Energy Consumption Prediction with Big Data: Balancing Prediction Accuracy and Computational Resources

Katarina Grolinger, Miriam A. M. Capretz
Department of Electrical and Computer Engineering
Western University
London, ON, Canada N6A 5B9
{kgroling, mcapretz}@uwo.ca

Luke Seewald
London Hydro
London, ON, Canada N6A 4H6
seewaldl@londonhydro.com

***Abstract*—In recent years, advances in sensor technologies and expansion of smart meters have resulted in massive growth of energy data sets. These Big Data have created new opportunities for energy prediction, but at the same time, they impose new challenges for traditional technologies. On the other hand, new approaches for handling and processing these Big Data have emerged, such as MapReduce, Spark, Storm, and Oxdata H2O. This paper explores how findings from machine learning with Big Data can benefit energy consumption prediction. An approach based on local learning with support vector regression (SVR) is presented. Although local learning itself is not a novel concept, it has great potential in the Big Data domain because it reduces computational complexity. The local SVR approach presented here is compared to traditional SVR and to deep neural networks with an H2O machine learning platform for Big Data. Local SVR outperformed both SVR and H2O deep learning in terms of prediction accuracy and computation time. Especially significant was the reduction in training time; local SVR training was an order of magnitude faster than SVR or H2O deep learning.**

Keywords: consumption prediction; Big Data; local learning; local SVR; deep learning, deep neural networks, Oxdata H2O.

I. INTRODUCTION

Modeling and forecasting electrical energy consumption has been an active research area for more than a decade. In the United States, retail sales of electricity exceed $3,760 billion [1], and the electricity sector generates the largest share of greenhouse gas emissions (31%) [2]. Today, with climate change and the focus on environment, it is even more important to model and forecast electricity consumption accurately in pursuit of conservation opportunities.

The importance of measuring and collecting electricity data, together with recent advances in sensor technology, have led to the proliferation of smart meters that measure and communicate electricity consumption. These smart meters measure electricity at intervals of an hour or less, whereas some sensor devices can measure consumption in real time. These Big Data have created opportunities to develop new ways of analyzing energy consumption, identifying potential savings, and measuring energy efficiency.

Sensor-based approaches to energy forecasting rely on readings from sensors or smart meters and contextual information such as meteorological information or work schedules to infer future energy behaviour. Typically, historical data such as temperature, day of the week, time of day, and energy consumption are fed into a machine learning model that learns from them and consequently can forecast future energy consumption. The accuracy of these sensor-based approaches is comparable or superior to traditional approaches based on modeling in depth the properties of a building [3].

A typical assumption of Big Data is that more data can lead to deeper insights and higher business value. This is especially true in machine learning, where algorithms can learn better from bigger data sets. However, massive data sets can be challenging to process [4, 5]. Many machine learning algorithms were designed with the assumption that the whole dataset fits into the memory [4]. Often these algorithms are of high algorithmic complexity and require large amounts of memory [6]. This gave rise to distributed processing approaches, such as MapReduce, which are suitable for algorithms that can be parallelized to a degree sufficient to take advantage of available nodes. Apache Mahout [7] is an example of a platform for machine learning based on the MapReduce paradigm. Another recent development is Apache Spark [8], which is a cluster computing framework based on distributed data sets and in-memory processing. Although Mahout and Spark offer machine learning capabilities with a constantly increasing number of algorithms, algorithms such as support vector regression (SVR) and neural networks (NN), which are the dominant approaches to electricity consumption prediction [9], are only available in a limited context.

Local learning has also been suggested as a suitable approach for Big Data [6]. This approach reduces computation time by dividing the training set into clusters of similar samples and building a separate model for each cluster. However, it is not clear if and how the use of Big Data approaches affects energy prediction accuracy or computation time.

In previous work, the authors have used traditional SVR and NN to predict the consumption of an event venue [9]. With readings at 15-min intervals and one year of training data, SVR parameter optimization using cross validation and SVR model training exceeded 24 hours. Therefore, it is important to seek other solutions to reduce training time.

This study explores Big Data approaches, specifically local learning and deep learning, in the context of electricity consumption prediction; it looks at how those approaches

K. Grolinger, M. A. M. Capretz, L. Seewald, Energy Consumption Prediction with Big Data: Balancing Prediction Accuracy and Computational Resources, Proc. of the IEEE BigData Congress, 2016;

compare to traditional SVR with respect to prediction accuracy and computation time. An approach to electricity consumption prediction based on local learning with support vector regression is presented. This local SVR approach is compared to traditional SVR and Oxdata H2O deep neural networks [10]. Although the data set used in the case study is not very large, it demonstrates how Big Data approaches can benefit energy prediction. Moreover, the advantage of the presented approach will be even larger with bigger data sets.

The rest of this paper is organized as follows: Section II introduces local SVR and Oxdata H2O, and Section III reviews related work. The methodology, including the data set, energy prediction with local SVR, and performance metrics, is described in Section IV. An evaluation is presented in Section V, and Section VI concludes the paper.

II. BACKGROUND

This section introduces local SVR and Oxdata H2O.

A. Local SVR

Support vector machines (SVM) [11] are supervised learning algorithms characterized by a high degree of generalization, which indicates the model's ability to perform accurately on new, previously unseen data. A form of SVM known as support vector regression (SVR) is used for regression tasks. From a training data set $\{(X_i, Y_i)\}_{i=1}^{i=N}$, where X is a vector of input variables and Y is a vector of output variables, SVR approximates the relationship between input and output variables as:

$$Y = W \cdot \Phi(X) + b, \tag{1}$$

where $\Phi(X)$ is a kernel function that non-linearly maps from the input space X to the high-dimensional feature space. Coefficients W and b are determined by minimizing the objective function:

$$Minimize\ \frac{1}{2}\|w\|^2 + C\frac{1}{N}\sum_{i=1}^{N}\xi_i + \xi_i^* \tag{2}$$

subject to the following constraints:

$$Y_i - W \cdot \Phi(X_i) - b \le \varepsilon + \xi_i, \tag{3}$$
$$W \cdot \Phi(X_i) + b - Y_i \le \varepsilon + \xi_i^*, \tag{4}$$
$$\xi_i, \xi_i^* \ge 0\ . \tag{5}$$

A weight vector W should be as flat as possible to achieve good generalization. The terms ξ_i and ξ_i^* capture residuals beyond the prescribed tolerance ε, and cost C is the penalty for errors greater than ε.

A common choice of kernel is the radial basis kernel, which is efficient to compute and has only one parameter γ (influence of each data point) that needs to be determined; hence, this work also uses the radial basis kernel.

Local SVR uses a local learning principle with SVR as a local predictor. Local learning is based on the assumption that training samples in the neighbourhood of the test sample are the best indicators of the response variable. This is not a new concept; in 1992, Bottou and Vapnik [12] presented local learning as a way of dealing with training data that are unevenly distributed in the input space. Their work examines two approaches: the simple approach selects k training samples in the vicinity of the test sample, trains the prediction model using only these k samples, and applies this model to the test sample. In the second approach, the structure of the learning model ensures that only neighbouring samples affect the response variable.

Although the idea of local learning is old, in recent years it has emerged as a feasible approach in the context of Big Data [6]. Specifically, the solutions based on the following idea are promising:

- Training: partition the training set into clusters, build prediction model for each cluster;
- Testing/prediction: for the test sample, determine the cluster membership and apply that cluster model to determine the response/prediction value.

The reasoning behind the local learning concept in Big Data is that for computationally intensive algorithms, it is faster to find solutions for k problems of size m/k than to find the solution for one problem of size m. For example, standard SVM training has $O(m^3)$ time and $O(m^2)$ space complexity, where m is the size of the training set [13]. For large data sets, this is computationally infeasible. By splitting the set into k clusters and training each cluster separately, the overall training time should be significantly reduced.

B. Oxdata H2O

Oxdata H2O is a scalable, open source machine learning platform for Big Data analytics [10]. Its in-memory distributed parallel processing enables massively scalable data analysis and therefore allows H2O to harness Big Data for business benefit. H2O can be run stand-alone or on top of the Big Data platforms like Hadoop and Spark when H2O brings in-memory machine learning to these Big Data platforms. Presently, H2O includes a number of common machine learning algorithms such as generalized linear models (linear regression, logistic regression, etc.), decision trees, gradient boosting, k-means, deep learning, and Naïve Bayes. For energy prediction, this work uses H2O deep learning.

The term *deep learning* refers to a family of algorithms that model data using multiple layers, with each one performing a non-linear transformation. Examples of such algorithms include deep neural networks, deep belief networks, and convolutional and recurrent deep NN.

H2O's deep learning follows the model of multi-layer, feedforward neural networks with entirely supervised training. Its in-memory processing, columnar compression, MapReduce capability, multi-threaded computation, and distributed computation provide efficient processing. Distributed computation uses the MapReduce approach: in the map phase, each node trains with local data with asynchronous threads, whereas in the reduce phase, model averaging is performed. In contrast to a Hadoop MapReduce task, an H2O MapReduce tasks is performed in memory. Repeated training produces different results because the Hogwild! approach [14] used for parallelization. In contrast to other parallelization techniques that require performance

degrading memory locking, the Hogwild! approach implements stochastic gradient descent without any locking. This lock free implementation is achieved by allowing threads to access shared memory, with the possibility of overwriting each other's results. By allowing these race conditions, H2O improves performance.

H2O deep learning is very flexible; it supports manual and adaptive learning rates with a number of tuning parameters, different regularization techniques such as dropout, L1 (Lasso), and L2 (Ridge), early stopping, and others. This makes it possible to fine-tune the prediction model, but determining the optimal parameters becomes challenging and time consuming.

III. Related Work

This section reviews related work in machine learning with Big Data, and in electricity consumption prediction.

A. Machine Learning with Big Data

Machine learning (ML) has been attracting renewed attention with the emergence of Big Data as it has been seen as a way of extracting value from data. ML platforms for Big Data started with disk-based approaches such as Apache Mahout [7] which inherits disk orientation from the underlying Hadoop architecture. Because disk access is slow, new memory-based approaches have been developed. Apache Spark and Oxdata H2O are examples of memory-based platforms, and even Mahout machine learning algorithms are transitioning to these platforms. Zhang *et al*. [15] reviewed in-memory Big Data management and processing. They distinguished two types of in-memory systems: batch-oriented systems such as Spark and H2O, and real time or stream processing systems such as Storm. The systems relevant to energy consumption prediction primarily belong to the batch category.

Al-Jarrah *et al*. [6] reviewed energy efficient machine learning approaches and new approaches with reduced memory requirements. They saw local learning as one of the key mechanisms for machine learning with Big Data because of its ability to reduce computation cost. They also considered deep learning to be an important technique as it promises to provide representation learning for complex problems. Although deep learning is not a new concept, it is experiencing a rebirth with recent developments in distributed processing. H2O deep learning is an example of recent deep learning approaches for Big Data.

The publications of Chen and Lin [16] and Najafabadi *et al*. [17] examined deep learning with Big Data and discussed the associated challenges. Both studies highlighted the role of dimensionality reduction, parallel processing, and distributed processing in deep network training. Our work takes advantage of parallel and distributed processing and performs dimensionality reduction, but only after the training data have been partitioned into clusters.

B. Electricity Consumption Prediction

In recent years, with the proliferation of smart meters, prediction efforts have shifted from annual to daily, hourly, and even 10- or 15-min consumption prediction. Approaches with such granularity are typically sensor-based; they rely on historical energy readings and meteorological information without the need for a deep understanding of the physical building structure. For example, Jain *et al*. [18] and Grolinger *et al*. [9] considered daily, hourly, and 10- or 15-min intervals and explored the prediction accuracy achieved with different data granularities.

Sensor-based approaches to electricity forecasting are diverse; a few examples are support vector regression (SVR), neural networks (NN), autoregressive integrated moving average (ARIMA) models, and gray prediction [19]. Suganthi and Samuel [19] reviewed models for electricity demand prediction and noted that NN have been used extensively. Ahmad *et al*. [20] also reviewed energy prediction, but they focussed strictly on the use of NN and SVR.

Variants of the SVR approach have also been proposed: Jung *et al.* [21] added a genetic algorithm to the least-squares support vector machine (LSSVM), whereas Elattar *et al*. [22] used locally weighted support vector regression. Our local SVR and the approach proposed by Elattar *et al.* are both based on the assumption that the neighbours are the best indicators of the response variable. However, while Elattar *et al*. modify the SVR risk function to accommodate a distance measure, our approach classifies training data and builds an SVR model for each cluster.

Jovanović *et al*. [23] examined an ensemble of various neural networks to predict heating energy consumption. The impact of various climatic variables on prediction has also been studied [24].

Whereas the studies discussed above focus on prediction accuracy and application of a prediction approach in a specific context, the present work explores if and how recent developments from the Big Data domain can benefit electricity consumption prediction. Frincu *et al.* [25] and Anjos *et al.* [26] have been concerned with Big Data in the energy sector. Frincu *et al*. proposed an approach for selecting the prediction model, whereas Anjos *et al*. took a streaming approach to energy management. In contrast, the work reported here looks at adapting Big Data machine learning to energy prediction. Kejela *et al*. [27] used H2O for energy prediction: whereas they used a gradient boosting machine, the present study used deep learning. Moreover, H2O is just one approach considered in the present work.

IV. Methodology

This section first introduces the data set. Next, a local SVR approach is described and performance metrics presented.

A. Data Set

The Green Button initiative [28] is an effort to provide utility consumers with automated access to their energy usage and the ability to securely share these data with third parties. Through this initiative, data from smart meters are provided in a standardized Green Button format. Presently, over 60 million consumers have access to their energy use in this format [28]. Consequently, this study uses past energy consumption available through Green Button.

The specific scenario considered is electricity consumption prediction for event venues such as sports

arenas, theatres, and conference centres. In this scenario, consumption patterns are not as strongly related to hours of the day and days of the week, as is the case with office buildings, but are driven by event schedules and event attributes such as event type (basketball, hockey, …) and seating capacity. In addition to electricity readings, the following attributes are considered:

- Day of the year: 1 to 365
- Day of the week: 1 to 7
- Hour of the day: 1 to 24
- Event day: indicates whether there was an event on the day of the reading
- Event type: category of events, such as basketball and hockey. Three input features are used, one for each of basketball, hockey, and other.
- Seating configuration: captures seating capacity for an event.

The data consist of one sample for each electricity reading, and the event schedule is captured through date/time attributes (day of the year, day of the week, hour of the day) and the event type. Samples corresponding to non-event periods have 0 for the event type, whereas those corresponding to time periods during events have an event type describing the category of event, such as basketball or hockey.

B. *Local SVR for Energy Prediction*

The objective of this paper is to evaluate various suggested approaches for Big Data processing with respect to accuracy and time, not necessarily to create a completely new approach. Hence, prediction with local SVR as described in this section relies mostly on already available components, but it combines them in a way that enables efficient energy prediction.

Fig. 1 describes the training and testing process for energy prediction using local SVR. First, in step 1, the data set is divided into a training and a testing set. Because energy prediction is a time series problem in which older data are used to predict newer data, a portion of the data at the end of the time series is reserved solely for testing. The remainder of the set is used for training and parameter optimization.

1) *Training*

The training phase, as typical in machine learning, starts with normalization (step A.2), which adjusts variables to a common scale, in this case zero to one, to avoid dominance of high-valued features.

Next, step A.3 performs feature weighting to capture the different relevance of predictor variables and to improve unsupervised clustering in step A.4. Feature weights represent the degree of influence of individual variables on the predicted value. The feature weighting is performed on the scaled data obtained from step A.2. Specifically, correlations are used for feature weighting. The correlation between each input/independent variable x and the output/dependent variable y, in this case energy consumption, is calculated as follows:

$$corr(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n\sigma_x\sigma_y}, \quad (6)$$

where n is the number of samples in the training data set, $\bar{x}$ and $\bar{y}$ are the means of x and y, and σ_x and σ_y are the standard deviations of x and y.

Each value for each input feature is weighted according to the correlation coefficient calculated for that specific feature. Here, correlation is used to weight the features, but other more sophisticated approaches could also be used, such as those based on mutual information criteria [29].

Next, the training data set is partitioned using k-means clustering (step A.4). Empirical methods exist for determining the number of clusters, such as those based on distortion, which measures the distance between each observation and its closest cluster center [30]. However, this study is not concerned with cohesion within or between clusters, but rather with selecting the value of k that results in the highest prediction accuracy. Therefore, k is selected by repeating the training process with different k values and choosing the value of k that achieves the highest prediction accuracy.

Clustering is followed by feature reduction, step A.5. Feature reduction is carried out separately for each cluster; hence, models corresponding to different clusters may have different parameters. In the case study, because of the small number of input features, a simple approach was used: features that had the same value for all data points in a particular cluster were removed. For machine learning with a large number of features, it is better to use other dimensionality reduction techniques such as principal component analysis (PCA). PCA transforms a set of possibly correlated variables into a set of linearly uncorrelated variables, referred to as principal components, using orthogonal transformation. Then dimensionality can be

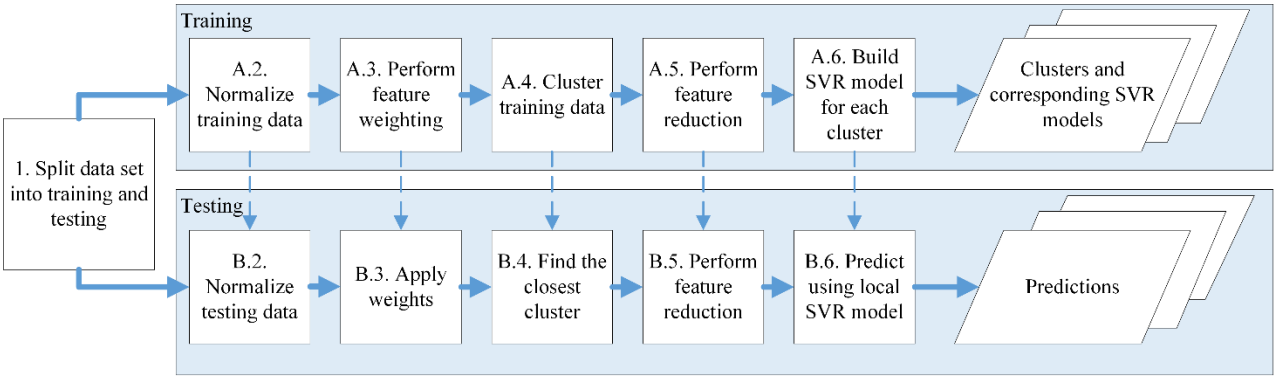

Figure 1. Local SVR process.

reduced by choosing only the first *p* principal components.

This dimensionality reduction is especially important for wide Big Data sets because it can reduce computational complexity. Nevertheless, in the presented case study, even a simple removal of features with a single value within a cluster resulted in a greatly reduced feature space and improved performance.

The process continues by building a separate SVR model for each cluster (step A.6). This includes selecting model parameters and training the model. For SVR, the two main parameters to be selected are ε, which defines which residuals are not penalized and the cost *C*, which determines the penalty for errors greater than ε. In addition, for the radial basis kernel used in this study, the width γ of the radial basis kernel must be selected.

For each cluster, parameter selection is performed using grid search with *k*-fold cross validation. Parameter combinations form a grid, and *k*-fold cross validation is repeated for each grid element to assess prediction error. The parameter combination with the smallest error is selected. Next, for each cluster, the SVR model is built using all data from that cluster. Note that parameter optimization and SVM model training are performed separately for each cluster and using only the training data set.

After step A.6, clusters and their corresponding SVR models are ready for use in prediction.

2) Testing

The steps of the testing or prediction phase correspond to the training steps. Test data are normalized in step B.2 using statistics from training step A.2. Next, the weights calculated during training in step A.3 using the correlation approach are applied to test data. Each value for each input feature in the testing set is weighted using the correlation coefficient calculated according to Eq. (6) for that feature.

Next, cluster membership (step B.4) is determined by finding the nearest cluster mean in terms of Euclidean distance. The distance of the data point *x* to cluster *s* is:

$$Dist_s = \sqrt{\sum_{i=1}^{M}(x_i - \mu_i)^2} \tag{7}$$

where *M* is the number of independent variables and μ is the mean of cluster *s*.

The SVR model for the nearest cluster might not be using all features, and therefore features not used in that cluster SVR are removed. Note that feature removal depends only on what was determined during the training stage and is not affected by the feature values of the test data.

Finally, the SVR model corresponding to the identified cluster is used for prediction (step B.6).

C. Performance Metrics

The two metrics often used in electricity prediction studies are the mean absolute percentage of error (MAPE) and the coefficient of variance (CV) [3, 9, 18]; hence, this work also uses these metrics.

The MAPE metric expresses average absolute error and is calculated as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100, \tag{8}$$

where y_i is the actual consumption, $\hat{y}_i$ is the predicted consumption, and *N* is the number of observations.

The CV metric expresses error variation with respect to the mean and is calculated as follows:

$$CV = \frac{\sqrt{\frac{1}{N-1}\sum_{i-1}^{N}(y_i - \hat{y}_i)^2}}{\bar{y}} \times 100, \tag{9}$$

where y_i, $\hat{y}_i$, and *N* represent the same elements as in MAPE and $\bar{y}$ is the average actual consumption.

V. Evaluation

This section first introduces the data set. Next, results are presented and discussed and threats to validity described.

A. Implementation

The evaluation was carried out on data from Budweiser Gardens, an event venue with a capacity of over 10,000 seats located in London, Ontario, Canada. This venue hosts professional sport events, including basketball and hockey, and a variety of other entertainment shows such as concerts and theatre productions.

Electricity consumption data were obtained through Green Button (GB) Connect My Data. London Hydro, the local electricity utility, has developed the first cloud-based Green Button Connect My Data environment to provide data access to academic partners with the customer's consent. The data consisted of 15-minute electricity consumption readings from revenue grade utility meters from January 1, 2013 to March 31, 2014.

This period generated a total of 43,680 data points. Although this is not a very large data set in a Big Data context, it can result in significant computation requirements, especially when parameter selection is involved. For example, in previous work by the authors [9], traditional SVR for the same data set with five-fold cross validation for parameter selection took over 24 hours. This was for only 10 values for each of two prediction model parameters and on a two node cluster, with each node having 24 cores and 96 GB memory. Consequently, even for a data set of this size, computation time needs to be reduced.

In addition to energy consumption, the data set included event-related data as described in Section IV.A. 80% of the data were used for parameter selection and training, and 20% were used for testing. The training set contained readings for all of 2013, thus accounting for all seasons. The testing set included data for the first 3 months of 2014.

Three prediction approaches were implemented: SVR, local SVR as presented in Section IV.B, and H2O deep learning. Each implementation uses the grid search approach with five-fold cross validation for the parameter selection:

- **SVR**: Implemented in R language [31] using the "e1071" package. Two parameters were tuned: 10 values for the cost C from 1e-6 to 1e+3 with exponential increments and

10 values for the radial basis parameter γ from 1e-8 to 1e+1. This makes for a total of 100 configurations.

- **Local SVR**: Implemented in R language [31] using the "stats" package for *k*-means clustering and the "e1071" package for SVR. The SVR model for each cluster was tuned using the same approach as in standalone SVR; 10 values for the cost *C* and 10 values for the parameter γ. In addition, ten values of the number of clusters *k* (from 20 to 110 by increments of 10) were considered.
- **H2O deep learning**: An H2O implementation of distributed deep neural networks was used. It was accessed from R through the "h2o" package. H2O deep learning has a large number of parameters, including number of hidden layers, number of neurons in each layer, adaptive learning rate ε, adaptive learning rate time decay ρ, and regularization parameters l1 and l2. To keep the grid search size reasonable, only the number of layers, number of neurons, ε, and ρ were considered, as presented in Table I. For other parameters default values were used. This made for 81 considered configurations, which was fewer than in SVR or local SVR, but was kept low to keep training time reasonable.

Experiments were carried out on a two node cluster (Gigabit Ethernet); each node had 24 cores (Intel Xeon CPUs) and 96 GB RAM. For SVR and local SVR, the code was parallelized to run different configurations on different cores and nodes. H2O itself performs distributed computations, and hence no additional parallelization was implemented.

B. *Results and Discussion*

The prediction approaches (SVR, local SVR, and H2O deep learning) were evaluated with hourly and with 15-min readings. Two aspects of the prediction were evaluated: accuracy and training time. In the case of machine learning with Big Data, a small drop in accuracy can be warranted by a large reduction in training time.

MAPE and CV consumption prediction errors obtained with each approach on testing data are presented in Table II. The same data are displayed in Fig. 2. For hourly readings, local SVR achieved slightly lower error rates in terms of both MAPE and CV errors than traditional SVR, with MAPE errors of 16.806 and 17.860 and CV errors of 19.612 and 20.428 for local SVR and SVR respectively. H2O accuracy was lower, with an MAPE error of 20.261 and a CV error of 22.703.

With 15-min data, traditional SVR and local SVR also outperformed H2O, with the lowest error rates obtained with local SVR (MAPE error of 19.407 and CV error of 21.517).

For all three approaches, accuracy with 15-min readings was lower than accuracy with hourly readings. This can be explained by the models inability to capture random consumption variations between 15-min intervals.

TABLE I. H2O DEEP LEARNING PARAMETERS

Parameters		Considered values
Hidden layer sizes	1-layer	(16), (32), (64)
	2-layer	(16,16), (32,32), (64,64)
	3-layer	(16,16,16), (32,32,32), (64,64,64)
ρ		0.95, 0.99, 0.999
ε		1e-10, 1e-8, 1e-6

As already mentioned, it is crucial to consider training time in addition to prediction accuracy. For both hourly and 15-min intervals, the same time periods were considered; therefore, the 15-min data set was four times the size of the hourly data set. Fig. 3 compares the training times for the three approaches for hourly and 15-min readings. The training time included parameter optimization using grid search with five-fold cross validation. For SVR, two parameters with 10 values each were considered (as described in Section V.A), which made for a total of 100 configurations. For local SVR, exactly the same configurations were considered, with the difference that optimization was performed at the cluster level. Moreover, 10 values for the number of clusters were considered (from 20 to 110 by increments of 10). Finally, for H2O, 81 configurations were considered (as described in section V.A) to keep the training time reasonable. This grid parameter optimization with five-fold cross validation was a large contributing factor to overall training time.

As seen from Fig. 3, the time to train the local SVR was several times shorter than to train the SVR or H2O models. The difference between training time for local SVR and the

TABLE II. ERRORS : SVR, LOCAL SVR, AND H2O DEEP LEARNING

	Hourly readings		15-min readings	
	MAPE	***CV***	***MAPE***	***CV***
SVR	17.860	20.428	19.973	21.964
Local SVR	16.806	19.612	19.407	21.517
H2O deep learning	20.261	22.703	21.329	22.151

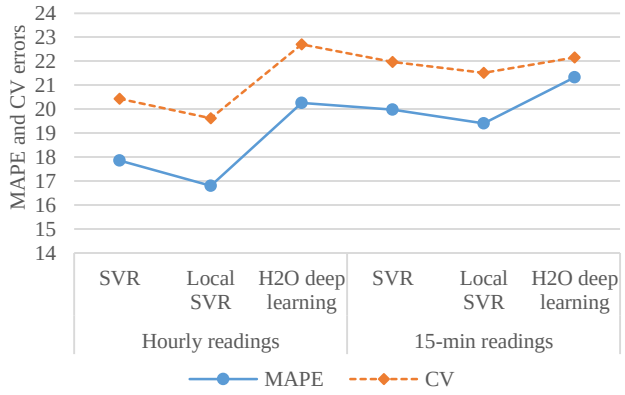

Figure 2. Prediction accuracy: SVR, local SVR, and H2O deep learning

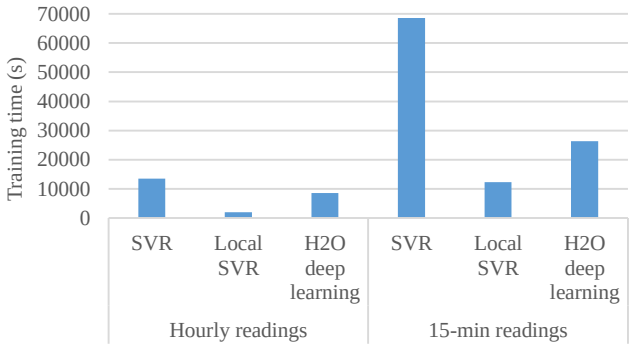

Figure 3. Training times: SVR, local SVR, and H2O deep learning

other two models remained significant for 15-min readings. Local SVR training time was reduced compared to SVR because instead of training one model with a large data set, several cluster models were trained with smaller data sets.

Fig. 4 and 5 show training time, MAPE errors, and CV errors versus the number of clusters, for hourly and 15-min readings respectively. For parameter optimization, 10 values of the number of clusters were considered, but here the value domain was extended to consider larger numbers of clusters. For both hourly and 15-min readings, training time decreased sharply when the number of clusters was increased from 20 to 60. When the number of clusters was increased beyond 120, no significant further change in training time occurred. In contrast, as the number of clusters increased, the MAPE and CV errors gradually increased. 50 clusters gave error rates close to minimums with reasonably short training time.

Another reason for short training time with local SVR is feature reduction; therefore, the relation between training time and number of removed features was explored. Fig. 6 and 7 show the average number of removed features per cluster and the training error for hourly and 15-min readings respectively. With an increasing number of clusters, the average number of removed features increases, and the training time decreases. Considering that there were only eight input features in this case study, on average, more than half features were removed.

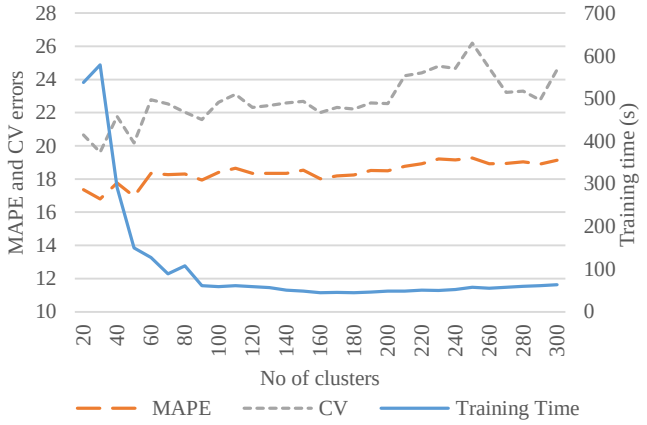

Figure 4. Local SVR, hourly data: errors and training time

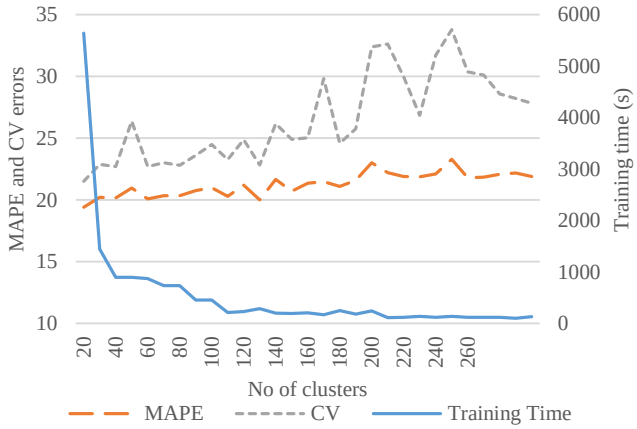

Figure 5. Local SVR, 15-min data: errors and training time

C. *Threats to validity*

H2O deep learning has a large number of parameters that can be tuned in an attempt to increase accuracy. The authors believe that by including other parameters, especially regularizations l1 and l2, accuracy could be improved. However, further parameters were not considered in an attempt to keep the grid search at a similar size to the other two models and the training time reasonably short.

Similarly, deep learning in general is successful with complex problems, and for energy prediction, its power might be excessive. In these experiments, the accuracy among different H2O runs varied greatly, which can be explained by getting stuck in local minima and by the use of the Hogwild! approach.

Nevertheless, the experiments performed in this research still demonstrate that the local SVR approach presented in this paper outperforms traditional SVR in terms of accuracy and training time. Moreover, local SVR is easier and less resource intensive to tune than H2O deep learning.

I. CONCLUSIONS

In recent years, development and proliferation of sensors and metering devices have enabled collection of fine-grained

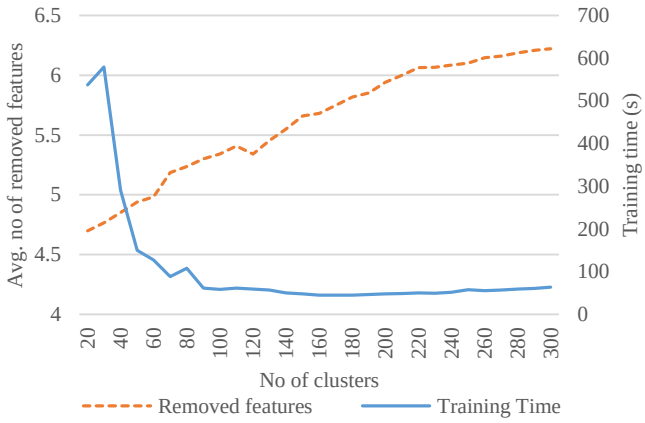

Figure 6. Local SVR, hourly readings: training time and average number of removed features

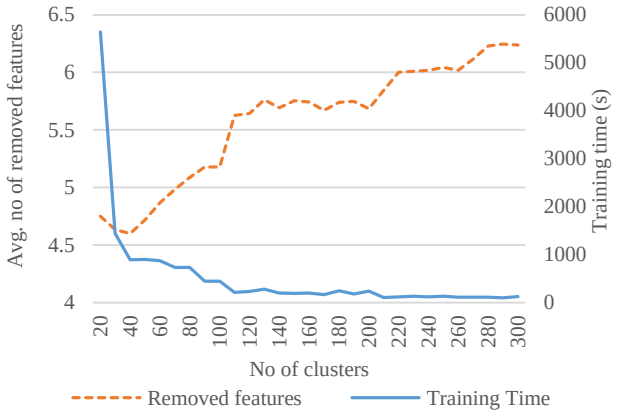

Figure 7. Local SVR, 15-min readings: training time and average number of removed features

energy consumption data. These sensor Big Data have the potential to improve energy prediction greatly, but they also pose major challenges.

This paper explores the ability of recently developed Big Data approaches with respect to energy consumption prediction. The focus is on evaluating if and how findings from machine learning with Big Data can benefit consumption prediction. An approach based on local learning with support vector regression is presented. The approach takes advantage of parameter reduction to increase training speed. The presented case study compares traditional SVR, local SVR, and H2O deep learning in terms of accuracy and training time. Local SVR outperformed H2O in both accuracy and training time. The presented local SVR was evaluated on the energy consumption prediction for event venues, but it could be applied for other energy consumption prediction scenarios.

Future work will evaluate the same approaches on much larger data sets to determine their performance on truly Big Data. Comparison with other distributed Big Data algorithms such as those supported by Spark will be performed. To partition the data, locality-sensitive hashing (LSH) will be evaluated as a potential replacement for k-means.

ACKNOWLEDGMENT

This research has been partially supported by NSERC CRD at Western University (CRD 477530-14). The authors would like to thank London Hydro for supplying industry knowledge, the Green Button platform, and the data used in this study. They also would like to thank Budweiser Gardens for providing valuable data for this project.

REFERENCES

[1] U.S. Energy information administration, "Electricity Data." http://www.eia.gov/electricity/data/browser/2015.

[2] United States environmental protection agency, "Sources of greenhouse gas emissions." http://www3.epa.gov/climatechange/ghgemissions/sources.html2013.

[3] H. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

[4] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. M. Capretz, "Challenges for MapReduce in Big Data," *Proceedings of the IEEE World Congress on Services*, pp. 182–189, 2014.

[5] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, 2013.

[6] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidatc, G. K. Karagiannidis, and K. Tahaa, "Efficient machine learning for Big Data: A review," *Big Data Research*, vol. 2, no. 3, pp. 87–93, 2015.

[7] "Apache Mahout." http://mahout.apache.org/.

[8] "Apache Spark." http://spark.apache.org/.

[9] K. Grolinger, A. L'Heureux, M. A. M. Capretz, and L. Seewald, "Energy forecasting for event venues: Big Data and prediction accuracy," *Energy and Buildings*, vol. 112, pp. 222–233, 2016.

[10] Oxdata, "H2O." http://www.h2o.ai/2016.

[11] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data, New York: Springer-Verlag*. 1982.

[12] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Computation*, vol. 4, no. 6, pp. 888–900, 1992.

[13] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research*, pp. 363–392, 2005.

[14] F. Niu, B. Recht, R. Christopher, and S. J. Wright, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," *Advances in Neural Information Processing Systems*, pp. 693–701, 2011.

[15] H. Zhang, G. Chen, and B. Ooi, "In-memory Big Data management and processing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920–1948, 2015.

[16] X.-W. Chen and X. Lin, "Big Data deep learning: challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[17] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[18] R. K. Jain, K. M. Smith, P. J. Culligan, and J. E. Taylor, "Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy," *Applied Energy*, vol. 123, pp. 168–178, 2014.

[19] L. Suganthi and A. Samuel, "Energy Models for Demand Forecasting-A Review," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1223–1240, 2012.

[20] A. S. Ahmad, M. Y. Hassan, M. P. Abdullah, H. A. Rahman, F. Hussin, H. Abdullah, and R. Saidur, "A review on applications of ANN and SVM for building electrical energy consumption forecasting," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102–109, 2014.

[21] H. C. Jung, J. S. Kim, and H. Heo, "Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach," *Energy and Buildings*, vol. 90, pp. 76–84, 2015.

[22] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man. and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 4, pp. 438–447, 2010.

[23] R. Ž. Jovanović, A. A. Sretenović, and B. D. Živković, "Ensemble of various neural networks for prediction of heating energy consumption," *Energy and Buildings*, vol. 94, pp. 189–199, 2015.

[24] S. S. Abdelkader, K. Grolinger, and M. A. M. Capretz, "Predicting energy demand peak using M5 model trees," *Proceedings of the IEEE International Conference on Machine Learning and Applications*, 2015.

[25] M. Frincu, C. Chelmis, M. U. Noor, and V. Prasanna, "Accurate and efficient selection of the best consumption prediction method in smart grids," *Proceedings of the 2014 IEEE International Congress on Big Data*, pp. 721–729, 2014.

[26] D. Anjos, P. Carreira, and A. P. Francisco, "Real-time integration of building energy data," *Proceedings of the 2014 IEEE International Congress on Big Data*, pp. 250–257, 2014.

[27] G. Kejela, R. M. Esteves, and C. Rong, "Predictive analytics of sensor data using distributed machine learning techniques," *Proceedings of the 6the IEEE International Conference on Cloud Computing Technology and Science*, pp. 626–631, 2014.

[28] US Department of Energy, "Green Button." http://energy.gov/data/green-button.

[29] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[30] C. A. Sugar and G. M. James, "Finding the number of clusters in a dataset," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, 2003.

[31] R Core Team, "R: A language and environment for statistical computing." R Foundation for Statistical Computing, Vienna, Austria, http://www.r-project.org, 2015.