Training Subset Selection in Hourly Ontario Energy Price Forecasting using Time Series Clustering-based Stratification

Karol Lina López[a], Christian Gagné[a], Germán Castellanos-Dominguez[b], Mauricio Orozco-Alzate[b]

[a] *Computer Vision and Systems Laboratory (CVSL), Département de génie électrique et génie informatique, Université Laval, Québec (Québec), Canada G1V 0A6*

[b] *Universidad Nacional de Colombia - Sede Manizales - Departamento de Ingeniería Eléctrica, Electrónica y Computación - Grupo de Control y Procesamiento Digital de Señales - Kilómetro 7 vía al Magdalena, Manizales, 170003 - Colombia*

Abstract

Training a given learning-based forecasting method to a satisfactory level of performance often requires a large dataset. Indeed, any data-driven methods require having examples that are providing a satisfactory representation of what we want to model to work properly. This often implies using large datasets to be sure that the phenomenon of interest is properly sampled. However, learning from time series composed of too many samples can also be a problem, given that the computational requirements of the learning algorithms can easily grow following a polynomial complexity according to the training set size. In order to identify representative examples of a dataset, we are proposing a methodology using clustering-based stratification of time series to select a training data subset. The principle for constructing a representative sample set using this method consists in selecting heterogeneous instances picked from all the various clusters composing the dataset. Results obtained show that with a small number of training examples, obtained through the proposed clustering-based stratification, we can preserve the performance and improve the stability of models such as artificial neural networks and support vector regression, while training at a much lower computational cost. We illustrate the methodology through forecasting the one-step ahead Hourly Ontario Energy Price (HOEP).

Keywords: Stratification, Data Selection, Stratified sampling, Forecasting models, Hourly Ontario Energy Price

1. Introduction

As time series are generally produced through regular sampling of a given phenomenon over a period of time, it is common to obtain very large set of redundant data using a relatively high sampling frequency (e.g., a sample every minute) over a long period of time (e.g., several years). A large training set increases the memory and processing required to generate the forecasting function. This problem can be particularly acute in situations requiring the repeated generations of a forecasting function from the data set (e.g., adjusting the hyper-parameters to learn a given forecasting function). There is thus considerable interest in reducing the training set size to remove redundancy in a training set, which can improve the space and time efficiency of the forecast models.

To reduce the training set, we propose a stratified sampling of an input space time series by clustering of the data based on a state representation of each instance. Particularly, we investigate the problem of selecting a subset of available candidate examples so as to obtain a representative description of a large dataset, in order to conduct supervised learning. We aim at removing redundancy in the training set by assuming that with a representative subset of examples, we can obtain a generalization error close to the one obtained with the full data set. For that purpose, we assume that we already have a sufficient amount of representative data instances. We make a deterministic selection of representative examples from the different clusters to be used further in forecasting model training. This clustering-based stratification is concretely carried out for Hourly Ontario Energy Price (HOEP) forecasting. Lagged values of the HOEP as well as the lagged values of the Hourly Ontario Demand (HOD) are considered as explanatory variables.

The paper is organized as follows: Sec. 2 is an overview of relevant work concerning the proposed approach. In Sec. 3, we describe the proposed methodology of clustering-based stratification so as to select the training subsets. In Sec. 4, the Hourly Ontario Energy Price data set is presented, as well as the experimental set-up to evaluate performance of artificial neural networks and support vector regression trained on the data subsets generated by clustering-based stratification. Results and discussions are presented in Sec. 5, followed by some conclusions in Sec. 6.

2. Related work

The major source of inspiration of our own work originates from [1], where two methods for constructing the cross-validation folds from a dataset are presented to deterministically assess classifiers. The folds are constructed

using unsupervised stratification by exploiting the instance distribution in the input space. The first proposed approach ranks the samples according to their distance to the data set centroid, and then this distance it used for partitioning. The second approach clusters and sorts the data (using the well-known K-means algorithm [2]) according to their cluster centre in order to conduct the partitioning. Since both methods attempt to construct more representative allocation of observations into folds, they reduce the bias of the resulting estimator.

Nonetheless, the scope of the current work is to extract the representative data subset for regression-type analysis in a context of time series forecasting, for which the explanatory variable depends on its own history. The starting point is to embed the data into a time-delayed space of suitable dimension [3]. Specifically, time series data are represented by a data point sequence typically measured at successive moments sampled over uniform time intervals. In that case, it is common to collect a large number of redundant observations. Consequently, it is important to choose a small but representative subset of training examples in order to reduce the computational burden while preserving performances and possibly improving stability.

Active learning [4] is also closely related to the current work, although dataset selection is made on-the-fly during the training. The idea of active learning is to query, and eventually label, the data samples dynamically during the learning phase. The selection of the next training samples is carried out according to some criterion, for example the level of uncertainty the learner has on the data available in the pool. Active learning is generally considered useful when all data are not labelled and the labelling operation has a given cost, as the method is able to limit greatly the number of samples requiring labelling.

Another method based on an Artificial Neural Network (ANN) for selecting examples was proposed by [5]. In particular, patterns are grouped into pairs located on both sides of a classification boundary by considering the Hamming distance. To improve the ANN generalization ability, training is accomplished as suggested in [6]. Namely, the network training is initiated with a small subset. During the training process, generalization of the network is estimated using an independent test set and a new pattern is selected when the generalization estimate exceeds the apparent network error on the current training set. The new training example is selected to have the maximal error. A similar algorithm was developed by [7], called *active selection of training sets*. New patterns having the maximal error are added to the current subset using an integrated mean square error estimate. The main focus lies on reduction of the training set size exploiting information obtained from the model due to learning from previous examples. Likewise, [8] proposes *cross validation with active pattern selection* based on leave-one-out cross-validation of ANN.

On the other hand, [9] introduces two new data selection methods to train Support Vector Machines (SVMs) for classification: the first one selects training data based on an introduced statistical confidence measure, whereas the second one uses the Hausdorff distance measure as a criterion to decide which training examples should belong to the reduced training set. In turn, [10] and [11] propose a procedure based on clustering by K-means to accelerate the training of SVMs. Clusters with mixed composition are likely to occur near the separation margins and they may hold some support vectors. Consequently, the number of vectors in a SVM training set is smaller and the training time can be decreased without compromising the generalization capability.

Some dimensionality reduction methods can be used to select a subset of data if the time series is considered as a point in a *N-dimensional* space. The problem of dimensionality reduction in a time series has been addressed mainly by transform methods. Particularly, [12] introduced the *Adaptive Piecewise Constant Approximation* (APCA) that approximates each time series by a set of constant value segments of varying length such that their individual reconstruction errors are minimal. [13] propose an index compression method named *Grid-based Datawise Dimensionality Reduction* which attempts to preserve the characteristics of the time-series.

The method we are proposing differs from previous ones in that the selection of samples is performed directly in the input space. Moreover, we take into account the history of the time series since inputs are composed of lags. Note that, we do not consider the example selection and training of the forecast model to be conducted simultaneously, as this can be computationally expensive and depends on to the training algorithm used. Furthermore, in the procedure proposed herein, no prior knowledge of the desired outputs is required, as the method is unsupervised. Thus, it can be applied to either regression or classification problems, including those cases when labels are not available beforehand.

3. Selection of representative training examples

The selection of representative training data examples is carried out in two steps: 1) applying the clustering procedure to the time series, in order to discover pattern behaviours on input space, and 2) selecting the data from the clusters obtained, building stratified sample sets that form a parsimonious data representation.

The main goal is to select data best representing the structure of the inputs. For that purpose, we apply clustering methods on the input space, in order to determine the different groups of similar instances. From that point on, we divert the data of a given cluster evenly into the different folds (data subsets). Clustering is achieved through the classical K-means algorithm along with the Euclidean distance between each instance and the associated cluster centre.

For time series forecasting, we predict the value of a given variable at the current time step using as input some of its past values predefined during time steps, termed

lagged values. The lagged values of the HOEP as well as the lagged values of the HOD are considered as explanatory variables [14]. More formally, let the list $\boldsymbol{a} \in \mathbb{R}^{1\times n}$ be the n lags of the HOEP and $\boldsymbol{b} \in \mathbb{R}^{1\times m}$ be the m lags of the HOD at current time t used to build up the list $\boldsymbol{l}(t) \in \mathbb{R}^{1\times n+m}$, which represents the lagged values (*Input Space*) of the forecasted variable:

$$\boldsymbol{a} = [a_1, \ldots, a_n], \tag{1a}$$
$$\boldsymbol{b} = [b_1, \ldots, b_m], \tag{1b}$$
$$\boldsymbol{v}_1(t) = [HOEP(t-a_1), \ldots, HOEP(t-a_n)], \tag{1c}$$
$$\boldsymbol{v}_2(t) = [HOD(t-b_1), \ldots, HOD(t-b_m)], \tag{1d}$$

where $\boldsymbol{v}_1(t)$ are the HOEP lagged values and $\boldsymbol{v}_2(t)$ are the HOD lagged values.

Concatenating both vectors provides:

$$\boldsymbol{l}(t) = [\boldsymbol{v}_1(t) \| \boldsymbol{v}_2(t)], \tag{2}$$

where the notation $\|$ represents the concatenation operator.

3.1. Mapping the input space into states

When measuring the Euclidean distance between two input lagged sets, both observations may have the same shape but differ only on the lag at the current time step. Such vectors can be very different in terms of distance, whereas actually they only differ in their alignment. For this reason, using distance-based clustering may lead to poor results when alignment issues occur in the data.

In order to overcome alignment issues, we are proposing to transform the input space into a new space called the input state, which allows invariance to data shifts. It consists on considering the minimum, mean, and maximum of the different lags of the input values as the state $\boldsymbol{s}_i(t) \in \mathbb{R}^{1\times 3}$ of given lagged values:

$$\boldsymbol{s}_i(t) = \{\min(\boldsymbol{v}_i(t)), \mathrm{mean}(\boldsymbol{v}_i(t)), \max(\boldsymbol{v}_i(t))\}. \tag{3}$$

Though other measures can be used as the state from the lagged values, these three features appear sufficient in the present work. The new state $\boldsymbol{s}'(t) \in \mathbb{R}^{1\times 6}$ is the representation of six values at time t of the time series:

$$\boldsymbol{s}'(t) = [\boldsymbol{s}_1(t) \| \boldsymbol{s}_2(t)]. \tag{4}$$

3.2. Clustering and distributing data into folds

A survey of clustering time series problems for various applications can be found in [15]. For instance, for forecasting price curves, [16] conducts an experiment with two clustering techniques, K-means and Expectation Maximization, demonstrating that the application of these techniques is effective for splitting the whole year into different day groups, according to their price variations.

In the current paper, we chose to use the K-means method due to its simplicity, speed, and general robustness, allowing *good enough* solutions to be found for most cases. This method relies on an iterative scheme starting with arbitrarily (randomly) chosen centres. Then, it alternates between two steps until convergence or the exhaustion of resources [15]: 1) distributing of objects among the clusters, and 2) updating of the cluster centres. The main idea behind the method is to minimize an objective function, usually the sum of the squared distance between the instances from their respective cluster centres.

Thus, the K-means algorithm is applied to cluster the time series at hand allowing a partition of the data into a number of groups (the clusters). We then want to assign the data in the different clusters to the folds, so that each fold contains the representative subset of the original dataset. In order to distribute the data into folds, the clusters are first sorted based on the distance between each cluster centre and the centre of the first cluster at hand. Then, the data are sorted forming each cluster according to its distance from the cluster centre [1]. This second sorting step is required to ensure that the data within a cluster are distributed evenly between the folds, since instances within a cluster are likely to be more related to nearby data according to the distance to the cluster centre than to data farther away using the same measure. Moreover, instances close to the cluster centre probably resemble each other more that they resemble data of the clusters located far from the centre.

With this sorting in two stages, between the clusters and within the clusters, data instances can then be distributed to the folds. This is done simply in a round-robin fashion, processing the data sequentially according to the two stages of sorting, and alternating between the folds when making the assignments. For instance, we begin by processing the first cluster, from which the closest data to the cluster centre is assigned to the first fold, the second closest data to the centre is assigned to the second fold, and so on. When all folds have received one instance, we then return to the first fold to proceed with the assignments, and when all data of a cluster have been processed, we continue with the next cluster according to its distance from the first one.

To ensure a good stability of the stratification process [17], it is convenient to carry out several repetitions of K-means to produce the clusters, each with a new set of initial cluster centroid positions. The final solution is the one with the lowest value for within-cluster sums of point-to-centroid distances. This way, the stratification by clustering methodology becomes almost a deterministic approach, with a clustering method that produces the same clusters from one run to another, thus generating the same stratified partitioning of a given dataset for a given parametrization.

The procedure consists of the following steps:

1. Set K as the predefined number of clusters.
2. Apply K-means clustering algorithm with the Euclidean distance into the input space $\boldsymbol{l}(t)$ or input state $\boldsymbol{s}'(t)$. Repeat this procedure a number of times

(e.g., 30 experiments), each with a new set of initial centroids. Return and record the matrix solution $\boldsymbol{C}_l = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K] \in \mathbb{R}^{K \times n+m}$ for the input space or: $\boldsymbol{C}_s = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K] \in \mathbb{R}^{K \times 6}$ for the input state with the lowest value for within-cluster sums of point-to-centroid distances.

3. Order the clusters to obtain the ordinal set $\{g_i : i = 1, 2, \ldots, K\}$. The clusters are sorted according to the Euclidean distance between each of the cluster centroids and the centroid of the first cluster, whose centroid is $\boldsymbol{c}_1$. To select the first cluster, we calculate the number of instances in each cluster $n(g_i)$ and then we choose the one with the smallest number of elements. All clusters are ranked according to the similarity distance $\|\boldsymbol{c}_1 - \boldsymbol{c}_i\| \leq \|\boldsymbol{c}_1 - \boldsymbol{c}_j\|, \forall i \neq j$, with $i, j = 1, 2, \ldots, K$, where $\boldsymbol{c}_i$ is the i-th cluster centroid.
4. Sort instances of i-th cluster according to the similarity of their distance to the corresponding centroid, $\boldsymbol{c}_i$: $\|\boldsymbol{c}_i - \boldsymbol{x}_i(y)\| \leq \|\boldsymbol{c}_i - \boldsymbol{x}_i(z)\|, \forall y \neq z$, with $y, z = 1, 2, \ldots, n(g_i)$, where $\boldsymbol{x}_i(\cdot)$ denotes the ordered instance that belongs to i-th cluster for $\boldsymbol{l}(\cdot) \in \mathbb{R}^{1 \times n+m}$ to input space and $\boldsymbol{s}'(\cdot) \in \mathbb{R}^{1 \times 6}$ to input state.
5. Distribute the instances into F folds, where each $\{f = 1, 2, \ldots, F\}$ fold will contain instances from each cluster using interleaved indices, as proposed by [1], following Algorithm 1, using the clusters sorted in step 3 as groups g_i and instances $\boldsymbol{x}_i(\cdot)$ sorted in step 4. The number of instances in each fold is defined as N_f.
6. Create the training subset $\boldsymbol{X}^{tr}$ with the data of one or several folds.
7. (Optional) Further split the training subset $\boldsymbol{X}^{tr}$ into several subsets, according to the needs of the learning methodology used. This is achieved by applying Algorithm 1 to the whole training subset $\boldsymbol{X}^{tr}$ as a single input group g_1, to produce a certain number of new partitions according to the number and size of the new subsets required. For example, if we want to split the training subset into two new subsets, i.e. a learning subset with 67 % of the training subset data and a validation subset with the remaining 33 %, we first create three partitions from the training subset with Algorithm 1, then we concatenate the first two partitions to create the learning subset and we use the last one as the validation subset. Such an interleaved distribution of the training subset data will preserve the stratification in the various subsets.

3.3. Number of clusters

The K-means algorithm requires the user to specify the number of clusters to model in advance, referred to as the variable K. We are proposing here a method for the estimation of the number of clusters in the context of time series data stratification. Thus, we define a centroid $\boldsymbol{\gamma}_f$ for each fold, that is the mean value along each dimension of $\boldsymbol{X}^f = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_f}] \in \mathbb{R}^{N_f \times (n+m)}$, that is $\boldsymbol{\Gamma}_l = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_F] \in \mathbb{R}^{F \times n+m}$ for the input space or $\boldsymbol{\Gamma}_s = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_F] \in \mathbb{R}^{F \times 6}$ for the input state.

Algorithm 1 Interleaved data distribution in F subsets.

Require: groups g_i and instances $\boldsymbol{x}_i(\cdot)$
  $h = 1$
  for each group g_i **do**
    for each instance $\boldsymbol{x}_i(\cdot)$ **do**
      $m = h \mod f$
      if $m = 0$ then $m = f$
      assign instance $\boldsymbol{x}_i(\cdot)$ to fold m
      $h = h + 1$
    end for
  end for

We enforce the fact that each fold includes elements from all clusters, and that the number of elements from each cluster should be roughly comparable in each fold. Therefore, we expect that the centroids of the folds should be very close to each other when the clustering is well done. This is the criteria we are proposing to use to select the appropriate number of clusters, which corresponds to minimizing the average Euclidean distance between centroids for each pair of folds:

$$K = \underset{k}{\operatorname{argmin}} \ \frac{F(F-1)}{2} \sum_{i=1}^{F} \sum_{j=i+1}^{F} \|\boldsymbol{\gamma}_i^k - \boldsymbol{\gamma}_j^k\|. \qquad (5)$$

The proposed procedure for choosing the number of clusters involves the following steps:

1. Set the number of folds F;
2. Apply the procedure of clustering and data distribution into folds (see 3.2), using different values of K;
3. Compute the Euclidean distance between centroids of each pairs of folds, for each value of K;
4. Return the value of K that minimizes the average distance between fold centroids (Eq. 5).

An illustrative two-dimensional problem is presented in Fig. 1. It presents the idea behind clustering-based stratification with $K = 3$ clusters and $F = 3$ folds. The application of clustering allows the formation of groups of instances (the clusters), where the instances in each group are relatively similar. Then, these instances are distributed between the folds in order to obtain balanced sets that are capturing the essence of the various clusters with only a subset of the data. In Fig. 1, the plot at the left shows which instances of fold 1 (data in blue) are chosen. Note that this fold contains data from different regions of each cluster according to the distance between each data and the centroid of the corresponding cluster. The middle plot in the figure shows the instances selected from fold 1 (blue) and fold 2 (yellow). The plot at the right shows how the full data were divided into the three folds.

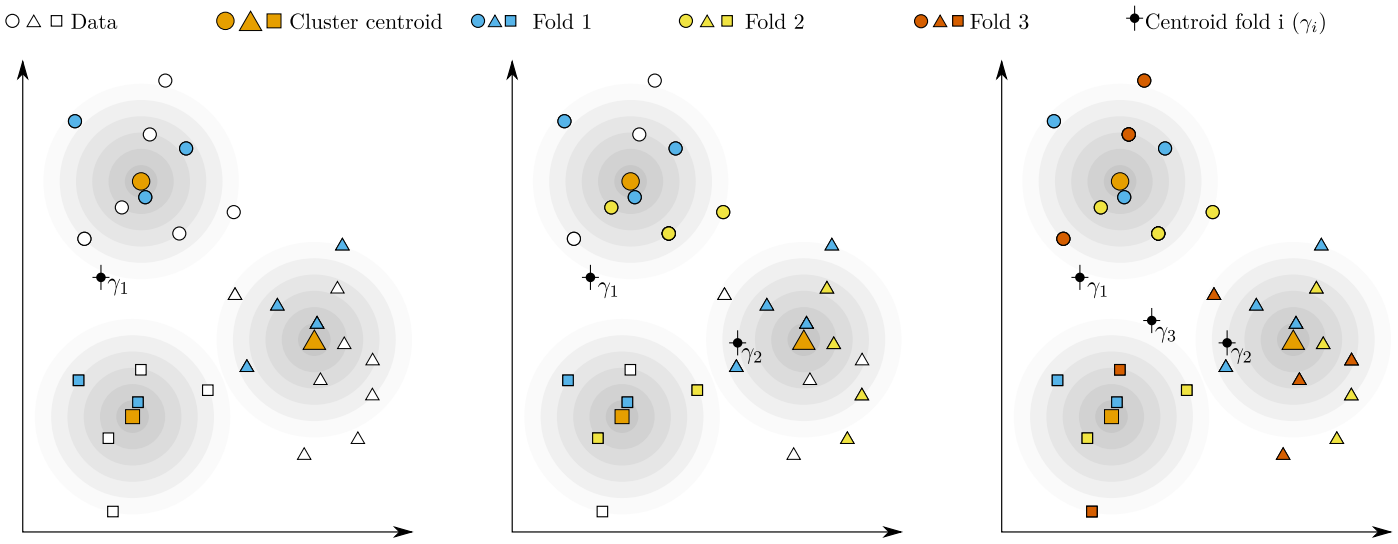


Figure 1: Illustrative example of the proposed method and how the number of clusters is selected, with $F = 3$ folds and $K = 3$ clusters in two dimensions. The left plot shows assignment of data to fold 1 (blue), the middle plot adds the assignment to fold 2 (yellow), while the right plot presents complete assignment to the three folds.

The figure illustrates the idea that in a good stratification, the distance between centroids ($\boldsymbol{\gamma}_i$) should be as small as possible. Thus, the proposed criterion allows the determination of the number of clusters by taking into account the heterogeneity of data in the clusters and the homogeneity of data in the folds.

4. Experimental methodology

In the remaining parts of the paper, we are assessing the stratification by clustering methodology presented in the previous section in the context of one-step ahead Hourly Ontario Energy Price (HOEP) forecasting. Note that the central idea of this work is neither improving the qualities of generalization of an estimator nor to develop the most generalized and robust forecast model. The idea is rather to analyze the relation between the error and the number of elements used to train a model for time series forecasting, in order to observe whether the method proposed for clustering-based stratification of time series allows the training of a model with a smaller data set while having little or no impact on performance.

4.1. Hourly Ontario Energy Price

To assess our method, we selected the Hourly Ontario Energy Price, a dataset which has been used in several other papers [14, 18–20].

In the Ontario market, the Independent Electricity System Operator (IESO)[1] publishes the Hourly Ontario Energy Price (HOEP) and Hourly Ontario Demands (HOD). The HOEP is the hourly price ($\$/MWh$) that is charged to local distribution companies, other non-dispatchable loads, and self-scheduling generators. Currently, Ontario has a unique hourly price for the whole province, that is, the price we address in this paper. Market demand represents the total energy that was supplied from the IESO-administered market.

Raw data of HOEP and HOD is shown in Figure 2. The peaks HOEP presented in this figure are related with to high reserve requirements.

4.2. Data pre-processing

In order to limit the negative effects of the price outliers and to allow a comparison of the variables of different units and scales such as price (HOEP) and demand (HOD), a suitable pre-processing of the original database has been proposed. Since the prices over $\$200/MWh$ are treated as anomalous prices[2], the pre-processing scheme is formulated such that if the HOEP is above than $\$200/MWh$, it will be replaced with a demand weighted average of the HOEPs of three previous days as shown below [21]:

$$P_t = \begin{cases} P_t & \text{if } P_t < \$200/MWh \\ \dfrac{\sum_{i=1}^{3}(P_{t-168i} \times D_{t-168i})}{\sum_{i=1}^{3}(P_{t-168i})} & \text{if } P_t \geq \$200/MWh \end{cases}. \tag{6}$$

where $P = HOEP$ and $D = HOD$. Then, standardization is used on the feature component of data set $\boldsymbol{L} = [\boldsymbol{l}(1), \boldsymbol{l}(2), \ldots, \boldsymbol{l}(N)] \in \mathbb{R}^{N \times n+m}$ to provide the zero mean and unit standard deviation:

$$\hat{\boldsymbol{l}}^i = \frac{\boldsymbol{l}^i - \mu^i}{\sigma^i}, \; i = \{1, \ldots, n+m\}, \tag{7}$$

[1] http://www.ieso.ca/imoweb/marketdata/marketData.asp

[2] See http://www.ontarioenergyboard.ca/OEB/Industry/About%20the%20OEB/Electricity%20Market%20Surveillance/Market%20Surveillance%20Panel%20Reports

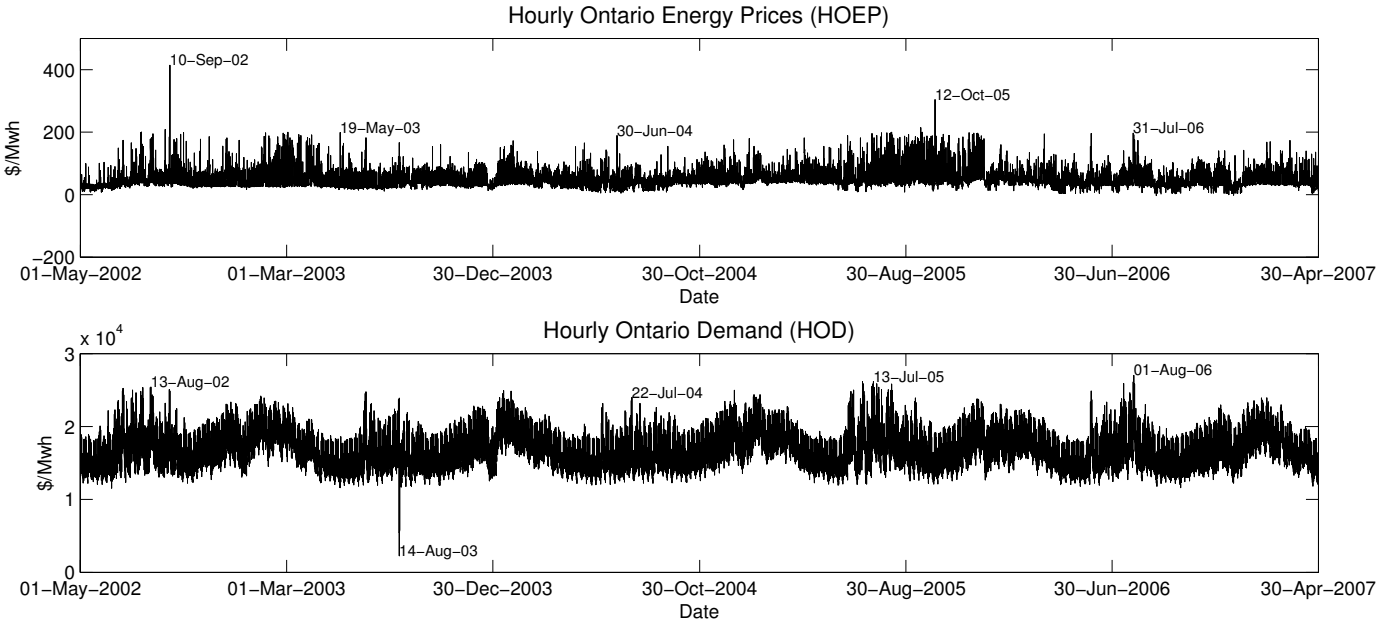


Figure 2: Hourly Ontario Energy Price (HOEP) and Hourly Ontario Demands (HOD) from May 1, 2002 to April 30, 2006.

where $\boldsymbol{l}^i$ is the i-th component of data set $\boldsymbol{L}$, $\hat{\boldsymbol{l}}^i$ is the corresponding normalized value, μ^i is the mean of the i-th component, and σ_i is the standard deviation.

4.3. Training and testing sets

First we divide the HOEP series into different years:

$$\mathcal{X} = \mathcal{X}_o \cup \mathcal{X}_{o+1} \cup \ldots \cup \mathcal{X}_e. \tag{8}$$

In the current case, we used $o = 2002$ as the initial year and $e = 2006$ as the final year composing the datasets. A year of HOEP time series begins on May 1 and ends on April 30. Then we apply a sliding window to create training and test datasets. Training years correspond to years o to $e-1$, that is years 2002 to 2005. For each training year, the following one is used for testing. For instance, for training year 2005 the corresponding testing year is 2006.

Each training set $\mathcal{X}_i$ is processed using the stratification by clustering approach described in Sec. 3.

4.4. Measures of accuracy

To assess the prediction accuracy of forecasting models, the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) criteria are commonly used [22]. The MAPE is calculated as:

$$\text{MAPE} = \frac{100}{N^T} \sum_{i=1}^{N^T} \left| \frac{p_t^{act} - p_t^{for}}{p_t^{act}} \right|, \tag{9}$$

while the RMSE is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{N^T} \sum_{i=1}^{N^T} (p_t^{act} - p_t^{for})^2}, \tag{10}$$

where p_t^{act} is the actual value at time step t and p_t^{for} is the forecasted value of HOEP. N^T is the number of data used in test.

However, it appears that with HOEP, there are some data that are deviating significantly from the average values of the time series, which would have a exaggerated effect on the MAPE and RMSE given their scale. For that purpose, we decided to make use of the trimmed MAPE and trimmed RMSE instead, where the highest 5 % error values are removed. We defined these measures first by defining a trimming function that is applied on the absolute errors:

$$E^t = \left| p_t^{act} - p_t^{for} \right|, \tag{11}$$

$$E = \{E^1, E^2, \ldots, E^N\}, \tag{12}$$

$$\text{trim}(E, \theta) = \{E^t \in E \mid E^t \leq \theta\}. \tag{13}$$

Then, we evaluate the trimming threshold E' as the following:

$$U = \{E^t \in E \mid \text{n}(\text{trim}(E, E^t)) \geq 0.95N\}, \tag{14}$$

$$E' = \underset{u \in U}{\text{argmin}}(\text{n}(\text{trim}(E, u))), \tag{15}$$

where $\text{n}(\cdot)$ is the cardinality of a set.

From this, the trimmed MAPE criteria with the 5 %

highest error values removed are calculated as:

$$E_{\text{trim}} = \text{trim}(E, E'), \quad (16)$$

$$\text{trimmed MAPE} = \frac{100}{\text{n}(E_{\text{trim}})} \sum_{E^t \in E_{\text{trim}}} \left| \frac{E^t}{x^t} \right|. \quad (17)$$

Likewise, the trimmed RMSE criteria is calculated as follows:

$$\text{trimmed RMSE} = \sqrt{\frac{1}{\text{n}(E_{\text{trim}})} \sum_{E^t \in E_{\text{trim}}} (E^t)^2}. \quad (18)$$

5. Results and discussion

In this section we examine whether the performance of the proposed stratification using the clustering methodology (*strat*) is better than a random selection of training subsets (*rand*). For this, we proposed to carry out a comparison between samples of different size obtained from a *strat* and *random* methodology, and use them to forecast one-step ahead Hourly Ontario Energy Price with two regression techniques: Artificial Neural Networks (ANN) and Support Vector Regression (SVR).

For the ANN, the configuration used consists in a feed-forward network trained using the Levenberg Marquardt algorithm, as implemented in the Matlab Neural Network toolbox [23]. The training subset is further split into two subsets, that is the learning subset, which contains 67% of the data, and the validation subset, which includes the remaining 33%. For the stratification by clustering method, these subsets are generated following instructions given in step 9 of the procedure presented in Sec. 3.2. For the random selection, the two subsets are produced through a random partitioning. The learning subset is used to carry out the actual training of the ANN with the backpropagation method, while the validation subset is used to measure the generalization capability (empirical error) of the network. Over all networks produced at each training epoch, we retain the one minimizing the error rate on the validation subset. A three-layered ANN has been selected having 10 nodes on each of the two hidden layers with a sigmoid transfer function and one output node with a linear transfer function.

For the SVR, the ν-SVR version with RBF (Gaussian) kernel, as implemented in LIBSVM [24], is used. LIBSVM allows a grid search to be conducted over all combinations of hyper-parameters considered for the learning algorithm. This is achieved by evaluating the cross-validation (CV) accuracy for each hyper-parameter combination, and then returning the one with the highest CV accuracy. For our experiments, the hyper-parameter values evaluated are C and ν. Once the SVR hyper-parameters have been determined, the final training is carried out, this time using the complete training subset.

The two regression models have been run 30 times for each size of training subset. The training subset sizes tested correspond to 5 %, 10 %, 25 %, 50 %, and 100 % of the available data.

5.1. Features

Since variable selection is not the topic of the paper, we used the following HOEP and HOD lagged values for our experiments, as proposed by [18]:

$$\boldsymbol{a} = (1, 2, 23, 24, 25, 48, 120, 144, 168, 169, 192),$$
$$\boldsymbol{b} = (1, 23, 24, 25, 144, 167, 168, 169, 192),$$

where $\boldsymbol{a}$ are lags used for price values used as inputs, and $\boldsymbol{b}$ are the lags for the demand values.

5.2. Clusters

In order to select the number of clusters to use, we tested stratification with $F \in \{2, 4, 10, 20\}$ folds corresponding to a subset with 50%, 25%, 10%, and 5% of the full dataset, respectively, in combination with $K \in \{2, 3, ..., 15\}$ clusters. For each configuration, we repeated the clustering 10 times, each with a new set of initial cluster centroid positions, selecting the solution with the lowest value of Eq. 5 on average over the various number of clusters tested.

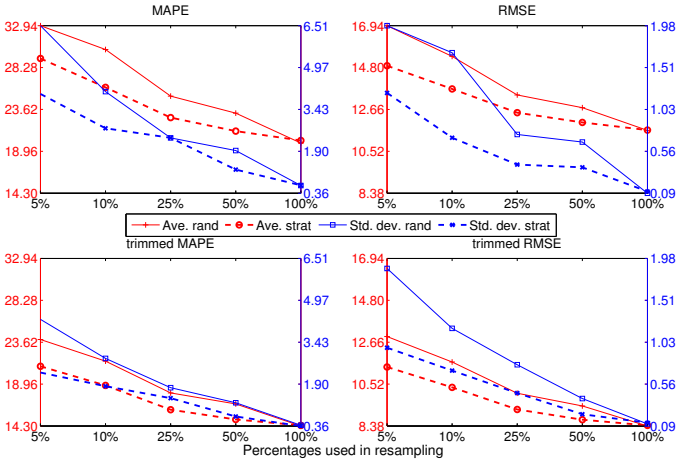
5.3. Results on using ANN and SVR on data selected with stratification by clustering

Figure 3 and 4 present visually the results obtained with the ANN and SVR, respectively. Figures show the average and standard deviation of the trimmed MAPE, the trimmed RMSE, the MAPE, and the RMSE for the results obtained on the test years, for 1-hour ahead HOEP forecasts using data selected with the methodology applied in the input space and the input state. The year represents the period used for the training.

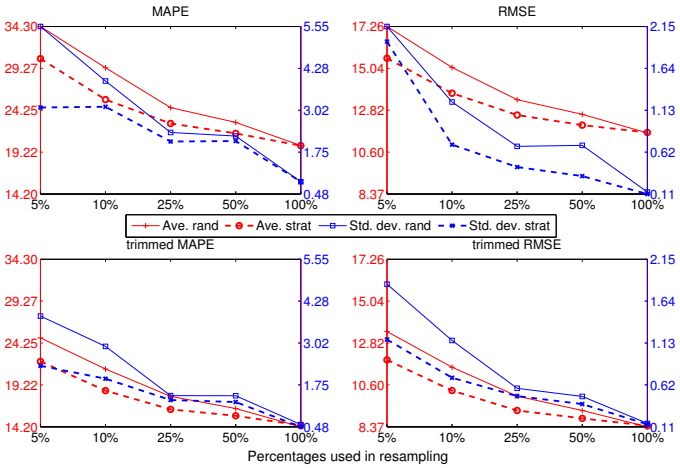
Tables 1 and 2 present trimmed MAPE and RMSE obtained on the test sets for 30 experiments for 1-hour ahead HOEP forecasts generated using ANN or SVR with different training subset sizes. The year represents the period of training, *strat* represents the results generated with samples obtained with the methodology of stratification, while *rand* represents the results produced with samples randomly obtained.

It can be observed that errors obtained with subsets produced with the stratification by clustering method are generally lower in comparison with errors obtained with randomly generated subsets. To assess the statistical validity conducted with the stratification methodology proposed, a non-parametric Mann-Whitney U-test has been used [25]. The null hypothesis is that the means of the random or stratified methodology are equal at a 95% significance level. Looking at the standard deviation, it also appears that the results generated by stratification are much more stable than those produced with randomly selected subsets, demonstrating the general robustness of the stratification by clustering approach.

Moreover, we explain the slight difference in the response mean with 100% of the data by the application of step 7 of the procedure presented in Sec. 3.2, in which the learning and validation sets are generated following the



(a) Input state



(b) Input space

Figure 3: ANN forecasting 1-step ahead HOEP for year 2006, with year 2005 used as training set and stratification by clustering applied in the (a) input state and (b) input space

(a) Input space

		5%	10%	25%	50%	100%
MAPE	strat	**19.22** (1.95)	**18.31** (1.79)	**15.20** (0.77)	**14.38** (0.58)	13.68 (0.57)
2002	rand	23.28 (3.60)	19.95 (1.81)	17.07 (1.38)	15.29 (0.61)	13.87 (0.63)
RMSE	strat	**11.21** (0.81)	**10.58** (0.71)	**9.25** (0.29)	**8.87** (0.19)	8.53 (0.12)
2002	rand	13.58 (1.59)	11.51 (0.92)	10.06 (0.60)	9.19 (0.23)	8.61 (0.16)
MAPE	strat	**14.38** (1.15)	**13.27** (0.94)	**11.97** (0.51)	**11.40** (0.24)	11.05 (0.25)
2003	rand	16.08 (1.51)	14.17 (1.55)	12.83 (0.71)	12.01 (0.46)	10.96 (0.21)
RMSE	strat	**9.14** (0.70)	**8.49** (0.59)	**7.71** (0.24)	**7.39** (0.13)	7.18 (0.11)
2003	rand	10.29 (0.88)	9.11 (0.97)	8.19 (0.39)	7.75 (0.26)	7.16 (0.09)
MAPE	strat	**19.34** (1.77)	**17.46** (1.11)	**16.15** (1.18)	**14.88** (0.75)	14.35 (0.50)
2004	rand	24.07 (5.74)	20.72 (2.12)	17.48 (1.31)	16.04 (1.25)	14.41 (0.47)
RMSE	strat	**17.82** (2.47)	**16.20** (2.07)	**15.77** (2.34)	**14.10** (1.17)	13.50 (0.78)
2004	rand	22.24 (6.85)	19.36 (3.05)	16.38 (1.91)	15.31 (1.99)	13.60 (0.86)
MAPE	strat	**22.03** (2.33)	**18.53** (1.94)	**16.28** (1.29)	**15.51** (1.23)	14.36 (0.48)
2005	rand	24.86 (3.83)	21.13 (2.91)	17.84 (1.42)	16.38 (1.42)	14.20 (0.55)
RMSE	strat	**11.92** (1.17)	**10.29** (0.70)	**9.24** (0.48)	**8.82** (0.38)	8.42 (0.14)
2005	rand	13.43 (1.84)	11.53 (1.16)	10.00 (0.57)	9.24 (0.48)	8.37 (0.15)

(b) Input state

		5%	10%	25%	50%	100%
MAPE	strat	19.97 (2.71)	**17.32** (1.32)	**15.29** (1.13)	**14.32** (0.71)	13.69 (0.49)
2002	rand	21.44 (2.96)	20.23 (2.98)	16.94 (1.20)	15.45 (0.87)	13.73 (0.53)
RMSE	strat	11.55 (1.19)	**10.30** (0.58)	**9.26** (0.43)	**8.78** (0.23)	8.55 (0.13)
2002	rand	12.62 (1.55)	11.66 (1.47)	10.01 (0.52)	9.34 (0.45)	8.57 (0.14)
MAPE	strat	**14.40** (1.21)	**13.28** (1.03)	**11.97** (0.50)	**11.31** (0.31)	11.12 (0.20)
2003	rand	16.48 (2.15)	14.18 (1.03)	12.78 (0.61)	12.00 (0.51)	**11.00** (0.17)
RMSE	strat	**9.18** (0.72)	**8.42** (0.52)	**7.70** (0.26)	**7.35** (0.16)	7.23 (0.07)
2003	rand	10.31 (1.23)	8.98 (0.57)	8.17 (0.35)	7.73 (0.29)	**7.19** (0.11)
MAPE	strat	**20.21** (2.45)	**17.68** (1.39)	**16.19** (1.05)	**14.94** (0.65)	14.18 (0.40)
2004	rand	24.14 (4.80)	20.38 (2.89)	18.20 (2.65)	16.24 (1.16)	14.44 (0.45)
RMSE	strat	**19.06** (4.12)	**16.43** (2.04)	**15.39** (1.74)	**14.03** (1.02)	13.27 (0.56)
2004	rand	23.71 (7.89)	19.57 (4.31)	18.07 (4.94)	15.61 (2.09)	13.65 (0.77)
MAPE	strat	**20.94** (2.32)	**18.82** (1.82)	**16.11** (1.38)	**15.03** (0.71)	14.36 (0.36)
2005	rand	23.92 (4.27)	21.52 (2.84)	17.98 (1.77)	16.75 (1.21)	14.30 (0.39)
RMSE	strat	**11.39** (0.97)	**10.36** (0.71)	**9.23** (0.46)	**8.71** (0.22)	8.41 (0.13)
2005	rand	12.95 (1.87)	11.65 (1.19)	10.06 (0.78)	9.41 (0.40)	8.38 (0.11)

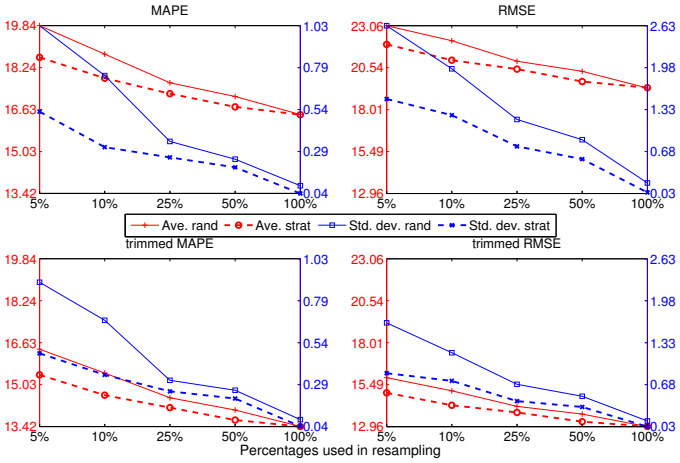
Table 1: Trimmed MAPE and RMSE in 1-hour ahead HOEP forecast with ANN with stratification done in the (a) input space and (b) input state. The values in parentheses correspond to the standard deviation over the 30 experiences. The percentage indicates the size of the sample taken from the data of the corresponding year. Reported values are testing errors over the year following the training year mentionned in first column. Results in bold are statistically better with a 95% significance level according to a Mann-Whitney U-test, when comparing the selection stratified of data with the corresponding one obtained with random selection.

stratification by clustering approach, while our comparison with the full training set assumes that the learning and validation subsets are produced through a random partition. It appears that this difference decreases as the subsets become larger for both the stratification and random selection approaches, which is expected. But the difference is generally smaller and less dispersed for stratification by clustering in comparison to the random selection. These results suggest that with 25% data generated by stratification by clustering, the MAPE and RMS errors are still relatively close to what is obtained when using the full training set in most cases.

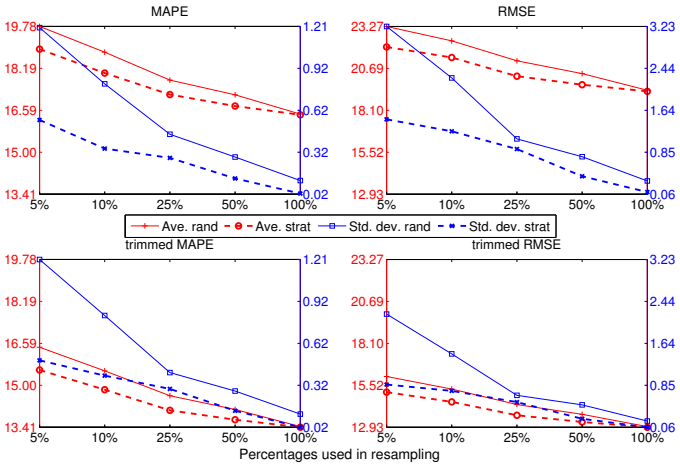
Figure 5 presents the differences between the trimmed MAPE of the 30 experiments and the trimmed average error when all training data is used. Results are for forecasting 1-step ahead HOEP for year 2004, trained on year 2003 data. It appears that this difference decreases as the subsets become larger for both the stratification and random selection approaches, which is expected. But the difference is generally smaller when the examples of training were selected with the stratification methodology proposed.

As indicated in the LIBSVM guide [24], although the RBF kernel is a reasonable first choice, there are some situations where the RBF kernel is not suitable. In particular, when the number of features is very large. In this case, other improvements, such as the stratification of the folds (construct folds deterministically rather than randomly) to use in the v-fold cross-validation, may be needed to reduce the bias. Other techniques such as feature selection may be needed.

The K-means procedure is deterministic as far as a given set of initial centre positions will always lead to the same final results when using the same parametrization. However, the choice of the initial positions of the centres is arbitrary and often made through a random process, for example by generating K random positions uniformly in the data domain or by making a random selection of K instances in the dataset. So, from one execution to another, the results of K-means may differ given that different initial positions for the K centres will be used. Nevertheless, K-means is known to be relatively stable in practise, as the



(a) Input state



(b) Input space

Figure 4: SVR forecasting 1-step ahead HOEP for year 2005, with year 2004 used as training set and stratification by clustering applied in the (a) input state and (b) input space

number of distinct clustering results obtained at convergence is relatively limited, such that carrying out several repetitions of K-means should provide a good sampling of the possible solutions.

6. Conclusion

In this paper, we proposed a new stratification by clustering method that generates training subsets for forecasting. This method enables the production of small subsets composed of representative samples of the original dataset. The stratification method has been tested on forecasting Hourly Ontario Energy Price (HOEP), with results for the years 2002 to 2006. The stratification by clustering method proposed has also been compared to a random selection of subsets, using Artificial Neural Network (ANN) and Support Vector Regression (SVR) as forecasting models.

Results obtained by the stratification by clustering are

(a) Input space

		5%	10%	25%	50%	100%
MAPE	strat	**13.64** (0.54)	**13.03** (0.43)	**12.50** (0.20)	**12.18** (0.13)	11.93 (0.06)
2002	rand	14.65 (0.80)	13.52 (0.48)	12.95 (0.34)	12.50 (0.18)	11.92 (0.07)
RMSE	strat	**8.77** (0.18)	**8.50** (0.14)	**8.27** (0.05)	**8.15** (0.04)	8.04 (0.02)
2002	rand	9.27 (0.34)	8.76 (0.20)	8.47 (0.10)	8.27 (0.05)	8.04 (0.03)
MAPE	strat	**11.41** (0.24)	**11.01** (0.26)	**10.75** (0.14)	**10.65** (0.09)	**10.53** (0.01)
2003	rand	11.97 (0.45)	11.45 (0.34)	10.87 (0.16)	10.78 (0.17)	10.59 (0.04)
RMSE	strat	**7.39** (0.12)	**7.19** (0.15)	**7.06** (0.07)	**7.03** (0.05)	7.03 (0.02)
2003	rand	7.68 (0.22)	7.41 (0.18)	7.12 (0.07)	7.08 (0.08)	**7.00** (0.03)
MAPE	strat	**15.58** (0.50)	**14.83** (0.39)	**14.05** (0.29)	**13.69** (0.14)	**13.41** (0.02)
2004	rand	16.45 (1.21)	15.55 (0.82)	14.61 (0.41)	14.08 (0.28)	13.43 (0.12)
RMSE	strat	**15.10** (0.87)	**14.50** (0.75)	**13.67** (0.53)	**13.26** (0.22)	**12.93** (0.06)
2004	rand	16.06 (2.20)	15.29 (1.44)	14.33 (0.66)	13.73 (0.48)	12.97 (0.18)
MAPE	strat	**14.67** (0.64)	**14.01** (0.44)	**13.61** (0.22)	**13.53** (0.14)	13.38 (0.01)
2005	rand	15.75 (1.49)	14.78 (0.95)	14.07 (0.34)	13.73 (0.21)	13.37 (0.01)
RMSE	strat	**8.68** (0.16)	**8.38** (0.12)	**8.19** (0.07)	**8.11** (0.04)	8.03 (0.01)
2005	rand	9.15 (0.46)	8.71 (0.29)	8.37 (0.11)	8.21 (0.06)	8.03 (0.01)

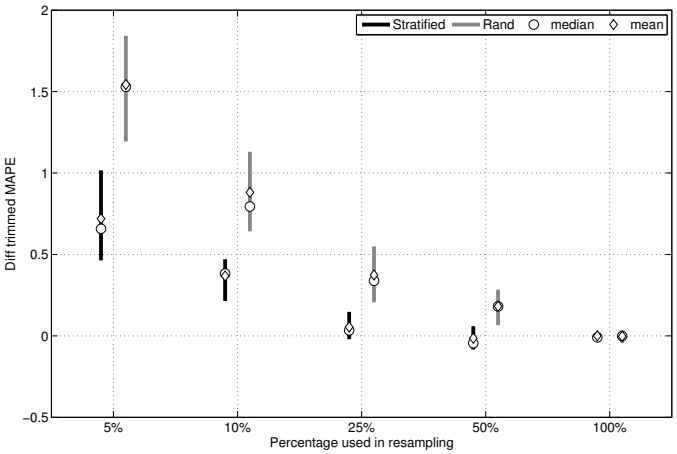
(b) Input state

		5%	10%	25%	50%	100%
MAPE	strat	**13.39** (0.48)	**12.96** (0.24)	**12.40** (0.08)	**12.16** (0.13)	11.91 (0.04)
2002	rand	14.51 (0.82)	13.69 (0.52)	12.93 (0.26)	12.48 (0.18)	11.91 (0.07)
RMSE	strat	**8.70** (0.18)	**8.47** (0.10)	**8.30** (0.04)	**8.14** (0.05)	8.04 (0.02)
2002	rand	9.20 (0.23)	8.81 (0.23)	8.43 (0.07)	8.26 (0.06)	8.04 (0.02)
MAPE	strat	**11.37** (0.38)	**11.01** (0.19)	**10.69** (0.12)	**10.62** (0.10)	10.63 (0.01)
2003	rand	12.14 (0.42)	11.48 (0.34)	10.95 (0.21)	10.77 (0.15)	**10.58** (0.05)
RMSE	strat	**7.41** (0.20)	**7.20** (0.08)	**7.04** (0.05)	**7.01** (0.05)	7.04 (0.01)
2003	rand	7.81 (0.25)	7.41 (0.16)	7.16 (0.10)	7.07 (0.07)	**7.00** (0.03)
MAPE	strat	**15.40** (0.47)	**14.63** (0.35)	**14.15** (0.25)	**13.67** (0.21)	13.42 (0.04)
2004	rand	16.38 (0.89)	15.47 (0.67)	14.53 (0.31)	14.06 (0.26)	13.43 (0.08)
RMSE	strat	**14.99** (0.86)	**14.24** (0.74)	**13.81** (0.42)	**13.26** (0.33)	12.97 (0.03)
2004	rand	15.91 (1.64)	15.12 (1.17)	14.18 (0.68)	13.71 (0.50)	12.96 (0.12)
MAPE	strat	**14.45** (0.49)	**14.19** (0.32)	**13.63** (0.24)	**13.52** (0.22)	**13.37** (0.00)
2005	rand	15.89 (1.23)	14.65 (0.74)	14.02 (0.41)	13.66 (0.23)	13.37 (0.02)
RMSE	strat	**8.63** (0.17)	**8.40** (0.10)	**8.20** (0.08)	**8.11** (0.06)	**8.03** (0.00)
2005	rand	9.20 (0.37)	8.69 (0.19)	8.34 (0.11)	8.21 (0.07)	8.03 (0.01)

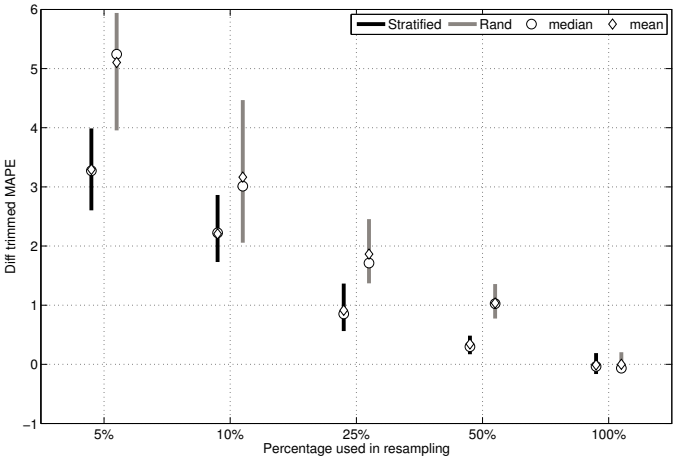
Table 2: Trimmed MAPE and RMSE in 1-hour ahead HOEP forecast with SVR with the stratification conducted in the (a) input space and (b) input state. The values in parentheses correspond to the standard deviation over the 30 experiments. The percentage indicates the size of the sample taken from the data of the corresponding year. Reported values are testing errors over the year following the training year mentionned in first column. Results in bold are statistically better with a 95% significance level according to a Mann-Whitney U-test, when comparing the selection stratified of data with the corresponding one obtained with random selection.

shown to be better in general than the a random selection method for producing a representative subset of a given size. Moreover, the stratification procedure appears relevant for creating subsets having less than 50% of the original dataset, whose subsets still allow the learning of forecasting models having performances comparable to the models trained on the full dataset.

Using small training subset is interesting in many ways. First, it can significantly decrease the time required to train a forecasting model. For instance, state-of-the-art SVR implementation has quadratic time complexity according to the number of data in the training set. Thus, training a SVR with more than 100 000 training instances is hardly feasible in practise. The current procedure is quite straightforward, with a linear complexity in time and space, such that it can be applied to select smaller subsets from large datasets, in order to keep training time at reasonable levels. Another situation where this is particularly interesting for study of forecasting models. An example of this would be to determine at first the relevant inputs to use with the forecasting methods, such as conducting feature selections using a wrapper approach [26].



(a) SVR in input state



(b) ANN in input space

Figure 5: Average and dispersion of the trimmed MAPE for stratification by clustering and random selection over years 2002-2006 obtained with (a) SVR in the input state and (b) ANN in the input space.

These methods may have prohibitive computational cost when working with the full datasets, while increasing the risk of oversearching the space of forecasting methods [27]. Working on smaller but representative subsets for hyperparameter tuning or feature selection allows the computation time to be reduced, while optimizing over only a small part of the full training set, keeping the rest of the training set untouched for the final training.

Acknowledgements

This work was supported by funding from the Fonds de Recherche Québécois sur la Nature et les Technologies (FRQNT) and access to computational resources of Calcul Québec/Compute Canada and the program *Fortalecimiento a programas de Doctorados, Maestrías y Especialización* (RES CFIA 330 de 2010) financed by *Fondo de Investigación de la Universidad Nacional de Colombia.*

We are grateful to Annette Schwerdtfeger for proofreading this manuscript.

References

[1] N. Diamantidis, D. Karlis, E. A. Giakoumakis, Unsupervised stratification of cross-validation for accuracy estimation, Artif. Intell. 116 (1-2) (2000) 1–16. doi:http://dx.doi.org/10.1016/S0004-3702(99)00094-6.

[2] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. M. Cam, J. Neyman (Eds.), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, California, USA, 1967, pp. 281–297.

[3] A. Maus, J. Sprott, Neural network method for determining embedding dimension of a time series, Commun. Nonlinear Sci. Numer. Simulat. 16 (8) (2011) 3294–3302. doi:10.1016/j.cnsns.2010.10.030.

[4] S. Tong, D. Koller, Support Vector Machine active learning with applications to text classification, JMLR 2 (2002) 45–66. doi:10.1162/153244302760185243.

[5] L. Franco, S. A. Cannas, Generalization and selection of examples in feed forward neural networks, Neural Comput. 12 (2000) 2405–2426. doi:10.1162/089976600300014999.

[6] A. Röbel, Dynamic pattern selection: Effectively training backpropagation neural networks, in: M. Marinaro, P. G. Morasso (Eds.), ICANN '94, Springer London, 1994, pp. 643–646. doi:10.1007/978-1-4471-2097-1_151.

[7] M. Plutowski, H. White, Selecting concise training sets from clean data, IEEE Trans. on Neural Netw. 4 (2) (1993) 305–318. doi:10.1109/72.207618.

[8] F. Leisch, L. Jain, K. Hornik, Cross-validation with active pattern selection for neural-network classifiers, IEEE Trans. on Neural Netw. 9 (1) (1998) 35–41. doi:10.1109/72.655027.

[9] J. Wang, P. Neskovic, L. N. Cooper, Training data selection for support vector machines, in: L. Wang, K. Chen, Y. Ong (Eds.), Advances in Natural Computation, Vol. 3610, Springer Berlin Heidelberg, 2005, pp. 554–564. doi:10.1007/11539087_71.

[10] M. Barros de Almeida, A. de Padua Braga, J. Braga, SVM-KM: speeding SVMs learning with a priori cluster selection and k-means, in: Proc. of the Brazilian Symposium on Neural Networks, 2000, pp. 162–167. doi:10.1109/SBRN.2000.889732.

[11] Z. Songfeng, L. Xiaofeng, Z. Nanning, X. Weipu, Unsupervised clustering based reduced support vector machines, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 2003, pp. 821–824. doi:10.1109/ICASSP.2003.1202493.

[12] K. Chakrabarti, E. Keogh, S. Mehrotra, M. Pazzani, Locally adaptive dimensionality reduction for indexing large time series databases, ACM T. Database Syst. 27 (2) (2002) 188–228. doi:10.1145/568518.568520.

[13] J. An, Y.-P. P. Chen, H. Chen, DDR: an index method for large time-series datasets, Information Systems 30 (5) (2005) 333–348. doi:10.1016/j.is.2004.05.001.

[14] H. Zareipour, K. Bhattacharya, C. Canizares, Forecasting the hourly Ontario energy price by multivariate adaptive regression splines, in: Power Engineering Society General Meeting, 2006, pp. 1–7. doi:10.1109/PES.2006.1709474.

[15] T. W. Liao, Clustering of time series data – a survey, Pattern Recogn. 38 (11) (2005) 1857–1874. doi:10.1016/j.patcog.2005.01.025.

[16] F. Martínez-Álvarez, A. Troncoso, J. Riquelme, J. Riquelme, Partitioning-clustering techniques applied to the electricity price time series, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), Intelligent Data Engineering and Automated Learning, Vol. 4881, Springer Berlin Heidelberg, 2007, pp. 990–999. doi:10.1007/978-3-540-77226-2_99.

[17] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd Edition, Wiley, New York, 2001.
URL http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html

[18] N. Amjady, A. Daraeepour, F. Keynia, Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network, IET Gener. Transm. Distrib. 4 (3) (2010) 432–444. doi:10.1049/iet-gtd.2009.0297.

[19] A. I. Arciniegas, I. E. Arciniegas Rueda, Forecasting short-term power prices in the Ontario Electricity Market (OEM) with a fuzzy logic based inference system, Utilities Policy 16 (1) (2008) 39–48. doi:10.1016/j.jup.2007.10.002.

[20] C. Rodriguez, G. Anders, Energy price forecasting in the Ontario competitive power system market, IEEE Trans. Power Syst. 19 (1) (2004) 366–374. doi:10.1109/TPWRS.2003.821470.

[21] H. Zareipour, Price forecasting and optimal operation of wholesale customers in a competitive electricity market, Ph.D. thesis, University of Waterloo, Waterloo, ON, Canada (2006).
URL https://uwspace.uwaterloo.ca/bitstream/handle/10012/2611/Hamid_Zareipour_Thesis.pdf

[22] J. Zhang, C. Cheng, Day-ahead electricity price forecasting using artificial intelligence, in: Electric Power Conference, 2008, pp. 1–5. doi:10.1109/EPC.2008.4763317.

[23] M. H. Beale, M. T. Hagan, H. B. Demuth, Neural Network Toolbox User's Guide, Available online at http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf., 2013.
URL www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf

[24] C.-C. Chang, C.-J. Lin, Libsvm: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 1–39. doi:10.1145/1961189.1961199.

[25] N. Kottegoda, R. Rosso, Applied Statistics for Civil and Environmental Engineers, John Wiley & Sons, 2008.
URL http://books.google.ca/books?id=S7FSnnA6kBIC

[26] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, JMLR 3 (2003) 1157–1182.
URL http://jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[27] D. D. Jensen, P. Cohen, Multiple comparisons in induction algorithms, Mach. Learn. 38 (3) (2000) 309–338. doi:10.1023/A:1007631014630.