Industrial Engineering
& Management Systems
Vol 14, No 3, September 2015, pp.318-324
ISSN 1598-7248 │ EISSN 2234-6473 │

http://dx.doi.org/10.7232/iems.2015.14.3.318
© 2015 KIIE

Short-Term Load Forecasting Based on Sequential Relevance Vector Machine

Youngchan Jang*
Weapon System Engineering Department, Korea Army Academy at Yeongcheon, Yeongcheon, Korea

(Received: May 13, 2015 / Accepted: September 14, 2015)

ABSTRACT

This paper proposes a dynamic short-term load forecasting method that utilizes a new sequential learning algorithm based on Relevance Vector Machine (RVM). The method performs general optimization of weights and hyperparameters using the current relevance vectors and newly arriving data. By doing so, the proposed algorithm is trained with the most recent data. Consequently, it extends the RVM algorithm to real-time and nonstationary learning processes. The results of application of the proposed algorithm to prediction of electrical loads indicate that its accuracy is comparable to that of existing nonparametric learning algorithms. Further, the proposed model reduces computational complexity.

Keywords: Sequential Relevance Vector Machine, Support Vector Machine, Online Learning, Short-Term Load Forecast, Prediction

* Corresponding Author, E-mail: mapsossa@gmail.com

1. INTRODUCTION

Load forecasting plays an important role in electric power systems. Accurate forecasting of information is critical for both the electricity supplier and end-users. For the electricity supplier, owing to the competitive nature of energy markets, accurate estimation of electrical loads is vital for many operational decisions, such as pricing, generating capacity, scheduling of operations, and operating cost (Fan and Chen, 2006). For end-users, accurate load forecasting provides the information needed to develop maintenance plans, allocate reserves, manage loads, and exchange redundant load with other facilities.

Depending on a time period, load forecasting is divided into long term, medium term, and short term. Long term load forecasting covers period of one to several years and its goal is for planning of utility investment such as whether to build or upgrade new electric utilities. Medium term load forecasting is related to time period from few months to a year and it is used for power system planning, scheduling for generation companies for load peak time like summer and winter seasons. Short term load forecasting is made for a period from one hour to few days. And it plays an important role in real time generation control, energy management optimization, and energy transaction planning.

A variety of forecasting tools based on various techniques, such as time-series and artificial intelligence methods, have been utilized over the years especially for short term load forecasting. Time-series models such as moving average, exponential smoothing methods, autoregressive and moving averages models with exogenous input model (ARMAX) (Taylor and McSharry, 2007), and Kalman filter-based methods (Al-Hamadi and Soliman, 2004) are employed for short-term load forecasting. However, these time-series models have linearity assumptions; consequently, their prediction accuracy may not be satisfactory when there are nonlinear relationships among the electrical loads, previous loads, and weather conditions. As a result, to improve the prediction power, nonlinear models have been proposed. Neural Network (NN) (Benaouda *et al*., 2006; Rocha *et al*., 2005; Zhang *et al*., 2001; Pandey *et al*., 2010; Bashir *et al*., 2009; Amjady *et al*., 2009) based methods have been applied and shown to effectively learn the time dependent load series and capture the nonlinearity characteristics. Recently, Support Vector Machine (SVM), a machine learning technique, has also been used for load forecasting (Dongxiao *et al.*, 2009; Che, 2012; Ping-Feng and Wei-Chiang, 2005). SVM is used extensively for classification and regression problems because of its high generalization performance. But SVM only provides a point forecasting value; it does not provide the estimate in a probabilistic sense. In addition, k-fold cross-validation is usually employed to optimize its parameters (Hsu *et al.*, 2003), which is computationally expensive. To overcome these disadvantages of SVM, Tipping (2001) developed a new machine learning technique, called the Relevance Vector Machine (RVM). RVM is a nonparametric model with most of its characteristics similar to those of SVM. However, RVM has a sparseness characteristic, i.e. it uses a subset of training points called Relevance Vectors (RVs) to predict the output given the new input data. Usually, the number of RVs is much smaller than that of SVs. Thus, RVM is more advantageous in terms of memory efficiency. Further, the accuracy of RVM is comparable to that of SVM.

A small number of studies utilize RVM for short term load forecasting (Zhinong *et al.*, 2013, Qing *et al.*, 2008). And almost all of these studies are designed for handling batch data. It means that when electricity usage datasets increase dynamically in an hour, the original RVM operates in a batch manner by discarding the previous result of forecasts and relearning with all the arrived data. In this case, it takes a lot of time to train and forecast the next hour electrical load. Therefore, the batch RVM needs to be modified to adapt itself to dynamic cases.

Previous study (Nikolaev *et al.*, 2005) proposed the Sequential RVM, which organizes adaptive model selection through simultaneous incremental optimization of both the weight parameters and their hyperparameters. But it does not preserve the sparseness property of RVM itself. To preserve the advantage of RVM, we propose a new dynamic learning algorithm based on RVM without losing the sparseness property. Our proposed algorithm updates the model sequentially when new observations are obtained. We call the proposed approach Sequential RVM (SRVM).

Electricity usage data results for our implementation show that the proposed sequential model approximates the original model without losing the sparseness property of the original RVM. The results also indicate that the proposed model has improved prediction accuracy over other nonparametric regression algorithms and the original RVM in batch manner. Moreover, the proposed model reduces computational complexity.

The remainder of this paper is organized as follows. Section 2 reviews the original RVM and explains the proposed algorithm. Section 3 discusses a numerical example used to evaluate the performance of the proposed Sequential RVM using electrical load data. Finally, we summarize and conclude this paper in Section 4.

2. METHODOLOGY

In this section, we first review the original RVM proposed by Tipping (2001). We then extend it for online learning, leading to our proposed sequential algorithm.

2.1 Original RVM

Given a set of input vectors and corresponding target values, $\{x_i, y_i\}_{i=1}^N$, it is assumed that

$$t_i = y(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2) \tag{1}$$

Function $y(x_i)$ is defined as a combination of design matrix $\Phi \in R^{N\times(N+1)}$ and its corresponding weight vector, $w \in R^{N+1}$:

$$y(x_i) = \sum_{i=0}^{N} w_i \phi(x_i) = \Phi w \tag{2}$$

$w = (w_0, w_1, \cdots, w_N)^T$, $\Phi = [\phi(x_1), \phi(x_2), \cdots, \phi(x_N)]^T$
with $\phi(x_i) = [1, K(x_i, x_1), \cdots, K(x_i, x_N)]$.
The Gaussian kernel, or Radial Basis Function (RBF), is widely utilized for the kernel function:

$$K(x_m, x_n) = \exp\{-l^2 \| x_m - x_n \|^2\} \tag{3}$$

where l is the Gaussian kernel width.

A target value t_i follows a Gaussian distribution with mean $y(x_i)$ and covariance σ^2. Therefore, the likelihood of the complete data set is

$$p(t \mid w, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\{-\frac{1}{2}\sigma^2 \| t - \Phi w \|^2\} \tag{4}$$

Since the maximum likelihood estimation of w and σ^2 from the likelihood of the complete dataset could lead to a severe over-fitting problem, RVM imposes constraints on parameter w by defining a prior probability distribution over it:

$$p(w \mid \alpha) = \sum_{i=0}^{N} N(w_i \mid 0, \alpha_i^{-1}) \tag{5}$$

where $\alpha \in R^{N+1}$ is a hyperparameter whose distribution is

$$p(\alpha) = \prod_{i=0}^{N} Gamma(\alpha_i \mid a, b) \tag{6}$$

$$p(\sigma^{-2}) = \prod_{i=0}^{N} Gamma(\sigma^{-2} \mid c, d) \tag{7}$$

To make non-informative prior distributions, the parameters are set to zero: $a = b = c = d = 0$.

The weight prior, $p(w_i) = \int p(w_i \mid \alpha_i) p(\alpha_i) d\alpha_i$, corresponds to the density of a Student-t distribution, which causes the weight vector to be sparse, because weight components are sharply peaked at zero. The posterior distribution of the weight is represented by

$$p(w \mid t, \alpha, \sigma^2) = \frac{p(t \mid w, \sigma^2) p(w \mid \alpha)}{p(t \mid \alpha, \sigma^2)}$$
$$= (2\pi)^{-\frac{(N+1)}{2}} \mid \Sigma \mid^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2(w-\mu)^T \Sigma^{-1} (w-\mu)} \right\} \quad (8)$$

where the posterior weight covariance and mean are represented by

$$\Sigma = (\alpha^{-2} \Phi^T \Phi + A)^{-1} \quad (9)$$
$$\mu = \sigma^{-2} \Sigma \Phi^T t \quad (10)$$

with hyperparameter matrix $A = diag(\alpha_0, \alpha_1, \cdots, \alpha_N)$.

RVM searches for the maximization of hyperparameters, which means that $p(\alpha, \sigma^2 \mid t) \propto p(t \mid w, \sigma^2) p(\alpha)$ $p(\sigma^2)$ with respect to α and σ^2. With uniform hyperpriors, it is required to maximize

$$p(t \mid \alpha, \sigma^2) = \int p(t \mid w, \sigma^2) p(w \mid \alpha) dw$$
$$= (2\pi)^{-\frac{1}{2}} \mid \sigma^2 I + \Phi A^{-1} \Phi \mid^2 \exp\left\{ -\frac{1}{2} t^T (\sigma^2 I + \Phi A^{-1} \Phi)^{-1} t \right\} \quad (11)$$

The likelihood is marginal, with maximization called type-Ⅱ maximization. Because maximization of α and σ^2 cannot be estimated in a closed form, iterative re-estimation is used to compute the maximization of the hyperparameters. Therefore, differentiation of the marginal likelihood with respect to α and σ^2, respectively, is utilized in the re-estimation.
For the diagonal matrix α,

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} = \frac{(1 - \alpha_i \Sigma_{ii})}{\mu_i^2} \quad (12)$$

where Σ_{ii} is the diagonal element of the posterior weight covariance with current α and σ^2 values.
For the noise variance σ^2,

$$(\sigma^2)^{new} = \frac{\| t - \Phi\mu \|^2}{N - \Sigma_{ii} \gamma_i} \quad (13)$$

The learning algorithm is repeated until some convergence criteria are satisfied; then, the posterior weight mean and ovariance are updated with current α_i^{new} and $(\sigma^2)^{new}$. During the iterations, a large number of α_i s tend to infinity, which means that the corresponding posterior weight components become zero; i.e. the posterior distributions of many weights, $p(w \mid t, \alpha, \sigma^2)$, are highly peaked at zero. The surviving components and their corresponding data points are called RVs. The number of RVs are, in practice, less than that of SVs in SVM, which leads to the prediction time in RVM being shorter than that in SVM; hence, we believe that RVM is more memory efficient than SVM.

The predictive distribution for new data, x_*, can be computed based on the posterior distribution of the weight, with maximum values α_{MP} and σ_{MP}^2 :

$$p(t_* \mid t, \alpha_{MP}, \sigma_{MP}^2) = \int p(t_* \mid w, \sigma_{MP}^2) p(w \mid t, \alpha_{MP}, \sigma_{MP}^2) dw$$
$$= N(t_* \mid y_*, \sigma_*^2) \quad (14)$$

where $y_* = \mu^T \phi(x_*)$ and $\sigma_*^2 = \sigma_{MP}^2 + \phi(x_*)^T \Sigma \phi(x_*)$

For the computation of the inverse matrix in posterior weight covariance, Cholesky decomposition is used for numerical stability with $O(n^3)$. In the first iteration, n is equivalent to the amount of data, N. During subsequent iterations, n corresponds to the number of RVs. Therefore, learning time is improved during the iterations. However, sufficient time to compute matrix inversion is required in the first iteration. Furthermore, if the dimension of the covariance weight is large, the degenerate problem, in which some columns of the covariance weight can be represented by the combinations of the other columns, arises, and results in the algorithm being inoperable. Therefore, it is recommended that RVM be used only with a small amount of data.

2.2 Proposed Algorithm

In this section, we introduce our proposed algorithm, Sequential RVM. Sequential RVM trains and updates the model sequentially whenever new data are observed. The idea underlying Sequential RVM is that there might be a relationship between the RVs obtained previously and those obtained after new data arrive. Therefore, the proposed algorithm trains the model with the set of RVs and newly arrived data, which means that the RVs obtained in the previous stage are used in the next learning stage with the new data.

On observing new data, first a new kernel function is defined using the current kernel function and the new data. Next, the observed data are added to the current set of RVs and arbitrary values γ_i and α_i placed in the current sets of γ and α matrices. Iteration then begins with the newly arrived data and a set of RVs obtained prior to the new data being observed. A more detailed explanation of the algorithm is given in Table 1.

Usually the values for α_i increase during the iterations, and when α_i tends to infinity (a certain preset large number), the corresponding columns of the covariance of the weight are deleted. As stated in Section 2.1, the surviving columns are called RVs. In the proposed algorithm, α_i s are re-estimated from the α_i s computed in the previous iteration. Therefore, in practice, the RVs that arrive first are deleted first. Consequently, the algorithm always has RVs consisting of the most current data; i.e. the learning algorithm is more weighted by the

Table 1. Outline of the proposed algorithm

```
Initialize $\alpha$ and $\sigma^2$ : $\alpha \leftarrow 0.1$, $\sigma^2 \leftarrow 0.1 \times \mathrm{var}(t)$
if  the number of the arrived data = 1
    1. Kernel matrix, $\Phi = K(X, X)$, given current data $\{X, y\}$
      for  iteration = 1 to maximum iteration or convergence
            2. Covariance of weights, $\Sigma = (\alpha^{-2}\Phi(:, used)^T \Phi(:, used) + A(used))^{-1}$
            3. Mean of weights, $\mu = \sigma^{-2}\Sigma\Phi(:, used)^T t$
            4. Hyper-parameter, $\alpha_i^{new} = \frac{(1-\alpha_i\Sigma_{ii})}{\mu_i^2}$ and Noise variance, $(\sigma^2)^{new} = \frac{\|t-\Phi\mu\|^2}{N-\Sigma_{ii}\gamma_i}$
            5. Delete non-zero
      end for
else (whenever the new data arrives)
    6. Insert the new data point $\{x_{new}, y_{new}\}$ to the current data, $X_{new} = [X; x_{new}]$ and $y_{new} = [y, y_{new}]$
    7. Update the kernel matrix, $\phi_{new} = [K(X, X)\ K(X, x_{new});\ K(X, x_{new})\ K(x_{new}, x_{new})]$
    8. $used_{new} = [used;\ 1]$, previous used is equivalent to the indices of previous RVs
    9. $\gamma_{new} = [\gamma;\ 1]$
    10. $\alpha_{new} = [\alpha_i;\ initial\ alpha]$
    11. By using $\Phi_{new}(:, used_{new})$, updates $\Sigma$, $\mu$, $\alpha$, and $\sigma^2$ (i.e. implement 2-5)
end if
```

$used \in (0,\ 1)$ is the indicator vector. "1" represents survival column, and "0" deleted column.

$A(used)$ represents the diagonal matrix.

current data.

Like the original RVM, the computational complexity of Sequential RVM is $O(n^3)$. However, n in the original RVM is equivalent to the amount of data, whereas n in the Sequential RVM is equivalent to the number of RVs.

Therefore, the training time of Sequential RVM is substantially improved in an online manner compared with the original RVM in a batch manner. Further, because the number of RVs in sequential RVM is small compared with the number of RVs in the original RVM, the degenerate covariance problem does not arise. Thus, the algorithm requires a smaller amount of time to train the model and is also able to train large amounts of data.

3. NUMERICAL EXAMPLE

In this section, we validate the efficacy of the proposed algorithm by applying it to real-world data.

3.1 Data Description and Implementation

In this paper, ISO New England the hourly electric load data is used; for which one year's worth of electric load data are available. Figure 1 depicts the electricity usage data. It is clear that the electricity usage has daily, weekly, and seasonal patterns.

Previous studies have shown that the load patterns on weekdays and weekends, including holidays, are quite different (Song *et al.*, 2005). Therefore, we train two models separately, one for weekday load prediction and the other for predictions on weekends and holidays, to achieve accurate load forecasting. For the input features, the list of input variables in Table 2 is used for each model.

We standardize each variable, such that each variable can range from zero to one:

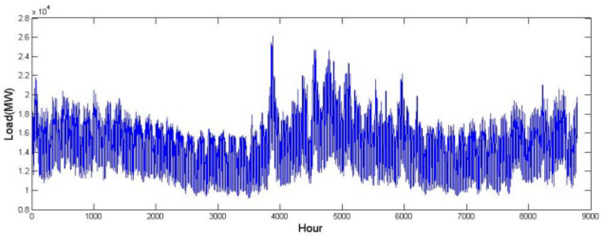

Figure 1. Hourly electricity usage in 2008.

Table 2. List of input data

Index	Dry bulb/dew point temperature
	Hour of day, Day of the week
	Holiday/weekend indicator (0 or 1)
	Previous 24-hr average load
	1, 2, 3, 4, 5, 6, 7, 12, 24, 168-hr lagged load

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{15}$$

Both batch and online learning algorithms utilize the data arrived so far to forecast the next hour load iteratively, all the data sets are both training and test data.

To validate the performance of our proposed algorithm, we computed several criteria; specifically, Mean Square Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE)

$$MSE = \frac{1}{n}\sum_i (y_i - f_i)^2 \tag{16}$$

$$MAE = \frac{1}{n}\sum_i | y_i - f_i | \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_i (y_i - f_i)^2} \tag{18}$$

where y_i is the real hourly electricity load and f_i is the predicted hourly electricity load by the forecasting tools.

To compare the efficiency of the model with other existing algorithms, we measured the training time required. We used the validation data to conduct cross-validation and to determine the optimal Gaussian kernel width. With the optimal Gaussian kernel width obtained using the validation data, the models were then trained with the training dataset and electricity load predicted one hour ahead with the test dataset.

3.2 Results

We compared the prediction performance of our proposed approach with that of ε -SVR, NN, and RVM by batch manner. The prediction error, learning time which contains the prediction time for each hourly forecast, and the number of RVs/SVs are shown in Table 3 and 4. Table 3 shows the results for weekdays (regular days), whereas Table 4 shows the results for weekends including holidays. The proposed model clearly outperforms SVM, NN, and RVM in terms of the accuracies on both weekdays and weekends. Especially, the accuracy of Sequential RVM on weekends is clearly better than that on weekdays.

Sequential RVM learns the model and predicts the subsequent load one hour ahead whenever new data are observed. The type of learning in Sequential RVM is quite different from that of the other methods we implemented for comparison; specially, ε -SVM, NN, and RVM are batch learning algorithms. The main difference between the batch and online learnings depends on whether the algorithms utilize the prediction results in the previous learning stage in the next learning stage. The batch learning algorithms ignore the results of the previous learning results in the next learning processes. Thus, the data comes, the learning time increases and is accumulated. However, Sequential RVM utilizes RVs computed in the previous learning stage and newly arrived data instead of the whole data set. It saves the learning time to train and forecast the load.

Table 3. Experimental results of batch learning algorithms for weekdays

	ε -SVM	NN	RVM	SRVM
MSE	135,685	132,376	128,565	128,053
MAE	261.1	253.6	247.8	249.3
RMSE	368.4	363.9	358.6	357.9
Number of RVs/SVs	3,021	-	45	(12.3)
Learning time(sec)	1,499.2	1,488.1	1,306.3	114.8

Table 4. Experimental results of batch learning algorithms for weekends

	ε -SVM	NN	RVM	SRVM
MSE	80,806	75,521	85,825	69,951
MAE	209.7	202.6	220.1	192.1
RMSE	284.3	274.8	293.0	264.5
Number of RVs/SVs	1,528	-	39	(11.1)
Learning time(sec)	135.6	225.6	284.7	25.6

Numbers in parenthesis refer to the average number of RVs.

As stated in Section 2.2, the proposed algorithm adds the new observation to the set of previous RVs. The relationship between the previously obtained RVs and the new RVs can be determined after new data are observed from Figure 2 and Figure 3 and Table 5 and Table 6. The original RVM does not have any relationship among the indices of RVs when the amount of data increases, whereas Sequential RVM preserves or deletes the previous RVs during the learning process. Moreover, it can be seen that the more recent data are included in the set of RVs in Sequential RVM. Thus, it is clear that our proposed algorithm utilizes the more recent data in its learning processes.

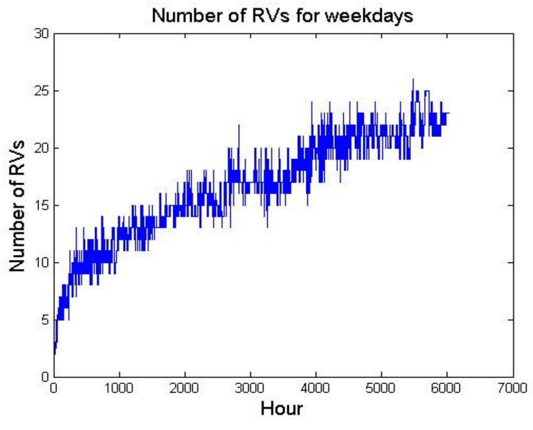

Figure 2. Number of RVs at each learning process for weekdays.

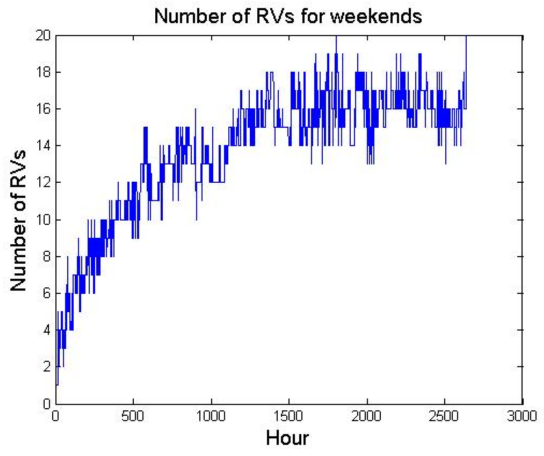

Figure 3. Number of RVs at each learning process for weekends.

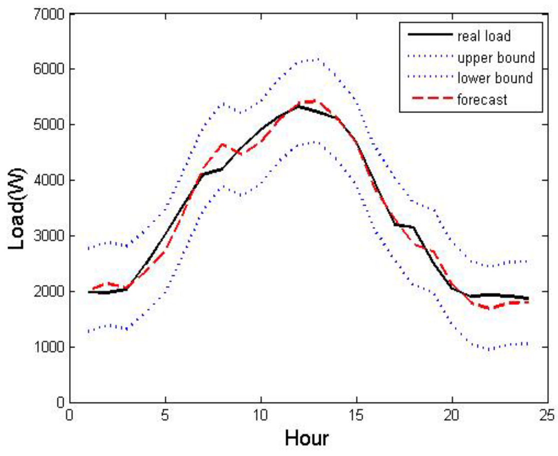

Figure 4. 95% confidence interval of the load forecast.

Table 5. Indices of RVs with different number of input data with original RVM

N	Number of RVs	Indices of RVs
500	9	100, 107, 210, 228, 262, 359, 380, 412, 497
600	10	77, 100, 228, 248, 262, 291, 380, 497, 506, 507
700	12	100, 176, 220, 248, 262, 267, 290, 291, 380, 497, 507, 563
800	14	44, 107, 219, 220, 221, 380, 399, 490, 491, 497, 507, 554, 673, 720
900	13	98, 129, 262, 267, 397, 490, 491, 497, 597, 640, 673, 871, 872
1,000	12	104, 129, 262, 397, 490, 491, 497, 597, 640, 673, 872, 931

Table 6. Indices of RVs with different number of input data with Sequential RVM

N	Number of RVs	Indices of RVs
500	11	88, 184, 227, 290, 356, 380, 434, 455, 458, 492, 499
600	10	88, 184, 227, 290, 356, 434, 455, 458, 550, 595
700	13	88, 184, 227, 290, 356, 434, 455, 458, 595, 673, 689, 696, 699
800	11	184, 227, 290, 356, 673, 716, 720, 725, 739, 750, 797
900	12	184, 227, 290, 356, 673, 716, 720, 739, 750, 797, 825, 900
1,000	12	184, 227, 290, 356, 673, 716, 720, 739, 797, 825, 901, 910

Another advantage of an RVM-based algorithm over SVM is that it provides probabilistic inference, including the prediction intervals for one-hour ahead predictions. By using the probabilistic information, electricity reserves can be prepared upfront for emergency situations such as peak electricity usage.

4. CONCLUSION

This paper proposed an RVM-based online learning algorithm in which the posterior weight, mean, and covariance hyperparameters, and the noise variance are updated from the previous learning process by utilizing the most recent RVs. Because the proposed method uses recent data to update the prediction model, it can estimate loads more flexibly in nonstationary data scenarios. The results of a comparative evaluation using real-world electricity usage data indicate that the proposed method is superior to ε-SVM, original RVM, and NN in batch learning in terms of the accuracy.

The approach used extends the batch learning algorithm to online learning based on RVM using the most recent data to update the prediction model. The proposed algorithm is weighted by more recent data in the learning process. Moreover, it preserves the advantages of the original RVM over SVM and overcomes the disadvantages of the original RVM. However, the accuracy of the proposed algorithm depends on the proper choice of the Gaussian kernel width. Cross-validation or maximum likelihood estimation could be used as an option to obtain better performance in the proposed algorithm.

REFERENCES

Al-Hamadi, H. M. and Soliman, S. A. (2004), Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model, *Electric, Power Syst. Res.*, **68**(1), 47-59.

Amjady, N. and Keynia, S. (2009), Short term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm, *Energy*, **34**, 46-57.

Bashir, Z. A. and El-Hawary, M. E. (2009), Applying wavelets to short-term load forecasting using PSO-based neural networks, *IEEE Trans. Power Syst.*,

24, 20-27.

Benaouda, D., Murtagh, F., Starck, J. L. and Renaud, O. (2006), Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting, *Neurocomputing*, **70**, 139-154.

Che, J., Wang, J., and Wang G. (2012), An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting, *Energy*, **37**(1), 657-664.

Dongxiao, N., Wang, Y., and Wu, D. D. (2009), Power load forecasting using support vector machine and ant colony optimization, *Expert Systems with Applications*, **37**(3), 2531-2539.

Fan, S. and Chen, L. (2006), Short-term load forecasting based on an adaptive hybrid method, *IEEE Trans. Power Syst*., **21**(1), 392-401.

Hagan, M. T. and Behr, S. M. (1987), The time series approach to short term load forecasting, *IEEE Trans. Power Syst*., **2**(3), 785-791.

Hippert, H. S., Pedreira, C. E., and Souza, R. C. (2001), Neural networks for short-term load forecasting: A review and evaluation, *IEEE Trans. Power Syst*., **16**(1), 44-55.

Hsu, C. W., Chang, C. C., and Lin, C. J. (2003), A practical guide to support vector classification, Department of Computer Science, *National Taiwan University*.

Nikolaev, N. and Tino, P. (2005), Sequential relevance vector machine learning from time series, Neural Networks, IJCNN, Proceedings, *IEEE International Joint Conference on*, **2**, 1308-1313

Pandey, A. S., Singh, D., and Sinha, S. K. (2010), Intelligent hybrid wavelet models for short-term load forecasting, *IEEE Trans. Power Syst*., **25**, 1266-1273.

Ping-Feng, P. and Wei-Chiang, H. (2005), Support vector machines with simulated annealing algorithms in electricity load forecasting, *Energy Conversion and Management*, **46**(17), 2669-2688.

Qing Duan, J.-G., Zhao, L. N., and Ke, L. (2008), Regression Based on Sparse Bayesian Learning and the Applications in Electric Systems, *Natural Computation*, Fourth International Conference on, 106-110.

Rocha Reis, J. and da Silva, A. P. A. (2005), Feature extraction via multiresolution analysis for short-term load forecasting, *IEEE Trans. Power Syst*., **20**, 189-198.

Song, K. B., Baek, Y. S., Hong, D. H., and Jang, G. (2005), Short-term load forecasting for the holidays using fuzzy linear regression method, *IEEE Trans. Power Syst*., **20**(1), 96-101.

Taylor, J. W. and McSharry, P. E. (2007), Short-Term Load Forecasting Methods: An Evaluation Based on European Data, *IEEE Trans. Power Syst*., **22**(4), 2213-2219.

Tipping, M. E. (2001), Sparse Bayesian learning and the relevance vector machine, *J. Mach. Learn. Res*., **1**, 211-244.

Zhang, B.-L. and Dong, Z.-Y. (2001), An adaptive neural-wavelet model for short term load forecasting, *Electr. Power Syst. Res*., **59**, 121-129.

Zhinong, W., Xiaolu, L., Cheung, K. W., Jiang, W., and Shuaidong, H. (2013), A new short-term load forecasting model based on relevance vector machine, *Electricity Distribution*, 22nd International Conference and Exhibition on, 10-13.