



**HAL**  
open science

## SVM hyperparameters tuning for recursive multi-step-ahead prediction

Jie Liu, Enrico Zio

► **To cite this version:**

Jie Liu, Enrico Zio. SVM hyperparameters tuning for recursive multi-step-ahead prediction. *Neural Computing and Applications*, 2016, 28 (12), pp.3749 - 3763. 10.1007/s00521-016-2272-1 . hal-01342889

**HAL Id: hal-01342889**

**<https://hal.science/hal-01342889>**

Submitted on 6 Jul 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SVM hyperparameters tuning for recursive multi-step ahead prediction

Jie Liu<sup>1</sup> and Enrico Zio<sup>1, 2, \*</sup>

<sup>1</sup>Chair on System Science and the Energetic Challenge, EDF Foundation,  
CentraleSupélec, Paris, France

<sup>2</sup> Energy Departement, Politecnico di Milano, Milano, Italy

\*Corresponding author: enrico.zio@ecp.fr

## Abstract

Prediction of time series data is of relevance for many industrial applications. The prediction can be made in one-step and multi-step ahead. For predictive maintenance, multi-step ahead prediction is of interest for projecting the evolution of the future conditions of the equipment of interest, computing the remaining useful life and taking corresponding maintenance decisions. Recursive prediction is one of the popular strategies for multi-step ahead prediction. SVM is a popular data-driven approach that has been used for recursive multi-step ahead prediction. Tuning the hyperparameters in SVM during the training process is challenging, and normally the hyperparameters are tuned by solving an optimization problem. This paper analyses the possible objectives of the optimization for tuning hyperparameters. Through experiments on one synthetic dataset and two real time series data, related to the prediction of wind speed in a region and leakage from the reactor coolant pump in a nuclear power plant, a bi-objective optimization combining mean absolute derivatives and accuracy on all prediction steps is shown to be the best choice for tuning SVM hyperparameters for recursive multi-step ahead prediction.

Key words: recursive multi-step ahead prediction, support vector machine, hyperparameters tuning, single-objective optimization, bi-objective optimization

## 1. Introduction

Prognostics and Health Management (PHM) is becoming an important topic of interest for industrial applications (Vichare and Pecht, 2006; Lei *et al.*, 2007; Wu and Chau, 2013;

Zhao *et al.*, 2014; Javed *et al.*, 2015; Lei *et al.*, 2015). Prognostics includes the prediction of the health condition of the equipment of interest in the future (Jardine *et al.*, 2006; Palm, 2007; Saha *et al.*, 2009; Zio, 2012). This can be done with reference to the next time instant, i.e. one-step ahead prediction or multiple time instants, i.e. multi-step ahead prediction (Jardine *et al.*, 2006; Wang and Han, 2015). The latter is definitely more difficult than the former.

Support Vector Machine (SVM) (Drucker *et al.*, 1997; Ding *et al.*, 2013; Wang *et al.*, 2013) is a popular data-driven approach, also widely used for prognostics. The inputs are features extracted from measurements directly or indirectly related to the prediction of interest. SVM has been used for example in Muller *et al.* (1997), Min and Lee (2005), Asefa *et al.* (2006), Widodo *et al.* (2011), Zio and Di Maio (2012), Benkedjough *et al.* (2013b), Liu *et al.* (2014), Namdari and Jazayeri-Rad (2014). Multi-step ahead prediction, e.g. to compute the remaining useful life, i.e. the time at which the component degradation state reaches a predefined failure threshold, has been done for example in Asefa *et al.* (2006), Lin *et al.* (2006), Sorjamaa *et al.* (2007), Sapankvych and Sanka (2009), Nuhic *et al.* (2013), Benkedjough *et al.* (2013a), Bao *et al.* (2014). Recursive (iterative) prediction is a popular strategy for multi-step ahead prediction of time series data. Recursive SVM achieves the multi-step ahead prediction on the basis of the predictions of previous steps, which are used as in inputs for the prediction in the following step. For example, in Benkedjough *et al.* (2013a), SVM is used to assess the current status and predict the Remaining Useful Life (RUL) of bearings. In Nuhic *et al.* (2013), SVM is used to recursively predict the future capacity of lithium-ion batteries to estimate the time left for reaching 80% of the nominal capacity.

There are three types of hyperparameters in SVM, i.e. the regularization parameter which balances the prediction accuracy and the flatness of the model, the hyperparameters related to the kernel function which define the high-dimensional feature space and the hyperparameters related to the loss function which controls the number of data points in the model.

Tuning the hyperparameters of SVM for application to prognostics is a challenge. For the same dataset, different hyperparameters values may give either very good or very bad prediction results. The best hyperparameters values are those which give best prediction on the new data. How to find the optimal hyperparameters values effectively and efficiently and how to guarantee the prediction performance without overfitting on the dataset used for SVM training are critical problems.

Cross-Validation is used in Nuhic *et al.* (2013) to find the best hyperparameters of SVM considering the Mean Squared Error (MSE). A hybrid of Particle Swarm Optimization (PSO) and Differential Evolution (DE) is used to search the optimal parameter combination in the search space in Zhang *et al.* (2015) with minimal prediction error. PSO is employed also in Aich and Banerjee (2014) for the purpose of finding the SVM hyperparameters that minimize the Mean Absolute Percentage Error (MAPE) on the training dataset. Adaptive PSO is used in Wang *et al.* (2015) for tuning hyperparameters. Hyperparameters tuning in Chapelle *et al.* (2002) is done by minimizing some estimates of the generalization error of SVM using a gradient descent algorithm over the parameters space. In Liu *et al.* (2006), Grid Search (GS) is combined with evolutionary strategies for optimizing hyperparameters in SVM, aiming at minimizing the cross-validation accuracy. In Wu *et al.* (2007), a genetic-based SVM is developed to determine automatically the hyperparameters values with the highest predictive accuracy and generalization ability simultaneously. A fast messy GA is used in Chou *et al.* (2014) to find the best hyperparameters values with a minimum number of support vectors and optimal SVM parameters to preserve an acceptable level of prediction accuracy. In Igel (2005), different bi-objective optimizations considering model complexity and prediction accuracy for hyperparameters tuning are proposed and tested. Aydin *et al.* (2011) uses the percentage of support vector and prediction accuracy as the objectives for tuning hyperparameters in SVM. A comprehensive learning PSO-based memetic algorithm that evolves feature selection and parameter optimization simultaneously is proposed in Hu *et al.* (2014) to minimize the MAPE on the training dataset. Chen *et al.* (2014) hybridizes a SVM model with an adaptive Genetic Algorithm (GA) search and the seasonal index adjustment to forecast daily flow of holiday tourist. The maximum-margin principle and the jackknife technique are used in Chang and Chou

(2015) for tuning hyperparameters of L2-loss SVM. Fruit fly optimization algorithm is used in Shen *et al.* (2016) for tuning hyperparameters.

As discussed above, tuning hyperparameters in SVM is normally treated as an optimization problem, single-objective or bi-objective. Most methods developed in the literature, then, focus on faster processes for optimization. The objective is always prediction accuracy for single-objective optimization or prediction accuracy combined with some metrics of model generalization ability for bi-objective optimization. These objectives are computed on the one-step ahead prediction, even for recursive multi-step ahead prediction.

Indeed, recursive SVM for multi-step ahead prediction is based on the one-step ahead prediction and that is why the prediction accuracy on the one-step ahead prediction is the objective of the optimization problem to find the best hyperparameters during the training process. This, however, does not always guarantee satisfactory prediction results at multi-step ahead prediction, as pointed out in Taieb *et al.* (2010).

To the authors' knowledge, there are many good reviews and original research on tuning hyperparameters in SVM for one-step ahead prediction, whereas there has not been much published research reporting specifically on the problem of the objectives for hyperparameters tuning in SVM multi-step prediction. In this paper, we try to address this by considering the prediction accuracy at all steps, also including the last step, during the training process for multi-step ahead prediction. Various possibilities for defining the objectives for tuning hyperparameters of recursive SVM are compared. Considering prediction accuracy, the objectives could be the accuracy (MSE or MAPE) on the first step, on all the steps and on the last step of the multi-step ahead prediction on the training dataset. Model complexity is also considered as proposed in Igel (2005) to quantify the generalization ability: the smaller the model complexity is, the stronger its generalization ability is. The Mean of the Absolute value of the Derivatives (called MAD, in this paper) of the estimate function in SVM on all training data points is used to describe the robustness of the model under small perturbations in the inputs. Smaller MAD means stronger robustness of the SVM model.

Single-objective optimization for tuning hyperparameters considering only the prediction accuracy (for multi-step ahead prediction, as explained previously) and bi-objective

optimization considering the prediction accuracy (on multi-step ahead prediction) and the model inherent characteristics (model complexity and MAD) are compared on the multi-steps ahead prediction of one synthetic dataset and two real time series datasets.

One real series relates to the prediction of the leakage from the first seal of the Reactor Coolant Pump (RCP) in a Nuclear Power Plant (NPP). The other real series regards predicting wind speed. The experimental results obtained in these three cases show that 1) optimization considering the prediction accuracy on all the steps of the prediction horizon as objective of the optimization for tuning hyperparameters gives better results than the one of considering only the prediction accuracy on the first or last step of the prediction horizon; 2) the bi-objective optimization gives better results than the single-objective optimization.

The rest of the paper is structured as follows. Section 2 describes briefly the hyperparameters to tune in SVM and the objectives in the associated optimization. The case studies on one synthetic dataset and two real time series datasets are carried out in Section 3 to compare the different objectives. Based on the results obtained, some discussions are also given in this Section. Conclusions are drawn in Section 4.

## **2. Methodology**

In this Section, the basic strategy for recursive multi-step ahead prediction by SVM is introduced at the beginning. Different objectives that can be used for optimizing the hyperparameters in SVM for one-step ahead prediction are presented. Their extension to multi-step ahead prediction are also presented in this Section.

### **2.1 Recursive multi-step ahead prediction**

In recursive multi-step ahead prediction, the model is trained for one-step ahead prediction and the predicted value is appended to previous predicted values in input to predict the value for the next step. For simplicity, we consider a single-variable time series whose values until time  $T$  are  $\{x_1, x_2, \dots, x_T\}$ . The aim is to predict the future values of  $N$  steps ahead. The inputs are  $M$  previous values, i.e. the input-output pairs  $(\mathbf{x}_t, \mathbf{y}_t)$  are  $([x_{t-M+1} \ x_{t-M+2} \ \dots \ x_t], [x_{t+1} \ x_{t+2} \ \dots \ x_{t+N}])$ . For a time series data with  $T$  values, the

number of constructed data points is  $K = T - N - M + 1$ . In Figure 1,  $\hat{x}_{t+i}, i = 1, 2, \dots, N$  are the predicted values for each step. Figure 1 shows the schematic illustration of the recursive prediction process. It is seen that the predictions on all the steps, except the first step, use the values predicted at the previous steps.

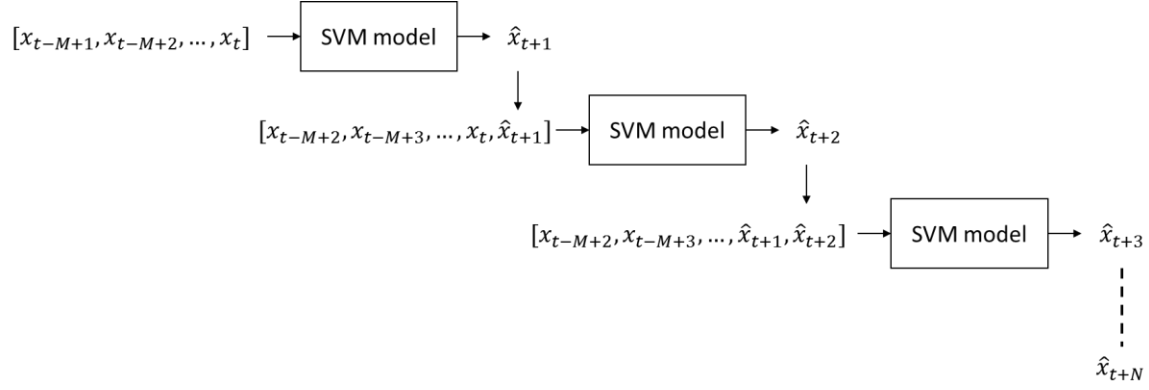


Fig. 1 Schematic illustration of recursive multi-steps ahead prediction using SVM.

## 2.2 Support Vector Regression

In this paper, we consider regression by SVM with  $\epsilon$ -insensitive loss function. The basic idea of SVM is to map the training data points into a high-dimensional space, i.e. a feature space, where the relation between the training data points becomes linear.

Suppose we have  $K$  data points of the form  $(\mathbf{x}_t, y_t), t = 1, 2, \dots, K$  and that  $k(\mathbf{x}_i, \mathbf{x})$  is the kernel function that represents the inner product of two training data vectors (points) in the feature space. SVM tries to estimate the function of interest  $f(\mathbf{x})$  as the linear combination (as in Equation (1) below) that best approximates the underlying relation between the input and output, as represented by the data:

$$f(\mathbf{x}) = \sum_{i=1}^K (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b . \quad (1)$$

In Equation (1), the unknown Lagrange multipliers  $\alpha_i$  and  $\alpha_i^*$ , for  $i = 1, 2, \dots, K$ , can be calculated by solving the following optimization problem:

$$\begin{aligned} & \text{Maximize } -\frac{1}{2} \sum_{i,j=1}^K (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_{i=1}^K (\alpha_i + \alpha_i^*) + \sum_{i=1}^K y_i (\alpha_i - \alpha_i^*) \\ & \text{subject to } \sum_{i=1}^K (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \quad (2)$$

with  $C$  being the penalty hyperparameter and  $\epsilon$  the hyperparameter related to the loss function. There are also hyperparameters related to the kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$ , e.g.  $\sigma$  in a Radial Basis Function (RBF), with  $k(\mathbf{x}_i, \mathbf{x}_j) = e^{(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / (2\sigma^2))}$ . The values of these hyperparameters are critical for the prediction performance of the SVM model and, thus, hyperparameters tuning is very important for implementing SVM.

The unknown  $b$  in Equation (1) can be calculated by imposing the so-called Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951).

### 2.3 Objectives for tuning hyperparameters

Normally the hyperparameters introduced in Section 2.2 are tuned by solving an optimization problem. As shown in Figure 1, recursive prediction is achieved by recursively applying one-step ahead prediction. In most of the published papers, the objective of the optimization problem for tuning hyperparameters in SVM for recursive multi-step ahead prediction is the same as for one-step ahead prediction, i.e. to minimize the prediction error of the one-step ahead prediction on the training dataset.

In this section, the possible objectives for hyperparameters tuning of SVM for one-step ahead prediction are considered and some of them are extended for multi-steps ahead prediction. In Section 3, experiments are carried out comparing these different objectives in the case studies to draw some practical conclusions.

#### 2.2.1 Accuracy-related objectives

In the published papers, two widely used accuracy-related objectives are MSE and MAPE on the one-step ahead prediction:

$$MSE = \frac{1}{K} \sum_{t=1}^K \|\hat{x}_{t+1} - x_{t+1}\|^2 \quad (3)$$

$$MAPE = \frac{1}{K} \sum_{t=1}^K \frac{\|\hat{x}_{t+1} - x_{t+1}\|}{\|x_{t+1}\|} \quad (4)$$

where  $K$  is the number of training data points,  $\hat{x}_{t+1}$  and  $x_{t+1}$  are the predicted value and true value of the one-step ahead prediction, respectively and  $\|\bullet\|$  is the absolute value.

In the problem of multi-step ahead prediction, other accuracy-related objectives can be used. In order to distinguish these different accuracy-related objectives, the previous *MSE*



and *MAPE* on one-step ahead prediction in Equations (3) and (4) are named *MSE\_one* and *MAPE\_one*.

For the multi-step ahead prediction, the accuracy on the first step is not the only objective; actually, the prediction performance on the following steps is the most important. Thus, four other prediction accuracy-related objectives are proposed in this paper for optimizing the hyperparameters of SVM, which are the MSE and MAPE on the last step, i.e. *MSE\_last* and *MAPE\_last*, and the MSE and MAPE on all the steps, i.e. *MSE\_all* and *MAPE\_all*:

$$MSE\_last = \frac{1}{K} \sum_{t=1}^K \|\hat{x}_{t+N} - x_{t+N}\|^2, \quad (5)$$

$$MAPE\_last = \frac{1}{K} \sum_{t=1}^K \frac{\|\hat{x}_{t+N} - x_{t+N}\|}{\|x_{t+N}\|}, \quad (6)$$

$$MSE\_all = \frac{1}{K} \sum_{t=1}^K \sum_{i=1}^N \|\hat{x}_{t+i} - x_{t+i}\|^2, \quad (7)$$

$$MAPE\_all = \frac{1}{K} \sum_{t=1}^K \sum_{i=1}^N \frac{\|\hat{x}_{t+i} - x_{t+i}\|}{\|x_{t+i}\|}. \quad (8)$$

The previous four accuracy-related objectives for hyperparameters tuning give the information of interest on the trained SVM model for multi-step ahead prediction and remaining useful life estimation.

### 2.2.2 Model-characteristics-related objectives

The accuracy measures in Equations (3-8) can be used separately as objectives of single-objective optimization or jointly with other model-characteristics-related objectives for tuning hyperparameters in a bi-objective optimization framework. One widely used model-characteristics-related objective is the model complexity of SVM, which describes the complexity (or simplicity) of the model. The simpler the model is, the stronger the model generalization ability is. The model complexity of SVM is calculated as Igel (2005):

$$M_{comp} = R^2 \sum_{t=1}^K \|\alpha_t - \alpha_t^*\|, \quad (9)$$

where  $\alpha_t$  and  $\alpha_t^*$  are the Lagrange multipliers, and  $R$  denotes the radius of the smallest sphere in the feature space containing all training data points. The minimal radius  $R$  can be calculated by:

$$R = \sqrt{\sum_{i=1}^K \beta_i k(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^K \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j)} , \quad (10)$$

where the values of  $\beta_i$  for  $i = 1, 2, \dots, K$  are calculated by solving the quadratic optimization problem:

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^K \beta_i k(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^K \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) \\ & \text{Subject to } \sum_{i=1}^K \beta_i = 1 \text{ and } \beta_i > 0, i = 1, 2, \dots, K \end{aligned} \quad (11)$$

The estimate function in Equation (1) is differentiable if the kernel function is differentiable, which is the case most times, e.g. RBF, polynomial kernel function etc. Thus, if the kernel function is differentiable, the derivative of the estimate function on each training data point characterizes the robustness of the trained SVM model on the training data point under small noise. The Mean Absolute Derivative (MAD), proposed in this paper for the first time, can be used to characterize the robustness of the model under small noise.

The derivative of the estimate function on the training data point  $(\mathbf{x}, y)$  is:

$$\frac{df(\mathbf{x})}{dx} = \frac{d(\sum_{i=1}^K (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b)}{dx} = \sum_{i=1}^K (\alpha_i - \alpha_i^*) \frac{dk(\mathbf{x}_i, \mathbf{x})}{dx} . \quad (12)$$

Then, MAD can be calculated as:

$$MAD = \sum_{t=1}^K \left\| \left. \frac{df(\mathbf{x})}{dx} \right|_{\mathbf{x}_t} \right\| = \sum_{t=1}^K \sum_{i=1}^K \|\alpha_i - \alpha_i^*\| \left\| \left. \frac{dk(\mathbf{x}_i, \mathbf{x})}{dx} \right|_{\mathbf{x}_t} \right\| . \quad (13)$$

As an example, for the RBF kernel function we have:

$$MAD = \sum_{t=1}^K \sum_{i=1}^K \|\alpha_i - \alpha_i^*\| \left\| \frac{\mathbf{x}_i - \mathbf{x}_t}{\sigma^2} \right\| e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_t\|^2}{2\sigma^2}} . \quad (14)$$

## 2.4 Possible objectives for tuning hyperparameters in SVM

Considering the objectives introduced in Sections 2.3 and 2.4, the possible optimization for tuning hyperparameters in SVM during the training process can be formed as single-objective or bi-objective.

Single-objective optimization considers only the prediction accuracy, which includes  $MSE\_one$ ,  $MSE\_last$ ,  $MSE\_all$ ,  $MAPE\_one$ ,  $MAPE\_last$  and  $MAPE\_all$ . Bi-objective optimization combines the prediction accuracy and the characteristics of the model to find

compromised hyperparameters values, i.e.  $MSE\_one$  and  $model\ complexity$ ,  $MSE\_last$  and  $model\ complexity$ ,  $MSE\_all$  and  $model\ complexity$ ,  $MSE\_one$  and  $MAD$ ,  $MSE\_last$  and  $MAD$ ,  $MSE\_all$  and  $MAD$ ,  $MAPE\_one$  and  $model\ complexity$ ,  $MAPE\_last$  and  $model\ complexity$ ,  $MAPE\_all$  and  $model\ complexity$ ,  $MAPE\_one$  and  $MAD$ ,  $MAOE\_last$  and  $MAD$ ,  $MAPE\_all$  and  $MAD$ .

In Section 3, experiments on real time series data are carried out to compare these different objectives.

## 2.5 Pareto-optimal solution set for bi-objective optimization

For tuning hyperparameters in SVM, the bi-objective optimization problem can be formulated considering the prediction accuracy and the characteristics of the SVM model, as introduced in Section 2.4. A Pareto-front results from the bi-objective optimization and solutions of compromise of the two objectives can be identified (Kim and De Weck, 2005; Blasco *et al.*, 2008). For example, the “Min-Max” method provides a widely accepted way to find a compromised solution (Belegundu and Chandrupatla, 1999). Suppose  $\mathbf{g}_n = (g_{n,1}, g_{n,2})$ ,  $n = 1, 2, \dots, P$  are the solution points in the bi-dimensional Pareto front and  $g_1^{max}, g_2^{max}$  denote the maximal value of the first and second objective functions on this front, respectively, i.e. the maximal accuracy-related objective value and model-characteristics-related objective value, respectively. The relative deviation of each objective of each solution  $(g_{n,1}, g_{n,2})$ ,  $n = 1, 2, \dots, P$  is calculated as  $z_{n,i} = (g_i^{max} - g_{n,i})/g_i^{max}$   $i = 1, 2$  and  $z_n = \min(z_{n,1}, z_{n,2})$  represents the solution  $(g_{n,1}, g_{n,2})$ . The optimal compromised solution  $\mathbf{g}$  for the bi-objective optimization is  $\arg(\max(z_n))$ ,  $n = 1, 2, \dots, P$ . Such a solution represents the “center” of the Pareto-front (Figure 2).

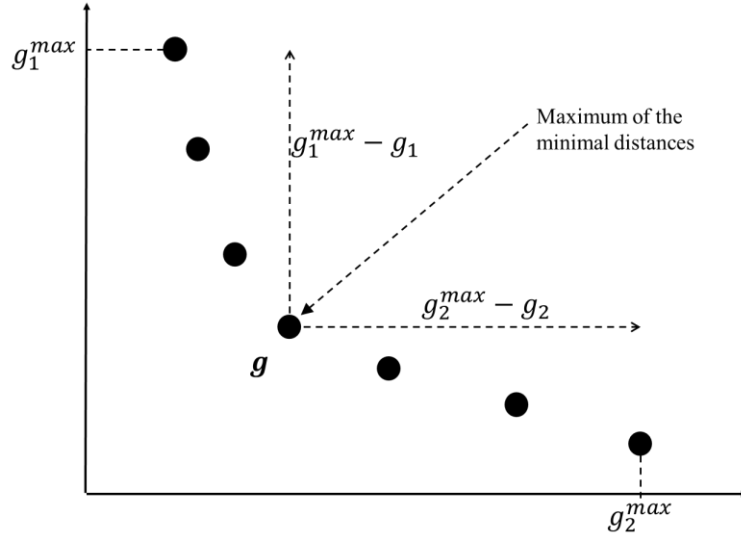


Fig. 2 Best compromised solution  $g$  selected from the bi-dimensional Pareto-front using “Min-Max” method.

### 3. Numerical experiments

Case studies of one synthetic dataset and two real time series datasets are considered with respect to the optimization problem of tuning hyperparameters in recursive SVM. One real case study considers the leakage from the first seal of RCP in NPP and the other one the prediction of wind speed.

The procedure of the experiments is shown in Figure 3. The time series data  $a(t)$  is reconstructed into data points that can be treated with SVM. Partial autocorrelation analysis (Rodgers and Nicewander, 1988) between the different time lag and the next step value is carried out to decide the best number  $M$  of historical values in the input vector. The reconstructed data points are of the form  $([a(t - M + 1), a(t - M + 2), \dots, a(t)], [a(t + 1), a(t + 2), \dots, a(t + N)])$ , for  $N$ -step ahead prediction. The optimization objective for tuning hyperparameters in Figure 3 can be any of the objectives considered in Section 2.4. In order to simplify the experiments, the searching method implemented in these experiments is grid search method and k-fold cross-validation. Grid search method finds the optimal hyperparameters among the given discrete values for each hyperparameter and k-fold cross validation aims at reducing the risk of overfitting during the training part. The best solution is either the one that gives the minimal prediction error for single-objective

optimization or the compromised one that is the center of the Pareto-front for bi-objective optimization, as shown in Section 2.5.

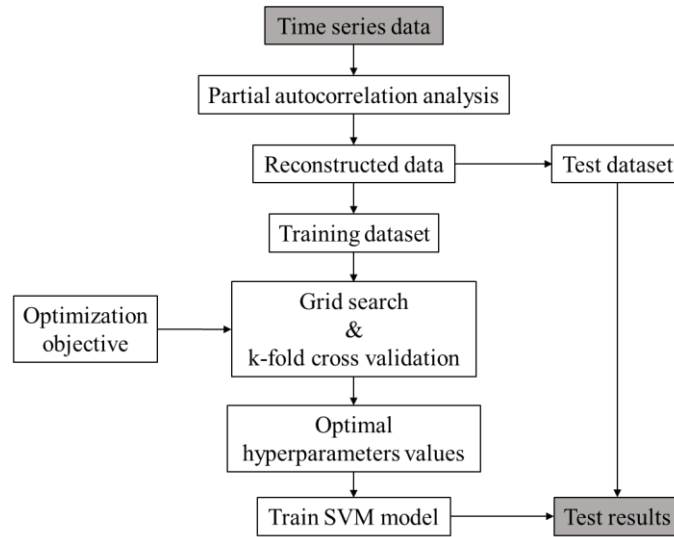


Fig. 3 Flow chart of the experiment.

The prediction errors for the last step ( $N$ th step) on the test dataset are compared among different optimization objectives for hyperparameters tuning. The best objectives for tuning hyperparameters in SVM are identified, for the experiments considered.

### 3.1 Prediction of one synthetic time series data

In this section one synthetic dataset is used to test different objectives for optimizing SVM hyperparameters. The synthetic data, as shown in Figure 4 is a time series data of *sinc* function added with a certain noise. The results are shown in Tables 1 and 2.

In all the Tables in Sections 3.1, 3.2 and 3.3, the first column and the first line list the possible objectives that can be used for tuning hyperparameters during training. The second line shows the prediction results on the test dataset using only accuracy-related objectives, and the third and fourth lines show the prediction results on the test dataset using bi-objective optimization, i.e. combining the prediction accuracy on the training dataset with the model complexity or the MAD.

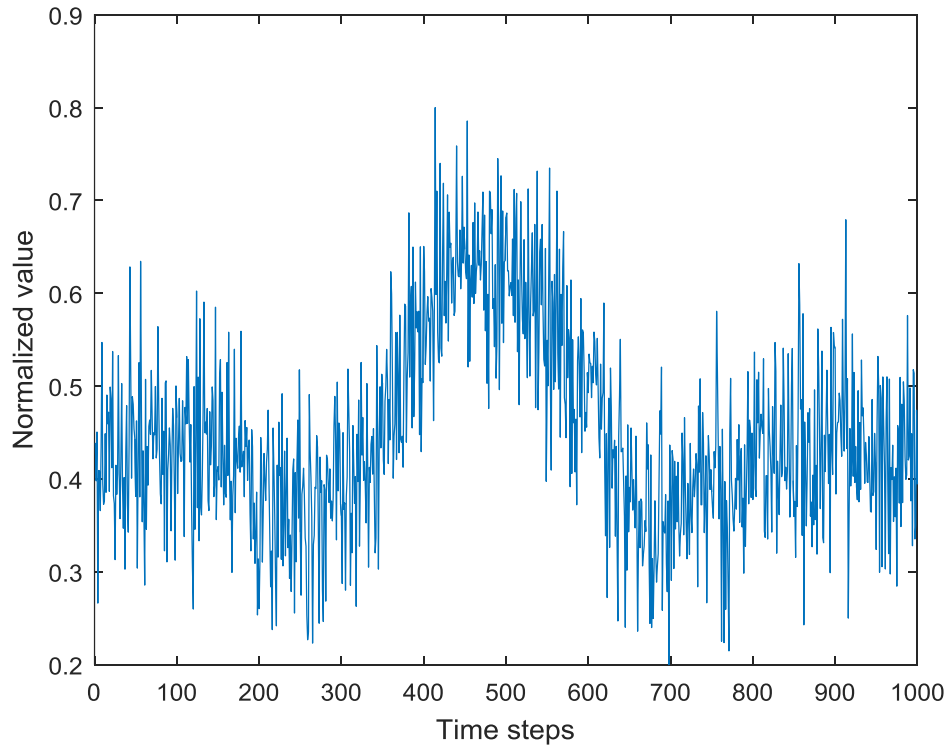


Fig. 1 Normalized values of synthetic time series data.

Table 1 MSE of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for synthetic data prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	0.0658	0.0062	0.0056	0.0415	0.0072	0.0071
Bi-objective with <i>model complexity</i>	0.0415	0.0055	0.0055	0.0168	0.0066	0.0127
Bi-objective with <i>MAD</i>	0.0307	0.0055	0.0054	0.0070	0.0066	0.0127

Table 2 MAPE of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for synthetic data prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	0.4316	0.0319	0.0333	0.2711	0.0326	0.0323
Bi-objective with <i>model complexity</i>	0.2711	0.0328	0.0328	0.1021	0.0314	0.0833
Bi-objective with <i>MAD</i>	0.1941	0.0328	0.0326	0.0034	0.0314	0.0833

### 3.2 Prediction of the leakage from the RCP of a NPP

RCP is a very important component of a NPP, as it must provide sufficient coolant to transport the heat from the reactor core to the steam generator. Radioactive coolant leaked from RCP may endanger the personnel and equipment in NPP. In the extreme case of loss of coolant accident, the reactor core could also melt down if proper safety actions are not taken. Thus, the prediction of leakage is critical for the operation of a NPP.

Figure 5 is a time series dataset which contains 1553 values, measured every four hours. The values are normalized for confidentiality reasons. The objective of the analysis is the ten-step ahead prediction. The leakage in the last 10 4-hour time steps are used to predict the leakage values in the next 10 4-hour time steps. A set of 1534 data points are constructed as explained in Section 2.1. The first 1234 data points are used as training dataset and the remaining 300 data points form the test dataset. The prediction results, including MSE and MAPE of the prediction on the last step for the test dataset are shown separately in Tables 3 and 4, using different objectives for optimizing the SVM hyperparameters during the training process.

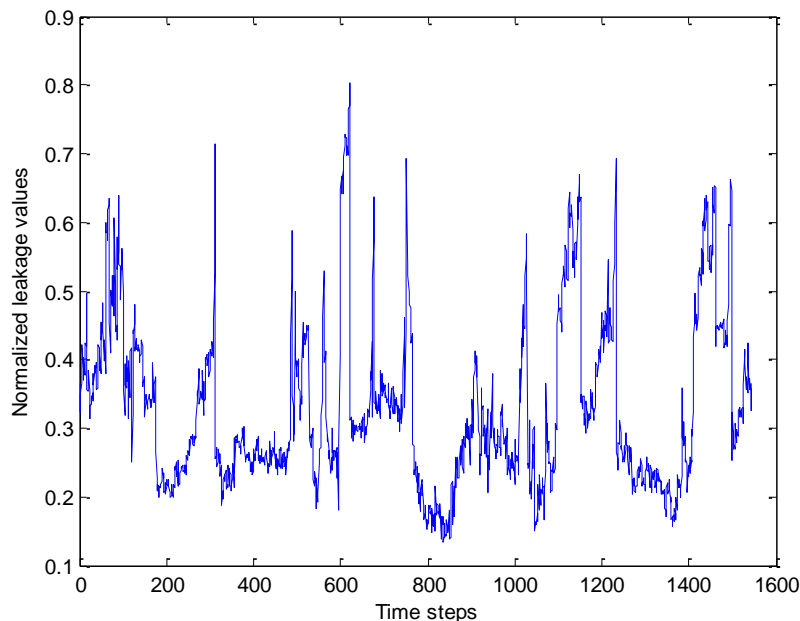


Fig. 5 Normalized values of the leakage measured at time steps of 4 hours.

Table 3 MSE of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for leakage prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	0.4709	0.8397	1.4053	0.0187	0.0166	0.0166
Bi-objective with <i>model complexity</i>	0.0101	0.0098	0.0104	0.0095	0.0094	0.0095
Bi-objective with <i>MAD</i>	0.0103	0.0102	0.0102	0.0187	0.0166	0.0125

Table 4 MAPE of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for leakage prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	2.5034	4.9544	8.2861	0.1067	0.1014	0.1014
Bi-objective with <i>model complexity</i>	0.1212	0.1173	0.1281	0.1073	0.1085	0.1082
Bi-objective with <i>MAD</i>	0.1204	0.1152	0.1252	0.1067	0.1014	0.1000

### 3.2 Wind speed prediction

As more wind power generation is connected to the electrical network, prediction of wind speed becomes more and more important for effective electricity generation planning and management, while guaranteeing network stability and end-users service satisfaction (AK *et al.*, 2013).

In this case study, we consider the wind speed data measured of Regina, Saskatchewan, a region of central Canada (Canadian Weather Office, 2012) over a period of two months, from 1st of February 2012 to 31st of March 2012. According to a partial autocorrelation analysis, wind speed in the last two 1-hour time steps are used as inputs to predict the wind at the next five 1-hour time steps. The total data set includes 1438 values (as shown in Figure 6) (Ak *et al.*, 2013). The first 800 of the total constructed 1432 data points are used as training dataset and the rest as test dataset. The prediction results are compared by using different objectives for optimizing the SVM hyperparameters during the training process, as shown in Tables 5 and 6.



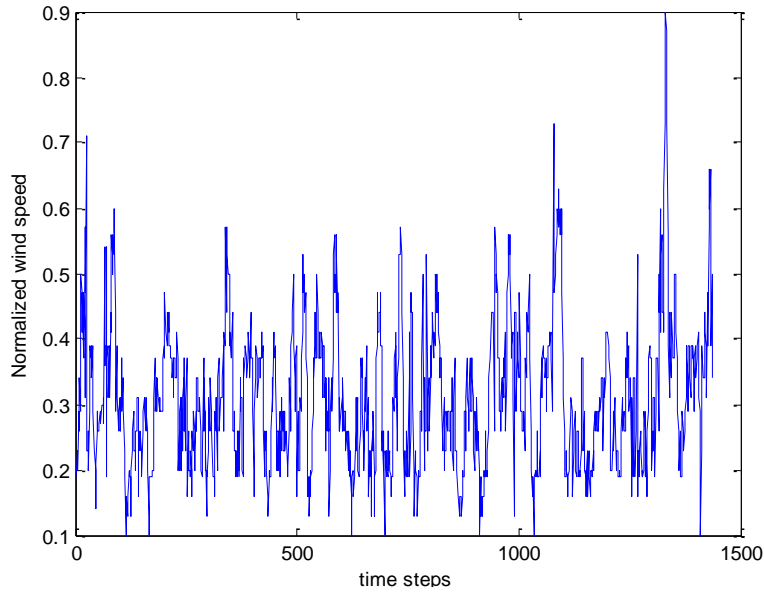


Fig. 6 Normalized values of the wind speed with a time step of 1 hour.

Table 5 MSE ( $10^{-4}$ ) of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for wind speed prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	9.5748	5.1991	5.2952	9.5748	6.2335	6.2335
Bi-objective with <i>model complexity</i>	7.7724	6.0290	6.3404	5.9548	5.6379	5.6379
Bi-objective with <i>MAD</i>	5.8448	5.2072	5.2816	6.0042	5.3199	5.3199

Table 6 MAPE of the prediction results on the last step for the test dataset with respect to the different objectives used for optimizing the SVM hyperparameters for wind speed prediction.

	<i>MSE_one</i>	<i>MSE_all</i>	<i>MSE_last</i>	<i>MAPE_one</i>	<i>MAPE_all</i>	<i>MAPE_last</i>
Single-objective	0.0062	0.0048	0.0049	0.0062	0.0050	0.0050
Bi-objective with <i>model complexity</i>	0.0059	0.0055	0.0072	0.0052	0.0058	0.0058
Bi-objective with <i>MAD</i>	0.0050	0.0049	0.0050	0.0050	0.0047	0.0047

### 3.3 Discussions

Several insights can be drawn from the results of the three case studies.

(1) Single-objective optimization based on MSE for tuning SVM hyperparameters for leakage prediction, leads to prediction results that are not satisfactory, as the MSE during the search for optimal hyperparameters values is strongly influenced by the bad predictions of only few training data points. This is one drawback of using MSE as accuracy measure, because the MSE can be dominated by relatively big errors on only a small number of data points and the optimization efforts are, then, specifically dedicated to improving these errors. Figure 7 is an example of this drawback using MSE as the objective for tuning SVM hyperparameters. The solid line without marker is the output of the training data points. If the MSE on the training data points is the sole objective for tuning SVM hyperparameters, the first set of hyperparameters with a MSE of 0.0013 is slightly better than the second set of hyperparameters with a MSE of 0.0018. But the fact is that the second set of hyperparameters gives better results than the first set, except for the really bad prediction on the 15<sup>th</sup> and 24<sup>th</sup> data points. In such case, the MSE as objective does not choose the best set (the second set of hyperparameters).

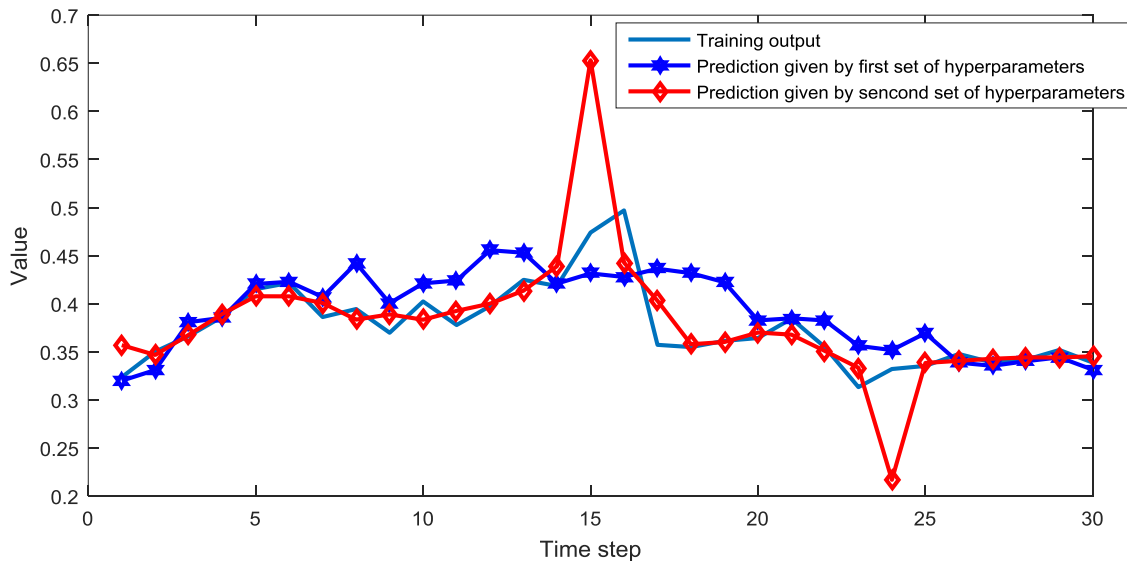


Fig. 7 Illustration of the possible drawback of using MSE as objective for tuning SVM hyperparameters.

(2) For predictions of the three case studies, Figures 8, 9, 10, 11, 12 and 13 show the comparisons of the prediction results of each line of Tables 1, 2, 3, 4, 5 and 6, i.e. the prediction accuracy obtained by SVM when tuning the hyperparameters calculated on

different time horizons as explained in Section 2.3.1 e.g. comparing the prediction accuracy obtained by using bi-objective optimization with model complexity and MSE with single-objective MSE at the first step, at all the prediction steps or at the last step as objective, for tuning hyperparameters. It is observed that compared to using only the prediction accuracy on the one-step ahead prediction, for most of the cases the prediction accuracy on the test dataset can be improved by using the prediction accuracy (MSE or MAPE) on all the steps or on the last step as optimization objective for tuning the SVM hyperparameters. This seems to confirm the observation in Taieb *et al.* (2010) that the relation between the input and the one-step ahead prediction cannot guarantee the relation between the input and the multi-step ahead prediction. If the training focuses only on minimizing the prediction error of the one-step ahead prediction, the trained model can be trapped in overfitting such prediction, and the prediction accuracy on the following steps may decrease. From Figures 8, 9, 10, 11, 12 and 13, it can also be observed that in the experiments considered, using the prediction accuracy on all steps as objective in the optimization problem for tuning the SVM hyperparameters leads, in general, to better results than using the prediction accuracy only on the last step as objective of the optimization.

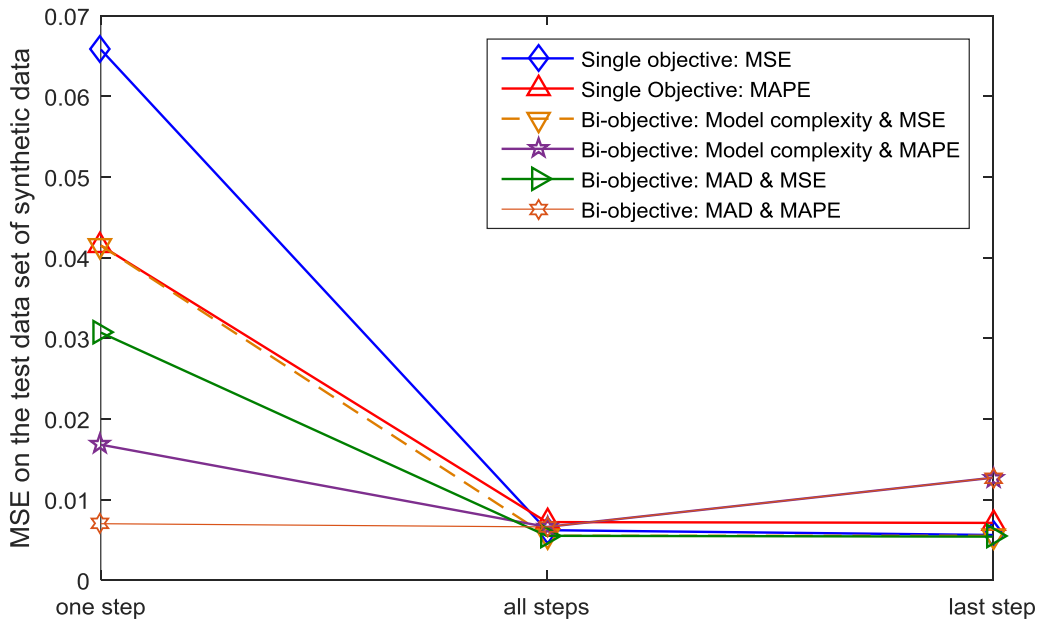


Fig. 8 Comparison of MSE on the test dataset of synthetic data prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

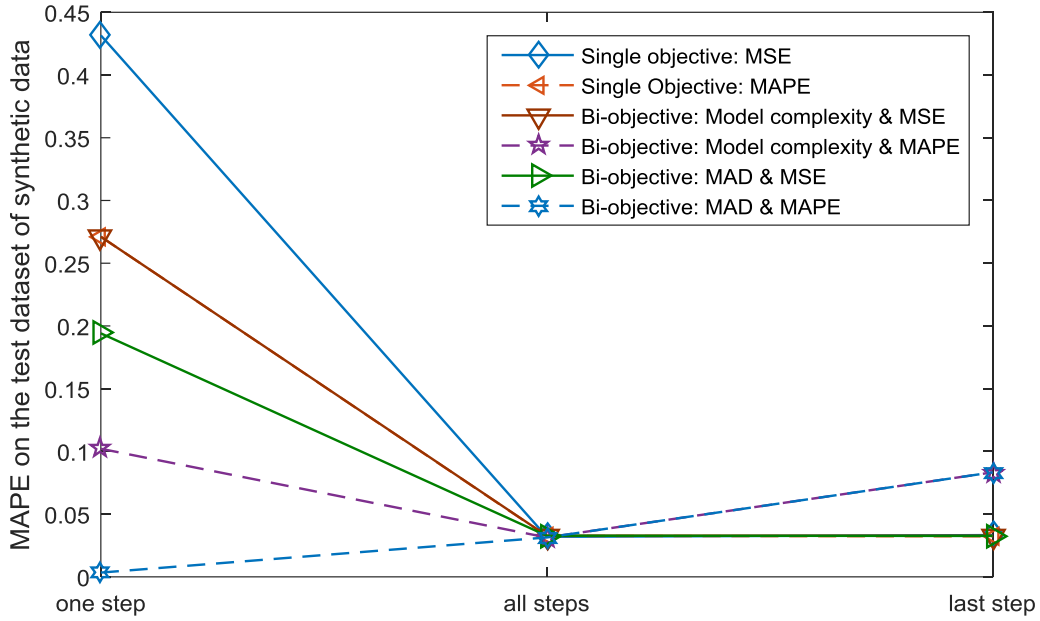


Fig. 9 Comparison of MAPE on the test dataset of synthetic data prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

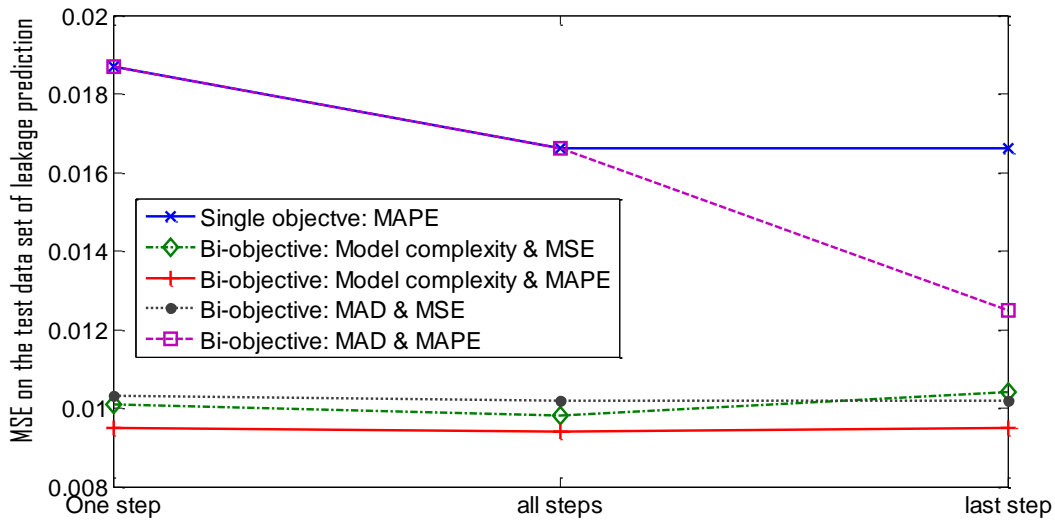


Fig. 10 Comparison of MSE on the test dataset of leakage prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

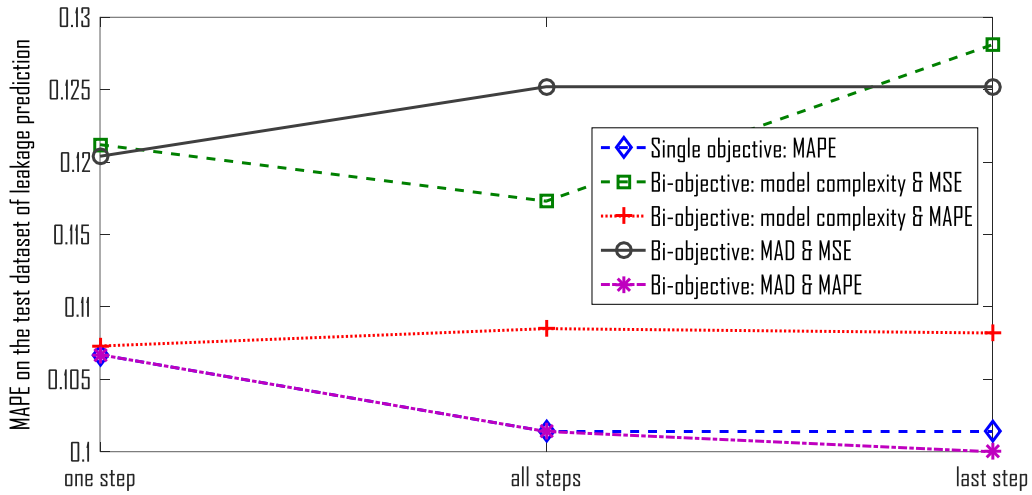


Fig. 11 Comparison of MAPE on the test dataset of leakage prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

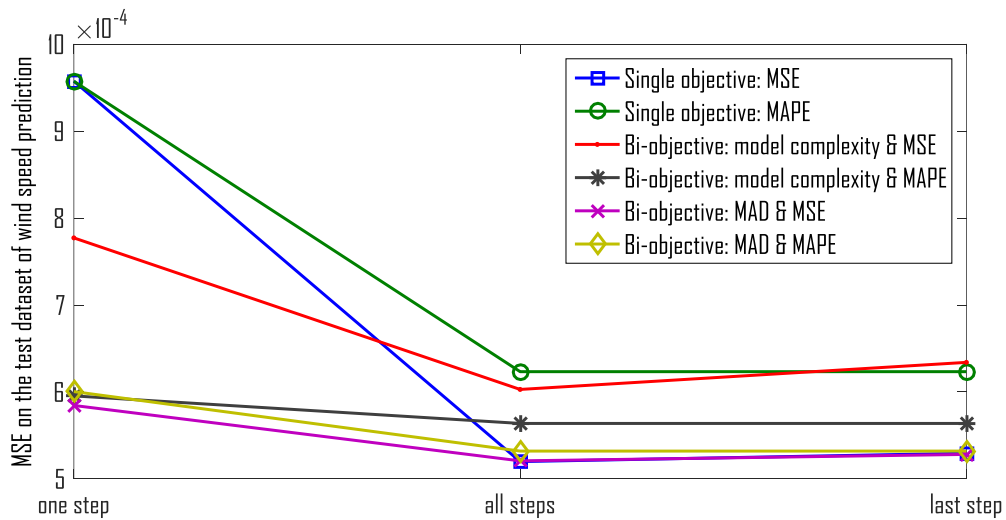


Fig. 12 Comparison of MSE on the test dataset of wind speed prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

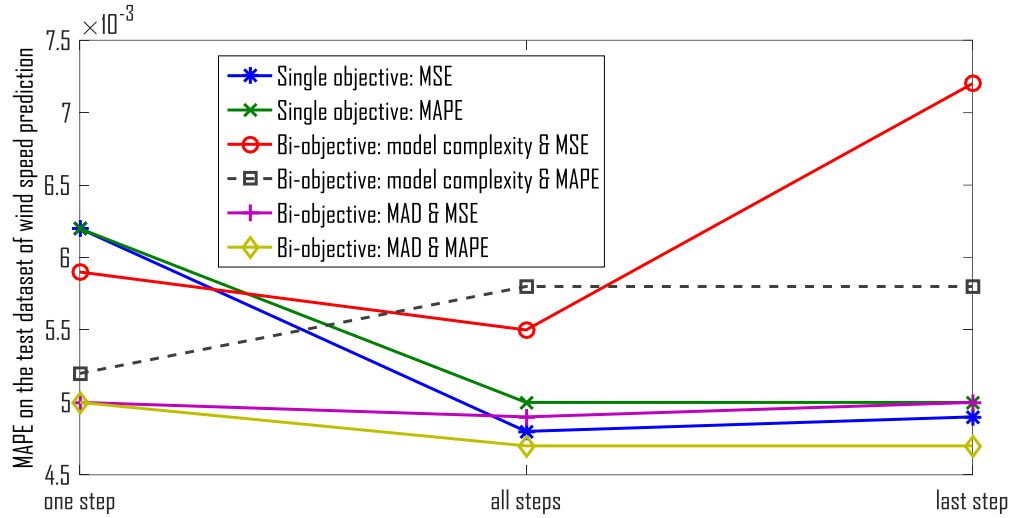


Fig. 13 Comparison of MAPE on the test dataset of wind speed prediction considering the prediction accuracy on different time horizons, i.e. one step, all steps and last step as optimization objective for tuning hyperparameters.

(3) Figures 14, 15, 16, 17, 18 and 19 show the comparisons of the prediction results on the test datasets using single-objective and bi-objective optimization for the SVM hyperparameters tuning, with the same prediction-accuracy-related objective, e.g. MSE (i.e. MSE on all steps, MSE on all steps and model complexity, MSE on all steps and MAD). Generally, the bi-objective optimization gives better results on the test datasets than the single-objective optimization, as simply minimizing the prediction accuracy on the training dataset can fall into overfitting of the training dataset. For the two bi-objective optimizations, the combination of prediction accuracy with MAD, which characterizes the robustness of the model, gives more stable and accurate results than the combination with model complexity. This can be explained by the inherent meaning of the two model characteristics: MAD characterizes the robustness of the model against small noises, which can decrease the influence of the prediction error that may cumulate during the recursive prediction process; model complexity characterizes the generalization ability of the model, which guarantees the prediction performance for different inputs.

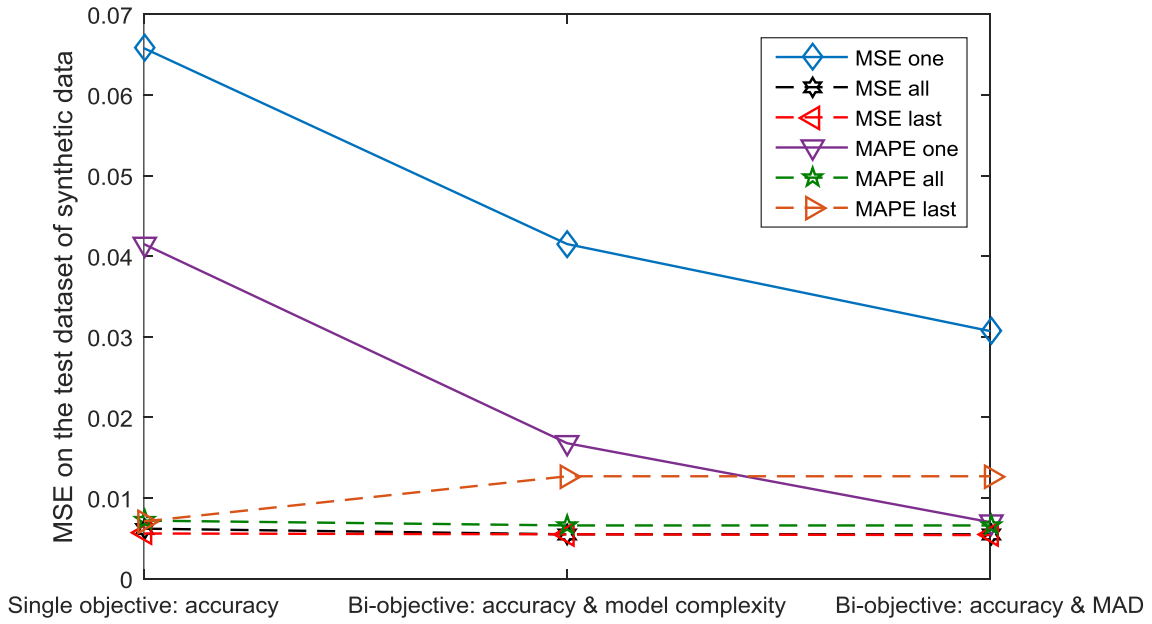


Fig. 14 Comparison of MSE on the test dataset of synthetic data prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.

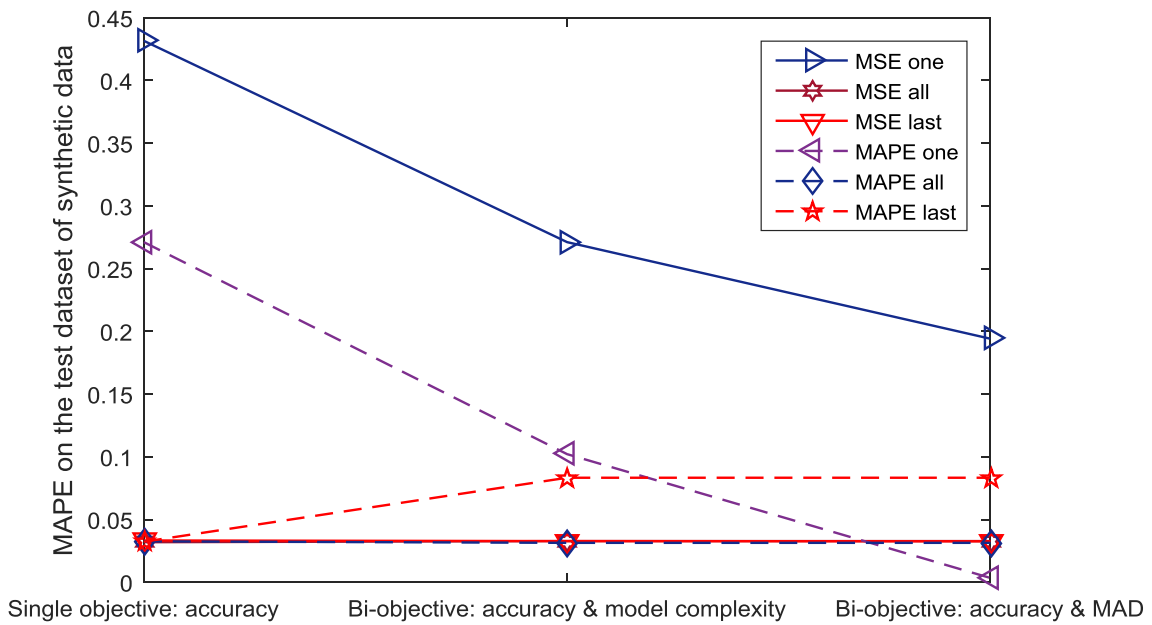


Fig. 15 Comparison of MAPE on the test dataset of synthetic data prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.

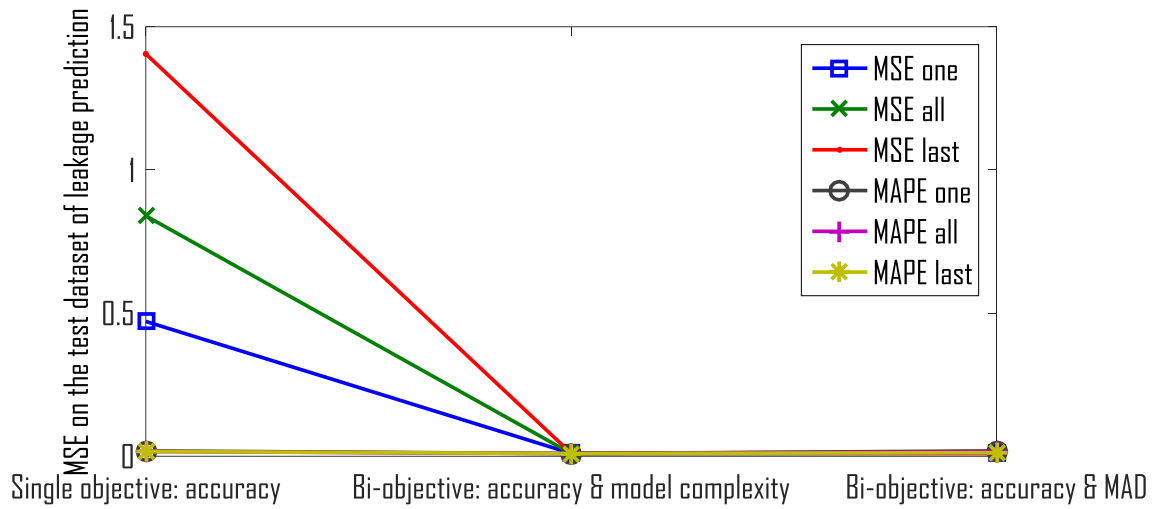


Fig. 16 Comparison of MSE on the test dataset of leakage prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.

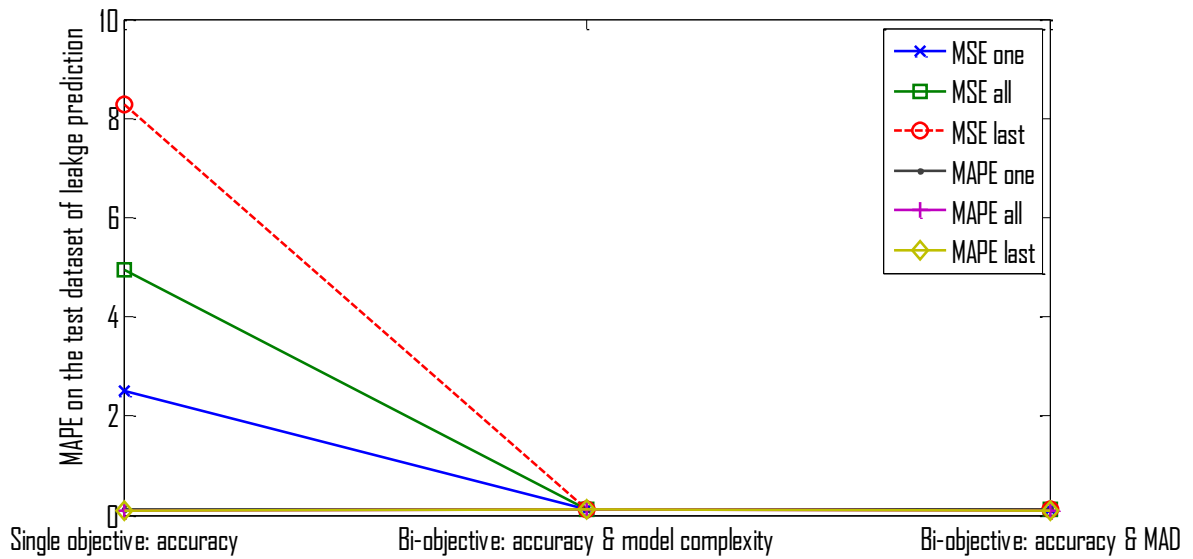


Fig. 17 Comparison of MAPE on the test dataset of leakage prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.



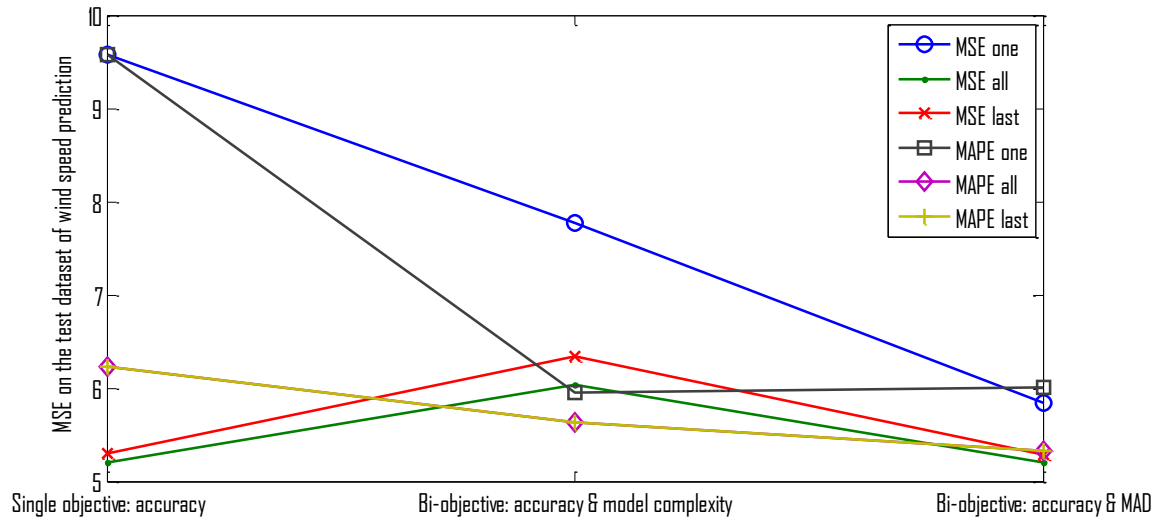


Fig. 18 Comparison of MSE on the test dataset of wind speed prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.

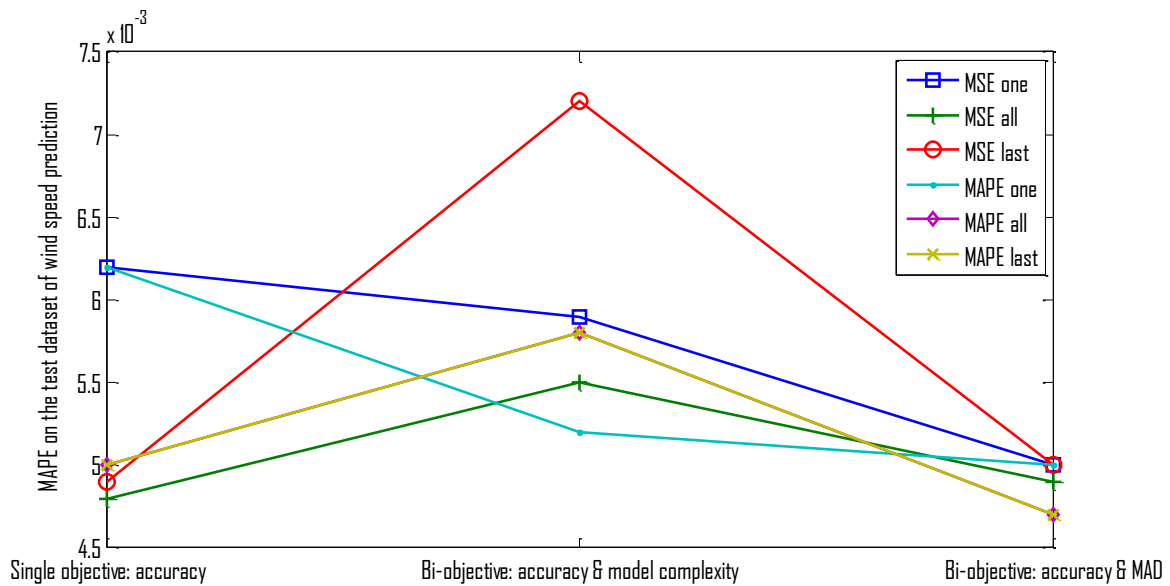


Fig. 19 Comparison of MAPE on the test dataset of wind speed prediction considering accuracy-related measures in combination with model complexity and MAD, as optimization objectives for tuning SVM hyperparameters.

(4) From the discussions above, we can draw the conclusion that for recursive multi-step ahead prediction by SVM, the combination of model complexity or MAD with prediction

accuracy on all the steps as objectives of the optimization for hyperparameters tuning give the best results, with MAD guaranteeing more robustness than model complexity.

## 4. Conclusions

In this paper, we have considered the objectives for optimizing the hyperparameters of SVM for recursive multi-step ahead prediction. The candidate objectives have been presented and discussed, distinguishing between accuracy-related and model-characteristics-related objectives, and considering different step horizons (single step, all steps and last step). Experiments on one synthetic time series data and two real time series data (leakage from RCP in NPP and wind speed) show that the bi-objective optimization combining the MAD and the prediction accuracy (MSE or MAPE) on all the steps is a preferable choice for SVM hyperparameters tuning for recursive multi-step ahead prediction.

## References

- Aich, U., Banerjee, S., 2013. Modeling of EDM responses by support vector machine regression with parameters selected by particle swarm optimization. *Appl. Math. Model.* 38, 2800–2818. doi:10.1016/j.apm.2013.10.073
- Ak, R., Li, Y., Vitelli, V., Zio, E., López Droguett, E., Magno Couto Jacinto, C., 2013. NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Expert Syst. Appl.* 40, 1205–1212. doi:10.1016/j.eswa.2012.08.018
- Asefa, T., Kemblowski, M., McKee, M., Khalil, A., 2006. Multi-time scale stream flow predictions: The support vector machines approach. *J. Hydrol.* 318, 7–16. doi:10.1016/j.jhydrol.2005.06.001
- Aydin, I., Karakose, M., Akin, E., 2011. A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Appl. Soft Comput.* 11, 120–129. doi:10.1016/j.asoc.2009.11.003
- Bao, Y., Xiong, T., Hu, Z., 2014. Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing* 129, 482–493. doi:10.1016/j.neucom.2013.09.010
- Benkedjough, T., Medjaher, K., Zerhouni, N., Rechak, S., 2013. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Eng. Appl. Artif. Intell.* 26, 1751–1760. doi:10.1016/j.engappai.2013.02.006
- Benkedjough, T., Medjaher, K., Zerhouni, N., Rechak, S., 2015. Health assessment and life prediction of cutting tools based on support vector regression. *J. Intell. Manuf.* 26, 213–223. doi:10.1007/s10845-013-0774-6
- Blasco, X., Herrero, J.M., Sanchis, J., Martínez, M., 2008. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Inf. Sci. (Ny)*. 178, 3908–3924. doi:10.1016/j.ins.2008.06.010
- Chang C-C, Chou S-H (2015) Tuning of the hyperparameters for L2-loss {SVMs} with the {RBF} kernel by the maximum-margin principle and the jackknife technique. *Pattern Recognit* 48:3983–3992. doi: <http://dx.doi.org/10.1016/j.patcog.2015.06.017>

Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., 2002. Choosing multiple parameters for support vector machines. *Mach. Learn.* 46, 131–159. doi:10.1023/A:1012450327387

Chen, R., Liang, C.-Y., Hong, W.-C., Gu, D.-X., 2015. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Appl. Soft Comput.* 26, 435–443. doi:10.1016/j.asoc.2014.10.022

Chou, J.-S.S., Cheng, M.-Y.Y., Wu, Y.-W.W., Pham, A.-D.D., 2014. Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification. *Expert Syst. Appl.* 41, 3955–3964. doi:10.1016/j.eswa.2013.12.035

Ding, S., Han, Y., Yu, J., & Gu, Y. (2013). A fast fuzzy support vector machine based on information granulation. *Neural Computing and Applications*, 23(1), 139-144.

Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V., 1997. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* 9, 155–161. doi:10.1.1.10.4845

Hu, Z., Bao, Y., Xiong, T., 2014. Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression. *Appl. Soft Comput.* 25, 15–25. doi:10.1016/j.asoc.2014.09.007

Igel, C., 2005. Multi-objective model selection for support vector machines. In *Evolutionary Multi-Criterion Optimization*, 534-546. Springer Berlin Heidelberg.

Jardine, a. K.S., Lin, D., Banjevic, D., 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech. Syst. Signal Process.* 20, 1483–1510. doi:10.1016/j.ymsp.2005.09.012

Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P., 2015. Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Trans. Ind. Electron.* 62, 647-656. doi:10.1109/TIE.2014.2327917

Karush, W. 1939. Minima of functions of several variables with inequalities as side constraints (Doctoral dissertation, Master's thesis, Dept. of Mathematics, Univ. of Chicago).

Kim, I.Y., De Weck, O.L., 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct. Multidiscip. Optim.* 29, 149–158. doi:10.1007/s00158-004-0465-1

Kuhn, H.W., Tucker, A., 1951. Nonlinear Programming, Proceedings of the Second Symposium on Mathematical Statistics and Probability. doi:10.1007/BF01582292

Lei Y, He Z, Zi Y, Hu Q (2007) Fault diagnosis of rotating machinery based on multiple ANFIS combination with GAs. *Mech Syst Signal Process* 21:2280–2294. doi: 10.1016/j.ymsp.2006.11.003

Lei Y, Liu Z, Wu X, et al. (2015) Health condition identification of multi-stage planetary gearboxes using a mRVM-based method. *Mech Syst Signal Process* 60-61:289–300. doi: 10.1016/j.ymsp.2015.01.014

Lin, J.-Y., Cheng, C.-T., Chau, K.-W., 2006. Using support vector machines for long-term discharge prediction. *Hydrol. Sci. J.* 51, 599–612. doi:10.1623/hysj.51.4.599

Liu, R., Liu, E., Yang, J., Li, M., Wang, F., 2006. Optimizing the hyper-parameters for SVM by combining evolution strategies with a grid search. *Lect. Notes Control Inf. Sci.* 344, 712–721. doi:10.1007/11816492\_87

Liu, S., Xu, L., Jiang, Y., Li, D., Chen, Y., Li, Z., 2014. A hybrid WA-CPSO-LSSVR model for dissolved oxygen content prediction in crab culture. *Eng. Appl. Artif. Intell.* 29, 114–124. doi:10.1016/j.engappai.2013.09.019

Min, J.H., Lee, Y.-C., 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Syst. Appl.* 28, 603–614. doi:10.1016/j.eswa.2004.12.008

Müller, K., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V., 1997. Predicting Time Series with Support Vector Machines, in: *Artificial Neural Networks ICANN97*. pp. 999–1004.

- Namdari, M., Jazayeri-Rad, H., 2014. Incipient fault diagnosis using support vector machines based on monitoring continuous decision functions. *Eng. Appl. Artif. Intell.* 28, 22–35. doi:10.1016/j.engappai.2013.11.013
- Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., Dietmayer, K., 2013. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *J. Power Sources* 239, 680–688. doi:10.1016/j.jpowsour.2012.11.146
- Palm, R., 2007. Multiple-step-ahead prediction in control systems with Gaussian process models and TS-fuzzy models. *Eng. Appl. Artif. Intell.* 20, 1023–1035. doi:10.1016/j.engappai.2007.02.003
- Vichare, N.M., Pecht, M.G., 2006. Prognostics and health management of electronics. *IEEE Trans. Components Packag. Technol.* 29, 222–229. doi:10.1109/TCAPT.2006.870387
- Rodgers, J.L., Nicewander, W.A., 1988. Thirteen Ways to Look at the Correlation Coefficient. *Am. Stat.* 42, 59 – 66. doi:10.2307/2685263
- Saha, B., Goebel, K., Poll, S., Christophersen, J., 2009. Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework. *IEEE Trans. Instrum. Meas.* 58, 291–297.
- Sapankevych, N., Sankar, R., 2009. Time Series Prediction Using Support Vector Machines: A Survey. *IEEE Comput. Intell. Mag.* 4, 24 –38. doi:10.1109/MCI.2009.932254
- Shen L, Chen H, Yu Z, et al. (2016) Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Syst* -. doi: <http://dx.doi.org/10.1016/j.knosys.2016.01.002>
- Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A., 2007. Methodology for long-term prediction of time series. *Neurocomputing* 70, 2861–2869. doi:10.1016/j.neucom.2006.06.015
- Ben Taieb, S., Sorjamaa, A., Bontempi, G., 2010. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing* 73, 1950–1957. doi:10.1016/j.neucom.2009.11.030
- Wang, X., Han, M., 2015. Improved extreme learning machine for multivariate time series online sequential prediction. *Eng. Appl. Artif. Intell.* 40, 28–36. doi:10.1016/j.engappai.2014.12.013
- Wang, Z., He, X., Gao, D., & Xue, X. (2013). An efficient Kernel-based matrixized least squares support vector machine. *Neural Computing and Applications*, 22(1), 143-150.
- Wang S, Han Z, Liu F, Tang Y (2015) Nonlinear system identification using least squares support vector machine tuned by an adaptive particle swarm optimization. *Int J Mach Learn Cybern* 6:981–992. doi: 10.1007/s13042-015-0403-0
- Widodo, A., Shim, M.C., Caesarendra, W., Yang, B.S., 2011. Intelligent prognostics for battery health monitoring based on sample entropy. *Expert Syst. Appl.* 38, 11763–11769. doi:10.1016/j.eswa.2011.03.063
- Wu, C.-H., Tzeng, G.-H., Goo, Y.-J., Fang, W.-C., 2007. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Syst. Appl.* 32, 397–408. doi:10.1016/j.eswa.2005.12.008
- Wu, C.L., Chau, K.W., 2013. Prediction of rainfall time series using modular soft computing methods. *Eng. Appl. Artif. Intell.* 26, 997–1007. doi:10.1016/j.engappai.2012.05.023
- Zhang, X., Qiu, D., Chen, F., 2015. Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis. *Neurocomputing* 149, Part , 641–651. doi:<http://dx.doi.org/10.1016/j.neucom.2014.08.010>
- Zhao, Z., Quan, Q., Cai, K.-Y., 2014. A Profust Reliability Based Approach to Prognostics and Health Management. *IEEE Trans. Reliab.* 63, 26–41. doi:10.1109/TR.2014.2299111
- Zio, E., 2012. Prognostics and health management of industrial equipment. *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, 333-356.
- Zio, E., Di Maio, F., 2012. Fatigue crack growth estimation by relevance vector machine. *Expert Syst. Appl.* 39, 10681–10692. doi:10.1016/j.eswa.2012.02.199