Nonlinear regression in environmental sciences by support vector machines combined with evolutionary strategy

Aranildo R. Lima*[a], Alex J. Cannon[a], William W. Hsieh[a]

[a]*Department of Earth and Ocean Sciences, University of British Columbia, Vancouver, BC V6T 1Z4, Canada*

Abstract

A hybrid algorithm combining support vector regression with evolutionary strategy (SVR-ES) is proposed for predictive models in the environmental sciences. SVR-ES uses uncorrelated mutation with p step sizes to find the optimal SVR hyper-parameters. Three environmental forecast datasets used in the WCCI-2006 contest – surface air temperature, precipitation and sulphur dioxide concentration – were tested. We used multiple linear regression (MLR) as benchmark and a variety of machine learning techniques including bootstrap-aggregated ensemble artificial neural network (ANN), SVR-ES, SVR with hyper-parameters given by the Cherkassky-Ma estimate, the M5 regression tree, and random forest (RF). We also tested all techniques using stepwise linear regression (SLR) first to screen out irrelevant predictors. We concluded that SVR-ES is an attractive approach because it tends to outperform the other techniques and can also be implemented in an almost automatic way. The Cherkassky-Ma estimate is a useful approach for minimizing the mean absolute error and saving computational time related to the hyper-parameter search. The ANN and RF are also good options to outperform multiple linear regression (MLR). Finally, the use of SLR for predictor selection can dramatically reduce computational time and often help to enhance accuracy.

Keywords: Support Vector Machine, Evolutionary Strategy, Artificial Neural Network, Atmospheric Forecasts, Hybrid Systems, Machine Learning

*Corresponding author. Department of Earth and Ocean Sciences, University of British Columbia, 6339 Stores Road, Vancouver, BC V6T 1Z4, Canada, Tel.: +1-604-822-5691; Fax.: +1-604-822-6088, E-mail: arodrigu@eos.ubc.ca

1. Introduction

The main idea of machine learning is that computer algorithms are capable of automatically distilling knowledge from data. From this knowledge they can construct models capable of making predictions from novel data in the future. However, environmental modeling problems are typically very noisy (Cawley et al., 2007), hence it is not easy to build successful predictive models.

Due to data modeling complexity, even for a particular machine learning method, there is often more than one way to build the model (Zhang, 2007). To build a successful predictive model, the correct adjustment of the model hyper-parameters is necessary. For example, in artificial neural network (ANN) models, the number of hidden processing units, the choice of activation functions and the regularization parameter all need to be specified (Haykin, 1998; Hsieh, 2009).

Similarly, in support vector machines (SVM) for regression (SVR), typically two or three hyper-parameters have to be tuned, such as the cost of constraint violation (C), the insensitive-loss (ϵ) and, if the Gaussian function is used as the kernel function, the width of the Gaussian (γ). In theory, the establishment of these hyper-parameters requires an optimal search in full state space.

Much effort has been spent on improving the efficiency of the SVR hyper-parameter search (Cherkassky and Ma, 2004; Friedrichs and Igel, 2005; Fan et al., 2005; Huang and Wang, 2006; Lin et al., 2008). The most common approach is the simple grid search (Fan et al., 2005; Ortiz-García et al., 2009; Pino-Mejías et al., 2010; Zeng et al., 2011), an exhaustive approach that is computationally expensive. Furthermore, the hyper-parameters are varied by fixed step-sizes through a wide range of values, limiting the search to discrete values.

Some of the most promising approaches in tuning the SVM hyper-parameters are based on evolutionary algorithms (EA), e.g. genetic algorithms (GA) (Pai and Hong, 2005; Huang and Wang, 2006; Tripathi et al., 2006), particle swarm optimization (PSO) (Lin et al., 2008), and evolutionary strategies (ES) (Friedrichs and Igel, 2005). However, these approaches need a set of adjustable parameters called EA parameters (Smit and Eiben, 2009) and typically there are more EA parameters to adjust than the number of hyper-parameters. For example, GA has population size, mutation rate, crossover

rate, number of generations, etc., and PSO has population size, acceleration coefficients, inertia weight, number of generations, etc. Kramer et al. (2007) showed that the choice of the EA parameters has a decisive impact in the final results and can undervalue the algorithm performance. In other words, to use a GA it is necessary to adjust at least four parameters, which is more than the three SVR hyper-parameters when using a Gaussian kernel.

Another important issue is the data quality upon which machine learning methods operate. Indeed, machine learning algorithms may produce less accurate and less understandable results if the data are inadequate or contain extraneous and irrelevant information (Hall and Smith, 1996). It is also possible to make the modeling process less time consuming and sometimes more accurate by removing predictors that are irrelevant or redundant with respect to the task to be learned.

In this paper, our main goals are: (i) reduce the number of hyper-parameters which require estimation; (ii) use an accurate initialization of the hyper-parameter search; and (iii) discard irrelevant and redundant predictors. We propose a hybrid algorithm called SVR-ES which uses a simple evolutionary strategy called "uncorrelated mutation with p step sizes" (Eiben and Smith, 2003) to find the optimal SVR hyper-parameters. We also combine the SVR-ES with stepwise linear regression (SLR) (Draper and Smith, 1998) to screen out irrelevant predictors.

Three environmental forecast problems used in the WCCI-2006 contest – surface air temperature (TEMP), precipitation (PRECIP) and sulphur dioxide concentration (SO2) – are tested (Cawley et al., 2007). These three datasets contain different amounts of non-linearity and noise. Several other machine learning techniques successfully used in the environmental forecast problems are considered, including bootstrap-aggregated ensemble ANN (Cannon and Lord, 2000; Krasnopolsky, 2007), SVR using the Cherkassky-Ma hyper-parameter estimates (Cherkassky and Ma, 2004), the M5 regression tree (Quinlan, 1992; Solomatine and Xue, 2004; Haupt et al., 2009) and random forest (RF) (Breiman, 2001; Pino-Mejías et al., 2010). We also use SLR with these techniques to prescreen and reduce the number of predictors.

Section 2 describes the data sets used in our study. The forecasting methods are presented in Sections 3 and 4. Results and discussion of the experiments are given in

Section 5, followed by summary and conclusion in Section 6.

2. Data Description

Environmental data normally contain properties that are difficult to model by common regression techniques. For example, response variables may be strictly non-negative, highly skewed, non-Gaussian distributed and heteroscedastic (Cawley et al., 2007). Some examples are the modeling of SO_2 air pollution (Kurt et al., 2008) or statistical downscaling of temperature and precipitation (Schoof and Pryor, 2001).

We used as benchmark three datasets which were originally used at the WCCI-2006 Predictive Uncertainty in Environmental Modeling Challenge. These benchmarks, characterized by a non-Gaussian, heteroscedastic variance structure (Cawley et al., 2007), are freely available from the challenge website (`http://theoval.cmp.uea.ac.uk/~gcc/competition/`).

Predicting precipitation accurately is still one of the most difficult tasks in meteorology (Fritsch et al., 1998; Kuligowski, 1998; Strangeways, 2007). Factors responsible for the difficulty in predicting precipitation are e.g. the chaotic nature of the atmosphere and the complexity of the processes involved in its creation (Fritsch et al., 1998), seasonal variations (Wallace and Hobbs, 2006), non-stationary statistical behavior (Von Storch and Zwiers, 2001), difficulties in precipitation measurements including problems with rain gauges, radar and satellites (Strangeways, 2007) and the limited temporal and spatial scales of global circulation models (GCMs) (Kuligowski, 1998).

To forecast precipitation at regional scale, GCM outputs cannot be used directly due to coarse spatial resolution. For example, GCMs do not provide information on the spatial structure of temperature and precipitation in areas of complex topography and land use distribution. To use the output of a GCM for this task, statistical downscaling models are often used (Hashmi et al., 2009).

In the PRECIP benchmark the input variables are large-scale circulation information, such as might be obtained from a GCM, and the target is daily precipitation data recorded at Newton Rigg, a relatively wet station located in the northwest of the United Kingdom. This benchmark is an example where the response variable is skewed to the right. The PRECIP dataset contains 106 predictors and 10,546 daily patterns.

With similar large-scale circulation features to those used for the PRECIP benchmark, the TEMP benchmark is another statistical downscaling problem, in this case one in which the target is the more normally distributed daily maximum temperature at the Writtle station in the southeast of the United Kingdom. The TEMP dataset has 106 predictors and 10,675 daily patterns.

Air pollution is a major environmental problem in cities. Local emissions and topographic factors, allied with meteorological conditions such as high atmospheric pressure and temperature inversions, cause poor dispersion of atmospheric pollutants due the stagnant conditions. Human health problems (both short term and chronic problems) can occur when pollutant concentrations exceed the allowable limits. Therefore predicting the concentration of pollutants such as sulfur dioxide (SO_2) is crucial to providing proper actions and control strategies in extreme situations (Kurt et al., 2008).

The target of the SO2 dataset is the SO_2 concentration in urban Belfast. In order to forecast the SO_2 concentration twenty-four hours in advance, meteorological conditions and current SO_2 levels were used as input variables. Similar to PRECIP, SO2 is a right skewed variable with a heavy tail. The SO2 dataset has 27 predictors and 22,956 hourly patterns, more than double the number of patterns in PRECIP and TEMP.

Irrelevant predictors can unnecessarily increase the time needed for learning a sufficiently accurate forecast model and in many cases also increase the size of the search space. To reduce the number of predictor variables, we use the well-known prescreening technique of SLR (Hocking, 1976; Venables and Ripley, 2002; Hastie et al., 2009).

3. Support Vector Regression

Support vector machines (SVM) have been widely used in the environmental sciences for classification and regression problems (Pino-Mejías et al., 2010; Tripathi et al., 2006). SVM were originally designed for nonlinear classification problems, then extended to nonlinear regression problems (SVR) (Muller et al., 1997).

Suppose we are given the training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with n patterns. After mapping the input pattern $\mathbf{x}$ into a higher dimensional feature space using a nonlinear mapping function $\mathbf{\Phi}$, the nonlinear regression problem between $\mathbf{x}$ and y can be converted

to a linear regression problem between $\mathbf{\Phi}(\mathbf{x})$ and y, i.e.

$$f(\mathbf{x};\mathbf{w}) = \langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}) \rangle + b, \tag{1}$$

where $\langle , \rangle$ denotes the inner product, and $\mathbf{w}$ and b are the regression coefficients obtained by minimizing the error between f and the observed values of y. Instead of the commonly used mean squared error (MSE) norm, SVR uses the ϵ-insensitive error norm to measure the error between f and y,

$$|f(\mathbf{x};\mathbf{w}) - y|_\epsilon = \begin{cases} 0, & \text{if } |f(\mathbf{x};\mathbf{w}) - y| < \epsilon \\ |f(\mathbf{x};\mathbf{w}) - y| - \epsilon, & \text{otherwise,} \end{cases} \tag{2}$$

i.e., small errors ($|f - y| < \epsilon$) are ignored, whereas for large errors, the error norm approximates the mean absolute error (MAE). A key issue is that an error norm based on MAE is more robust to outliers in the data than the MSE. Using pattern data ($\mathbf{x}_i$, y_i), the $\mathbf{w}$ and b coefficients are estimated by minimizing the objective function:

$$J = \frac{C}{n} \sum_{i=1}^{n} |f(\mathbf{x}_i, \mathbf{w}) - y_i|_\epsilon + \frac{1}{2} \|\mathbf{w}\|^2, \tag{3}$$

where C (which controls the regularization) and ϵ are the hyper-parameters.

The global minimum solution to the linear regression problem (1) can be achieved without iterative nonlinear optimization, hence local minima in the objective function are not a problem. However, the linear regression problem can be very expensive or impossible to compute without truncating infinite dimensional vectors. This occurs because $\mathbf{\Phi}(\mathbf{x})$ may be a very high (or even infinite) dimensional vector. To counter this drawback a kernel trick is used in which the inner product $\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}') \rangle$ in the solution algorithm is replaced by a kernel function $K(\mathbf{x}, \mathbf{x}')$, which does not involve the difficult handling of $\mathbf{\Phi}(\mathbf{x})$. The minimization of (3) uses the method of Lagrange multipliers, and the final regression estimate can be expressed in the form (Bishop, 2006):

$$f(\mathbf{x}) = \sum_i K(\mathbf{x}, \mathbf{x}_i) + b, \tag{4}$$

where the summation is only over a subset of the given data $\mathbf{x}_i$ called the support vectors.

3.1. Evolutionary Strategies

Evolutionary algorithms (EA) mimic nature's way of evolving successful organisms (individuals) (Haupt et al., 2009). An Evolutionary Strategy (ES) is a particular class

of EA that is normally used for continuous parameter optimization. An attractive point of the ES is the self-adaptation of some strategy parameters. The strategy parameters are the set of adjustable parameters used by the algorithm. Self-adaptivity means that some EA parameters are varied during a run in a specific manner: the parameters are included in the chromosomes and co-evolve with the solutions (Eiben and Smith, 2003), i.e., the algorithm is capable of adapting itself autonomously.

3.1.1. Mutation and Self-adaptation

Mutation is the name given to a genetic operation which uses only one parent and creates one offspring by applying some type of randomized change to the representation. Let $\mathbf{G}$ be a chromosome defined by,

$$\mathbf{G} = (g_1, g_2, \ldots, g_p), \tag{5}$$

where g_i $(i = 1, 2, \ldots, p)$ are the solution parameters. Mutations are realized by adding some Δg_i to each g_i, where Δg_i are values from a Gaussian distribution $N(0, \sigma)$, with zero mean and standard deviation σ, i.e.,

$$g_i' = g_i + N(0, \sigma), \quad i = 1, 2, \ldots, p. \tag{6}$$

The self-adaptation consists of including the step size σ in the chromosomes so it also undergoes variation and selection, i.e. mutations are realized by replacing $(g_1, g_2, \ldots, g_p; \sigma)$ by $(g_1', g_2', \ldots, g_p'; \sigma')$, where σ' is the mutated value of σ, also called the mutation step size.

3.1.2. Uncorrelated Mutation with p Step Sizes

An attractive feature of the "mutation with p step sizes" method is its ability to treat each dimension differently, i.e., using different step sizes for different dimensions. The chromosome given in (5) is extended to p step sizes, resulting in

$$\mathbf{G} = (g_1, g_2, \ldots, g_p; \sigma_1, \sigma_2, \ldots, \sigma_p), \tag{7}$$

and the mutation rules are given by:

$$\sigma_i' = \sigma_i e^{\tau' N(0,1) + \tau N_i(0,1)}, \tag{8}$$

$$g_i' = g_i + \sigma_i N_i(0,1), \quad i = 1, 2, \ldots, p, \tag{9}$$

where $\tau' \propto 1/\sqrt{2p}$, and $\tau \propto 1/\sqrt{2\sqrt{p}}$. The common base mutation $e^{\tau' N(0,1)}$ allows an overall change of the mutability. The flexibility to use different mutation strategies in different directions is provided by the coordinate-specific $e^{\tau N_i(0,1)}$ (Eiben and Smith, 2003).

3.2. Hyper-parameter Optimization and SVR-ES

The performance of an SVR model is highly dependent on the choice of the kernel function and the hyper-parameters (Hastie et al., 2009; Hsieh, 2009). The use of a suitable nonlinear kernel function in SVR allows it to be fully nonlinear, while the use of a linear kernel function restricts SVR to a linear model. The SVR with the linear kernel is a linear regression model but with the robust ϵ-insensitive error norm instead of the non-robust MSE norm as used in multiple linear regression (MLR). In this study, we used the radial basis function (RBF) kernel (also called the Gaussian kernel) given by $K(\mathbf{x}, \mathbf{x}_i) = \exp[-\|\mathbf{x} - \mathbf{x}_i\|^2/(2\sigma_K^2)]$, with the hyper-parameter $\gamma = 1/(2\sigma_K^2)$ controlling the width σ_K of the Gaussian function.

Cherkassky and Ma (2004) proposed how to estimate values for the hyper-parameters C and ϵ in SVR. The estimated regularization parameter C is $\max(|\overline{y} + 3\sigma_y|, |\overline{y} - 3\sigma_y|)$ where $\overline{y}$ and σ_y are the mean and the standard deviation of the y values of the training data; and $\epsilon \sim 3\ \sigma_N\sqrt{\ln n/n}$, where n is the size of data set and σ_N is the standard deviation of the noise. The noise level is estimated by using the k-nearest-neighbors regression method, where $\hat{y}$, the value of a new point, is estimated from the average of the k nearest points (Hastie et al., 2009). Hence the estimated noise is:

$$\hat{\sigma}_N^2 = \frac{n^{1/5}k}{n(n^{1/5}k - 1)} \sum_{m=1}^{n} (y_m - \hat{y}_m)^2, \tag{10}$$

where y_m and $\hat{y}_m$ are the observed and estimated output values, respectively, for case m.

Cherkassky and Ma (2004) suggested that with d predictor variables the RBF width σ_K should behave as $\sigma_K^d \sim (0.1 - 0.5)$. Hence, no precise estimate of γ was given. From now on we will call the SVR with the Cherkassky-Ma hyper-parameters just SVR for brevity.

In SVR-ES, the ES is initialized from a starting solution $\mathbf{h}$ containing the Cherkassky-Ma hyper-parameters. The local search algorithm searches the candidate solutions in $N(\mathbf{h})$, a set of neighbors, for an $\mathbf{h}'$ that has a better fitness function than $\mathbf{h}$. Basically, the fitness function assigns a fitness value to each point in the parameter space (adaptive landscape (Eiben and Schoenauer, 2002)), where this value can be seen as a measure of how good a solution, represented by that point in the landscape, is to the given problem (Hordijk, 1996). In this way, if a solution exists (i.e. has better fitness function) it is accepted as the new incumbent solution, and the search proceeds by examining the candidate solution in $N(\mathbf{h}')$. In our particular case, we minimize the MSE ($f(\mathbf{h}) = (\text{MSE})^{-1}$). The necessary steps for implementing the SVR-ES algorithm are shown in Figure 1.

Eventually, this process will lead to the identification of a local optimum (Eiben and Smith, 2003). However, as this was done for a single individual instead of a population (which is necessary for a global search), this approach is highly dependent on its initial condition (initial individual). Assuming that the initial hyper-parameters from Cherkassky-Ma are reasonable, our local search with ES is adequate.

In particular, the use of SVR-ES is attractive because it can be implemented in an almost automatic way, uses variable steps to find the optimal hyper-parameters, is self-adaptive, and, by using $\tau' = 1/\sqrt{2p}$ and $\tau = 1/\sqrt{2\sqrt{p}}$, the only EA algorithm parameter that remains to be adjusted is the number of iterations.

4. Artificial Neural Network and Regression Trees

For comparison with the SVR and SVR-ES models, an artificial neural network (ANN) and two regression tree algorithms are also applied to the WCCI-2006 datasets.

An ANN is a biologically inspired mathematical model which is composed of a large number of highly interconnected perceptrons divided in layers (input, hidden, output), but working in union to solve a problem. Training of the ANN model involves adjusting the parameters iteratively so the MSE between the model output $\hat{y}$ and target y is minimized. Overfitting is a known problem with ANN. If p, the number of model parameters, is too large, the ANN may overfit the data, producing a spuriously good fit that does not lead to better predictions for new cases. This motivates the use of model comparison

criteria, such as the corrected Akaike information criterion (AIC_c), which penalizes the MSE as a function p. The AIC_c, a bias-corrected version of the original AIC, is given by

$$AIC_c = n \ln(\text{MSE}) + 2p + 2(p+1)(p+2)/(n-p-2), \tag{11}$$

where n is the number of effective observations (Faraway and Chatfield, 1998). Models with increasing numbers of hidden neurons n_h are trained and the model that minimizes AIC_c is chosen as the optimum model. The ANN architecture and training algorithm used in this study are based on Cannon and Lord (2000), with one hidden layer, early stopping to avoid overfitting, an ensemble (or committee) of ANN models to deal with local minima in the objective function, and with bootstrap aggregation (bagging) (Breiman, 1996).

Conceptually simple yet powerful, regression tree analysis, applicable to data sets with both a large number of patterns and a large number of variables, is extremely resistant to outliers. Among the decision trees used in machine learning, the classification and regression tree (CART) model, first introduced by Breiman et al. (1984), is the most commonly used. A random forest (RF) (Breiman, 2001) is a bootstrap ensemble of many decision trees (CART). Each tree is grown over a bootstrap sample from the training data set using a randomly selected subset of the available predictors at each decision branch.

Like CART, M5 builds a tree-based model, however the tree constructed by M5 can have multivariate linear models at its leaves - the model trees are thus analogous to piecewise linear functions. The advantage of M5 over CART is that M5 model trees are generally much smaller and more accurate in some problem domains (Quinlan, 1992).

5. Experimental Results

To demonstrate the practical use of the forecasting methods for environmental problems, experiments were performed on the datasets outlined in Section 2. In order to compare linear versus non-linear approaches, we also performed experiments with MLR. Linear models are simple, fast, and often provide adequate and interpretable descriptions of how the inputs affect the output. In particular for prediction purposes they can sometimes outperform fancier nonlinear models (Hastie et al., 2009). All the forecast models can be built using the free R software environment for statistical computing (R Development Core Team, 2011). For RF we used the R package randomForest (Liaw and Wiener,

2002). For SVR we used the R package e1071 (Dimitriadou et al., 2011). We developed the SVR-ES using the R package e1071 and the R native libraries; SVR-ES code is freely available from the project website: `http://forai.r-forge.r-project.org/`. For M5 we used the package RWeka (Hornik et al., 2009). ANN uses the monmlp package. For MLR and SLR we used the package stats.

Data were standardized (zero mean and unit variance) and divided into a training set (75% of the data) and an independent test set (last 25% of the data) which is used to test the trained model. For the SVR we used a 5-fold cross validation within the training set to train and validate the model. For SVR-ES we subdivided the training set into two parts leaving the final 20% to validate the model. The RF and ANN perform their own split-sample validation via the out-of-bootstrap samples in the bootstrap aggregation.

The ANN setup was as follows: the range of initial random weights was between $[-0.5 : 0.5]$, 30 ensemble members were used for bagging, and 5000 was used as the maximum number of iterations. For RF, we used 500 as the number of generated trees and 5, the default value, as the minimum size of terminal nodes. For M5 we used the default values of the package RWeka. For the SRV-ES, C and ϵ were initialized by the Cherkassky-Ma guidelines and γ by 0.001.

For SVR, the C and ϵ values used were from Cherkassky-Ma, but for γ we did a grid search using the range suggested by Cherkassky-Ma (see section 3.2) and the extended range $[2^{-10}, 2^{4}]$ suggested by Lin and Lin (2003). Table 1 shows that using γ from Cherkassky-Ma yielded poorer results than those from Lin and Lin. Henceforth, SVR will refer to the model using a γ search following Lin and Lin.

In order to compute the relative accuracy between the MLR and the non-linear methods we calculated the skill score (SS) (Hsieh, 2009) of the MAE and MSE. The SS is defined by:

$$\mathrm{SS} = \frac{A - A_{\mathrm{ref}}}{A_{\mathrm{perfect}} - A_{\mathrm{ref}}}, \tag{12}$$

where A is a particular measure of accuracy, A_{perfect} is the value of A for a set of perfect forecasts (in our case MAE and MSE equal to zero), and A_{ref} is the value of A computed over the set of reference forecasts (in our case the MAE or MSE value of the MLR model using all predictors). Positives SS means better performance than the reference model and negative values, worse performance.

According to the skill scores for PRECIP, the SVR (with the extended γ range) achieved the best MAE results (Figure 2) and the SVR-ES the best MSE results (Figure 3). That the two SVR methods perform better than the other methods relative to the MAE could be expected due to the ϵ-insensitive error norm of the SVR being more similar to the MAE. The SLR reduced the number of predictors from 106 to 59, cutting approximately 45% of the predictors. Basically, the SLR did not have an impact on the MLR accuracy. For the ANN, the MAE SS is also essentially the same with and without prescreening by SLR. However, an improvement is noted in the MSE when the SLR was applied. The combination of SLR with M5 and RF tended to diminish the skill. As M5 and RF do their own variable selection, it makes sense that SLR is not helpful. On other hand, SVR-ES had a minor improvement with fewer predictors. All the non-linear methods had better MAE results than the MLR (Figure 2). However the M5 had poor MSE performance when compared with the MLR (around 10% worse) (Figure 3). The SVR-ES had good results when compared with MLR, with skill scores exceeding 20% in MAE and over 10% in MSE.

Figures 4 and 5 show that for TEMP the SVR-ES achieved the best results in both MAE and MSE. Excluding M5, all the non-linear methods again had better MAE SS than the MLR. The SLR reduced the 106 predictors to 60, cutting approximately 44% of the predictors. Again the strongest improvement was ANN combined with SLR. SLR was also beneficial to SVR-ES, SVR, and RF but, as in PRECIP, detrimental to M5.

For the SO2 dataset, Figures 6 and 7 show RF as the best performer. However, SVR-ES kept its good performance in both MSE and MAE. Figure 6 shows again that MLR is the worst in terms of MAE performance. SVR-ES has MAE SS performance around 10% better than the MLR, and RF and SVR around 15% better. SLR reduced the number of predictors for SO2 from 27 to 21, cutting approximately 23% of the predictors. Although SLR did not have major impact on the forecast accuracy for the SO2 dataset, the results are still desirable since the reduction of predictors reduces computing time.

To clarify why SVR performs better than SVR-ES in Figures 2 and 6, we performed a small experiment on the PRECIP and SO2 datasets (with all predictors used) by varying the fitness function in ES. Instead of maximizing only $(\text{MSE})^{-1}$, we also tested using $(\text{MAE})^{-1}$ as fitness function, and calculated the skill scores using the MLR values

as reference. Figure 8 compares the skill score of MAE and MSE with $f_1 = (\text{MSE})^{-1}$ as fitness function, and $f_2 = (\text{MAE})^{-1}$ as fitness function.

For the PRECIP dataset (Figure 8a), with f_1 as fitness function, SVR-ES has worse MAE SS than SVR, but the two are comparable when f_2 is used instead. However, for the MSE SS, SVR-ES did better than SVR, regardless of whether f_1 or f_2 was used. Similarly, for the SO2 dataset (Figure 8b), SVR-ES with f_1 has worse MAE SS than SVR, but slightly better MAE SS than SVR with f_2. For the MSE SS, SVR-ES (with either f_1 or f_2) did better than SVR, thought the difference between f_1 and f_2 is more pronounced than in the PRECIP dataset. More guidelines on why an error measure can be minimized while a different error measure can remain unchanged or diminish can be found at Jachner et al. (2007) and Lima et al. (2010). In general, it is not possible for a single error measure to perform best on all criteria (e.g. cost, reliability, sensitivity, resistance to outliers, relationship to decision making, etc.) (Armstrong and Collopy, 1992), hence the choice of the fitness function is dependent on the desired goal.

The heart of the learning problem is generalization (Witten et al., 2011), i.e., whether the methods can retain satisfactory performance on new datasets. Based on the previous results (Figures 3, 4 and 5), M5 is not recommended. On the other hand, ANN, SVR, SVR-ES, and RF have better performance than the MLR when considering both MAE and MSE over the three tested datasets.

It is difficult to evaluate the computing time of two methods under every parameter setting because different values of the SVR hyper-parameters affect the training time (Fan et al., 2005). To illustrate the time reduction provided by SLR, a small experiment using the PRECIP dataset was performed. Varying the size of the training dataset, we measured the cpu time used to train the SVR model (with γ fixed at 0.001) and forecast the test set, using (i) all predictors and (ii) only the predictors selected by SLR. For 1000, 3000, 5000 and 7000 points in the training set, using all predictors required cpu times of 10, 135, 371 and 727 s, respectively, while using only the predictors selected by SLR required 5, 60, 212 and 469 s, respectively, thus confirming that screening of predictors by SLR saves computing time.

To provide some guidelines and explanation for SVR results, Figures 9 and 10 show changes related to the range of the γ value, with γ varying between $[2^{-10}, 2^4]$. The

MAE and MSE values were rescaled to range between $[0, 1]$ in order to compare the sensitivity between the different datasets. First, as discussed in Cherkassky-Ma, indeed there is dependency related to the γ values and the dataset dimensionality. However, this dependency is not present over the full tested range. For example, in Figures 9 and 10, the solid lines with triangles correspond to the full dimensionality (106 predictors) and the dashed lines with triangles correspond to a reduced dimensionality (60 predictors) of the TEMP dataset; within the range of $[2^{-10}, 2^{-6}]$ there is no difference between the dimensionality and the values of MAE/MSE. The same behavior occurs in the SO2 dataset over the range of $[2^{-10}, 2^{-4}]$ (solid and dashed lines with x symbols). Another interesting point is that the best γ values are independent of the dimensionality, i.e., solid lines and dashed lines converge to the same minimal point. This means that the dependency of γ is not totally related to the dataset dimensionality but with the dataset characteristics, as can be seen in the range of $[2^{-10}, 2^{-6}]$. However, in this range, the MAE and MSE results for the PRECIP dataset had different optimal γ values, though this difference is only one unit ($\log_2\gamma$) as the grid search used discrete steps of this step size.

Finally, to show the differences between various approaches to setting SVR hyper-parameters, we performed an experiment on the PRECIP dataset again, using all predictors and the same test set as before, but only the last 500 points of the training set. We tested the modified GA proposed by Leung et al. (2003) with different parameters settings (GA1, GA2 and GA3), SVR using the procedure recommended by Cherkassky and Ma (2004) but with the extended range $\gamma = [2^{-10}, 2^4]$ as suggested by Lin and Lin (2003), SVR-ES with 200 generations (which is less than the 500 used in GA1 and GA2) and SVR using 3-D grid search with the range of the three hyper-parameters recommended by Fan et al. (2005). For GA1, we used 500 as the number of generations, 0.1 as the mutation rate, 0.9 as the crossover weight and 50 as the size of the population (which was initialized randomly). For GA2, the corresponding values were 500, 0.2, 0.8 and 20, respectively, while for GA3, the values were 200, 0.3, 0.5 and 100, respectively. The reference model was again the MLR (trained on the 500 points).

SVR-ES and SVR had the best results according to the MSE and MAE SS (Figure 11). As shown by Kramer et al. (2007), changing the GA parameters can improve/worsen the

final results. Similar to Figure 2, SVR had better performance in the MAE SS than SVR-ES, but this can be reversed by using $(\text{MAE})^{-1}$ as the fitness function (Figure 8). The 3-D grid search had similar performance as GA1. In terms of computing efforts, SVR required 15 evaluations (for the 15 different γ values used in the search), SVR-ES 200 evaluations, 3-D grid search about 1500, and the three GA models 1400–3500 evaluations.

6. Summary and Conclusion

In summary, we used three environmental datasets to test five non-linear forecasting methods (four well-known and one that we proposed). Except for M5, the nonlinear methods (ANN, SVR, SVR-ES and RF) generally outperformed the linear method (MLR). Prescreening of predictors by SLR is generally beneficial for the nonlinear models (except for M5), as it reduces the computing time and may increase the forecast skills (especially for ANN). As explained in Witten et al. (2011), there is no universal best learning method. SVR-ES had very good accuracy when the datasets were TEMP and PRECIP, while RF had the best accuracy when the dataset was SO2. During pollution events, the SO_2 concentration spikes much more dramatically than the high values observed in the temperature and precipitation data, hence the architecture of RF may be particularly suited for predicting variables with a heavy-tailed distribution.

The best overall method tends to be SVR-ES. SVR (with the Cherkassky-Ma estimates for C and ϵ and an extended grid search in γ) worked well in terms of the MAE skill score, and provided satisfactory performance in terms of the MSE. It used a relatively modest amount of computing time for the hyper-parameter search. When using the fitness function of $(\text{MSE})^{-1}$ in the ES, SVR-ES may sometimes underperform SVR in terms of the MAE skill score, but changing the fitness function to $(\text{MAE})^{-1}$ appears to eliminate this problem.

As recommended by Eiben and Smith (2003), a common approach is to start with simple models; if the results are not of good enough quality, then move to more complex models. The SVR-ES accuracy can be improved using more complex ES such as correlated mutations (Eiben and Smith, 2003) or covariance matrix adaptation evolution strategy (CMA-ES) (Friedrichs and Igel, 2005). Multi-processors can also be used in the

computation, i.e. given a chromosome $\mathbf{G}$ (equation 7), k independent mutations can be made among ρ processors (Verhoeven and Aarts, 1995).

References

Armstrong, J.S., Collopy, F., 1992. Error measures for generalizing about forecasting methods: Empirical comparisons. International Journal of Forecasting 8, 69–80.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140. 10.1007/BF00058655.

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

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth Inc.

Cannon, A.J., Lord, E.R., 2000. Forecasting summertime surface-level ozone concentrations in the lower fraser valley of british columbia: An ensemble neural network approach. J. Air & Waste Manage 50, pp. 322–339.

Cawley, G.C., Janacek, G.J., Haylock, M.R., Dorling, S.R., 2007. Predictive uncertainty in environmental modelling. Neural Networks 20, 537 – 549. Computational Intelligence in Earth and Environmental Sciences.

Cherkassky, V., Ma, Y., 2004. Practical selection of svm parameters and noise estimation for svm regression. Neural Netw. 17, 113–126.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., , Weingessel, A., 2011. e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.5-26.

Draper, N.R., Smith, H., 1998. Applied Regression Analysis (Wiley Series in Probability and Statistics). Wiley series in probability and mathematical statistics. Applied probability and statistics, John Wiley & Sons Inc. 2 sub edition.

Eiben, A.E., Schoenauer, M., 2002. Evolutionary computing. Information Processing Letters 82, 1–6.

Eiben, A.E., Smith, J.E., 2003. Introduction to Evolutionary Computing. Natural Computing Series, Springer, Berlin.

Fan, R.E., Chen, P.H., Lin, C.J., 2005. Working set selection using second order information for training support vector machines. J. Mach. Learn. Res. 6, 1889–1918.

Faraway, J., Chatfield, C., 1998. Time series forecasting with neural networks: a comparative study using the air line data. Journal of the Royal Statistical Society: Series C (Applied Statistics) 47, 123–140. 10.1111/1467-9876.00109.

Friedrichs, F., Igel, C., 2005. Evolutionary tuning of multiple svm parameters. Neurocomputing 64, 107–117. Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004.

Fritsch, J.M., Houze, R.A., Adler, R., Bluestein, H., Bosart, L., Brown, J., Carr, F., Davis, C., Johnson, R.H., Junker, N., et al., 1998. Quantitative precipitation forecasting: Report of the eighth prospectus

development team, us weather research program. Bulletin of the American Meteorological Society 79, 285–299.

Hall, M., Smith, L., 1996. Practical feature subset selection for machine learning, in: McDonald, C. (Ed.), Computer Science '98 Proceedings of the 21st Australasian Computer Science Conference ACSC'98, Springer. pp. 181–191.

Hashmi, M.Z., Shamseldin, A.Y., Melville, B.W., 2009. Statistical downscaling of precipitation: state-of-the-art and application of bayesian multi-model approach for uncertainty assessment. Hydrology and Earth System Sciences Discussions 6, 6535–6579.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics, Springer. 2nd ed. 2009. corr. 3rd printing 5th printing. edition.

Haupt, S., Pasini, A., Marzban, C., 2009. Artificial intelligence methods in the environmental sciences. Springer.

Haykin, S., 1998. Neural Networks - A Comprehensive Foundation. Pearson Education. second edition.

Hocking, R.R., 1976. A biometrics invited paper. the analysis and selection of variables in linear regression. Biometrics 32, pp. 1–49.

Hordijk, W., 1996. A measure of landscapes. Evolutionary Computation 4, 335–360.

Hornik, K., Buchta, C., Zeileis, A., 2009. Open-source machine learning: R meets Weka. Computational Statistics 24, 225–232.

Hsieh, W.W., 2009. Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels. Cambridge University Press, New York, NY, USA.

Huang, C.L., Wang, C.J., 2006. A ga-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31, 231–240.

Jachner, S., van den Boogaart, K.G., Petzoldt, T., 2007. Statistical methods for the qualitative assessment of dynamic models with time delay (r package qualv). Journal of Statistical Software 22, 1–30.

Kramer, O., Gloger, B., Goebels, A., 2007. An experimental analysis of evolution strategies and particle swarm optimizers using design of experiments, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA. pp. 674–681.

Krasnopolsky, V.M., 2007. Neural network emulations for complex multidimensional geophysical mappings: Applications of neural network techniques to atmospheric and oceanic satellite retrievals and numerical modeling. Reviews of Geophysics 45.

Kuligowski, Robert J., A.P.B., 1998. Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. Wea. Forecasting 13, 1194?1204.

Kurt, A., Gulbagci, B., Karaca, F., Alagha, O., 2008. An online air pollution forecasting system using neural networks. Environment International 34, 592–598.

Leung, F., Lam, H., Ling, S., Tam, P., 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. Neural Networks, IEEE Transactions on 12, 79–88.

Liaw, A., Wiener, M., 2002. Classification and regression by randomforest. R News 2, 18–22.

Lima, A.R., Silva, D.A., Mattos Neto, P.S., Ferreira, T.A., 2010. An experimental study of fitness function and time series forecasting using artificial neural networks, in: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, ACM, New York, NY, USA. pp. 2015–2018.

Lin, K.M., Lin, C.J., 2003. A study on reduced support vector machines. Neural Networks, IEEE Transactions on 14, 1449 – 1459.

Lin, S.W., Ying, K.C., Chen, S.C., Lee, Z.J., 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications 35, 1817–1824.

Muller, K., Smola, A., Rtsch, G., Schlkopf, B., Kohlmorgen, J., Vapnik, V., 1997. Predicting time series with support vector machines, in: Gerstner, W., Germond, A., Hasler, M., Nicoud, J.D. (Eds.), Artificial Neural Networks ICANN'97. Springer Berlin / Heidelberg. volume 1327 of *Lecture Notes in Computer Science*, pp. 999–1004. 10.1007/BFb0020283.

Ortiz-García, E.G., Salcedo-Sanz, S., Pérez-Bellido, A.M., Portilla-Figueras, J.A., 2009. Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions. Neurocomputing 72, 3683–3691.

Pai, P.F., Hong, W.C., 2005. Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. Electric Power Systems Research 74, 417–425.

Pino-Mejías, R., de-la Vega, M.D.C., Anaya-Romero, M., Pascual-Acosta, A., Jordán-López, A., Bellinfante-Crocci, N., 2010. Predicting the potential habitat of oaks with data mining models and the r system. Environmental Modelling & Software 25, 826 – 836.

Quinlan, J.R., 1992. Learning with continuous classes, in: Adams, A., Sterling, L. (Eds.), Proceedings of the 5th Australian joint Conference on Artificial Intelligence, World Scientific. pp. 343–348.

R Development Core Team, 2011. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Schoof, J., Pryor, S., 2001. Downscaling temperature and precipitation: a comparison of regression-based methods and artificial neural networks. International Journal of Climatology 21, 773–790.

Smit, S., Eiben, A., 2009. Comparing parameter tuning methods for evolutionary algorithms, in: Evolutionary Computation, 2009. CEC '09. IEEE Congress on, pp. 399 –406.

Solomatine, D.P., Xue, Y., 2004. M5 model trees and neural networks: Application to flood forecasting in the upper reach of the huai river in china 9, 491–501.

Strangeways, I., 2007. Precipitation: theory, measurement and distribution. Cambridge University Press.

Tripathi, S., Srinivas, V., Nanjundiah, R.S., 2006. Downscaling of precipitation for climate change scenarios: A support vector machine approach. Journal of Hydrology 330, 621–640.

Venables, W., Ripley, B., 2002. Modern applied statistics with S. Statistics and computing, Springer.

Verhoeven, M., Aarts, E., 1995. Parallel local search. Journal of Heuristics 1, 43–65. 10.1007/BF02430365.

Von Storch, H., Zwiers, F., 2001. Statistical Analysis in Climate Research. Cambridge University Press.

Wallace, J., Hobbs, P., 2006. Atmospheric Science: An Introductory Survey. International Geophysics

519 Series, Elsevier Academic Press.
520 Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Tech-
521 niques. Morgan Kaufmann, Burlington, MA. 3 edition.
522 Zeng, Z., Hsieh, W.W., Burrows, W.R., Giles, A., Shabbar, A., 2011. Surface wind speed prediction
523 in the canadian arctic using non-linear machine learning methods. Atmosphere-Ocean 49, 22–31.
524 http://www.tandfonline.com/doi/pdf/10.1080/07055900.2010.549102.
525 Zhang, G.P., 2007. Avoiding pitfalls in neural network research. IEEE Transactions on Systems, Man,
526 and Cybernetics, Part C 37, 3–16.

527 List of Tables

Table 1: Comparison of MAE and MSE values from the test set using SVR with the range of γ suggested by Cherkassky and Ma (2004) and by Lin and Lin (2003) for the PRECIP, TEMP, and SO2 datasets, using either all predictors (ALL) or predictors selected by SLR.

	MAE				MSE			
	Cherkassky-Ma		Lin and Lin		Cherkassky-Ma		Lin and Lin	
	ALL	SLR	ALL	SLR	ALL	SLR	ALL	SLR
PRECIP	0.0534	0.0530	0.0379	0.0380	0.00951	0.00944	0.00572	0.00572
TEMP	0.2520	0.2415	0.0622	0.0618	0.09199	0.08543	0.00642	0.00630
SO2	0.0277	0.0263	0.0244	0.0244	0.00386	0.00371	0.00325	0.00326

List of Figures

Figure 1: Procedure of the SVR-ES

begin

  $\tau \rightarrow 0$; // τ: Iteration number;

  if *Screenout irrelevant Predictors* = *TRUE* **then**

    Perform stepwise regression;

  end

  Initialize $\mathbf{G}(\tau)$; // $\mathbf{G}(\tau)$: Chromosome according to Equation (7);

  Evaluate $f(\mathbf{G}(\tau))$; // $f(\mathbf{G}(\tau))$: Fitness function value ;

  while *not termination condition* **do**

    $\tau \rightarrow \tau + 1$;

    Perform *mutation* operation according to Eqns. (8) and (9) to generate new chromosome $\mathbf{G}'$;

    Evaluate $f(\mathbf{G}'(\tau))$;

    if $f(\mathbf{G}') > f(\mathbf{G})$ **then**

      $f(\mathbf{G}) \leftarrow f(\mathbf{G}')$;

    end

  end

end

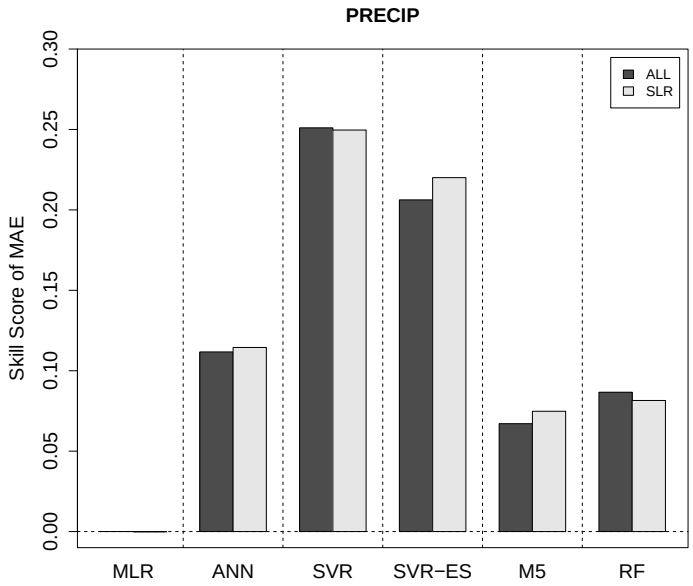

Figure 2: Skill score of MAE for the PRECIP test dataset with all predictors used (dark bar) and with predictors selected by SLR (light bar). The reference forecast, MLR with all predictors, has simply 0 for the skill score.

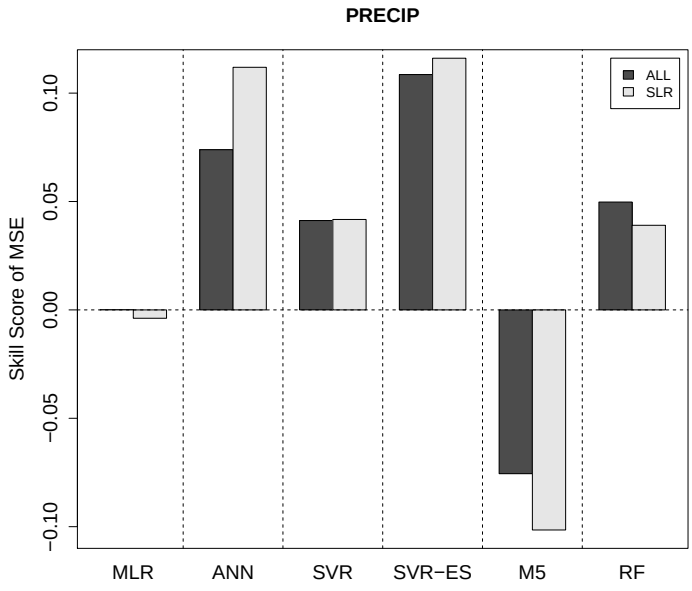

Figure 3: Skill score of MSE for the PRECIP test dataset.

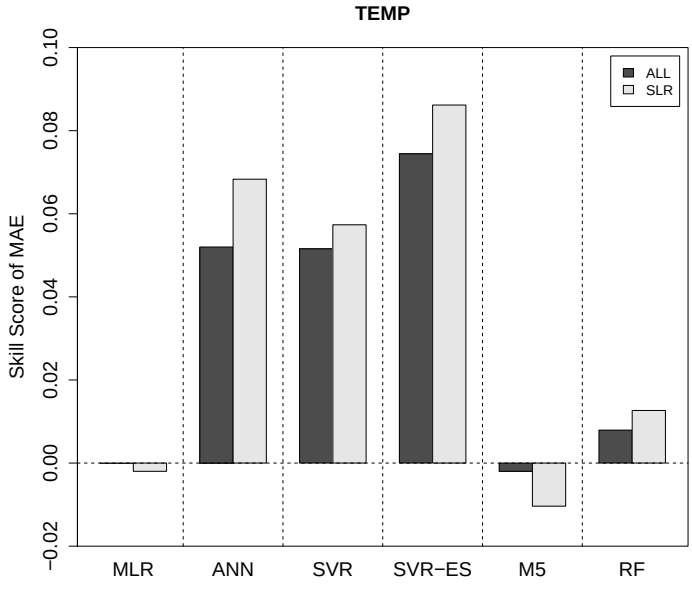

Figure 4: Skill score of MAE for the TEMP test dataset.

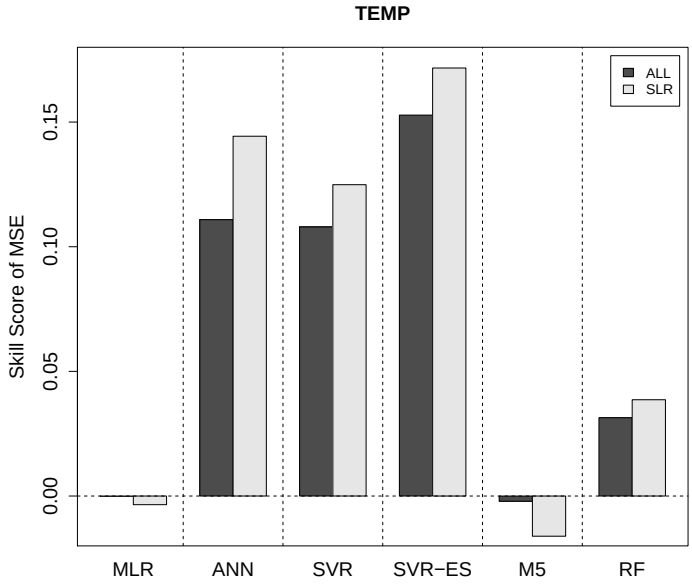

Figure 5: Skill score of MSE for the TEMP test dataset.

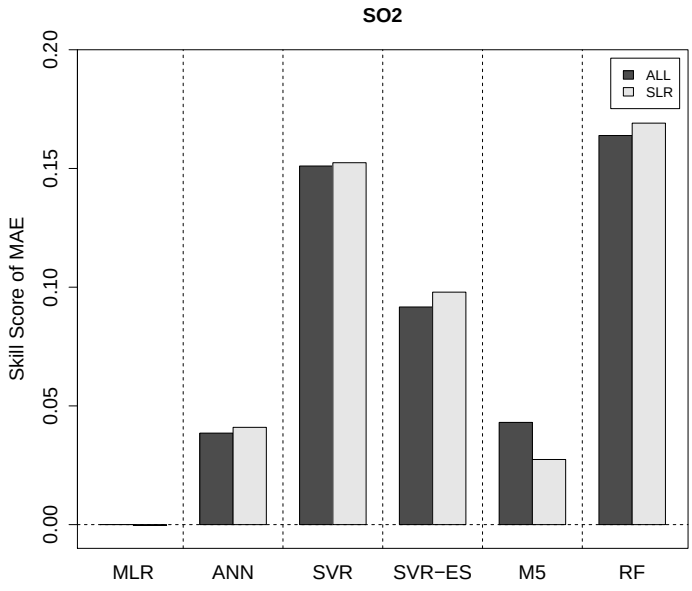

Figure 6: Skill score of MAE for the SO2 test dataset.

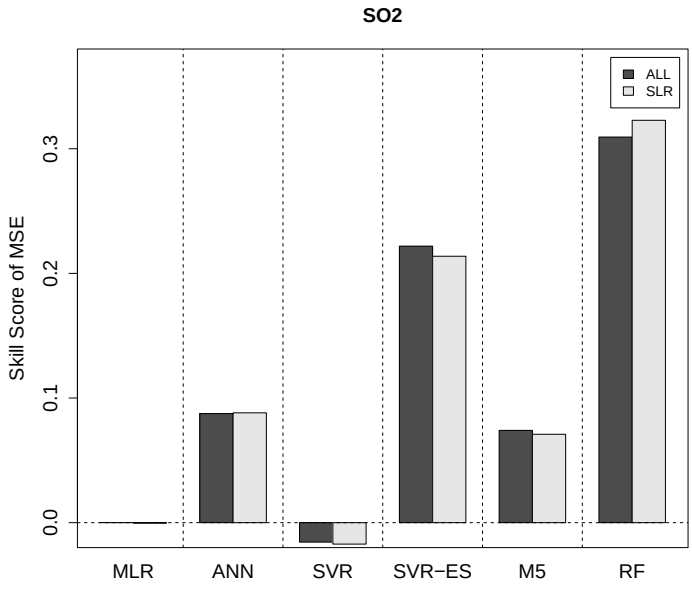

Figure 7: Skill score of MSE for the SO2 test dataset.

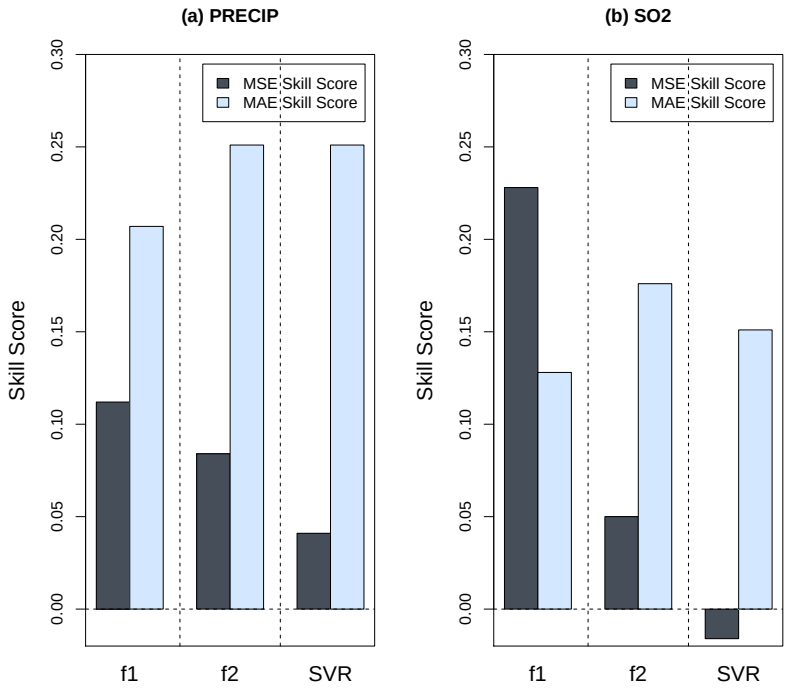

Figure 8: Skill score of MSE (dark bar) and MAE (light bar) for (a) PRECIP and (b) SO2 test datasets, where f1 denotes SVR-ES with f_1=(MSE)$^{-1}$ as fitness function and f2 denotes SVR-ES with f_2=(MAE)$^{-1}$.

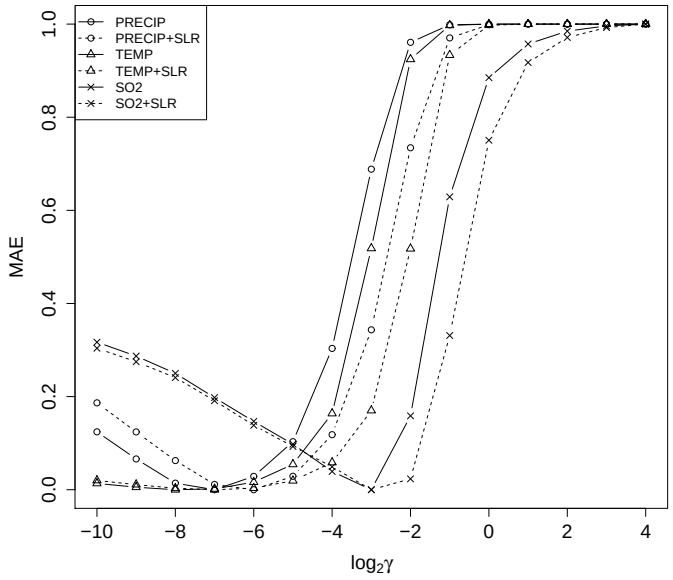

Figure 9: Results of MAE as γ varies between $[2^{-10}, 2^4]$, for the PRECIP test dataset with all predictors used (solid line with circles), with predictors selected by SLR (dashed line with circles), for the TEMP test dataset with all predictors used (solid line with triangles) and with predictors selected by SLR (dashed line with triangles), for the SO2 test dataset with all predictors used (solid line with x) and with predictors selected by SLR (dashed line with x).

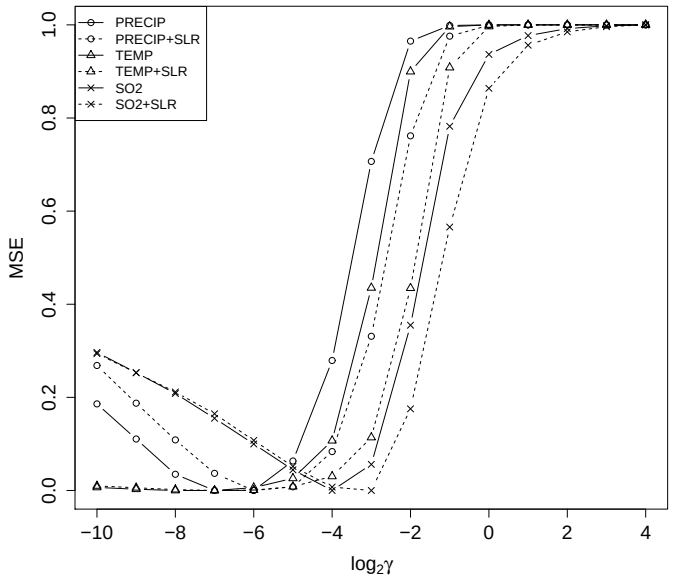

Figure 10: Results of MSE as γ varies between $[2^{-10}, 2^4]$.

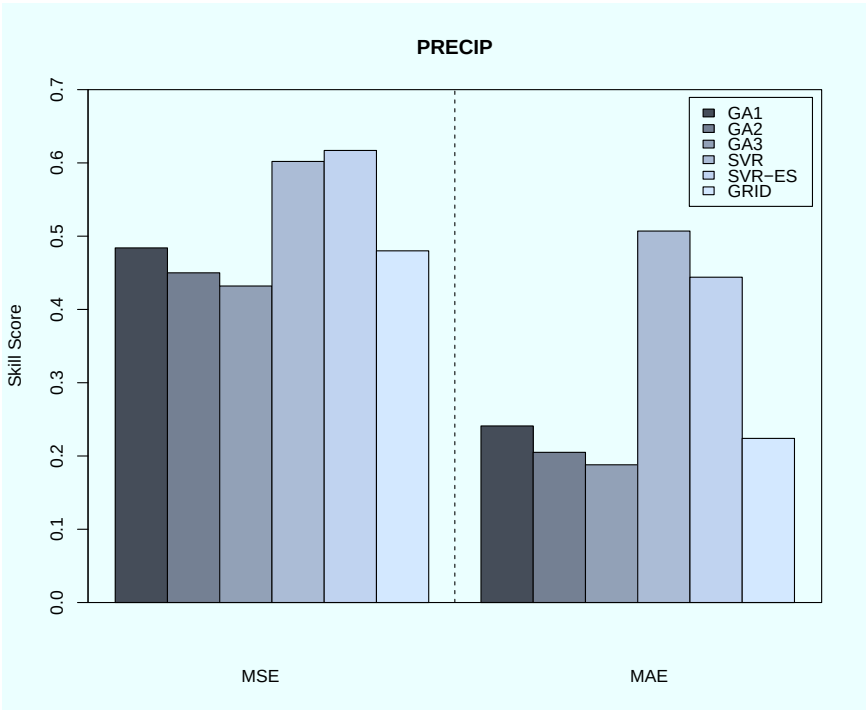

Figure 11: Results of MSE skill score (left side) and MAE skill score (right side) for the PRECIP test dataset using the last 500 points of the training set and all the predictors to train the models. The models are respectively: the modified GA proposed by Leung et al. (2003) with different parameters settings (GA1, GA2 and GA3), SVR with the procedure recommended by Cherkassky and Ma (2004) and the extended range $\gamma = [2^{-10}, 2^4]$ suggested by Lin and Lin (2003), SVR-ES with 200 generations and 3-D grid search with the hyper-parameters' range recommended by Fan et al. (2005).