Machine Learning and Statistical Methods for Time Series Forecasting: a Case Study with Water Demand

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

19-07-2021 / 26-07-2021

CITATION

Zerara, Mohamed (2021): Machine Learning and Statistical Methods for Time Series Forecasting: a Case Study with Water Demand. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.15019698.v2>

DOI

[10.36227/techrxiv.15019698.v2](https://doi.org/10.36227/techrxiv.15019698.v2)

Machine Learning and Statistical Methods for Time Series Forecasting: a Case Study with Water Demand

Mohamed Zerara, Pierre Sibut-Bourde, Mohamed Abid, Askar Syrlybekov
Matthias Schneider, Udeneev Semen, Bakytzhan Salimov

Abstract—Nowadays, time series forecasting is commonly performed using several methods which can be classified in two groups; the stochastic methods and the machine learning based approaches. The former group includes amongst other auto-regressive (AR), moving average (MA), a combination of AR and MA called ARIMA (with a differential term), the seasonal ARIMA called SARIMA, the vector auto-regression (VAR) technique for multi-variate series, and others. The latter group includes regular neural networks such as the multi-layer perceptron (MLP), more sophisticated deep learning algorithms such as GRU and LSTM, two recurrent neural networks, support vector machines (SVM) and its application to regression problems (SVR), a more recent algorithm called XGBoost, and others. In this study, most of these approaches and techniques will be used to forecast water demand of a public company which is in charge of distributing water, as potable water, to several hundred thousand of end-customers. These different methods will be briefly introduced, then assessed with the water demand case study and discussed in details.

Index Terms—Machine learning, Forecasting, Time series, Water demand, ARIMA, SARIMA, MLP, LSTM, SVR, Random Forest Regressor, XGBoost

I. INTRODUCTION

THIS section introduces some general issues on water management and scarcity. Then, internet of things (IoT), machine learning are introduced. Finally, time series forecasting are briefly introduced.

A. Water loss and scarcity issues

Nowadays, with the rapid population increase and the development of countries, water has experienced a high pressure for expansion and betterment. Effectively, the economic development with the related consequence of urbanization, global warming, and industrialization- it is estimated that almost half

of the population will live in water-stressed areas by 2025. This means that water will become one of the biggest challenges and financial expenses for cities and thus directly impact the economies. For example, industries are already looking for a system to reduce their expenses on water and water waste treatment. Furthermore, in most of the countries, 20-25% of the fresh water produced is lost due to leakages in the public water network.

In the last few years, the rapid development of Internet of Things (IoT) which enables a real-time monitoring of infrastructure (temperature, humidity, light, energy and water consumption, and others) is spreading rapidly in cities around the world [1]–[11]. In the field of Water management, the development of this technology is still in his infancy and the monitoring platforms rely on expensive IoT platforms. The impact of this technology based on nodes of sensors will provide a real-time monitoring, thus improving water management.

B. Internet of things and machine learning

By combining the IoT and machine learning, we developed an automated low-cost, compact and low-consumption system named LinkH2O which monitors in real-time the flow, pressure, temperature, and also controls the flow in a particular points of the water infrastructure. Furthermore, the use of artificial intelligence is helping us to manage (forecasting, detecting anomalies) and responding appropriately to any damage (burst and freezing pipe) by controlling the electrovalves. More generally, our system technology is not solely suitable for detecting leaking but also for water scarcity management by rerouting water for strategic facilities (hospital, fire department) when necessary.

Finally, in order to forecast and detect anomalies in the water network, time series analysis of the incoming data is of tremendous importance.

C. Time series and forecasting

Time series forecasting is the subject of intense research in many different fields such as physical sciences, medical and health monitoring, engineering (electricity, water demand, engines), meteorology etc [12]–[22].

Here, we propose to assess, test and compare in our set of data the most commonly used methods for forecasting, in order to choose and validate and the most suitable method.

Corresponding author : Mohamed Zerara is with University of Applied Sciences, Geneva, Switzerland (email: mohammed.zerara@hesge.ch)

Pierre Sibut-Bourde is with Institut National des Sciences Appliquées (INSA), Lyon, France (email: pierre.sibut-bourde@insa-lyon.fr)

Mohamed Abid was with Beijing Institute of Technology, Beijing, China (email: mohamed.abid@bit.edu.cn)

Askar Syrlybekov is with Nazarbayev University, Nursultan, Kazakhstan (email: askar.syrlybekov@nu.edu.kz)

Matthias Schneider is with University of Applied Sciences, Geneva, Switzerland (email: matthias.schneider@hesge.ch)

Udeneev Semen is with Instrumentation and Control Department, Astana su arnasy, Nursultan, Kazakhstan

Bakytzhan Salimov is with Jsc Kazakhstan Center for Modernization and Development House and Utilities Services, Nursultan, Kazakhstan

Water demand will be analyzed as time series to determine seasonality, trend and any other feature. Correlations, autocorrelations will be evaluated using statistical methods. Then, statistical methods will be applied to forecast both pressure and water demand. Then, state-of-the-art machine learning algorithms will be used to perform predictions. Finally, performance of each approach will be measured and discussed.

II. WATER INFRASTRUCTURE

The water-supply infrastructure in Nursultan city consists of what is built to pump, transport, filter, treat, store, and deliver technical and drinkable water. In Nursultan, this infrastructure includes two main reservoirs (sources), surface-water intakes, secondary reservoirs, storage tanks, drinking-water facilities, and pipes.

ACA company is a public company in charge of water treatment and distribution in the capital city of Nur-Sultan (Kazakhstan). Figure 1 represents a schematic of ACA main infrastructure.

The water-supply system generally includes water transmission system which is the network of pipes that is used to convey water from water treatment plants to service reservoirs, while the water distribution system refers to the network of pipes that supplies water from the service reservoirs and balancing reservoirs to end-consumers. In the present work, water demand from reservoirs of the water transmission system, that is the main infrastructure, is considered.

Forecasting water demand on several thousands of LinkH2O end-node devices can be simplified by doing this study on the main water infrastructure of ACA. Subsequent studies will be performed to a larger scale by including more sensors, but are not included in the present work.

III. THEORY

There exist plethora of methods and approaches for time series forecasting. In this section, we will describe two classes of models; the stochastic based one and the machine learning based ones which encompasses neural networks, support vector machines and tree based methods. The aim of this section is to introduce the theoretical foundations of some of the most popular methods. It is a prerequisite before assessing their performance on water demand predictions, and discussing their strengths and weaknesses.

A. Time series : theoretical considerations

a) Definition: Time series is a finite sequence of data measured over successive times, usually at fixed intervals, e.g. every day, month, hour, minute, second, etc. Given a random variable X which is assumed real valued, a timespan T , a time series is defined as a series of observations X_t measured at time $t \in \mathbb{R}$. If $T \subseteq \mathbb{Z}$, the time series is said to be discrete. In contrast, if $T \subseteq \mathbb{R}$, time series is said to be continuous. A mathematical expression for time series is $(X_t : t \in \mathbb{Z})$; X_t is a vector. Although there exist continuous time series, the purpose of this study is the forecasting of discrete time series with hourly measured data.

A time series in general is composed of different terms : trend, seasonality and cycle, and a random component. Any time series can be decomposed according to, either a multiplicative model defined as:

$$X_t = m_t \times S_t \times Y_t \quad (1)$$

or an additive model, defined as:

$$X_t = m_t + S_t + Y_t \quad (2)$$

It is worth notice that the additive model can be obtained from the multiplicative one by applying an algorithmic transformation.

$$\log(X_t) = \log(m_t) + \log(S_t) + \log(Y_t)$$

In the above expressions,

- $\{m_t, t \in T\}$ is the trend component, with slow variations.
- $\{s_t, t \in T\}$ is the seasonality : there exist a minimal d , called seasonal periodicity, such that for any $t \in T$, $s_{t+d} = s_t$
- $\{Y_t, t \in T\}$ is the random component of the time series.

b) Stationarity:

c) Weak stationarity: A time series $\{X_t\}$ is called *weakly stationary* or just *stationary* if

- $E[X_t] < \infty$; the expectation of X_t is finite
- $\gamma(X_t, X_{t+\tau}) = \gamma(X_\tau)$; for each time lag τ the autocovariance of $\{X_t\}$ and $\{X_{t+\tau}\}$ is constant and does not dependent on t .

d) Strict stationarity: A time series $\{X_t\}$ is called *strictly stationary* if

- All random variables X_t are identically distributed for all t ,
- Pairs of random vectors $(X_t, X_{t+\tau})^T$ are identically distributed for all times t and lags τ . That is,

$$(X_t, X_{t+\tau})^T \stackrel{d}{=} (X_1, X_{1+\tau})^T$$

e) ACF and PACF: The autocorrelation function (ACF) and partial autocorrelation (PACF) are statistical measures that help understand how observations in a time series are related (correlated) for different time shifts (lags).

The autocorrelation function is defined as ratio of the covariance functions γ_k and γ_0

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (3)$$

where

$$\gamma_k = \text{cov}(X_t, X_{t+k}) = E[(X_t - \mu)(X_{t+k} - \mu)]$$

here k is the lag, cov is the covariance, $\mu = E[X_t]$ the mean of the time series, γ_k and γ_0 are respectively the autocovariance at lags k and 0 (no lag) [23].

On the other hand, partial autocorrelation is the partial correlation of a time series with its own lagged values, that is correlations for lag k are estimated for values $0 \rightarrow k$, controlling for the correlations at all shorter lags. Detailed mathematical expressions can be found elsewhere [24]–[26].

Plots of ACF and PACF functions against time lags are called correlograms. They are used to help us determine the orders of AR and MA, the auto-regressive and moving average models respectively.

Fig. 1. Schematic diagram of ACA water-supply infrastructure in Nursultan (Kazakhstan). It is composed of pipes, pumping facilities, a filtering station, and several reservoirs (noted Res. and R1 to R9).

f) *Box-Jenkins transformation*: The Box-Jenkins methodology [27]–[30] is used to select the most appropriate statistical ARIMA model that can lead to accurate forecasts of water demand. Following the preliminary analysis, that is detection of trend, seasonality and cycles, a three stage process is applied to

- Define what class of models are able to reproduce the time series
- Determine the model parameters; the orders of the various terms in ARIMA
- Validate the model and its parameters using the residuals

The Box–Jenkins methodology for ARMA models relies on visual inspection of the (partial) correlograms to estimate the orders p and q of an $AR(p)$, $MA(q)$ or $ARMA(p, q)$.

- 1) Apply preliminary transformations of the X_t data
 - a) such as Box–Cox algorithm
 - b) differencing to remove the time series trend so that the average of the time series does no longer depend on time.
- 2) Correlogram
 - a) Determine the $MA(q)$ order by looking at ACF function, at the points for which $\rho_k \neq 0$ for $k \leq q$ and $r_k \approx 0$ for $k > q$.
 - b) For an $AR(p)$ process, the auto-correlation function should decay exponentially, with possible oscillation patterns.
 - c) For an $ARMA(p, q)$ model, the pattern is irregular for lags $k \in \{1, \dots, p\}$ and go to zero as $k \rightarrow \infty$.
- 3) Partial correlogram
 - a) Parameters should be zero at lags $k > p$ for the $AR(p)$ model, and nonzero otherwise
 - b) The parameters decay exponentially in the $MA(q)$ model
 - c) The parameters decrease to zero as $k \rightarrow \infty$ for the $ARMA(p, q)$ model.

Determination of optimal model parameters is crucial to avoid either bias or high variance leading respectively to underfitting and overfitting. ACF and PACF are one approach to determine the optimal parameters. Another approach is based on measures called *Akaike Information Criterion (AIC)* [31]–[33] and *Bayesian Information Criterion (BIC)* [34]–[36]

$$AIC(p) = n \ln(\hat{\sigma}_e/n) + 2p \quad (4)$$

$$BIC(p) = n \ln(\hat{\sigma}_e/n) + p + p \ln(n) \quad (5)$$

where p is the number of parameters in the model, n is the number of observations used for fitting the model, $\hat{\sigma}_e$ is the sample squared residuals (errors).

B. Stochastic methods

Two linear models are widely used for time-series predictions; the *auto-regressive (AR)* and *moving average (MA)* models. Combinations of AR and MA with additional terms lead to popular models such as the *auto-regressive moving average (ARMA)*, the *auto-regressive integrated moving average (ARIMA)*, the *seasonal auto-regressive integrated moving average (SARIMA)*.

a) *AR - Auto-regressive model*: In the auto-regressive model, the future value is given as a linear combinations of past values, e.g. previous water demand or water pressure values. If p past observations are considered, $AR(p)$ can be expressed as a regression against the p past values :

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t = c + \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \varepsilon_t \quad (6)$$

where $\varepsilon_t \sim WN(0, \sigma^2)$ is the random shock or white noise at time t , and p is the *order* of the $AR(p)$ model. ε_t is uncorrelated with X_s for each $s < t$. c is a constant that depends on the mean $E[X_t] = \mu$,

$$c = \mu(1 - \phi_1 - \dots - \phi_p)$$

Equation 6 can be rewritten in vector notation :

$$X_t = \phi_t^T \mathbf{X}_{t-1} + \varepsilon_t$$

with vectors

$$\phi_t = (\phi_1, \dots, \phi_p)^T$$

$$\mathbf{X}_{t-1} = (X_{t-1}, \dots, X_{t-p})^T$$

b) *MA - Moving average*: In the moving average model, X_t is given by a linear combination of the q past errors :

$$X_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i} = \mu + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} \quad (7)$$

where μ is the average of the q past values, ε_t is the random shock at time t . Here also, q is called the order of the $MA(q)$ model.

c) *ARMA*: $ARMA(p, q)$ is the combination of $AR(p)$ (Eq. 6) and $MA(q)$ (Eq. 7) :

$$X_t = c + \varepsilon_t + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} \quad (8)$$

The orders p and q refer to p auto-regressive and q moving average terms. Eq. 8 can be rewritten as

$$\underbrace{X_t - \phi_1 X_{t-1} - \dots - \phi_p X_{t-p}}_{AR(p)} = \underbrace{\varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}}_{MA(q)}$$

d) *ARIMA*: While AR , MA and their combination $ARMA$ can only be applied to stationary time series, the $ARIMA$ can be used even for non-stationary time series. In $ARIMA$, stationarity is obtained by introducing the differencing between successive data points into the model

$$\nabla^d X_t = (1 - B)^d X_t = \sum_{k=0}^d \binom{d}{k} (-1)^k X_{t-k}$$

Where B is the backshift operator

$$BX_t = X_{t-1}$$

From the above notations, one can express $ARIMA(p, d, q)$ as

$$\left(1 - \sum_{i=1}^p \phi_i B^i\right) (1 - B)^d X_t = \left(1 + \sum_{j=1}^q \theta_j B^j\right) \varepsilon_t \quad (9)$$

Here, p , d , q are the orders of the auto-regressive (AR), integrated (I), and moving average (MA) terms.

e) *SARMA and SARIMA*: In these models, seasonality is dealt with by differencing at appropriate order s to remove non-stationarity from the time series. For a time series with hourly measured data, the seasonality order is then $s = 24 \times 365$ (or 366 days for leap years). The mathematical expression of $SARIMA(p, d, q) \times (P, D, Q)_s$ is

$$\Phi_P(B^s) \phi_p(B) \nabla_s^D \nabla^d X_t = \Theta_Q(B^s) \theta_q(B) \varepsilon_t \quad (10)$$

where

$$\nabla_s^D X_t = (1 - B^s)^D X_t, D \in \mathbb{N}^*$$

where

Fig. 2. Network graph of a $(L+1)$ -layer perceptron with d input units and C output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_P B^{Ps}$$

and

$$\Theta_Q(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{2s} - \dots - \Theta_Q B^{Qs}$$

are respectively the seasonal AR and seasonal MA operators with period s .

Seasonal $ARMA(p, q) \times (P, Q)_s$ can be deduced from Eq. 10 by omitting the differencing terms

$$\Phi_P(B^s) \phi_p(B) X_t = \Theta_Q(B^s) \theta_q(B) \varepsilon_t \quad (11)$$

C. Neural networks

a) *Preliminaries*: To make the study self-contained, a brief review of multi-layer perceptrons (MLP), long-short term memory (LSTM) networks, support vector regressors (SVR), XGBoost and Random Forests are given as preliminaries.

b) *MLP*: The multi-layer perceptrons (MLPs) are the most widely used artificial neural networks (ANN) for either classification or regression problems. This class of models is composed of an input layer, one or more hidden layers, and an output layer. Figure 2 is a representation of a three layers MLP.

Such as a network graphs is able to compute the following output

$$X_t = \alpha_0 + \sum_{l=1}^L \left(\sum_{j=1}^q \alpha_{jl} g \left(\beta_{0jl} + \sum_{i=1}^p \beta_{ijl} X_{t-i} \right) \right) + \epsilon_t \quad (12)$$

Here, the integers L, p, q are respectively the number of hidden layers, the number of inputs $X_t (i = 1, 2, \dots, p)$, the number of nodes in a given hidden layer. The activation function g can be for instance the *sigmoid* ($g(x) = 1/(1 + e^{-x})$), *tanh* ($((e^x - e^{-x})/(e^x + e^{-x}))$) or *ReLU* ($g(x) = \max(0, x)$) function.

For a network with a single hidden layer, Eq. 12 simplifies to

$$X_t = \alpha_0 + \sum_{j=1}^q \alpha_j g \left(\beta_{0j} + \sum_{i=1}^p \beta_{ij} X_{t-i} \right) + \epsilon_t \quad (13)$$

c) *LSTM*: The LSTM is a recurrent neural network architecture that is formed of three gates and two states :

- forget gate
- input gate
- output gate
- hidden state
- cell state

The overall diagram of this network is shown in figure 3. We will always refer to it in the following explanation. In this diagram, the sigmoid $\sigma(x) = 1/(1 + e^{-x})$ and the hyperbolic tangent $\tanh(x) = (e^x + e^{-x})/(e^x - e^{-x})$ activation functions are applied to vector's components.

We also have element-wise addition and multiplication operations, noted respectively $\odot$ and $\oplus$. Finally, when two lines meet, there is a vertical concatenation between the two associated vectors. This operation is defined as $\dagger$: $X \dagger Y = \begin{pmatrix} X \\ Y \end{pmatrix}$.

Unsing these notations, one can express $h^{(t)}$ and $c^{(t)}$ as a function of $h^{(t-1)}$ and $c^{(t-1)}$ and $x^{(t)}$.

$$\begin{aligned} c^{(t)} &= \left(c^{(t-1)} \odot \sigma \left(\Theta_f \left(h^{(t-1)} \dagger x^{(t)} \right) \right) \right) \oplus \\ &\quad \left(\left(\tanh \left(\Theta_c \left(h^{(t-1)} \dagger x^{(t)} \right) \right) \odot \sigma \left(\Theta_i \left(h^{(t-1)} \dagger x^{(t)} \right) \right) \right) \right) \\ h^{(t)} &= \tanh \left(c^{(t)} \right) \odot \sigma \left(\Theta_o \left(h^{(t-1)} \dagger x^{(t)} \right) \right) \end{aligned}$$

where $x^{(t)}$, $c^{(t)}$, $h^{(t)}$ are respectively the input signal (value of the time series at time t), the estimated output value for time t and the cell state at time t . Matrices Θ_f , Θ_i , Θ_c , Θ_o are parameters of the *LSTM* model.

Fig. 3. Graph of a LSTM cell.

D. Support vector machines

a) *SVR*: The support vector regression (SVR) algorithm uses the ϵ -insensitive loss function, proposed by Vapnik [37]–[39]. In SVR, a non-linear transformation Φ is applied to the time series data X_t , from the input space to a higher dimensional space :

$$\Phi : \mathbb{R}^n \rightarrow F$$

$$f(x) = \langle w, \Phi(x) \rangle + b$$

Here, $w \in F$ is parameters (also called weights) vector that minimizes the linear function f , $b \in \mathbb{R}$ is a constant.

For the minimization process, SVR generally chooses the ϵ -insensitive loss function [40] rather than more conventional loss functions such as the least mean average error (MAE) and the mean absolute percentage error (MAPE).

To minimize the weights vector w and hence the function f , one has to minimize the risk regularized function

$$\begin{aligned} \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s.t. } y_i - \langle w, \Phi(x_i) \rangle - b \leq \epsilon + \xi_i \\ \langle w, \Phi(x_i) \rangle + b - y_i \leq \epsilon + \xi_i^* \end{aligned} \quad (14)$$

where $\epsilon \geq 0$ is the distance between the actual value y and the estimated function f . Slack variables $\xi, \xi^* \geq 0$ are introduced to allow errors greater than ϵ . C is the regularization constant used to specify the trade-off between generalization and accuracy while fitting training data [41].

In practice expressions of w and f based on Lagrangian multipliers are used

$$\begin{aligned} w &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(x_i) \\ f(x) &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x) - b \end{aligned}$$

where, $\alpha_i, \alpha_i^* \leq C$, and $K(x_i, x)$ represents the inner product between $\phi(x_i)$ and $\phi(x)$, which are called the kernel functions. More details on support vector machines and their use for regression problems can be found in literature [42], [43].

E. Tree based methods

a) *Random forest regressor*: Random forests (RF) are ensemble learning models [44] that can be used for regression by averaging a large number of individual regression trees $r_n(x, \Theta_n, D_n)$, where Θ is a parameters vector of the model, and D_n is the training set $(X_1, Y_1), \dots, (X_n, Y_n)$. The regression function which is the estimate of the aggregated trees is

$$\bar{r}_n(X, D_n) = \mathbb{E}_{\Theta} [r_n(X, \Theta, D_n)] \quad (15)$$

where $\mathbb{E}_{\Theta}$ is the conditional mean or expectation for a given parameter vector Θ . Both bagging and feature randomness are used for building a forest of uncorrelated individuals trees. Then prediction by committee leads to more accurate result than that of any individual tree.

Detailed implementations of the random forest algorithm are given in references [45], [46].

b) *XGBoost*: XGBoost (eXtreme Gradient Boosting), an implementation of the gradient tree boosting (GBT) [47], is also a tree ensemble machine learning method [48]. The prediction at time (or step) t is defined as

$$\hat{y}^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_i(x_i)$$

where x_i is the feature vector or input observation, that is the previous time values from the time series data set. And $f_i(x_i)$ is the learner, typically a regression tree, at time t . In order to prevent the over-fitting of the training data, the XGBoost model uses a regularized objective function

$$O^{(t)} = \sum_{k=1}^n l(\hat{y}, y) + \sum_{k=1}^t \Omega(f_i)$$

Where l is the loss function (e.g: *RMSE*, the cross entropy function, ...), n the number of learners (trees), and Ω represents the regularization term, which is defined as follows

$$\Omega(f) = \gamma T + \frac{1}{2} \|w\|^2$$

where T is the number of leaves, Ω is the leaf score, and λ is the regularization parameter. Parameter γ represents the minimum loss value required to split the leaf node.

More details on the XGBoost model and its implementation can be found in the work of Chen and Guestrin [48].

IV. RESULTS AND DISCUSSION

A. Dataset, trends and patterns

The dataset considered here consists of hourly potable (reservoirs R1 to R7) and technical (reservoirs R8 and R9) water demands (see Fig. 1) recorded from 1st of January 2017 to 31st of December 2019. Here, the water demand is expressed in cubic meters (m^3) measured every hour. As mentioned previously, the data of the main infrastructure were provided by ACA, the governmental water company located in the capital of Kazakhstan, a city with more than one million inhabitants.

The dataset is recorded hourly for a period ranging from 1st of January 2017 to 31st of December 2019. It includes data related to the water potable reservoirs (R1 to R7) and technical water (reservoirs R8 and R9) which correspond to industrial consumption.

Figure 4 shows the water consumption of reservoirs (R8 to R9) for the period from 2017.01.01 to 2019.12.31. First, we can note that there is over the years a slight linear increase for all reservoirs, which can be correlated to a population increase (Figure 4).

Fig. 4. Water demand from reservoirs R1 to R9 (from 2017.01.01 to 2019.12.31).

In narrower time spans (12 days and 21 days), Figures 5 and 6 reveal periodic features with a period of 24 hours. This main feature is not surprising as the water consumption is related to human activities. For example, the water demand is lower in the night and we observe peaks of consumption outside the working hours (in the morning and evening). In the morning and in the evening, water use increases resulting in peaks as shown in figure 5.

On the other hand, the consumption pattern on technical water (reservoirs R8 and R9) is very different than that for the potable water.

Figure 5 shows the water consumption of reservoirs R1 to R9 for a period of 12 days.

Fig. 5. This plot shows the water consumption of reservoirs R1 to R9 for a period of 12 days.

Fig. 6. This plot represents the water consumption in the potable reservoirs R1 and technical reservoir R8 for three weeks data.

However, the demand pattern from technical reservoirs R8 and R9 is different than that for the other water potable reservoirs (R1 to R7). This is due to different usage of technical water because industrial plants, cooling infrastructure or irrigation do not obey to same usage characteristics. Figure 6 shows the difference of patterns in a potable water reservoir (R1) and a technical water reservoir (R9). The curve corresponding to reservoir R9 does not show any obvious cycle nor repeated pattern. Technical water use might be influenced by exogenous variables such as weather conditions especially in outdoor water use (e.g. irrigation). The present study focuses on domestic

water use, therefore no further investigation on technical water use will be performed.

Trend and seasonal components have been removed from the water demand curves and a resulting curve is shown for reservoir R1 (Fig. 7). From this figure it appears that the demand has a periodic components; every sundays and national holidays (2nd and 16th-17th of december 2019) one can see a similar pattern and an increase of the water consumption. This is certainly related to the fact that people spend more time at home rather than in their offices. Information on official holidays in 2019 from *Official Holidays* web site [49].

A seasonal component with a periodicity of twenty-four hours is also observed. It is due to characteristics and habits in water usage of the population.

Fig. 7. This plot represents the trends, seasonality and residues corresponding to the demand from reservoir R1.

Figure 8 shows the residue after removing both trend and seasonality from demand arising from R1 reservoir. From this figure and the zoomed inset, it appears that no trend nor seasonality is visible on the graph. The Augmented Dickey-Fuller test and p -value analysis for the resulting curve have been performed on the entire R1 data set (three years) and both confirm the the residual curve is stationary. The next table displays results of the test: a statistical value of -41.3987 lower than critical values -3.4306, -2.8616, -2.5668 at 1%, 5%, and 10% confidence intervals respectively. The p -value is very close to zero (rounded to zero by python calculation). These values confirm the visual inspection and hence that the resulting residues curve is stationary with a confidence interval up to 99%.

TABLE I
P-VALUE

t -Statistics	p -value	CI	Critical values
-41.3987	0.0000	1%	-3.4306
		5%	-2.8616
		10%	-2.5668

B. ARIMA

The computation of auto-correlation (ACF) and partial autocorrelation (PACF) functions are represented on figure 9 as correlograms for different lag values.

Fig. 8. The current plot represents the residue to the water consumption from reservoir R1.

Fig. 9. Autocorrelation and partial autocorrelation for different lag values (demand from R1 reservoir).

Both *ACF* and *PACF* estimates by default start with lag 0, at which it always takes the value 1. A cyclic behavior with a period of 24 is apparent in the time series plot of the original water demand data corresponding to any potable water reservoir (R1 to R7). Moreover, the absolute value of the correlations attenuates with increasing lag. Next, we will discuss the interpretation of the correlograms.

ACF correlogram (top of figure 9) indicates a strong spike at lag 1 and smaller spikes at lags 2 and 3. This can be represented by a moving average process of order 1,2,3; $MA(1)$, $MA(2)$, $MA(3)$ for potable water demand. A similar behaviour can be observed for any of the stationarized potable water reservoirs demand. Spikes are also observed around lag 24. This may be due to the demand characteristics ; water usage being somehow dependent on human habits. Indeed, most of people sleep at night, have a shower in either the morning or evening etc.

On the other hand, *PACF* shows significant correlations up to lag 2 indicating that $AR(2)$ could be a good enough model to describe all contributing terms in the moving average approximation.

However, both *Akaike Information Criterion (AIC)* and *Bayesian Information Criterion (BIC)* suggest that orders $p = 4$, $q = 3$ for $ARMA(p,q)$ lead to more accurate regression on the R1 training data set. The results are also confirmed by

the log-likelihood analysis which shows a minimal value for $p=4$, $q=3$.

Therefore, one can conclude that a reasonably good model for the stationarized water demands (arising from reservoirs R1 to R7) would be $ARIMA(4,0,3)$ or $ARMA(4,3)$.

In this study, we were not able to go to high orders of p and q . Indeed, it should be kept in mind that the time series studied has more than 26,280 values. Thereby, several problems arised.

- Time problem: a time consuming algorithm does not provide a response in a reasonable time, and so has been inoperative with our computing tools (a single personal computer).
- Memory problem: considering the large size of our data, any calculation requires significant memory space. Memory appears to be a crucial point in modeling such a large series.

TABLE II
BIC

	0	1	2	3	4
0	342543.41	323252.69	316784.07	314724.09	314317.57
1	318474.97	314957.58	314485.38	314309.99	312964.70
2	314037.64	313935.72	311170.17	311149.01	311155.80
3	313990.59	311228.95	311153.85	311158.02	310767.16
4	313816.15	311137.90	310813.85	310512.58	311017.78

TABLE III
AIC

	0	1	2	3	4
0	342527.06	323228.16	316751.37	314683.20	314268.51
1	318450.44	314924.87	314444.50	314260.93	312907.46
2	314004.93	313894.84	311121.11	311091.77	311090.39
3	313949.71	311179.89	311096.62	311092.60	310693.57
4	313767.09	311080.66	310748.44	310438.99	310936.02

TABLE IV
LOG-LIKELIHOOD

	0	1	2	3	4
0	-171261.53	-161611.08	-158371.68	-157336.60	-157128.25
1	-159222.22	-157458.43	-157217.25	-157124.46	-156446.73
2	-156998.46	-156942.42	-155554.55	-155538.88	-155537.19
3	-156969.85	-155583.94	-155541.31	-155538.30	-155337.78
4	-156877.54	-155533.33	-155366.22	-155210.49	-155458.01

C. Machine learning

Several machine learning methods, including simple neural networks, recurrent neural networks, support vector machines, tree-based methods have been studied in order to evaluate their predictive abilities on the time series that constitute the data water consumption.

Those algorithms were implemented using Scikit-learn, Statsmodels, Keras and TensorFlow libraries which were installed along with the python interpreter and Jupyter notebook.

a) *Train, validation and test sets*: The dataset has been split into train, validation and test sets. The tree sets representing 71%, 10% and 19% of the whole dataset (3 years data of

hourly measured demand) were used for every method to train, validate and test the best model.

The datasets were cleansed; e.g linear interpolation was used to fill in missing values. Each value of the three datasets was then scaled to help prevent any convergence and saturation issues. Data were shrinked within the range $[0, 1]$.

b) *Model optimization*: Learning curves of model performance on the train and validation datasets were used to diagnose any underfit or overfit. The performance of each candidate model was plotted as a loss function (one for the training set and another for the validation set) vs the epochs. By comparing and checking the shapes of the plotted loss functions, one could easily check whether the model may lead or not to high variance (overfit) or bias (underfit).

When the model has too many hyperparameters, k-fold cross-validation method is used since there are too many combinations of hyperparameter values. Therefore a manual approach is tedious and must be avoided. In this approach, the training set is split into k smaller sets. The following procedure is followed for each of the k folds:

- a model is trained using $k - 1$ of the folds as training data.
- the model is validated on the remaining part of the data.
- the two steps above are repeated k times using another part of the data for validation.

The performance reported by k-fold cross-validation is obtained by averaging the values computed for the k steps. This approach can be computationally expensive, but does not require too much data as is the case when fixing an dedicated validation set. there are other approaches that may slightly differ but generally follow the same principles.

c) *Inputs*: To train our neural network, we have decided to use the past 3 values as inputs with the value at any corresponding time instant. This means we will have now an additional 3 columns where each row will have the water demand at any hour and the corresponding demands of the past 24 hours, the past 7×24 hours, the past 365×24 hours. These past values are included to account for correlations that have been observed previously from visual inspection of the water demand curves. Instead of using a single column, the actual water demand at each reservoir (apart from the datetime index of course), we use a four columns vector (Fig. 10 and Tab. V).

d) *Results*: MLP The simple Multi-layer Perceptron has an input layer with 4 nodes (accounting for each of the 4 past regressor inputs). The inner layer has 4 nodes and the *ReLU* activation was applied to input data. As for regression applications of *MLP*, the output layer has a single node while the linear activation function was used. *Adam*, the algorithm for first-order gradient-based optimization of stochastic objective functions [50] which is based on adaptive estimates of lower-order moments, is used. The learning rate was set to 0.001, while the exponential decay rates were set to $\beta_1=0.9$, $\beta_2=0.999$. The mean squared error (*MSE*) was used as a loss function.

From tables VI and VII one can see that the error roughly increases in proportion with the number of days included in the test set. R^2 , the square of the Pearson's correlation coefficient, and *MAE*, the mean average error indicate how the regression

Fig. 10. Slicing of one dimensional timeseries to give slices of 4 values; at times t , $t - 24$, $t - 168$ (24 hours $\times$ 7 days), $t - 8760$ (24 hours $\times$ 365 days). The obtained input is a two dimensional array $[X_1 \ X_2 \ \dots \ X_n]$.

TABLE V
FIRST TEN LINES OF THE INPUT DATA FOR R1 LOCATION; EACH LINE IS AN INPUT VECTOR X_i WHOSE COMPONENTS ARE DATA AT TIMES t , $t - 24$, $t - 168$ AND $t - 8760$.

	$R1_t$	$R1_{t-24}$	$R1_{t-168}$	$R1_{t-8760}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_{i-4}	1994	2152	1882	1515
X_{i-3}	1680	1758	1488	1642
X_{i-2}	1882	1454	1267	1498
X_{i-1}	1718	1244	1080	1317
X_i	1544	1150	1040	1180
X_{i+1}	1380	1128	1094	1085
X_{i+2}	1284	1212	1574	1049
X_{i+3}	1266	1332	2172	1049
X_{i+4}	1284	1466	2056	1143
X_{i+5}	1354	2102	2076	1375
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

predictions fits the data. One can see to what extent the trained model is able to perform forecasts at different times. The same model (not updated) can be used up to 90 days; however the forecasts are less reliable at 90 days than forecasts between 60 days and 5 days. The best results are found for forecasts at 1 day where almost 98% (R^2) of the data is explained by the *MLP* model. Figure 11 shows an example of prediction for water demand related to *R1* reservoir.

Fig. 11. Plot of actual vs predicted values using MLP model (demand from R1 reservoir).

LSTM The network has a single LSTM layer with 16 internal units. This was an optimal value that prevented from both overfitting and underfitting. The *sigmoid* activation function was chosen for the LSTM network. As for regression applications, the output layer consisted in a single node and the linear activation function was applied to output values. As for *MLP*, the *Adam* algorithm is used with the same parameters values : the learning rate was set to 0.001, and the decay rates were set to $\beta_1=0.9$, $\beta_2=0.999$. The mean squared error (*MSE*) was also used as a loss function. On figure 12 one can see an example of prediction for water arising from *R1* reservoir.

Fig. 12. Plot of actual vs predicted values using LSTM model (demand from R1 reservoir).

From tables VI and VII one can see that the error roughly increases in proportion with the number of days included in the test set. Based on R^2 coefficients and *MAE* analysis, it can be seen that the performance of the initial model (without further training) leads to less reliable forecasts at 90 days than forecasts in between 60 days and 30 days. For 7 days and 5 days, R^2 which has a value of almost 0.97 shows that these predictions are more reliable. The best results are found for forecasts at 1 day where almost 99% of the data is explained by the *LSTM* model.

SVR Several combinations of C , the regularization constant, and ϵ parameter (see Eq. 14) have been used to find an optimal one that leads to the best SVR model, while using the radial basis function (*RBF*) as a kernel function. Values of $1 \leq C \leq 10$ and $\epsilon = 0.1$ give the best results; C and ϵ were set to 1 and 0.1 respectively.

TABLE VI
FORECASTING RESULTS USING TEST DATA SET FOR WATER DEMAND ARISING FROM RESERVOIR *R1* (R^2 VALUES).

<i>Model</i>	90 days	60 days	30 days	15 days	7 days	5 days	1 day
<i>MLP</i>	0.9109	0.9442	0.9414	0.9442	0.9541	0.9489	0.9786
<i>LSTM</i>	0.9151	0.9474	0.9413	0.9480	0.9665	0.9634	0.9898
<i>SVR</i>	0.9335	0.9469	0.9455	0.9502	0.9635	0.9624	0.9873
<i>XGBoost</i>	0.9171	0.9271	0.9248	0.9284	0.9382	0.9340	0.9635
<i>Random Forest</i>	0.9164	0.9232	0.9168	0.9220	0.9316	0.9275	0.9566

TABLE VII
FORECASTING ERRORS USING TEST DATA SET FOR WATER DEMAND ARISING FROM RESERVOIR *R1* (*MAE* ERRORS). FOR OBVIOUS PRACTICAL REASON (SMALLER NUMBERS) *MAE* WERE CALCULATED ON SCALED DATA (SCALED TEST DATA AND SCALED PREDICTED VALUES).

<i>Model</i>	90 days	60 days	30 days	15 days	7 days	5 days	1 day
<i>MLP</i>	132.14	109.33	113.40	109.94	100.16	114.40	58.83
<i>LSTM</i>	136.04	113.87	122.46	115.04	95.73	105.87	54.12
<i>SVR</i>	128.02	115.36	120.17	115.75	101.55	109.27	60.22
<i>XGBoost</i>	143.29	135.14	138.03	135.98	130.02	141.73	115.49
<i>Random Forest</i>	142.70	137.62	144.85	139.34	136.01	147.53	119.05

Tables VI and VII show the performance of the *SVR* model in predicting water demand for reservoir *R1*. Based on R^2 coefficients and *MAE* analysis, it can be seen that the model is able to make forecasts up to 90 days and 30 days ($R^2 \sim 0.93 - 0.94$) which are almost as reliable as those at 7 days and 5 days ($R^2 \sim 0.95 - 0.96$). The best results are found for forecasts at 1 day ($R^2 \sim 0.99$). Figure 13 shows an example of prediction for water demand related to *R1* reservoir.

Fig. 13. Plot of actual vs predicted values using SVR model (demand from *R1* reservoir).

e) XGBoost: XGBoost models can be defined by selecting specific values for several parameters. They should be selected to maximize the model performance on a given dataset and in such a way to prevent overfitting and unnecessary complexity as well as underfitting.

XGBoost uses multiple parameters, which makes parameter tuning relatively difficult. Such parameters include number of boosting iterations, maximum depth, random subsampling, learning rate, lambda, and alpha. The number of boosting iterations in *XGBoost* refers to the number of consecutive trees that are fitted. The maximum depth of the tree is the maximum number of splits; too high a maximum depth can cause overfitting. Randomly subsampling corresponds to a specific ratio of the training dataset in each iteration prior to growing

trees. A learning rate is used to shrink the weights at each step and make the optimization process more robust; essentially, the learning rate reduces the impact of each individual tree and let future trees improve the model (Chen and Guestrin, 2016). The parameters λ and α respectively are regularization terms on weights, and by increasing them the model becomes more conservative. Using a 10-fold cross-validation process combined to a grid search algorithm [51]–[53], one obtains the following values for each parameter: number of boosting iterations: 500; maximum tree depth: 25; subsample ratio of training dataset in each split: 1 (0.8); subsample ratio of columns in each split: 1 (0.5); $\lambda = 1$; $\alpha = 0.2$; learning rate = 0.1.

The performance of XGBoost in predicting water demand (test set from reservoir *R1*) without being retrained does not significantly improve while decreasing the forecasting period from 90 days to 5 days (~ 0.92 - 0.93). The prediction significantly improves for one day (24 hours) prediction ($R^2 \sim 0.96$). On figure 14 one can see an example of prediction for water arising from *R1* reservoir.

Fig. 14. Plot of actual vs predicted values using XGBoost model (demand from *R1* reservoir).

f) RF Regressor: Here, we have chosen to optimize two parameters of the model; the maximum depth of the tree and

Fig. 15. Plot of actual vs predicted values using the Random Forest regressor model (demand from R1 reservoir).

the number of trees in the forest. Depending on the values of these parameters, one can achieve better performance from the model while increasing the number of trees. However, one must be cautious since the model may lead to high variance (overfitting) issues.

For tuning the hyperparameters of the RF regressor model, a 10-fold cross-validation combined with the grid search algorithm [54] has been performed. Three parameters have been tuned; different values and ranges of values have been tried for the number of estimators, the maximum depth and the number of random states. Based on *RMSE* error, the following optimal values have been found : number of estimators ($n_estimators=100$), the maximum depth ($max_depth=6$), and number of random states ($random_state=100$).

The performance of the Random Forest regressor model in predicting water demand (test set from reservoir *R1*) with the initially trained model is somewhat similar than the performance of *XGBoost*. It does not significantly improve while decreasing the forecasting period from 90 days to 5 days ($\sim 0.92-0.93$). The prediction significantly improves for one day (24 hours) prediction ($R^2 \sim 0.96$). Figure 15 shows an example of prediction for water demand related to *R1* reservoir.

V. CONCLUSION

First, it is important to notice that all drinkable water reservoirs (*R1* to *R7*) have similar pattern and characteristics which demonstrate that they are connected to the same principal water network and not affected to specific city area (e.g Business, Commercial, residential). Therefore, the approach used in the present study can be easily used with the same level of reliability and confidence for any of those reservoirs. In contrast, technical water reservoirs (*R8* to *R9*) which are out of the current study's scope, need more features to be considered in order to have a reliable model for demand forecasting. For instance, the features may include weather conditions such as rain falls, temperature, etc. Several works have outlined the importance of these features [55]–[58].

The current study confirms that statistical methods have limited predictive power in practice due to the nonlinear nature of changes in water demand. In particular, forecasting beyond

a few hours time-step may not be accurate when using those models. Especially, when the demand drastically changes; these changes occur after a few hours as shown by the various plots of the present publication.

More sophisticated methods that are able to fit non-linear models such as *MLP*, *LSTM*, *SVR*, *XGBoost* and *Random Forest* perform significantly better when forecasting water demand. Forecasts based on the initially trained model (with no further updates of the model parameters) are very good for twenty-four hours (one day) for any these models even though one can see from table VII that *MLP*, *LSTM* and *SVR* models have much better predictive abilities. Then, the accuracy of the predictions decreases significantly for five days forecasts and beyond (seven days to 90 days) for all the methods that have been studied. However, *MLP*, *LSTM* and *SVR* have relatively good predictions up to sixty days.

Therefore, it is possible to use one of the studied machine learning algorithms after initial training for up to 90 days model. Then, it is highly recommended to update the model by further training using recent measured data. The input must include the best predictors which are the water consumption values recorded at the previous hour, the previous day, the previous week and the previous year (four predictors).

As a conclusion, one can easily deduce that the good forecasting abilities of artificial neural networks (*MLP*), recurrent neural network (*LSTM*), support vector machine regressors (*SVR*) and tree based regressors (*XGBoost* and *Random forest*) are twofold. First, their non-linear nature can fit water demand which are inherently non-linear. In contrast, the statistical methods such as *AR*, *MA* and *ARIMA* are unable to fit water demand and more generally non-linear models.

REFERENCES

- [1] H. Zhan, L. Wang, S. Chen, P. M. Kumar, and P. M. Shakeel, "Detection and alerting system of nearby medical facilities during emergency using iot sensors," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.
- [2] D. K. Sharma, S. Brahmachari, and I. Srivastav, "Role of iot and sensors in achieving sustainability," in *Green Automation for Sustainable Environment*, pp. 65–93, CRC Press, 2020.
- [3] S. Tanwar, S. Tyagi, and S. Kumar, "The role of internet of things and smart grid for the development of a smart city," in *Intelligent communication and computational technologies*, pp. 23–33, Springer, 2018.
- [4] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.
- [5] S. P. Mohanty, U. Choppali, and E. Kougiannos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.
- [6] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on emerging telecommunications technologies*, vol. 25, no. 1, pp. 81–93, 2014.
- [7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [8] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
- [9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [10] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things journal*, vol. 1, no. 2, pp. 112–121, 2014.
- [11] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz, J. A. Galache, M. Presser, L. A. H. Gómez, and J. Pettersson, "Smart cities at the forefront of the future internet," in *Future internet assembly*, pp. 447–462, 2011.
- [12] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.
- [13] F. Piccialli, F. Giampaolo, E. Prezioso, D. Camacho, and G. Acampora, "Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion," *Information Fusion*, vol. 74, pp. 1–16, 2021.
- [14] J. F. Torres, A. M. Fernández, A. Troncoso, and F. Martínez-Álvarez, "Deep learning-based approach for time series forecasting with application to electricity load," in *International Work-Conference on the Interplay Between Natural and Artificial Computation*, pp. 203–212, Springer, 2017.
- [15] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
- [16] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou, "Deep learning with a long short-term memory networks approach for rainfall-runoff simulation," *Water*, vol. 10, no. 11, p. 1543, 2018.
- [17] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
- [18] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [19] A. K. Palit and D. Popovic, *Computational intelligence in time series forecasting: theory and engineering applications*. Springer Science & Business Media, 2006.
- [20] Z. Tang, C. De Almeida, and P. A. Fishwick, "Time series forecasting using neural networks vs. box-jenkins methodology," *Simulation*, vol. 57, no. 5, pp. 303–310, 1991.
- [21] J. C. B. Gamboa, "Deep learning for time-series analysis," *arXiv preprint arXiv:1701.01887*, 2017.
- [22] L. Cao, "Support vector machines experts for time series forecasting," *Neurocomputing*, vol. 51, pp. 321–339, 2003.
- [23] Wikipedia contributors, "Autocorrelation — Wikipedia, the free encyclopedia." <https://en.wikipedia.org/w/index.php?title=Autocorrelation&oldid=1027440461>, 2021. [Online; accessed 8-July-2021].
- [24] P. J. Brockwell, "Time series," in *International Encyclopedia of Statistical Science* (M. Lovric, ed.), pp. 1601–1605, Springer, 2011.
- [25] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. Springer, 2000.
- [26] Wikipedia contributors, "Partial autocorrelation function — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Partial_autocorrelation_function&oldid=967803127, 2020. [Online; accessed 8-July-2021].
- [27] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.
- [28] O. Anderson, "The box-jenkins approach to time series analysis," *RAIRO-Operations Research*, vol. 11, no. 1, pp. 3–29, 1977.
- [29] A. C. Harvey and P. H. Todd, "Forecasting economic time series with structural and box-jenkins models: A case study," *Journal of Business & Economic Statistics*, vol. 1, no. 4, pp. 299–307, 1983.
- [30] S. Makridakis and M. Hibon, "Arma models and the box-jenkins methodology," *Journal of forecasting*, vol. 16, no. 3, pp. 147–163, 1997.
- [31] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, "Akaike information criterion statistics," *Dordrecht, The Netherlands: D. Reidel*, vol. 81, no. 10.5555, p. 26853, 1986.
- [32] H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.
- [33] J. E. Cavanaugh and A. A. Neath, "The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 11, no. 3, p. e1460, 2019.
- [34] A. E. Raftery, "Bayesian model selection in social research," *Sociological methodology*, pp. 111–163, 1995.
- [35] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461–464, 1978.
- [36] S. S. Chen and P. S. Gopalakrishnan, "Clustering via the bayesian information criterion with applications in speech recognition," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 2, pp. 645–648, IEEE, 1998.
- [37] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
- [38] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [39] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [40] V. N. Vapnik, "Methods of function estimation," in *The Nature of Statistical Learning Theory*, pp. 181–224, Springer, 2000.
- [41] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *International Conference on Artificial Neural Networks*, pp. 999–1004, Springer, 1997.
- [42] S. F. Crone, J. Guajardo, and R. Weber, "The impact of preprocessing on support vector regression and neural networks in time series prediction," in *DMIN*, pp. 37–44, 2006.
- [43] Y. Wang, B. Wang, and X. Zhang, "A new application of the support vector regression on the construction of financial conditions index to cpi prediction," *Procedia Computer Science*, vol. 9, pp. 1263–1272, 2012.
- [44] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [45] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
- [46] E. Scornet, G. Biau, J.-P. Vert, et al., "Consistency of random forests," *The Annals of Statistics*, vol. 43, no. 4, pp. 1716–1741, 2015.
- [47] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.
- [48] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.
- [49] "Official holidays."
- [50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [51] L. Sun, "Application and improvement of xgboost algorithm based on multiple parameter optimization strategy," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pp. 1822–1825, IEEE, 2020.
- [52] T. Fushiki, "Estimation of prediction error by using k-fold cross-validation," *Statistics and Computing*, vol. 21, no. 2, pp. 137–146, 2011.
- [53] Y. Jung and J. Hu, "Ak-fold averaging cross-validation procedure," *Journal of nonparametric statistics*, vol. 27, no. 2, pp. 167–179, 2015.
- [54] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, pp. 76516–76531, 2020.

- [55] M. Bakker, H. Van Duist, K. Van Schagen, J. Vreeburg, and L. Rietveld, "Improving the performance of water demand forecasting models by using weather input," *Procedia Engineering*, vol. 70, pp. 93–102, 2014.
- [56] M. Ghiassi, D. K. Zimbra, and H. Saidane, "Urban water demand forecasting with a dynamic artificial neural network model," *Journal of Water Resources Planning and Management*, vol. 134, no. 2, pp. 138–146, 2008.
- [57] P. Rajalakshmi and S. D. Mahalakshmi, "Iot based crop-field monitoring and irrigation automation," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–6, IEEE, 2016.
- [58] A. An, C. Chan, N. Shan, N. Cercone, and W. Ziarko, "Applying knowledge discovery to predict water-supply consumption," *IEEE Expert*, vol. 12, no. 4, pp. 72–78, 1997.