Electrical Power Load Forecasting using Hybrid Self-Organizing Maps and Support Vector Machines

J. Nagi, K. S. Yap
S. K. Tiong, S. K. Ahmed, *Member, IEEE*

Abstract—Forecasting of future electricity demand is very important for decision making in power system operation and planning. In recent years, due to the privatization and deregulation of the power industry, accurate forecasting of future electricity demand has become an important research area for secure operation, management of modern power systems and electricity production in the power generation sector. This paper presents a novel approach for mid-term electricity load forecasting. It uses a hybrid artificial intelligence scheme based on self-organizing maps (SOMs) and support vector machines (SVMs). According to the similarity degree of time series input samples, the SOM is used as a filtering scheme to cluster historical electricity load data into two subsets using the Kohonen rule in an unsupervised manner. As a novel learning machine, the SVM based on statistical learning theory is used for prediction, using support vector regression (SVR). Two epsilon-SVRs are employed to fit the training data of each SOM clustered subset individually in a supervised manner for load prediction. The proposed hybrid SOM-SVR model is evaluated in MATLAB on the electricity load dataset used in the European Network on Intelligent Technologies (EUNITE) competition, arranged by the Eastern Slovakian Electricity Corporation. This proposed model is robust with different data types and can deal well with non-stationarity of load series. Practical application results show that this hybrid technique gives far better prediction accuracy for mid-term electricity load forecasting compared to previous research findings.

Keywords—Support vector machine, self-organizing map, electrical power load forecasting, artificial intelligence.

I. INTRODUCTION

Load forecasting has always been a key instrument in power system operation. Many operational decisions in power systems, such as unit commitment, economic dispatch, automatic generation control, security assessment, maintenance scheduling, and energy commercialization depend on the future behavior of loads. In particular, with the rise of deregulation and free competition of the electric power industry all around the world, load forecasting has become more important than ever before. Load forecasts are currently being considered vital for energy transactions in competitive electricity markets[1]. In the recent years, accuracy of electricity load forecasting has received more attention on a regional and national scale. The error of electricity load forecasting may increase the cost of operation, where overestimation of the future load can result in excess supply. In contrast, underestimation of load leads to failure in providing enough electricity and implies high costs in peaking units[5].

During the last four decades, a wide variety of techniques have been used for the problem of load forecasting. Such a long experience in dealing with the load forecasting problem has revealed some time series modeling approaches based on artificial neural networks (ANNs) and statistical methods. Statistical models include moving average and exponential smoothing methods such as, multi-linear regression models, stochastic process, data mining approaches, autoregressive and moving averages (ARMA) models, Box-Jenkins methods, and Kalman filtering-based methods[1]. However, electric load time series are usually nonlinear functions of exogenous variables. Therefore, to incorporate non-linearity, ANNs have received much more attention in solving problems of electricity load forecasting. However, one major risk in using ANN models is the possibility of excessive training data approximation, i.e., overfitting, which usually increases the out-of-sample forecasting errors[4]. Hence, due to the empirical nature of ANN procedures their application is cumbersome and time consuming.

Recently, new approaches based on machine learning techniques using support vector machines (SVMs) have been proposed for electricity load forecasting. Support vector regression (SVR) used in SVMs is a new and powerful machine learning technique for regression based analysis in statistical learning theory. SVRs established on the structure risk minimization principle (SRM) have shown to be very resistant to the overfitting problem of ANNs, by achieving a high generalization performance in solving forecasting problems of various time series[1]. Electricity demand now days is becoming difficult to forecast because of the variability of non-stationarity of load series that results from non-uniform demand supply, price-dependent loads, and time-varying prices. Therefore, more sophisticated forecasting tools with higher accuracy are required for modern power systems.

This paper focuses on mid-term electricity load forecasting to predict the month-ahead electricity load profile, using historical electricity data from the 2001...
EUNITE competition[7] with a time series modeling approach. The EUNITE competition data is an internet-based dataset and has been employed to allow reproduction of the results presented in this paper. The theoretical parts are addressed in Sections II-IV, self contained as much as possible. Section V discusses the design and architecture of the hybrid electricity load forecasting model. Section VI shows experimental results and Section VII presents concluding remarks.

II. EUNITE DATA

A. EUNITE Dataset

The historical electricity load dataset used in the EUNITE competition[7] contains the following data: maximum daily electricity loads, average daily temperatures and annual holidays from January 1, 1997 to December 31, 1998 as shown in Table I. The maximum daily values of the electricity load for the 31 days of January 1999 are to be forecasted using the given data for the preceding two years.

![Graph of Average maximum daily load data from 1997 to 1998](image)

![Graph of Average maximum daily load data for January 1997](image)

B. Data Analysis

Observations regarding the EUNITE electricity load dataset are investigated to determine the relations between the load demand and other information such as, temperature and annual holidays. The following observations are concluded for the given dataset:

1. **Electricity Load Data**: Firstly, through simple analysis of the graphs representing the data, it is observed that electricity load data follows seasonal patterns, i.e. high demand for electricity in the winter, while low demand in the summer, as shown in Figure 1. Secondly, another load pattern is observed, where load periodicity exists in the profile of every week, i.e. load demand on the weekend is usually lower than that on weekdays, as shown in Figure 2. In addition, electricity demand on Saturday is a little higher than that on Sunday[3] and the peak load occurs in the middle of the week, i.e., Wednesday.

2. **Temperature Influence**: Through analysis it is observed that the load data has seasonal variation, which indicates a great influence in climatic conditions. A negative correlation between the load data and daily average temperature is observed which is found be -0.868[3] as shown in Figure 3. This observation concludes that due to the use of heating, higher temperature causes lower electricity demands.

C. **Mean Absolute Percentage Error**

Accuracy of load forecasting depends upon the error metric, Mean Absolute Percentage Error (MAPE) of the predicted result. MAPE is defined as follows[4]:

$$MAPE = 100 \frac{1}{n} \sum \frac{|L_{ia} - L_{ip}|}{L_{ia}}, \quad n = 31$$

where $L_{ia}$ and $L_{ip}$ are the actual and the predicted values of maximum daily electrical load on the $i^{th}$ day of the year 1999 respectively, and $n$ is the number of days in January 1999.

![Table I: Given EUNITE Electricity Load Dataset](image)

III. SELF-ORGANIZING MAPS

A. **Overview**

Self-organizing maps (SOMs) also known as a **Kohonen Maps** are well known subtypes of ANNs. SOMs are an unsupervised learning process, which learn the distribution
of a set of patterns without any class information, while having the property of topology preservation. In a SOM, there is a competition among the neurons to be activated or fired. A SOM network identifies a winning neuron using the same procedure as employed by a competitive layer model. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen Rule[3].

B. Kohonen Rule

The Kohonen rule allows the weights of a neuron to learn an input vector. During the learning phase, the neuron with weights closest to the input data vector is declared as the winner. Then, weights of all of the neurons in the neighborhood of the winning neuron are adjusted by an amount inversely proportional to the Euclidean distance. This algorithm clusters and classifies the dataset based on the set of attributes used. The algorithm is as follows[3]:

1. Initialization: Choose random values for the initial weight vectors \( w_j(0) \), the weight vectors being different for \( j = 1, 2, ..., l \) where \( l \) is the total number of neurons.

2. Sampling: Draw a sample \( x \) from the input space with a certain probability.

3. Similarity Matching: Find the best matching (winning) neuron \( i(x) \) at time \( t \), \( 0 < t \leq T \) by using the minimum distance Euclidean criterion:

\[
   i(x) = \arg \min \| x(n) - w_j \|, \quad j = 1, 2, ..., l
\]

4. Updating: Adjust the synaptic weight vector of all neurons by using the update formula:

\[
   w_j(n+1) = w_j(n) + \eta(n) h_{i,j}(n) (x(n) - w_j(n))
\]

where \( \eta(n) \) is the learning rate parameter, and \( h_{i,j}(n) \) is the neighborhood function centered around the winning neuron \( i(x) \). Both \( \eta(n) \) and \( h_{i,j}(n) \) are varied dynamically during learning for best results.

5. Repetition: Continue with step 2 until no noticeable changes in the feature map are observed.

IV. SUPPORT VECTOR MACHINES

Support vector machines (SVMs), introduced by Vapnik, are a set of related supervised learning methods used for classification and regression. Support vector regression (SVR) for SVMs can be used for time series prediction, which is useful for problems characterized by non-linearity and high dimension. The basic concept of SVR is to map the input data, \( x \), non-linearly into a higher dimensional feature space[5]. Given training data \((x_i, y_i), ..., (x_n, y_n)\) where \( x_i \) is the input pattern, and \( y_i \) is the associated output value of \( x_i \), to solve an optimization problem with[1,3]:

\[
   \min_{w, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i^*)
\]

subject to the following constraints:

\[
   y_i - w^T \Phi(x_i) - b \leq \epsilon + \xi_i, \quad i = 1, 2, ..., N
\]

\[
   w^T \Phi(x_i) + b - y_i \leq \epsilon + \xi_i^*, \quad i = 1, 2, ..., N
\]

where \( x_i \) is mapped to a higher dimensional space by the function \( \Phi \), \( \xi_i \) and \( \xi_i^* \) are slack variables representing lower and upper training errors respectively, subject to the \( \epsilon \)-insensitive tube \((w^T \Phi(x) + b) \leq \epsilon \). The constant \( C > 0 \) determines the tradeoff between the flatness and losses. The parameters which control regression quality are: the cost of error \( C \), width of the \( \epsilon \)-insensitive tube \( \epsilon \), and the tolerance of termination criterion \( \sigma[1,3] \).

The constraints of (4.1) imply that data \( x_i \) should be put in the tube \( \epsilon \). If \( x_i \) is not in the tube, there is an error \( \xi_i \) or \( \xi_i^* \) that can be minimized in the objective function. SVR avoids under-fitting and over-fitting of the training data by minimizing the training error \( C \sum_i (\xi_i + \xi_i^*) \) for \( i = 1, 2, ..., n \) as well as the regularization term \( \frac{1}{2} w^T w \). Since \( \Phi \) might map \( x_i \) to a high or infinite dimensional space, instead of solving for (4.1) in a high dimension, its dual problem is solved with[1,3]:

\[
   \min_{\alpha, \alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)
\]

subject to the following constraints:

\[
   \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0
\]

\[
   0 \leq \alpha_i, \quad \alpha_i^* \leq C
\]

where \( Q_{ij} = \Phi(x_i)^T \Phi(x_j) \). However, this inner product is computationally heavy because \( \Phi(x) \) has too many elements. Hence a “kernel trick” is applied to do the mapping implicitly, i.e. employing some special forms, inner products in a higher space can be calculated in the original space[1,3]. The four kernel functions used are listed below:

- **Linear kernel**

  \[
  K(x, y) = x^T y
  \]

- **Polynomial kernel**

  \[
  K(x, y) = (y^TXy + r)^d, \quad \gamma > 0
  \]

- **Radial basis function (Gaussian) kernel**

  \[
  K(x, y) = \exp(-\gamma \|x - y\|^2), \quad \gamma > 0
  \]

- **Sigmoid kernel**

  \[
  K(x, y) = \tanh(\gamma x^T y + r), \quad \gamma > 0
  \]

Fig. 3. Correlation between maximum daily load and daily temperature

IV. SUPPORT VECTOR MACHINES

Support vector machines (SVMs), introduced by Vapnik and co-workers, are a set of related supervised learning methods used for classification and regression. Support vector regression (SVR) for SVMs can be used for time series prediction, which is useful for problems characterized by non-linearity and high dimension. The basic concept of SVR is to map the input data, \( x \), non-linearly into a higher dimensional feature space[5]. Given training data \((x_i, y_i), ..., (x_n, y_n)\) where \( x_i \) is the input pattern, and \( y_i \) is the associated output value of \( x_i \), to solve an optimization problem with[1,3]:

\[
   \min_{w, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i^*)
\]

subject to the following constraints:

\[
   y_i - w^T \Phi(x_i) - b \leq \epsilon + \xi_i, \quad i = 1, 2, ..., N
\]

\[
   w^T \Phi(x_i) + b - y_i \leq \epsilon + \xi_i^*, \quad i = 1, 2, ..., N
\]

where \( x_i \) is mapped to a higher dimensional space by the function \( \Phi \), \( \xi_i \) and \( \xi_i^* \) are slack variables representing lower and upper training errors respectively, subject to the \( \epsilon \)-insensitive tube \((w^T \Phi(x) + b) \leq \epsilon \). The constant \( C > 0 \) determines the tradeoff between the flatness and losses. The parameters which control regression quality are: the cost of error \( C \), width of the \( \epsilon \)-insensitive tube \( \epsilon \), and the tolerance of termination criterion \( \sigma[1,3] \).

The constraints of (4.1) imply that data \( x_i \) should be put in the tube \( \epsilon \). If \( x_i \) is not in the tube, there is an error \( \xi_i \) or \( \xi_i^* \) that can be minimized in the objective function. SVR avoids under-fitting and over-fitting of the training data by minimizing the training error \( C \sum_i (\xi_i + \xi_i^*) \) for \( i = 1, 2, ..., n \) as well as the regularization term \( \frac{1}{2} w^T w \). Since \( \Phi \) might map \( x_i \) to a high or infinite dimensional space, instead of solving for (4.1) in a high dimension, its dual problem is solved with[1,3]:

\[
   \min_{\alpha, \alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)
\]

subject to the following constraints:

\[
   \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0
\]

\[
   0 \leq \alpha_i, \quad \alpha_i^* \leq C
\]

where \( Q_{ij} = \Phi(x_i)^T \Phi(x_j) \). However, this inner product is computationally heavy because \( \Phi(x) \) has too many elements. Hence a “kernel trick” is applied to do the mapping implicitly, i.e. employing some special forms, inner products in a higher space can be calculated in the original space[1,3]. The four kernel functions used are listed below:

- **Linear kernel**

  \[
  K(x, y) = x^T y
  \]

- **Polynomial kernel**

  \[
  K(x, y) = (y^TXy + r)^d, \quad \gamma > 0
  \]

- **Radial basis function (Gaussian) kernel**

  \[
  K(x, y) = \exp(-\gamma \|x - y\|^2), \quad \gamma > 0
  \]

- **Sigmoid kernel**

  \[
  K(x, y) = \tanh(\gamma x^T y + r), \quad \gamma > 0
  \]
A. Feature Selection

Features have been selected for use in the hybrid forecasting model to provide better prediction accuracy. Four features from the EUNITE competition dataset have been selected to evaluate the proposed forecasting model, which are as follows:

- Maximum daily electricity load
- Average daily temperature
- Day type (weekday or weekend)
- Annual holidays

B. Data Preprocessing for Time Series Modeling

For a time series alternating in time, a non-linear discrete-time dynamic model for load forecasting is represented by:

\[ y(t + k) = f(y(t), y(t + 1), \ldots, y(t + k - 1)) \quad k = 7 \quad (5.1) \]

where \( y(t) \) is a vector representing the daily electricity load profile at time \( t = 0, 1, 2, \ldots, N \), and \( k \) is the order of the dynamic system, which is a pre-determined time series shift constant. For the case of this experiment \( k = 7 \) is used in (5.1); i.e., historical load data of the first one week is used to calculate the load data of eighth day. Similarly, historical data from the second day until the eighth day (next 7 days) is used to calculate the load data of the ninth day, and so on, for the entire load data for the preceding two years.

C. Architecture of Hybrid Network

A hybrid artificial intelligence-based network is applied to reconstruct the dynamics of electricity load consumption using the time series of its observables. The proposed hybrid electricity forecasting model is shown in Figure 4, where the parameter subscripts: \( \text{train}, \text{clust} \) and \( \text{test} \) represent the training, clustered and testing data respectively. This model is based on a two stage architecture of the SOM and the epsilon-SVR.

The hybrid electricity load forecasting model presented in this paper is developed, trained, and tested using MATLAB™ R2007b v7.5.0. The computer used was a Dell PowerEdge SC1420 Workstation with Windows XP, a 3.00 GHz Intel Xeon Processor and 512 MB of RAM. The Kohonen SOM model was implemented using the MATLAB Neural Network Toolbox and the epsilon-SVR models were implemented using the LIBSVM v2.85-1 Toolbox[6] for MATLAB.

D. Data Representation

Firstly, load and temperature time series data are linearly scaled (normalized) in the range from 0 to 1. Secondly, useful information is selected and a proper combination of features is selected to prepare the training dataset[3]. To represent calendar information, the day type feature uses seven binary digits [0, 1] to encode information for weekdays and weekends. The annual holidays feature uses one binary digit to encode information. To represent the historical electricity load for the preceding two years from January 1, 1997 to December 31, 1998, seven vector numerics for \( k = 7 \) in (5.1), from the preprocessed time series are used, each with a shift of one day. For temperature data only one numerical attribute is used from the normalized temperature data.
optimization (SMO) algorithm individually, for each SOM clustered subset in a supervised manner.

6. Next, two Euclidean distances between the cluster centers and the testing data are calculated; \(d_1\) and \(d_2\).

7. Logical comparison of the Euclidean distances \((d_1 < d_2)\) and \((d_2 < d_1)\) as shown in Figure 4, provide the decision of selecting the appropriate \(\varepsilon\)-SVR model to fit the training data. The input variables for testing the two \(\varepsilon\)-SVR models are shown in Table III.

### Table III
<table>
<thead>
<tr>
<th>Input</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>Load vector ((L_{train}))</td>
<td>Seven vector numerics of one week from 97 to 98</td>
</tr>
<tr>
<td>8</td>
<td>Temperature ((T_{train}))</td>
<td>One vector numeric for average daily temperature from 97 to 98</td>
</tr>
<tr>
<td>9-15</td>
<td>Day type ((D_{train}))</td>
<td>Seven binary digits representing calendar information from 97 to 98</td>
</tr>
<tr>
<td>16</td>
<td>Holiday ((H_{train}))</td>
<td>One binary digit representing annual holidays from 97 to 98</td>
</tr>
</tbody>
</table>

8. Cross-validation is used by dividing the training data into two sets: training set and validation set. The validation set uses different sets of one week historical data from the preceding two years data (from January 1, 1997 to December 31, 1998) to perform cross-validation.

9. The best suited \(\varepsilon\)-SVR kernel, its parameters and optimal \(\varepsilon\)-SVR parameters: cost of error \(C\), width of the \(\varepsilon\)-insensitive tube (loss function) \(\varepsilon\), and the tolerance of termination criterion \(\sigma\), are selected by generating 1 million random values for each parameter, and iterating them with different combinations to find the optimal set.

10. Finally, the month-ahead electricity load forecast for the 31 days of January 1999 is predicted using cross validation, and the MAPE is calculated.

### VI. EXPERIMENTAL RESULTS

#### A. Kernel Selection

The behavior of four different kernels namely: linear, polynomial, radial basis function (RBF) and sigmoidal for the \(\varepsilon\)-SVRs was observed using 10-fold cross-validation. Comparison of these kernels based on the accuracy of prediction of the forecasting model is shown in Figure 5.

The parameter values selected for the kernels (defined in 4.3, 4.4, 4.5 and 4.6) are shown in Table IV, where the cost of error \(C\), and kernel parameter \(\gamma\), were selected to be 100 and 0.001 respectively. From the 10-fold cross-validation results obtained, the RBF kernel yielded the best accuracy, resulting in a MAPE of 1.73%. Hence, from this point onwards, all tests performed were done using the RBF kernel.

#### B. Parameter Optimization

Through cross-validation trials, the best values for the \(\varepsilon\)-SVR parameters \(\varepsilon\) and \(\sigma\) were found to be 0.02 and 0.1 respectively. For finding the optimal values of \(C\) and the RBF kernel parameter \(\gamma\), prediction accuracy of the hybrid electricity load forecasting model using different combinational values of the two parameters was determined. From the results obtained, two parameter values for \(\varepsilon\)-SVR-1 and \(\varepsilon\)-SVR-2 were found, where \(C_1 = 213.8381\), \(\gamma_1 = 0.0010\), \(C_2 = 52.8170\) and \(\gamma_2 = 0.0069\) resulting in a MAPE of 1.42%.

#### C. Comparison of Forecasting Models

For a comparative study, numerical simulations comparing with other load forecasting methods were also conducted. Besides using a hybrid SOM-SVR model, simulations using the EUNITE dataset were conducted for a stand-alone SVR and Multi-layer Back-propagation Neural Network (ML-BPNN) model, as shown in Figure 6.

Overall results obtained, indicate that by far neural networks alone are not satisfactory as shown in Table V. The hybrid SOM-SVR model proved to be superior to the SVR and MLP-BPNN models, resulting in a MAPE of...
1.42%. In addition, comparisons with previous work done on electricity load forecasting in [3] and [4], using the EUNITE competition[7] dataset revealed that, the proposed hybrid electricity load forecasting model has far better prediction accuracy than both of the forecasting models.

VII. CONCLUSION

In this paper, a novel technique for mid-term electricity load forecasting has been presented based on a hybrid SOM-SVR model. Experimental results obtained demonstrate the feasibility of successfully applying this new hybrid model for electricity load forecasting. The proposed model has three notable advantages. Firstly, it has the ability to tackle with the non-stationarity in the electricity load time series. Secondly, it can treat regular days (weekdays and weekends) and annual holidays with different schemes. Lastly, it has strong robustness and can be easily modified for different power systems. In addition, the structural risk minimization principle in SVRs proved to be superior compared to the empirical risk minimization principle employed by conventional ANNs. Comparisons with the previous winner[3] of the EUNITE competition[7], and [4] shows that this hybrid electricity load forecasting model has far better prediction accuracy compared to its predecessors.

REFERENCES


