

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

**ELECTRICAL LOAD DEMAND FORECASTING APPLICATION USING
SUPPORT VECTOR MACHINES**

**M.Sc. Thesis by
Dilara DEMREN**

Department : Electrical Engineering

Programme : Electrical Engineering

JUNE 2011

**ELECTRICAL LOAD DEMAND FORECASTING APPLICATION USING
SUPPORT VECTOR MACHINES**

**M.Sc. Thesis by
Dilara DEMREN
(504061028)**

**Date of submission : 06 May 2011
Date of defence examination: 08 June 2011**

**Supervisor (Chairman) : Assoc. Prof. Belgin TÜRKEY (ITU)
Members of the Examining Committee : Prof. Dr. İbrahim EKSİN (ITU)
Assoc. Prof. Mustafa BAĞRIYANIK
(ITU)**

JUNE 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DESTEK VEKTÖR MAKİNELERİ İLE ELEKTRİKSEL YÜK TALEP
TAHMİNİ UYGULAMASI**

**YÜKSEK LİSANS TEZİ
Dilara DEMREN
(504061028)**

Tezin Enstitüye Verildiği Tarih : 06 Mayıs 2011

Tezin Savunulduğu Tarih : 08 Haziran 2011

**Tez Danışmanı : Doç. Dr. Belgin TÜRKEY (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. İbrahim EKSİN (İTÜ)
Doç. Dr. Mustafa BAĞRIYANIK (İTÜ)**

HAZİRAN 2011

FOREWORD

I would like to express my gratitude to my thesis supervisor, Assoc. Prof. Belgin TRKAY. Without her support, this thesis would not have been completed. Her advices and feedbacks were the most important factors that helped me continue my studies.

I am grateful to Berkay Bulut for all his help in documentation and bearing me throughout this thesis.

I would like to thank Suna Bolat Sert with her technical guidance and suggestions.

I would like to express my special thanks to Teresa Quain for helping me in editing this thesis.

I also would like to thank Zeynep Glin Ko, Nagehan Fatma Gek, Sezen Yldırım, Duygu Bayram and Lale Erdem for their endless support.

Finally, I must thank my family for their encouragement and support of all kinds.

May 2011

Dilara DEMREN

Electrical Engineer

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	v
TABLE OF CONTENTS	vii
ABBREVIATIONS	ix
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF SYMBOLS.....	xv
SUMMARY.....	xvii
ÖZET	xix
1. INTRODUCTION	1
2. MACHINE LEARNING.....	5
2.1 Machine Learning Discipline.....	5
2.2 Machine Learning Methods	6
2.2.1 Supervised Learning.....	6
2.2.2 Unsupervised Learning.....	7
2.2.3 Semi-supervised Learning.....	8
2.2.4 Reinforcement Learning.....	9
2.3 Statistical Learning Theory.....	9
3. SUPPORT VECTOR MACHINES	13
3.1 Historical Overview of Support Vector Machines	13
3.2 Support Vector Machines in Regression	14
3.3 Kernel Induced Feature Space.....	19
4. ELECTRICAL LOAD DEMAND FORECASTING.....	23
4.1 Planning in Generation/ Transmission/ Distribution	23
4.2 Factors in Load Demand.....	27
4.3 Load Forecasting Approaches	28
4.3.1 Classical Approaches.....	28
4.3.2 Computational Intelligence Methods	29
5. ELECTRICAL LOAD DEMAND FORECASTING APPLICATION.....	33
5.1 Data Sets	34
5.2 Implementation	39
5.2.1 Support Vector Machine Implementation using Libsvm	40
5.2.2 Artificial Neural Networks Implementation using MATLAB	41
5.3 Results.....	43
6. CONCLUSION	49
REFERENCES	53
APPENDICES	57

ABBREVIATIONS

AI	: Artificial Intelligence
ANN	: Artificial Neural Networks
AR	: Autoregressive
ARIMA	: Autoregressive Integrated Moving Average
ARIMAX	: Autoregressive Integrated Moving Average with Exogenous Variables
ARMA	: Autoregressive Moving Average
ARMAX	: Autoregressive Moving Average with Exogenous Variables
EP	: Electricity Price
ERM	: Empirical Risk Minimization
GDP	: Gross Domestic Product
KKT	: Karush-Kuhn-Tucker
OCR	: Optical Character Recognition
SOM	: Self Organizing Maps
SRM	: Structural Risk Minimization
SVM	: Support Vector Machines
SVR	: Support Vector Regression
TUIK	: Turkish Statistical Institute (Türkiye İstatistik Kurumu)
VC	: Vapnik-Chervonenkis

LIST OF TABLES

	<u>Page</u>
Table 5.1 : Electricity Price in TL/ kWh	36
Table 5.2 : C , ε , γ values used in models	41
Table 5.3 : MAPE of SVM and ANN models using different data preparation.....	43

LIST OF FIGURES

	<u>Page</u>
Figure 2.1: VC dimension illustration.....	11
Figure 2.2: Diagram of SRM.....	12
Figure 3.1: The parameters used in (one dimensional) support vector regression....	16
Figure 3.2: Kernel transformation.....	19
Figure 4.1: The relation between Gross Domestic Product per Capita and Electrical Energy Consumption per Capita.....	24
Figure 4.2: System Planning in Electrical Industry.....	25
Figure 4.3: Factors that affect load demand.....	27
Figure 4.4: A multilayer feed-forward neural network architecture	30
Figure 5.1: Istanbul European Region daily peak load demand.....	35
Figure 5.2: Average daily temperature for Istanbul	36
Figure 5.3: Daily peak load demand curve from April 2006	37
Figure 5.4: Daily peak load demand curve from April 2007	38
Figure 5.5: Daily peak load demand curve from April 2008	38
Figure 5.6: Daily peak load demand curve from April 2009	39
Figure 5.7: Flow chart of the forecasting application	40
Figure 5.8: Sample artificial neural network diagram in MATLAB.....	42
Figure 5.9: Comparison graph between real and predicted values with past load, temperature and EP	44
Figure 5.10: Comparison graph between real and predicted values with past load and EP	44
Figure 5.11: Comparison graph between real and predicted values with past load values.....	45
Figure 5.12: Error analysis for the first data set.....	46
Figure 5.13: Error analysis for the second data set	46
Figure 5.14: Error analysis for the third data set.....	47

LIST OF SYMBOLS

b	: Bias term
C	: The cost of error
ε	: Width of the tube
γ	: Gamma value in radial basis function
h	: VC dimension
$R(\alpha)$	: Risk functional
$R_{emp}(\alpha)$	: Empirical risk functional
$\mathbf{w}$	: Weight vector

ELECTRICAL LOAD DEMAND FORECASTING APPLICATION USING SUPPORT VECTOR MACHINES

SUMMARY

Load forecasting is a very critical issue for the operational planning of electric power systems. Today, the electrical industry requires forecasts in short terms (hours, days or a month in advance), mid-terms (from a month to a year) and long terms (more than a year in advance). Among these, mid-term forecasts plays a vital role in defining the pricing policies, scheduling maintenances, organizing fuel supply and determining the power plants which can be used in urgent conditions.

In this study, Support Vector Machines, one of the machine learning techniques based on Statistical Learning Theory, is used in application for mid-term electrical load forecasting in Istanbul European Side. Three data sets are constructed, first one with the past daily peak load data, average temperature, calendar days and electricity price data of 2006 through to 2009, second one including all but temperature values and the third one is with only calendar days and past load values. Using these three data sets, models are obtained. With those models, April 2010's maximum daily peak load demand is predicted.

The same three data sets are used with Artificial Neural Networks and the resulting April 2010's peak load values are compared with SVM and real values.

It is concluded that Support Vector Machine algorithm is superior in all data sets to Artificial Neural Networks and is rather suitable for mid-term electrical load demand forecasting applications.

DESTEK VEKTÖR MAKİNALARI İLE ELEKTRİKSEL YÜK TALEP TAHMİNİ UYGULAMASI

ÖZET

Elektriksel güç sistemlerinin işleyişlerinin planlanmasında yük tahmini oldukça kritik bir sorundur. Günümüzde elektrik endüstrisi kısa dönemde (saatler, günler ya da bir ay öncesinden) orta dönemde (bir aydan bir yıla kadar olan süre öncesinden) ve uzun dönemde (bir yıldan fazla süre öncesinden) yapılan tahminlere ihtiyaç duymaktadır. Bunların arasından orta dönem tahminler özellikle fiyat politikalarının belirlenmesinde, bakım onarım çalışmalarının programlanmasında, yakıt tedarikinin organizasyonunda ve acil durumlarda kullanılacak ünitelerin belirlenmesinde önemli rol oynamaktadır.

Bu çalışmada, İstanbul Avrupa Yakası orta dönem yük tahmini, İstatistiksel Öğrenme Teorisinden ortaya çıkan ve bir makine öğrenmesi dalı olan Destek Vektör Makinaları ile yapılmıştır. Bu amaçla üç adet veri kümesi oluşturulmuştur. Bunlardan ilki 2006 yılından 2009 yılına kadar olan geçmiş yük verileri, ortalama hava sıcaklığı, takvim günleri ve elektrik fiyatı ile oluşturulurken, ikincisi yine aynı değerlerle ancak sıcaklık verileri çıkarılarak kurulmuştur. Üçüncü veri kümesinde ise yalnızca takvim günleri ve geçmiş yük değerleri kullanılmıştır. Bu üç veri kümesi kullanılarak modeller oluşturulmuştur. Elde edilen modeller ile de 2010 yılı Nisan ayının günlük puant yük değerleri tahmin edilmiştir.

Aynı üç veri seti Destek Vektör Makinelerinde kullanıldığı şekilde Yapay Sinir Ağları ile de kullanılmıştır. Bulunan 2010 Nisan ayı değerleri Destek Vektör Makineleri ile bulunan sonuçlarla ve gerçek değerler ile karşılaştırılmıştır.

Bu çalışmanın sonucunda Destek Vektör Makinaları algoritmasının Yapay Sinir Ağlarına oranla daha üstün ve orta dönem yük talep tahmini için uygun olduğu ortaya çıkmıştır.

1 INTRODUCTION

Technology plays an important role in life today. It evolves rapidly and the resulting products become irreplaceable even in our daily lives. The main input for technology is without a doubt the energy which is generated by a wide variety of resources.

Energy, which is the criterion for economical and social development, has gained more importance all around the world, especially since the 1970's. Accordingly energy management and planning has become more essential in developing countries, especially in the ones who have deregulated energy markets.

The main goal of the electrical energy supplier is to provide economical and reliable energy to the consumers. The reliable energy means the electricity with constant voltage, frequency and without outages. The necessity of this reliability in electrical energy forces the research and investments in this area in order to have more economic, flexible planning for the future. Therefore, to create feasible plans, the first and one of the most important step is to have accurate load demand forecasts.

Load demand forecast basically depends on the analysis of the past, current load, and the factors that affect the load. Based on this analysis, an assumption can be made. The factors that affect the load demand course can be economical, demographical, geographical, etc. The factors which will be used as input data in load forecast is determined by the forecasting term, the methodology and, naturally, the availability of the data which will be used in the forecasting method. [1]

The importance of load forecasting makes it an attractive research field among the electrical industry. With this popularity, the number of methods used has increased accordingly. Techniques used for electrical load demand forecasts should correlate the load and the factors that affect the load, such as meteorological conditions, economic and demographic changes, etc.. For this purpose, today statistical and computational intelligence methods are widely used in this field. Alternatively, different methods can be combined in order to have good forecasting results.

In the literature, many approaches can be found in electrical load forecasting field. For instance, Hor et al. [2] proposed a regression model and analyzed the impact of meteorological conditions on the load demand for England and Wales. Their aim was to provide an accurate model for a long-term prediction of the monthly demand. Their obtained models include weather conditions and the socio-economic factors, such as gross domestic product. Bruhns et al. [3] presented a non-linear regression model for medium-term load forecasting with the past load and temperature values of France. An hourly prediction of the load is made by the obtained model. In another study, Amjady [4] uses Autoregressive Integrated Moving Average (ARIMA) with past load values of the Iran's power network and temperature values, and then uses the model to predict the hourly load demand and the daily peak loads of this system.

In addition to the classical approaches, computational techniques are commonly used. For example, Salama et al. [5] used artificial neural networks for load prediction of Egypt with respect to one year and five years plan. In this medium-term load forecasting application, several models are constructed for predicting the load demand and energy sales. Inputs consist of many parameters such as past annual loads, population, oil price, gross domestic product, energy sales. Bakirtzis et al. [6] developed an artificial neural networks based short-term load forecasting model for the Energy Control Center of the Greek Public Power Corporation. For constructing the model, they used feed-forward multi-layer network architecture with back-propagation algorithm. Input variables include historical hourly load data, temperature, and the day of the week. The model is used for forecasting load demand from one to seven days. In another study, Yalçınöz et al. [7] proposed an artificial neural networks model for predicting monthly energy consumption of Niğde region in Turkey. Here, only past load values are used for constructing the model.

Hybrid approaches are also very common. As an example, Aquino et al. [8] combined genetic algorithm with artificial neural networks to realize the hourly load forecasting based on the monthly total load consumption of the year 2005 registered by the Energy Company of Pernambuco in Brazil. Kim et al. [9] presented a hybrid model that forecasts the load demand for special days in Korea. These days include public holidays, consecutive holidays, and days preceding and following holidays. The proposed method uses a hybrid approach of ANN based technique and fuzzy inference method to forecast the hourly loads of special days.

Since the area is quite popular, there are numerous surveys about forecasting methods in the literature; such as Hahn et al. [10], Campell and Adamson. [11], Feinberg and Genethliou [12], Alfares and Nazeeruddin [13], Gross and Galiana [14] Kyriakides and Polycarpou in [15], Tzafestas and Tzafestas in [16] and Hippert et al. [17]. In the surveys [10-13] the general approaches used in electrical load forecasting are explained. On the other hand, Gross and Galiana, Tzafestas and Tzafestas, Kyriakides and Polycarpou explained only short-term load forecasting approaches, whereas Hippert et al. discussed specifically artificial neural networks models for short-term load forecasting.

The focus of this study is, Support Vector Machines, one of the machine learning techniques, application in mid-term electrical load forecasting. SVM is based on the statistical learning theory, Vapnik- Chervonenkis theory. It was developed in the 1990's and is used for machine learning tasks such as pattern recognition, classification, and regression analysis. The motivation of use SVM in this study is the good convergence to the optimum solution ability of this technique in nonlinear problems without depending on a model.

The outline of this study is as follows; in the second chapter machine learning discipline is introduced and some of the main machine learning techniques are briefly explained. Afterwards, statistical learning theory which is the underlying idea of SVM is overviewed.

The third chapter explains the support vector machines in regression case. Then, commonly used kernel functions are summarized in kernel-induced feature space section.

The fourth chapter summarizes electrical load demand forecasting and its importance. Also, commonly used methods are explained in this section.

The fifth chapter is about the application of SVM to the electrical load demand forecasting problem. The results are compared with the ones that are found with ANN and real values of the application region in related time.

In the last chapter, chapter six, the conclusion of the study is given.

2 MACHINE LEARNING

Machine learning is the study based on designing algorithms that allows computers to evolve or imitate behaviour according to a given set of data. This data represents a set of relations between the variables of the data. Here, the main objective of the learner is to find out and capture this unknown relationship.

In this chapter, first the machine learning discipline is overviewed. Next, the most common and well-known algorithms are briefly explained and afterwards, the statistical learning theory is described.

2.1 Machine Learning Discipline

Machine learning is a very active sub-field of artificial intelligence concerned with the development of computational models of learning. Machine learning is inspired by the work in several disciplines: computer science, statistics, information theory, control theory, philosophy, and biology.

There are several reasons why machine learning is important. Apart from the fact that the achievement of learning in machines might help us understand how animals and humans learn, there are important engineering reasons as well. Some of these are [18]:

- Some tasks cannot be defined well unless samples are provided; that is, we might be able to specify input/output pairs but not a concise relationship between inputs and desired outputs. We would like machines to be able to adjust their internal structure to produce correct outputs for a large number of sample inputs and thus suitably constrain their input/output function to approximate the relationship implicit in the examples.
- It is possible that hidden among large piles of data are important relationships and correlations. Machine learning methods can often be used to extract these relationships (data mining).

- Human designers often produce machines that do not work as well as desired in the environments in which they are used. In fact, certain characteristics of the working environment might not be completely known at design time. Machine learning methods can be used for on-the-job improvement of existing machine designs.
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans. Machines that learn this knowledge gradually might be able to capture more of it than humans would want to write down.
- Environments change over time. Machines that can adapt to a changing environment would reduce the need for constant redesign.
- Humans are constantly discovering new knowledge about tasks. Vocabulary changes. There is a constant stream of new events in the world. Continuing redesign of AI systems to conform to new knowledge is impractical, but machine learning methods might be able to track much of it.

2.2 Machine Learning Methods

The systems that learn through data require a learning method, a strategy which defines the learning algorithm. Commonly used strategies are as follows;

2.2.1 Supervised Learning

Using computers to solve practical problems usually depends on defining the output as a function of the input set. The job of the system designer and programmer is to translate or program the function in a way the computer may process [19].

Computers; when used to solve more complex problems; may come across problems with no known simple functions, or problems with functions that require extensive research and funding to solve. An example of this type of problem may be modelling chemical reactions with unknown endings or classification of protein series with regard to DNA structures they may create. These problems may not be able to be solved with classical approaches due to the fact that the system designer cannot find a definite way to calculate the outputs using the input data. An alternative strategy to solve these kinds of problems may lie in the process of teaching the computer to analyze and learn the input-output relationship by itself. Imagine trying to teach a

group of children about sport cars or alternatively showing them sports cars in a group of other cars [19]. This type of learning may be used for classifying hand writings, forecasting for stock market prices, weather forecasts and classification of news in news agencies [20].

The classification of the sports car problem is a basic binary yes/no label. However the protein recognition problem has a limited set of more than two, and the chemical reaction modelling problem will have an output of real numbers describing the amount or reactants required for the solution. So the sports car problem can be defined as a binary classification, while the protein recognition is a multi-class classification, and the real number output question of chemical reactants is a regression problem [19].

Learning by training set instead of a function of input-output, is called supervised learning. In supervised learning all the training set is labelled. One data pack includes the data itself and which one of the data packs it belongs to. The labelling of the data sets is usually done by an outside mechanism; a human; thus the process itself is called supervised learning [21].

In a strategy such as this, the supervisor feeds the system with examples as an input-output training set. The task of the system here is to map the training data the supervisor provided, thus the system has a healthy amount of data to learn the outcomes of specific type of data inputs [19].

The training set is expanded and continuously fed into the system as long as the error percentage of the outputs to the real figures is above a certain line. When the system error percentage drops below the requirements, the system is considered trained and the training phase is completed. Input-output relationship is usually a function that addresses the inputs for the outputs. However this addressing does not work perfectly all the time as the outputs may corrupt due to interference [22].

Neural Networks (multilayer perceptron), Support Vector Machines, linear regression are all some of the examples of supervised learning.

2.2.2 Unsupervised Learning

In addition to supervised learning there is another method called unsupervised learning. In supervised learning training sets, images or pixels are given to construct

a whole object, such as to process interior or exterior images, or to define face and background pixels [20].

In these kinds of systems, there is no supervisor to help the system learn. The system only receives the input data. The training is done solo by the systems to find out the relationships of the parameters in the examples. There is no data about the correct outputs.

All data packs in such systems include the information required except for the labels. This type of learning attempts to find out rules, labels and sometimes number of sets in unlabeled training sets [21].

This type of learning is mainly used for classification problems. After training, the user is required to label the outputs for the next training session. Unsupervised training method maybe used for taxonomy problems when the data is required to be categorized under significant categories [18].

Self organizing maps (SOM) can be an example of unsupervised learning.

2.2.3 Semi-supervised Learning

Semi-supervised learning sits right in between supervised learning and unsupervised learning having a bit of both positive and negative sides of both learning methods. A small amount of labelled data used in conjunction with a large set of unlabeled data which increases the efficiency of training sets in some occasions.

Labelling training sets takes a huge amount of time and funding and requires a fully competent person to work for hours on every training data. When considering using more than thousands of data packs for a training data it may be even impossible to fully label whole training sets. Thus, instead of not using the data, the unlabeled data may be used with semi-supervised learning systems.

A common procedure is to use a small set of labelled data with unlabeled data in a first training session to train the computer to semi-label the data groups. The second part is to use this trained and semi-labelled data train the system and use the model for producing outputs. The first part of the procedure might be repeated to decrease the error percentage.

A real life example is filtering spam e-mails or organizing web filters for spam web sites. Considering that this method works really well with text document filtering and

the great amount of unlabeled test data on internet, semi-supervised learning shows great success.

However; when used with big groups of labelled sets; it is not worthwhile, time wise to add unlabeled data to the training set as the error percentage difference with or without the unlabeled data, rapidly decreases as the amount of labelled data increase.

2.2.4 Reinforcement Learning

In machine learning, the system is required to choose a method of approach while taking into consideration the current situation. Reinforced learning problems are subsets of this tasks in which the systems does not see the correct approach. Instead of seeing the correct approaches the system is rewarded or penalized for the methods it uses. Supported or reinforced learning is the method of trying to maximize the rewards, thus to teach the best mapping approach for the situations. The supervisor helps the systems by constantly monitoring the output of the system and rewarding or penalizing the system according to the “correct” output data he has. The actual correct output data is never given to the system. The system, considering the rewards or the penalties given by the supervisor, improves its training procedure [20].

On the contrary to supervised learning, where approach in any given situation is never fed to the system, this system is required to take consideration of the given penalty and reward signals and create a better training mapping. Robot control, computer gaming and system optimization may be perfect examples of reinforced learning in real world [23].

One hardship for reinforcement learning that can be seen on systems is the fact that researching for new approaches and trying the old ways with known rewards seem to contradict each other. As the systems struggles to maximize rewards, it must use old approaches with positive results and also try new approaches to improve the learning process. While there is no single solution to this paradox, it is obvious the fact that one of these strategies cannot just bring the system to optimal solution [20].

2.3 Statistical Learning Theory

Statistical learning theory was developed in 1960s by Vladimir Vapnik and Alexey Chervonenkis, which is basically a mathematical structure for non-parametric

dependency estimation from a given data. This theory combines fundamental concepts and principles related to learning, well-defined problem formulation, and self-consistent mathematical theory [24].

Statistical learning theory is based on theoretical analysis of empirical risk minimization (ERM) inductive principle. Empirical risk minimization principle is a method where the unknown dependencies are estimated by minimization of average training error called the empirical risk with respect to model parameters. Since the mathematical theory behind Statistical Learning Theory is quite complicated, only an overview will be given in this part.

The learning problem can be constructed as follows. Let variable x and y has a relationship according to an unknown joint probability distribution $P(x, y)$ [24].

$$(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l), \quad x, y \in \mathcal{R}^n \times \mathcal{R} \quad (2.1)$$

Consider the expected value of the loss, given by the risk functional;

$$R(\alpha) = \int L(y, f(x, \alpha)) dP(x, y) \quad (2.2)$$

The goal is to find the function $f(x, \alpha_0)$ which minimizes the risk functional $R(\alpha)$ in the situation where the joint probability distribution $P(x, y)$ is unknown and the only available information is contained in the training set [25]. Here, α represents functions and $L(y, f(x, \alpha))$ is the loss which is caused by using $f(x, \alpha)$ while estimating y .

In order to minimize the risk functional (2.2), for an unknown probability measure $P(x, y)$ the following induction principle is usually used. The expected risk functional is replaced by the empirical risk functional [25];

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l L(y_i, f(x_i, \alpha)) \quad (2.3)$$

which is constructed on the grounds of training set given in (2.1). Here l is the number of random independent identically distributed observations drawn according to the probability distribution.

This principle is called the empirical risk minimization induction principle (ERM principle).

With an η such that $0 \leq \eta \leq 1$, the following bound holds with probability $1 - \eta$ [24];

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h \ln \left(\frac{2l}{h} + 1 \right) - \ln \frac{\eta}{4}}{l}} \quad (2.4)$$

Here, $R(\alpha)$ is limited by the empirical risk $R_{emp}(\alpha)$ and an additional term, which is called confidence interval. The confidence interval includes the parameter h , or VC (Vapnik-Chervonenkis)-dimension and the number of training samples.

The VC-dimension, h is a value that measures the capacity of a set of functions [26]. The VC dimension of a set of functions is the maximum number of points that can be separated (shattered) in all possible ways by that set of functions. For hyperplanes in $\mathbb{R}^n$, the VC dimension can be $n + 1$.

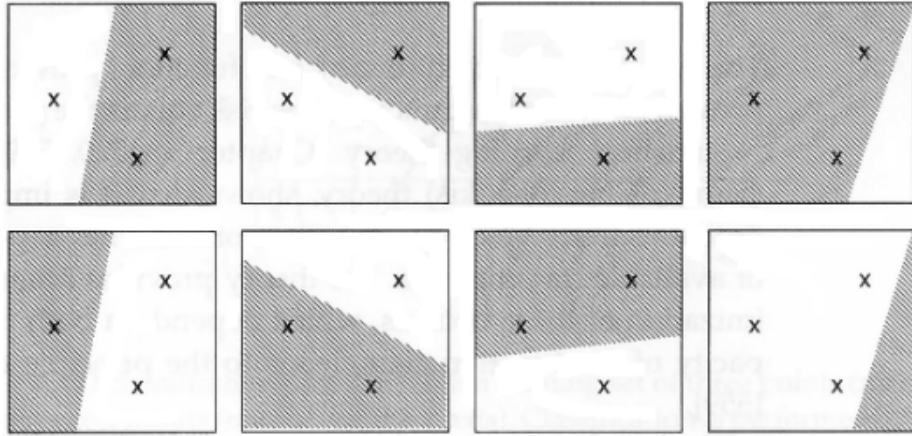


Figure 2.1: VC dimension illustration [27]

Figure 2.1 shows a simple VC dimension example. There are $2^3 = 8$ ways of assigning 3 points to two classes. For the displayed points in $\mathbb{R}^2$, all 8 possibilities can be realized using separating hyperplanes, in other words, the function class can shatter 3 points. This would not work if we were given 4 points, no matter how they are located. Therefore, the VC dimension of the class of separating hyperplanes in $\mathbb{R}^2$ is 3[28].

As stated in [29], the ERM principle is aimed for dealing with large sample sizes. This can be justified by considering the inequalities in (2.4). When l/h is large, the

confidence interval becomes small. The actual risk $R(\alpha)$ is then close to the value of the empirical risk $R_{emp}(\alpha)$. In this case, a small value of $R_{emp}(\alpha)$ provides a small value of $R(\alpha)$. However, if l/h is small, then even a small $R_{emp}(\alpha)$ does not guarantee a small value of $R(\alpha)$.

In order to minimize $R(\alpha)$, a new principle is needed, for minimizing the empirical risk as well as the VC-dimension of the set of functions [29]. This is called Structural Risk Minimization (SRM) principle. SRM is intended to minimize the risk functional with respect to both empirical risk and VC-dimension of the set of functions. [25]

Let S the set of functions $f(x, \alpha)$ be provided with a structure: so that S is composed of the nested subsets of functions S_k such that; [25]

$$S_1 \subset S_2 \subset \dots \subset S_\infty \quad (2.5)$$

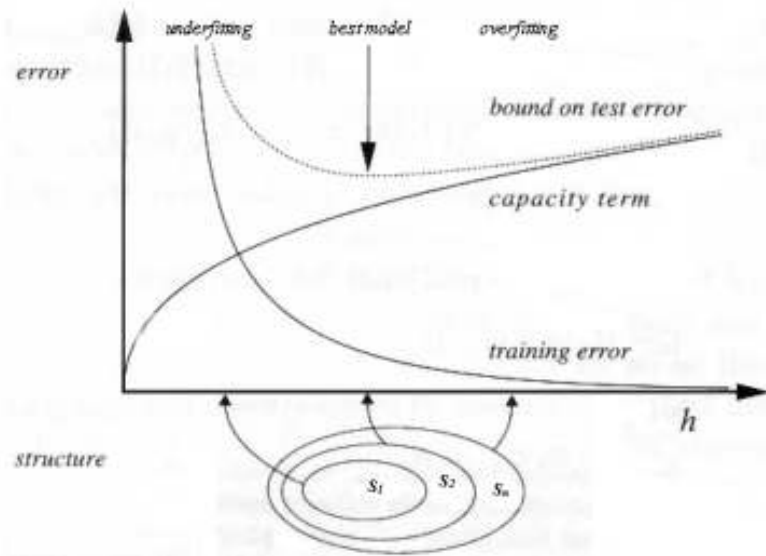


Figure 2.2: Diagram of SRM [27]

Two methods exist to achieve the SRM principle. One is to search for the minimized empirical risk in each subset, then select the subset that minimizes the sum of the minimized empirical risk and the confidence interval. In fact, this method is time-consuming. When the number of subsets is very large, even infinite, it becomes impractical. The other is to design a structure for the function set, which makes each subset get the minimized experiential risk, then select an appropriate subset to make the confidence interval minimal. In this subset, the function that makes the experiential risk minimized is optimal. [24]

3 SUPPORT VECTOR MACHINES

In this chapter, the concept of Support Vector Machines (SVMs) and their application areas are examined. Firstly, a brief history of SVMs is given. Secondly, the basics of SVMs are handled in respect to regression case. Lastly, kernel functions and their selections are described in Kernel-Induced Feature Spaces section of this chapter.

3.1 Historical Overview of Support Vector Machines

Foundation of SVM dates back to 1936, with the first algorithm for pattern recognition from R.A Fisher. In 1950, Aronszajn brings up kernels, which are the integral parts of SVMs, with his study “Theory of Reproducing Kernels”. Nevertheless, the first and biggest step among the machine learning area was Frank Rosenblatt’s linear classifier called the “perceptron”, which was invented in 1957 [30]. Perceptron can be described as the simplest kind of feed-forward neural network. With the invention of perceptron model, the mathematical analysis of the learning systems were began.

In 1963, Vapnik and Lerner introduced The Generalized Portrait algorithm. The main importance of The Generalized Portrait algorithm is, SVM is the nonlinear generalized version of this algorithm. Vapnik and Chervonenkis continued the researches about The Generalized Portrait algorithm and developed it in 1964 [30].

With the help of other research in this area, including kernels, hyperplanes and etc., in 1974 the area Statistical Learning Theory began to take shape. It can be said that SVMs have started when statistical learning theory has developed further with Vapnik in 1979 [30].

In its present form, Vapnik and co-workers developed the SV machine at AT&T Bell Laboratories. Due to this industrial context, SV research has up to date had a sound orientation towards real-world applications. Initial works focused on OCR (Optical Character Recognition). Within a short period of time, SV classifiers became

competitive with the best available systems for both OCR and object recognition tasks [31].

In 1995 the soft margin classifier was introduced by Cortes and Vapnik . In the same year the algorithm was extended to the case of regression by Vapnik in The Nature of Statistical Learning Theory [30].

With the development of regression and time series prediction option, today SVM has a wide variety of application area, such as financial market applications, control systems and signal processing, machine reliability forecasting and electric utility forecasting.

3.2 Support Vector Machines in Regression

SVM is a powerful technique used in solving main learning problems. It is derived from the statistical learning theory. Basically, it was first used in pattern recognition and classification problems but it is modified to be used in regression problems

Basically, an SVM creates a hyperplane or a set of hyperplanes in a high or infinite dimensional space for solving these classification or regression problems. With the help of hyperplane, which is located in between the maximum distance to the closest data points of each class, the separation of the given data set is made.

Generally speaking, for learning problems in the case of regression, training data is given to the learner, in order to find out the relation (correlation, mapping or function) between inputs and outputs of a function $f(\mathbf{x})$. Training set;

$$D = \{[\mathbf{x}_i, y_i] \in \mathcal{R}^n \times \mathcal{R}, i = 1, \dots, l\} \quad (3.1)$$

consists of input vectors $\mathbf{x}$, $\mathbf{x} \in \mathcal{R}^n$, and outputs y , $y \in \mathcal{R}$, that constructs $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)$ pairs.

The main idea of the support vector machines in regression is that the inputs $\mathbf{x}_i$ needs to be mapped into a higher dimensional feature space with the function ϕ [28].

$$\begin{aligned} \phi(\mathbf{x}): \mathcal{R}^n &\rightarrow \mathcal{R}^f \\ \mathbf{x} \in \mathcal{R}^n &\rightarrow \phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \ \phi_2(\mathbf{x}) \ \dots \ \phi_n(\mathbf{x})]^T \in \mathcal{R}^f \end{aligned} \quad (3.2)$$

After the inputs are mapped into higher dimensional space, the problem converts from a nonlinear regression case into a linear regression case, which enables the quadratic programming formulation of the problem.

The SVM considers approximating functions of the form [32];

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b \quad (3.3)$$

where the $\boldsymbol{\phi}(\mathbf{x})$ are called mapping function, which is an advanced defined function.

In formulation, parameters $\mathbf{w}$ and b can be found by minimizing expression of the error function R [32];

$$R = \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^l |y_i - f(\mathbf{x}_i, \mathbf{w})|_{\varepsilon} \right) \quad (3.4)$$

where;

$$|y - f(\mathbf{x}, \mathbf{w})|_{\varepsilon} = \begin{cases} 0 & \text{if } |y - f(\mathbf{x}, \mathbf{w})| \leq \varepsilon \\ |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon & \text{otherwise} \end{cases} \quad (3.5)$$

Here, $\|\mathbf{w}\|^2$ is the norm of the weight vector that defines the capacity of the model for an optimum generalization. The term ε , represents the Vapnik's ε -insensitive loss function which defines the width of the tube as shown in Figure 3.1. If the predicted value is within the tube, the cost (error) is equal to zero. But, if the predicted value is outside of the radius ε , then the cost is equal to the distance between the tube and the predicted value. On the other hand, ε affects the amount of support vectors that is used in order to construct Support Vector Regression model. If the value of ε is big, the amount of support vectors used on the model is few and the calculations become quicker. But, the result may be unsatisfactory. Conversely, if the value of ε is small, the amount of support vectors used is more which results longer calculation durations. But the result becomes more accurate. Therefore, the choice of the parameter ε has an important role in Support Vector Regression performance.

The penalty parameter C determines the trade-off between an approximation error and the weights vector norm. An increase in C penalizes larger errors (large ξ and ξ^*) and in this way leads to a decrease in approximation error. However, this can be

achieved only by increasing the weights vector norm. At the same time an increase in $\|\mathbf{w}\|$ does not guarantee good generalization performance of a model [32].

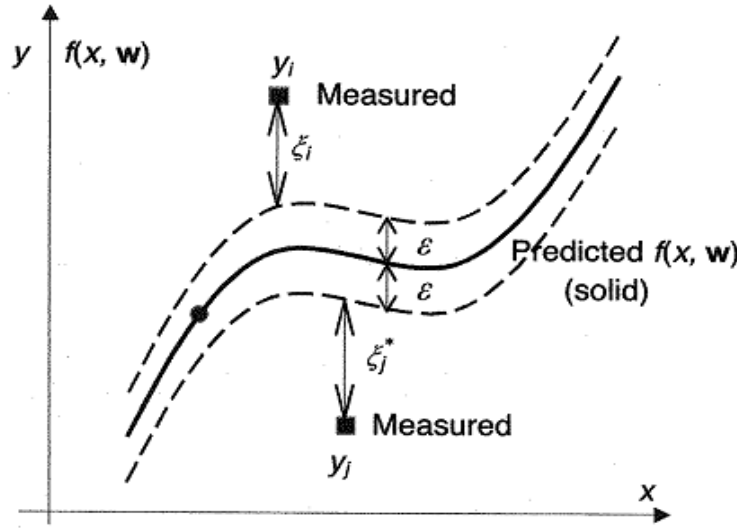


Figure 3.1: The parameters used in (one dimensional) support vector regression [32]

As can be seen from the Figure 3.1 above, the training data points that lies outside of the ε -insensitive tube can be written as;

$$|y - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi \text{ for the points above the tube}$$

$$|y - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi^* \text{ for the points below the tube}$$

where ξ and ξ^* are slack variables which are positive values. Substituting the equations above into the R ;

$$R_{\mathbf{w}, \xi, \xi^*} = \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \right] \quad (3.6)$$

subject to the constraints,

$$\begin{aligned} y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - b &\leq \varepsilon + \xi_i & i = 1, \dots, l \\ \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b - y_i &\leq \varepsilon + \xi_i^* & i = 1, \dots, l \end{aligned} \quad (3.7)$$

$$\xi \geq 0$$

$$\xi^* \geq 0 \quad i = 1, \dots, l$$

The described regression problem now can be solved with the help of Lagrange multipliers [32];

$$\begin{aligned}
L(\mathbf{w}, b, \xi, \xi^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) \\
&= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\
&\quad - \sum_{i=1}^l \alpha_i (\varepsilon + \xi_i - y_i + \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b) \\
&\quad - \sum_{i=1}^l \alpha_i^* (\varepsilon + \xi_i^* + y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - b) - \sum_{i=1}^l (\beta_i \xi_i + \beta_i^* \xi_i^*)
\end{aligned} \tag{3.8}$$

The Lagrangian above should be minimized according to $\mathbf{w}, b, \xi, \xi^*$ and maximized according to $\alpha_i, \alpha_i^*, \beta_i, \beta_i^*$. The derivatives in respect to $\mathbf{w}, b, \xi, \xi^*$;

$$\frac{\partial L}{\partial b} = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0 \tag{3.9}$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^l (\alpha_i^* - \alpha_i) \boldsymbol{\phi}(\mathbf{x}_i) = 0 \tag{3.10}$$

$$\frac{\partial L}{\partial \xi} = C - \alpha_i - \beta_i = 0 \tag{3.11}$$

$$\frac{\partial L}{\partial \xi^*} = C - \alpha_i^* - \beta_i^* = 0 \tag{3.12}$$

Applying Karush-Kuhn-Tucker (KKT) conditions for regression maximize dual variables Lagrangian [32];

$$\begin{aligned}
L_d(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) &= -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) \\
&\quad + \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j)
\end{aligned} \tag{3.13}$$

subject to the constraints

$$C \geq \alpha_i, \alpha_i^* \geq 0, i = 1, \dots, l$$

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \quad (3.14)$$

where Kernel function is defined as;

$$K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}_j) \quad (3.15)$$

β_i and β_i^* are eliminated with equations (3.11) and (3.12) in (3.13). $\mathbf{w}$ can be calculated with (3.10). As a result, the regression function takes the form;

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_j) + b \quad (3.16)$$

In order to find b , Karush-Kuhn Tucker complementarity conditions can be used.

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b) &= 0 \\ \alpha_i^*(\varepsilon + \xi_i^* - y_i + \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b) &= 0 \\ (C - \alpha_i)\xi_i &= 0 \\ (C - \alpha_i^*)\xi_i^* &= 0 \end{aligned} \quad (3.17)$$

As can be seen from the formulations above, if α_i and α_i^* are within the range of $(0, C)$, ξ_i and ξ_i^* equal to zero. Hence,

$$\begin{aligned} y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - \varepsilon &= b, & 0 < \alpha_i < C \\ y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + \varepsilon &= b, & 0 < \alpha_i^* < C \end{aligned} \quad (3.18)$$

Support Vector Machine generalization performance (estimation accuracy) depends on a good setting of parameters; cost of error or the penalty parameter C , the width of the tube ε and the kernel function parameters.

The type kernel function is decided by the type of the data set. Since they define the structure of the high dimensional feature space, they can directly affect the prediction quality. There is no general rule for the kernel function decision which makes this choice even more critical.

Existing software implementations of SVM regression usually treat SVM parameters as user-defined inputs. Selecting a particular kernel type and kernel function parameters is usually based on application-domain knowledge and also should reflect distribution of input values of the training data.

3.3 Kernel Induced Feature Space

A function $\phi_i: X \rightarrow \mathfrak{R}$ that maps each object $x \in X$ to a real value $\phi_i(x)$ is called a mapping function. Combining mapping functions $\phi_1, \dots, \phi_n$ results in a feature mapping $\phi: X \rightarrow F$ and the space F is called a feature space [19]. The Support Vector Regression model described in previous chapter mentioned that the nonlinear input data points need to be mapped to a higher dimensioned feature space in order to solve the issue with the help of a optimal hyperplane. As stated before, the kernel functions are used to map the input values to this feature space with the calculations made in the input space, which helps to simplify the given task.

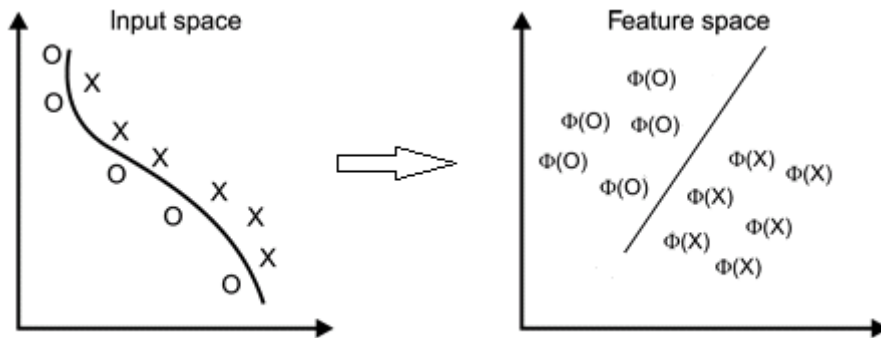


Figure 3.2: Kernel transformation

The kernel function is the key element in a Support Vector Machine. The importance of choice is explained in the previous sections. In this part, the basics of kernel functions are discussed.

A kernel is a function K , such that for all $x, y \in X$

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi_i(\mathbf{x}), \phi_i(\mathbf{y}) \rangle \quad (3.19)$$

where ϕ is a mapping from X to an inner product feature space F [28]. The usage of kernel functions allows the input data to be mapped into a feature space by

eliminating the computational problems caused by the high dimensionality. During the mapping process, only the inner product matrix, which is called Kernel Matrix, is used.

The kernel function $K(\mathbf{x}, \mathbf{y})$ should meet some requirements in order to be used on Support Vector Machine Algorithms. Any symmetric positive definite function can represent a scalar product in a feature space that satisfies following [26];

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}) \quad \lambda_i \geq 0 \quad (3.20)$$

$$\iint K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \quad g \in L_2 \quad (3.21)$$

can represent a kernel function.

There are many kernel functions used in literature. The most common ones are described as follows;

Linear Kernel Function: This is the simplest form of kernel functions. It can be represented by the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ with an additional constant c ;

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y}) + c \quad (3.22)$$

Polynomial Kernel Function: This is a popular kernel function among nonlinear problems. It can be represented as;

$$K(\mathbf{x}, \mathbf{y}) = [(\mathbf{x}^T \mathbf{y}) + c]^d \quad (3.23)$$

Note that, if the parameter d is equal to 1, the resulting function is the same as linear kernel function.

Gaussian Radial Basis Function: This is the most common used kernel function in the literature because it enables the support vector machine to have good interpolation ability.

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2) \quad (3.24)$$

When a kernel is used it is often unclear what the properties of the mapping and the feature space are. It is always possible to make a mapping into a potentially very high dimensional space and to produce a classifier with no classification errors on the training set. However, then the performance of the classifier can be poor. On the other hand, it is possible that a classifier with an infinite dimensional feature space performs well. Thus, the dimension of the feature space is not the essential quantity when choosing the right kernel [33].

The choice of kernel function depends on the user and the type of problem; amount of data and class, noise ratio, computational skills and -most importantly- experience. Since there is no general answer, the functions should be tried and the one which suits the model should be chosen [32].

4 ELECTRICAL LOAD DEMAND FORECASTING

In the electrical industry, to make precise decisions at the right time, the first step is to make an accurate electrical load forecasting. In order to make profitable and efficient investments, build new systems and expand the capacity of the existent systems, the energy demand needs to be known. Especially in deregulated markets, an investor who wants to compete with others must foresee the load demand at a certain time. This is where load forecasting comes into play.

In forecasting, an underestimation can cause a limitation in the energy supply, decrease the quality of this energy and the reliability of the system. Besides, since the supplier may need to purchase energy from other suppliers in the market, the supplier will be affected economically. On the other hand, an overestimation can lead to unnecessary investments for generation or systems with low generation and inefficient facility conditions.

Above and beyond, one of the main factors that determine the price is the demand. The forecasting of load demand helps the price policy. In a nutshell, load forecasting is the centre of electrical industry.

In this chapter, first the importance of planning in generation, transmission and distribution will be briefly explained. Then, the factors that affect the load demand will be discussed. The load forecasting methods will be described in Load Forecasting Approaches section of this chapter.

4.1 Planning in Generation/ Transmission/ Distribution

Planning can be described as defining the pathways to achieve a goal. In industrial manner, planning is a process to achieve a goal that includes gathering all the information at hand (resources, costs, laws and even politics); identifying the alternatives; checking the availability, pros and cons of the selected option and proceeding.

Planning in electrical industry is essential because of economical and operational aspects. Economical reasons include purchasing and pricing factors whereas operational factors consist of generating, load switching, unit commitment and maintenance. All the factors affect the planning process and in order to perform an efficient planning, it all begins with an accurate forecast.

The total energy consumption is directly proportional to the Gross Domestic Product (GDP), one of the measurements for a country's overall economic output. For Turkey, the projection can be seen in Figure 4.1.;

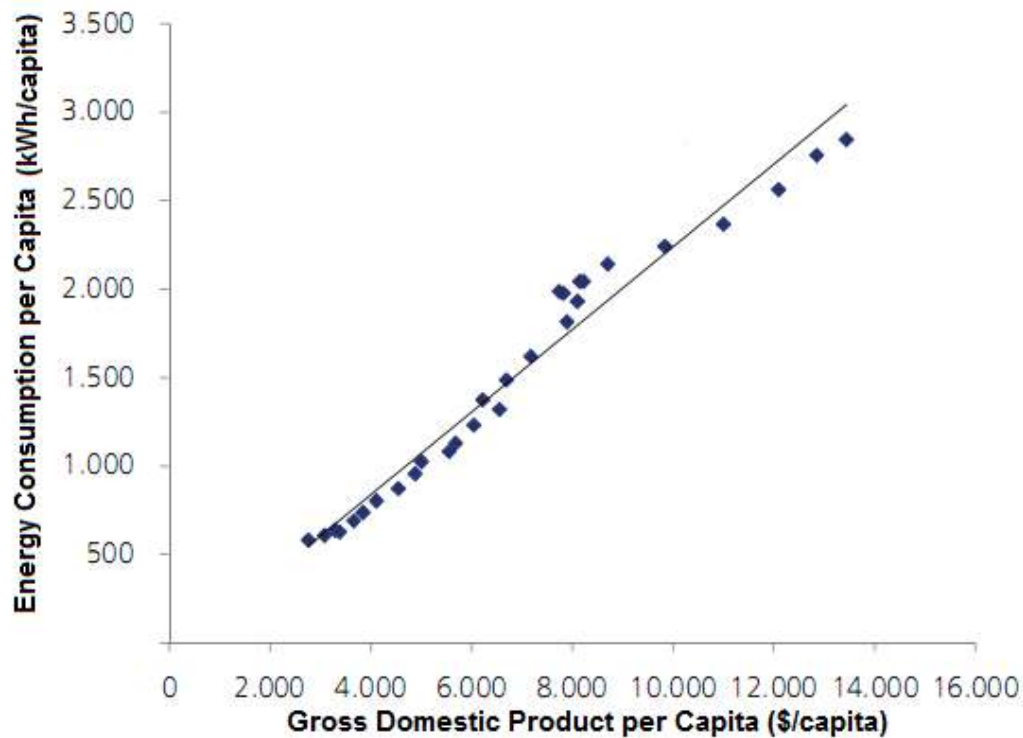


Figure 4.1: The relation between Gross Domestic Product per Capita and Electrical Energy Consumption per Capita [34]

The projection above starts from the 1980s, which has an average GDP of 2100 \$/capita with a consumption 750 kWh/capita, and continues to present day, 2000s, which has an average GDP of 6350 \$/capita with a consumption 2300 kWh/capita. As the required energy is increased with the economic growth, the investments in electrical industry should be planned accordingly.

The main goal of generation, transmission and distribution of an electrical energy is to provide economical and high quality energy to the consumers. Since the population grows rapidly, it is vital to keep up the quality and reliability of the

energy provided. In order to maintain this, expanding the existing systems to supply the demand is required.

Electrical systems require a complete planning process starting from purchasing fuels to generating, transmitting and dispatching the energy to the consumers. In Figure 4.2, the process is described briefly.

In generation systems planning, the construction of power plants, which will meet the forecasted energy demand in a suitable and economical way, and the schedules of these plants are determined. In transmission and distribution planning, the main goal is to build the distribution and transmission lines and stations which will ensure providing the end users qualified, economical and reliable energy.

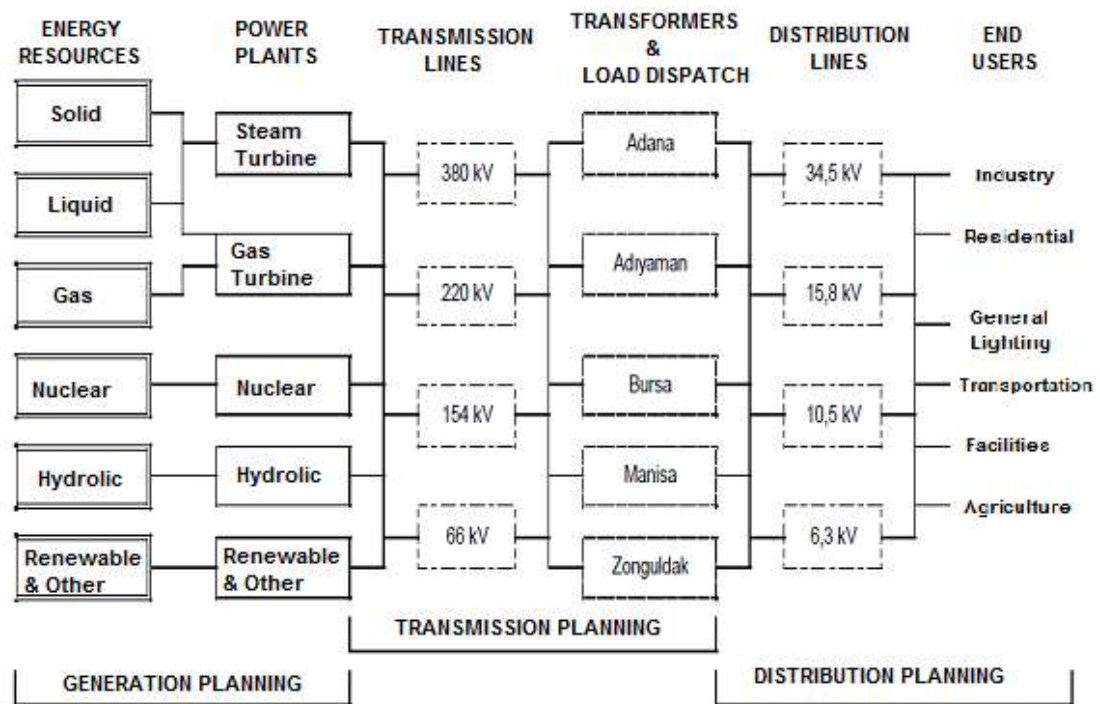


Figure 4.2: System planning in electrical industry [35]

As can be seen above, planning is the first step in providing reliable and economical energy to the end users. In order to have an efficient planning, the energy demand should be predicted accurately.

Electrical load forecasts can be handled in three categories [12]:

1. Short-term load forecasts: from one hour to a day or a week
2. Medium-term load forecasts: from a week to a year
3. Long-term load forecasts: more than a year to numerous years

These forecasts for different time ranges are important for different operations.

Short-term load forecasts include the forecasts from an hour to daily or weekly forecasts. It is used for the day-day operations of the power system, such as hydro-thermal coordination, scheduling of energy transactions in deregulated markets, scheduling of start-up times of new units, load flow analysis, and power system security studies. The main variables of these forecasts are weather conditions and time [15].

Medium-term load forecasts include the forecasts from a week to a year [10]. The main achievements with these forecasts are defining the pricing policies, scheduling maintenances, organizing fuel supply, minor infrastructure adjustments and determining the power plants which can be used in urgent conditions. Especially in the deregulated electricity sector, medium term load forecasting has gained even more importance since the market players generally need to sign annual contracts for energy transactions. Any major deviations of the forecasted quantities from the actual demand lead to financial penalties. The medium term forecasting algorithm may need to take into account seasonal patterns, weekly patterns, and economic and demographic factors such as the Consumer Price Index and the Average Salary Earning. The medium term load forecast provides estimates of the peak load and the daily energy requirement [15].

Long-term load forecasts aim at load prediction for more than a year ahead. This type of forecast is needed for planning purposes such as constructing new power stations, increasing the transmission system capacity, and in general for expansion planning of the electric utility. Expansion planning requires a significant time period since it involves feasibility studies, expropriation of land, design and operational analysis, and international competitions for the supply of equipment. Long term forecasting takes into account the population growth, industrial expansion, local area development, the gross domestic product, and past annual energy consumption. The output of this type of forecast is the annual peak load demand and the annual energy demand for the years ahead [15].

4.2 Factors in Load Demand

In general, the electric energy can be used in a wide variety, such as industrial activities, public and private usages, etc. Factors affecting the load demand depend on the particular consumption division. The electric usage pattern is different for different divisions and for different time ranges. It can be affected by economical situation, population, weather, holidays and even random events like TV-shows, etc. Figure 4.3 shows the relation between load and its major factors.

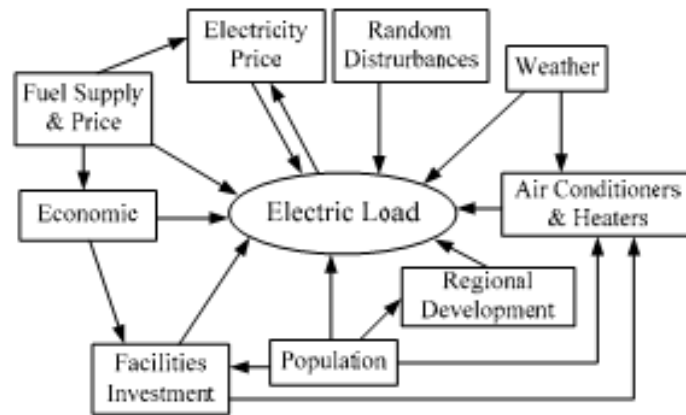


Figure 4.3: Factors that affect load demand [36]

Although there are many factors that influence the energy consumption, the usual approach in load forecasting is to concentrate on the total load of the whole utility. This approach reduces the number of factors that can be taken into account. In general, there are four main factors that affect the load demand;

- In short term, the meteorological conditions have the major impact on load demand. In addition to the temperature, also wind speed, cloud cover, and humidity have an influence.
- In long term, economic and demographic factors are gaining importance on load demand.
- In general, time factors need to be taken into account, such as seasonal effects, daily and weekly rhythms, holidays, etc.
- Random factors, from minor private events to large community activities, can have an effect on the load demand.

4.3 Load Forecasting Approaches

A large number of methods and techniques have been developed to perform electric load forecasting. These range from regression-based approaches over time-series approaches towards artificial neural networks and expert systems. In the following, a short overview over some of the models and methods is given.

4.3.1 Classical Approaches

Classical, which can be named statistical, approaches necessitate a mathematical model that describes the relationship between load and several input factors. Numerous classical models are applied for electrical load forecasting, such as end-use, econometric, regression and time series models. In the following, regression-based and time series models are briefly explained.

- Regression Models: These models are quite common in load forecasting and used to model the relationship between the load and external factors, for example weather and calendar information or customer types. Regression methods are relatively easy to implement. The most common, linear regression is used. The coefficients of these variables are estimated using least squares or other regression techniques. Another aspect is that the relationship between input and output variables is easy to understand [10].

Although regression-based methods are widely used by electric utilities, they suffer from a number of negative aspects. Due to the nonlinear and complex relationship between the load demand and the influencing factors, it is not easy to develop an accurate model. As indicated in Kyriakides et al., on site tests of regression-based methods have shown deterioration in performance in cases where the load deviates due to sudden weather changes and load events. One of the main reasons for this problem is that the model is linearized in order to estimate its coefficients. However, the load patterns are nonlinear and it is not possible to represent the load demand during distinct time periods using a linearized model. In order to partially reduce this issue, it is often necessary to use statistical techniques to enable the forecaster to capture the load deviations due to sudden weather changes and special events.

- Time Series Models: Time series techniques model the load demand as a function of historical data. These techniques assume that the data follow a certain stationary pattern that depends on autocorrelation, trends in the data, and daily, weekly and seasonal variations [15]. These approaches are amongst the oldest techniques that have been used for load forecasting applications. Time series models appear in the literature in different forms such as Box-Jenkins, autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), autoregressive moving average with exogenous variables (ARMAX) and autoregressive integrated moving average with exogenous variables (ARIMAX), etc.

As stated by Kyriakides et al., if there isn't any change in the variables that affect load demand, such as environmental or social changes, time series methods generally give satisfactory results. If there is an unexpected change in any of these variables, then time series methods may not be precise. Time series methods assume that the load demand is a stationary time series and has normal distribution characteristics. When the historical load data is different from the general course, the time series forecasting accuracy decreases significantly, similar to the regression model. Moreover, since there is a need to use a considerable amount of historical data and a large number of complex relationships, time series techniques require a significant computational time.

4.3.2 Computational Intelligence Methods

Computational intelligence methods started to be applied in electrical load forecasting problems in 1990s. Since then, especially artificial neural networks, these methods have become the most popular and common methods in this field. The expression computational intelligence is commonly used to refer to the fields of artificial neural networks, fuzzy systems, support vector machines, evolutionary computation, swarm intelligence and hybrid systems which may combine two or more different approaches in order to overcome some issues of the original methods. In this chapter, artificial neural networks models and fuzzy logic models are explained.

- Artificial Neural Networks Models: An artificial neural network, basically, is an algorithm to be formed to imitate the human brain functions. Their basic unit is the artificial neuron, which receives (numerical) information through a number of input nodes, processes it internally, and puts out a response. The processing is usually done in two stages: first, the input values are linearly combined, and then the result is used as the argument of a nonlinear activation function.

An activity on one neuron can be described as [37];

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (4.1)$$

$$y_j(n) = \varphi_j(v_j(n)) \quad (4.2)$$

where m is the total number of inputs applied to the neuron, y_i is the input signals, w_{ji} is synaptic weights of neuron i , v_j is the linear combiner output and φ is the activation function. Possible forms of the activation function are linear function, step function, logistic function and hyperbolic tangent function.

The neurons are organized in a way that defines the network architecture. The most common architecture is the multilayer perceptron (MLP) type, in which the neurons are organized in layers. The neurons in each layer may share the same inputs, but are not connected to each other. If the architecture is feed-forward, the outputs of one layer are used as the inputs to the following layer. The layers between the input nodes and the output layer are called the hidden layers.

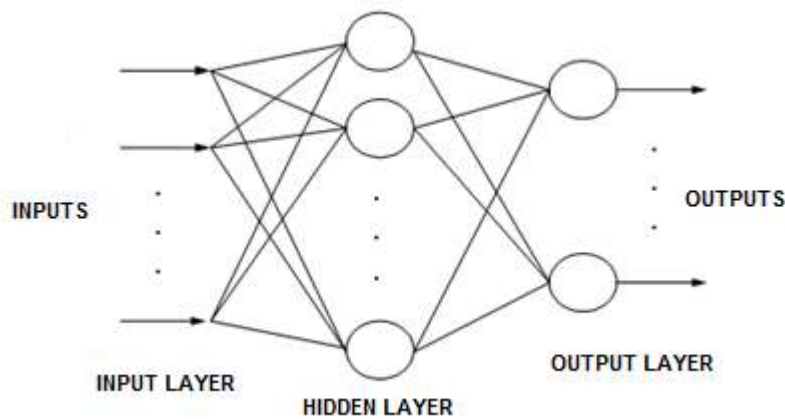


Figure 4.4: A multilayer feed-forward neural network architecture

The estimation of the parameters is called the “training” of the network, and is done by the minimization of a loss function (usually a quadratic function of the output error). Many optimization methods have been adapted for this task. The first training algorithm to be devised was the back-propagation one, which uses a steepest-descent technique based on the computation of the gradient of the loss function with respect to the network parameters (that is the reason why the activation functions must be differentiable). Many other training algorithms are now available.

In load forecasting applications, this basic form of multilayer architecture shown above is still the most popular. Nevertheless, there are a large number of other designs, which might be suitable for other applications.

The approximation capability of Neural Networks has made it convenient for common usage. For instance, a software based on the Artificial Neural Networks method was used by several electric utilities in the US and Canada for hourly short-term load forecasting [38].

- Fuzzy Logic Models: The first Fuzzy Logic researches started in 1965 with the study of Lofti A. Zadeh and continued to nowadays with an inclining popularity. The most important aspect of fuzzy logic is it allows the description of uncertainties within a mathematical description.

Fuzzy logic is a generalization of the usual Boolean logic used for digital circuit design. An input under Boolean logic takes on a truth value of "0" or "1". Under fuzzy logic an input has associated with it a certain qualitative ranges [12]. For instance, in real life one may not be exactly true or false, it can be partially true or partially false. With respect to this statement, fuzzy logic takes the infinite range between $[0, 1]$ into account.

Fuzzy logic has a numerous application areas. The main advantage of this approach is easy modelling of learning through experience problems, which necessitates the mathematical description of uncertain terms. Ergo, the fuzzy logic approach is suitable for nonlinear cases.

Fuzzy logic models are applied to electrical load forecasting problems which consists of numerous independent inputs, such as temperature, day type (holiday, weekdays, and weekends). In addition to that, there are numerous applications in this area which combines the artificial neural network and fuzzy logic approaches. These

combinations allow the user to have more efficient results in comparison to use just one approach.

5 ELECTRICAL LOAD DEMAND FORECASTING APPLICATION

In this chapter, an application in medium-term electrical load forecasting is made by SVM using the program Libsvm. Load forecasting with SVM can model the nonlinear relationship with load and the factors that affect the load. For instance, Chen et al. [39] used a Support Vector Regression technique to solve an electrical load forecasting problem, which was a competition organized by EUNITE network (European Network on Intelligent Technologies for Smart Adaptive Systems). Their approach in fact won the competition. In addition to that, Mohandes [40] used the Support Vector Regression model for short term electrical load forecasting and compared the results with the autoregressive (AR) model. As a result, the performance of the SVM was much lower than the AR method. In another study, Guo et al. [41] represented a monthly electrical load forecasting application of Hebei Province of China with SVM. They used only the load values from years 1990 to 2000 for training the model and predict the total monthly loads of the years 2001 to 2005 accordingly. The comparison with the actual loads showed that SVM is quite successful in this electrical load forecasting application. Li et al. [42] proposed a method that forecasts the daily energy consumption 24h ahead using SVM and Similar Day Method. The purpose of the Similar Day Method was to identify days according to electrical load, weather and day type which is similar to the forecasting day and use this information to adjust the result given by the SVM result. This approach corrects the outputs of the SVM and gave more accurate results. Zhang [43] proposed a model using SVM for 24 h load forecasting application. The data set includes load and weather data of previous 50 days and the results are compared to ANN. SVM gives better results than ANN. Escobar and Perez [44] represented a study for hourly load demand forecasting of a week. They used SVM and ANFIS for building the models and compared the results. SVM shows superiors results over ANFIS. Guo et al. [45] suggested an SVM model for hourly and weekly electrical load forecasting application of a city in China. They used hourly load and weather data for constructing the data set and chose the data whose weather situation is similar to that of the forecasting day to form the sample set. Obtained results are

compared with the actual loads. Shortly, many researches in literature showed that with its superior generalization capability, Support Vector Machines are successful in forecasting applications.

In this study, the data gathered from the Istanbul European Side are used as a sample for the application and according to the obtained model the daily maximum loads of April 2010 are predicted. In addition to the past load data, electricity price in Turkish Lira for economic effects, daily average temperature and calendar days are considered as attributes in forecasting.

In order to observe the influence of those factors in such a medium-term load forecasting application, three data sets are constructed. The first data set consists of past load data, calendar days, daily average temperatures and electricity price. In the second data set, only daily average temperature values are extracted. In the third data set, in addition to daily average temperatures, electricity price values are extracted and a model is obtained with only past load values and calendar days. With constructed models related loads are predicted.

The same three data sets are used with the explained ANN model and the results are compared with SVM results and real load values.

5.1 Data Sets

The relation between meteorological conditions, time, economic, demographic factors and electrical load is important for an accurate load forecasting. For this purpose, it is crucial to determine the factors that affect the load. In this study, three data sets are used for load forecasting purpose. The first data set consists of daily maximum past loads, weekdays/weekends, holidays, daily average temperature and electricity price in Turkish Lira as the economic factor. The second data set is constructed with the same data except daily average temperatures. The third data set is built with only past load values and calendar days.

The values that are used in data sets are as follows;

-Electrical Data:

For the data set, from the year 2006 to 2009 Total Istanbul European Side's daily peak loads in MW are used. In Figure 5.1 the daily peak load demand curve for the related area is given.

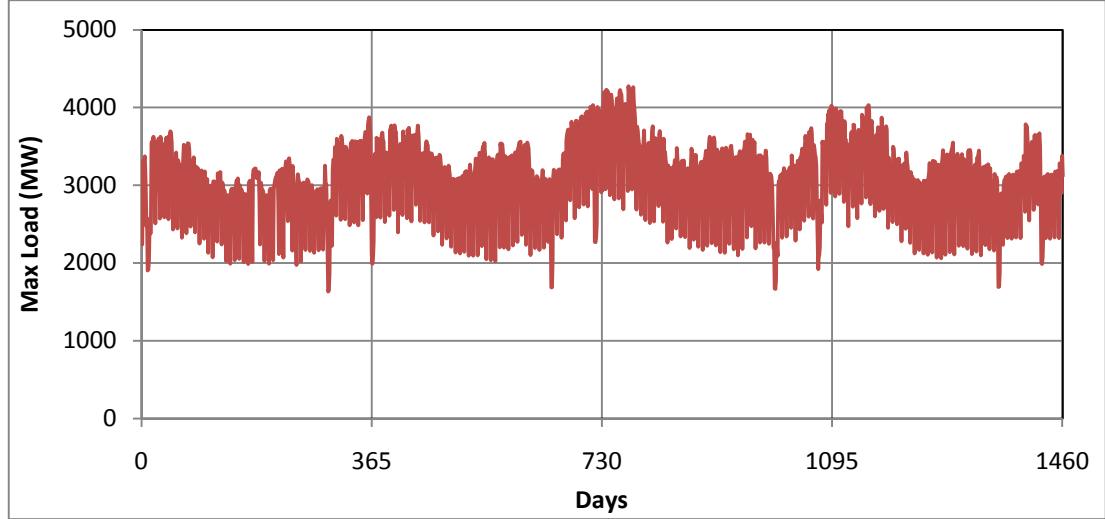


Figure 5.1: Istanbul European Region daily peak load demand

Since Istanbul European Region has both residential and industrial areas, there are no significant differences in load demand in weekdays, weekends. However, the demand in Saturdays and Sundays is slightly lower than the other weekdays. It can also be observed from the Figure 5.1, the load demand on public and religious holidays, such as Ramadan Festival, are lower than normal days.

- Calendar Days:

As it is stated in previous chapters, during holidays, public and religious holidays there are differences in load demand can be observed. In order to have an accurate forecast, these calendar days should be taken into account. In this application, the calendar information for the years 2006 through 2009 is considered in the data set.

-Temperature:

Temperature and seasonal factors (humidity, wind etc.) generally have an important effect on load demand. Here, daily average temperatures for Istanbul between the years 2006 and 2009 are given in Figure 5.2;

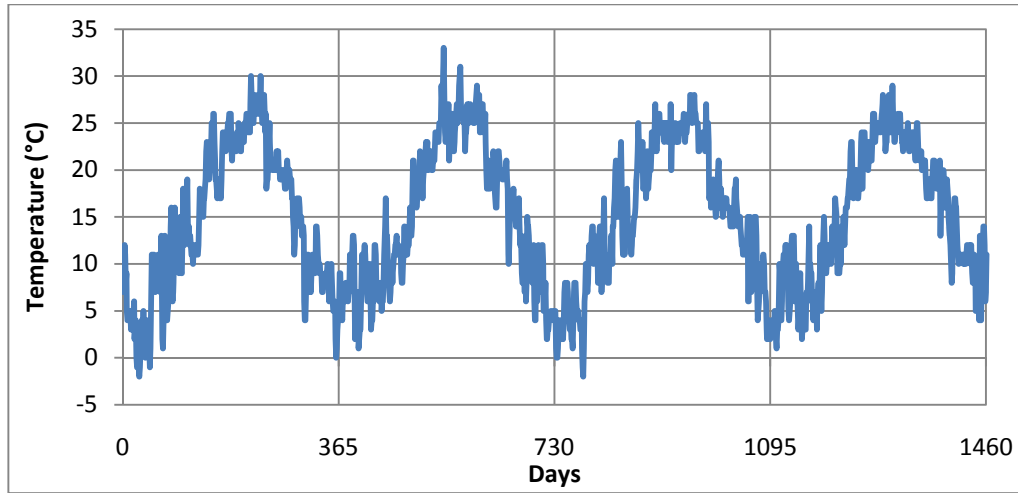


Figure 5.2: Average daily temperature for Istanbul

Seasonal effects can be clearly seen on the Figure 5.2 above. Since only temperature value gives an overall idea of the weather conditions, only temperature factor is taken into account in this study.

It can be observed that the seasonal effect of the temperature is influential on the electrical load demand by comparing Figure 5.1 and Figure 5.2. In order to find out if weather conditions should be considered in a medium-term load forecasting application for dynamically changing areas, such as the Istanbul European Side, daily average temperature values are included in the first data set and excluded from the second and third data sets.

- Economical Factors:

Especially in medium and long term electrical load forecasting, it is important to take economic and demographic factors into consideration to accurately project the demand course. In this study, Electricity Price in Turkish Lira (TL/ kWh) is used as the economic factor. The Table 5.1 shows the annual change in electricity prices from year 2006 through 2010 for only the months April, May and June.

Table 5.1: Electricity Price in TL/ kWh

Years	Electricity (TL/kWh)
2006	0.16
2007	0.16
2008	0.19
2009	0.25
2010	0.27

The first training set is built with 17 inputs including historical load data, calendar dates, daily average temperature values and electricity price assigned to each attribute as follows;

- Seven attributes for indication of the day of the week
- One attribute for indication of whether that day is holiday or not
- One attribute for the daily average temperature
- Seven attributes for maximum loads of the past seven days
- One attribute for electricity price in TL/ kWh

The second data set has the same structure except the daily average temperature. Thus, 16 inputs are encoded as above. The third data set has the same format with previous two data sets but this time 15 inputs are encoded with only past load values and calendar information.

During the training process, in order to take similar days into account, only April, May and Junes of the years 2006 through 2008 values are used for all features. The values for the related months of the year 2009 are used for testing the models, which are for adjustment of the parameters of SVM and ANN.

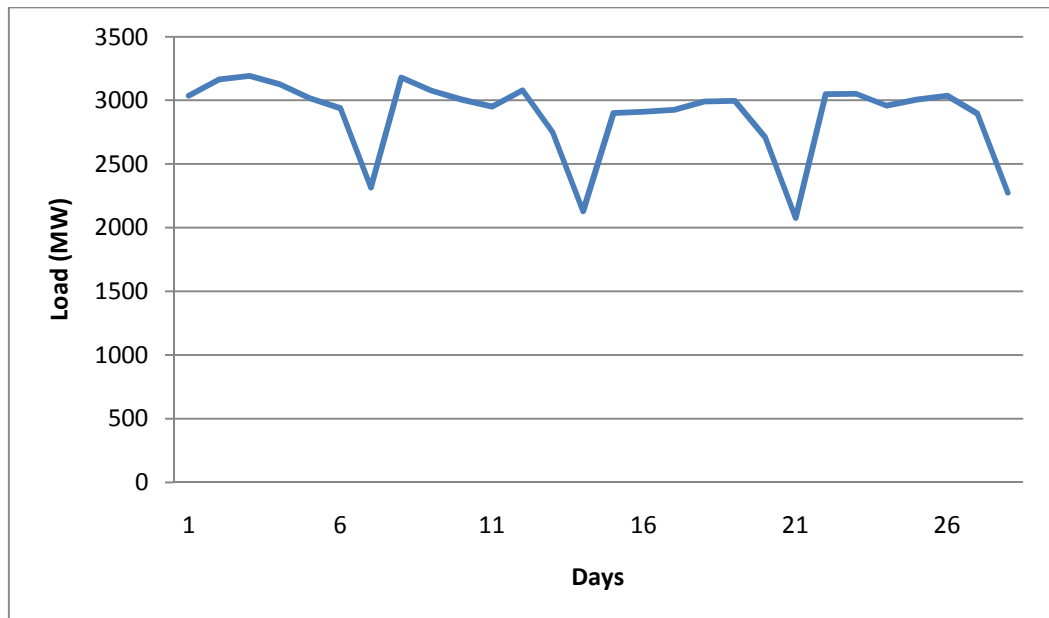


Figure 5.3: Daily peak load demand curve from April 2006

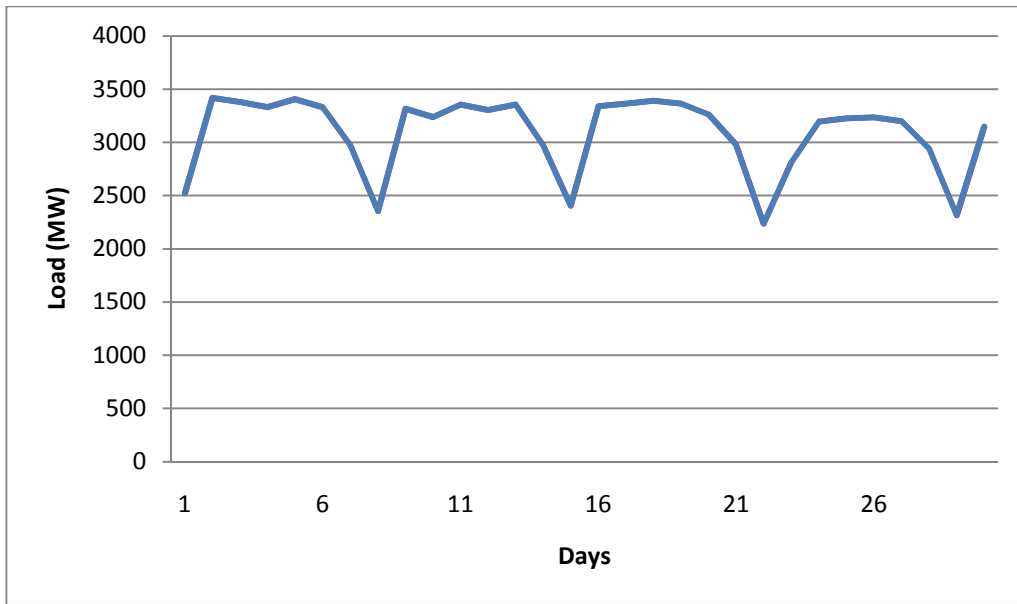


Figure 5.4: Daily peak load demand curve from April 2007

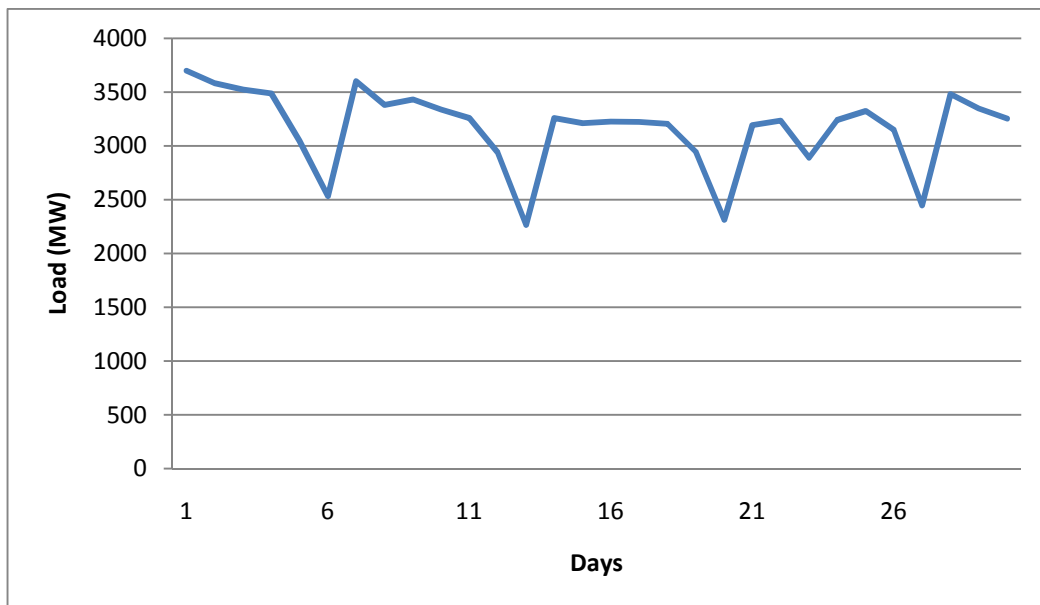


Figure 5.5: Daily peak load demand curve from April 2008

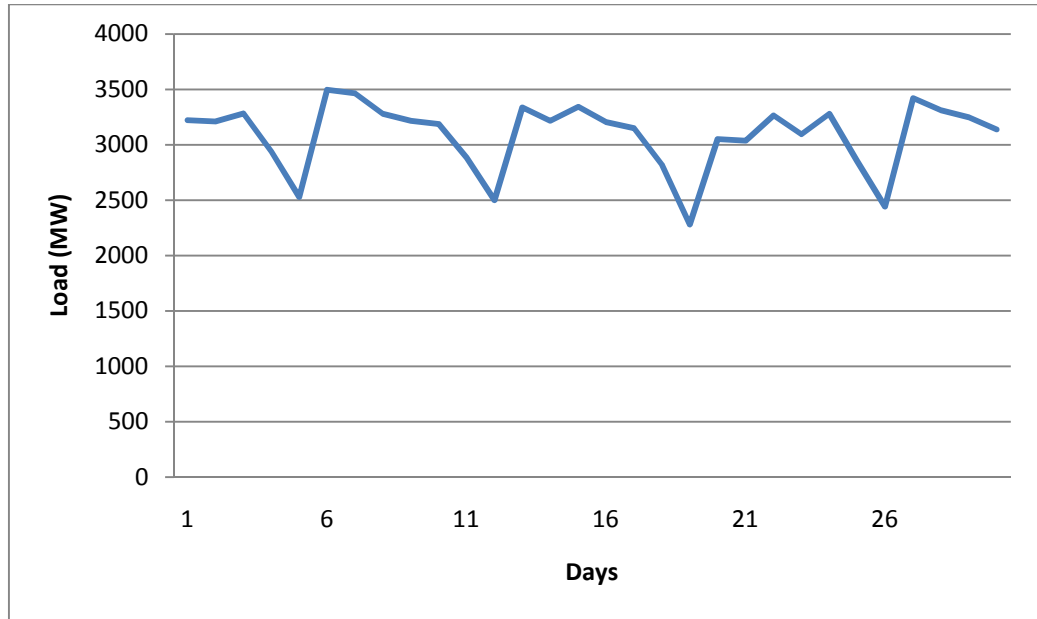


Figure 5.6: Daily peak load demand curve from April 2009

In figures above, load curve for April 2006-2007-2008 and 2009 is given as an example to the load demand course. It is obvious that load demand has a periodicity on a weekly basis. As can be seen from the load curve course above, weekends generally have lower demand than weekdays. Therefore, the maximum load of past seven days is used to effectively project the load periodicity to the model.

In forecasting process, newly predicted load is used as an attribute and included to the next prediction. For instance, after having the prediction of 1st of April, 2010, it is used within the input data of predicting 2nd of April, 2010. This process is repeated until obtaining the estimated maximum load value of 30th of April, 2010.

In order to have a good training set, it is important to scale input values. With this purpose, the entire past load inputs and, in the first data set, temperature values are scaled within the range [0, 1]. Libsvm's scale command is used for this intent.

5.2 Implementation

The general scheme of the forecasting process is as follows;

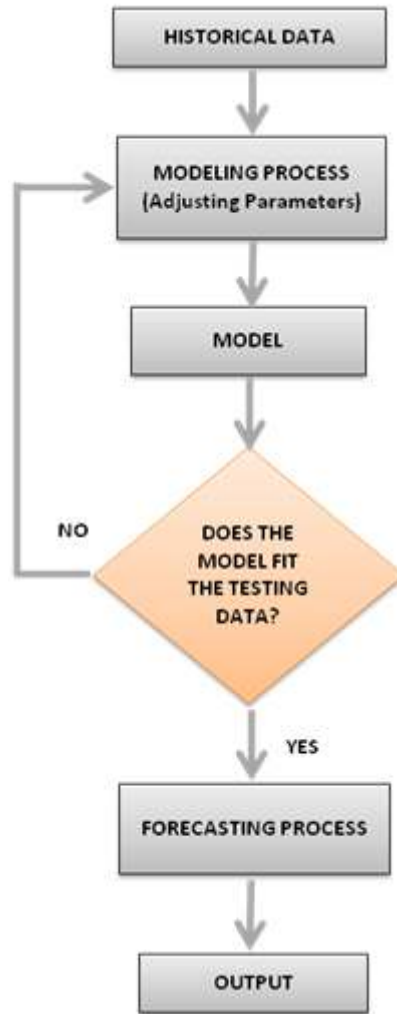


Figure 5.7: Flow chart of the forecasting application

5.2.1 Support Vector Machine Implementation using Libsvm

The program Libsvm is simple, easy-to-use, and efficient software for SVM classification and regression. In this study, the built SVM model has a single output which gives the predicted peak load for each day. When building an SVM model, there are some parameters to choose that would determine the performance. Therefore, in order to get a good model, these parameters need to be selected properly;

- 1) The kernel function
- 2) The width of the ε -insensitive tube, ε
- 3) Cost of error, C

In this study, Gaussian Radial Basis Function (RBF) is selected as the kernel function, which is the most commonly used kernel [31, 32].

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2) \quad (5.1)$$

With the selection of RBF kernel, the problem of selecting the gamma value γ in the kernel arose. In order to overcome this problem a simple cross-validation is applied to the training data. In cross-validation process, the data set is randomly divided into n sets, namely n -folded. First, training is performed with $n-1$ sets and test is performed with the rest. For instance, if a 3-fold cross-validation is performed in a data set, first 1st and 2nd data sets are used for training and 3rd set is used for testing; second 1st and 3rd data sets are used for training and 2nd data set is used for testing; finally 2nd and 3rd data set is used for training and 1st data set is used for testing. According to this process the best parameters are chosen.

The program Libsvm offers cross-validation for the parameter selection in regression cases, which results in finding the best gamma value γ in addition to cost of error C and the width of ε -insensitive tube, ε . In this study, a 5-fold cross-validation is performed on all data sets using this feature of Libsvm.

Table 5.2 : C , ε , γ values used in models

	With Past Load, Temperature and EP	With Past Load and EP	With Past Load
C	65536	2048	4096
ε	0.5	0.5	0.5
γ	0.007812	0.125	0.125

A general usage of Libsvm via command line interface is given in Appendix A. With the given data sets and parameters, SVM models are obtained. Using all three model files, daily peak loads of April 2010 are predicted.

5.2.2 Artificial Neural Networks Implementation using MATLAB

A feed-forward multilayer network is a supervised artificial neural networks model. There are different kinds of learning algorithms. Back-propagation algorithm is one of the most common learning algorithms used with this network structure. The

underlying idea is adjusting the weights and biases according to the error between target values and the outputs of the network for given inputs.

The behaviour of an ANN depends on both the weights and the input-output function, which is called transfer function. Some of the most common used transfer functions are linear transfer function, log-sigmoid transfer function and hyperbolic tangent sigmoid transfer function. In this application, log-sigmoid transfer function is used in both hidden layers and output layers.

In this study, the feed-forward network is obtained using MATLAB Neural Network Toolbox codes given in Appendix B.

Gradient descent with momentum and adaptive learning rate back-propagation (traingdx) is used as training algorithm in this two hidden layered network. Figure 5.8 shows a sample diagram of this architecture. Numbers of input neurons, output neurons, training patterns, hidden layer neurons are entered along with the training algorithm parameters learning rate, momentum constant, maximum number of epochs to train, performance goal. The main difficulties for constructing such architecture are to determine what type of training algorithm should be used and the number of hidden layer units. These training parameters are determined by trial-error and experience. As the transfer function, log-sigmoid function [37] is used in both hidden and output layers.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (5.2)$$

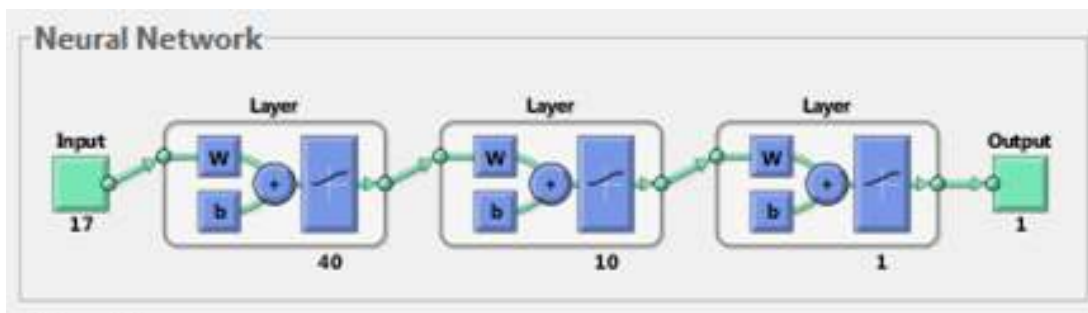


Figure 5.8: Sample artificial neural networks diagram in MATLAB

Predicted values are compared to real load values of April 2010. As the error metric, mean absolute percentage error (MAPE) is used.

$$MAPE = \frac{100}{T} \sum_{t=1}^T \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (5.3)$$

In the formula above y_i represents the real value whereas $\hat{y}_i$ represents the predicted value [10]. The value T corresponds to the total number of values, in our case this is the total number of predicted days.

5.3 Results

All three models are constructed via the training of the data sets that include the values of April, May and June of 2006-2008 as explained above. The same month's values of 2009 are used for testing the obtained models. Afterwards, the model files are used for prediction of the daily maximum load demand in April 2010. In order to compare, both models and the factors that affect the load, prediction results are given with the real values of April 2010 in Appendix C.

And the metric of error, MAPE is show in Table 5.3 for all data sets of SVM and ANN models;

Table 5.3 : MAPE Of SVM And ANN models using different data preparation

	With Past Load, Temperature and EP	With Past Load and EP	With Past Load
SVM	3.67%	2.70%	2.94 %
ANN	4.97%	2.92%	4.99 %

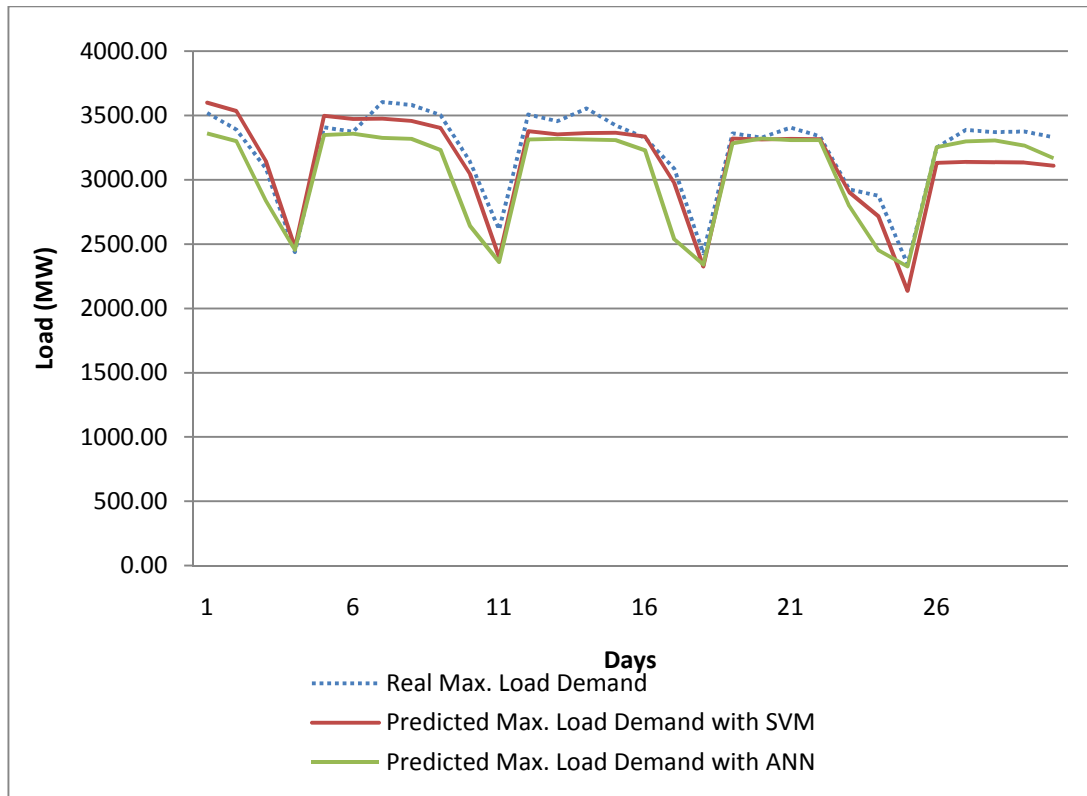


Figure 5.9: Comparison graph between real and predicted values with past load, temperature and EP

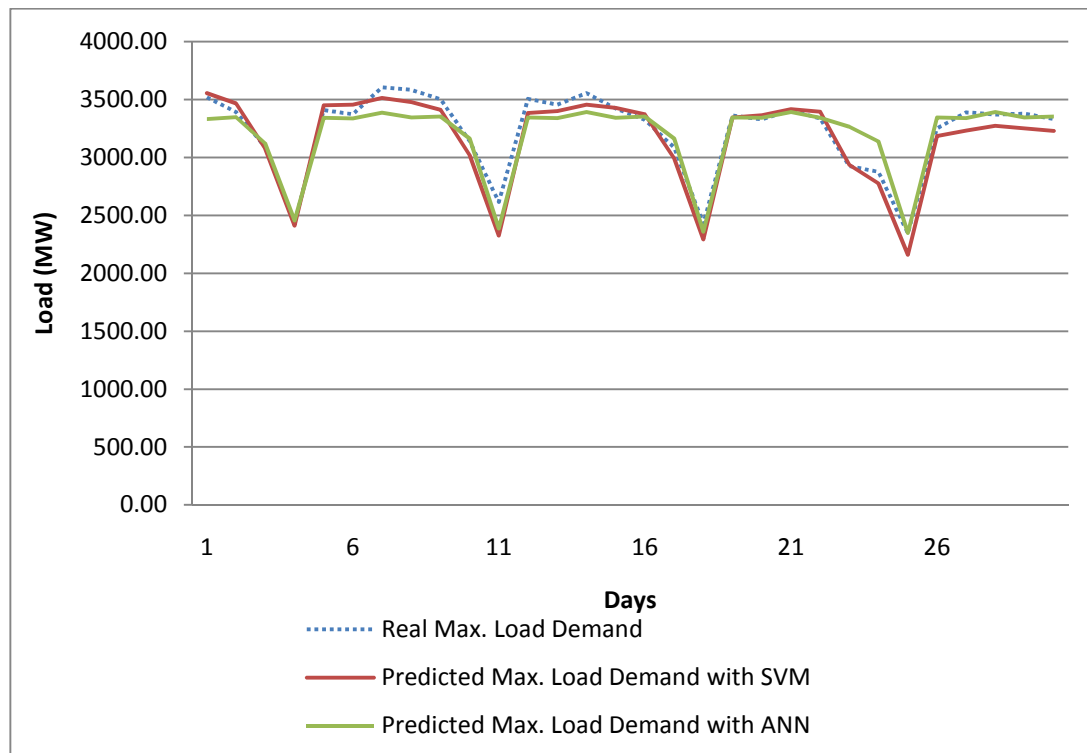


Figure 5.10: Comparison graph between real and predicted values with past load and EP

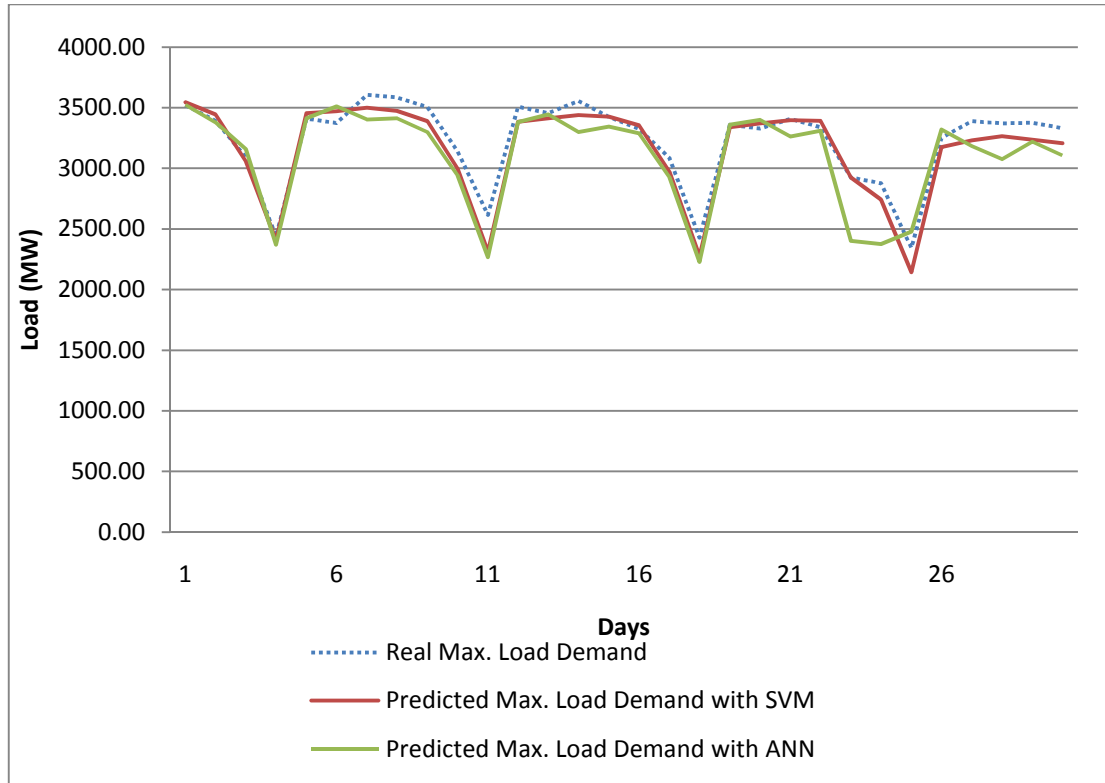


Figure 5.11: Comparison graph between real and predicted values with past load values

In general, all three models gave satisfactory results for predicting a load demand of an area with dynamic nature. Here, the main factor of the good forecast results is the well prepared training set. Since it contains the data of 3 previous years, the prediction ability of the programs is increased. It can be observed from Figure 5.9, Figure 5.10 and 5.11 that the prediction in all three models clearly projected the periodicity of the load demand on weekly basis. But as can be seen from the results in Table 5.3, SVM has given better results than ANN model in different data preparations. That means with the same training set, since SVM has a better ability to converge optimal solution than ANN, it gives better outputs. Therefore, it can be said that the learning ability of the SVM is suitable for electrical load prediction tasks, even in constantly changing areas.

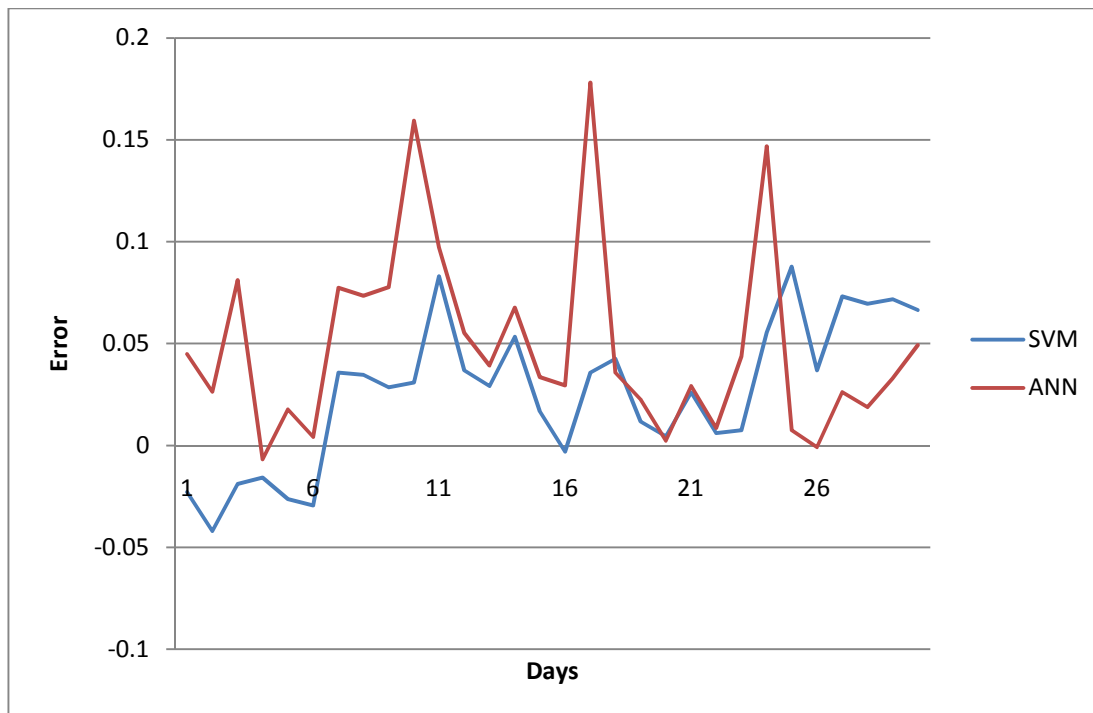


Figure 5.12: Error analysis for the first data set

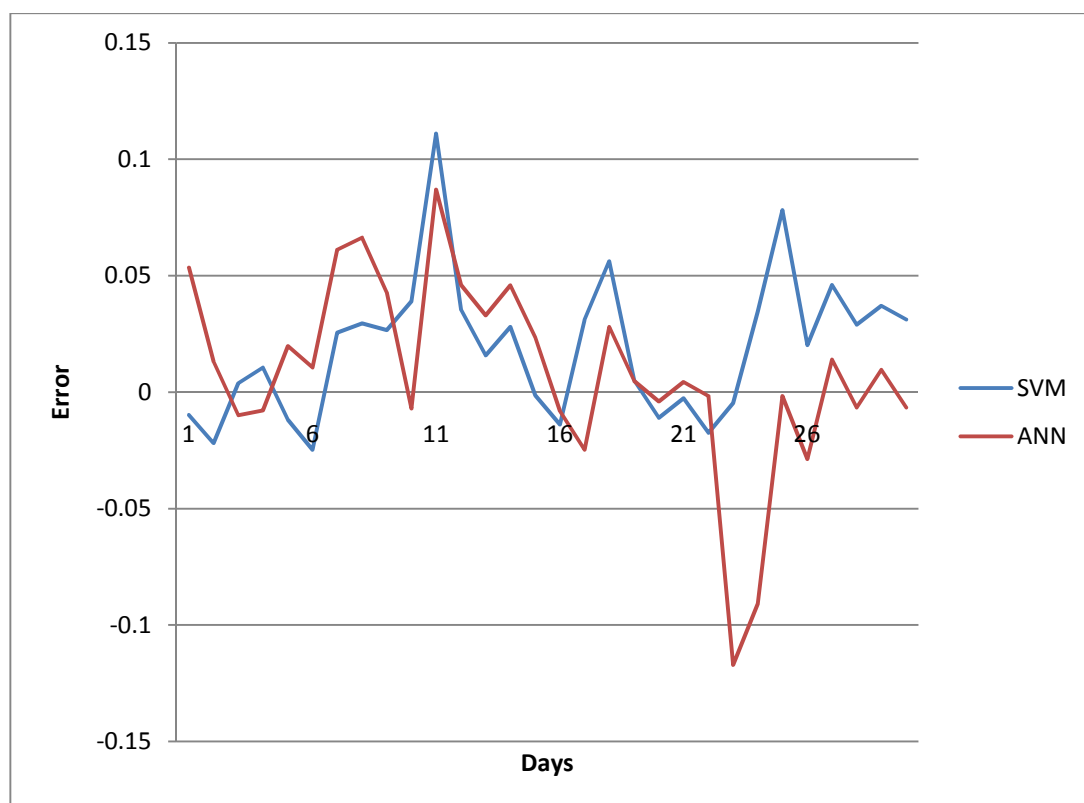


Figure 5.13: Error analysis for the second data set

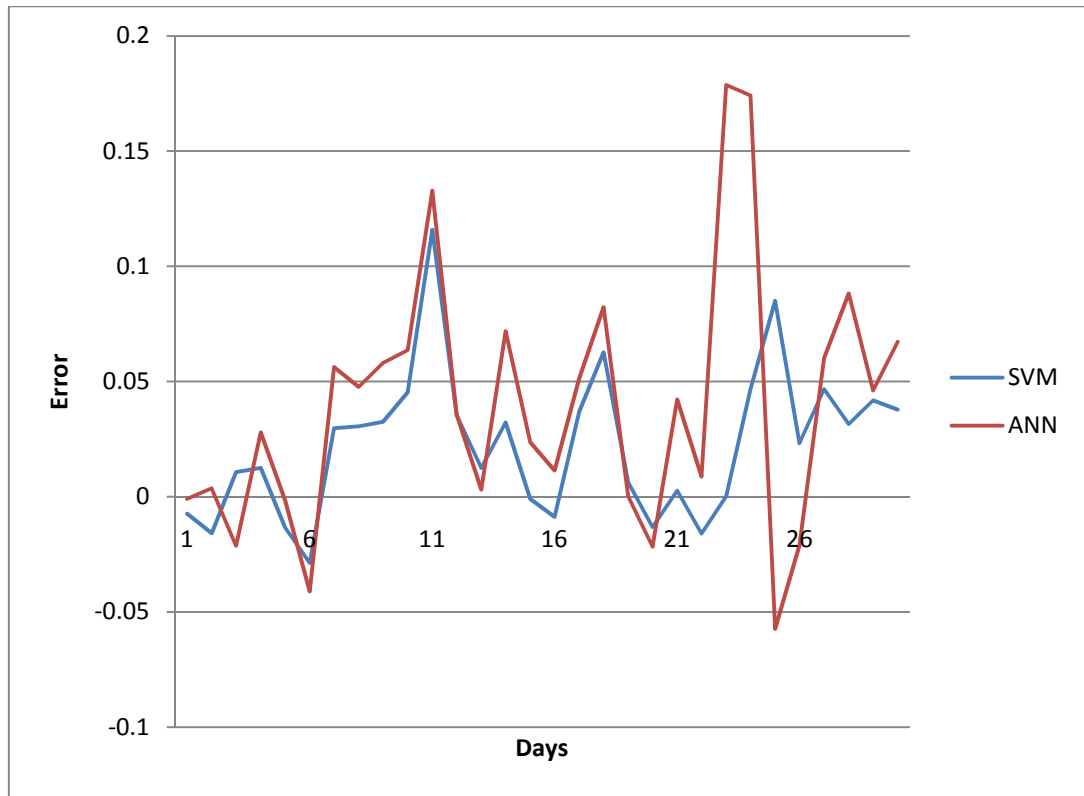


Figure 5.14: Error analysis for the third data set

The error analysis graphs in Figure 5.12, Figure 5.13 and Figure 5.14 show the general course of the error in all three data sets for the models SVM and ANN. Even though the forecasted demands are within acceptable range, it is clearly seen that, especially in dynamic, large areas, the weather conditions may not be considered as an attribute, in a mid-term load demand forecasting applications.

6 CONCLUSION

Machine learning is described as the ability of the computer to learn how to solve problems and develop behaviours based on empirical data. Support Vector Machines is one of the supervised learning methods in machine learning discipline which is based on statistical learning theory and risk minimization. Thus, the algorithm behind this methodology becomes suitable for real life applications.

Support Vector Machines were first applied in classification problems. The usage of kernel functions enables the learning algorithm to be applied in nonlinear cases. Today, SVMs have a wide variety of application areas from financial markets to environmental research.

SVM also offers a powerful technique for solving regression problems. Since the learning process is based on observed data, SVM estimates a function according to it. Thus, SVM is independent of the model and it has a good convergence to the optimal solution. Moreover, it has small number of parameters to be adjusted which enables a computationally efficient method. But this efficiency is proportional to the empirical adjusted parameters. Therefore, the main requirement of this technique is experience.

In this study, SVM is used in solving a medium term load demand forecasting problem using the program Libsvm. The main reason for using SVM is that it can easily model the load curve; the relation between the load and the dynamics that change the load demand, such as calendar days, special days (holidays, festive, etc.), temperature, economic and demographic factors. Load forecasting is a necessity for future planning in electrical generation, transmission and distribution. Especially in deregulated energy markets, medium-term load forecasting has gained more importance since it can be used in defining the pricing policies, organizing fuel supply etc. which is vital for the investor.

In medium-term electrical load demand forecast generally weekly patterns, seasonal patterns and economic factors are taken into account. In order to determine the effects of these kind of factors, three data sets are constructed. The first data set is

prepared with the daily peak load data, average daily temperature, and electrical price as the economic factor from year 2006 through 2009. The second data set, only daily average temperature values are extracted. In the third data set, in addition to daily average temperatures, electricity price values are extracted and a model is obtained with only past load values and calendar days. The reason to separate temperature values is to determine whether it is necessary to take weather conditions into consideration in an electrical load forecasting application for next 30 days of a constantly changing, dynamic area such as Istanbul European Side. In addition to these parameters, the weekly parameters are projected via calendar days, such as week days, weekends and holidays. During the training only the values of the months April, May and June are used. The same values of the year 2009 on April, May and June are used for testing of the models.

When constructing the SVM regression model, Gaussian Radial Basis Function is used as the kernel function in the algorithm. The parameters; cost of error C kernel parameter γ and width of ε -insensitive tube ε are selected with the 5-fold cross validation using Libsvm's cross validation feature.

The same application is made with ANN using MATLAB Neural Network Toolbox with the same three data sets as they are applied to SVM. A feed-forward back-propagation with two hidden layers architecture is used when constructing the ANN models. The training algorithm Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation is selected. The parameters for the hidden units, momentum constant and learning rate are determined by trial& error and experience.

With the obtained three models April 2010 daily peak load demand is predicted using SVM and ANN. The results are compared to real maximum daily load demand values of April 2010 and the errors are computed with mean absolute percentage error, MAPE.

In the literature, vast of techniques are used for medium-term load forecasting applications. As it is stated in many of those, the key point is the selection of technique and the parameters that will be used in the data set. For a medium-term electrical load demand forecasting application, one can use many economical and demographical data in order to project the growth conditions of the related area. Since finding the accessible data for solving a forecasting problem is usually

difficult, the user should select the most suitable technique according to the data that she/ he has at hand.

This study shows that considering weather conditions in a medium-term electrical load forecasting application, especially for dynamic areas, is decreasing the general accuracy of the prediction. Thus, it can be concluded that depending on the prediction area, meteorological conditions may not be included in a medium-term load forecasting application.

Apart from weather conditions, naturally, there are many parameters that can be used as economic/demographic indicators of the related area. In this study electricity price is chosen as the indicator of the economic conditions, in the course of the related time. The main reason to use electricity price as an economical factor is the availability of this data from TUIK. It is believed that with accurate population growth values especially for Istanbul European Side, better results can be achieved for medium-term electrical load forecasts.

Under the light of this study, it can be said that the SVM regression technique has better generalization capability than ANN and suitable for mid-term electrical load forecasting applications, even in constantly changing, crowded metropolitan areas, as Istanbul European Side.

REFERENCES

- [1] **Wang, X.**, 1994. Modern Power System Planning, McGraw-Hill, London.
- [2] **Hor, C.-L., Watson, S.J., Majithia, S.**, 2005. Analyzing the impact of weather variables on monthly electricity demands. *IEEE Transactions on Power Systems*, Vol. **20**, No. 4, pp. 2078–2085.
- [3] **Bruhns, A., Deurveilher, G., Roy, J.-S.**, 2005. A non-linear regression model for midterm load forecasting and improvements in seasonality. *15th Power Systems Computation Conference*.
- [4] **Amjady, N.**, 2001. Short-term hourly load forecasting using time-series modelling with peak load estimation capability. *IEEE Transactions on Power Systems*, Vol. **16**, No. 3, pp. 498–505.
- [5] **Salama, H. A.E., El-Gawad, A. F. A., Sakr, S.M., Mohamed, E. A., Mahmoud, H. M.**, 2009. Applications on medium-term forecasting for loads and energy scales by using Artificial Neural Network. *20th International Conference and Exhibition on Electricity Distribution - Part 1*, CIRED 2009, June.
- [6] **A. G. Bakirtzis, V. Petridis, S. J. Kiartzis, and M. C. Alexiadis**, 1996. A neural network short term load forecasting model for the Greek power system. *IEEE Transactions on Power Systems*, Vol. **11**, No. 2, pp. 858–863.
- [7] **Yalçınöz, T., Herdem, S., Eminoğlu, U.**, 2002. Yapay Sinir Ağları İle Niğde Bölgesinin Elektrik Yük Tahmini, *ELECO'2002*, Bursa.
- [8] **de Aquino, R.R.B., Noberto Neto, O., Lira, M.M.S., Ferreira, A.A., Santos, K.F.**, 2007. Using genetic algorithm to develop a neural-network-based load forecasting. *Lecture Notes in Computer Science*, Part II, Vol. 4669, Springer, pp.738–747.
- [9] **K. H. Kim, H. S. Youn, and Y. C. Kang**, 2000. Short-term load forecasting for special days in anomalous load conditions using neural networks and fuzzy inference methods. *IEEE Transactions on Power Systems*, Vol. **15**, No. 2, pp. 559–565.
- [10] **Hahn, H., Meyer-Nieberg, S., Pickl, S.**, 2009. Electric load forecasting methods: Tools for decision making, *European Journal of Operational Research*, Vol. **199**, Issue 3, December, pp. 902-907.
- [11] **Campbell, P.R.J., Adamson, K.**, 2006. Methodologies for Load Forecasting. *3rd International IEEE Conference on Intelligent Systems*, September.
- [12] **Feinberg, E.A., Genethliou, D.**, 2005. Load forecasting. In *Applied Mathematics for Restructured Electric Power Systems: Optimization*,

Control and Computational Intelligence, Power Electronics and Power Systems. Springer, US, pp. 269–285.

- [13] **Alfares, H. K., Nazeeruddin, M.,** 2002. Electric load forecasting: literature survey and classification of methods. *International Journal of Systems Science*, Taylor& Francis Group, Vol. **33**, No. 1, pp. 23-34.
- [14] **Gross, G., Galiana, F. D.,** 1987. Short-Term Load Forecasting. *Proceedings of the IEEE*, Vol. **75**, No. 12.
- [15] **Kyriakides, E., Polycarpou, M.,** 2007. Short term electric load forecasting: A tutorial. In Trends in Neural Computation, Studies in Computational Intelligence, Vol. **35**, Ch. 16, pp. 391–418, Springer.
- [16] **Tzafestas, S., Tzafestas, E.,** 2001. Computational intelligence techniques for short-term electric load forecasting. *Journal of Intelligent and Robotic Systems* 31, pp. 7–68.
- [17] **Hippert, H.S., Pedreira, C.E., Souza, R.C.,** 2001. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, Vol. **16**, No. 1, pp. 44–55.
- [18] **Nilsson, N.J.,** 1996. Introduction to Machine Learning: An Early Draft of a Proposed Textbook, Stanford University.
- [19] **Cristianini, N., Shawe-Taylor, J.,** 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, UK, Cambridge University Press.
- [20] **Herbrich, R.,** 2002. Learning Kernel Classifiers: Theory and Algorithms, Massachusetts, The MIT Press.
- [21] **Sebe, N., Cohen, I., Garg, A., Huang, T.S.,** 2005. Machine Learning in Computer Vision, Springer Verlag, The Netherlands, May.
- [22] **Elmas, Ç.,** 2003. Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama), İstanbul, SeçkinYayıncılık.
- [23] **Mitchell, T.,** 1997. Machine Learning, Mc. Graw-Hill Science/ Engineering/ Math, March.
- [24] **Wang , X., Zhong, Y.,** 2003. Statistical learning theory and state of the art in SVM, *The Second IEEE International Conference on Cognitive Informatics, Proceedings*, August.
- [25] **Vapnik, V.,** 2000. The Nature of Statistical Learning Theory, Second Ed., New York, Springer.
- [26] **Gunn, S.R.,** 1998. Support Vector Machines for Classification and Regression, Technical Report, University of Southampton, May.
- [27] **Url-1** <<http://www.svms.org/srm/>>, accessed at 01.04.2011.
- [28] **Schölkopf, B., Burges, C.J.C, Smola, A.J.,** 1999. Advances in Kernel Methods: Support Vector Learning, The MIT Press, Cambridge, London, England.
- [29] **Nasien, D., Yuhaniz, S.S., Haron, H. ,** 2010. Statistical Learning Theory and Support Vector Machines, *2010 Second International Conference on Computer Research and Development*, May.

- [30] **Url-2** < <http://www.svms.org/history.html>>, accessed at 05.11.2010.
- [31] **Smola, A.J., Schölkopf, B.**, 2004. A tutorial on support vector regression, *Journal Statistics and Computing*, Vol. **14**, Issue 3, August.
- [32] **Kecman, V.**, 2001. Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models, MIT Press, Cambridge, Massachusetts, England.
- [33] **Erastö, P.**, 2001. Support Vector Machines-Backgrounds and Practice, Academic Dissertation for the Degree of Licentiate of Philosophy, Rolf Nevanlinna Institute, Helsinki.
- [34] **Deloitte**, 2010. Türkiye Elektrik Enerjisi Piyasası 2010-2011, Beklentiler ve Gelişmeler, Retrieved November 02, 2010, from http://www.deloitte.com/view/tr_TR/tr/sektorler/enerjivedogalkaynaklar/115bd4c6937b9210VgnVCM100000ba42f00aRCRD.htm
- [35] **Şahin, V.**, 1993. Türkiye Bölgesel Elektrik Talebi (1990-2010), Türkiye Elektrik Kurumu, Ankara.
- [36] **Daneshi, H., Shahidehpour, M., Choobbari, A.L.**, 2008. Long-Term Load Forecasting in Electricity Market, *IEEE International Conference on Electro/Information Technology*, June.
- [37] **Haykin S.**, *Neural Networks*, Upper Saddle River, N. J., Prentice Hall, 1999.
- [38] **Khotanzad, A., Afkhami- Rohani, R., Maratukulam, D.**, 1998. ANNSTLF-Artificial Neural Network Short-Term Load Forecaster generation three, *IEEE Transactions on Power Systems*, Vol. **13**, No. 4, pp. 1413 - 1422, November.
- [39] **Chen, B.-J., Chang, M.-W., Lin, C.-J.**, 2004. Load forecasting using support vector machines: A study on EUNITE competition 2001, *IEEE Trans. Power Syst.*, Vol. **19**, No. 4, pp. 1821-1830, November.
- [40] **Mohandes, M.**, 2002. Support vector machines for short-term load forecasting, *Int. J. Energy Res.*, Vol. **26**, No. 4, pp. 335–345, March.
- [41] **Guo, Y.-C., Niu, D.-X., Chen, Y.-X.**, 2006. Support Vector Machine Model In Electricity Load Forecasting, *Fifth International Conference on Machine Learning and Cybernetics*, Dalian, August.
- [42] **Li, X., Sun, C., Gong, D.**, 2005. Application of support vector machine and similar day method for load forecasting. *Lecture Notes Computer Science*, Vol. **3611**, pp. 602–609.
- [43] **Zhang, M.-G.**, 2005. Short-term load forecasting based on support vector machine regression, *Proc. 4th Int. Conf. on Machine Learning and Cybernetics*, Vol. **7**, pp. 4310–4314, August.
- [44] **Escobar, A.M., Perez, L.P.**, 2008. Application of Support Vector Machines and ANFIS to the Short-term Load Forecasting, *Transmission and Distribution Conference and Exposition: Latin America*, IEEE/PES.
- [45] **Guo, X.C., Liang, Y.C., Wu, C.G., Wang, H.Y.**, 2006. Electric Load Forecasting using SVMS, *International Conference on Machine Learning and Cybernetics*, March.

- [46] **Sapankevych, N. I., Sankar, R.**, 2009. Time series prediction using support vector machines: a survey, *IEEE Computational Intelligence Magazine*, Volume 4 , Issue 2, May.
- [47] **Aksel, F.**, 2000. Regresyon analizi ve yapay sinir ağı yöntemleri ile uzun dönem yük tahmini. Master's Thesis, ITU Institute of Science and Technology, Istanbul.
- [48] **Yoldaş, U.C.**, 2006. Elektrik enerjisinde yük tahmini yöntemleri ve Türkiye'nin 2005–2020 yılları arasındaki elektrik enerjisi talep gelişimi ve arz planlaması. Master's Thesis, Gazi Universtiy Institute of Science and Technology, Ankara.
- [49] **Ceylan, G.**, 2004. Yapay Sinir Ağı İle Kısa Dönem Yük Tahmini, Master's Thesis, ITU Institute of Science and Technology, Istanbul.
- [50] **Pınarbaşı, M.**, 2009. Elektrik Enerji Sistemlerinde Talep Tahmin Yöntemleri Ve Yapay Sinir Ağları Uygulaması, Master's Thesis, Institute of Science and Technology, Istanbul University.
- [51] **Tolun, S.**, 2008. Destek Vektör Makineleri: Banka Başarısızlığının Tahmini Üzerine Bir Uygulama. PhD. Thesis, Institute of Social Sciences, Istanbul University.
- [52] **Url-3** <http://videlectures.net/mlas06_mitchell_sla/>, Semisupervised Learning Approaches, accessed at 07.10.2010.
- [53] **Demuth, H., Hagan, M., Beale, M.**, 2010. Neural Network Toolbox™ 7 User's Guide, the Math Works, Inc.
- [54] **Chang, C.-C., Lin, C.-J.**, LIBSVM: a Library for Support Vector Machines <http://www.csie.ntu.edu.tw/~cjlin/Libsvm/>

APPENDICES

APPENDIX A : Libsvm codes.

APPENDIX B : MATLAB Neural Networks Toolbox Codes.

APPENDIX C: Prediction Results

APPENDIX A

Libsvm is an integrated software for support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM).

Input variables can be scaled using svm-scale command as follows;

```
svm-scale [options] data_filename
options:
-l lower: x scaling lower limit (default -1)
-u upper: x scaling upper limit (default +1)
-y y_lower y_upper: y scaling limits (default: no y scaling)
-s save_filename: save scaling parameters to save_filename
-r restore_filename: restore scaling parameters from
restore_filename
```

“grid.py” is a parameter selection tool using the RBF kernel. It uses cross validation to estimate the accuracy of each parameter combination in the specified range and helps to decide the best parameters for the problem. The parameters γ -gamma value in kernel function, C - cost of the error, and ε - the width of ε -insensitive tube can be found with the command below;

```
grid.py [-log2c begin,end,step] [-log2g begin,end,step] [-log2p
begin,end,step] [-v fold] [-svmtrain pathname] [-gnuplot pathname]
[-out pathname] [-png pathname][additional parameters for svm-train]
dataset
```

The program conducts v-fold cross validation using parameter C , γ and ε as;

$= 2^{\text{begin}}, 2^{(\text{begin}+\text{step})}, \dots, 2^{\text{end}}$.

Training function can be used as follows;

```
svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
    0 -- C-SVC
    1 -- nu-SVC
    2 -- one-class SVM
    3 -- epsilon-SVR
    4 -- nu-SVR
-t kernel_type : set type of kernel function (default 2)
    0 -- linear: u'*v
    1 -- polynomial: (gamma*u'*v + coef0)^degree
    2 -- radial basis function: exp(-gamma*|u-v|^2)
    3 -- sigmoid: tanh(gamma*u'*v + coef0)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
```

```
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR
(default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR
(default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR
(default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking: whether to use the shrinking heuristics, 0 or 1
(default 1)
-b probability_estimates: whether to train a SVC or SVR model for
probability estimates, 0 or 1 (default 0)
-wi weight: set the parameter C of class i to weight*C, for C-SVC
(default 1)
```

After the training process, a model is obtained. Using this model, prediction can be done with the command;

```
svm-predict [options] test_file model_file output_file
```

options:

```
-b probability_estimates: whether to predict probability estimates,
0 or 1 (default 0); for one-class SVM only 0 is supported
```

Here, `model_file` is the model file generated by `svm-train`, `test_file` is the test data that user wants to predict. As a result, `svm-predict` will produce output in the `output_file`.

APPENDIX B

The following MATLAB Neural Network Toolbox codes are used for load demand forecasting application.

```
ni=17;%number of the input neurons
no=1;%number of outputs
np=266;%%number of the training patterns
nh1=40;%%number of hidden layer neurons
nh12=10;%%number of hidden layer neurons
load TrainInput.mat;%loading the output pattern vector
load traintarget.mat;%loading the output pattern vector

x=traininput';
t=traintarget';

net=newff(minmax(x), [nh1 nh12
no],{'logsig','logsig','logsig'},'traingdx');
net.trainParam.lr=0.01;
net.trainParam.mc=0.9;
net.trainParam.epochs=10000;
net.trainParam.goal=0.0001;
net.trainParam.show=100;
net=train(net,x,t);
```

After the network is obtained, the following commands are used for predicting the values;

```
load TestInput.mat
xnew=testinput';
output=sim(net,xnew)
```


APPENDIX C

Table C.1 : Real and prediction results with past load, temperature and EP

Date	Real Max. Load Demand (MW)	Predicted Max. Load Demand (MW) with SVM	Predicted Max. Load Demand (MW) with ANN
Thursday, 1 April 2010	3520.18	3600.45	3361.90
Friday, 2 April 2010	3392.23	3534.38	3302.40
Saturday, 3 April 2010	3090.15	3148.26	2839.00
Sunday, 4 April 2010	2437.09	2475.20	2453.70
Monday, 5 April 2010	3409.07	3498.34	3348.50
Tuesday, 6 April 2010	3373.12	3472.27	3358.90
Wednesday, 7 April 2010	3606.03	3476.80	3326.90
Thursday, 8 April 2010	3583.20	3458.53	3319.70
Friday, 9 April 2010	3504.11	3404.01	3231.40
Saturday, 10 April 2010	3142.03	3044.65	2641.40
Sunday, 11 April 2010	2615.01	2397.70	2361.00
Monday, 12 April 2010	3507.00	3377.78	3313.20
Tuesday, 13 April 2010	3455.12	3354.41	3319.60
Wednesday, 14 April 2010	3554.16	3364.36	3313.70
Thursday, 15 April 2010	3423.97	3366.51	3309.20
Friday, 16 April 2010	3327.09	3337.30	3229.00
Saturday, 17 April 2010	3086.14	2975.38	2536.40
Sunday, 18 April 2010	2428.05	2324.80	2341.20
Monday, 19 April 2010	3360.64	3321.26	3285.20
Tuesday, 20 April 2010	3328.17	3313.00	3320.50
Wednesday, 21 April 2010	3406.76	3318.43	3307.70
Thursday, 22 April 2010	3338.09	3317.57	3309.50
Friday, 23 April 2010	2924.32	2902.09	2796.00
Saturday, 24 April 2010	2875.82	2716.43	2453.60
Sunday, 25 April 2010	2343.46	2137.23	2325.40
Monday, 26 April 2010	3252.12	3132.08	3254.80
Tuesday, 27 April 2010	3388.23	3140.34	3299.20
Wednesday, 28 April 2010	3371.14	3136.54	3307.50
Thursday, 29 April 2010	3377.25	3134.48	3266.30
Friday, 30 April 2010	3332.17	3110.92	3167.90

Table C.2 : Real and prediction results with past load and EP

Date	Real Max. Load Demand (MW)	Predicted Max. Load Demand (MW) with SVM	Predicted Max. Load Demand (MW) with ANN
Thursday, 1 April 2010	3520.18	3554.46	3331.70
Friday, 2 April 2010	3392.23	3466.05	3347.90
Saturday, 3 April 2010	3090.15	3078.36	3120.70
Sunday, 4 April 2010	2437.09	2411.23	2456.30
Monday, 5 April 2010	3409.07	3449.21	3341.40
Tuesday, 6 April 2010	3373.12	3456.37	3337.50
Wednesday, 7 April 2010	3606.03	3514.05	3385.70
Thursday, 8 April 2010	3583.20	3477.20	3345.40
Friday, 9 April 2010	3504.11	3410.63	3353.90
Saturday, 10 April 2010	3142.03	3019.43	3164.10
Sunday, 11 April 2010	2615.01	2324.78	2387.60
Monday, 12 April 2010	3507.00	3382.90	3345.80
Tuesday, 13 April 2010	3455.12	3400.38	3341.10
Wednesday, 14 April 2010	3554.16	3454.51	3391.10
Thursday, 15 April 2010	3423.97	3428.87	3343.70
Friday, 16 April 2010	3327.09	3373.16	3354.10
Saturday, 17 April 2010	3086.14	2990.33	3162.40
Sunday, 18 April 2010	2428.05	2291.48	2360.00
Monday, 19 April 2010	3360.64	3344.14	3345.10
Tuesday, 20 April 2010	3328.17	3364.77	3341.20
Wednesday, 21 April 2010	3406.76	3415.84	3392.30
Thursday, 22 April 2010	3338.09	3396.23	3343.50
Friday, 23 April 2010	2924.32	2937.55	3266.40
Saturday, 24 April 2010	2875.82	2776.57	3137.50
Sunday, 25 April 2010	2343.46	2159.77	2346.80
Monday, 26 April 2010	3252.12	3186.39	3345.60
Tuesday, 27 April 2010	3388.23	3232.08	3340.70
Wednesday, 28 April 2010	3371.14	3273.28	3393.30
Thursday, 29 April 2010	3377.25	3251.75	3344.60
Friday, 30 April 2010	3332.17	3228.18	3354.20

Table C.3 : Real and prediction results with past load

Date	Real Max. Load Demand (MW)	Predicted Max. Load Demand (MW) with SVM	Predicted Max. Load Demand (MW) with ANN
Thursday, 1 April 2010	3520.18	3545.69	3523.60
Friday, 2 April 2010	3392.23	3445.54	3379.90
Saturday, 3 April 2010	3090.15	3057.04	3155.80
Sunday, 4 April 2010	2437.09	2406.54	2369.00
Monday, 5 April 2010	3409.07	3454.63	3414.20
Tuesday, 6 April 2010	3373.12	3469.89	3511.70
Wednesday, 7 April 2010	3606.03	3498.79	3403.00
Thursday, 8 April 2010	3583.20	3473.58	3412.40
Friday, 9 April 2010	3504.11	3390.08	3300.40
Saturday, 10 April 2010	3142.03	2999.73	2942.10
Sunday, 11 April 2010	2615.01	2311.82	2267.50
Monday, 12 April 2010	3507.00	3382.90	3381.70
Tuesday, 13 April 2010	3455.12	3411.91	3444.50
Wednesday, 14 April 2010	3554.16	3439.36	3298.90
Thursday, 15 April 2010	3423.97	3427.16	3343.00
Friday, 16 April 2010	3327.09	3355.95	3288.90
Saturday, 17 April 2010	3086.14	2972.07	2927.40
Sunday, 18 April 2010	2428.05	2275.82	2228.40
Monday, 19 April 2010	3360.64	3339.95	3360.60
Tuesday, 20 April 2010	3328.17	3371.73	3400.00
Wednesday, 21 April 2010	3406.76	3397.80	3263.20
Thursday, 22 April 2010	3338.09	3391.56	3308.70
Friday, 23 April 2010	2924.32	2923.74	2401.70
Saturday, 24 April 2010	2875.82	2741.57	2375.20
Sunday, 25 April 2010	2343.46	2143.73	2477.30
Monday, 26 April 2010	3252.12	3176.41	3320.50
Tuesday, 27 April 2010	3388.23	3229.95	3184.30
Wednesday, 28 April 2010	3371.14	3264.73	3073.80
Thursday, 29 April 2010	3377.25	3235.64	3221.40
Friday, 30 April 2010	3332.17	3206.11	3107.70

CURRICULUM VITAE

Candidate's full name: Dilara DEMREN

Place and date of birth: 19 January 1985/ Izmir

Permanent Address: Goruntu Sok. No:8/5 Mecidiyekoy-ISTANBUL

Universities and Colleges attended:

BSc- Istanbul Technical University/ Electrical Engineering

MSc- Technische Universität Kaiserslautern (Erasmus Student Exchange Programme)