Tuning Hyperparameters of a SVM-based Water Demand Forecasting System through Parallel Global Optimization

Antonio Candelieri[1], Ilaria Giordani[1], Francesco Archetti[1]

Konstantin Barkalov[2], Iosif Meyerov[2], Alexey Polovinkin[2], Alexander Sysoyev[2], Nikolai Zolotykh[2]

[1] *University of Milano-Bicocca, Dept. of Computer Science, Systems and Communication, Milan, Italy*

[2] *Lobachevsky State University of Nizhni Novgorod*

Abstract. Recently, the number of machine learning based water demand forecasting solutions has been significantly increasing. Different case studies have already reported practical results proving that accurate forecasts may support optimization of operations in Water Distribution Networks (WDN). However, tuning the hyper-parameters of machine leaning algorithms is still an open problem.

This paper proposes a parallel global optimization model to optimize the hyperparameters of Support Vector Machine (SVM) regression trained to provide accurate water demand forecasts in the short-time horizon (i.e. 24 hours). Every SVM has the first 6 hourly water consumptions as input features and a specific hourly water demand as target to be predicted, among the remaining 18. The Mean Average Percentage Error (MAPE), computed on leave-one-out validation, is the black-box objective function optimized.

Moreover, a preliminary time-series clustering has been applied in order to evaluate if this can improve the accuracy of the forecasting mechanism. Time-series clustering implies that the overall number of SVMs, whose hyperparameters are optimized through parallel global optimization, increases, with a SVM trained for each cluster identified and for each hourly water demand to be predicted, making even more critical a quick tuning of the hyperparameters.

Results on the urban water demand data in Milan prove that forecasting error is significantly low and that preliminary clustering allows for further reducing error while also improving computational performances.

Keywords*: short-term water demand forecasting, support vector machine, global optimization, time-series clustering*

1. Introduction

Water Distribution Networks (WDN) are large-scale complex systems whose management requires many interrelated decision making activities. One of the most relevant tasks is Pump Scheduling Optimization (PSO), which aims at reducing energy costs while matching the water demand. In the case that storage tanks are in the WDN, costs reduction can be achieved by scheduling most of the pumping activities within time windows associated to a low price of energy and by using the stored water in the remaining hours of the day. According to this goal, accurate demand forecasting solutions represent an effective tool to estimate the demand to use for solving the PSO problem with sufficient advance [1]. Water demand forecasting in the short-term, typically on the 24 hours, is a sufficient information, if accurate, to operate pumps and tanks effectively and efficiently [2]. Water demand forecasting systems already proved to be able in supporting energy costs reduction around 5% in a WDN in the Netherlands [3].

Several water demand forecasting approaches have been proposed in the literature, however this problem still remains an open challenge, mostly due to the wide set of different characteristics for every case study. Recently

a meta-analysis of the empirical literature on the topic has been published [4] with the aim to identify the possible motivations about the differences in the accuracies resulting and reported according to the different approaches proposed. The most important conclusion is that accuracy depends on relevant characteristics such as demand periodicity, modelling strategy, forecasting horizon, sample size and available variables (only internal/endogenous or also external/exogenous). The availability, as well as the selection, of specific variables definitely drives the selection of the approach to adopt. Usually, approaches working only with variables that can be easily collected, monitored and used by the water utility, such as those collected through Supervisory Control And Data Acquisition (SCADA) system and Automatic Metering Readers (AMR), are preferred, since this is perceived as a possible reduction of the risk to add noise/errors from "external" data/information sources (e.g., weather forecast services) avoiding, for cyber security reasons, the connection of internal systems to the Internet.

Furthermore, an useful categorization of water demand forecasting solutions has been reported in [5], taking into account relevant characteristics such as difference between linear and nonlinear methods – where linear methods are usually not as effective as the nonlinear ones due to the intrinsic nonlinearity of water demand data – and the difference between "modelling" and "predicting" [6], where modelling is devoted to identify periodicity, such as seasonality as well as trends, while predicting usually uses short memory data, together with a model of the underlying data generation process, to provide predictions.

Relevant advances have been achieved through the adoption of machine learning for the implementation of effective short-term water demand forecasting, as well as in hydraulic engineering issues, in general [7]. In particular, Support Vector Machine (SVM) regression has gained an increasing interest [8][9], In particular, as reported in [10], SVM regression proved to be the best choice to implement hourly water demand forecasting when compared with Artificial Neural Networks (ANNs), Projection Pursuit Regressions (PPR), Multivariate Adaptive Regression Splines (MARS), Random Forests and weighted pattern-based water demand forecasting. Furthermore, SVM regression proved to be effective for dynamic forecasting [13].

The practical application of many machine learning algorithms is often limited, since the accuracy of the methods strongly depends on the choice of their hyperparameters. However, the successful application of machine learning in several areas is recently increasing the demand for machine learning systems to be used also by non-experts, making global optimization the most promising tool for disclosing machine learning [11][12] potential taking, in some sense, the human out of the loop [14].

In this paper, we solve the problem of optimizing the hyperparameters of a complex forecasting system whose components are SVM regression models with radial basis function kernel, one for each hour to be predicted. Two hyperparameters for each one of the SVM regression models are tuned through parallel global optimization. It can be considered as a bound-constrained optimization problem with a "black-box" objective function specified algorithmically. The simplest method of searching for a global extremum is to calculate the values of the objective function at the nodes of a two-dimensional grid. This procedure has a computational complexity $O(m^2)$, where m is the number of points for each hyperparameter. Given that the calculation of an objective function's value requires the construction of a model with subsequent verification of its quality, this procedure is time-consuming for large values of m. The application of advanced methods of global optimization can significantly decrease the computational load. To determine hyperparameters, many methods of global optimization have been used: genetic algorithms, particle swarm optimization methods, chaos optimization algorithm, pattern search approach and others. With respect to decision support in WDN management, global optimization, in particular Bayesian Optimization, has been recently proposed for tuning hyperparameters of a Kernel-based clustering that is the core of an analytical leakage localization system [15].

In this paper, we use a parallel global optimization algorithm [16] to optimize hyperparameters of a SVM regression model [17] for the water demand forecasting problem. The algorithm employs the space-filling Peano curves to reduce complicated multiextremal multidimensional optimization problems to the one-dimensional ones. To solve the arising one-dimensional problems we use an information-statistical global search algorithm, empirically competitive with other global optimization methods both in accuracy and performance [16][18]. In the case when the objective function satisfies the Lipschitz condition, the convergence of these methods to the exact solution is proved analytically. Unfortunately, in this particular case, we are working with an algorithmically specified "black-box" function, whose properties are unknown. Nevertheless, the use of these methods still looks more promising compared to grid search and, as established empirically, allows to significantly reducing the calculation time as well as to identify more effective hyperparameters configurations out of the grid.

While hyperparameters optimization for machine learning algorithms has been largely addressed through Random Search, such as in [19][20][21] and Bayesian Optimization, such as in [14][22][23], this is, at our knowledge, the first application of a deterministic global optimization algorithm in a real life application (i.e., machine learning based water demand forecasting).

The rest of the paper is organized as follows: section 2 provides an overview on the methodological background, including SVM regression, parallel global optimization and time-series clustering, along with the description of the dataset used to validate the proposed approach. Section 3 reports all the information about the parallel global optimization and the computing system used to perform the experiments. In Section 4 the relevant results are reported, when a comparison between using and not using the preliminary time-series clustering phase before training the SVM regression models. Finally, Section 5 provides some relevant conclusions about this work.

2. Materials & Methods

This section provides the background about the methodologies adopted. Basics about SVM regression are presented along with the parallel global optimization approach proposed to optimally tune the SVM's hyperparameters. Time-series clustering is then presented, where its application, as first stage of the overall forecasting system, allows for the preliminary identification of typical water consumption patterns/behaviours and, subsequently, the improvement of forecasting accuracy through specialized SVM models specifically trained for each one of the behaviours (i.e. clusters) identified.

2.1. Support Vector Machine for regression

Given a dataset *D*, defined as:

$$D = \left\{ \left(x^i, y^i\right) \mid x^i \in \mathbb{R}^d, y^i \in \mathbb{R} \right\}$$

with $i = 1, \ldots, n$, the basic idea of using SVM [24] for regression [25] consists of searching for a function $f(x)$ that has at most ε deviations from the actual targets y^i for all the data in *D* and, at the same time, is as "flat" as possible. The role of ε is to define a ε-insensitive the following piecewise linear loss function:

$$L_\varepsilon\big(y, f(x)\big) = \max\left(0, |y - f(x)| - \varepsilon\right)$$

According to the loss function, only predictions $f(x)$ differing more than ε from y account for the empirical error computation. The easiest solution is a linear function in the form:

$$f(x) = \langle w, x \rangle + b \text{ with } w \in \mathbb{R}^d \text{ and } b \in \mathbb{R}$$

where <.,.> is the dot product in the *d*-dimensional space. "Flatness" of the solution is represented by small values of *w*. To address the feasibility of the linear solution, the parameter *C* is introduced in order to manage the trade-off between the complexity of the SVM model (i.e. "flatness") and the empirical error (i.e. the amount to which deviations larger than ε are tolerated):

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi^i + \xi^{i*}\right)$$

$$\text{subject to} \begin{cases} y^i - \left\langle w, x^i \right\rangle - b \le \varepsilon + \xi^i \\ \left\langle w, x^i \right\rangle + b - y^i \le \varepsilon + \xi^{i*} \\ \xi^i, \xi^{i*} \ge 0 \end{cases}$$

To solve the optimization problem above, the dualization method based on Lagrange multipliers is applied. By solving the dual problem, the resulting formulation of *f*(*x*) is usually known as the Support Vector expansion because *w* is expressed as a linear combination of the training patterns x^i, making *f*(*x*) completely independent on the dimensionality *d* of the input; it depends only on the number of Support Vectors (x^i such that the associated Lagrange multiplier is not zero). As *f*(*x*) is described in terms of dot products between data, it is not necessary to compute *w* explicitly, an important consideration when formulating the extension to the nonlinear case.

The simplest method to extend the Support Vector regression to nonlinear data is to pre-process the training set by using a mapping function ϕ from the original space (Input Space) to some other space (Feature Space) where the linear approach may be successfully applied. The important result is that, rather than explicitly mapping all the data into the new space through the mapping $\phi(x)$, one can use a kernel function (also knonwn as "kernel trick"). Several types of kernel have been proposed (e.g., Polynomial, Radial Basis Functions, Sigmoid, etc.), each one with at least an internal parameter to be tuned [26]. One of the most widely used kernel is the RBF that is computed as:

$$K(x, x') = e^{\frac{-\gamma\|x - x'\|^2}{2}}$$

This is the kernel used in this study.

2.2. Parallel global optimization algorithm

In this section we consider optimal parameters selection for the SVM ε-regression algorithm with the RBF kernel. Let the training set $D = \left\{ \left(x^i, y^i\right) \mid x^i \in \mathbb{R}^d, y^i \in \mathbb{R} \right\}$ be given, where $x^i \in R^d$ is a feature vector, $y^i \in R$ is the target (numeric) feature, $\mu(x, \theta)$ is a SVM-regression function that minimizes the value of empirical risk (prediction error on the training set), where θ are internal parameters of the function that are unique for each training data and algorithm's hyperparameters C, γ and ε. As is shown in [27] a generalization error of SVM-regression algorithm strictly depends on the choice of C, γ and ε parameters.

One of the common approaches for optimal parameters selection based on cross-validation error optimization was proposed in [28]. The idea of the method is to split the overall dataset randomly to S subsets $\{G_s, s = 1, \dots, S\}$, train the model on $(S-1)$ subsets (i.e. training set) and use the remaining subset (test set) to calculate the validation error.

The error averaged over all the subsets is used as an estimate of the algorithm's generalization error:

$$MSE_{CV} = \frac{1}{S}\sum_{s=1}^{S}\sum_{i\in G_s}\left(y^i - \mu(x^i, \theta^s)\right)^2$$

where θ^s are parameters of the SVM-regression function trained on the $T\backslash G_s$ data. If the number of samples in the training set is not large, leave-one-out error can be used:

$$MSE_{LOO} = \frac{1}{S}\sum_{i=1}^{S}(y^i - \mu(x^i, \theta^i))^2$$

where θ^i are parameters of the SVM-regression function trained on $T\backslash\{x_i\}$ data. Due to the fact that values of θ^i parameters are unique for each set (C, γ, ε) we can consider the leave-one-out error as the function of C, γ and ε:

$$MSE_{LOO} = \frac{1}{S}\sum_{i=1}^{S}(y^i - \mu(x^i, \theta^i(C, \gamma, \varepsilon)))^2 = F(C, \gamma, \varepsilon)$$

Let value of ε be fixed. We will find the optimal value of the parameters C and γ in the hypercube $X = [C_{min}; C_{max}] \times [\gamma_{min}; \gamma_{max}]$. Let us define $\varphi(y) = F(C, \gamma)$, where $y = (C, \gamma)$, and consider the following global optimization problem (in our case $N = 2$):

$$\varphi^* = \varphi(y^*) = \min\{\varphi(y): y \in X\}$$

$$X = \{y \in \mathbb{R}^N: -2^{-1} \le y^i \le 2^{-1}, 1 \le i \le N\}$$

Let the objective function $\varphi(y)$ satisfies the Lipschitz condition[1]

$$|\varphi(y) - \varphi(y')| \le K\|y - y'\|, y, y' \in X$$

with constant *K* which in a general case is unknown. We note that any hyperinterval

$$S = \{y \in \mathbb{R}^N: a^i \le y^i \le b^i, 1 \le i \le N\}$$

can be reduced to a hypercube *X* using a linear coordinate transformation.

The main idea of the algorithm [16] is to reduce the optimization problem to a one-dimensional problem (dimension reduction)

$$\varphi^* = \varphi(y^*) = \varphi(y(x^*)) = \min\{\psi(x) = \varphi(y(x)) : x \in [0,1]\}$$

For this a continuous single-valued mapping such as Peano curve

$$\{y \in \mathbb{R}^N: -2^{-1} \le y^i \le 2^{-1}, 1 \le i \le N\} = \{y(x): 0 \le x \le 1\}$$

[1] If properties of the objective function are unknown, the algorithm can still be applied, but the convergence is not guaranteed.

is used. Numerical methods that allow efficient constructing such mappings with any given accuracy are considered in [16]. The arising one-dimensional problem is solved using an information-statistical global search algorithm [16].

Let us assume $k > 1$ iterations of the method to be completed (the point of the first trial x^1 can be an arbitrary point of the interval $[a; b]$, for example, the middle of the interval). Then, at the $(k+1)$-th iteration, the next trial point is selected according to the following rules.

Rule 1. Renumber the points of the preceding trials (including the boundary points of the interval $[a; b]$) such that

$$0 = x^0 < x^1 < \cdots < x^k < x^{k+1} = 1.$$

The function values $z^i = \psi(x^i)$ have been calculated at all points x^i $(i = 1, 2, \dots, k)$. At the points $x^0 = 0$ and $x^{k+1} = 1$ the function values have not been computed (these points are used for convenience of further explanation).

Rule 2. Compute the values:

$$\mu = \max_{1 \le i \le k} \frac{|z^i - z^{i-1}|}{\Delta^i}, \qquad M = \begin{cases} r\mu, & \mu > 0, \\ 1, & \mu = 0, \end{cases}$$

where $r > 1$ is the *reliability* parameter of the method, $\Delta^i = x^i - x^{i-1}$.

Rule 3. Compute the characteristics for all intervals $(x^{i-1}; x^i)$, $(i = 1, 2, \dots, k+1)$, according to the formulae:

$$R(1) = 2\Delta^1 - 4\frac{z^1}{M}; \qquad R(k+1) = 2\Delta^{k+1} - 4\frac{z^k}{M};$$

$$R(i) = \Delta^i + \frac{(z^i - z^{i-1})^2}{M^2\Delta^i} - 2\frac{z^i + z^{i-1}}{M}, (i = 1, 2, \dots, k+1).$$

Rule 4. Arrange the characteristics of the intervals in decreasing order:

$$R(t_1) \ge R(t_2) \ge \cdots \ge R(t_k) \ge R(t_{k+1})$$

and select p intervals with the highest values of characteristics (p is the number of processors/cores used for the parallel computations).

Rule 5. Execute new trials at the points

$$x^{k+j} = \begin{cases} \dfrac{x^{t_j} + x^{t_j - 1}}{2}, t_j \in \{1, k+1\}, \\ \dfrac{x^{t_j} + x^{t_j - 1}}{2} - \operatorname{sign}(z^{t_j} - z^{t_j - 1}) \dfrac{1}{2r} \left[\dfrac{|z^{t_j} - z^{t_j - 1}|}{M} \right]^N, 1 < t_j < k+1. \end{cases}$$

The termination condition should be checked for all intervals, in which the scheduled trials are executed

$$\Delta^{t_j} \le \varepsilon, 1 \le j \le p$$

where ε is the predefined accuracy of the problem solution. The detailed description of the method is presented in [18].

2.3. Time-series clustering

Every clustering algorithm is aimed at grouping data, represented as vectors in a multi-dimensional space, by maximizing a given measure of similarity within groups while minimizing the same measure between data points belonging to different groups. This general goal is valid also for time-series data but the sequential nature of this type of data requires specific choices for data representation, pre-processing, and selection of the similarity measure to use [29][30].

With respect to data representation, and according to the idea proposed in this paper, the choice was to work directly with the raw data (i.e. the hourly water demand data represented as 24-dimensional vectors). Although this choice can be computational intensive when data dimensionality is high, it is well suited for the short term water demand forecasting. Another relevant point is the choice of a suitable similarity measure to compare time-series data. A useful characterization, proposed in [31], considers three different types of similarity measure:

- Similarity in time. The goal is to cluster together time series that vary in a similar way at each time step. In this case, time series can be clustered by capturing repetitive behaviours occurring always at the same time step or in the same time window (e.g., peak/burst hours)
- Similarity in shape. The goal is to cluster together time series having common shape features e.g., common trends occurring at different times or similar sub-patterns.
- Similarity in change. The goal is to cluster together time series that vary similarly from time step to time step. In this case, data are clustered with respect to the variations between two successive time stamps.

In this paper, clustering is performed, as first analytical stage, with the aim to capture typical consumption behaviours which are characterized by recurrent peak/burst hours depending on water consumption habits. Similarity in time measures are more suitable in capturing classes of typical behaviours, and cosine similarity was chosen for the implementation of the preliminary clustering phase. More in detail, cosine similarity is given by the cosine of a triangle between two vectors, so the value range of cosine similarity is [−1 to 1].

$$s(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \, \|x_j\|}$$

As the components of the urban water demand vectors are not negative, triangle similarity may vary from [0 to 1]. The spherical k-means algorithm provided by the R package “skmeans” is used, which implements a simple k-means strategy based on the cosine distance:

$$(x_i, x_j) = 1 - s(x_i, x_j) = 1 - \frac{\langle x_i, x_j \rangle}{\|x_i\| \, \|x_j\|}$$

Since the clustering algorithm used is basically a k-means, the number *k* of desired clusters must be set up in order to identify a suitable set of different patterns representing typical water consumption behaviours. To select the most suitable value for *k*, two cluster validity measures have been considered, namely Silhouette and Calinski-Harabatz.

The basic advantage of using time-series clustering is that it might allow for the identification of typical water consumption behaviours and, consequently, for splitting the entire dataset into subsets (i.e. clusters) which are used to train behaviour-specific SVMs-based forecasting models.

The application of time-series clustering according to a two-levels schema makes possible to identify possible seasonality. In particular, a new dataset is generated from the original one by computing the average hourly water demand vector for each month:

$$z_{month} = \frac{1}{N_{month}} \sum_{i=1,\dots,N_{month}} x_i$$

where *month* is the index of the month in the year, and N_{month} is the number of days in that month. Thus, the new dataset consists of *Months* time-series data where *Months* is the number of months in the dataset. At the first level, time-series clustering is performed on this new dataset with the aim to identify k_1 clusters related to seasonality (i.e., months characterized by similar average daily consumption patterns). Cluster assignment at this first level is used to label the original time-series dataset; then, according to these labels, k_1 sub-datasets are split from the original one and time-series clustering is performed on each one of them (second level). The best k_2^q is selected for each sub-dataset, with $q = 1, \dots, k_1$.

Although the combination of time-series clustering and SVM regression was initially proposed for both urban and individual customer demand forecasting [32][33], and successively extended to anomaly detection [34], global optimization of the SVMs' hyperparameters is addressed for the first time in this paper.

2.4. Available data set

A real world data set has been used to validate the proposed approach. More in detail, the available data are related to urban water demand in Milan in the period 1 October 2012 to 30 September 2013. Data are collected through the Supervisory Control And Data Acquisition (SCADA) system used to monitor and control the overall urban water distribution network, serving more than 5000 customers (buildings) corresponding to approximately 1.5 million habitants.

The available hourly water demand data have been organized into a time-series dataset $D = \{x_1, \dots, x_l\}$ consisting of l vectors, one for each day in the observation period, and where each vector x_i is a set of 24 ordered values corresponding to the hourly water delivered during the *i*-th day (i.e. time-series data).

3. Experimental setting

3.1. Hyperparameters: definition of the search space

In this work we consider parameter ε of SVM regression be fixed and equal to 1, optimum values of C and γ parameters are searched in $[C_{min}; C_{max}] \times [\gamma_{min}; \gamma_{max}]$ hyperrectangle, where $C_{min} = 1$, $C_{max} = 10$, $\gamma_{min} = 10^{-4}$, $\gamma_{max} = 10^{-1}$.

3.2. Forecasting error as objective function

The frequent evaluation of the forecasting error has a relevant operational impact because the degradation of the forecasting model's accuracy can significantly affect decision making processes, in particular the PSO problem, leading to unnecessary costs. A lot of error measures have been proposed in the literature, where the basic common idea is to compare forecasts with actual observations. The most widely adopted error measure

for time series forecasting, and in particular for the water demand forecasting, is the Mean Absolute Percentage Error (MAPE) [2,7,22–24,26], which is also used in this study. Let us denote

A_t—water demand observed at the time *t*,

F_t —water demand forecasted at the time *t*, and

L—time-series length;

Then, the MAPE is computed as the average of the absolute values of the difference, in percentage, between the forecasted and actual data at each time step

$$\text{MAPE} = \frac{100}{L}\sum_{t=1}^{L}\left|\frac{A_t - F_t}{A_t}\right|$$

MAPE is the black-box and expensive function we want to optimize. It is important to highlight that every SVM regression model is optimized individually – and in parallel. This objective function is clearly black-box since the MAPE value can be computed only after the training and test of every SVM regression model. On the other hand, it is also expensive because MAPE is computed through Leave-One-Out validation (i.e. k-fold cross validation with k equals to the number of instances of the dataset considered) so training and test have to be repeated as many times as the number of time-series into the dataset considered (i.e. a given cluster of those identified through clustering).

3.3. Computing system configuration

Computational experiments were conducted on the Lobachevsky supercomputer at the University of Nizhni Novgorod. We used 18 computational nodes, each with two 8-core Intel Sandy Bridge E5-2660 CPUs (2.2 GHz), 64 GB RAM. We employed the SVM implementation from the OpenCV 3.2 library and the parallel global optimization methods from the Globalizer solver. The code was compiled with Intel C++ Compiler 17 from the Intel Parallel Studio XE tool suite.

4. Results and discussion

This section reports the results obtained on the experiments performed. In the following Table 1 the prediction error is reported, taking into account the case of preliminary application of time-series clustering at a first stage as well as the case of the use of the entire dataset as is.

Results from time-series clustering have been already reported in [34]; in particular, the two level clustering procedure reported in the previous section 2.3 has been adopted in order to better identify possible seasonality. The relevant information for this study is that the preliminary time-series clustering phase is able to identify 6 different typical behaviours – according to best value of Silhouette and Calinski-Harabaz indices – and, therefore, allows for splitting the entire dataset into 6 subsets.

Table 1. Prediction error results for water demand forecasting with (ClusterID = 1..6) and without (ClusterID = ALL) time-series clustering.

ClusterID	MAPE_LOO	MIN	MAX	STD
1	0.033837	0.007719	0.129705	0.027154
2	0.063427	0.023554	0.210359	0.037293
3	0.097701	0.013512	0.209294	0.053995
4	0.072746	0.018296	0.291297	0.076188
5	0.057024	0.020734	0.130043	0.028881
6	0.043711	0.008002	0.217389	0.038703
ALL	0.082123	0.018346	0.381675	0.058579

ClusterID is a number of a time-series cluster, the "ALL" row corresponds to all data without any time-series clustering performed at the first stage, *MAPE_LOO* is the MAPE computed on leave one out validation, *MIN* is minimum MAPE, *MAX* is maximum MAPE, *STD* is standard deviation of the MAPE.

Along with the prediction accuracy performance, we also provide results about computational performances in order to assess scalability of the algorithm. The computational experiment consists of solving 18 tasks, each corresponding to optimizing the SVM regression error in the water demand forecasting with time-series clustering, at the appropriate hour in a day. When solving each of the tasks for calculating the value of the objective function at each point, a series of SVM models are trained for different values of two SVM parameters, followed by the calculation of the prediction error. Note that the calculation of the objective function is quite computationally intensive. The number of such calculations and overall run time are the objects of our analysis.

The experiment is organized as follows: each task was run on a separate node of the supercomputer. Then, in solving each problem, we used the previously described scheme of parallel computations. Each node of the supercomputer uses 16 cores using OpenMP for multithreading. Note that the scheme of calculations in sequential and parallel experiments differs. So, in the sequential case, the algorithm chooses the next point in the two-dimensional parameter space at the best-characteristic (an estimate of the probability of finding the global optimum) interval. In parallel mode, we select 16 such intervals and calculate 16 values of the function in parallel, subsequently choosing the best. In this regard, the number of calculated values of the objective function in the sequential and parallel experiments can be significantly different. Table 2 shows the performance results for each of the 18 tasks; the *Problem ID* column contains the task number. The *Sequential version* and *Parallel version* columns contain the results of the experiments using 1 and 16 threads, respectively. The numbers of points in which the value of the objective function is calculated and the total run time before the termination criterion is fulfilled are given. The results of experiments have shown that the run time varies slightly from task to task, exception is task #13. The situation changes in the parallel experiments. The run time from task to task changes, but intra-node speedup from 1 core to 16 cores at the same node is 9 to 24 times. Acceleration of more than 16 in some experiments is explained by the fact that the method, calculating the value of the objective function in 16 perspective points at every step. During this process, it is possible to build a better trajectory on the way to the global extremum than in the sequential case. The average strong scaling efficiency in calculations at one node is 82%. In whole, the wall time for a parallel experiment is 4.5 minutes approximately.

Table 2. Performance results in the sequential and parallel experiments during the SVM regression error optimization in the water demand forecasting WITHOUT time-series clustering. 18 nodes of the supercomputer are used for the experiments.

Problem ID	Sequential version (1 thread)		Parallel version (16 threads)		Speedup
	Number of Points	Run time (seconds)	Number of Points	Run time (seconds)	
1	4055	2 079	3936	160	13
2	4301	2 205	4384	178	12
3	4296	2 206	6064	249	9
4	4302	2 224	3296	133	17
5	4303	2 227	2944	121	18
6	4303	2 223	5936	235	9
7	4308	2 243	5248	211	11
8	4301	2 213	6656	269	8
9	4049	2 081	6048	241	9
10	4301	2 221	5952	242	9
11	4304	2 252	4816	200	11
12	4301	2 213	5184	207	11
13	6986	3 640	6144	246	15
14	4055	2 120	2240	91	23
15	4296	2 237	2368	93	24
16	4299	2 206	5792	234	9
17	4304	2 202	3584	142	16
18	4297	2 193	4416	175	13

In the second experiment we collect performance data during optimization of the SVM regression error in the water demand forecasting with time-series clustering. First, we split the data into 6 clusters.

Next, we optimize the hyperparameters of SVM regression for each of the 18 tasks using the clustering results. The results are presented in Table 3. The column *Problem ID* indicates the number of the task. The number of computed objective function values and corresponding run time in seconds are given in the columns *Number of points* and *Run time*, respectively. The numbers in these columns are aggregated on clusters. The last column, *Performance improvement*, shows the advantage of time-series clustering in terms of run time compared to the algorithm without clustering. The number of calculated values of the objective functions is much larger than before. This is due to the fact that we aggregate data from all 6 clusters. Note that the computational complexity of computing the objective function is much smaller after clustering, so the aggregated run time is even less than in the experiment without clustering. The wall time of this experiment can be computed as a maximum value in the second column of the table and is 1 minute approximately. It means that time-series clustering not only decreases the SVM regression error but also results in 4.5-fold performance improvement compared to the previous experiment.

Table 3. Performance results in the parallel experiments during the SVR error optimization in the water demand forecasting with time-series clustering. 18 nodes of the supercomputer (288 cores) are used for the experiments. Run time is compared to the previous results.

Problem ID	Number of points	Run time (seconds)	Performance improvement
1	41 296	40	4
2	40 560	29	6
3	38 880	21	12
4	46 032	59	2
5	32 528	38	3
6	38 480	22	11
7	29 936	15	14
8	32 336	23	12
9	40 416	39	6
10	36 704	25	10
11	33 968	15	14
12	37 008	37	6
13	39 472	27	9
14	34 704	23	4
15	37 792	39	2
16	34 848	19	13
17	38 064	24	6
18	38 576	39	5

The results of the experiments showed that the value of the objective function (prediction error) varies insignificantly in the two-dimensional parameters search space. It means that the regularities in the analyzed data are sufficiently obvious. Therefore, SVM with a Gaussian kernel is good enough for practical use in case of reasonable choice of its parameters. Meanwhile, this fact is empirical and there is no guarantee that this is true for any dataset, such as an urban water data of a different town or individual customer water consumptions data, as well as forecasting in similar domains, such as energy/gas demand data.

In this regard, the optimization of hyperparameters of SVM-based forecasting system proposed in this paper is still an actual problem for its subsequent practical use.

5. Conclusions

The short-term demand forecasting system proposed in this paper offers several interesting innovations. It uses SVM regression as base algorithm to provide predictions but, with respect to other papers using SVM, the proposed approach does not aim at inferring a relation like $predicted_{t+1} = f(actual_t, \dots, actual_{t-h})$, with h to define how much past information is required to perform a prediction. Instead, the proposed system addresses the short-term water demand forecasting problem by training one SVM regression for every hour of the day which the water demand has to predicted for. Only the first 6 actual hourly water demand data are used as input of all the SVM models: this allows to have a complete forecast for the remaining hours of the day avoiding the propagation of a prediction error from and SVM to the next one, as it would be in a "staked" system.

Another important innovation is the adoption of parallel global optimization for tuning the hyperparameters of every SVM regression model. With respect to grid search, global optimization allows for identifying a better hyperparameters configuration – if any – out of the grid, by using the same number of objective function evaluations of the grid (i.e. computations of forecasting error, MAPE, on leave one out validation). This is

crucially important for the hyperparameters tuning of individual machine learning algorithms as well as of more complex machine learning based system such as the one presented in this paper.

Finally, using a preliminary time-series clustering phase, proved to be able to improve forecasting accuracy, based on the idea that clustering can capture the limited set of typical water usage behaviours occurs, recurring on different time scales, such as seasonality – at a monthly time scale – and type of day – at a finer scale. More important, even if time-series clustering increases the number of SVM regression model to be learned – in particular, one SVM for each pair *(cluster, hour to predict)* – the parallel global optimization algorithm is able to scale efficiently, reducing the wall clock time thanks to the distribution of the computational load and the reduced size of each subsets on which leave one out validation is performed.

Authors are aware that the proposed approach does not deal with uncertainty which could potentially affect the water consumption data. Indeed, even if SCADA systems are usually highly reliable, sensors could be affected by some fault, resulting in noising data. To overcome this limitation, authors will replace SVM with robust chance-constrained SVM, recently proposed in the literature [35][36] and able to deal with data with uncertainties.

Acknowledgments

Alexander Sysoyev and Konstantin Barkalov acknowledge the support from the Russian Science Foundation project No. 16-11-10150.

University of Milano-Bicocca acknowledge the support from the European Project "DATA4WATER" – Project ID: 690900 – founded under H2020-EU.4.b. - Twinning of research institutions

References

[1] Mala-Jetmarova, H., Sultanova, N., Savic, D. (2017), Lost in optimisation of water distribution systems? A literature review of system operation, Environ. Model. & Softw., 93, 209–254.

[2] Mamo, T.G., Juran, I., Shahrour, I. (2013), Urban water demand forecasting using the stochastic nature of short term historical water demand and supply pattern, J. Water Resour. Hydraul. Eng., 2, 92–10.

[3] Bakker, M., van Duist, H., van Schagen, K., Vreeburg, J., Rietveld, L. (2014), Improving the performance of water demand forecasting models by using weather input. Proced. Eng., 70, 93-102, DOI: 10.1016/j.proeng.2014.02.012.

[4] Sebri, M. (2016), Forecasting urban water demand: a meta-regression analysis. J. Environ. Manag., 183, 777–785, DOI: 10.1016/j.jenvman.2016.09.032.

[5] Donkor, E.A., Mazzucchi, T.A., Soyer, R., Roberson, J.A. (2014), Urban water demand forecasting: review of methods and models. J. Water Resour. Plan. Manag., 140, 146–159, DOI: 10.1061/(ASCE)WR.1943-5452.0000314.

[6] Know, H., So, B., Kim, S., Kim, B. (2014) Development of ensemble model based water demand forecasting model, EGU Gen. Assem. Conf. Abstr., 16, 3711.

[7] Wu, M.C., Lin, G.F. (2015), An Hourly Streamflow Forecasting Model Coupled with an Enforced Learning Strategy, Water, 7(11), 5876-5895. DOI:10.3390/w7115876.

[8] Ji, G., Wang, J., Ge, Y., Liu, H. (2014), Urban Water Demand Forecasting by LS-SVM with Tuning Based on Elitist Teaching-Learning-Based Optimization. In The 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, May 31 2014-June 2 2014, 3997–4002.

[9] Sampathirao, A.K., Grosso, J.M., Sopasakis, P., Ocampo-Martinez, C., Bemporad, A., Puig, V. (2014), Water Demand Forecasting for the Optimal Operation of Large-Scale Drinking Water Networks: The Barcelona Case Study. In Proceedings of the 19th International Federation of Automatic Control (IFAC) World Congress, Cape Town, South Africa, 10457–10462.

[10] Herrera, M., Torgo, L., Izquierdo, J., Pérez-García, R. (2010), Predictive models for forecasting hourly urban water demand. J. Hydrol. Eng., 387, 141–150, DOI: 10.1016/j.jhydrol.2010.04.005.

[11] Learning and Intelligent Optimization - 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013 co-editors: Giuseppe Nicosia, Panos M. Pardalos, Lecture Notes in Computer Science 7997, Springer 2013, ISBN 978-3-642-44972-7.

[12] Learning and Intelligent Optimization - 8th International Conference, LION 8, Gainesville, Florida,February 16-21, 2013 co-editors: Panos M. Pardalos, Mauricio G.C. Resende, Chrysafis Vogiatzis, and Jose L. Walteros. Lecture Notes in Computer Science 8426, Springer 2014, ISBN 978-3-319-09583-7.

[13] Bai, Y., Wang, P., Li, C., Xie, J., Wang, Y., (2015) Dynamic forecast of daily urban water consumption using a variable-structure support vector regression model. J. Water Resour. Plan. Manag., 141, 04014058, DOI: 10.1061/(ASCE)WR.1943-5452.0000457.

[14] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., de Freitas, N. (2016), Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1), 148-175.

[15] Candelieri, A., Giordani, I., Archetti, F. (2017), Automatic Configuration of Kernel-based Clustering: an optimization approach, in Proceedings of Learning and Intelligence Optimization –11th International Conference LION 11, Nizhny Novgorod, Russia, June 19-21, 2017, 34-49..

[16] Strongin, R.G., Sergeyev, Ya. D. (2000), Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht.

[17] Barkalov K., Polovinkin A., Meyerov I., Sidorov S., Zolotykh N. (2013) SVM Regression Parameters Optimization Using Parallel Global Search Algorithm. In: Malyshkin V. (eds) Parallel Computing Technologies. PaCT 2013. Lecture Notes in Computer Science, vol 7979. Springer, Berlin, Heidelberg.

[18] Barkalov, K., Gergel, V. (2016), Parallel global optimization on GPU. J Glob Optim (2016) 66: 3.

[19] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. (2016), Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. arXiv preprint arXiv:1603.06560.

[20] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F. (2015), Efficient and robust automated machine learning. In Advances in Neural Information Processing Systems, 2962-2970.

[21] Mantovani, R. G., Rossi, A. L., Vanschoren, J., Bischl, B., de Carvalho, A. C. (2015), Effectiveness of random search in SVM hyper-parameter tuning. In Neural Networks (IJCNN), 2015 IEEE International Joint Conference on, 1-8.

[22] Snoek, J., Larochelle, H., Adams, R.P. (2012), Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems.

[23] Brochu, E., Cora, V.M., De Freitas, N. (2010), A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599.

[24] Vapnik, V.N. (1995), The Nature of Statistical Learning Theory, Springer: New York, NY, USA, 1995.

[25] Vapnik, V.N. (1998), Statistical Learning Theory, Wiley: New York, NY, USA, 1998.

[26] Schölkopf, B., Smola, A.J. (2002), Learning with Kernels: Support. Vector Machines, Regularization, Optimization, and Beyond, MIT Press: Cambridge, MA, USA, 2002.

[27] Ren, Y., Bai, G. (2010), Determination of Optimal SVM Parameters by Using GA/PSO. Journal of Computers 5(8), 1160–1168.

[28] Ito, K., Nakano, R. (2003), Optimization Support Vector Regression Hyperparameters Based on Cross-Validation. In: Proceedings of the International Joint Conference on Neural Networks, 3, 2077–2083.

[29] Liao, T.W. (2005), Clustering of time series data—a survey. Pattern Recognit., 38, 1857–1874, DOI: 10.1016/j.patcog.2005.01.025.
[30] Kavitha, V., Punithavalli, M. (2010), Clustering time series data stream-a literature survey. J.Comput. Sci. Inf. Secur., 8.
[31] Zhang, X., Liu, J., Du, Y., Lv, T. (2011), A novel clustering method on time series data. Expert Syst. Appl., 38, 11891–11900, DOI: 10.1016/j.eswa.2011.03.081.
[32] Candelieri, A., Archetti, F. (2014), Identifying typical urban water demand patterns for a reliable short-term forecasting - The icewater project approach, Procedia Eng., 89, 1004–1012.
[33] Candelieri, A., Soldi, D., Archetti, F. (2015), Short-term forecasting of hourly water consumption by using automatic metering readers data, Procedia Eng., 119, 1, 844–853, 2015.
[34] Candelieri, A., (2017) Clustering and support vector regression for water demand forecasting and anomaly detection, Water, vol. 9, no. 3, p. 224, 2017.
[35] Wang, X., Fan, N., Pardalos, P. M. (2017). Stochastic subgradient descent method for large-scale robust chance-constrained support vector machines. Optimization Letters, 11(5), 1013-1024.
[36] Wang, X., Fan, N., Pardalos, P. M. (2015). Robust chance-constrained support vector machines with second-order moment information. Annals of Operations Research, 1-24.