1-1-2007

Application of least squares support vector machines in medium-term load forecasting

Mohammadreza Afshin
Ryerson University

Application of Least Squares Support Vector Machines in Medium-Term Load Forecasting

by

Mohammadreza Afshin

A project
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Engineering
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2007

Author's Declaration

I hereby declare that I am the sole author of this project.
I authorize Ryerson University to lend this project to other institutions or individuals for the purpose of scholarly research.
Signature

I further authorize Ryerson University to reproduce this project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
Signature

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this project. Please sign below, and give address and date.

Abstract

Power load forecasting is essential in the task scheduling of every electricity production and distribution facility. In this project, we study the applications of modern artificial intelligence techniques in power load forecasting. We first investigate the application of principal component analysis (PCA) to least squares support vector machines (LS-SVM) in a week-ahead load forecasting problem.

Then, we study a variety of tuning techniques for optimizing the least squares support vector machines' (LS-SVM) hyper-parameters. The construction of any effective and accurate LS-SVM model depends on carefully setting the associated hyper-parameters. Popular optimization techniques including Genetic Algorithm (GA), Simulated Annealing (SA), Bayesian Evidence Framework and Cross Validation (CV) are applied to the target application and then compared for performance time, accuracy and computational cost.

Analysis of the experimental results proves that LS-SVM by feature extraction using PCA can achieve greater accuracy and faster speed than other models including LS-SVM without feature extraction and the popular feed forward neural network (FFNN). Also, it is observed that optimized LS-SVM by bayesian evidence framework can achieve greater accuracy and faster speed than other techniques including LS-SVM tuned with genetic algorithm, simulated annealing and *10-fold* cross validation.

Acknowledgments

I would like to thank my supervisor, Dr. Alireza Sadeghian, for his valuable support during this project, for his comments and insights, and for introducing and initiating the idea of this work.

Also, I appreciate Mr. Douglas Barnard, Customer Communications, Ontario's IESO for valuable guide and providing information for this project.

I cannot end without thanking my family. This work was not possible without their support, love and encouragement. They were always there for me and it is to them I dedicate this project.

Acronyms

ANN	Artificial Neural Network
BEF	Bayesian Evidence Framework
BP	Back Propagation
CV	Cross Validation
ERM	Empirical Risk Minimization
FFNN	Feed Forward Neural Network
GA	Genetic Algorithm
GP	Gaussian Process
ICA	Independent Component Analysis
IESO	Independent Electricity System Operator
KPCA	Kernel Principal Component Analysis
LOO-CV	Leave-One-Out Cross-Validation
LS-SVM	Least Squares Support Vector Machines
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layered Perceptron
PCA	Principal Component Analysis
QP	Quadratic Programming
RBF	Radial Basis Function
SA	Simulated Annealing
SRM	Structural Risk Minimization
SVM	Support Vector Machines

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Load forecasting has always been a very important issue in economic and reliable power systems planning and operation [1], [2]. Short term forecasting (up to week ahead) is required for the optimum allocation of generation, unit commitment and scheduling functions, evaluation of net interchange, and system security analysis. Medium term forecasts ranging between one week and one year are used for maintenance planning, fuel scheduling and hydro reservoir management. Both energy and peak demand estimates are required for medium term forecasting to assess the fuel requirements and to check the adequacy of plant margins after allowing for unavailability of system components due to maintenance and breakdown. The time horizon for long term demand projections is from one to ten years, which corresponds to lead times required for planning and development of transmission and distribution systems, and generation facilities [1].

Another area of application involves load flow studies, including contingency planning, load shedding, and load security strategies [3]. Economically, accuracy in load forecasting can allow utilities to operate at lower cost which can potentially contribute to millions of dollars in savings in major electric power utilities.

When analyzing the reports and trends in the industry, it is clear that Week-Ahead daily peak load forecast plays an important role in the day-to-day operations of every power producing company.

1.1 Motivations and Objectives

A wide variety of power forecasting techniques have been investigated and introduced so far. Traditional prediction techniques treat the problem as a special instance of parameter estimation where the estimated model is regarded as the predictor. Regression methods necessitate the identification of relevant variables with strong correlation to electric loads such as temperature, humidity and winds, etc. The time series method of load forecasting involves the examination of historical data, extracting essential data characteristics, and effectively projecting these characteristics into the future. The essential requirements for a good forecasting model are accuracy and reliability during all seasons and varying weather conditions [4], [5].

These statistical methods of load forecasting have some theoretical limitations that make difficult the fulfillment of the above mentioned requirements. They are inefficient due to their dependence on the functional relationship between the load and the weather variables, and also they are numerically unstable. Recognizing various limitations of these time-series and stochastic methods in terms of computational effort, amount of historical data required and accuracy of results, the emphasis has shifted to application of Artificial Intelligence based methods to load forecasting [6].

During the past decade, artificial neural networks have emerged as a very successful approach to power load forecasting [7]. However, this methodology has weaknesses when load patterns are not similar to those of weekdays, e.g., on weekends and public holidays. Also, the neural networks require many training samples and frequent retraining due to changes in seasonal conditions, and learning speed is reported to be comparatively slow [8], [9].

Recently, support vector machines (SVM) have attracted much attention in load forecasting field [10]. They have been successfully employed to solve most nonlinear regression and time series problems [11], [12]. Typical advantages of SVMs include good generalization performance and the absence of local minima.

The theory of SVM is based on statistical learning theory pioneered by Vapnik *et al* [13].

Unlike most of the traditional methods which implement the empirical risk minimization (ERM) principal, SVMs implement the structural risk minimization (SRM) principal, which seeks to minimize an upper bound of the generalization error rather than minimizing the training error. Essentially, SVMs map the inputs into a higher dimensional feature space in which a linear regressor is constructed by minimizing an appropriate cost function. Using Mercer's theorem [14], [15], the regressor is obtained by solving a finite dimensional quadratic programming (QP) problem in the dual space avoiding explicit knowledge of the high dimensional mapping and using only the related kernel function. Therefore, the solution of SVM is always globally optimal. SVM is a powerful solution for problems with nonlinearity and high dimension, and it improves both training time and accuracy in comparison with other competitor forecasting tools [11], [16].

LS-SVM is a simplified form of SVM that uses equality constraints (instead of the inequality constraints implemented in standard SVMs) and a least squares error term to obtain a linear set of equations in the dual space [17].

The other aspect of load forecasting rather than prediction algorithm is the feature selection and extraction. Here, Principal Component Analysis (PCA) is used in order to identify the most influential inputs in the context of the forecasting model [18]. It evaluates the input variables according to the projection of the largest eigenvector of the correlation matrix on the initial basis vector. This technique creates a new set of input variables which are orthogonal, so that they are uncorrelated with each other; the resulting orthogonal, principal components are ordered so that those with the largest variation take precedence; and components that contribute the least to the variation in the data set can be eliminated. In this work, the original inputs are first transformed into uncorrelated principal components using PCA. These new features are then used as inputs of LS-SVMs to solve time series load forecasting problems [19].

In any LS-SVM used for either classification or regression tasks, if embedded hyperparameters are not well chosen, results will not be satisfactory. The set of parameters that we select depends on the type of SVM used. The majority of works in the literature, rely on

cross validation for their optimization of SVM parameters (leave-one-out cross validation) [10], [17] - [20]. Global optimization techniques such as genetic algorithm and simulated annealing are used in exceptional cases [21] - [22]. Popular optimization techniques including genetic algorithms (GA), simulated annealing (SA), bayesian evidence framework (BEF) and cross validation (CV) are applied to tune the LS-SVM hyper-parameters, and then they are compared for speed, accuracy and computational complexity [23]. In this project, the kernel function variable (δ) and the regularization parameter (γ) in SVM are the hyper-parameters of interest. Our goal is to predict the daily peak load demand of the coming week. Evaluation of algorithms is typically based on two error metrics, namely, mean absolute error (MAE) and mean absolute percentage error (MAPE). Here, we rely on MAPE for general evaluation, and the cost function of the algorithms is defined as:

$$MAPE = \frac{\sum_{i=1}^{n} \left| \frac{L_i - \hat{L}_i}{L_i} \right|}{n} \times 100\%, \tag{1.1}$$

$$MAE = \frac{\sum_{i=1}^{n} \left| L_i - \hat{L}_i \right|}{n} \tag{1.2}$$

where L_i and $\hat{L}_i$ represent the actual and predicted peak daily loads and n is the number of the days in forecasting period.

1.2 Organization

The organization of this project is as follows: In Chapter 2, we describe the feature selection and extraction processes. In addition, the analysis of load trend is also propounded. In Chapter 3 the concepts and fundamentals of PCA are introduced (Section 3.1), followed by an overview of least squares support vector machines theory and its hyper-parameters tuning algorithms (Sections 3.2 and 3.3). In Chapter 4, the results of the proposed algorithms and other competitors are presented and compared. Finally, discussions related to this research along with conclusions are presented in Sections 5.1 and 5.2.

Chapter 2

Background

2.1 Historical Data

2.1.1 Load Demand Analysis

The following information are usually used for load forecasting:

1. Calendar information
2. Temperature profiles
3. Historical load demands

It is known that the load consumption has multiple seasonal patterns, corresponding to a weekly and annually periodicity. Depending on geographical location, we may encounter different climatic conditions. Fig. 2.1 gives a simple description of the maximum daily load demand from 2003 to 2006 in Toronto, Canada. This pattern implies the relation between electricity usage and weather conditions in different seasons. Also in Fig. 2.2, we can clearly see the seasonal periodicity of temperature during winters and summers for the same period in Toronto. Moreover, a closer look at Fig. 2.3 shows that load consumption in summer is extensively more than winter, due to extreme increases in temperature. In comparison with other Canadian cities, Toronto has a milder winter. Moreover, during June to August, the load consumption reaches the maximum of the year.

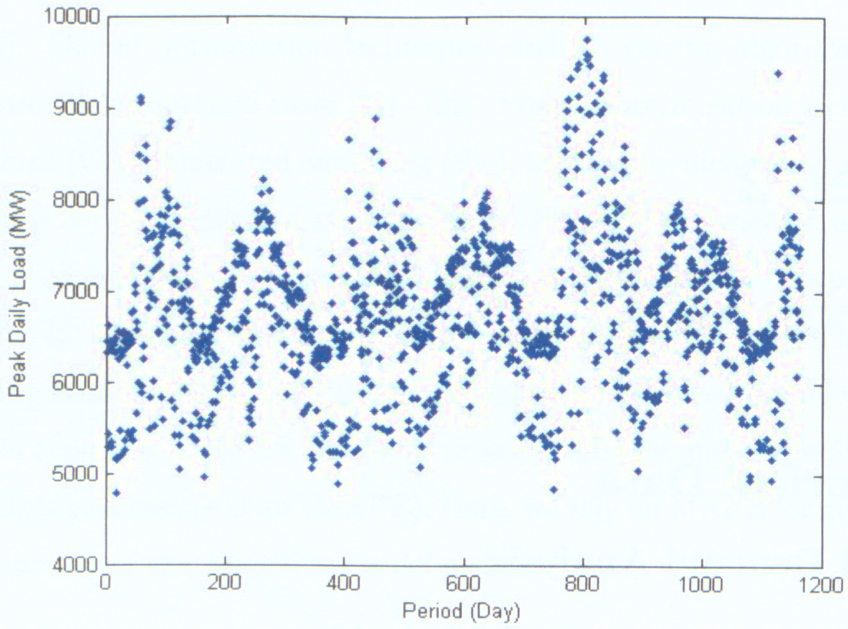

Figure 2.1: Load vs. Time, Toronto

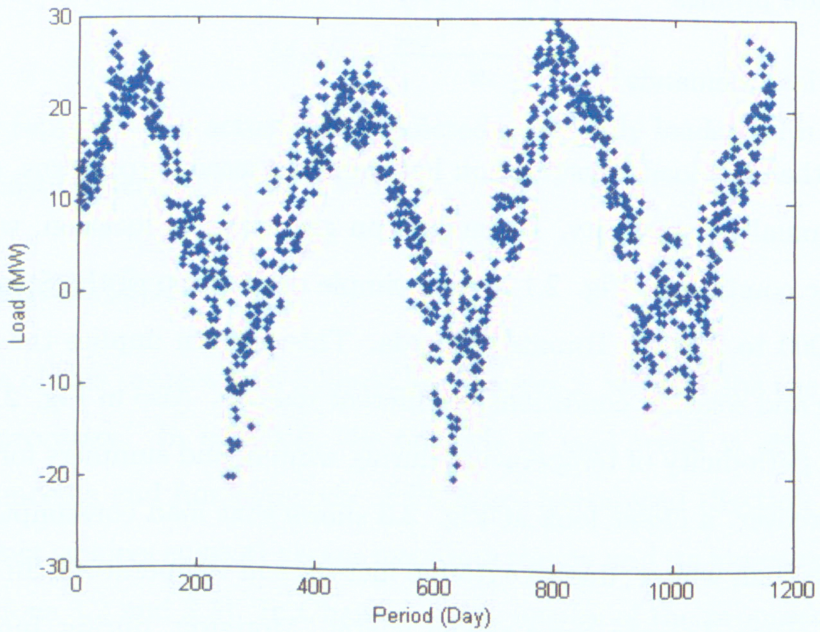

Figure 2.2: Temperature vs. Time, Toronto

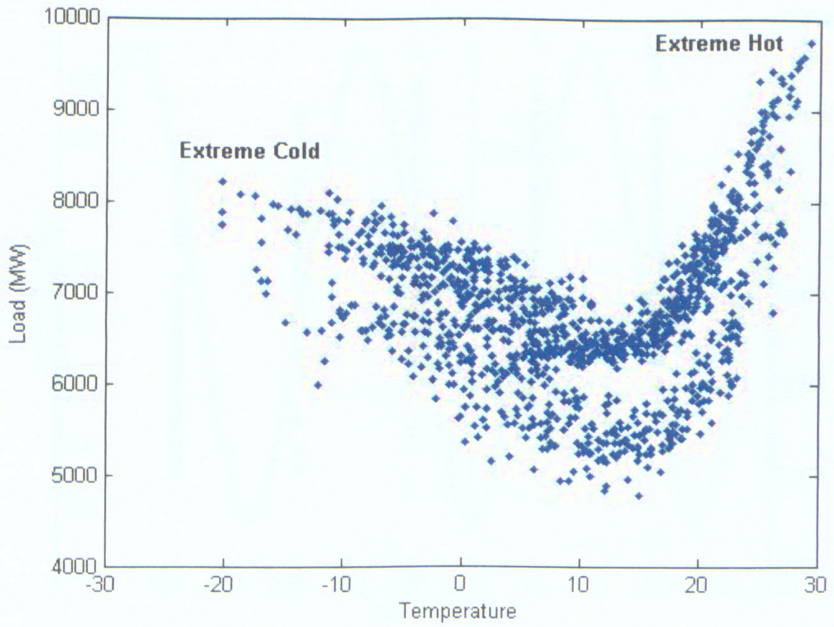

Figure 2.3: Load vs. Temperature Profile, Toronto

Besides, a load periodicity exists in every week. Fig. 2.4 shows that load demand in weekends and holidays is usually lower than weekdays or in other words working day (Monday to Friday). Besides, Electricity demand on Saturday is a little higher than Sunday, because some businesses are open on Saturday and it will increase the amount of consumption in a noticeable manner. On the whole, it can be observed that broadly speaking, the peak load happens in the middle of the week ,i.e., Wednesday. But there are always exceptions to all this.

A. Holiday Exceptions

As it was noted, the load demand is usually lower on holidays. With further study, we will figure out that load usage also depends on what holiday it is. On some major holidays such as Christmas or New Year, the demand may be more affected compared with other holidays.

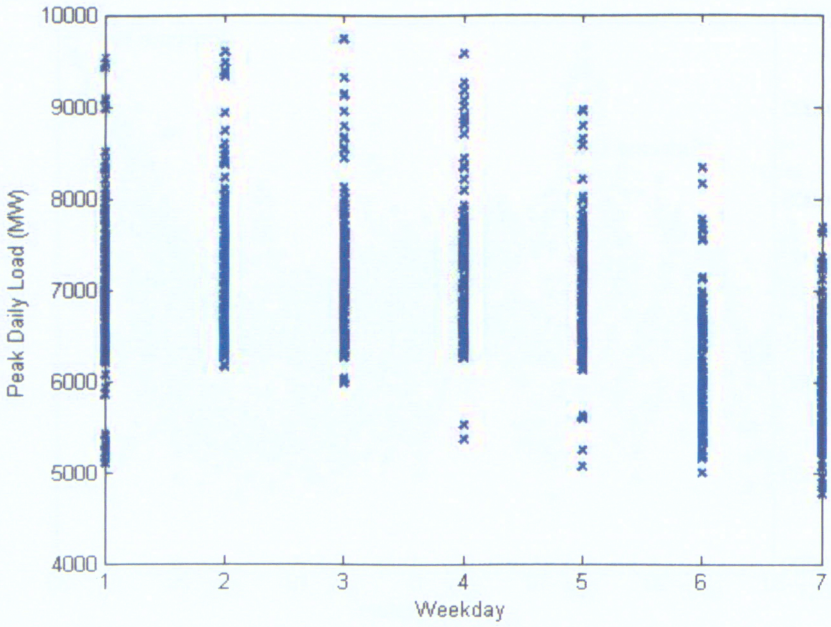

Figure 2.4: Load vs. Period (Day), Toronto

B. Extreme Temperatures

There is a complicated correlation between load demand and daily temperature [20]. As temperature increments, the load demand is decreasing. But there is always exceptions, e.g., in cases that temperatures passes the 30 celsius degrees (e.g. in Toronto) because of the Humidex effect, the load consumption will grow tremendously due to heating, ventilation, and air conditioning systems.

2.1.2 Feature Selection

Historical load data was obtained from Ontario's IESO [24] for Canadian major city, Toronto for the period May 2003 to July 2006 (hourly load).

The Temperature info was acquired from Environment Canada [25] for the same period. The minimum, maximum and average daily temperatures, precipitation and the snow level on the ground are the most important parameters influencing the load usage in a typical

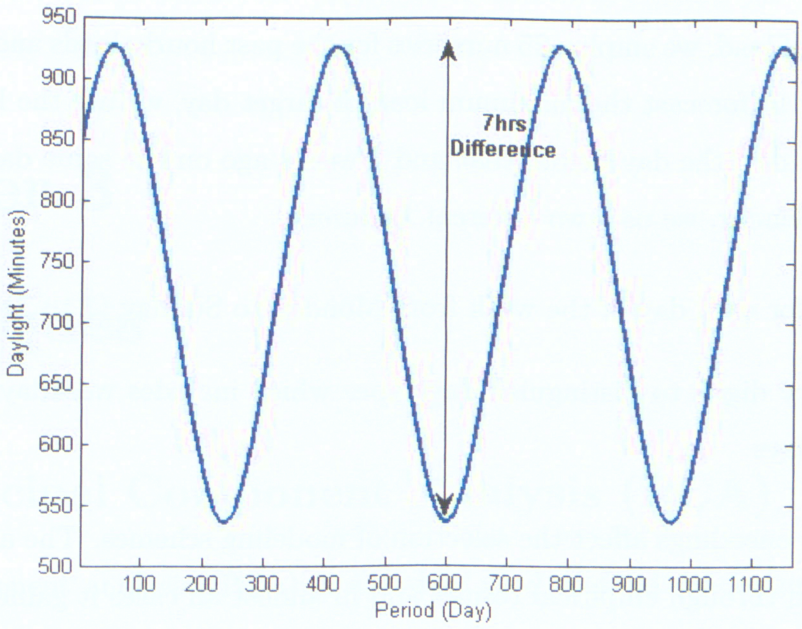

Figure 2.5: Daylight Profile based in minutes, Toronto

day. Other sources, [26], suggest that occasionally cloud cover is being considered in load forecasting, but this parameter alters instantaneously in real time, and for this reason, it is not proved to have a reliable impact on prediction accuracy. Moreover, wind speed and humidity are being used exceptionally, but in this work, we have decided not to consider them, due to rapid changes in their values in Toronto area. Finally, official holidays are determined for the relative years.

Furthermore, there is another parameter to be taken into account in every load forecasting issue and that is the daylight hours. As illustrated in Fig. 2.5, the daytime during summer and winter fluctuates enormously and depending on the geographical location, this has to be factored into data selection. To do so, the time between sunrise and sunset is readily determined as light hours during daytime.

To represent temperature, we use 3 numerical attributes for normalized temperature data. That includes the minimum, maximum and average daily temperatures of the target day (which is itself an accurate forecast) followed by one and two weeks before temperatures

on the same day.

In the matter of load, we employ 25 numerics for the past hourly loads and daily peak load demand. In order to forecast the maximum load of target day, we use the load information of last week same day, the day before that and 2 weeks ago on the same day.

Regarding Calendar, we use two different features:

- An integer for each day of the week from Monday to Sunday (1 to 7)
- Three binary digits to distinguish day types which includes weekdays, weekends and public holidays

Different data encodings affect the selection of modeling schemes. The above selection of data was obtained through empirical results and in almost all cases it gained the least error (MAPE).

2.1.3 Feature Extraction

In developing any forecasting problem, the first step is feature selection (new features are selected from the original inputs) and then feature extraction (new features are transformed from the original inputs) [18]. In other words, all available information can be used as inputs, but irrelevant or strongly correlated features could unfavorably impact the generalization performance due to the dimensionality problem [27].

On this basis, there are changes to the original load data received from Ontario's IESO. A quick look at the historical load data shows that starting $14th$ August 2003 at 16:10, due to huge blackout in southern Ontario and northeastern US, the load in transmission grid decreased tremendously and it remained same for the next day, until the full recovery from the widespread power outage. Thus, to improve the accuracy and be more realistic, the hourly loads of those two days are substituted with the average of the hourly loads of the days before and after the Blackout occurred.

Eventually, the data considered for training, validation and then testing have to be separately clarified.

Chapter 3

Techniques

3.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a widely used method for feature extraction. By calculating the eigenvectors of the sample covariance matrix, PCA linearly transforms the original inputs into uncorrelated new features (called principal components), which are the orthogonal transformation of the original inputs based on the eigenvectors. The obtained principal components in PCA have second-order correlations between the original inputs [28].

Given a set of centered input vectors $x_t(\sum_{t=1}^{l} x_t = 0)$, each of which is of m dimension

$$x_t = (x_t(1), x_t(2), ..., x_t(m))^T,$$

(usually $m < l$) , PCA linearly transforms each vector x_t, into a new one s_t, by

$$s_t = U^T x_t, \tag{3.1}$$

where U is the $m \times m$ orthogonal matrix whose i-th column u_i is the i-th eigenvector of the sample covariance matrix:

$$C = \frac{1}{l} \sum_{t=1}^{l} x_t x_t^T.$$

Basically, PCA firstly solves the eigenvalue problem (3.2):

$$\lambda_i u_i = C u_i, i = 1, ..., l \tag{3.2}$$

λ_i is one of the eigenvalues of C and u_i is the corresponding eigenvector. Based on the estimated u_i, the components of s_t, are then calculated as the orthogonal transformations of x_t .

$$s_t(i) = u_i^T x_t, i = 1, ..., m. \tag{3.3}$$

The new components are known as principal components. By using only the first several eigenvectors sorted in descending order of the eigenvalues, the number of principal components in s_t, can be reduced. This is the dimensional reduction characteristic of PCA. The size of the input vectors will be reduced by retaining only those components which contribute more than a specified fraction of the total variation in the data set. That means a new variable, minimum fraction variance component, should be defined. The comparison of different values of this parameter is explained in section 4.1.

3.2 Least Squares Support Vector Machines

In general, any ideal forecasting algorithm must satisfy the following criteria [20]:

1. *Non stationarity of load series:* When modeling the load series, it is important to consider the dynamic, nonlinear and complex input-output relationships that exist in the load trend.

2. *Adaptiveness of the forecasting model:* Previous researches have proved that the characteristics of load series between regular workdays and anomalous days (weekends and public holidays) are different.

3. *Robustness of the forecasting model:* A universal model is the top priority.

Kernel based estimation techniques, such as support vector machines (SVMs) and specially least squares SVMs (LS-SVMs) have shown to be powerful nonlinear classification and regression techniques and it has turned out that they can fulfill all of the above criteria perfectly [11], [12].

With the help of a kernel function, LS-SVMs perform the linear regression in the transformed space by nonlinearly mapping of the input data into a high dimensional feature space [17]. We will use x and z to denote the input vector and the feature space vector respectively, and $z = \phi(x)$.

Let the training set, x_i and y_i, consist of N data points, where x_i is the i-th input vector and y_i is the corresponding target value. The goal of LS-SVMs regression is to estimate a function that is as "close" as possible to the target values y_i for every x_i and at the same time, is as "flat" as possible for good generalization. The function f is represented using a linear function in the feature space:

$$y = f(x) = w \cdot \phi(x) + b, \tag{3.4}$$

where $\phi(x)$ is a function that maps the input space into a higher dimensional feature space. Also, b denotes the bias, as in all SVM designs, we define the kernel function, where "." represents inner product in the space.

$$k(x, \hat{x}) = \phi(x) \cdot \hat{\phi}(x)$$

This will result in the optimization problem in primal weight space:

$$\min_{w,b,e} J(w, e) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{k=1}^{N} e_k^2, \tag{3.5}$$

subject to

$$y_k = w^T \phi(x_k) + b + e_k, \quad k = 1, ..., N,$$

where w is weight vector in primal weight space and e_k is the error variable. The cost function J concludes a sum squared error and a regularization term. γ is a positive real constant that determines the penalties to estimation errors.

Next, the model in (3.5), can be computed in dual space instead of the primal space that results in Lagrangian with Lagrange multipliers $\alpha_k \in R$, called support values.

$$L(w, b, e; \alpha) = J(w, e) - \sum_{k=1}^{N} \alpha_k \left\{ w^T \varphi(x_k) + b + e_k - y_k \right\}. \tag{3.6}$$

The circumstances for optimality are as follows:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, k = 1, ..., N \\ \frac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \varphi(x_k) + b + e_k - y_k = 0, k = 1, ..., N \end{cases} \tag{3.7}$$

Based on [15], with the application of Mercers theorem, $K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$, and with a positive definite kernel function K, we can eliminate w and e_i , obtaining

$$y_j = \sum_{i=1}^{N} \alpha_i K(x_i, x_j) + b + \frac{\alpha_j}{\gamma}.$$

Building the kernel matrix $\Omega_{i,j} = K(x_i, x_j)$ and writing the equations in matrix notation will express the final system in dual form,

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x_j) + b. \tag{3.8}$$

The followings are popular kernel functions $K(x_i, x_j)$ used for SVM regression or classification problems:

- Linear kernel: $K(x_i, x_j) = x_i^T x_j$
- Polynomial kernel with degree d and tuning parameter c:

$$K(x_i, x_j) = (x_i^T x_j / c + 1)^d$$

- Radial basis function (RBF):

$$K(x_i, x_j) = \exp(-||x_i - x_j||_2^2 / \delta^2)$$

where δ is a tuning parameter.

It was observed in the literature that a nonlinear RBF kernel has been widely used than other types to fit the electricity load data and therefore in this project it is being applied in the LS-SVM algorithm. There is no doubt that the efficient performance of the LS-SVM model involves an optimal selection of the kernel parameter δ and regularization parameter γ , which can be done using one of several optimization techniques explained in the next section.

3.3 LS-SVM Hyper-parameters Optimization Algorithms

In general, in any classification or regression problem, if hyper-parameters of the model are not well selected, results will not be good enough. In this work, regularization parameter (γ) and kernel parameter (δ) of LS-SVM are called hyper-parameters, and their optimal values are of interest. There are roughly two classes of methods for LS-SVM parameter estimation (tuning):

1. *Experimental methods:* In practice, most researchers have so far used cross-validation.

2. *Theoretical methods:* Potentially, we can use global or local optimization techniques such as a genetic algorithm, simulated annealing and bayesian evidence framework.

Each of these techniques has its own pros and cons specific to the proposed application. In this section, different optimization techniques applied to the proposed LS-SVM algorithm are introduced in brief.

3.3.1 Genetic Algorithm

Genetic algorithms comprise a powerful stochastic search and optimization technique based on the processes of evolution theory. This method is reported to be suitable for a good approximate global maximum or minimum value. A genetic algorithm involves using three operators: reproduction, crossover, and mutation. The process of LS-SVM with Genetic Algorithm is illustrated in Fig. 3.1. Here, a genetic algorithm produces sets of individuals, which represent the LS-SVM parameters (γ) and (δ) [22], [29]. Each resulting LS-SVM is

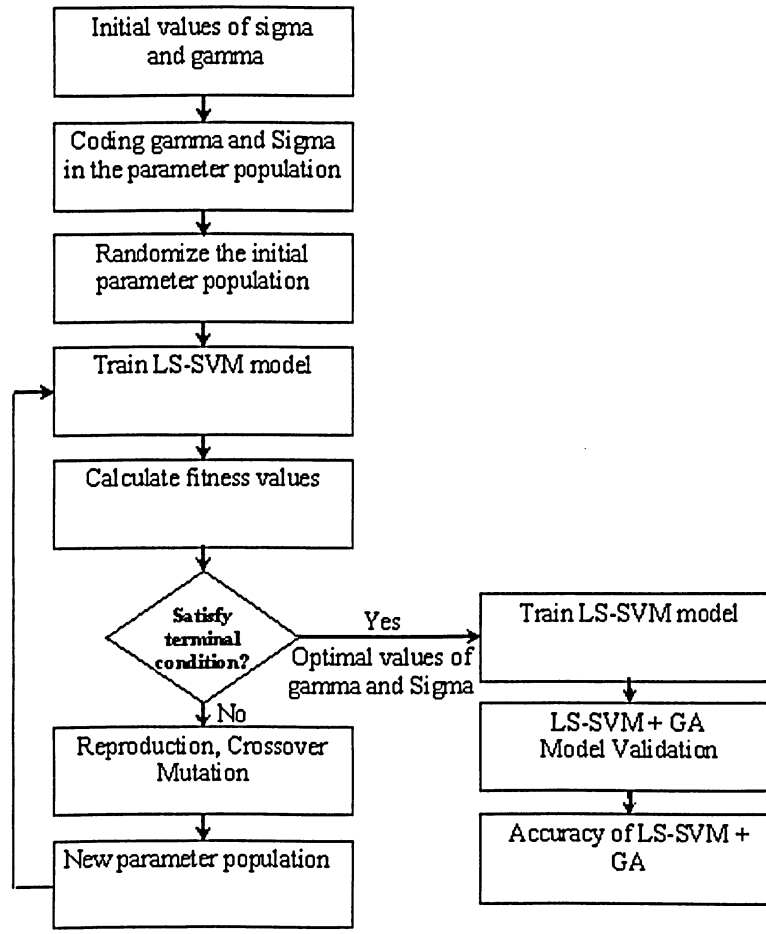

Figure 3.1: The proposed GA with LS-SVM

trained and is used to forecast the peak load demand. Parents of the next generation are selected according to a fitness function. Several measurement indicators have been proposed and used to evaluate the prediction accuracy of model such as MAPE, MAE, and maximum error in time series prediction problems. In this work, mean absolute percentage error (MAPE) is selected as fitness function. Individuals with larger fitness value have greater possibility of being selected as parents. The fitness function is defined as:

$$fitness = \frac{1}{MAPE(\sigma, \gamma)}. \tag{3.9}$$

Thus, maximizing the fitness value corresponds to minimizing the predicted error. When the termination criterion is met, the individual with the best fitness defines the optimal

parameters of the LS-SVM. We used the Matlab Genetic Algorithm Toolbox developed by C.R. Houck, J. Joines, and M. Kay, which by comparison is a really good and promising implementation of Genetic Algorithm [30].

3.3.2 Simulated Annealing

The simulated annealing algorithm is an optimization technique which simulates the annealing process of material physics [21]. Fig. 3.2 shows the general form of simulated annealing (SA). Based on the work of Boltzmann *et al.* [31] if a system is in thermal equilibrium at temperature T, then the probability $P_T(s)$ of the system being in a given state s is given by Boltzmann distribution:

$$P_T(s) = \frac{\exp(-E(s)/kT)}{\sum_{w \in S} \exp(-E(s)/kT)}, \tag{3.10}$$

where $E(s)$ represents the energy of state s, k is the Boltzmann constant and S is the set of all possible states. Metropolis *et al.* [32] developed an algorithm which simulates the process of Boltzmann. Based on this algorithm, when the system is in the original state s_{old} with energy $E(s_{old})$, a randomly selected atom is perturbed, resulting in a new state s_{new} with energy $E(s_{new})$. This new state could be either accepted or rejected depending on the Metropolis criterion which says if $E(s_{new}) < E(s_{old})$, the new state is accepted. Otherwise, if $E(s_{new}) > E(s_{old})$, then the probability of accepting the new state is given by the following probability function:

$$P(Accept\ \ s_{new}) = \exp\left(-\frac{E(s_{old}) - E(s_{new})}{kT}\right). \tag{3.11}$$

Following the work of Boltzmann and Metropolis, Kirkpatrick *et al.* [33] suggested that the Metropolis approach be conducted for each temperature on an annealing schedule until when the thermal equilibrium is accomplished. The SA algorithm in accordance with LS-SVM is described as follows:

Step 1 (Initialization): Set upper bounds of the two LS-SVM positive parameters γ and δ. Generate and feed the initial values of the two parameters into the LS-SVM model. The

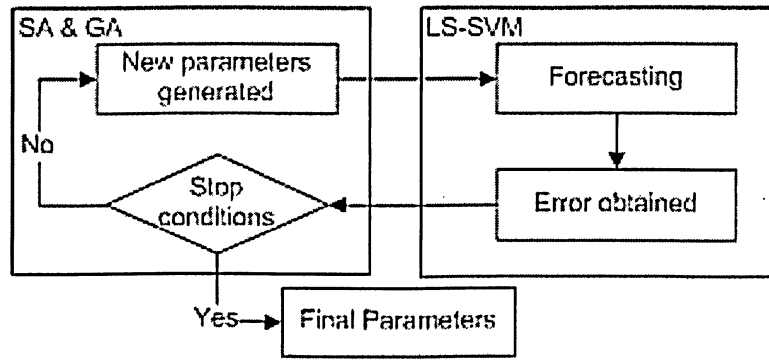

Figure 3.2: SA with LS-SVM algorithm

mean absolute percentage of forecasting error (MAPE) is defined as the system energy E. Thus, we have the energy of (E_0) as the initial state.

Step 2 (Temporary state): Make a random move to change the existing state to a temporary state. A new set of the parameters is obtained at this stage.

Step 3 (Acceptance criteria): The following conditions are employed to determine the acceptance or rejection of the temporary state:

$$\begin{aligned} &\text{Accept if } E(s_{new}) > E(s_{old}) \ \ and \ \ p < P(\text{Accept } s_{new}), \\ &0 \leq p \leq 1. \\ &\text{Also, accept if } E(s_{new}) \leq E(s_{old}). \\ &\text{Reject otherwise.} \end{aligned} \tag{3.12}$$

In (3.12), p is a random number to determine the probability of acceptance of the temporary state. If the temporary state is accepted, then, it will be set as the current state. If the temporary state is rejected, then, return to Step 2, and make another move. We may define a maximum number of repetitions N_{SA} to avoid infinite loops.

Step 4 (Temperature reduction): After the new system state is obtained, reduce the current temperature by some user-defined positive ratio. If the predetermined temperature (Stop criterion) is reached, then stop the algorithm, and the latest state is the approximate optimal solution. Otherwise, go back to Step 2.

We used the General simulated annealing algorithm developed by Joachim Vandekerck-

hove, which is found to be well-explained and very easily traced by comparison with other implementations of simulated annealing [34].

3.3.3 Bayesian Evidence Framework

The Bayesian evidence framework first introduced by Mackay [35] has been applied to the design of neural networks with great success. But, it was first applied to the standard SVM classification algorithm by Kwok [36]. Then Gestel *et al.* [37] extended its integration to the LS-SVM classifier and regression problems. This approach starts from the feature space formulation, while analytic expressions are obtained in the dual space on the different levels of bayesian inference, which yields the similar expressions of Gaussian Processes (GPs) [35]. It is known that this novel approach shows good generalization performances but with very complicated expressions for practical use. In this project, we apply the Bayesian evidence framework to the LS-SVM regression algorithm for load forecasting problem and use this practical approach to select optimal regularization parameter (γ) and optimal kernel parameter (δ). The method we are using here is quite simplified and similar to the Bayesian interpretation of standard SVM.

According to the Bayesian evidence theory, the inference is divided into three distinct levels. Training of the LS-SVM regression (i.e. support values and the bias) is interpreted as Level 1 inference. The optimal regularization parameter can be achieved as Level 2. The optimal kernel parameter selection can be performed as Level 3.

Level 1 inference

To be convenient [35], we divide optimization objective in (3.5) by γ and then replace $\frac{1}{\gamma}$ by λ. For a given value of λ, the first level of inference infers the posterior of ω by

$$p(\omega|D,\lambda,H) \propto p(D|\omega,\lambda,H)p(\omega|\lambda,H), \tag{3.13}$$

where D is the training dataset and H represents model with parameter vector ω. Assuming training data are independently identically distributed, and $p(\omega|\lambda,H)$ is the Gaussian

probability distribution, we finally will obtain

$$p(\omega|D,\lambda,H) \propto \exp\left\{-\frac{\lambda}{2}\omega^T\omega - \sum_{i=1}^{l} L(y_i, f(x_i))\right\}, \tag{3.14}$$

where x_i, y_i represent the input and output pairs, $f(x_i)$ is the LS-SVM model and finally $L(y_i, f(x_i))$ denotes the loss function. Level 1 inference, training of LS-SVM (3.5) can be interpreted as maximizing $p(\omega|D,\lambda,H)$ with respect to ω.

Level 2 inference

Applying the Bayesian rule in the second level of inference, we obtain the posterior probability of λ:

$$p(\lambda|D,H) \propto p(D|\lambda,H)p(\lambda|H) \propto p(D|\lambda,H). \tag{3.15}$$

The most possible value of λ can be determined by maximizing the posterior probability of λ as $p(\lambda|D,H)$.

Let us define $E_\omega = \omega T^\omega/2, E_D = \sum_{i=1}^{l} L(y_i, f(x_i))$ and then we will obtain

$$\ln(p(\lambda|D,H)) \propto \ln(p(D|\lambda,H)) = -\lambda E_\omega^{MP} - E_D^{MP} + \frac{k}{2}\ln\lambda - \frac{1}{2}\ln(\det A) + \text{constant.}, \tag{3.16}$$

where ω_{MP} is the most possible value of parameters ω and A is $A = \nabla^2\left(\lambda E_\omega + \sum_{i=1}^{l} L(y_i, f(x_i))\right)$.

Maximization of the log-posterior probability of $p(\lambda|D,H)$ with respect to λ leads to the most probable value of λ_{MP} obtained by the following equation

$$2\lambda_{MP}E_\omega^{MP} = \varsigma, \tag{3.17}$$

where ς is called the effective number of parameters. In the case of LS-SVM regression, use of cost function $L(y_i, f(x_i)) = \frac{1}{2}(y_i - \omega\phi(x_i) - b_i)^2$ yields

$$A = \nabla^2\left(\lambda E_\omega + \sum_{i=1}^{l} L(y_i, f(x_i))\right) = \lambda I + B, \tag{3.18}$$

where $B = \sum_{i=1}^{l} \varphi(x_i)\varphi(x_i)^T$. Denote the eigenvalues of B by ρ_i yields the effective number of parameters ς of LS-SVM as follows:

$$\varsigma = \sum_{i=1}^{N} \frac{\rho_i}{\rho_i + \lambda}, \tag{3.19}$$

where $N(N \leq 1)$ denotes the number of nonzero eigenvalues of $l \times l$ matrix $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_i), i, j = 1, 2, ...l$.

Level 3 inference

The third level of inference in the evidence framework compares the different models by examining their posterior probabilities $P(H|D) \propto P(D|H)P(H)$ and can be used to find the optimum kernel parameter. Assuming the prior probability $P(H)$ over all possible models is uniform, we have

$$P(H|D) \propto P(D|H) \propto \int P(D|\lambda, H)P(\lambda|H)d\lambda \propto P(D|\lambda_{MP}, H)/\sqrt{\cdot} \tag{3.20}$$

Therefore

$$\ln P(H|D) = -\lambda_{MP}E_\omega^{MP} - E_D^{MP} + \frac{k}{2}\ln\lambda_{MP} - \frac{1}{2}\ln(\det A) - \frac{1}{2}\ln(k - \lambda_{MP}\text{trace}A^{-1}) + \text{constant}. \tag{3.21}$$

The optimum kernel parameter can be obtained by maximizing log-posterior probabilities $\ln P(H|D)$ with respect to the kernel parameter. For practical use, the selection method of the kernel parameter δ of Gaussian kernel is illustrated in this subsection. To obtain the most possible value of the kernel parameter δ, we set the derivative of $\ln P(H|D)$ with respect to δ to zero we obtain the kernel parameter in the LS-SVM regression.

$$\frac{\partial \ln P(H|D)}{\partial \delta} = 0. \tag{3.22}$$

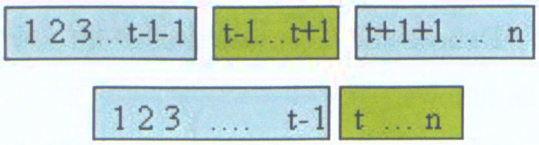

Figure 3.3: CV with LS-SVM algorithms

$$\delta = \left| \left[\frac{\lambda_{MP} \sum_{i,j=1}^{l} a_i a_j \exp\left(-\frac{(x_i - x_j)^2}{2\delta^2}\right)(x_i - x_j)^2}{\text{trace}\left(A^{-1}\left(\frac{\partial K}{\partial \delta}\right)\right) + \frac{\lambda_{MP}}{K - \lambda_{MP}\text{trace}A^{-1}}\text{trace}\left(A^{-2}\left(\frac{\partial K}{\partial \delta}\right)\right)} \right]^{1/3} \right| . \tag{3.23}$$

a_i and a_j represent the ith and jth element of A. The bayesian evidence framework in [38] is used with some modifications to best fit the load forecasting application.

3.3.4 Cross Validation

One of the most popular techniques of evaluating a set of parameter values is the use of cross-validation [39]. As shown in Fig. 3.3, in cross-validation [40], the training set T is divided up into M partitions $(T_1, T_2, ..., T_M)$. For each parameter setting, it trains the LS-SVM model M times when during each time one of M cases is held out while the remaining $(M-1)$ cases are used to train the model. Then, the trained model is used to test the held-out case. The average accuracy of these M trials is used to estimate what the generalization accuracy would be if the parameter value was used. The parameter value that yields the highest estimated accuracy, i.e. the least generalization error is, then, chosen. When more than one parameter needs to be tuned, the combined settings of all of the parameters can be measured using cross-validation in the same way. When M is equal to the number of training samples in T, the result is leave-one-out cross-validation (LOO-CV), in which each instance i is tested by all of the instances in T used for training except for i itself, so that almost all of the training data is available for each regression attempt. LOO-CV has been described as being desirable but computationally expensive.

Chapter 4

Results

4.1 Implementation and Evaluation

In this chapter, we explain the PCA-based LS-SVM approach used to forecast the week-ahead peak load demand:

Step 1: Preprocessing the historical load data sets (e.g., removing the abnormal samples of $13th$ and $14th$, August 2003 Blackout) and then normalizing all sample sets to zero mean and unit variance. As it was observed in the literature, there is no agreement on the use of target day's temperature for forecasting purposes, because the temperature itself is essentially a prediction and this will diminish forecasting accuracy. But, in this project, we observed that even by removing the target days temperature information, we would get almost the same results, since we are using also the data from the previous two weeks on the same day.

Step 2: Implementing PCA on the input data and based on trial-and-error, to determine the appropriate minimum fraction variance component, i.e., number of features to be entered into the LS-SVM model.

Step 3: Building the the target equation (3.8) and using the test data sets to predict the next seven day's maximum load demands. When training a LS-SVM model, there are some parameters to be selected. They would influence the performance of the model significantly.

- Regularization parameter (γ)
- Kernel bandwidth parameter (δ)

Table 4.1: LS-SVM with GA Optimization Training Parameters

Parameter	Value
Problem Type	Minimum
Population Size	10
Generations	20 - 500
Gamma Range	0 - 1000
Sigma Range	0 - 1000
Selection Method	Tournament
Mutation Method	Uniform

- The kernel function $k(x, \hat{x})$, RBF is used in our LS-SVM model.
- The size of training data sets, i.e. how many previous days are included for one training data.

Step 4: Applying the optimization algorithms to the LS-SVM regression and keep estimating the optimal hyper-parameters until enough accuracy is reached.

In cross validation the number of subsets M in the training data set needs to be selected. Empirically, the *10-fold* cross validation is the one most commonly used for both regression and classification purposes [39], [40]. The parameter value γ and δ that yields the minimum generalization error is, then, chosen.

For genetic algorithm, some parameters have to be determined in advance before using LS-SVM model. For instance, population size, range of parameters, selection and mutation operators have to be selected correctly. The values of individual parameters and the fitness value of the fitness function were based on prior experiences of training and on problem type. In Table 4.1 some of these parameters are shown.

Bayesian evidence framework does not need any initialization of parameters, and that is a very powerful and interesting aspect of the bayesian inference.

Simulated annealing however, requires more parameters to be determined such as cooling schedule, suitable initial and stop temperatures, annealing scheme and termination condition.

Table 4.2: LS-SVM with SA Hyper-Parameter Optimization

Level 1: Inference of Model Parameters	Infer ω_{MP} and b_{MP} or α and b_{MP}
Level 2: Inference of Hyper-Parameters	Infer σ_{MP}
Level 3: Inference of Model Parameters	Infer the Kernel Parameter γ_{MP}

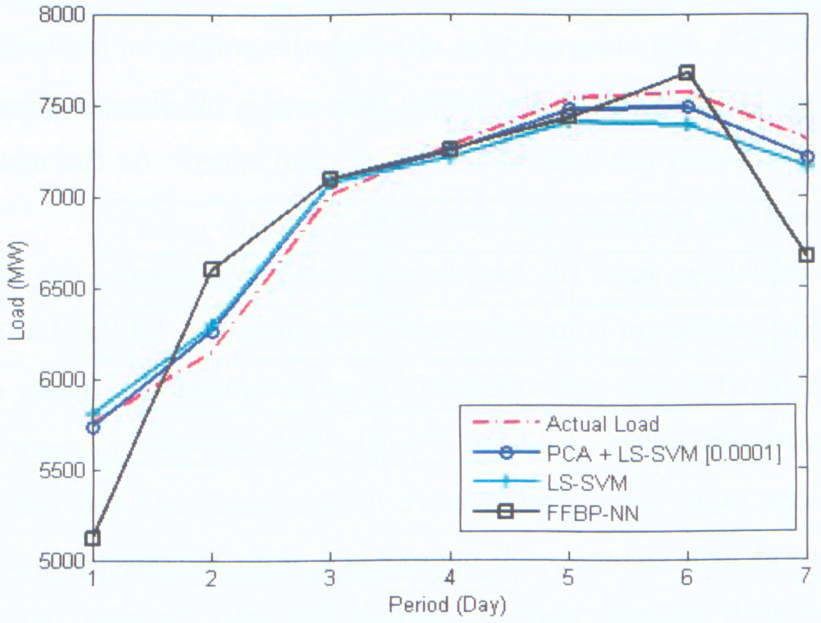

Figure 4.1: Actual vs. Predicted Load (LS-SVM with PCA for feature extraction)

Another important variable in both GA and SA is the maximum number of iterations. All of these parameters should be selected properly by an expert and they have significant influence on final result. Since this method is completely expert-dependent, it cannot be the algorithm of choice for our LS-SVM load forecasting model.

The comparison of different forecasting models with and without PCA is shown in Table 4.3 and Fig. 4.1. For instance, the 0.0001 for PCA + LS-SVM means those components that contribute more than one percent to the variance in the data set are kept and the remaining is eliminated. LS-SVM with PCA and minimum fraction variance of 1% obtained better results than LS-SVM without feature extraction and feed forward back propagation neural

networks. In this case the number of features decreased to 34 from the original 85. It can be observed from the Table 4.3 that this the PCA feature extraction resulted in better accuracy (MAPE) and also in faster speed. Since, the number of features is decreasing, therefore the amount of computations and calcualations will be reduced significantly and as a result, the algorithm will be faster. Also, it was found that using smaller minimum fraction variance improves MAPE and the model performance further. For example we used .1% minimum fraction variance of PCA and observed that although the number of features is growing, but the performance got better and the processing time was a bit slower. Generally speaking, there is a trade-off between the number of features and speed. As the number of features increasing, the speed will be slower and vice versa.

Here, we will explain in brief the feed forward back propagation neural networks used in this work for the purpose of comparison with our proposed LS-SVM with PCA feature extraction. An Artificial Neural Network (ANN) can be defined as a highly connected array of elementary processors called neurons. The the multi-layered perceptron (MLP) type ANN consists of one input layer, one or more hidden layers and one output layer. Each layer employs several neurons and each neuron in a layer is connected to the neurons in the adjacent layer with different weights. Signals flow into the input layer, pass through the hidden layers, and arrive at the output layer. With the exception of the input layer, each neuron receives signals from the neurons of the previous layer linearly weighted by interconnecting values between neurons. The neuron then produces its output signal by passing the summed signal through a sigmoid function [41] , [42].

A complete set of training data are assumed to be available. Inputs are imposed on the top layer. The ANN is trained to respond to the corresponding target vectors on the bottom layer. The training continues until a certain stop-criterion is satisfied. Typically, training is halted when the average error between the desired and actual outputs of the neural network over the training data sets is less than a predetermined threshold. The topology of the ANN for the peak load forecasting in our work consists of 5 hidden neurons in the hidden layer and we used the MATLAB Neural Network Toolbox which is a promising and powerful tool

Table 4.3: Forecasting errors for a typical week (LS-SVM with PCA for feature extraction)

Estimation Technique	No. features	MAPE (%)	MAE (MW)
Single LS-SVM	85	1.5755	110.3582
FFBP Neural Network	85	2.9231	203.1979
PCA + LS-SVM [0.0001]	34	1.1454	80.0693
PCA + LS-SVM [0.00001]	58	0.8082	56.1873

Table 4.4: Forecast errors for a typical Week (LS-SVM with hyper-parameter optimization techniques)

Tuning Technique	MAPE (%)	MAE (MW)
LS-SVM + CV	2.9231	203.1979
LS-SVM + GA	2.1454	142.0693
LS-SVM + SA	1.9082	125.1873
LS-SVM + BI	1.3057	86.5239

for any kind of artificial intelligence application [43].

The comparison of different hyper-parameter tuning techniques for a typical week is shown in Table 4.4 and Fig. 4.2. It can be observed that LS-SVM with Bayesian framework optimization outperformed other methods including simulated annealing, genetic algorithm and cross validation in case of yielding better performance and accuracy. The corresponding programs were developed from LS-SVMlab [44], with major changes to conform to the application. All the discussed algorithms and models were implemented and tested with the same processor.

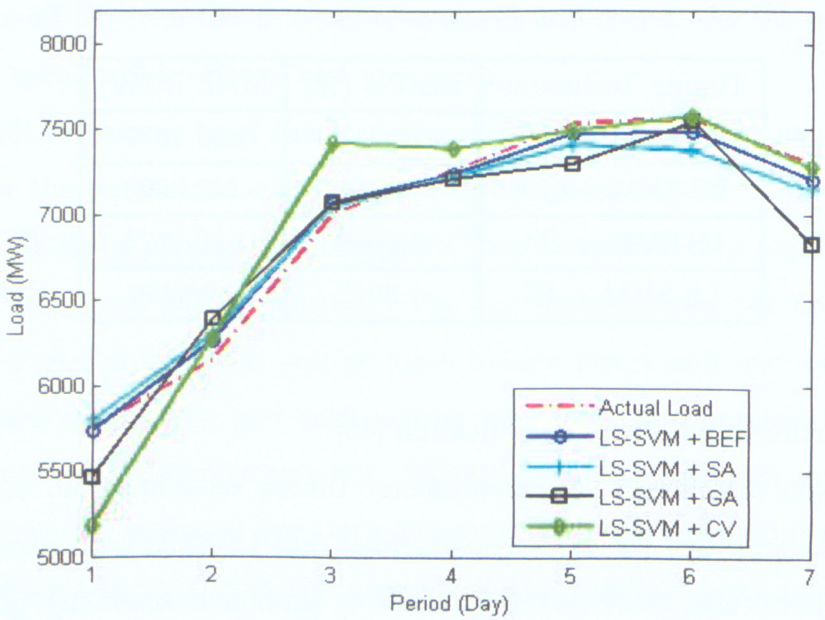

Figure 4.2: Actual vs. Predicted Load (LS-SVM with hyper-parameter optimization techniques)

Chapter 5

Conclusions

5.1 Conclusions

In this project, a PCA-based least squares support vector machine was presented and its performance was evaluated through a simulation study. As it was shown, the PCA was used to reduce the input variable dimension. A wide range of the minimum fraction variance were tested on the model and the results were satisfactory. Also, a crucial and effective feature was added to the data collection namely daylight time. Depending on the region, this feature could be varying tremendously and as a result it has to be taken into account.

LS-SVM by feature extraction using PCA outperformed other techniques in week-ahead load forecast including LS-SVM without feature extraction and the well-known feed forward back propagation neural networks. It showed better accuracy, faster speed and superb generalization.

Furthermore, various optimization methods for tuning the least squares support vector machine hyper-parameters were presented and performance was evaluated through a simulation study. The LS-SVM technique shows satisfactory performance, such as powerful regression ability, acceptable predicting accuracy and perfect foundations in theory. However, the efficient performance of the LS-SVM model depends on the optimum choice of the kernel and regularization parameters, which can be done using different optimization techniques. As a result, available parameter tuning techniques were applied into the LS-SVM regression model. It was observed that LS-SVM with bayesian framework outperformed

other tuning techniques in week-ahead load forecasting problem including LS-SVM with genetic algorithm, simulated annealing and cross validation. It showed better accuracy, faster speed and superb generalization.

5.2 Discussion

In general, simulated annealing (SA) and genetic algorithm (GA) are often viewed as different techniques, but in reality, it seems that they are more closely related than it is commonly thought. There is no concrete proof which one is more accurate or faster, since this issue is problem-dependent. Both SA and GA are stochastic, flexible and less likely to get trapped in local minima. On the other hand, they are both slow and their good movement is to some extent non-intuitive for a given task. As related to our application, it was observed that they have almost the same performance and close speed. Perhaps the Bayesian framework could be introduced as the most reliable, powerful and accurate algorithm which does not need any initialization of parameters. This last factor may significantly reduce the losses over the user-defined variables. Eventually, cross validation seemed to be the simplest, slowest and most popular. Although, it is slow and less accurate, but its computational simplicity has proved it as the first choice of try in most regression and classification problems.

5.3 Future Research

There is still additional research required to explore if the algorithms can yield in better performance. For instance, there are other advanced models of LS-SVM like Fixed-Size Support Vector Machines which is reported not only have identical performance as SVM and LS-SVM, but also can improve the computational complexity and training time tremendously. Also, there are a wide variety of LS-SVM implementations available in the literature and every one of them depending on the type of application can have end up in better results. So, there seems to be a really good potential for examining different implementations of SVMs on our load forecasting application.

In this project, for feature extraction, PCA was used and it resulted in good performance. But, it has been observed that Kernel Principal Component Analysis (KPCA) and Independent Component Analysis (ICA) have better generalization performance than PCA feature extraction [28]. Unlike PCA which linearly transforms the original inputs into uncorrelated features, KPCA is a nonlinear PCA developed by using the kernel method. In ICA, the original inputs are linearly transformed into statistically independent features.

Bibliography

[1] G. Gross and F. Galiana, "Short term load forecasting," *Proc. IEEE*, vol. 75, no. 12, pp. 1558–1573, 1987.

[2] R. Sadownik and E. Barbosa, "Short-term forecasting of industrial electricity consumption in brazil," *Int. J. Forecast.*, vol. 18, pp. 215–224, 1999.

[3] K. Kim, H. S. Youn, and Y. C. Kang, "Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference method," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 559–565, 2000.

[4] S. Tzafestas and E. Tzafestas, "Computational intelligence techniques for short-term electric load forecasting," *Journal of Intelligent and Robotic Systems*, vol. 31, p. 768, 2001.

[5] A. Lotufo and C. Minussi, "Electric power systems load forecasting: A survey," *Int. J. Forecast.*, vol. 18, pp. 215–224, 1999.

[6] D. Srinivasan and M. Lee, "Survey of hybrid fuzzy neural approaches to electric load forecasting," *IEE Proceedings*, vol. 5, no. 4004-4008, 1998.

[7] H. S. Hippert, C. E. Pedreira, and R. Castro, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, 2001.

[8] D. Srinivasan, A. Liew, and C. Chang, "Forecasting daily load curves using a hybrid fuzzy-neural approach," *IEE Proc.*, vol. 141, no. 6, 1994.

[9] D. Srinivasan, S. S. Tan, C. S. Chang, and E. K. Chan, "Practical implementation of a hybrid fuzzy neural network for one-day-a head load forecasting," *IEE Proceedings*, vol. 145, no. 6, 1998.

[10] B. Chen, M. Chang, and C. Lin, "Load forecasting using support vector machines: A study on eunite competition 2001," *IEEE Trans. Power System*, vol. 19, no. 4, pp. 1821–1830, 2004.

[11] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel Based Learning Methods.* Cambridge University Press, 2000.

[12] A. Smola and B. Scholkopf, "On a kernel-based method for pattern recognition, regression, approximation and operator inversion," *Algorithmica*, vol. 22, pp. 211–231, 1998.

[13] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[14] J. Suykens, "Least squares support vector machines for classification and nonlinear modeling," *Neural Network World*, vol. 10, pp. 29–48, 2000.

[15] V. Vapnik, *Statistical Learning Theory.* Wiley, 1998.

[16] B. Scholkopf, C. Burges, and A. Smola, *Advances in Kernel Methods - Support Vector Learning.* MIT Press, 1999.

[17] Y. Li, T. Fang, and E. Yu, "Short-term electrical load forecasting using least square support vector machines," *Proceedings. PowerCon*, vol. 1, pp. 230–233, 2002.

[18] J. Cao, K. Chua, and L. Guan, "Combining kpca with support vector machine for time series forecasting," *CIFEr Hong Kong*, 2003.

[19] M. Afshin and A. Sadeghian, "Pca-based least squares support vector machines in week-ahead load forecasting," *In Press. Accepted in IEEE ICPS Conference*, 2007.

[20] S. Fan and L. Chen, "Short-term load forecasting based on an adaptive hybrid method," *IEEE Trans. PowerSystem*, vol. 21, no. 1, pp. 392–401, 2006.

[21] P. Pai and W. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669–2688, 2005.

[22] C. Hsu, C. Wu, S. Chen, and K. Peng, "Dynamically optimizing parameters in support vector regression: An application of electricity load forecasting," *HICSS '06*, vol. 2, pp. 30c – 30c, 2006.

[23] M. Afshin, A. Sadeghian, and K. Raahemifar, "On efficient tuning of ls-svm hyper-parameters in short-term load forecasting: A comparative study," *In Press. Accepted in IEEE General Meeting*, 2007.

[24] Independent electricity system operator. [Online]. Available: http://www.ieso.ca.

[25] Environment canada. [Online]. Available: http://www.ec.gc.ca/

[26] "Methodology to perform long term assessments," Public, IESO, 2006.

[27] F. Tay and L. Cao, "Saliency analysis of support vector machines for feature selection," *Neural Network World*, vol. 2, no. 1, pp. 153–166, 2001.

[28] L. Cao and W. Chong, "Feature extraction in support vector machine: A comparison of pca, kpca and ica," *Proceedings of the 9th International Conference on Neural Information Processing*, vol. 2, pp. 1001–1005, 2002.

[29] X. Wang, H. Zhang, C. Zhang, X. Cai, J. Wang, and J. Wang;, "Prediction of chaotic time series using ls-svm with automatic parameter selection," *PDCAT 2005*, pp. 962–965, 2005.

[30] C. Houck, J. Joines, and M. Kay, "A Genetic Algorithm for Function Optimization: A Matlab Implementation," North Carolina State University, Raleigh, NC, Tech. Rep. NCSU-IE-TR-95-09, 1995.

[31] C. Cercignani, "The boltzmann equation and its applications," *Springer-Verlag*, 1988.

[32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller, "Equations of state calculations by fast computing machines," *Chem Phys*, vol. 21, pp. 1087–1092.

[33] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 4598, 1983.

[34] J. Vandekerckhove, "General simulated annealing algorithm," Tech. Rep., 2006. [Online]. Available: http://www.mathworks.com/matlabcentral/files/10548/anneal.m

[35] X. Wen, Y. Zhang, W. Yan, and X. Xu, "Nonlinear decoupling controller design based on least squares support vector regression," *Journal of Zhejiang University SCIENCE*, 2005.

[36] J. Kwok, "The evidence framework applied to support vector machines," *IEEE Transaction on Neural Network,*, vol. 11, pp. 1162–1173, 2000.

[37] J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Transaction on Neural Network,*, vol. 12, pp. 809–821, 2001.

[38] T. V. Gestel, J. Suykens, D. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. D. Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," pp. 809 – 821, 2001.

[39] C. Schaffer, "Selecting a classification method by cross-validation," *Machine Learning*, vol. 13, pp. 135–143, 1993.

[40] D. Wilson and T. Martinez, "Combining cross-validation and confidence to measure fitness," *IJCNN 99*, vol. 2, pp. 1409–1414, 1999.

[41] D. Park, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Trans. Power Syst.*, vol. 6, no. 2, pp. 442–449, 1991.

[42] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, and A. Azeem, "Practical experiences with an adaptive neural network short-term load forecasting system," *IEEE Trans. Power Syst.*, vol. 10, no. 1, pp. 254–265, 1995.

[43] A. Khotanzad, M. Davis, and A. Abaye, "An artificial neural network hourly temperature forecaster with applications in load forecasting," *IEEE Trans. Power Syst.*, vol. 11, no. 2, pp. 870–876, 1996.

[44] J. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle. (2002) Ls-svmlab: a matlab/c toolbox for least squares support vector machines. [Online]. Available: http://www.esat.kuleuven.ac.be/sista/lssvmlab.