Making urban water smart: the SMART-WATER solution


ABSTRACT

The rise of Internet of Things (IoT), coupled with the advances in Artificial Intelligence technologies and cloud-based applications, have caused fundamental changes in the way societies behave. Enhanced connectivity and interactions between physical and cyber worlds create ‘smart’ solutions and applications to serve society’s needs. Water is a vital resource and its management is a critical issue. ICT achievements gradually deployed within the water industry provide an alternative, smart and novel way to improve water management efficiently. Contributing to this direction, we propose a unified framework for urban water management, exploiting state-of-the-art IoT solutions for remote telemetry and control of water consumption in combination with machine learning-based processes. The SMART-WATER platform aims to foster water utility companies by enhancing water management and decision-making processes, providing innovative solutions to consumers for smart water utilisation.

Key words | Internet of Things, LPWAN, predictive big data analytics, smart metering, telemetry, water management

HIGHLIGHTS

- Smart Automated Metering and Remote Control – design and deploy a novel integrated infrastructure that encompasses telemetry and remote-control technologies for the efficient management of the water supply network.
- Device Connectivity and Data Management – develop alternative technologies for the implementation of a fixed wireless network that will support real-time telemetry and remote-control services. The fixed network’s performance and reliability are tested and evaluated in real conditions in Thessaloniki’s urban environment.
- Data Processing and Visualisation – develop data analysis and visualization tools for complex event processing, identification of patterns behind periodic peak demands, forecasting water consumption and demand, assisting the water utility company to efficient water management and decision-making processes.
- Water Management – create new products for the consumers and the water utility company, assisting the former to build consumption patterns and raise their water consciousness and awareness, and the latter to sustainably manage water, including issues concerning consumption, demand, resources as well as to detect leaks etc.
Feedback of water usage – evaluate the overall infrastructure based on operational efficiency, usability and reliability, and consumer satisfaction.

INTRODUCTION

Factors such as climate change, escalating urban water demand, customer requirements and the integration of new technologies in customers’ service have all been increasing the onus on water providers to adopt more effective and sustainable approaches for urban water management. The combination of smart metering and Internet of Things (IoT) allows the development of sophisticated systems to better manage resources and develop new applications to facilitate various parts of society; that is, citizens/consumers, public and private organizations, businesses etc. In the water industry, this combination is the key to develop smart water management systems serving both consumers and water utility companies and fostering sustainability, through water leak detection (Saraswathi et al. 2018), river water quality monitoring in real-time (Chowdury et al. 2019), water flow monitoring (Kusuma & Anil 2018), short-term water consumption and water demand forecasting (Antunes et al. 2018; Benítez et al. 2019). Smart water systems are inherently connected with the concept of smart cities, since they comprise part of the city’s infrastructure. Future smart cities will face challenges of privacy and cybersecurity that should be taken under consideration to minimize potential risks and strengthen the important role of smart water in urban areas (Moy de Vitry et al. 2019).

The smart metering industry is rapidly evolving. The term ‘smart’ defines any new systems employing the latest in communication capabilities and enhanced functionalities (Boyle et al. 2013). There is a plethora of innovative smart water meters that frequently record and transmit metering data. The absence of an open and widely accepted communication standard in the smart water metering industry led most vendors to introduce proprietary solutions at protocol and platform level (Diehl Stiftung & Co. KG 2020; Sensus 2020) adding deployment complexity and resulting in vendor and protocol ‘lock-in’ issues (Alvisi et al. 2019).

Water management infrastructures typically make use of traditional ‘walk-by’ and ‘drive by’ methods, which do not allow frequent data collection and cannot support bidirectional communication (Cole et al. 2012). Real time telemetry can only be feasible via a fixed wireless network consisting of data collectors at fixed installation points. Designing and implementing such a network in urban areas is challenging. Short-range technologies like BLE (Gomez et al. 2012) and ZigBee (Gislason 2008) can no longer meet the growing demands of modern applications. Alternatives based on cellular networks (2G/3G/4G) provide wider coverage with the expense of cost. Till now, the majority of Measuring Instruments Directive (M.I.D.) certified water meters by top-notch vendors adopt short range protocols, i.e. wM-Bus (EN 13757-4) (Carratù et al. 2017), which hinder large scale deployments due to the large number of required gateways. Low Power Wide Area networks (LPWAN) technologies (Sanchez & Cano 2016) provide increased connectivity range (>5 km), low power consumption and low infrastructural costs. The main LPWAN technologies used are: Sigfox (Sigfox 2016), LoRa (Semtech 2013) and NarrowBand-IoT (NB-IoT) (3GPP 2016).

Intelligent water metering effectively generates a big volume of data streams that can be stored into the water utility providers’ information systems. However, the need for real-time monitoring as well as for advanced automated reporting tools and predictive analytical processes have emerged. The reliable and accurate short-term water consumption and demand predictions can significantly improve water management, thus bringing significant financial savings to a water utility company. Hence, efficient demand forecasting facilitates on one hand identification of water network failure or water loss and on the other effective scheduling of the utilisation of the energy-related operations for treatment and pumping during the low-cost periods, etc. (Anele et al. 2017; Antunes et al. 2018; Guancheng & Shuming 2018; Xenochristou et al. 2018; Benítez et al. 2019; Oyebode & Ighravwe 2019). In the last decades, different linear and nonlinear approaches for short-term water demand forecasting have been proposed in the literature. Two well-known representatives of the first category are the Auto-Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA) approaches, which have been used in water demand time series modeling (Donkor et al. 2014; Kofinas et al. 2014; Antunes et al. 2018; Anele et al. 2017). The nonlinear ones aim to fit machine learning techniques to model the intrinsic nonlinearity into the water demand data and predict the time series behaviour, increasing the forecasting accuracy. A variety of studies have
explored the application of machine learning methods such as Artificial Neural Networks, Support Vector Machines for Regression (SVR), Random Forests, Deep Neural Networks etc. in short-term water demand forecasting or even hybrid models that combine one or more methods (Kofinas et al. 2014; Walker et al. 2015; Antunes et al. 2018; Guancheng & Shuming 2018; Vijai & Sivakumar 2018; Xenochristou et al. 2018; Oyebode & Ighravwe 2019).

In the framework of the SMART-WATER (Mourtzios et al. 2019) research project, a unified system consisting of smart water meters coupled with smart water valves and data analysis processes were utilized in the urban area of Thessaloniki, Greece. The proposed innovative combination makes its application an additional challenge.

In the following sections, a brief reference to related projects is exhibited and is followed with a short description of the SMART-WATER project. Then, the methodological framework, as well as the main methods and approaches, are mentioned in detail. Finally, the preliminary results of the project will be discussed in the last section of this work.

**RELATED WORK**

During the last decade, realizing the need to improve water management resources along with the achievements of evolving IoT technology, significant progress has been made and various projects have come up in the field of water, aiming to address the arising challenges. In the study of Tsavdaridou et al. (2012), Automated Meter Reading (AMR) was utilized with positive displacement water meters to detect leakages experimentally in Thessaloniki, in cooperation with the city’s water utility company (EYATH S.A.).

In the IceWater project, which is presented in Fantozzi et al. (2014), focusing on improving energy efficiency and leak detection, ICT solutions were utilized. The possibilities offered by using AMR systems to collect data at the consumers’ level were taken into account in Candelieri & Archetti (2014). In this work, an approach for water demand forecasting on an hourly basis was developed, aiming at optimizing operations of the water network in order to reduce energy costs.

Another project, the iWIDGET project (Makropoulos et al. 2014) proposed two web-based platforms developed mainly for consumers: (a) a web-based platform providing consumers with information for domestic water consumption accessible through the respective platform (Kossieris et al. 2014a). This web tool made use of time series charts helping consumers gain deeper insight into their consumption profile and past water consumption data, (b) an educational e-Learning platform, which was presented in Kossieris et al. (2014b), providing various applications, aiming to motivate and help consumers improve their water consumption profile. The iWIDGET project aimed at developing a collaboration between consumers and water utilities, exploring huge data sets from smart water meters (Ribeiro et al. 2015).

In the WATERNOMICS project, a system infrastructure and platform were developed using ICT technologies for water resource management aiming at increasing water efficiency. It proposed three (3) water consumption levels, namely domestic, corporate and municipal. The platform was the main tool to achieve improvements in water consumption behaviour (Perfido et al. 2015; Kouroupetroglou et al. 2015). Another smart water metering approach is presented in Mudumbe & Abu-Mahfouz 2015, including a web-based visualization tool for real-time and historical water consumption data. In Singapore, a Smart Water Grid is intended by its Public Utility Board for real-time monitoring of pressure and water quality throughout the network as well as for real-time water consumption with the use of Automated Meter Reading (AMR) (PUB 2016). Furthermore, the SmartH2O platform was developed (Novak et al. 2018; Rizzoli et al. 2018) aiming at water consumption reduction through behavioural change in consumers’ water consumption. It was based on the combination of smart metering data and visualization tools contributing to creating feedback between the consumers and the water utility sector.

**SMART-WATER PROJECT**

SMART-WATER is an ongoing pilot project that puts forward a novel design of an infrastructure that utilizes modern telemetry and remote-control technologies to provide innovative services to consumers and water utility companies. Custom-made gateways constitute the backbone of a fixed wireless and multiprotocol network that makes real time telemetry and remote control viable. The gateway functions overcome ‘lock-in’ restrictions since they can interface with any wM-Bus and LoRa sensors. SMART-WATER’s platform can decrypt and manage end device data from water valves and a wide range of different vendor water meters. State-of-the-art data analytics tools and techniques from the fields of statistics and Machine Learning are applied to predict consumption and identify patterns behind periodic peak demands. Finally, two web-based platform applications are developed to provide end users, consumers and water utility companies with real time telemetry and remote-control services. Specifically,
the goals of the SMART-WATER project can be summarised according to the five directions, as illustrated in the Figure 1. These goals have been contextualized in a five-layer unified generic architecture (Figure 2). The proposed platform supports the pipeline process from automatic metering of water consumption to the water management and decision support process by the water utility company as well as by the consumers.

**End Device Layer:** The primary goals of SMART-WATER project are the frequent recording and wireless transmission of water consumption (smart automated telemetry) data combined with remote control of water supply in almost real time.

**Device Connectivity Layer:** This layer is in charge of the bidirectional connection between end devices and the central server infrastructure using multiple wireless telecommunication protocols.

**Device and Data Management Layer:** At the 3rd level of the network architecture, the reception, decoding, decryption and storage of the data transmitted by the water meters, as well as the management of the actuation commands to the valves, take place.

**Data Processing and Analytics Layer:** In this level, the storage and processing of the metering data is realised. This layer employs state-of-the-art machine learning techniques to perform predictive analysis on water consumption and demand, identify patterns and trends in time series data, and detect abnormal water consumption behaviour of consumers.

**User Interaction Layer:** The system’s end-users are water utility company authorized personnel and consumers. Two different User Interfaces (U.I.) were designed and developed covering each group’s needs for real time water consumption monitoring and water supply control.

The main contribution of this work is to focus on the backbone aspects of the SMART-WATER project, by illustrating the proposed approaches, in each one of the above layers, which are able to tackle the aforementioned limitations and challenges. Specifically, in this work, a holistic framework is proposed that encapsulates innovative smart metering and remote-control services in real time, designs and deploys a novel fixed wireless network to connect the smart devices with the central infrastructure and develops state-of-the-art methods to manage and analyse the obtained data, aiming to assist the water utility company to achieve efficient water management and provide high quality services to consumers.

**METHODS**

In this section, the methods, tools and techniques that were designed, implemented and applied in the field through the whole life span of the project are exhibited thoroughly, by following the ordering of the architecture layers.
End device layer

The Sensors layer is the source of data to the SMART-WATER system infrastructure. Smart water meters consist a novel metering technology. Built-in communication systems enable efficient remote reading while sophisticated software alerts effectively for leak detection, bursts or other irregularities such as tampering attempts and reverse flows. The
prerequisite features they should demonstrate to constitute the building blocks of a water management infrastructure are M.I.D. certification, high metering accuracy (ultrasound or electromagnetic), wirelessly transmitting metering data at dense regular intervals and ultra-low consumption (10+ years autonomy). Based on the aforementioned features, water meters from five different vendors were chosen for the pilot testing in order to assess their metering and telecommunication performance. Water meters use wM-Bus or LoRa telecommunication protocols.

The second category of end devices are the smart wireless water valves, which make it possible to control water supply and comprise one of the main innovative features of the proposed system. The wireless valves should exhibit certain features: remotely controlled via a wireless and bidirectional telecommunication protocol, ultra-low consumption (10 years autonomy) and tamper alert. Telecommunications-wise, water valves use LoRaWAN protocol.

Device connectivity layer

Multi-Protocol gateways (MPG)

An important challenge tackled during the project was the lack of any commercially available gateway that can simultaneously communicate bidirectionally with wM-Bus and LoRa end devices and at the same time establish bidirectional communication with the central infrastructure. In the literature, there have been proposed IoT gateways that utilize multiple protocols. Amiruddin et al. (2018) proposed a gateway that could interface with Zigbee και BLE sensors while Kaur & Singh 2015 and Gunasagarana et al. 2015 proposed similar telecommunication nodes that could communicate with RF, Zigbee and Bluetooth sensors.

Water meters and water valves operate on different telecommunication protocols. An additional problem is that the two protocols used by end devices, namely wM-Bus and LoRaWAN, are not internet protocols, which does not allow end devices to communicate directly to the server. All the above issues led us to design and manufacture an electronic device that: (a) communicates bidirectionally and simultaneously with wM-Bus and LoRaWAN technology devices and (b) communicates bidirectionally via at least one internet protocol with the central server. The Printed Circuit Board (PCB) (Figure 3), together with the electronic components (modules/boards), comprise the multi-protocol gateway, which is placed in an IP55 plastic enclosure. The breakout boards are placed on the female pinheads, allowing the user to easily add-drop them depending on the connectivity scenario. The gateway uses Raspberry Pi Zero W (1 GHz, single-core CPU, 512MB RAM) as its main processing unit.

The Multi-Protocol Gateway performs numerous functions simultaneously. Firstly, it collects wM-Bus telegrams

---

**Figure 3 |** PCB with all modules attached on female pinheads. Dimensions: 22 cm(L) × 14 cm (W) × 5 cm(H).
(T, S, C modes) transmitted by wM-Bus water meters within range. Secondly, it communicates bidirectionally with LoRaWAN end devices. To overcome the issue that the telecommunications protocols used by end devices are not IP protocols, a GSM module was integrated in each gateway to allow two-way communication with the server. The gateways use NB-IoT network as a secondary two-way protocol in premises where there is limited GSM coverage or as a fail-over solution in cases where GSM outage take place.

Hybrid fixed network

One of the most challenging tasks of the SMART-WATER project was the design and implementation of the bidirectional and wireless fixed network, which is necessary for real time telemetry and remote-control services. The dense urban environment and reinforced structural materials in Greek cities, combined with the limitations of RF technologies, made network design and deployment more complicated. Figure 4 depicts the architecture of the fixed wireless network, realized through different connectivity scenarios. SMART-WATER's hybrid network makes use of multiple telecommunication protocols, both legacy (wM-Bus, GSM) and LPWAN (LoRa, NB-IoT), employing different gateway types; for example, off-the-shelf outdoor LoRaWAN Gateways, wM-Bus-to-LoRa Bridges, LoRa repeaters (the last two are realized through MPG), to ensure the optimal connectivity conditions and minimum packet loss.

The gateway’s installation points need to be carefully selected in terms of connectivity requirements and cost constraints. In this direction, a custom ‘tester’ device was designed and assembled in order to perform on site connectivity tests, which defined the optimal gateway installation point inside consumer’s premises and verified the connectivity scenarios described in Figure 4.

Device and data management layer

At the 3rd level of the proposed system architecture, the reception, decoding, decryption and storage of metering data, as well as the management of the downlink commands, take place. The server unit can simultaneously process and decode/decrypt payload data from water meters and water valves. This level hosts three main services: (a) end device management, (b) gateway monitoring and (c) LoRa and wM-Bus servers.

In the proposed architecture (Figure 2), the distributed layers from end devices to UIs are depicted. An MQTT
Broker (Mosquito Broker) has been used in each distributed unit of the system and an AMQP (Advanced Message Queuing Protocol) Broker (RabbitMQ Broker) in the central server. In addition, the IoT Core tool has been added to the system, a managed cloud service provided by Amazon Web Services (AWS), used to synchronize all MQTT Brokers in the system. Two cloud-based Linux computers are used through Amazon’s online services (AWS) and specifically through the Elastic Compute Cloud (EC2) service. The server uses a noSQL database (MongoDB) both for end device management and for data storage. Server’s API (Application Programming Interface) is a RESTful API, with token-based authentication which was developed to support consumer’s and administrator’s applications and provide them with the necessary data and metadata originating from the network’s devices (end devices and gateways).

Device management

Given that the end devices are remotely installed, it is crucial to develop a system that allows their monitoring and registration. The ‘Device Management System’ allows the network administrator to register new end devices with all necessary information (e.g. ‘Device id’, ‘Customer id’, ‘decryption keys’, ‘installation address’, ‘coordinates’).

Gateway monitoring

MPGs are remotely controlled and monitored. Virtual Private Network (VPN) technology is used for remote management and monitoring of all gateways. The user can perform routine debugging actions, software updates, system monitoring etc. The gateways additionally transmit frequent ‘heartbeat’ uplink messages to the server through the packet forwarder services.

Lora server/ WM-Bus server

When an uplink message from a LoRa end device reaches the central server, it is directed to the ‘LoRaWAN server’ as shown in Figure 2. The LoRa server is an open source application (ChirpStack 2017) developed according to the LoRaWAN protocol in the Go programming language. ‘wMBus Server’ is a custom application developed for SMART-WATER. wM-Bus messages are directly processed for the application of machine learning models, wM-Bus server was developed in the Python programming language.

Data processing and analytics layer

The main objective of this layer is to analyse the acquired time-series data in order to make short-term predictions of the water consumption in a specific urban region of interest. To achieve this goal, various methodologies from statistics and machine learning fields are implemented and evaluated. The proposed SMART-WATER analytical approach consists of the following seven (7) steps:

1. **Data acquisition**: the water consumption data for specific time period and region of interest will be acquired from the SMART-WATER data storage.

2. **Visualise the Time Series**: visualising the insight characteristics in the time series, such as the trend, seasonality, existence of outliers etc., is an essential step prior to deeper analysis and building a more refined model. The SMART-WATER dashboard aims towards this direction, providing filtering, searching and zooming functionalities to easily explore time-series data.

3. **Preprocessing**: Depending on the dataset, the steps of preprocessing will be defined, including the creation of timestamps, the detection and elimination of outliers, the stationary test, the normalisation process etc.
   i. **Outlier detection and removal**: Outliers are observations that are very different from the majority of the observations in the time series (Hyndman & Athanasopoulos 2018). They may be errors, due to hardware issues, sometimes extremely high values of water consumption appear, or they may simply be unusual. These values should be removed because the majority of analytical techniques are not able to deal with them efficiently. A common and simple way to detect outliers is using the z-score method.
   ii. **Test for Stationary Time Series**: A well-known test for stationarity is the Augmented Dickey Fuller Test (ADF) which is a form of Hypothesis Testing. The null hypothesis of the ADF test is that the time series is non-stationary. If the ADF Test Statistic is less than the Critical Value, the null hypothesis should be rejected, concluding that the series is stationary and it is expected with a very high probability that its previous behaviour will be continued in the future.
   iii. **Normalization**: Data normalization involves converting time series data into a given range. It is needed for the application of machine learning models,
4. **Choose the input data features**: a crucial step for the quality of the forecasting process is the right choice of input features, especially during the creation of a model. Each feature represents an input variable that highly affects the outcome of the forecast (Benitez et al. 2019). In the SMART-WATER framework, it is considered that previous water consumption observations affect the short-term water demand forecast. Although all the range of data has to be utilised during the training phase, only a small portion has a direct influence on each step of the forecasting process. Thus, the predicted water consumption for the timestamp \( T + 1 \), \( WCF(T + 1) \), (e.g. day, hour etc.) can be predicted by employing the past observations \( O_{T-n}, \ldots, O_T \) in the timestamps \( (T-n, \ldots, T) \), namely \( WCF(T + 1) = M(O_T, O_{T-1}, \ldots, O_{T-n}) \), where \( M \) is the model.

5. **Training phase**: The main goal of this step is the creation of a forecasting model that will enable correct prediction of the future water consumption. Thus, the appropriate method should be chosen along with splitting the time-series dataset for training and testing purposes.

   i. **Develop base-line approach**: create/train ARIMA and SARIMA models; namely, the appropriate models’ parameters (p, d, q, P, D, Q, m) should be determined.

   ii. **Develop machine learning approaches**: create/train machine learning models such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Long Short-Term Memory (LSTM) Neural Networks.

6. **Testing phase**: The trained model is evaluated in terms of its ability to correctly predict future water consumption. The testing set is employed and its performance is estimated using suitable metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and Coefficient of determination \( (R^2) \).

7. **Compare the results and choose the most accurate model**: the best model is chosen in terms of the performance indicators over the testing set. It should be noted that better forecasting models are extracted as the \( MAPE \) and \( RMSE \) values tend to zero (0) and the \( R^2 \) values approach unity (1).

The proposed schema is easily customised and self-learning as it has been designed and developed to be applicable both at the individual consumers’ level as well as for clusters of consumers with similar water consumption patterns. Furthermore, it is completely data-driven and independent of the machine learning model employed for forecasting. In this work, the ARIMA and LSTM methods are employed and their forecasting performance evaluated. A brief mathematical formalization of these methods is presented as follows.

**Auto-Regressive Integrated with Moving Average Model (ARIMA)** is an efficient approach for short-term forecasting (Hyndman & Athanasopoulos 2018). Usually, it denotes as \( ARIMA(p, d, q) \) model, where \( p \) is the order of Autoregression (AR) terms (number of time-step lags to be used), which represents the relation between past and current observations, \( d \) is the order of differencing for attaining stationarity (integrated) and \( q \) is the order of Moving Average (MA) terms, which means the size of the MA window. In the ARIMA model, the forecast value is a linear combination of past observations (linear combination lags of \( y \) up to \( p \)) and past errors (linear combination of lagged forecast errors, up to \( q \)), expressed as follows:

\[
y_t = c + \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t
\]

For the estimation of the \( p, d \) and \( q \) parameters, an iterative strategy is applied consisting of the identification step, where a sub-class of models that might better represent the data is chosen, the estimation step, where the parameters of the model are trained using the historical data and finally the diagnostic checking, where the trained model is validated using a disposal dataset.

**LSTM stands for Long Short-Term Memory**: Neural Network is an enhancement of the Recurrent Neural Network (RNN) in relation to their ability to capture the long-term dependencies (Hochreiter & Schmidhuber 1997). Specifically, RNNs use previous information in the present task allowing historical information to be stored in the network’s internal state and mapping them to the final output. RNNs suffer from short term memory due to the vanishing gradient problem (Masum et al. 2018; Hua et al. 2019). Alternatively, the LSTMs overcome this limitation as they are equipped with unique ‘Gates’ enabled to avoid the long terms for vanishing (Figure 5).

Particularly, an LSTM memory block is a core component that encompasses a memory cell and gates, namely an Input Gate, an Output Gate and a Forget Gate. It computes the mapping from an input sequence \( x = (x_1, \ldots, x_t) \) to an output sequence \( y = (y_1, \ldots, y_t) \). The following steps are executed inside the memory cell:
1. In the **Forget Gate**, the input sequence from the current input \( x_t \), is combined with the previous hidden state \( h_{t-1} \), through the sigmoid function \( \sigma(x) = \frac{1}{1 + e^{-x}} \), which returns values in the range between 0 and 1. Hence,

\[
ft = \sigma(WFx_t + Ufh_{t-1} + b_f)
\]

The closer to 0 means to forget, and the closer to 1 means to keep. In this way, the Forget Gate assists in removing information from the cell state.

2. The **Input Gate** aims to store new information into the memory cell state. Initially, a sigmoid layer determines what will be updated via the formula:

\[
i_t = \sigma(Wix_t + Uih_{t-1} + b_i)
\]

Secondly, the input will be updated by employing the tanh layer, by creating a vector of new candidate values \( z_t \) that can be added to the memory cell state, as stated in the following equation:

\[
z_t = \phi(W_zx_t + U_zh_{t-1} + b_z)
\]

where \( \phi(\cdot) \) denotes the tanh function, \( \phi(x) = 2\sigma(2x) - 1 \). The values fluctuate between -1 and 1.

3. The new state of the memory cell, \( c_t \), should be updated. The estimated values, \( f_t, i_t, z_t \), are fused along with the old memory cell state \( c_{t-1} \). Specifically:

\[
c_t = i_t \times z_t + f_t \times c_{t-1}
\]

4. The output of the memory cell is realised by element-wise multiplication between the value obtained from a tanh function of \( c_t \) and the output of a sigmoid layer, i.e., the Output Gate activation vector \( o_t \), as follows:

\[
\begin{align*}
of_t &= \sigma(W_ox_t + U_oc_{t-1} + V_oc_t + b_o) \\
h_t &= o_t \times \text{tanh}(c_t)
\end{align*}
\]

The weight matrices \( W_f, W_i, W_z, W_o, U_f, U_i, U_z, U_o \) and \( V_o \) along with the bias vectors \( b_f, b_i, b_z, b_o \) can be learnt in the training stage estimated. Through the cooperation between the memory cell and the gates, LSTM is endowed with a powerful ability to predict time series with long-term dependences (Hochreiter & Schmidhuber 1997; Masum et al. 2018; Hua et al. 2019).

**User interaction layer**

**Water utility company (administrator) Web application**

The **Main Dashboard** (Figure 6) is the cornerstone component of the Water Utility Company Web Application, as it encompasses crucial functionalities enabling the Water Utility organisation to improve its daily operations. Therefore, end devices are mapped based on their geographical location. Furthermore, it enables the operators to gain a general glance and overview of the water consumption by region of interest as well as monitor hourly water consumption in near ‘real-time’.
The **Single Consumer Dashboard** (Figure 7) enables the practitioners to monitor the water consumption of a specific consumer in a particular time horizon, to compare his/her consumption with personal past or average consumptions of the consumer’s region. The interactive visualisations fuel the operators with functionalities to navigate in specific time periods, aggregate and illustrate time series data hourly, every six hours, daily and monthly.

The **Alert Dashboard** (Figure 8) notifies in a concise way the alerts that are received from the end devices. The alerts are categorized into 14 classes, each one indicating a specific malfunction of the end device.

**Consumer’s Web application**

The consumer’s application is a web based and responsive application. After logging in (Figure 9), the consumer is
directed to the consumer **Main Homepage** (Figure 10) in case he/she owns more than one pair of end devices. Otherwise, he/she is directed to the **Main Dashboard** (Figure 11) where he/she can track water consumption by hour/day/week/month and actuate on water valves. Information about former bills is also available and a prediction based on current consumption, for the next billing is also depicted (Figure 10).

**RESULTS**

This section summarizes findings based on pilot tests in real conditions. Having installed the majority of end devices (75 water meters and valves out of 100) and MPGs, useful conclusions were drawn regarding SMART-WATER system’s infrastructure. The system underwent rigorous stress tests to validate its stability across all architecture layers and the ability to serve multiple users simultaneously. Furthermore, the data analysis processes for short-term forecasting of water consumption in the region of interest were assessed in terms of their accuracy.

**Time performance**

The server infrastructure was initially assessed in terms of its API performance and secondly for the latency of data acquisition. In the SMART-WATER case, users are consumers and water utility company’s administrators who access the respective web applications. It is of prime importance to design and develop a reliable and robust server system which can respond to multiple user calls simultaneously. In that direction, the server’s API was tested using performance tool Apache Jmeter ([https://jmeter.apache.org/](https://jmeter.apache.org/)). Several critical scenarios were simulated in order to extract **Response Time** (ms) (amount of time needed to complete a call, Wescott 2013) and **Throughput** (kB/s) (number of calls per second, Wescott 2013) as key performance metrics for the API. The API’s performance was assessed in different cases realizing 12 scenarios. The scenarios entail both GET and POST requests spanning from simple ‘Login’ requests to more demanding ones. For example, Scenario 10 describes the case of 200 (virtual) users doing consecutive calls (6 GET calls) for 200 seconds in the Consumers’ application home page. The description of the scenarios, including their specific features as well as the response times and throughput simulated results per scenario, are listed in Table 1.

It is evident that as the number of users increase, a decrease in throughput is observed as expected due to the additional load created. Scenarios 1, 2 and 3 exhibit the lowest response time (<100 ms) while Scenario 11 the highest one exceeding 1.8 sec. Particularly, it represents the most challenging test case, which is also reflected in the lowest throughput (53 kB/s). Scenario 1 exhibits the highest...
simulated throughput reaching 103 kB/s (Figure 12). The results are more than acceptable, since scenarios describe challenging cases hardly possible to occur with real users which means that the API server is designed to serve numerous users at the same time without any significant delay.

Secondly, data acquisition times were recorded to evaluate the system’s processing speed and traffic load management. The data acquisition pipeline was divided into three fundamental stages. Each stage is assigned to a respective time slot, namely ‘Gateway to Server’ stage (time for a message to reach the server upon reception at the gateway), ‘Processing’ stage (time for decoding, decryption, authentication processes after the message reaches the server) and ‘Persist’ stage (time for data aggregation). The total statistical sample was 3,000 received uplink messages from water meters installed in consumer’s premises. The total simulation time was 2.8 days. The time data points of the three stages are presented in respective box plots (Figure 13). For the ‘Gateway to server’ plot, the median time (50% percentile) value is 0.87 s (orange line in the box) while the inter-quartile range (IQR) of the distribution extends from 0.66 s (percentile 25%) to 0.99 s (percentile 75%). For the second stage of the data acquisition pipeline, ‘Processing’, the respective values are 0.00312 s (percentile 25%), 0.00319 s (median), and 0.00329 s (percentile 75%). Finally, the statistics for the ‘Persist’ time samples are 0.0334 s (percentile 25%), 0.034 s (percentile 50%), and 0.036 s (percentile 75%).

Figure 9 | SMART-WATER Consumer’s web application: Log-in page.
The longest timeslot among the three stages is the ‘Gateway to server’ case, which also exhibits the greatest deviation. Over-the-internet transmission introduces the longest time delay, due to the limited connectivity conditions at the gateways’ installation points. The remarkable fluctuation in time values is related to RF
propagation issues introduced by the dense urban environment. The time periods of the other two stages are considerably lower and the respective deviations denote the merest fluctuation. This stability implies that the system can handle all traffic generated by the end devices without delay (near real time).

Moreover, *Runtime Performance*, one of Google Chrome’s Developer Tools, was employed to effectively evaluate the response times for the two web applications (the water utility company’s and the consumer’s). More specifically, the above tool describes and analyzes the way that a website appears and loads. In order to measure the response time of the consumer application, the browsing conditions on a mobile device should be simulated. Therefore, 4xCPU slow down for throttling and the Fast 3G for the network were selected.

### Table 1

<table>
<thead>
<tr>
<th>Scenario No.</th>
<th>Scenario Description</th>
<th>GET/POST</th>
<th>No. Users</th>
<th>Duration (sec.)</th>
<th>Response time (ms)</th>
<th>Throughput (kB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>Consumer App home page</td>
<td>6 GET</td>
<td>10</td>
<td>60</td>
<td>96</td>
<td>103</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>Login request</td>
<td>1 POST</td>
<td>10</td>
<td>60</td>
<td>98</td>
<td>102</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>Valve request</td>
<td>1 POST</td>
<td>10</td>
<td>60</td>
<td>93</td>
<td>99</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>Consumer App home page</td>
<td>6 GET</td>
<td>10</td>
<td>60</td>
<td>581</td>
<td>77</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>Login request</td>
<td>1 POST</td>
<td>10</td>
<td>60</td>
<td>578</td>
<td>78</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>Valve request</td>
<td>1 POST</td>
<td>10</td>
<td>60</td>
<td>571</td>
<td>79</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>Consumer App home page</td>
<td>6 GET</td>
<td>10</td>
<td>120</td>
<td>848.8</td>
<td>69</td>
</tr>
<tr>
<td>Scenario 8</td>
<td>Login request</td>
<td>1 POST</td>
<td>10</td>
<td>120</td>
<td>798</td>
<td>69</td>
</tr>
<tr>
<td>Scenario 9</td>
<td>Valve request</td>
<td>1 POST</td>
<td>10</td>
<td>120</td>
<td>917</td>
<td>62</td>
</tr>
<tr>
<td>Scenario 10</td>
<td>Consumer App home page</td>
<td>6 GET</td>
<td>200</td>
<td>200</td>
<td>1,752</td>
<td>57</td>
</tr>
<tr>
<td>Scenario 11</td>
<td>Login request</td>
<td>1 POST</td>
<td>200</td>
<td>200</td>
<td>1,869</td>
<td>53</td>
</tr>
<tr>
<td>Scenario 12</td>
<td>Valve request</td>
<td>1 POST</td>
<td>200</td>
<td>200</td>
<td>1,749</td>
<td>58</td>
</tr>
</tbody>
</table>

![API performance per number of users and scenario process](image)

**Figure 12** | Response time (ms) and throughput (kB/s) estimations using the Apache Jmeter performance tool, for each of the simulated test scenarios and the number of users.
Runtime performance analysis is categorized into four main groups of functionalities: Scripting, Rendering, Painting and Loading. Scripting is the time for JavaScript analysis and evaluation. Rendering and Painting are related to HTML compilation and CSS editing on objects that can be displayed on the user’s screen. Finally, Loading refers to the interaction of the website with the network as well as the loading of HTML, CSS and Javascript (https://github.com/Google-Chrome/devtools-docs/blob/master/docs/timeline.md).

In Figure 14(a), Response Time for the water utility company’s web application main components, measured over the group of functionalities. As expected, scripting is a significantly time-consuming operation in all the main components and it is followed by Rendering. Similarly, the Response Time of the main components of the consumer application is presented in Figure 14(b). The consumer’s application includes the Consumer Homepage and Consumer Dashboard, consisting of the billing information page, water consumption and water valve actuation option.

Experiments and results of water consumption forecasting

Multi-step water consumption forecasting is performed using ARIMA and LSTM models. The goal is to compare the performance of the models over their ability to make accurate short-term predictions of total water consumption.

The available set of data is related to the water consumption, corresponding to the period between 2020-02-01 00:00:00 and 2020-05-18 23:00:00 local time. The collected data were provided in flow rates measured in an hourly frequency from 75 AMRs (total: 2,587 hourly observations). Using the z-score approach, 30 outliers were detected and their values were substituted by propagating the last known observation each time. The average consumption is around 263.75 l/h and the maximum consumption ranged to 1,209 l/h. The distribution of the time series water consumption data grouped by days and hours exhibits high variability over the specific period (Figure 15).

The stationary ADF test exhibits that the time series is stationary as its value (96.387) is less than the critical values 25, 50, and 75% (96, 209, and 371.5 respectively). The p-value is 0.002 less than 0.05, which indicates to reject the null hypothesis (H0), the data does not have a unit root and the time series is stationary. This is rational, as the limited size of the dataset did not imply any trend or seasonality.

In order to fit forecasting models for water consumption, the selected dataset needs to be transformed. The data should be re-scaled to values between 0 and 1 as described above. Applying the LSTM model, it is required that the data be...
within the scale of the activation function of the model. Then, the time series dataset was divided into input and output sub-ssets enabling supervised learning to be undertaken. The past \((T-n)\) time steps were used as input in order to predict the water consumption in the time step \(T+1\). Furthermore, the time series data was divided into 85.30\% (2,155 hourly observations) for training and the remaining 16.70\% (432 hourly observations) for validation/testing the models’ performance in terms of the RMSE, MAPE and \(R^2\) evaluation measures.

A series of experiments were realised to fine tune the training parameters of the LSTM network, namely the epochs, the number of neurons and the activation function. The RMSE is employed to evaluate the performance of the LSTM network for each set of parameters over the testing set. The best set of parameters, minimizing the RMSE, is \{\text{Epochs} = 50, \#\text{neurons} = 4, \text{activation function} = \text{‘sigmoid’}\}. The performance of the LSTM network in terms of the RMSE over the training and testing set after 10 iterations fluctuated from the training set from 129.15 to 135.59 with mean \pm standard deviation value equal to 130.96 \pm 1.761 and for the testing set from 161.83 to 177.08 with mean \pm standard deviation value equal to 166.55 \pm 4.668. Similarly, the \(R^2\) fluctuates from 53\% to 58\% in the training set (mean \pm standard deviation = 56.40\% \pm 1.35) and in the testing set from 57\% to 64\% (mean \pm standard deviation = 62.20\% \pm 2.15). Finally, the minimum MAPE value is 45.19\% for training and 58.52\% for testing and the maximum values are 65.17\% and 50.21\% respectively. The average MAPE is 54.59\% \pm 7.186 for training and 43.23\% \pm 4.401 for testing.

In the following figure (Figure 16), the performance of the LSTM network in the training and testing sets is exhibited in terms of its ability to learn the water consumption pattern hidden in the time series data (training) as well as to forecast the hourly water consumption in a forecasting horizon of the 18 days.

In order to compare the performance of the LSTM model with an ARIMA approach, we need to select an optimal time series model determining the appropriate set of parameters.
parameters \((p, d, q)\). Using the `auto_arima` function from the python `pmdarima` package, the ARIMA\((6,0,0)\) model attains to minimise the Akaike information criterion (AIC). However, the performance of this model over the testing set is poor, as the RMSE was around 321.17 and MAPE climbed to 82.28%. Hence, the LSTM model seems to outperform the baseline models such as ARIMA in this set of time series data (Figure 17).

**CONCLUSIONS**

In this work, the main aspects of the on-going research project, called SMART-WATER, is presented. Specifically, SMART-WATER is a holistic water management solution that aims to offer innovative services to consumers and water utility companies. Pilot tests of the SMART-WATER system took place in the wider area of Thessaloniki, Greece. Real time telemetry and remote control of water supply are feasible utilizing state-of-the-art end devices on a fixed bidirectional network. High precision, wireless and M.I.D. certified water meters are used in conjunction with wireless water valves. Real time telemetry and remote-control services are viable through a fixed wireless and bidirectional network, which can manage messages from multiple protocols and standards (wM-Bus, LoRaWAN, NB-IoT, GSM). One of the key elements of the proposed system are the custom-made multi-protocol gateways (MPG), which unlock protocol and vendor binding issues. They can simultaneously interface with a wide set of end devices (water meters and water valves) of different vendors and protocols, providing bidirectional connectivity. Given the limited indoor penetration of the wM-Bus protocol, the gateways need to be located close to the end devices but at the same time remain connected to the central infrastructure via GSM or NB-IoT. Field tests also denoted that LoRaWAN technology is significantly superior to legacy wM-Bus technology regarding connectivity range.

At the other side of the architecture, a universal server can decode and decrypt wireless messages from subscribed wM-Bus and LoRa end devices regardless of the internet protocol used. The performance of the server and its API was tested in terms of processing time, response time and throughput. Simulation campaigns from real metering data demonstrated excellent data processing time upon gateway reception to aggregation (less than a second on average). Response time and throughput for different scenarios render the infrastructure able to serve numerous users simultaneously. As expected, a larger number of users introduces increased response time and moderate throughput values.

The short-term water demand forecasting approach adopts a seven-step schema, from the acquisition of the time-series data to the evaluation of the forecasting performance. The approach has been already tested on real data retrieved from the end devices installed in the field. Although the preliminary results for the operation of the

![Figure 17](image-url) | Performance comparison between LSTM and ARIMA \((6,0,0)\) models.
system are encouraging, however, further work should be done in order to evaluate its performance in the entire pipeline. Especially, in the advanced analytical perspective, as all the end devices will be deployed and start to collect data seamlessly, more robust, statistically sound and safe conclusions will emerge in terms of the short-term water demand/consumption estimations. Furthermore, clustering techniques and forecasting water consumption models will be applied, aiming at the determination of consumption profiles of the consumers.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: TIEDK-04337).

DATA AVAILABILITY STATEMENT

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

REFERENCES


Semtech Corporation. 2015 AN1200.22 LoRaTM Modulation Basics Application Note Revision 2 (accessed 10/03/2020).


