBATS* library which is also fitted and tested by using a grid search automatically performed by the package. The impact of Box-Cox transforms to encourage a normal distribution was evaluated in initial testing but hindered results across all models, possibly explaining the lack of use in other studies for SARIMA. A summary of the models is given in 7 below. A training window of 4 weeks is initially chosen based on manual trial and error between 1, 4, 8, 32, and 52 weeks.

Appendix E.3. Isolation Forest

Isolation Forest is an ensemble tree-based method based on the idea that abnormal data usually takes a small proportion of the whole data set and tends to alienate the normal data in the feature space. Instead of first constructing a distribution of what the underlying data should follow, this method isolates using decision trees in a forest. This is important as the data provided often strays from any identifiable statistical distribution 4 and so many other anomaly methods would be inappropriate. A sub sample from the training data set is selected and used to build a decision tree by randomly splitting a value of demand between its minimum and the maximum in the hourly group. This is repeated

to produce multiple trees in a forest of a desired size. The anomaly score of each point is then found as:

$$s(x, n) = 2^{-E(h(x))/c(n)}$$

where $h(x)$ is the number of edges that a point x traverses from the root node until the traversal is terminated at an external node in the isolation tree. $E(h(x))$ is the expected path length of point x in all the isolation trees, n is the number of instances, and $c(n)$ is the average path length of n given points. Points are then considered anomalies if their anomaly score exceeds a certain threshold determined via trial and error. The threshold of the anomaly score is determined by the contamination level of anomalies. The contamination level indicates the percentage of the dataset that contains outliers and is calibrated through trial and error. A contamination level of 0.5% yields satisfactory results as judged by a visual inspection of the outliers marked.

Appendix E.4. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

Water Demand data can include large amounts of noise, particularly for DMAs with few, large users where there is a less predictable consumption pattern, e.g. DMA A. Noise can limit the capture of dependencies between future and past behaviour of the water demand time-series and therefore the decomposition of linear and non-linear aspects is useful. Separate models can be utilised to predict noise and more repeatable, periodic aspects.

Empirical Mode Decomposition (EMD) is utilised to decompose a nonlinear and non stationary signal into several series called Intrinsic Mode Functions (IMFs) alongside a residual. An example of this decomposition for DMA A can be seen in the Figure below E.19. Each IMF exhibits characteristics of local oscillation frequency of the original signal. This gives a set of the underlying oscillatory components of a series without prior knowledge of the time-series' underlying characteristics. This technique is particularly effective in analysing non-stationary signals such as the water demand data. The EMD algorithm involves decomposing the signal into a series of IMFs, which are obtained through a process called sifting. Sifting encompasses identifying the local extrema of the signal and fitting an envelope to this extrema (by connecting extrema). The difference between the signal and its envelope is then computed, and the process is repeated on the difference until a locally smooth IMF is obtained.

The resulting IMFs represent different oscillations scales in the signal, with the first IMF capturing the highest-frequency oscillations and the last IMF capturing the lowest-frequency oscillations. The final residual signal, obtained after all the IMFs have been extracted, represents the trend of the original signal. As a five-part decomposition is used, five forecasting models are trained, one for each component, to produce five separate forecasts which are summed.

EMD has the issue of mode mixing, where and IMF may contain signals with large variance in oscillations, or similar oscillations at the same position in multiple IMFs. EEMD overcomes this issue by adding white noise with finite variance to the original signal. Despite the issue of mode-mixing being partially removed by EEMD, the addition of white

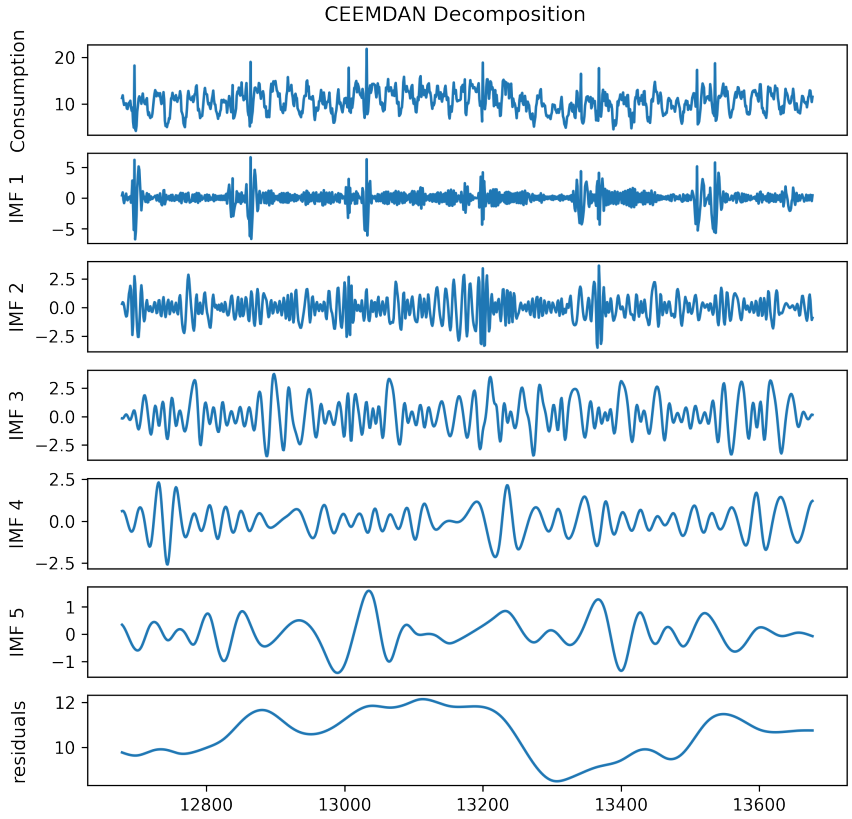

Figure E.19: Original signals, their corresponding Intrinsic Mode Functions (IMFs), and residue results of Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) for time series with Isolation Forest applied.

noise reduces precision of decompositions. Therefore, CEEMDAN introduces adaptive noise; the addition of a particular noise at each decomposition stage.

Frequency analysis is powerful for understanding the characteristics of a signal with strong periodicity. Given several obvious periodicities present in water consumption, training a model on a representation of each can eliminate some of the non-linearity which traditional models struggle to predict and make it easier to extract the sub-signals processed by CEEMDAN [32].

Appendix E.5. Virtual Data

Virtual data is implemented by linearly interpolating to find a selected number of *virtual* values between each set of two real demand values. [4] finds that just a single virtual point enhances the prediction accuracy the most and improves the models ability to forecast

highly non-linear data. This study will therefore utilise a single inserted virtual value.

Inserting virtual values enlarges the data input, thus increasing the training and forecasting time which should be considered in assessing feasibility. With just a single virtual point, the input doubles in length, and twice as many single-step forecasts are required (47 steps, one for each half-hour) to cover the following 24 hour period. Each intermediate point from the forecast is then removed to give a final set of 24 predicted values covering the upcoming day. DMAs with largely non-linear, erratic demand patterns such as the hospital district are expected to benefit the most from this method due to the large spikes in demand observed 2.

Appendix E.6. Discrete Wavelet Transforms

Discrete Wavelet Transforms (DWT) allow a more stable, low-volatility version of the time-series to be extracted in addition to the remaining noise component. A 'low-pass' filter is used to decompose the series into an *approximations* component which represents the low-frequency aspects of the data with the main features of the original series. Additionally, detail components are derived via a 'high-pass' filter and correspond to the high frequency aspects.

The derivation of a single detail and approximation component from a series is a single level decomposition. The process of multiple decomposition levels is shown in the Figure below E.20. The raw series X is decomposed into $X = A_1 + D_1$. The same decomposition is then repeated on A_1 for level 2 to give $X = A_2 + D_1 + D_2$. This can be repeated for as many levels as necessary. Two forecasting models are trained, one for each component, to produce two separate forecasts which are summed.

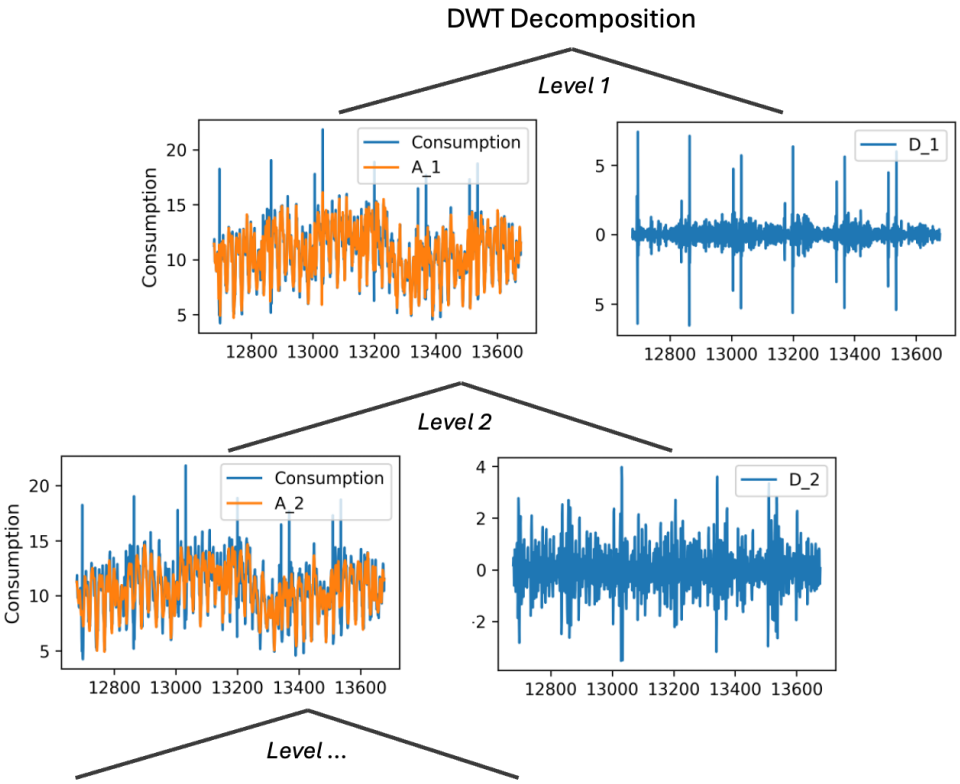

Figure E.20: Original signals, their corresponding DWT approximation and detail components at different levels of decomposition. The approximate component is shown on the left branches while the right branches show the detail components separated.

In order to utilise the DWT transform for forecasting, separate models are developed for both the approximate component and the remaining residual. The residual is given as $R = X - A_n$ where X is the actual consumption and A_n is the approximate wavelet series for the nth level of decomposition.

There is no clear theory to select the appropriate amount of decompositions for each time-series. [18] utilises the RMSE and signal-noise ratio as evaluation indexes to compare the performance of different decomposition levels. The level of volatility in the approximate series declines as the number of decompositions increases, therefore it is logical that the higher the level decomposition, the better the forecasting behaviour. This is however balanced by a more complex residual. The above paper finds the optimal decomposition level as 1 with a *db4* wavelet (Daubechies wavelet of order 4). As 1 level shows clear superiority, it will be utilised in this study without further testing for other levels. The residual series is then found using the imputed, anomaly-treated series derived in the methodology.

Appendix F. Model Development

Appendix F.1. Forecasting Technique

Numerous ANN multi-step forecasting techniques exist in the literature including;

- Direct Multi-step Forecasting
- Direct-Recursive Hybrid Multi-step Forecasting
- Multiple Output Forecasting
- Recursive Multi-step Forecasting

Direct multi-step forecasting requires the development of separate modes for each time-step forecasted. E.g.

$$prediction(t+1) = model1(obs(t), obs(t-1), ..., obs(t-n)$$

$$prediction(t+2) = model2(obs(t), obs(t-1), ..., obs(t-n)$$

However, as a separate model is used for each time-step there is no opportunity to model the dependencies between predictions which is often the case in time series. Direct-recursive hybrid forecasting instead uses the prediction from each step as an input into the next model which enables dependency between predictions to be established. Again, this requires separate trained models for each time step which is highly computationally expensive, especially for 24 different predicted values. Therefore, this method was also not used. The multiple output strategy uses a single model with past data as an input to predict all future values at once. E.g.

$$prediction(t+1), prediction(t+2) = model(obs(t), obs(t-1), ..., obs(t-n)$$

This method shows effective results, however it does not allow efficient use of the pre-processing technique *fragmenting* - a key method in recent literature to be tested in this study. A fragment is a reduced set of past data which is highly relevant to a single data point in the future, not a set of multiple future data points. For fragmenting to be effective, it may only be used to predict a single data point ahead. This single prediction can then be incorporated into a new fragment and fed back into the original model successively to form a new prediction.

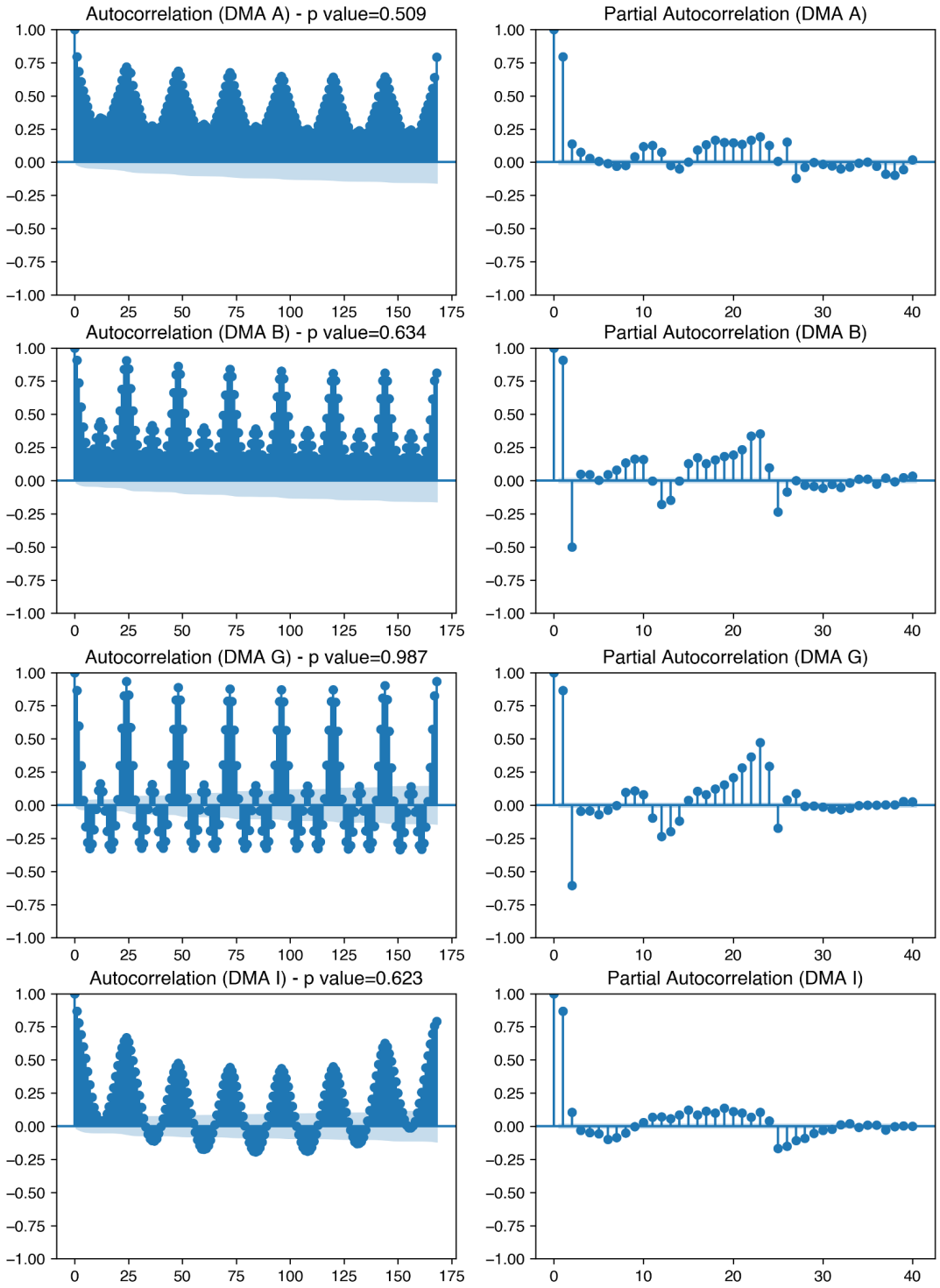

Figure F.21: Autocorrelation and Partial Autocorrelation plots for each DMA using original demand data with imputation and anomaly removal applied. The augmented Dicky-Fuller p-values are included in plot titles.

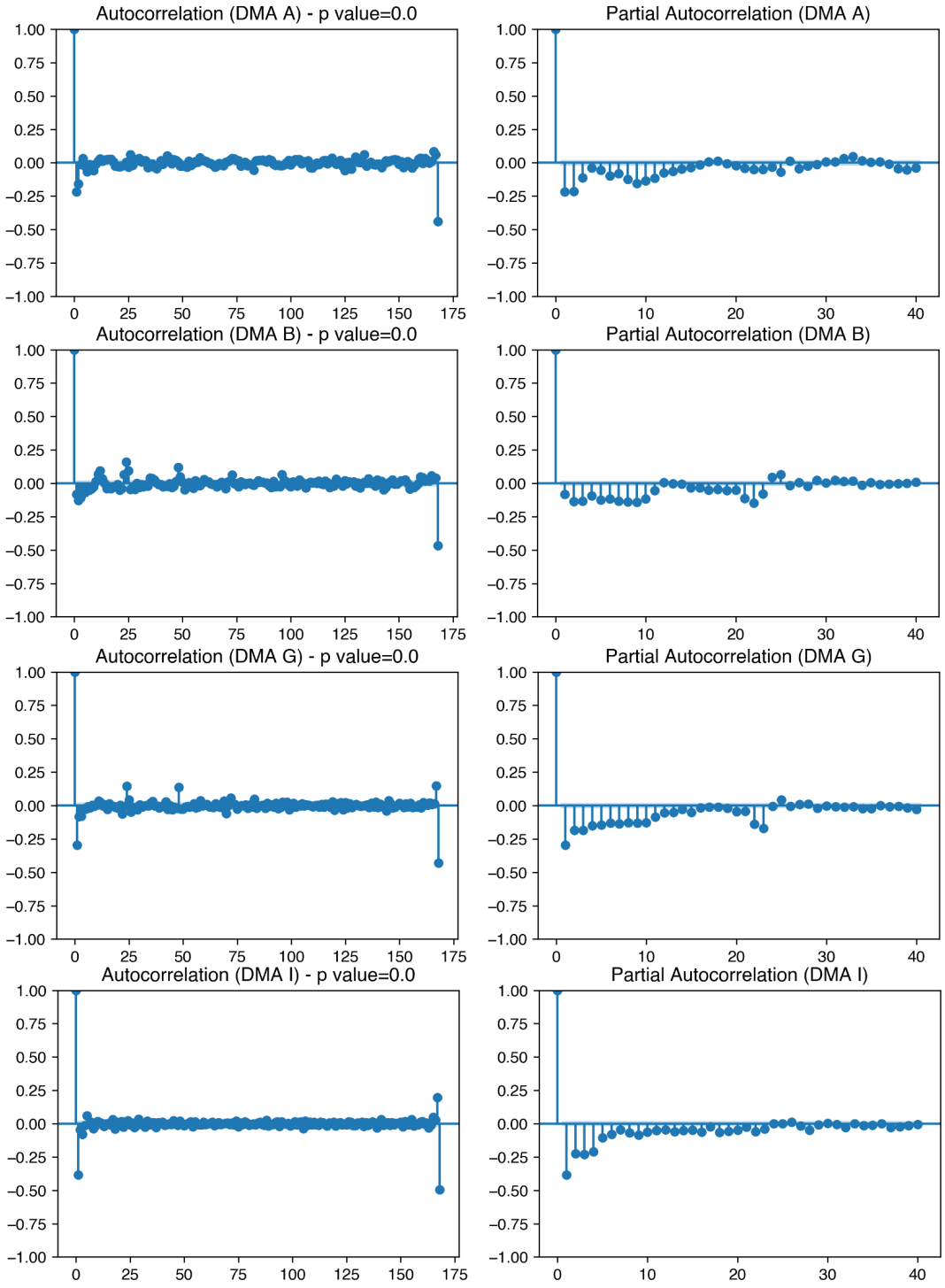

Figure F.22: Autocorrelation and Partial Autocorrelation plots for each DMA using a differeced set of data values with augmented Dicky-Fuller p-values included in plot titles.