Journal of Hydroinformatics

Journal of Hydroinformatics Vol 25 No 3, 895 doi: 10.2166/hydro.2023.163

Short-term water demand forecasting using data-centric machine learning approaches

Guoxuan Liu [a], Dragan Savic [a,b,c] and Guangtao Fu [a,*]

[a] Centre for Water Systems, University of Exeter, Exeter, EX4 4QF, United Kingdom
[b] KWR Water Research Institute, Nieuwegein, 3430 BB, The Netherlands
[c] Faculty of Civil Engineering, University of Belgrade, Bul. Kralja Aleksandra 73, 11120 Belgrade, Serbia
*Corresponding author. E-mail: g.fu@exeter.ac.uk

GL, 0000-0001-9837-8252; DS, 0000-0001-9567-9041

ABSTRACT

Accurate water demand forecasting is the key to urban water management and can alleviate system pressure brought by urbanisation, water scarcity and climate change. However, existing research on water demand forecasting using machine learning is focused on model-centric approaches, where various forecasting models are tested to improve accuracy. The study undertakes a data-centric machine learning approach by analysing the impact of training data length, temporal resolution and data uncertainty on forecasting model results. The models evaluated are Autoregressive (AR) Integrated Moving Average (ARIMA), Neural Network (NN), Random Forest (RF) and Prophet. The first two are commonly used forecasting models. RF has shown similar forecast accuracy to NN but has received less attention. Prophet is a new model that has not been applied to short-term water demand forecasting, though it has had successful applications in various fields. The results obtained from four case studies show that (1) data-centric machine learning approaches offer promise for improving forecast accuracy of short-term water demands; (2) accurate forecasts are possible with short training data; (3) RF and NN models are superior at forecasting high-temporal resolution data; and (4) data quality improvement can achieve a level of accuracy increase comparable to model-centric machine learning approaches.

Key words: autoregressive integrated moving average, data-centric machine learning, neural network, prophet, random forecast, short-term water demand forecasting

HIGHLIGHTS

- Data-centric machine learning approaches offer promise for improving the forecast accuracy of short-term water demands.
- Accurate forecasts are possible with short training data.
- Random forest and neural network models are superior at forecasting high-temporal resolution data.
- Data quality improvements can achieve a level of accuracy increase comparable to model-centric machine learning approaches.

INTRODUCTION

Water demand management is essential for ensuring water security in urban centres, which are increasingly coming under threat due to urbanisation, water scarcity and climate change. An effective way to mitigate the increasing threat is to make accurate demand and consumption forecasts, for short, medium and long forecasting horizons; these different horizons aid utilities with operation, financing and planning-related issues (Donkor *et al.* 2014). For operational management, short-term water demand forecasting is vital for pump operation and early leakage detection, which are key to improving service quality and minimising water loss. For demand forecasting to be successfully used for leakage detection, a high level of accuracy is necessary (Wu & Liu 2017).

With an aim to improve the accuracy of short-term water demand forecasting, much of the work has been using model-centric approaches (Lertpalangsunti *et al.* 1998; Herrera *et al.* 2010; Adamowski *et al.* 2012; Chen *et al.* 2017; Gagliardi *et al.* 2017; Chen & Boccelli 2018; Sardinha-Lourenço *et al.* 2018; Liu *et al.* 2022). These approaches focus on developing and adapting models to data, through various approaches including parameter optimisation, alterations to model structure and ensemble models. For machine learning models with well-defined structures, such as the Prophet (Taylor & Letham 2017)

and Autoregressive (AR) Integrated Moving Average (ARIMA), parameter optimisation has been the core focus when it comes to model-centric forecast accuracy improvement (Papacharalampous & Tyralis 2018; Weytjens *et al.* 2019; Menculini *et al.* 2021). While parameter optimisation is also employed in more complex models, such as neural network (NN) and random forest (RF), it is often not the focus of the investigation, as these models can be further improved in the model structure. Examples such as changes in the number of hidden layers for NN have shown to be effective at improving forecasting accuracy, but these improvements are gained at the cost of further computational complexity (Adamowski 2008; Ghiassi *et al.* 2008; Adamowski *et al.* 2012; Chen *et al.* 2017; Toharudin *et al.* 2020). Alternatively, ensemble modelling combines several forecasting models with either equal or different weights for individual models and this approach draws out the advantages of individual models, thus achieving higher accuracy compared to individual models (Lertpalangsunti *et al.* 1998; Grover *et al.* 2015; Bata *et al.* 2020).

In contrast to model-centric approaches, data-centric machine learning approaches have received limited attention in the field of time series forecasting, including short-term water demand forecasting (Fu *et al.* 2022). The idea of a data-centric approach has been popularised by Andrew Ng in recent years (DeepLearningAI 2021). The Data-Centric AI Competition (DeepLearning.AI & Landing AI 2021) followed not long after, where participants were asked to improve a dataset using data-centric techniques before it is fed to a fixed model and the level of accuracy improvement that can be achieved is evaluated. In the research communities, the terms data-centric and data-driven were generally used interchangeably, prior to Andrew Ng's definition in 2021. This is evident from many studies that explicitly mentioned the term 'data-centric', yet merely used data for forecasting purposes, the core research focuses still lies with the model or forecasting system (Faeldon *et al.* 2014; Grover *et al.* 2015; Böse *et al.* 2017). More recently, data-centric approaches were demonstrated more distinctly by Kang *et al.* (2021), whereby the research has forgone forecasting models. Instead, they use a large pool of real data from multiple sources as references. The similarity between the target series (series for forecasting) and the pool of reference series is measured and a subset of reference series that is most like the target series is chosen for forecasting the target series. However, a model-less approach is not the sole data-centric method. In this paper, all techniques that adapt data to models to improve forecast accuracy are considered data-centric approaches. The difference between data-driven and data-centric could then be noted as simply using data to develop a machine learning model in the former and optimising data to achieve the best model accuracy in the latter.

In this paper, data-centric machine learning approaches will be tested for short-term water demand forecasting to answer three questions: (1) How effective are model-centric approaches in improving forecast accuracy?; (2) Does forecast accuracy correlate with the training data length and data resolution?; and (3) How much data noise can models tolerate?

To answer the above questions, four forecast models are used – ARIMA, NN, RF and Prophet. ARIMA is commonly used as a benchmark model for comparative purposes (Adamowski *et al.* 2012; Tiwari & Adamowski 2013; Chen & Boccelli 2018; Guo *et al.* 2018; Sardinha-Lourenço *et al.* 2018). NN is a common model that has demonstrated high forecast accuracy (Maidment & Miaou 1986; Lertpalangsunti *et al.* 1998; Adamowski 2008; Ghiassi *et al.* 2008; Herrera *et al.* 2010; Adamowski *et al.* 2012; Tiwari & Adamowski 2013; Gagliardi *et al.* 2017; Guo *et al.* 2018). RF has received less attention compared to ARIMA and NN, but it has been shown to produce similar high forecast accuracy to NN (Herrera *et al.* 2010; Chen *et al.* 2017). Prophet is a relatively new forecasting model developed by Facebook (Taylor & Letham 2017) and it has yet to be applied to the field of short-term water demand forecasting. These models will be tested through a series of experiments to answer the three research questions. The demand data used are from four case studies with measured sub-daily water demand data. This study will help us understand how to optimally use data in the field of short-term water demand forecasting.

METHODS

This section starts with the source of data and then introduces the four forecasting models used: ARIMA, Prophet, NN and RF. Finally, performance indicators and experimental setups are explained.

Water demand data

The four case studies come from two different sources. Case Study 1 is collected hourly, while the other case studies are collected every 15 min. Some basic statistical information is shown in Table 1; this information includes the number of individuals or properties served in each case study, data stationarity and overall peaks and troughs. The high- and low- resolution data correspond to 15 min and hourly data. Case study 1 is collected on an hourly basis, thus, high-resolution information is not available (N/A); other case studies are collected every 15 min and the resolution is lowered by combining

Table 1 | Statistical information of water demand data used

		p-value		ADF value		Peak (m^3)		Trough (m^3)	
Case Study	Users (individuals/properties)	High res	Low res	High res	Low res	High res	Low res	High res	Low res
1	20,000 individuals	N/A	0	N/A	−5.8	N/A	110	N/A	9
2	351 properties	0	0	−24.9	−4.9	9.6	34.1	1.1	4.6
3	461 properties	0	0	−28.1	−6.9	12.9	49.2	1.6	6.8
4	669 properties	0	0	−28.7	−4.9	12.4	47.7	1.2	5.2

every four data points into an hourly demand. Data stationarity testing can confirm the presence of unit roots in data (Gupta *et al.* 2009) and this can be used to determine data trajectory; stationary data means the data have a consistent moving average (MA). Data stationarity can be confirmed if it has a *p*-value close to 0 and a large negative Augmented Dickey-Fuller (ADF) value. As the results show, all datasets are stationary across different temporal resolutions.

The raw datasets have several holiday periods, as well as single or continuous recording errors, these periods are all excluded from the datasets used for analysis. As a result, the datasets used in this research are without any known special event, anomaly or holiday effects.

Figure 1 shows an overview and basic analysis of Case Study 1. The MA at 24 (daily) and 168 points (weekly) can be seen in Figure 1(a). This confirms the daily and weekly seasonality within the demand data. While there is strong daily and weekly seasonality within the data, Figure 1(c) shows the demand is still highly varied for most of the day. Therefore, forecasting models are required to make detailed predictions, as highly accurate forecasts would benefit operational management.

Each dataset is split on a 60/40 basis for training and testing. As a result, 1,008 data points from Case Study 1 are used for training and 672 for testing. The other three case studies have a higher temporal resolution but have similar daily and weekly seasonality. Compared to 10 weeks of quality data used in Case Study 1, 30 weeks of data are retained for other case studies, with the higher data resolution and 12,096 and 8,064 data points are used for training and testing, respectively.

Based on the data overview, the number of features used for RF, the number of input neurons for NN and the seasonality factor for ARIMA are the number of measured demand points in a day. While there exists weekly periodicity, increasing the number of features for RF and input neurons for NN to weekly measured points would increase exponentially the forecast model's complexity.

ARIMA

ARIMA is a statistical model for time series, developed by Box and Jenkins in 1970 (Box George *et al.* 2015). ARIMA combines AR and MA models with a built-in differencing term.

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

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

where y_t is the information state at different time t, ε is error, ϕ and θ are AR and MA parameters and p and q are the total number of AR and MA terms. Equation (1) can be simplified as

$$\phi_p(B^p) y_t = \theta_q(B^q) \varepsilon_t \quad (2)$$

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

ARMA models only work on stationary data, where stationarity is defined by data having a constant mean and variance. Non-stationary data can become stationary by differencing data points. This differencing function can be integrated into the

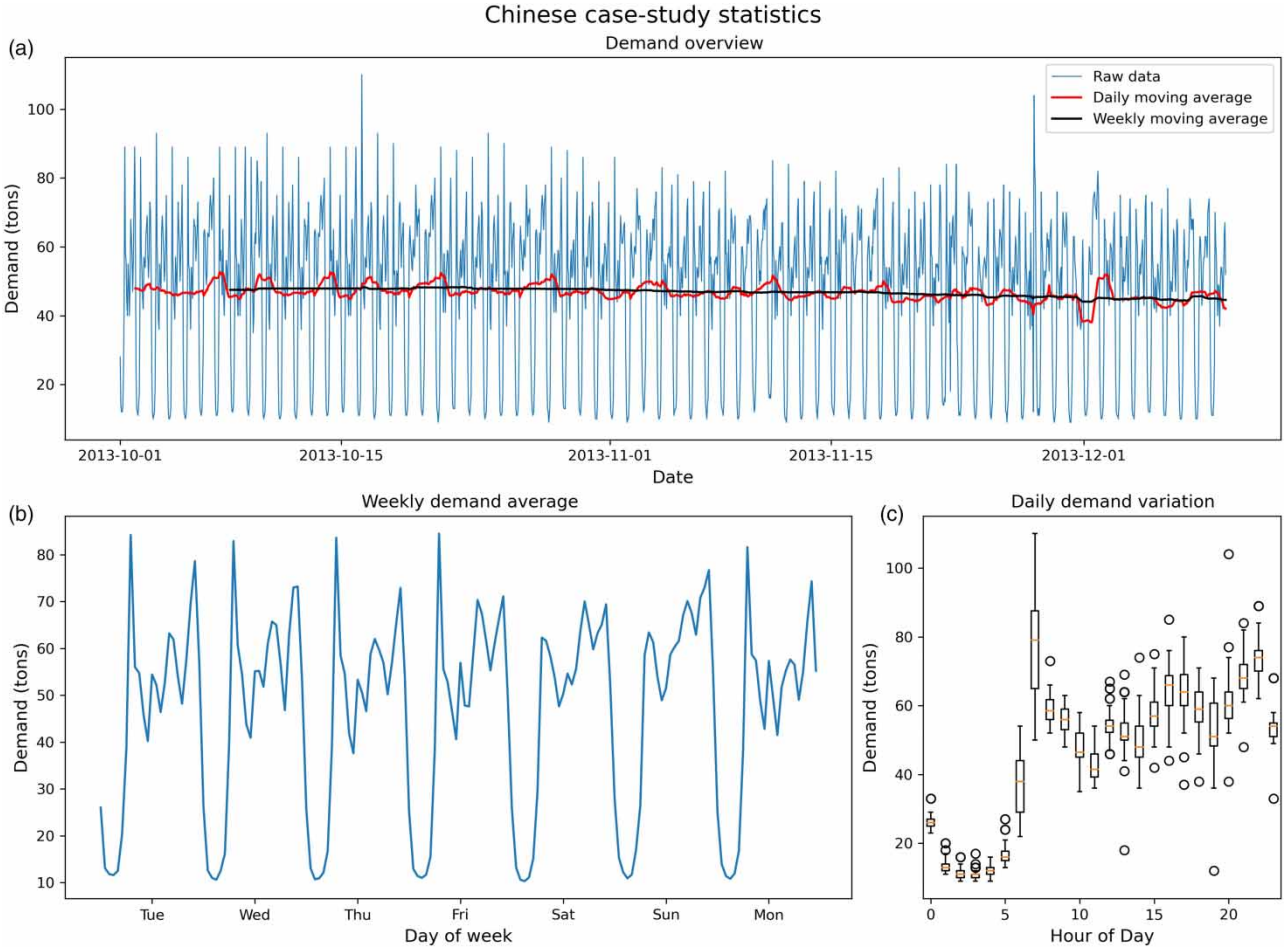

Figure 1 | Raw data plots of Case Study 1, (a) full demand data, with MA of 24 and 168 h; (b) averaged weekly demand data; and (c) daily demand data with median and percentiles (25 and 75%).

ARMA model and the result is known as an ARIMA model, as expressed in the following equation:

$$\phi_p(B^p)\Delta^d y_t = \theta_q(B^q)\varepsilon_t \tag{3}$$

where Δ is the differencing factor and d is the degree of difference.

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

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

$$\Phi_P(B^{sP})\phi_p(B^p)\Delta_s^D\Delta^d y_t = \Theta_Q(B^{sQ})\theta_q(B^q)\varepsilon_t \tag{4}$$

where s is the period of a known seasonality and P, D and Q, respectively, represent the AR, differentiation and MA terms of the seasonality, like their respective lower-case counterparts. The SARIMA model is presented in the form of SARIMA(p,d,q)(P,D,Q)$_s$.

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

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

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

Prophet

Prophet is a modular regression model for time series forecasting, developed by the Facebook research team (Taylor & Letham 2017). In the original paper, the number of events created on Facebook is used as source data to compare the forecast performance of Prophet, ARIMA, Exponential Smoothing and Random Walk. The results have shown that Prophet is superior to all other models tested.

Prophet works by decomposing any given time series into three main components, including trend, seasonality and holiday effects, as expressed in the following equation:

$$y_t = G_t + S_t + H_t + \varepsilon_t \tag{5}$$

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

The trend can be modelled as a linear function (Equation (6)) or a non-linear saturating growth (Equation (7)):

$$G_t = (k + a_t\delta)(t + (m + a_t\gamma)) \tag{6}$$

$$G_t = \frac{C_t}{1 + e^{-(k+a_t\delta)(t-(m+a_t\gamma))}} \tag{7}$$

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

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

$$S_t = \sum_{n+1}^{N} \left(a_n \cos\frac{2\pi nt}{P} + b_n \sin\frac{2\pi nt}{P} \right) \tag{8}$$

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

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

In terms of input parameters, Prophet can make forecasts without any parameter input. However, the key parameter – the number of change points and its scale – will be tested through grid search using data from Case Study 1 to determine its impact and the optimal pairing. The Prophet package is available in both R and Python, and the Python Prophet package is used in this research.

Neural network

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

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

$$h_j = f\left(\sum_{i=0}^{n_0} w_{ij}x_i + w_{0j}\right) \tag{9}$$

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

The NN model used in this research is a three-layer feed-forward model with back propagation. The model is applied using 'MLPRegressor' from the 'sklearn' library in Python. Available numerical parameters are all investigated using Case Study 1 to determine optimal parameter settings for further experiments.

Random forest

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

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

$$E = \sum_{y\in R_1} (y - \hat{y}_{R_1})^2 + \sum_{y\in R_2} (y - \hat{y}_{R_2})^2 \tag{10}$$

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

The RF model used in this research is the 'RandomForestRegressor' from the 'sklearn' library within Python. Available numerical parameters are first investigated using Case Study 1 to determine optimal parameter settings for further experiments.

Performance indicators

There are many measures to determine the performance of forecasting models. These include mean absolute percentage error (MAPE) (Bakker *et al.* 2013; Chen *et al.* 2017; Taylor & Letham 2017; Sardinha-Lourenço *et al.* 2018), root mean squared error (RMSE) (Chen & Boccelli 2018) and coefficient of determination (R^2) (Adamowski 2008; Bakker *et al.* 2013; Chen & Boccelli 2018).

This study has chosen to use RMSE and R^2 to measure model accuracy. RMSE focuses on measured and forecast demand error residual and R^2 focuses on measured and forecast demand correlation. While there are several different expressions of R^2, a detailed analysis of all expressions (Kvalseth 1985) recommends the use of an R^2 indicator in Equation (12). The expressions for RMSE and R^2 used in this research are shown in the following equations:

$$\text{RMSE} = \sqrt{\sum \frac{(y_o - y_f)^2}{n}} \tag{11}$$

$$R^2 = 1 - \sum \frac{(y_o - y_f)^2}{(y_o - \overline{y_o})^2} \tag{12}$$

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

While both RMSE and MAPE measure residual/error, RMSE is chosen because it is not affected by the size of the measured values. In comparison, MAPE gives high-measured values a greater degree of error leniency over low-measured values. Since short-term water demand data experience clear seasonality and knowing daily peak and trough demand are equally important, the high-value bias in MAPE is unnecessary. The RMSE value depends on the scale of data used, thus, there is no upper limit, but it does have a lower limit of 0.

The choice R^2 is a dimensionless measure, with an upper limit of 1. While the R^2 selected has no lower limit, the intuitive understanding is that any negative R^2 values would indicate that a model is worse than the observed mean, which renders models with negative R^2 statistically insignificant. Additionally, the selected R^2 is identical to the Nash–Sutcliffe Efficient Coefficient (NSE) (Gagliardi *et al.* 2017; Tyralis & Papacharalampous 2018), which is commonly used in water-related research.

Experiment set up

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

The first experiment aims to establish the effect of basic model-centric approaches on forecast accuracy, which addresses the first research question. This is done by evaluating the model parameters available for tuning. Prophet and SARIMA have a limited number of parameters available for tuning. Prophet has four parameters, but three overlap with each other; thus, only two parameters warrant investigation. SARIMA has seven parameters, the seasonality, differencing and seasonal differencing factors are fixed and the seasonal AR and MA factors are 0 as all case studies have a near-constant MA, thus only two parameters warrant investigations. In comparison, NN and RF have multiple tuneable numerical parameters. Therefore, all numerical parameters for NN and RF are first tested individually, using Case Study 1; from these results, two crucial parameters are selected for having the most significant effect on forecast accuracy. As this paper focuses on data-centric approaches, only two parameters are selected for each model for sampling analysis to demonstrate the effect of the basic model-centric machine learning approach.

The two selected parameters for NN and RF are carried forward and further investigated through sampling, along with two parameters each from Prophet and SARIMA, using Case Studies 1 and 2. For each test in Experiment 1, 60% of data are used for training and 40% for testing. The forecast horizon is set to 1 day (24 points for Case Study 1 and 96 for Case Study 2). After each forecast, a moving window is employed to move along the training and forecasting data forward by a day, to ensure the training data used is always 60% of the full data length prior to the point of the forecast. The model is then retrained and a new forecast is made; this retraining process is repeated for each forecast horizon until 40% of data is forecasted. The accuracy is then recorded between the 40% of data and its corresponding forecast.

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

For Experiments 2 and 3, the setup used is the same as for Experiment 1. The total data length used is 10 weeks for Case Study 1 and 30 weeks for others; the forecast horizon is 1 day; the testing period is 40% of the total data; and lastly, a moving window is used to maintain a consistent training length for each forecast.

Experiment 2 will investigate the effect of lengthening the training data length, starting at 2 days, with an increment of 1 day each time, up to 28 days of training data. This will determine if forecast accuracy correlates with training data length or if less can be beneficial. Experiment 2 will be applied to all case studies.

Experiment 3 will investigate the effect of data resolution. This experiment is only applied to Case Studies 2, 3 and 4, as Case Study 1 involves lower-resolution data. For the case studies tested, the data were aggregated using 30, 60 and 120 min-long steps by taking the MA of the raw data. Forecasts are made for each new data set and the average forecast accuracy is compared with the original data.

The final experiment aims to determine how well each model tolerates noise; this will address the third research question. A scaled Gaussian noise is added to the training data; the scale is set to be between 0 and 50% of the average data value. The number of forecasts made is significantly higher for this experiment because: (1) each model is repeated for each noise level and (2) each noise level is repeated ten times, due to the random nature of the added noise. To reduce the computation time, the forecast horizon is increased to 7 days and the noise scale increment is 5% for the case studies.

RESULTS AND DISCUSSION

Parameter analysis

Using Case Study 1, the numerical parameters for RF and NN are considered for evaluation. Based on this result, the two parameters with the most significant effect on accuracy will be selected for detailed sampling analysis.

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

Figure 2 shows the effect different numerical parameters have on forecast accuracy for NN and RF. In all figures, the *y*-axis shows the forecast accuracy and the *x*-axis shows the selected parameter orders (a detailed selection of parameter scale and values is provided in Appendix A).

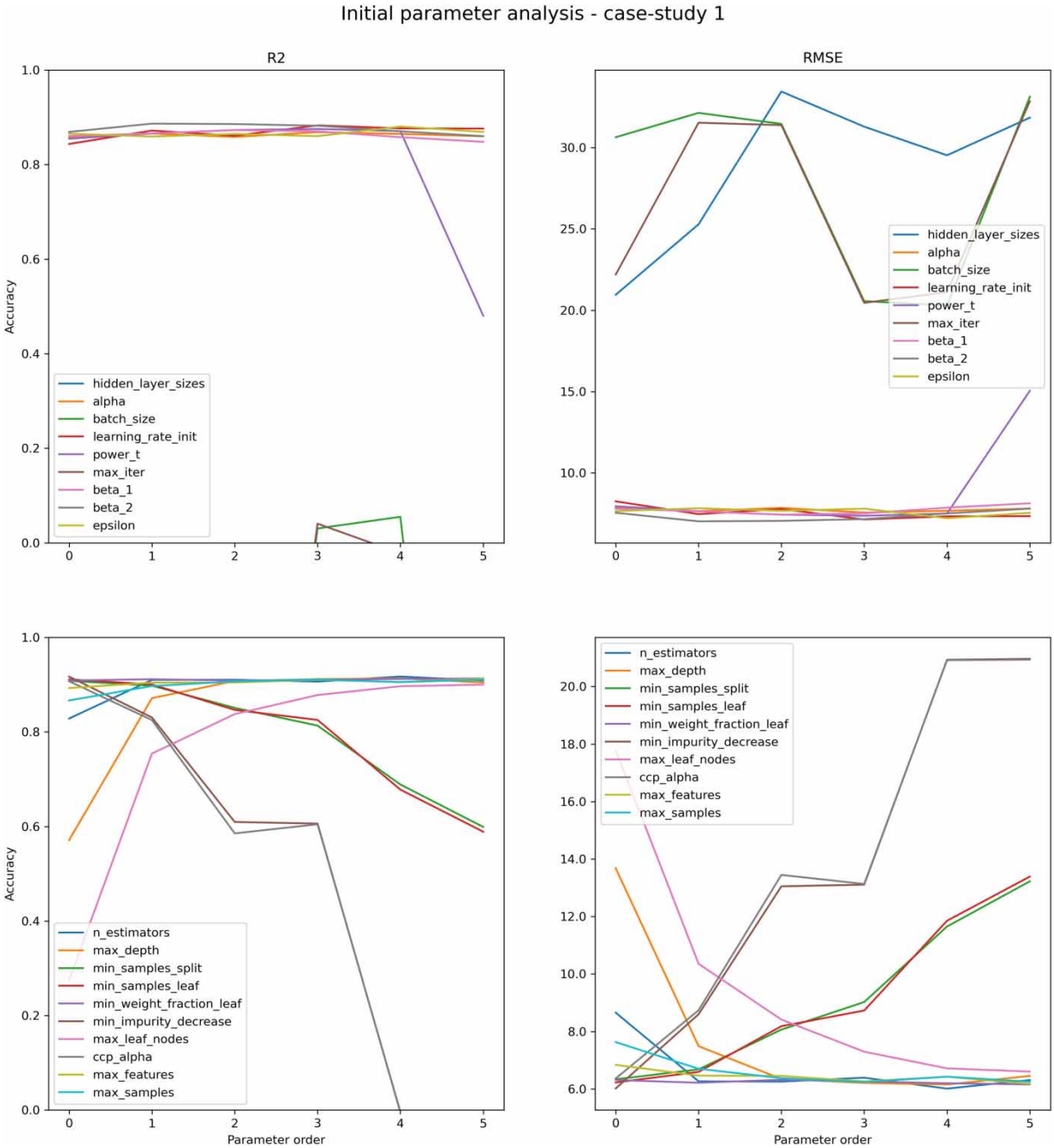

Figure 2 | Initial parameter analysis of all available numerical parameters for NN and RF.

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

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

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

Table 2 shows the selected sampling parameters and the chosen investigation range for the two case studies (as Case Studies 2, 3 and 4 are similar in resolution and data length, investigation on parameter choice is only performed on Case Study 2).

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

Figure 3 shows the accuracy results for parameter sampling for Case Studies 1 and 2. The heading above each figure in the top panels indicates the model used per column and the left headings for each figure in the left panels indicate the case study and accuracy measure per row. The axis headings and values for each panel show the feature and value sampled, as detailed in Table 2. The shade corresponds with accuracy levels, where lighter shades correspond with higher accuracy and vice versa. The text in each panel shows the accuracy values, rounded to two decimal places. Each panel also has three highlighted (bold) values; these correspond with the three most accurate forecasts within each panel.

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

The panels in the first and second columns show that for short-term water demand data, Prophet and ARIMA produce consistent forecasts, independent of parameter pairings; only extreme parameter choices have a slightly negative impact on accuracy for these models. RF too is not overly dependent on parameter choice, though a feature count equal to half of the features available and a small minimum sample split tend to produce slightly better results. The parameters in NN do

Table 2 | Parameter choice and analysed values for different models

Model	Parameter	Case Study 1	Case Study 2
Prophet	Number of change points	1,2,3,6,42	1,2,3,18,126
	Change point prior scale	0.0005,0.005,0.05,0.5,5	0.0005,0.005,0.05,0.5,5
ARIMA	p	0, 1, 2, 3, 4	0, 1, 2, 3, 4
	q	0, 1, 2, 3, 4	0, 1, 2, 3, 4
RF	Maximum feature	1,6,12,18,24	1,24,48,72,96
	Minimum sample split	2,4,6,8,10	4,13,22,31,40
NN	Hidden layer node count	12, 24, 48, 96, 192	24, 48, 96, 192, 384
	Maximum iteration	50, 100, 200, 400, 800	50, 100, 200, 400, 800

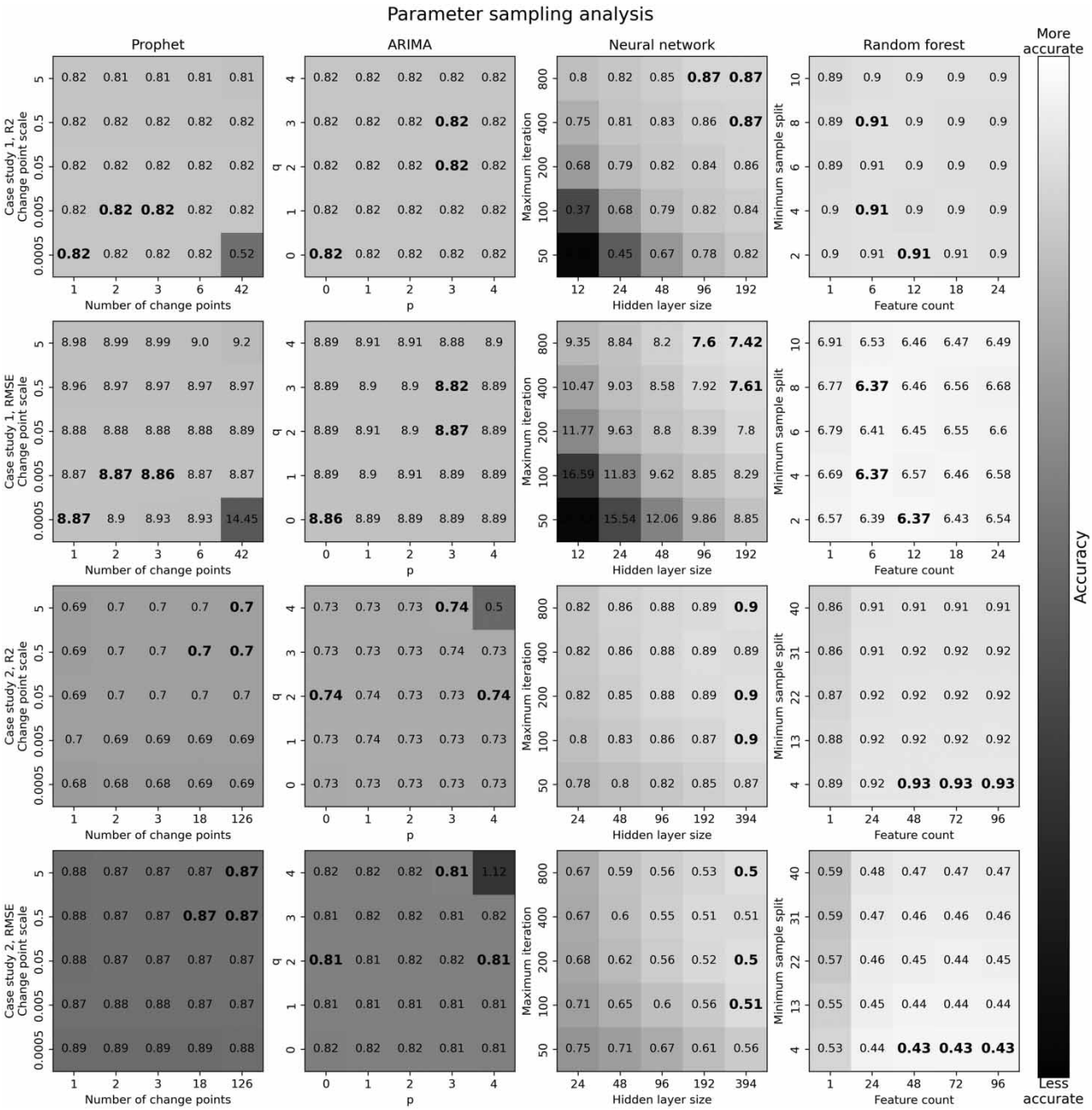

Figure 3 | Sampling analysis for Case Studies 1 and 2.

play a more significant role, where the higher computational complexity does result in a more accurate result, but the accuracy plateaus when the maximum iteration is 800 and the hidden layer size is more than double the number of features.

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

Training data length analysis

While there is sufficient quality data available in all case studies analysed, this may not be the case for real-life forecasting cases. Experiment 2 aims to establish a baseline for data required for each model to make a sufficiently accurate forecast.

The parameter values used were selected from Experiment 1. The effect of training data length can be visualised by varying the amount of training data used, starting at 2 days with an increment of 1 day each time, up until 28 days. This is applied to all case studies.

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

Figure 4 shows how the forecast accuracy reacts to reduced training data length for all case studies. The x-axis represents the length of data used for training, measured in days, and the y-axis represents the forecast accuracy (left panels for R^2 and

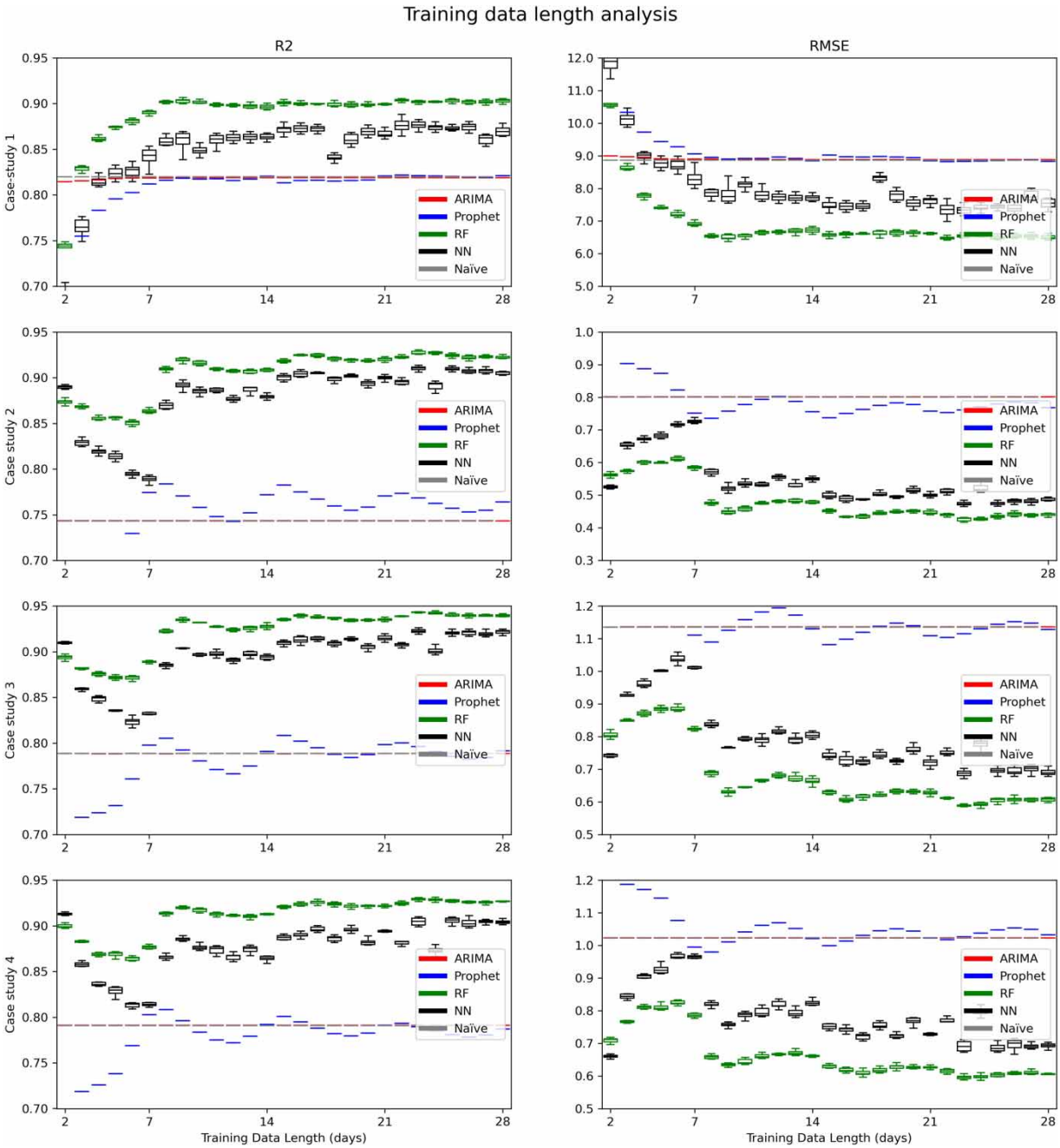

Figure 4 | Training data length analysis.

right panels for RMSE). The grey line in all figures corresponds to the accuracy level achieved by the Naïve method, where all demands in the forecast period are assumed to be equal to the demand from the same time on the previous day. It needs to be noted that ARIMA has similar forecast accuracy to the Naïve method in most cases; thus, the lines overlap once ARIMA plateaus.

Like Experiment 1, the R^2 and RMSE negatively correlate with each other, suggesting that forecasts with low accuracy are underperforming in both correlation and bias. Because of this correlation, all accuracy discussions that follow will not distinguish between R^2 and RMSE, unless specific accuracy values require discussion.

From Figure 4, the results from the top panels for Case Study 1 show that all models approach their optimal accuracy level with 10 days of training data, with only NN showing a significant further improvement, both in terms of accuracy and model stability (variance decrease). Prophet and ARIMA produce similar forecast accuracy when plateaus are reached.

Panels from rows 2, 3 and 4 show that there is a periodicity in how result accuracy changes with increased training length in all models for Case Studies 2, 3 and 4. The period identified is 7 days and the first peak appears on days 8 or 9, depending on the forecasting model.

RF and NN have reached their first local peak on day 9, then each subsequent local peak is reached 7 days after the previous peak. The improved results for 2 days beyond n whole weeks could be explained by the importance of weekend information. As Saturday and Sunday have slightly different demand patterns, two additional days of training data can improve the weekend forecast, especially when the training data are short. The accuracy oscillation effect diminishes with longer training data. Overall, RF has shown to be more accurate and stable compared to NN and it has reached a stable peak at 9 days compared to 25 days for NN.

In contrast, Prophet has reached global optimum at first accuracy peak at 8 days. The accuracy then oscillates around the Naïve method level, peaking every 7 days after day 8; however, the average accuracy slowly decreases with increased training data length. This effect can be explained by reviewing the Prophet model's structure. Due to Prophet's additive nature, the seasonal trends remain consistent. The overall trend change in the testing period follows the trend change frequency detected in the training period. Since short-term water demand does not experience significant overall trend change, prolonged training data would introduce unnecessary change points and could cause over fitting in the testing data. Additionally, shorter training data means that the training data more closely relates to the testing data in the temporal space. The first local peak in forecast accuracy for Prophet should be taken as the global peak.

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

The oscillation effect in Prophet, NN and RF suggests that more training data may have a negative effect on forecast accuracy. This is especially true when training data is limited; knowing the right amount of data to train models based on model character and data periodicity is more important in improving forecast accuracy. This analysis suggests that weekly or less frequent data features have a minimal impact on model forecast accuracy, models that consider these features, such as ARIMA and Prophet, pose no advantage to models that do not.

Temporal resolution analysis

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

Figure 5 shows how the forecasting models react to reduced data resolutions for Case Studies 2, 3 and 4. The R^2 measure is unit free, but the RMSE does have a unit, which correlates with the size of the measured demands. Since the lower-resolution datasets are generated by aggregating high-resolution data, the RMSE values at different resolutions cannot be directly compared. As a result, each RMSE result shown in the right panels of Figure 4 is divided by the number of aggregating points, namely 1, 2, 4 and 8, for 15, 30, 60 and 120 min sample rates, respectively.

Like previous experiments, the R^2 and RMSE results negatively correlate with each other, thus, the R^2 and RMSE results in Figure 5 will be jointly discussed. The results all show that the forecast accuracy increased with decreasing data resolution for

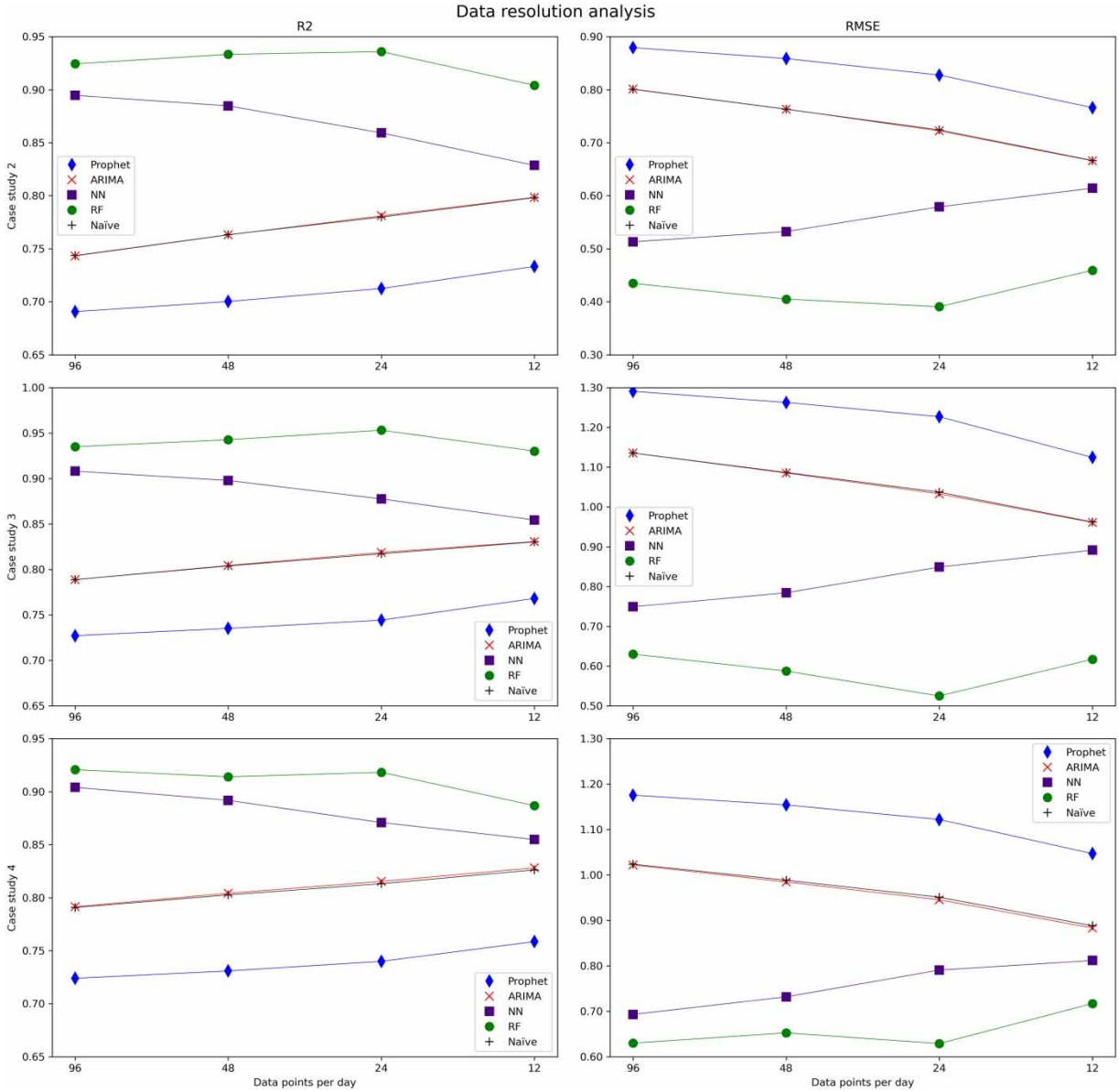

Figure 5 | Data resolution analysis.

Prophet, ARIMA and the Naïve method forecasting (accuracy overlaps with ARIMA results), and the opposite is true for RF and NN.

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

It is worth noting that while the forecast accuracy improved for Prophet and ARIMA when the resolution was decreased, it was at best on par with the Naïve method, still far worse than RF and NN. This analysis indicates that the holistic models (ARIMA and Prophet) are inferior for short-term water demand forecasting. Combining this with the results from the previous experiment, it can be generalised that for sub-daily water demand forecasting, daily data feature plays a key role and features

that are weekly or less frequent have minimal impact. However, the impact of less frequent seasonal factors increases with the decrease in data temporal resolution.

Data uncertainty analysis

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

Figure 6 shows the impact of noisy data on all case studies. The left panel shows R^2 accuracy, while the right panel shows RMSE accuracy. As it has been with R^2 and RMSE comparisons in previous experiments, the R^2 and RMSE accuracy

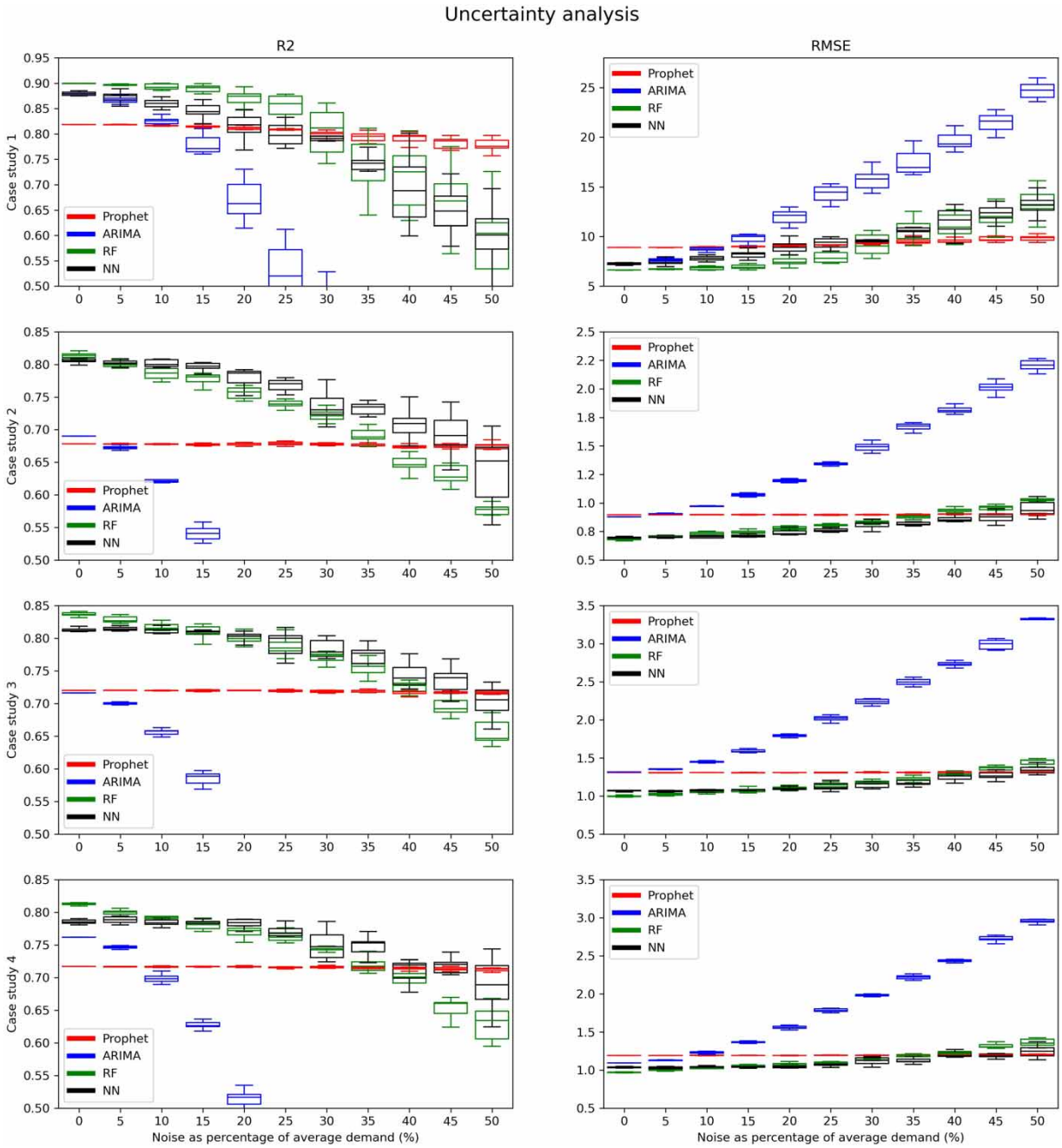

Figure 6 | Uncertainty analysis.

negatively correlate with each other, suggesting that better forecasting results are superior in both correlation and residuals. Therefore, subsequent discussions of accuracy will be done in terms of more and less accuracy when comparing methods or case studies.

The uncertainty results show that the impact of data noise differs greatly between models, with the most significantly affected being ARIMA and the least being Prophet. All figures show that Prophet can maintain the same level of accuracy regardless of noise, albeit the accuracy variance slightly increases towards a higher noise level. While Prophet has inferior accuracy using training data without noise, its robustness allows it to eventually outperform all other models.

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

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

CONCLUSIONS

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

The results show that all models can make highly accurate forecasts, both in terms of R^2 and RMSE. While all models have proven their ability in their application in the field of short-term water demand forecasting, the performance of different models varies with the same data set, with RF consistently producing forecasts with the highest R^2 and lowest RMSE. This implies that appropriate model choice is an important first step in ensuring accurate forecasts.

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

The training data length analysis has shown that more data does not necessarily provide better forecasts. This is especially true when using Prophet to forecast short-term water demands. The accuracy oscillations in Prophet, RF and NN suggest that when using shorter training data, high accuracy can still be achieved when using the right amount of training data. This study found that when using small training datasets, the optimal training data length is 1 day more than n whole weeks for Prophet and 2 days more than n whole weeks for RF and NN. Prophet performs better with shorter training data, for cases where data have little long-term value shift, such as short-term water demands.

When considering data temporal resolution and forecasting model pairing, RF and NN are better for high-resolution data, while ARIMA and Prophet are better for low-resolution. Due to the data used in this research, the resolution effect is present but varies for different models. This implies the significance of analysing data temporal resolution in the development of machine learning. This finding needs to be further confirmed by doing similar tests on short- to medium-term water demand predictions.

The findings from training data length analysis and data temporal resolution analysis show that daily data features play the most significant role in short-term water demand forecasting, while data features that are present with a frequency of weekly or less have a minimal impact. Therefore, models such as Prophet and ARIMA that consider longer-term seasonal factors have no advantage over other models and RF and NN that focus on near-term demand are more suitable.

Lastly, data quality is shown to have a significant impact on forecast accuracy in most models. Although Prophet has shown that it is immune to data noise, it has produced lower accuracy forecasts compared to other models with uncorrupted data. RF and NN data uncertainty testing has shown that 10% data quality improvement can improve R^2 by 0.05. This shows that high-accuracy forecasting models are sensitive to data quality and data quality improvements can offer similar accuracy improvement to that of complex model-centric approaches.

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

ACKNOWLEDGEMENTS

This work was supported by the Royal Society under the Industry fellowship scheme (Ref: IF160108) and the Alan Turing Institute under the EPSRC grant (Ref: EP/N510129/1). The first author was supported by the Doctoral Training Grant (DTG) from the UK Engineering and Physical Sciences Research Council (Ref: 620036449).

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

Adamowski, J. F. 2008 Peak daily water demand forecast modeling using artificial neural networks. *J. Water Resour. Plan. Manag.* **134** (2), 119–128. https://doi.org/10.1061/ASCE0733-94962008134:2119.

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

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

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

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

Box George, E. P., Jenkins Gwilym, M., Reinsel Gregory, C. & Ljung Greta, M. 2015 *Time Series Analysis: Forecasting and Control*. Wiley Blackwell, Hoboken, NJ.

Breiman, L. 2001 Random forests. *Machine Learning* **45**, 5–32. https://doi.org/10.1023/A:1010933404324.

Chen, J. & Boccelli, D. L. 2018 Forecasting hourly water demands with seasonal autoregressive models for real-time application. *Water Resour. Res.* **54** (2), 879–894. Blackwell Publishing Ltd. https://doi.org/10.1002/2017WR022007.

Chen, G., Long, T., Xiong, J. & Bai, Y. 2017 Multiple random forests modelling for urban water consumption forecasting. *Water Resour. Manag.* **31** (15), 4715–4729. Springer Netherlands. https://doi.org/10.1007/s11269-017-1774-7.

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

DeepLearning.AI and Landing AI 2021 *Data-Centric AI Competition*.

Donkor, E. A., Asce, S. M., Mazzuchi, T. A., Soyer, R. & Roberson, J. A. 2014 Urban water demand forecasting: review of methods and models. *J. Water Resour. Plan. Manag.* **140** (2), 146–159. https://doi.org/10.1061/(ASCE)WR.1943-5452.

Faeldon, J., España, K., Jay, D. & Ix, S. 2014 *Data-Centric HPC for Numerical Weather Forecasting*. https://doi.org/10.1109/ICPP.Workshops.2014.23.

Fu, G., Jin, Y., Sun, S., Yuan, Z. & Butler, D. 2022 The role of deep learning in urban water management: a critical review. *Water Res.* **223**, 118973.

Gagliardi, F., Alvisi, S., Kapelan, Z. & Franchini, M. 2017 A probabilistic short-termwater demand forecasting model based on the Markov chain. *Water (Switzerland)* **9** (7). MDPI AG. https://doi.org/10.3390/w9070507.

Ghiassi, M., Zimbra, D. K. & Saidane, H. 2008 Urban water demand forecasting with a dynamic artificial neural network model. *J. Water Resour. Plan. Manag.* **134** (2), 138–146. https://doi.org/10.1061/ASCE0733-94962008134:2138.

Grover, A., Kapoor, A. & Horvitz, E. 2015 A deep hybrid model for weather forecasting. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp. 379–386.

Guo, G., Liu, S., Wu, Y., Li, J., Zhou, R. & Zhu, X. 2018 Short-term water demand forecast based on deep learning method. *J. Water Resour. Plan. Manag.* **144** (12), 04018076. American Society of Civil Engineers (ASCE). https://doi.org/10.1061/(asce)wr.1943-5452.0000992.

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

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

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

Kvalseth, T. O. 1985 *Cautionary Note About R2*.

Lertpalangsunti, N., Chan, C. W., Mason, R. & Tontiwachwuthikul, P. 1998 A toolset for construction of hybrid intelligent forecasting systems: application for water demand prediction. *Artificial Intelligence in Engineering* **13** (1), 21–42. https://doi.org/10.1016/S0954-1810(98)00008-9.

Liu, X., Zhang, Y. & Zhang, Q. 2022 Comparison of EEMD-ARIMA, EEMD-BP and EEMD-SVM algorithms for predicting the hourly urban water consumption. *J. Hydroinform*. https://doi.org/10.2166/hydro.2022.146.

Maidment, D. R. & Miaou, S.-P. 1986 Daily water use in nine cities. *Water Resour. Res.* **22** (6), 845–851.

Menculini, L., Marini, A., Proietti, M., Garinei, A., Bozza, A., Moretti, C. & Marconi, M. 2021 Comparing prophet and deep Learning to ARIMA in forecasting wholesale food prices. arXiv:2107.12770v3.

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

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

Taylor, S. J. & Letham, B. 2017 Forecasting at scale. https://doi.org/10.7287/peerj.preprints.3190v2.

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

Toharudin, T., Pontoh, R. S., Caraka, R. E., Zahroh, S., Lee, Y. & Chen, R. C. 2020 Employing long short-term memory and Facebook prophet model in air temperature forecasting. In: *Communications in Statistics – Simulation and Computation*. Bellwether Publishing, Ltd. https://doi.org/10.1080/03610918.2020.1854302.

Tyralis, H. & Papacharalampous, G. A. 2018 Large-scale assessment of Prophet for multi-step ahead forecasting of monthly streamflow. *Adv. Geosci.* **45**, 147–153. Copernicus GmbH. https://doi.org/10.5194/adgeo-45-147-2018.

Weytjens, H., Lohmann, E. & Kleinsteuber, M. 2019 Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. In: *Electronic Commerce Research*. Springer New York LLC. https://doi.org/10.1007/s10660-019-09362-7.

Wu, Y. & Liu, S. 2017 A review of data-driven approaches for burst detection in water distribution systems. *Urban Water J.* **14** (9), 972–983.

First received 12 October 2022; accepted in revised form 5 March 2023. Available online 17 March 2023