ENERGY LOAD FORECASTING USING HYBRID K-MEANS CLUSTERING – SVM

A DISSERTATION/THESIS

SUBMITTED IN PARTIAL FULFILMENENT OF THE

REQUIREMENTS

FOR THE AWARD OF DEGREE OF

MASTER OF TECHNOLOGY

IN

CONTROL AND INSTRUMENTATION

Submitted By
Vineet Hooda
Roll No.: 2k14/C&I/24

Under the supervision and guidance of

Dr. Bharat Bhushan

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I, Vineet Hooda, Roll No. 2k14/C&I/24 student of M. Tech. (Control and Instrumentation), hereby declare that the dissertation/project titled "Energy Load Forecasting using hybrid K-means clustering – SVM" under the supervision of Dr. Bharat Bhushan of Electrical Engineering Department Delhi Technological University in partial fulfillment of the requirement for the award of the degree of Master of Technology has not been submitted elsewhere for the award of any Degree.

Place: Delhi **Vineet Hooda**
Date: 29/08/2016

Dr. Bharat Bhushan
SUPERVISOR
Associate Professor EED, DTU

ACKNOWLEDGEMENT

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor **Dr. Bharat Bhushan** for his invaluable guidance, encouragement and patient reviews. He kept on boosting me with time, to put an extra ounce of effort to realize this work. With their continuous inspiration only, it becomes possible to complete this dissertation.

I would also like to take this opportunity to present my sincere regards to all the faculty members of the Department for their support and encouragement. I am grateful to my parents for their moral support all the time; they have been always around to cheer me up, in the odd times of this work. I am also thankful to my classmates for their unconditional support and motivation during this work.

It is a great pleasure to have the opportunity to extend my heartiest felt gratitude to everybody who helped me throughout the course of this project.

Vineet Hooda

2k14/C&I/24

ABSTRACT

In Power Industry, Energy Load Forecasting is an important aspect. Determining the future demand for load in advance is very important. Once the company knows the future load, it can take much better investment decisions and decisions about expansion, maintenance and buying energy from the generating companies. Having some knowledge of future energy consumption is, therefore, an absolute necessity. Power distribution companies, therefore, require tools that can predict the load. Prediction of electrical load is difficult. A number of classical prediction models are available for this. But these models suffer from the problem of requirement of linearity and seasonality. For predicting electric load we have used K-Means Clustering and SVM. The results obtained using the technique are compared with energy load forecasting using SVM only and the performance of hybrid K-means clustering – SVM is found to be consistently better.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

1.1 Literature review

Energy forecasting refers to forecast of load in energy industry. Energy forecasting is not limited to this definition. It also includes forecasting of electric prices, fuel (like coal, oil, natural gas) prices, non-conventional energy resources, etc.

Power utility monopolies used short-term forecasting for ensuring supply of electricity and long-term forecasting for upgrading capacity of the system and for investing in new capacity [40][52]. Since early 1990s there has been deregulation of electricity around the world and there has been an introduction of competitive electricity market which has reshaped the entire power industry. Many countries have market rules for trade of electricity [49]. The load and price forecast data is very important for decision making of power utilities. It is very critical to get accurate load forecast as over or under estimation can eventually lead to bankruptcy of the power utility in extreme cases [50][51]. Utilities cannot pass this cost to the costumer and as a result their vulnerability is very high [34].

There are two types of studies for the forecast – point and probabilistic. The former is based on expected value or best guess of the spot price whereas the latter is based on interval and density. While there has been much emphasis on point forecast in the past, the probabilistic forecast has not seen much study [53]. This is, however, fast changing as the Global Energy Forecasting Competition in 2012 was based on point forecast of load and wind power whereas in 2014 it put emphasis on probabilistic forecast of load, electric prices and wind & solar power.

The wholesale electricity prices are very volatile. Any utility or industrial consumer who is able to forecast the electricity prices with a high level of accuracy can adjust its strategy and can modify its consumption pattern and as a result reduce the risks and maximize its profit. Since the load and electric price forecast are used by many departments of a utility, it is difficult to quantify it. An estimate of savings from a 1% reduction in MAPE (Mean Absolute Percentage Error) for a 1GW peak load is: [54]

- $500,000 per year from long-term load forecasting,
- $300,000 per year from short-term load forecasting,
- $600,000 per year from short-term load and price forecasting.

There is no clear boundary to define short-term, medium-term and long-term load forecasting:

- Short-term load forecasting includes forecasting from few minutes to few days ahead. This is important for day-to-day market operations. Another term known as very short-term load forecasting is used for lead time in minutes.
- Medium-term load forecasting is done from a few days to a few weeks. It is used for balance sheet calculations. In electric price forecast the evaluation is not based on point forecast but probabilistic forecast.
- Long-term load forecasting is done in months or quarters or years. It is used for investment profitability analysis.

Load forecasting generally means the power demand (in kW) or energy demand (in kWh). The magnitude of power and energy is same for an hour, so they are treated as same. The quantity of interest is the hourly load. However, hourly, weekly and monthly values of load are also important.

There is an estimate growth of 3-7% of electric load annually. There are various factors that affect power generation & consumption. They are management of load, energy exchange, non-conventional energy, etc. Load forecasting, thus, is very important for decisions like planning & operation. Network reliability can be improved and equipment failures and blackouts can be prevented with timely implementation of such decisions. There are two type of load forecasting, Spatial & Temporal. Spatial forecasting is forecasting of future load of a region, state or country. Forecasting of future load for a supplier or a group of consumers in hours, days or months is Temporal forecasting. Temporal forecasting has four types – very short, short, medium and long term. For planning the growth of generation capacity long term forecasting is done. On the basis of this it is decided whether up gradation of existing line is required or new line is to be set up. Load forecast at the height of summer or winter season is done using medium term forecast, which is carried out for few weeks or months in advance.

Short term load forecasting is generally done for one week. Load for each hour is calculated along with daily peak load and weekly generation of energy. Many real time operations and power generation control are done based on short term load forecasting. The real time operations include security analysis, energy exchange with other utilities, energy planning, etc. Accuracy of the load forecast determines the reliability and economy of operations. Load dispatch center must anticipate the energy load pattern in advance so that to have sufficient generation to meet consumer demand. Over-estimation of load will result in starting of too many generating units. This will cause an unwanted increase of the reserve and operating cost. Under-estimation will result in instability of the system because of the failure to provide adequate spinning reserve and standby reserve. Errors in load forecasting can lead to suboptimal decisions of unit commitment. Therefore, correct load forecast is a very important aspect of power system.

Many methods of load forecasting have been used in past years, with varying degree of accuracy. They may be classified as causal models and time series models. In causal, load is modeled as a function of external factors like temperature, humidity, etc. In time series models, we model the load as a function of past values observed. Some models of the 1st class are ARMAX, techniques of optimization, curve-fitting method, etc. Some of the methods of 2nd class are linear and non-linear dynamic models, models based on Kalman filter, multiplicative and threshold auto-regressive models. However, despite of these large numbers of models, the most used causal models are the models that decompose load into basic weather dependent elements and the linear ones. The above models are attractive because of their linearity and physical interpretation can be attached to them so that operators and engineers understand their behavior easily. The problem is they are linear models and the systems that they are trying to explain are known to be non-linear functions of external variables.

In recent years, most of the research on energy load forecasting is done with the application of artificial intelligence techniques. Many new systems have been tried and compared to the classical systems. Fuzzy inference and neural-fuzzy methods have also been tried. But most of the research using artificial intelligence technique has been carried out in artificial neural network (ANNs). The first use of artificial neural network for energy load forecasting was done in late 1980s. Taking a cue from the number of papers that are published on load forecasting using artificial neural network one has to say that it has not turned into a passing fad as once was perceived.

Factors like time, weather, customers, etc. are taken into account for Short-term load forecast. For medium-term and long-term forecast, factors such as historical data, weather, customer categories, data of sale of appliances in an area and age of the appliances, demographic data of that area are used. In time factors, we take into account the time of the day, month or year. Loads are generally different on weekdays and weekends. Weekdays adjacent to weekends such as Monday and Friday can also have structurally different loads. This is particularly true during the day time. Holidays are generally difficult to forecast because of their occurrence being low. The load is also influenced by weather conditions. In short-term forecast, the weather data is of paramount importance. Temperature and humidity are the two most important variables used in forecasting. Among all the weather variables used, the temperature humidity index (THI) and the wind chill index (WCI) are widely used by the utilities. Temperature-humidity index is for summer heat and likewise Wind-chill index is for winter chill. There are different types of consumers such as domestic and commercial. The energy use pattern is different for different classes but the pattern is very much similar for a particular class. Hence, the utilities generally differentiate load on class basis [20].

1.2 Methods of Forecasting

Over the last few years a number of methods have been developed for load forecasting. They include:

- similar day approach
- regression models
- time series models
- artificial neural networks
- expert system models
- fuzzy inference system
- statistical learning models

The development of sophisticated mathematical tools will provide more accurate forecasting methods. Statistical approaches usually require a mathematical model that represents load as function of different factors such as time, weather, and customer class.

1.2.1 Medium- and long-term load forecasting methods

The end-use and econometric modeling and their mix are the frequently utilized techniques for medium-term and long-term forecasting. Depictions of apparatuses utilized by clients, the house size, the equipment's age, technology advancement, client conduct, and populace elements are normally incorporated into the statistical models and simulation models in view of the end-use approach. Moreover, monetary elements, for example, per capita income, occupation levels, and power costs are incorporated into econometric models. These models are frequently utilized as a part of mix with the end-use approach. Long term forecasting incorporates the gauges on the populace changes, monetary advancement, mechanical development, and innovation improvement.

1.2.1.1 End-use models:

The end-use approach specifically assesses energy utilization by utilizing broad data on end-use and end-users, for example, apparatuses, the client use, age of the appliances, house sizes, etc. Factual data about clients alongside dynamics of change is the basis of the forecasting. End-use models concentrate on the different utilization of power in the domestic and commercial sector. These models depend on the rule that power demand is derived from client's demand for light, cooling, warming and so on. In this way end-use models explain power demand as a component of the number of appliances in use [18]. Ideally this methodology is very precise. In any case, it is sensitive to the nature of end-use data. For instance, in this technique the distribution of age of the equipment is vital for specific type of appliances. End-use load forecasting requires less information about historical data and more data about clients and their appliances.

1.2.1.2 Econometric models:

The econometric methodology consolidates financial hypothesis and statistical models for determining power demand. The methodology estimates the connections between power consumption (dependent variables) and components affecting power consumption. These connections are evaluated by the least square technique or time series model.

One of the choices in this structure is to total the econometric methodology, when consumption in various sectors (domestic, commercial and so on.) is ascertained as a function of climate, financial and other different variables, and after that estimates are collected utilizing recent historical data. Incorporation of the econometric methodology into the end-use model brings behavioral parts into the end-use equations.

1.2.1.3 Statistical model-based learning:

The end-use and econometric techniques require a lot of data significant to equipment, clients, financial matters, and so on. Their application is complex and requires human interpretation. Also such data is frequently not accessible with respect to specific clients and a utility keeps and backs a profile of a "normal" client or normal clients for various kinds of clients. The issue emerges if the utility needs to forecast next year’s load for sub-regions, which are regularly called load pockets. For this situation, the amount of work that ought to be done increases relatively with the number of load pockets. Moreover, end-use profiles and econometric information for various load pockets are generally diverse. The attributes for specific zones might be unique in relation to the normal characteristics for the utility and may not be accessible.

With a specific goal to simplify medium-term forecasting, make them more exact, and keep away from the use of the inaccessible data, Feinberg came up with a statistical model that learns the load model parameters from the historical data available.

1.2.2 Short-term load forecasting methods

For Short-term load forecasting, a large number of Statistical and Artificial Intelligence techniques have been formulated.

1.2.2.1 Similar-day approach:

This methodology depends on historical data of days within the last few years with comparative attributes to the forecast day. Comparable qualities include climate, day of the week, month of the year. The load of a comparative day is considered as a forecast. Rather than a single comparable day, the forecast can be a linear combination or regression methods that can incorporate several comparable days. The pattern coefficients can be utilized for similar days in the earlier years.

1.2.2.2 Regression methods:

Regression is one of the most widely utilized statistical systems. For energy forecasting regression strategies are normally used to model the relationship of power consumption and different variables, for example, climate, day sort, and consumer class. Engle introduced a few regression models for the following day peak load forecast [7]. These models include deterministic impacts, for example, holidays, stochastic impacts, for example, normal loads, and external impacts, for example, climate.

1.2.2.3 Time series:

Time-series techniques depend on the presumption that the information has an inside structure, for example, autocorrelation, pattern, or regular variation. Time-series forecasting techniques distinguish and investigate such a structure. Time-series method has been utilized for a considerable length of time as a part of such fields as economic aspects, DSP, and also electric load determining. Specifically, 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) are the frequently utilized traditional time-series techniques. ARMA models are normally utilized for stationary processes while ARIMA is an expansion of ARMA to non-stationary forms. ARMA and ARIMA utilize the time and load as the main data input parameters. Since load by and large relies upon the climate and time, ARIMAX is the most common model for load estimating among the established time-series models. Fan and McDonald [9] and Cho [13] portray usage of ARIMAX models for load estimating. Yang [21] utilized Evolutionary Programming (EP) way to recognize the ARMAX model parameters for a day to a week ahead hourly load forecast. EP [10] is a technique for reproducing advancement and constitutes a stochastic improvement algorithm. Yang and Huang [27] gave a fuzzy autoregressive moving average with exogenous input variables (FARMAX) for hourly estimates for a day.

1.2.2.4 Neural networks:

The utilization of simulated neural systems (ANN or basically NN) has been a widely studied energy forecasting model since 1990 [8]. NN are basically non-linear circuits that have shown the ability to do non-linear curve fitting.

The yields of an ANN system are some linear or nonlinear functions of its inputs. The inputs might be the yields of other network components and also actual system inputs. In practice system components are organized in a moderately small number of connected layers of components between system inputs and outputs. Feedback paths are also utilized sometimes.

While applying an artificial neural network to load forecasting, one must choose one of various designs such as back-propagation, the number of layers and components, utilization of bi-directional or uni-directional connections, and the number format (e.g. binary or persistent) to be utilized by inputs and outputs.

The most well-known ANN design for load estimating is back propagation. Back-propagation neural systems utilize continuous function and supervised learning. That is, in supervised learning, actual numerical values assigned the system inputs are determined by comparing historical data, (for example, time and climate) to desired output, (for example, historical data of load) in data training. ANN with unsupervised learning doesn’t require data training.

Bakirtzis [16] built up an artificial neural network based short term load estimating model. In this they utilized a fully connected 3-layer feed-forward artificial neural network & back-propagation algorithm was utilized for the training session. Input data include the recorded hourly load information, temperature, humidity and the day of the week. The model can estimate load profiles from one day to seven days. Additionally Papalexopoulos [12] created and implemented a multi layered feed-forward neural network for short term load determining. In the model three sorts of variables are utilized as inputs to the ANN: inputs related to season, climate related inputs, and recorded loads. Khotanzad [23] portrayed a load determining system known as ANNSTLF. ANNSTLF depends on multiple Neural Network methodologies that catch different patterns in the data. In this they utilized a multi-layer perceptron ANN trained with error back-propagation algorithm. ANNSTLF considered the impact of temperature and relative humidity on load. It additionally contains forecasters that generated the hourly estimates of temperature and relative humidity required by the system. An improvement of this system was portrayed in [25]. In the new era, ANNSTLF incorporates two neural network forecasters, with one predicting the load and the other predicting the change in load. The final estimate is done by an adaptive mix of these. The impact of the humidity and wind velocity are considered through a linear change of temperature. Chen [31] built up a three layer completely connected feed-forward ANN and the back-propagation algorithm was utilized as the training technique. Their neural network however considers the power cost as one of the principle characteristic of the system load. Numerous distributed studies use ANN in conjunction with other estimating strategies, (for example, with regression trees [33], time-series model [17] or fuzzy inference [26]).

1.2.2.5 Expert systems:

Rule-based estimating utilizes rules, which are most of the times of heuristic nature, to do accurate load estimating. Expert systems, includes rules & procedures used by humans in the field and incorporate it into software that then automatically do load forecasting without the assistance of humans.

The use of Expert system began in mid-1960 for applications such as geological prospecting & design of computer. Expert system works best when an expert works with software developers and imparts his knowledge to the developed software. Furthermore, his knowledge must be codified into rules of the software (i.e. the programmer must be able to put the decisions of the expert into the system). Hundreds and thousands of rules may be coded into the expert system.

A knowledge-based expert system was proposed for the short term load forecasting by Ho [5]. Among other things weather parameters were also taken into account. Knowledge of the load is represented in parameter form. Knowledge about the factors affecting load are also represented in the form of rules. The rule base and the database vary from one place to other. This technique was tested for different sites and it has low error.

1.2.2.6 Fuzzy logic:

Fuzzy logic uses Boolean logic used in digital circuit designing. The input of the Boolean logic is binary. In fuzzy logic we have qualitative ranges such as “low”, “medium”, “high”, etc. Fuzzy logic allows us to logically deduce the output. Fuzzy logic, like curve fitting, is one of the numerous techniques available for mapping inputs to outputs. Fuzzy logic has many advantages. It does not require a mathematical model for its working. Also, it does not require noise-free input. With such general rules, fuzzy inference system can be a very robust one. After the fuzzification and rule base, a de-fuzzification method is also required. The de-fuzzification method provides us with precise outputs required. References [29], [30], [26] describe applications of fuzzy logic to electric load forecasting.

1.2.2.7 Support vector machines:

SVMs are a very powerful technique for solving classification and regression problems. This theory was given by Vapnik. Unlike ANN, in which complex functions were defined for input feature space, SVMs use kernel functions to perform mapping of data into higher dimensional feature space. Then SVMs create linear decision boundaries by the use of linear functions in new space. In Support Vector Machines, we have to use a suitable kernel function whereas in artificial neural network the problem is of choosing architecture [28].

Mohandes [38] used the method of SVMs for short-term electrical load estimating. He compared the performance of this method with the autoregressive method. The results show that Support Vector Machines gave much better results than autoregressive method. Chen [35] gave a SVM model for predicting daily demand of load for a month.

CHAPTER – 2

K-MEANS CLUSTERING

2.1 Clustering

Clustering is grouping of data in a way such that objects belonging to same group or cluster are similar in some way to each other than to those belonging to other groups or clusters. The main task of clustering is data mining and it is used in a number of fields such as machine learning, image recognition, pattern analysis, etc.

Clustering is not some specific algorithm. It is a general task which can be solved by various algorithms which differ significantly from each other in the way they define cluster and how to find them. The algorithm to be used and the parameters such as distance function, density threshold and the number of clusters depend on the available data set and its intended use. Clustering is not an automatic process, but an iterative task of knowledge or multi-objective optimization which involves hit and trial. It is very often required to pre-process the data available to get the desired results.

2.1.1 Definition of clustering

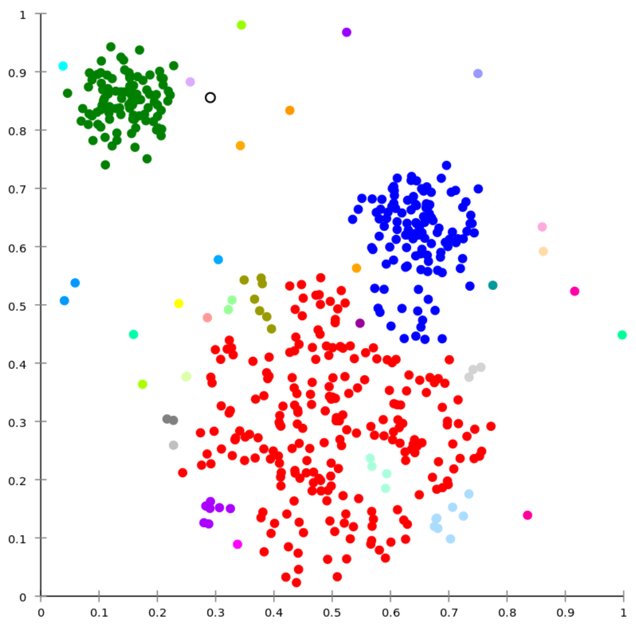
No particular algorithm can solve all the clustering problems. Therefore, there are so many clustering algorithms [41]. But all the algorithms have one thing in common: a group of data set. Different researchers have put forth different cluster models giving rise to different algorithms. For understanding these algorithms, we have to understand these cluster models.

Cluster models generally include:

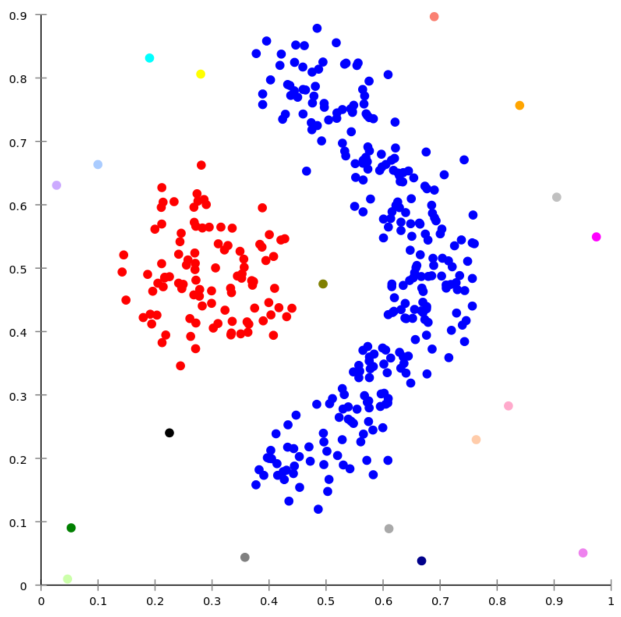
- Connectivity models: such as the hierarchical clustering models which are based on distance connectivity.

- Centroid models: such as the K-means clustering in which a single mean vector represents each cluster.
- Distribution models: in this the clusters are modeled by statistical distributions, such as the Expectation-maximization algorithm.
- Density models: such as DBSCAN & OPTICS defines clusters in the data space as connected dense regions.
- Subspace models: for example, Bi-clustering or Co-clustering or two-mode-clustering, in this clusters are modeled as cluster members and other relevant attributes.
- Group models: few algorithms just provide grouping information without any refined model.
- Graph-based models: any two nodes connected by an edge can be considered as a prototype.

The figure 2.1 shows some data clustering examples:



(a)



(b)

Figure 2.1 Data clustering example

2.2 K-means clustering

K-means clustering is a clustering method which originated from signal processing. It is basically a method of vector quantization. K-means clustering partitions n observations into k clusters such that each of the observation belongs to the cluster with the nearest mean.

The problem is computationally complex, but there are many efficient algorithms that can be employed and gives a quick convergence to a local optimum. Moreover, it uses cluster centers to model the given data.

If we have a set of n observations ($x_1, x_2, \ldots, x_n$) such that each observation is a d-dimensional vector, then the K-means clustering divides the observations into k clusters, where k ($\leq$ n) sets **S** = $\{S_1, S_2, \ldots, S_k\}$ so that the sum of square within cluster is minimized.

The objective is to find:

$$\arg min \sum_{i=1}^{k} \sum_{k \epsilon S_i} ||x - \mu_i||^2 \quad \ldots (2.1)$$

where μ_i is the mean of points in S_i.

2.2.1 History

James MacQueen was the first to use the term "k-means" in 1967 [4]. However, the idea for it was given by Hugo Steinhaus in 1957 [1]. The first algorithm was proposed in 1957 by Stuart Lloyd. IT was a technique for pulse modulation. But it wasn't until 1982 that it was published outside of Bell Labs [2]. E.W.Forgy published the same method in 1965 and hence, it is also referred as Lloyd-Forgy [3]. In 1975, Hartigan and Wong published its more efficient method.

2.2.2 Algorithm

Standard algorithm

The most common algorithm is the k-means clustering algorithm. It makes use of the iterative refinement technique. K-means clustering is also known as the Lloyd's algorithm. Let the initial set $m_1^{(1)},\ldots,m_k^{(1)}$, of k-means data is given. The algorithm has the following two steps [47]:

Assignment step:

Each of the observation is assigned to a particular cluster depending on whose mean gives the least WCSS (within-cluster sum of squares).We know that the sum of squares is the square of the Euclidean distance, which is basically the nearest mean [42].

$$S_i^{(t)} = \left\{x_p : \left|x_p - m_i^{(t)}\right|^2 \leq \left|x_p - m_j^{(t)}\right|^2 \forall j, 1 \leq j \leq k\right\}, \quad \ldots (2.2)$$

where m_i, m_j are the initial set of k-means data and S_i is the d-dimensional vector set and each x_p is defined to exactly one $S^{(t)}$

Update step:

New means are calculated. These calculated means would be used in the new clusters as the centroid of the observations.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \epsilon S_i^{(t)}} x_j \qquad \dots (2.3)$$

where m_i is the initial value of k-means data set and $S^{(t)}$ is the matrix set

The arithmetic mean is also a least square estimator. As a result, the minimization of the within-cluster sum of squares is also achieved.

The algorithm will be said to have converged when the means no longer change. Since both of the above steps optimize the within-cluster sum of squares objective, and there exists only finite such partitioning, the given algorithm will converge to local optimum. The possibility of global optimum is not guaranteed using this algorithm.

This algorithm assigns object to the nearest cluster on the basis of distance. The given algorithm minimizes the within-cluster sum of squares. Thus assigning the observations by “least sum of square”, this is exactly same as assigning the observations by smallest Euclidean distance. Use of a distance function different than the Euclidean distance may prevent convergence of the algorithm.

Initialization methods

Forgy and Random Partition are the commonly used methods of initialization. In Forgy method, k observations are chosen randomly from the data set and these are used as initial means. In Random Partition method a cluster is randomly assigned to each observation and then the update step is processed. And in this way the initial mean is to be computed. This mean becomes the centroid of the randomly assigned points of the cluster. In Forgy method the initial mean tend to spread out whereas in Random Partition method the means are placed at the center of the data set. For the k-means clustering, the preferable method is Forgy method.

2.2.3 An example of k-means clustering

Table 2.1: Data Set of the k-means clustering example:

Sepal length	**Sepal width**	**Petal length**	**Petal width**	**Species**
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*
4.4	2.9	1.4	0.2	*I. setosa*
4.9	3.1	1.5	0.1	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*

Sepal length	**Sepal width**	**Petal length**	**Petal width**	**Species**
5.5	2.3	4.0	1.3	*I. versicolor*
6.5	2.8	4.6	1.5	*I. versicolor*
5.7	2.8	4.5	1.3	*I. versicolor*
6.3	3.3	4.7	1.6	*I. versicolor*
4.9	2.4	3.3	1.0	*I. versicolor*
6.6	2.9	4.6	1.3	*I. versicolor*
5.2	2.7	3.9	1.4	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*
6.5	3.0	5.8	2.2	*I. virginica*
7.6	3.0	6.6	2.1	*I. virginica*
4.9	2.5	4.5	1.7	*I. virginica*

Sepal length	**Sepal width**	**Petal length**	**Petal width**	**Species**
7.3	2.9	6.3	1.8	*I. virginica*
6.7	2.5	5.8	1.8	*I. virginica*
7.2	3.6	6.1	2.5	*I. virginica*

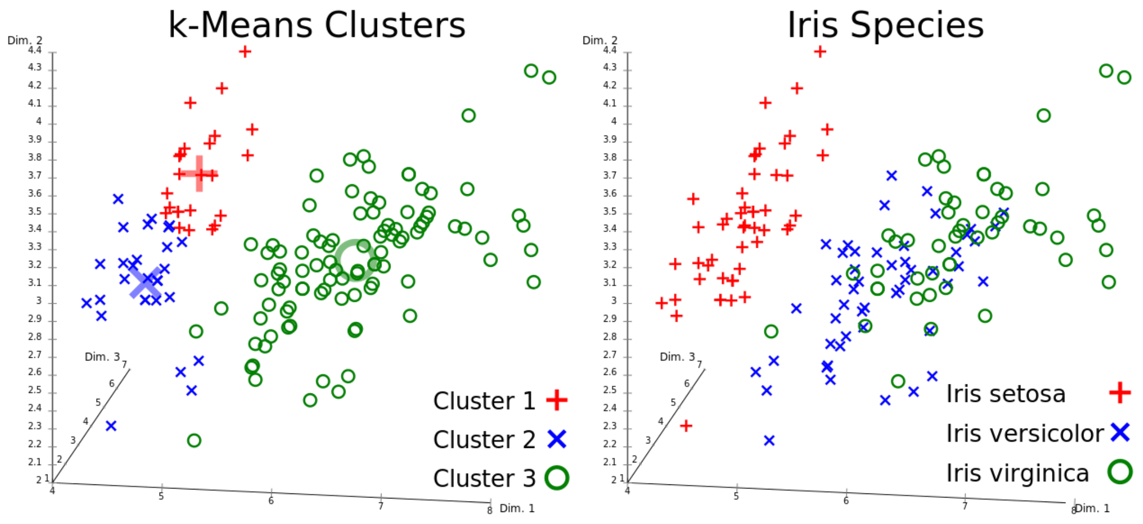


Figure 2.2: A k-means clustering example

The result of this algorithm may depend on initial cluster and hence there is no guarantee that this algorithm will converge to a global optimum. Since, it is a very fast algorithm; it is generally run multiple times with different initial values.

CHAPTER – 3

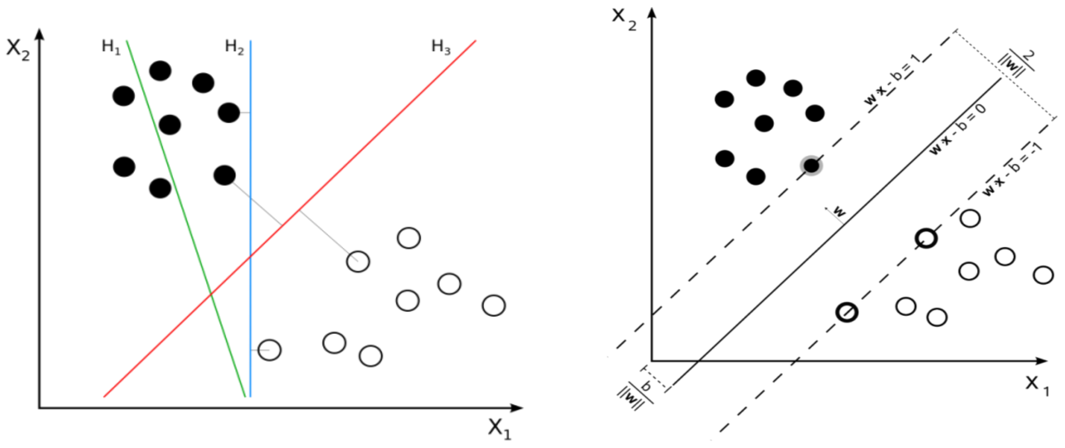
SUPPORT VECTOR MACHINES

3.1 Support Vector Machines

Support Vector Machines [15] or SVM is supervised machine learning method used for classification of data into two different categories. A SVM is trained with data, where the data is labeled for belonging to one or the other category. The SVM training algorithm builds a model that marks the new data into one of the two categories. The SVM model maps the data into space such that the two classes are separated by a clear gap which is as wide as possible. A new data is then mapped onto the same space and predicted to be belonging to one or the other class depending on which side of the margin it lies.

SVM can perform the linear as well as the non-linear classification. In non-linear classification, it maps the data into a higher dimensional feature space using what is known as a kernel trick. SVM constructs a hyperplane to divide the data into the two classes. A good hyperplane can be constructed if it at maximum distance from the training data of both the classes.

Figure 3.1(a) shows various hyperplanes separating the two classes. The hyperplanes H1 and H2 are not good hyperplanes whereas hyperplane H3 is a good hyperplane. In figure 3.2(b), there are three straight lines. The solid straight line in the middle shows the decision boundary separating the two classes shown by the equation $wx\text{-}b=0$, where w is the slope and b is a constant. The dotted lines pass through the support vectors of the two classes such that the distance of the support vectors is maximum from the decision boundary.



(a) (b)

Figure 3.1: SVM hyperplanes

3.1.1 History of SVM

The SVM algorithm was invented by Vladimir Vapnik and Alexey Chervonenkis in 1963. In the year 1992, Vladimir Vapnik, Isabelle Guyon and Bernhard Boser gave forth the non-linear classifier using the kernel trick. Because of its success in handwritten digit recognition, SVM became very popular. There was only 1.1% error rate in SVM for handwritten digit recognition. This was same as the error rate for a carefully constructed ANN.

3.1.2 Decision Boundary

The SVM classifier needs a decision boundary for separating the two classes. There are two lines, one at each of the data points of the two classes which are closest to the other class, so that the gap is maximum between the two classes. Then the decision boundary is at the center of both of these lines so that it is equidistant from both the classes. The data points on either side of the decision boundary which are closest to the decision boundary are known as support vectors.

3.1.3 Finding good decision boundary

Let us consider a linearly separable, two-class problem as shown in figure 3.2

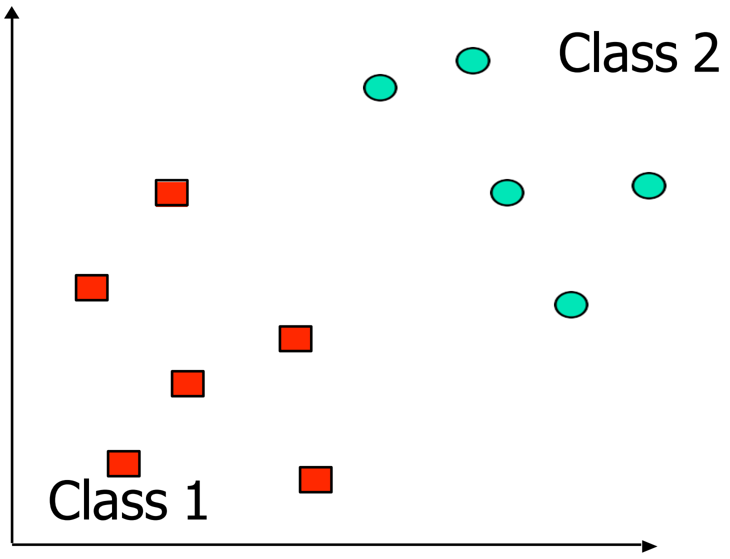


Figure 3.2: Representation of two-class system

Now, what is a good decision boundary that will separate the two classes? Some decision boundaries separating the two classes are shown in figure 3.3.

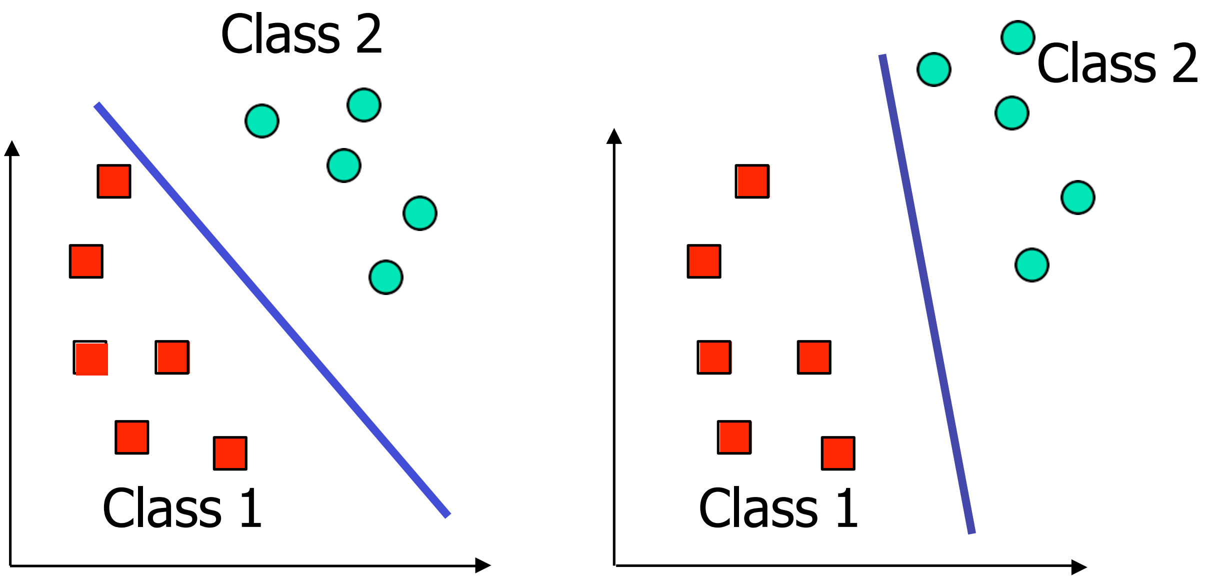


Figure 3.3: Bad decision boundaries

The above are the examples of bad decision boundaries. For a decision boundary to be a good decision boundary it must be a large margin decision boundary.

3.1.4 Large margin decision boundary

The decision boundary separating the two classes of data should be such that it is as far away as possible from both the classes. If we look at the below figure the problem is of maximizing the margin m. We know that the distance between origin and any line $w^T x = k$ is:

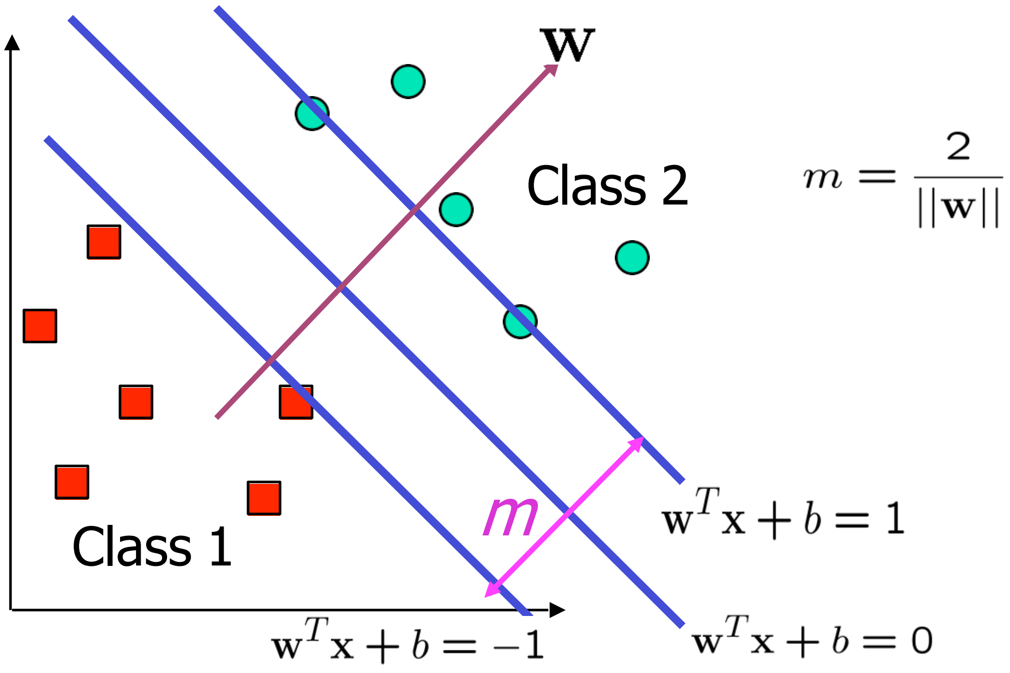
$$\frac{k}{||w||} \qquad \text{... (3.1)}$$

where k is a constant and **w** is the vector perpendicular to the separating hyperplane

Let the data set be $\{x_1, x_2, \ldots, x_n\}$ and the class label of our data set be y_i {-1, 1}.

The task of decision boundary is that it should classify all the points correctly into the two classes.

$$y_i(w^T x_i + b) \geq 1 \quad \forall i \qquad \text{... (3.2)}$$



w
Class 2
$m = \frac{2}{||\mathbf{w}||}$
m
$\mathbf{w}^T\mathbf{x} + b = 1$
Class 1
$\mathbf{w}^T\mathbf{x} + b = -1$
$\mathbf{w}^T\mathbf{x} + b = 0$

Figure 3.4: A two-class system with good decision boundary

Now we can find the decision boundary by solving the following:

Minimize:

$$\frac{1}{2}||w||^2$$

Subject to:

$$y_i(w^T x_i + b) \geq 1 \qquad \forall i$$

The above problem is that of constraint optimization.

3.1.5 Solving Constraint Optimization Problem

Suppose we have a constraint minimization problem in which we have to minimize a function $f(x)$ subject to the condition $g(x) = 0$

The condition for x_0 to be a solution of the given problem it should satisfy the below condition:

$$\begin{cases} \frac{\partial}{\partial x}\left(f(x) + \alpha g(x)\right)\Big|_{x=x_0} = 0 \\ g(x) = 0 \end{cases} \quad \dots (3.3)$$

where α is the Lagrange multiplier and *f(x)* and *g(x)* are functions of input set *x*.

For multiple constraint problem $g_i(x) = 0$. A Lagrange multiplier is required for each constraint.

$$\begin{cases} \frac{\partial}{\partial x}\left(f(x) + \sum_{i=1}^{n} \alpha_i g_i(x)\right)\Bigg|_{x=x_0} = 0 \\ g_i(x) = 0 \qquad for\ i = 1,2,\dots,m \end{cases} \quad \dots (3.4)$$

where α_i are the Lagrange multiplier and *f(x)* and $g_i(x)$ are functions of *x*.

For the inequality constraint problem $g_i(x) = 0$, the Lagrange multiplier should be positive. The solution of the inequality constraint optimization problem:

$$\min f(x) \text{ subject to } g_i(x) \leq 0 \text{ for } i = 1,2,\ldots,m$$

There must exist a positive Lagrange multiplier $\alpha_i \geq 0$ such that:

$$\begin{cases} \frac{\partial}{\partial x}\left(f(x) + \sum_i \alpha_i g_i(x)\right)\Big|_{x=jx_0} = 0 \\ g_i(x) \leq 0 \qquad for\ i = 1,2,\ldots,m \end{cases} \qquad \ldots (3.5)$$

The function $f(x) + \sum_I \alpha_i g_i(x)$ is known as the Lagrangian. For solving the problem we need to set its gradient to 0.

3.1.6 Back to finding decision boundary

Minimize:

$$\frac{1}{2}||w||^2$$

Subject to:

$$1 - y_i\ (w^T x_i + b) \leq 0 \qquad for\ i = 1,2,\ldots,n$$

x_i is the input set and y_i is the class label for the data set and b is a constant

w is the vector perpendicular to the hyperplane separating the two classes

The Lagrangian is given by:

$$\mathcal{L} = \frac{1}{2}w^T w + \sum_{i=1}^{n} \alpha_i\left(1 - y_i(w^T x_i + b)\right) \qquad \ldots (3.6)$$

Since

$$||\boldsymbol{w}||^2 = \boldsymbol{w}^T \boldsymbol{w}$$

w is the vector perpendicular to the hyperplane

Setting the Lagrangian gradient with respect to **w** and b equal to zero, we get:

$$w + \sum_{i=1}^{n} \alpha_i(-y_i)x_i = 0$$

$$\Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \text{... (3.7)}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \text{... (3.8)}$$

Now if we substitute $w = \sum_i \alpha_i y_i x_i$ to the Lagrangian, we get:

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i x_i^T \sum_{j=1}^{n} \alpha_j y_j x_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j y_j x_j^T x_i \right) \right)$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j x_j^T x_i - b \sum_{i=1}^{n} \alpha_i y_i$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i$$

Also we have $\sum_i \alpha_i y_i = 0$

We can see that it is a function of only α_i.

Now we have an objective function which is in terms of α_i only. This is also known as the dual problem as if we know **w,** we can find all α_i and if we know all α_i, we can find **w**. The objective function needs to be maximized. The dual problem, therefore, becomes maximizing:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \qquad \text{... (3.9)}$$

α_i, α_j are Lagrangian multiplier

y_i, y_j are class labels and x_i, x_j are input data set

Subject to:

$$\alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

w can be found out by:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

w has a sparse representation as many of the α_i's are zero. This can be viewed as data compression. The data points with non-zero α_i are known as the Support Vectors. Only Support Vectors are required for determining the decision boundary.

Let the s Support Vectors have the indices given by t_j (j = 1,2,…,s). Then we can write:

$$w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}$$

For a new data **z**, compute:

$$w^T z + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \left(x_{t_j}^T z \right) + b \qquad \text{... (3.10)}$$

If the sum is positive, we classify it as class 1 otherwise we classify it as class 2.

Table 3.1: An example with support vector weightage for geometrical interpretation

support vector	weightage
α_1	0.8
α_2	0
α_3	0
α_4	0
α_5	0
α_6	1.4
α_7	0
α_8	0.6
α_9	0
α_{10}	0

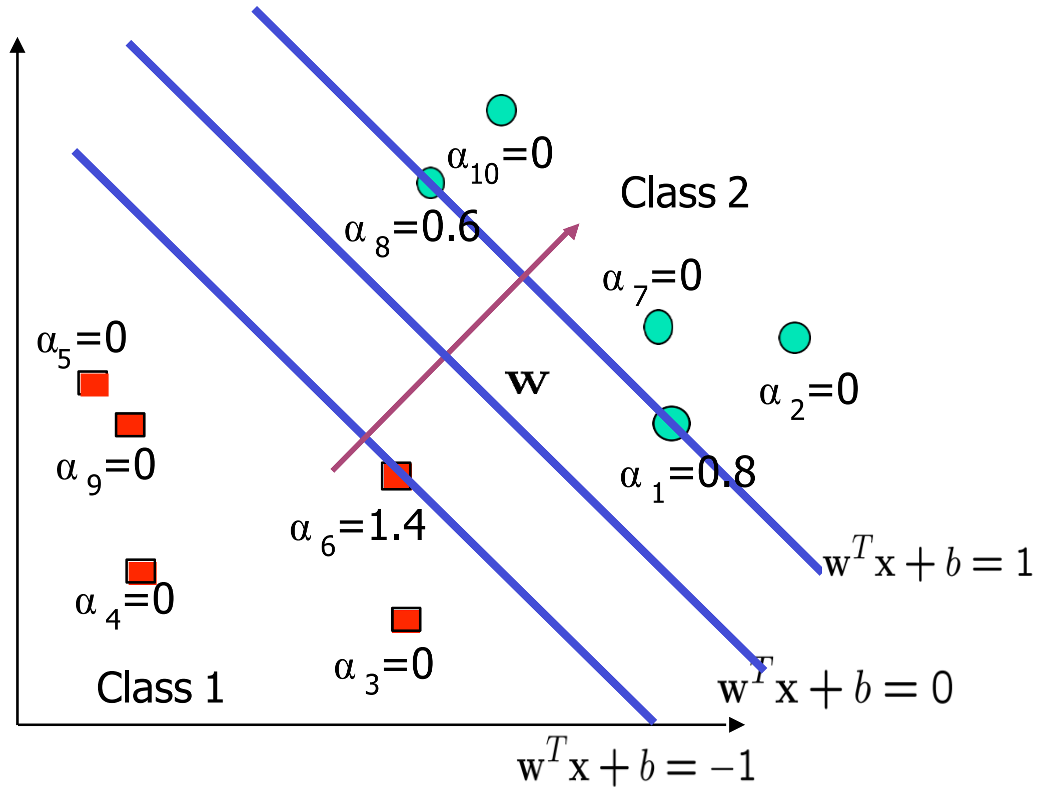


Figure 3.5: A geometrical interpretation

3.1.7 Non-linearly separable classes' problem

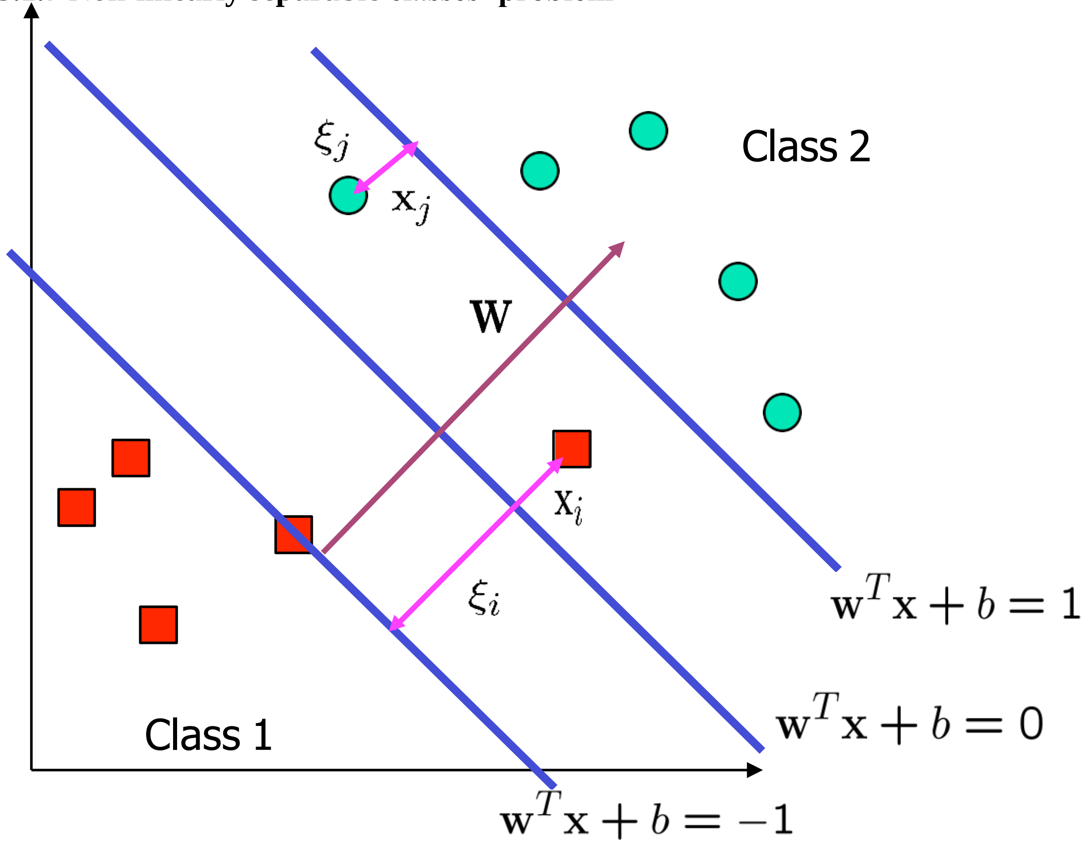

ξ_j
$\mathbf{x}_j$
Class 2
$\mathbf{W}$
$\mathbf{x}_i$
ξ_i
$\mathbf{w}^T\mathbf{x}+b=1$
$\mathbf{w}^T\mathbf{x}+b=0$
Class 1
$\mathbf{w}^T\mathbf{x}+b=-1$

Figure 3.6: Non-linearly separable problem

Let ξ_i be the error in classification based on the output of the function $\mathbf{w}^T\mathbf{x} + b$.

$$w^T x_i + b \geq \xi_i \qquad y_i = 1 \qquad \text{... (3.11)}$$

$$w^T x_i + b \leq -1 + \xi_i \qquad y_i = -1 \qquad \text{... (3.12)}$$

$$\xi i \geq 0 \qquad \forall\, i$$

ξ_i is the error, x_i is the input set, b is a constant

If there is no error for a data point, then ξ for that particular data point is zero. Now we have to minimize

$$\frac{1}{2}\,||w||^2 + C\sum_{i=1}^{n}\xi_i \qquad \text{... (3.13)}$$

where C is the trade-off parameter between margin & error.

The optimization problem now is maximizing:

$$\frac{1}{2}\,||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{Subject to} \qquad y_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi i \geq 0 \qquad \text{... (3.14)}$$

Now the dual of this new problem is maximizing:

$$W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \qquad \text{... (3.15)}$$

Subject to:

$$C \geq \alpha_i \geq 0, \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \text{... (3.16)}$$

w can be calculated as:

$$w = \sum_{j=1}^{s}\alpha_{t_j} y_{t_j} x_{t_j} \qquad \text{... (3.17)}$$

This is similar to the problem of optimization in linearly separable case. Only difference is that now there is an upper bound on α_i given by *C*.

3.1.8 Non Linear Decision Boundary

Until now we have only considered linear decision boundary. We can generalize it to include non-linear decision boundary. This is done by transforming $\mathbf{x}_i$ into a higher dimensional feature space.

Here input space is where $\mathbf{x}_i$ are located and feature space is where $\phi(\mathbf{x}_i)$ after transformation are located. With the help of transformation, the linear operation of the feature space becomes non-linear operation of the input space.

Computation can become very complex in the feature space because of it being a high dimensional space. A typical feature space is of infinite dimensions. Because of this we use what is known as a kernel trick. Figure 3.7 shows transformation into higher dimensional space:

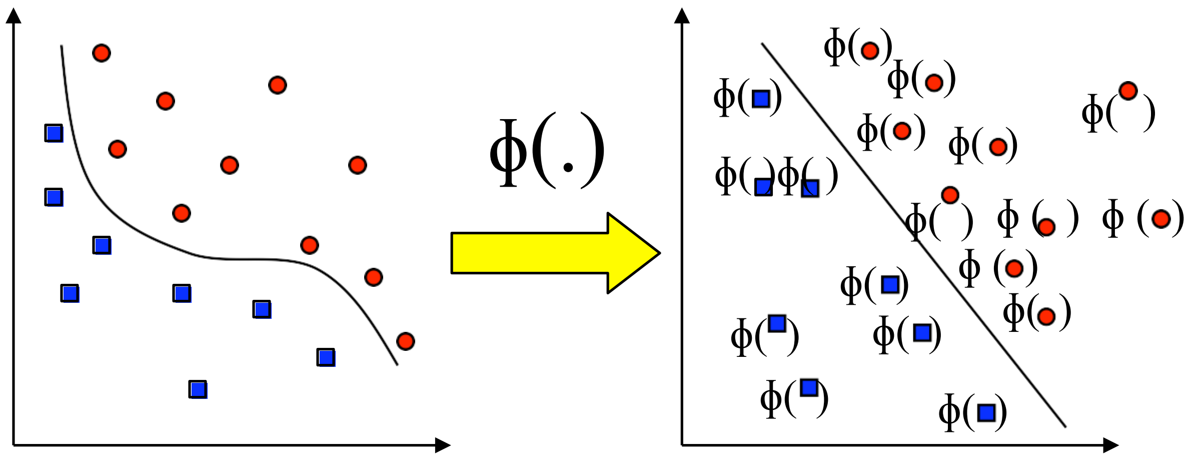


Figure 3.7: Transformation into higher dimensional feature space

The data points in the SVM optimization problem appear only as inner product. Therefore, as long as the inner product of the data points can be computed in the feature space, we need not explicitly map the data points to the higher dimension.

3.1.9 kernel trick

Let ϕ(.) is:

$$\varphi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$

Therefore the inner product in feature space is:

$$\langle\varphi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right), \varphi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right)\rangle = (1 + x_1y_1 + x_2y_2)^2$$

So the kernel function becomes:

$$K(x, y) = (1 + x_1y_1 + x_2y_2)^2$$

This use of the kernel function is known as the kernel trick. Here, there is no need of carrying out the mapping explicitly. In the practical use of Support Vector Machine (SVM), we just need to specify the kernel function. The transformation need not be explicitly stated.

3.1.10 Modification due to kernel trick

In the optimization problem equations, we replace all the data point inner product with the kernel function. As a result, for training the problem is now maximizing:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i\alpha_j y_i y_j K(x_i, x_j) \qquad \text{... (3.18)}$$

Subject to:

$$C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

For testing**,** a new data point is classified as:

1. Class 1, if $f \geq 0$
2. Class 2, if $f < 0$

And the equations without kernel trick are:

$$w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}$$

$$f = w^T z + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}^T z + b$$

And with kernel trick are:

$$w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \varphi(x_{t_j})$$

$$f = \langle w, \varphi(z) \rangle + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(x_{t_j}, z) + b \qquad \ldots (3.19)$$

where K represents the kernel function and other symbols have the meaning as defined earlier

3.2 Hybrid k-means clustering – SVM

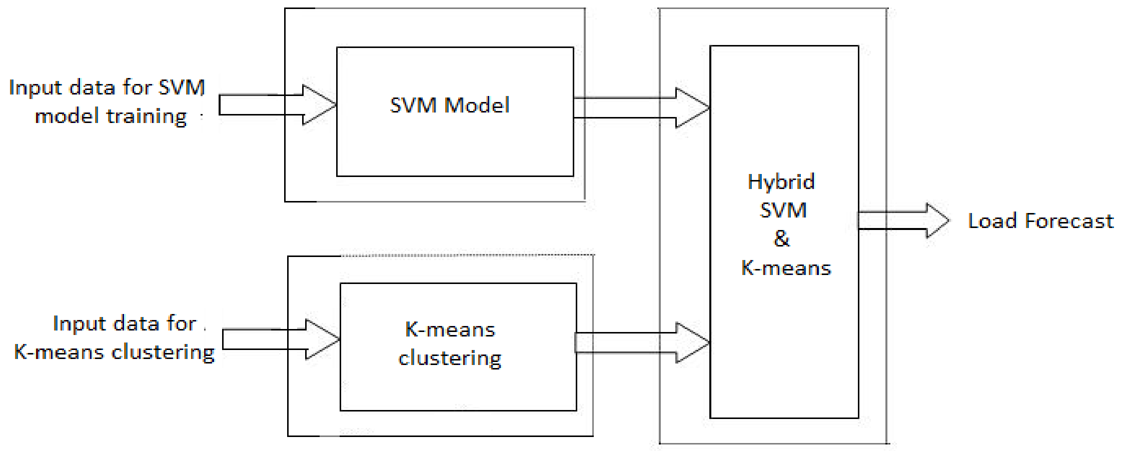
K-means clustering is used to distribute the given data points into clusters such that the data points belonging to a cluster have high degree of similarity. And the clusters so formed are as dissimilar as possible to each other. Each of the data points are distributed on the basis of cluster with nearest mean. That is, a data point is said to be belonging to that cluster whose mean is closest to the given data point.

SVM is used to classify a data point into two classes. In SVM two classes are separated by a hyperplane. The hyperplane should be such that it is at maximum distance from the data points of the two classes. This is also known as maximum margin criteria as the distance between the two hyperplanes passing though the data points of a particular class and separating it from the other class has to be maximum. The hyperplane separating the two classes or the decision boundary is then the one at the middle of this margin.

If a linear decision boundary is unable to separate the two classes, then we go for non-linear decision boundary. Kernel trick is used to transform the data points to higher dimensional feature space where the decision boundary is linear even if it was non-linear in the input space.

SVM is used to classify data into two classes; however, if data is to be classified into more than two classes then we use multi-class SVM.

In this methodology, we develop a novel technique for energy load forecasting by combining K-means clustering with SVM. K-means clustering is used to group the data into defined clusters and then SVM trains the model so developed using the data to give the load forecaster. The architecture of the proposed technology is shown in Figure 3.8:



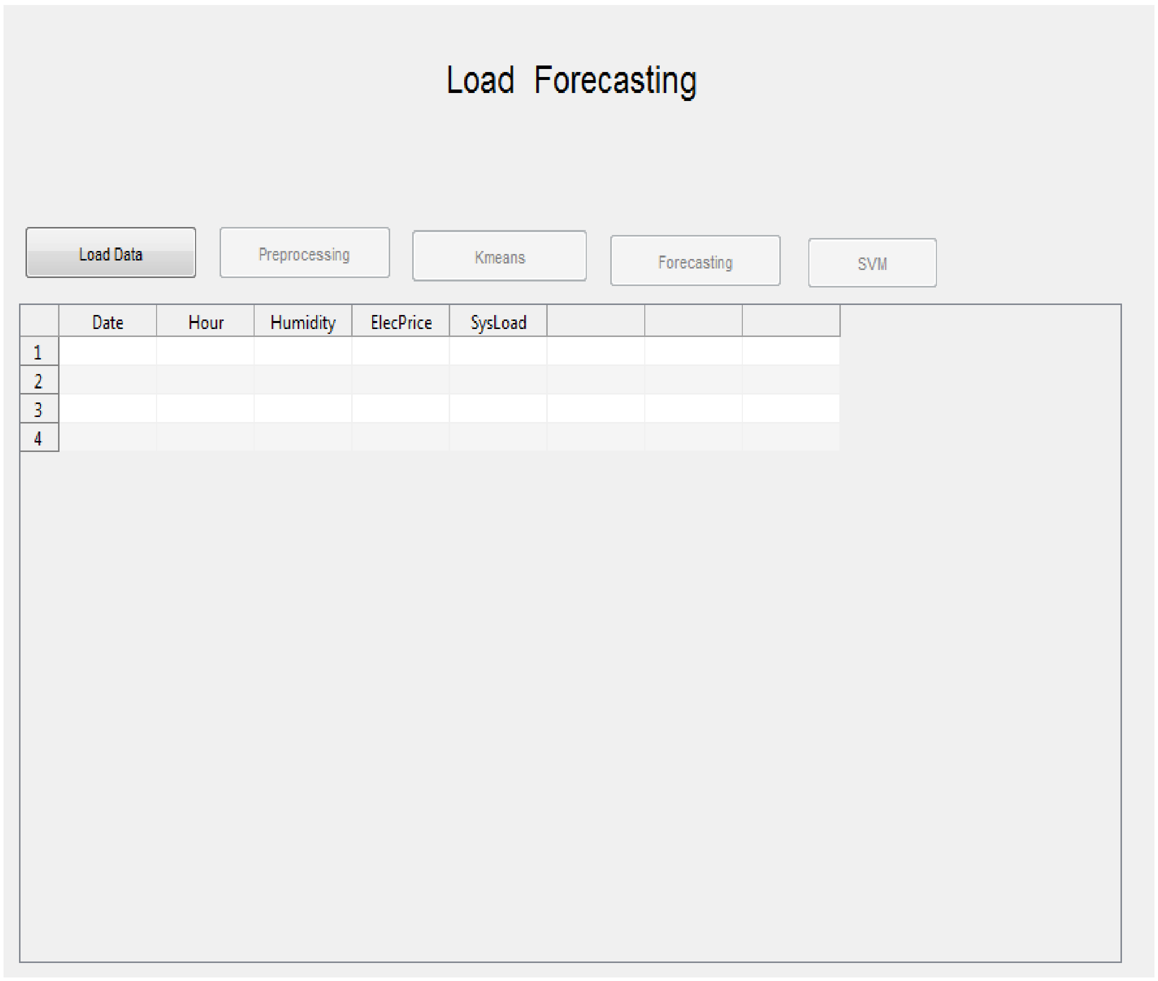
Input data for SVM model training
SVM Model
Hybrid SVM & K-means
Load Forecast
Input data for K-means clustering
K-means clustering

Figure 3.8: Architecture of the proposed methodology

CHAPTER – 4

RESULTS AND CONCLUSION

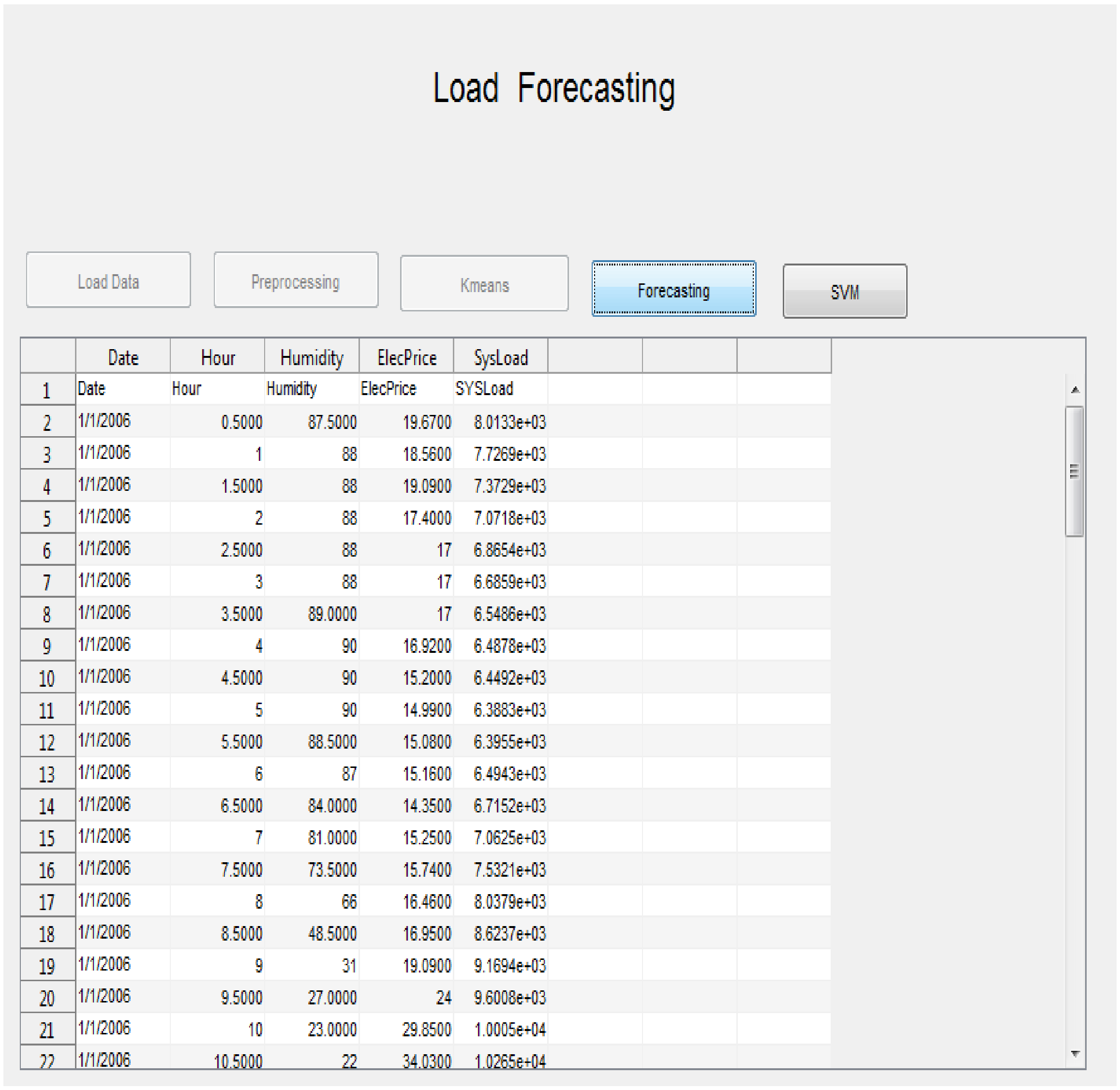
The technique of energy load forecasting developed by combining k-means clustering and SVM is run using the MATLAB R2014a software. The main Graphical User Interface or GUI developed for the technique is shown in figure 4.1:



	Date	Hour	Humidity	ElecPrice	SysLoad			
1								
2								
3								
4								

Figure 4.1: Graphical User Interface

In the Graphical User Interface, different pushbuttons are provided for carrying out different functions such as k-means clustering and SVM. First the data is loaded into the GUI. The data is taken over the period of 5 years from 2006 to 2010. The figure 4.2 shows the GUI with the loaded data.



	Date	Hour	Humidity	ElecPrice	SysLoad
1	Date	Hour	Humidity	ElecPrice	SYSLoad
2	1/1/2006	0.5000	87.5000	19.6700	8.0133e+03
3	1/1/2006	1	88	18.5600	7.7269e+03
4	1/1/2006	1.5000	88	19.0900	7.3729e+03
5	1/1/2006	2	88	17.4000	7.0718e+03
6	1/1/2006	2.5000	88	17	6.8654e+03
7	1/1/2006	3	88	17	6.6859e+03
8	1/1/2006	3.5000	89.0000	17	6.5486e+03
9	1/1/2006	4	90	16.9200	6.4878e+03
10	1/1/2006	4.5000	90	15.2000	6.4492e+03
11	1/1/2006	5	90	14.9900	6.3883e+03
12	1/1/2006	5.5000	88.5000	15.0800	6.3955e+03
13	1/1/2006	6	87	15.1600	6.4943e+03
14	1/1/2006	6.5000	84.0000	14.3500	6.7152e+03
15	1/1/2006	7	81.0000	15.2500	7.0625e+03
16	1/1/2006	7.5000	73.5000	15.7400	7.5321e+03
17	1/1/2006	8	66	16.4600	8.0379e+03
18	1/1/2006	8.5000	48.5000	16.9500	8.6237e+03
19	1/1/2006	9	31	19.0900	9.1694e+03
20	1/1/2006	9.5000	27.0000	24	9.6008e+03
21	1/1/2006	10	23.0000	29.8500	1.0005e+04
22	1/1/2006	10.5000	22	34.0300	1.0265e+04

Figure 4.2: GUI with loaded data

Table 4.1: A sample of the used data set [55]

S.No.	Date	Hour	Humidity	Price	Load
1	1/1/2006	0.5	87.5	19.67	8013.278
2	1/1/2006	1	88	18.56	7726.892
3	1/1/2006	1.5	88	19.09	7372.858
4	1/1/2006	2	88	17.4	7071.833
5	1/1/2006	2.5	88	17	6865.44
6	1/1/2006	3	88	17	6685.927
7	1/1/2006	3.5	89	17	6548.628
8	1/1/2006	4	90	16.92	6487.837
9	1/1/2006	4.5	90	15.2	6449.178
10	1/1/2006	5	90	14.99	6388.278
11	1/1/2006	5.5	88.5	15.08	6395.48
12	1/1/2006	6	87	15.16	6494.333
13	1/1/2006	6.5	84	14.35	6715.202
14	1/1/2006	7	81	15.25	7062.48
15	1/1/2006	7.5	73.5	15.74	7532.117
16	1/1/2006	8	66	16.46	8037.868
17	1/1/2006	8.5	48.5	16.95	8623.735
18	1/1/2006	9	31	19.09	9169.365
19	1/1/2006	9.5	27	24	9600.79
20	1/1/2006	10	23	29.85	10005.31
21	1/1/2006	10.5	22	34.03	10264.63
22	1/1/2006	11	21	43.63	10463.6
23	1/1/2006	11.5	20	45.6	10647.81
24	1/1/2006	12	19	47.21	10682.94
25	1/1/2006	12.5	18	68.11	10787.55
26	1/1/2006	13	17	99.57	10842.33
27	1/1/2006	13.5	16.5	153.18	10907.03
28	1/1/2006	14	16	100.75	10859.59
29	1/1/2006	14.5	16	103.6	10903.85
30	1/1/2006	15	16	103.65	10900.74
31	1/1/2006	15.5	15.5	104.24	10952.31
32	1/1/2006	16	15	104.36	10948.03
33	1/1/2006	16.5	15.5	104.25	10904.45
34	1/1/2006	17	16	103.93	10997.9
35	1/1/2006	17.5	16.5	73.93	11026.05
36	1/1/2006	18	17	60.38	11009.99
37	1/1/2006	18.5	20	48.13	10923.49
38	1/1/2006	19	23	45.25	10894.9
39	1/1/2006	19.5	25.5	49.16	10952.22
40	1/1/2006	20	28	49.01	11112.76
41	1/1/2006	20.5	48.5	40.53	10821.02
42	1/1/2006	21	69	26.9	10247.09

43	1/1/2006	21.5	72	20.94	9842.207
44	1/1/2006	22	75	18.48	9345.815
45	1/1/2006	22.5	75.5	17.63	8924.16
46	1/1/2006	23	76	16.87	8416.873
47	1/1/2006	23.5	76	16.84	8044.818
48	1/2/2006	0	76	16.37	7737.028

After the data is loaded into the Graphical User Interface, it undergoes preprocessing so that the data can be given as input to the k-means clustering step. After the data undergoes clustering, load is finally predicted using the SVM model.

The Actual and the Forecasted load are shown in figure 4.3. The actual load is shown in sky blue and the forecasted load is shown in navy blue.

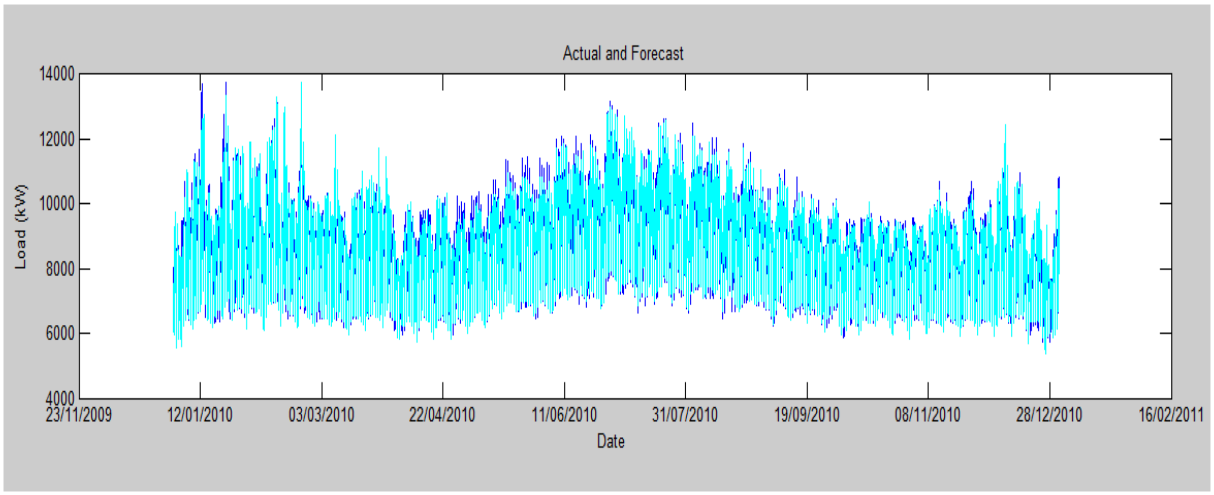


Figure 4.3: Actual and Forecasted load using hybrid K-means clustering – SVM

Below is the error between the forecasted load and the actual load. The error between the forecasted and actual load is quite low using this technique. However, we see a sharp increase in the error on the days which were holidays. This sharp increase in error is due to the unpredictability of load on holidays.

More data is required for the SVM trainer to accurately predict the load on holidays. With more information on load data on holidays, we can train the SVM model to predict load with less error.

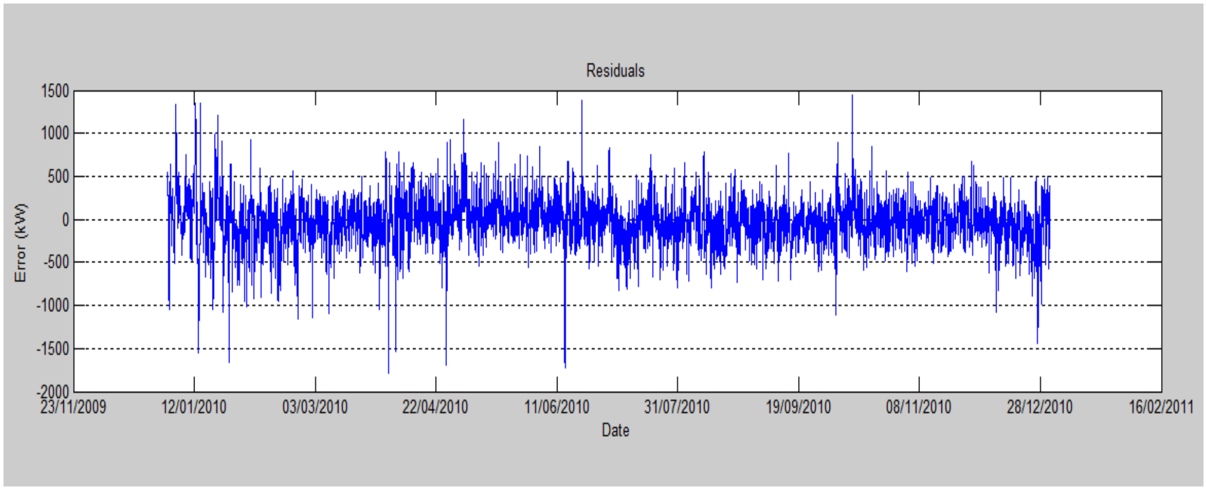


Figure 4.4: Error using hybrid k-means clustering – SVM

The month-wise load forecast using the hybrid technique of k-means clustering – SVM is also obtained.

The load forecast for the month of January 2015 is shown in the figure 4.5:

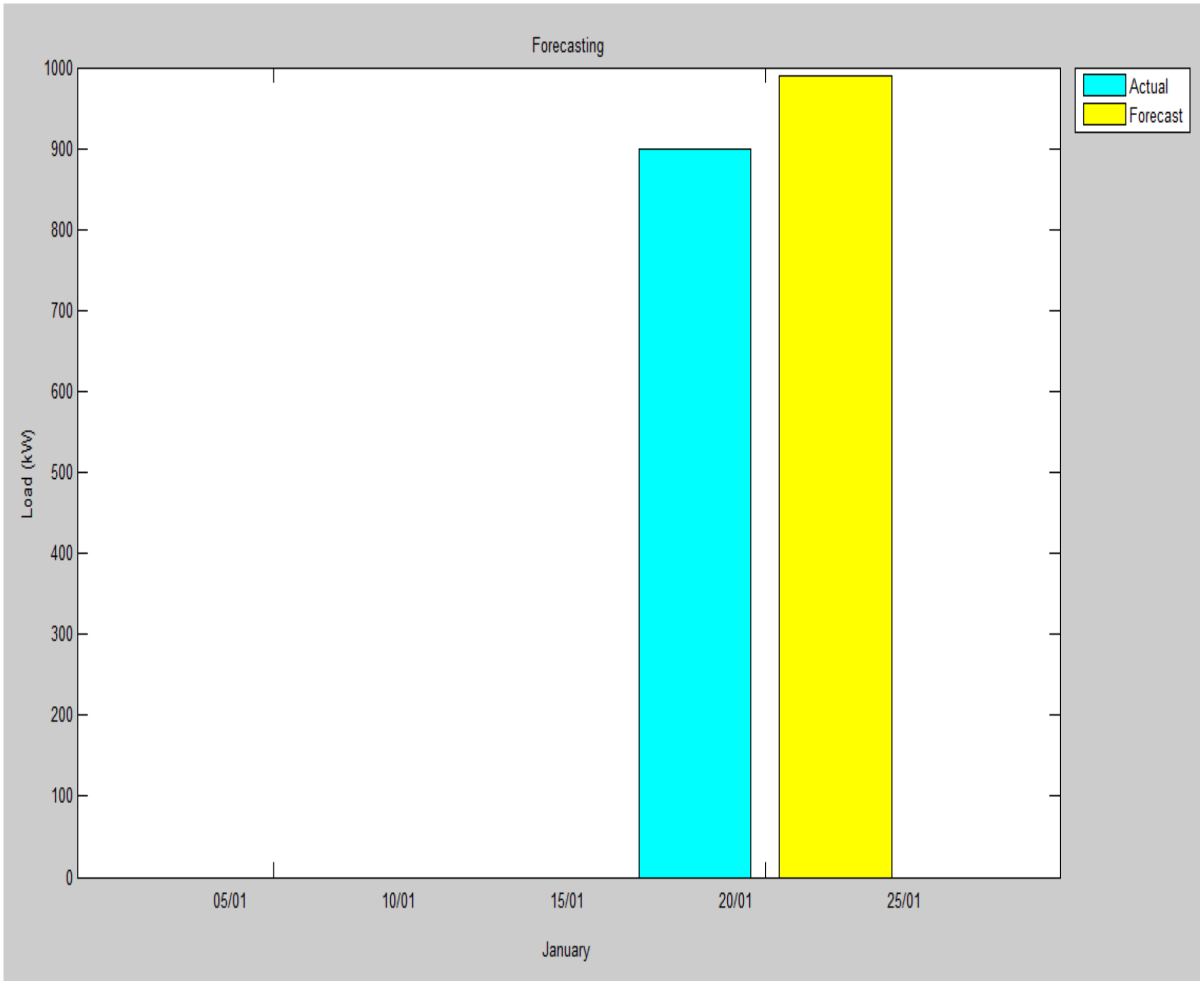


Fig 4.5: Load forecast for the month of January 2015

The load forecast for the month of February 2015 is shown in figure 4.6:

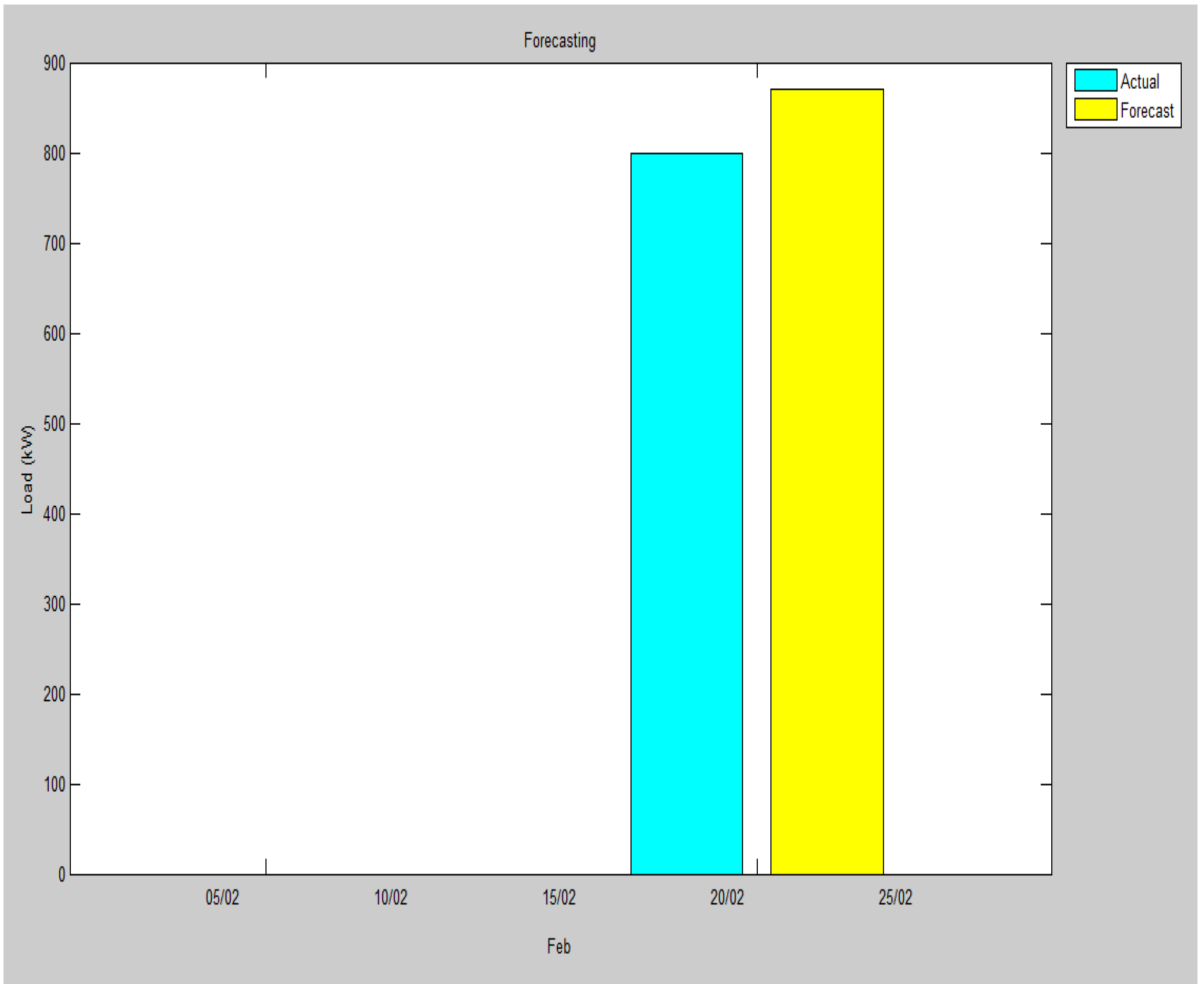


Fig 4.6: Load forecast for the month of February 2015

The load forecast for the month of March 2015 is shown in figure 4.7:

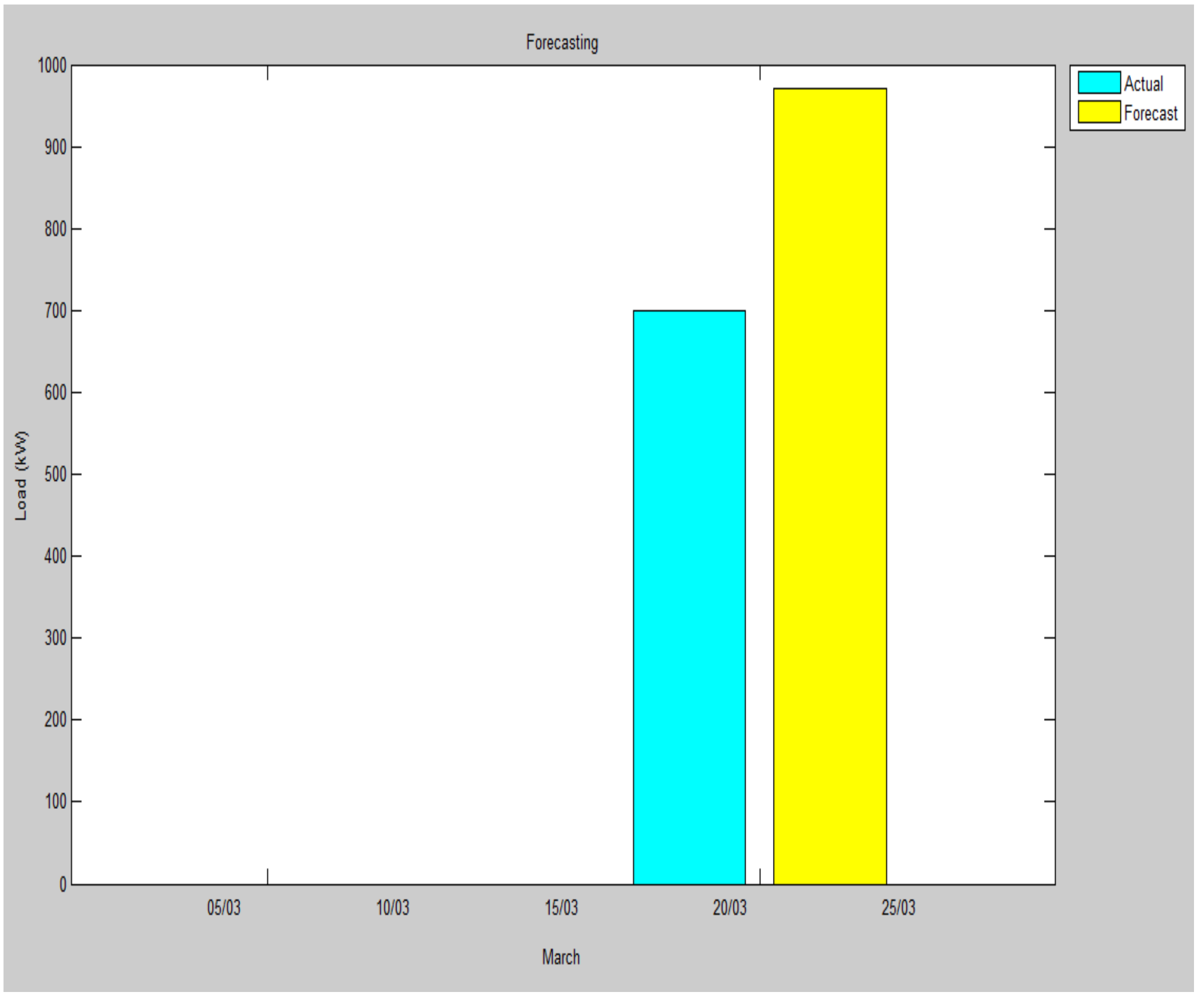


Fig 4.7: Load forecast for the month of March 2015

The load forecast for the month of April 2015 is shown in figure 4.8:

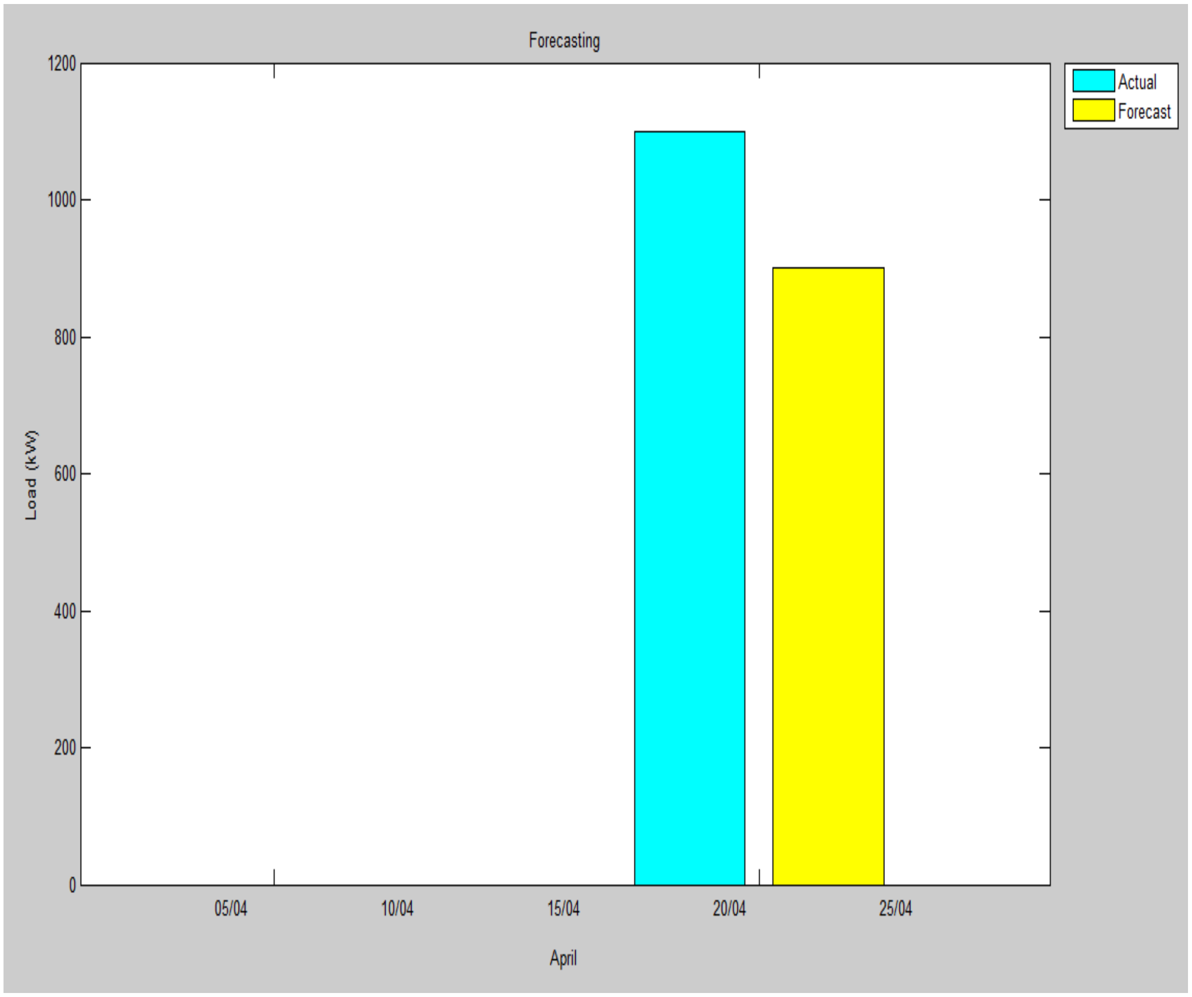


Fig 4.8: Load forecast for the month of April 2015

The load forecast for the month of May 2015 is shown in figure 4.9:

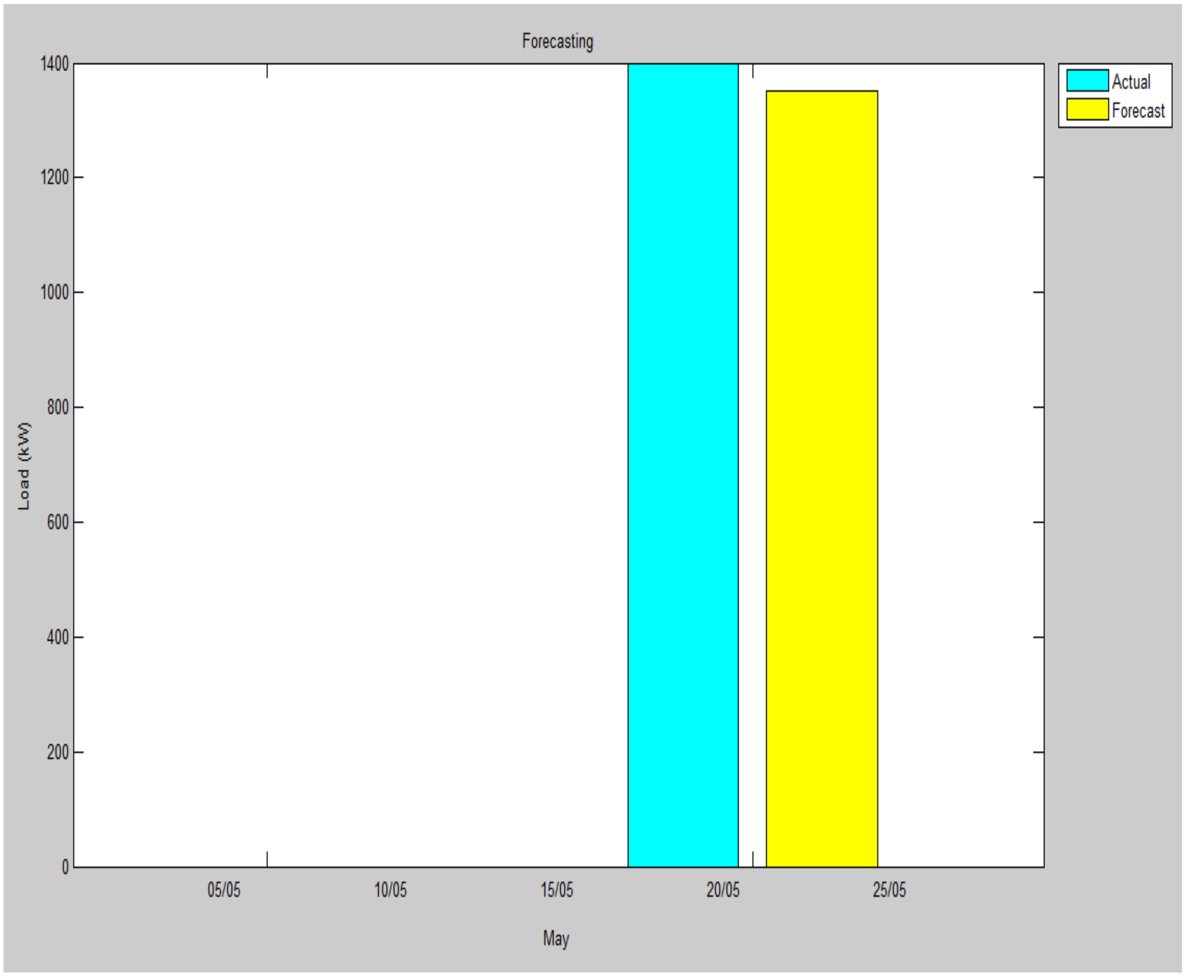


Fig 4.9: Load forecast for the month of May 2015

The load forecast for the month of June 2015 is shown in figure 4.10:

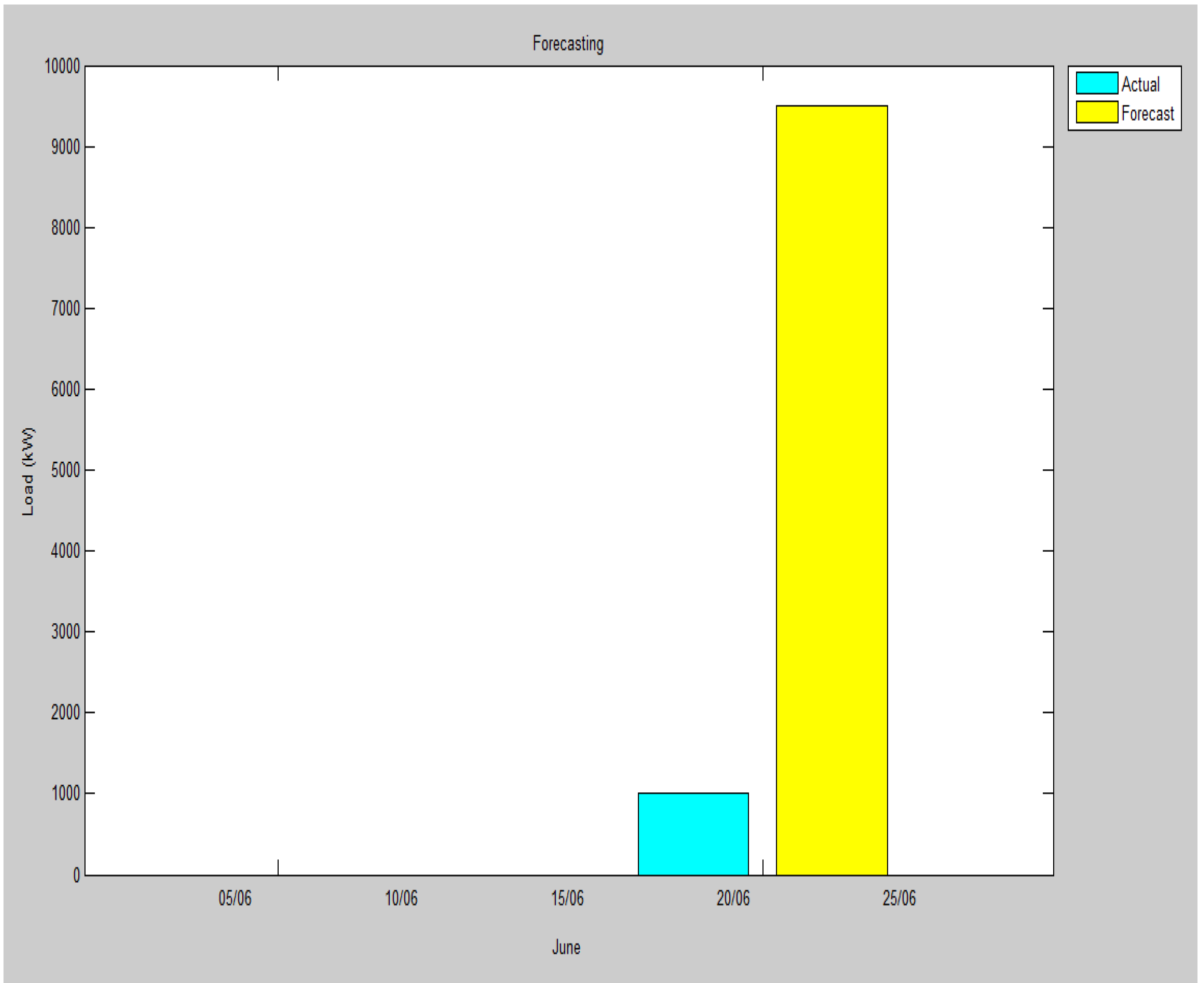


Fig 4.10: Load forecast for the month of June 2015

The load forecast for the month of July 2015 is shown in figure 4.11:

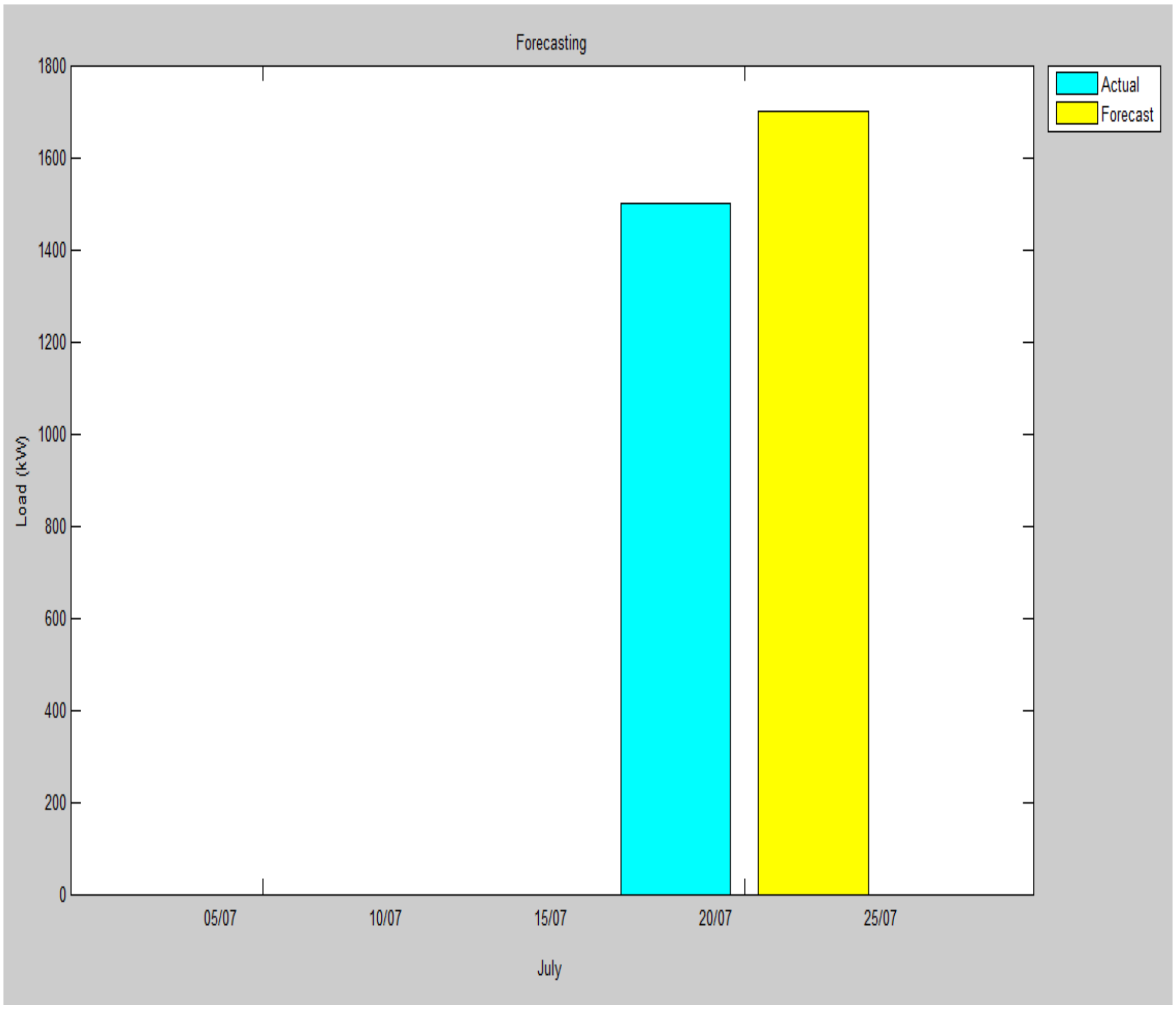


Fig 4.11: Load forecast for the month of July 2015

The load forecast for the month of August 2015 is shown in figure 4.12:

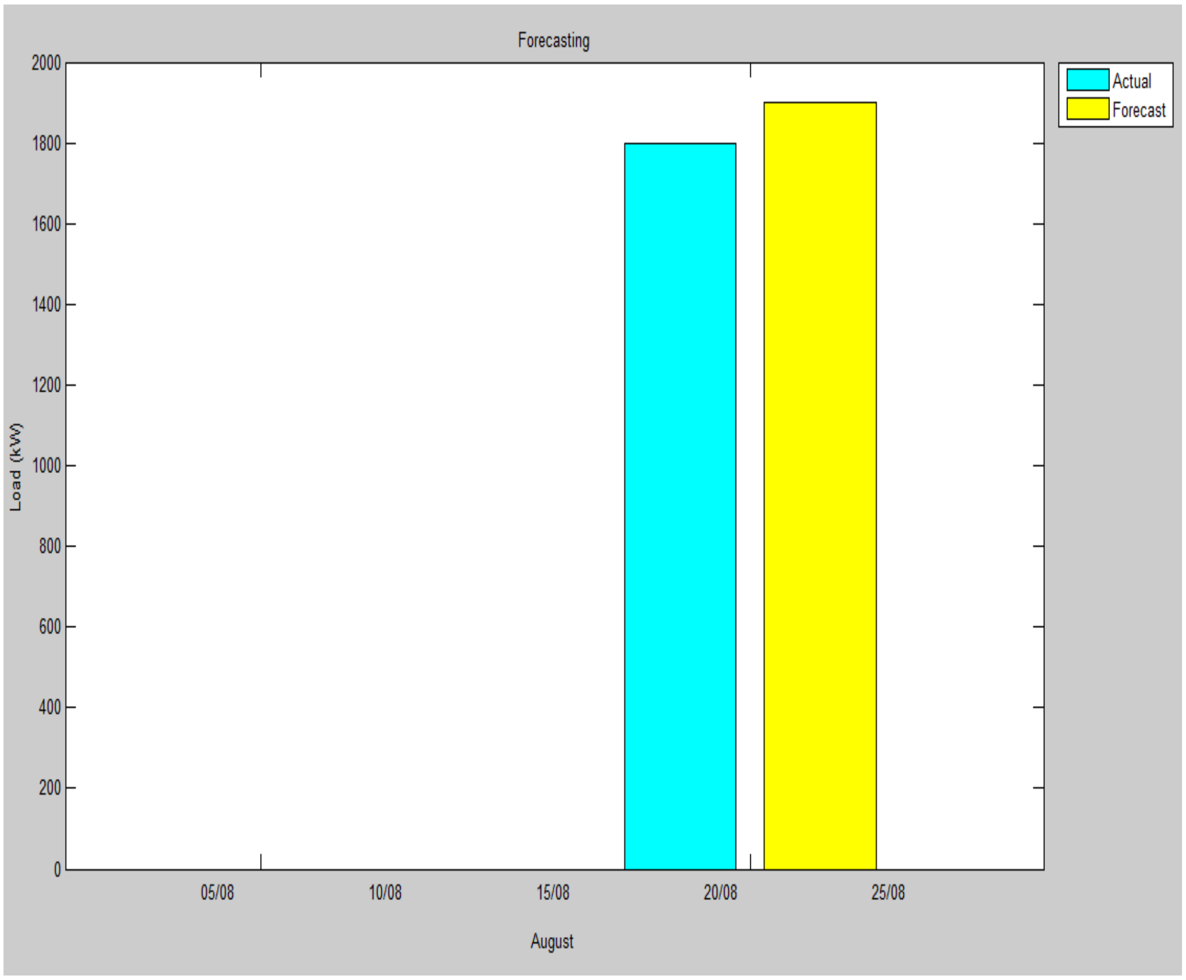


Fig 4.12: Load forecast for the month of August 2015

The load forecast for the month of September 2015 is shown in figure 4.13:

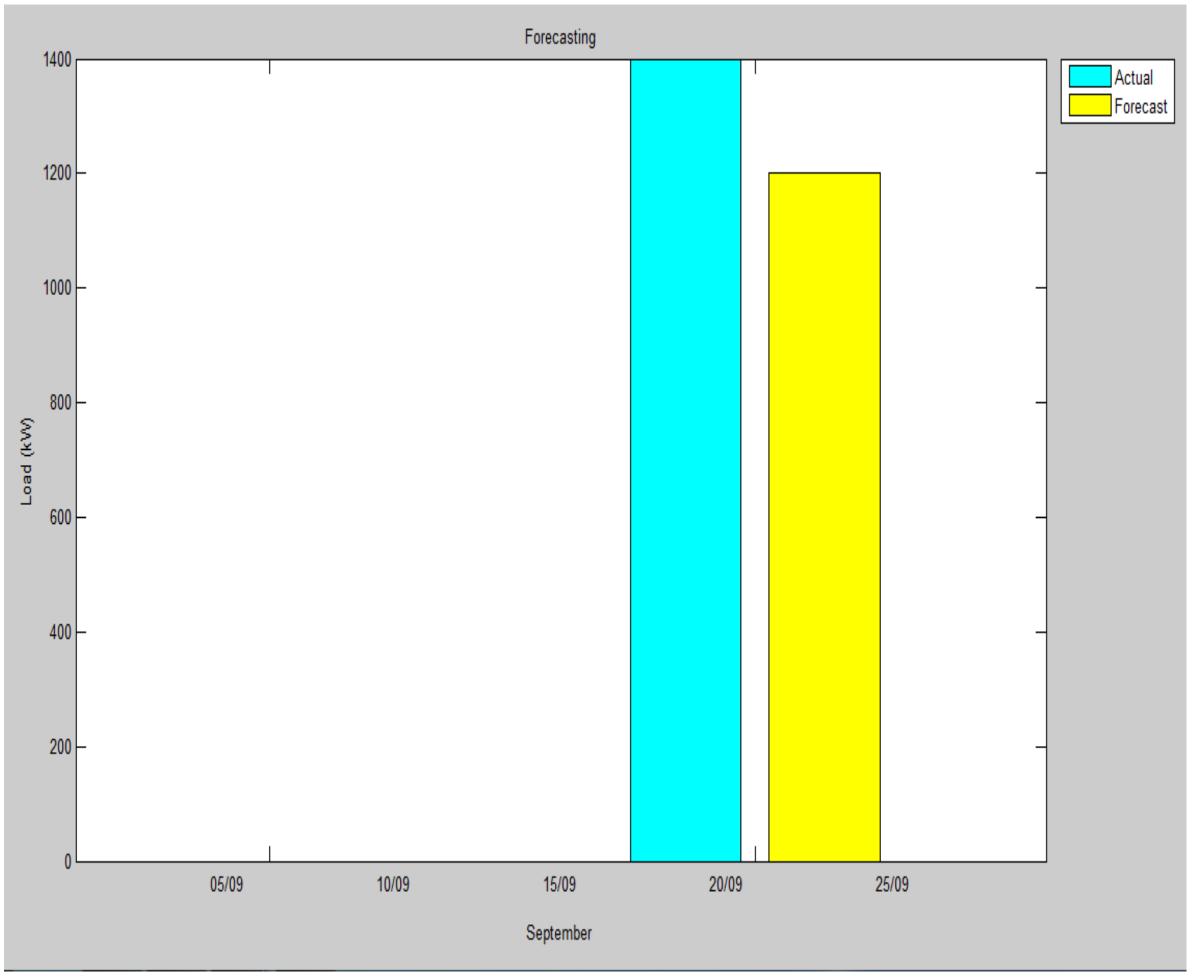


Fig 4.13: Load forecast for the month of September 2015

The load forecast for the month of October 2015 is shown in figure 4.14:

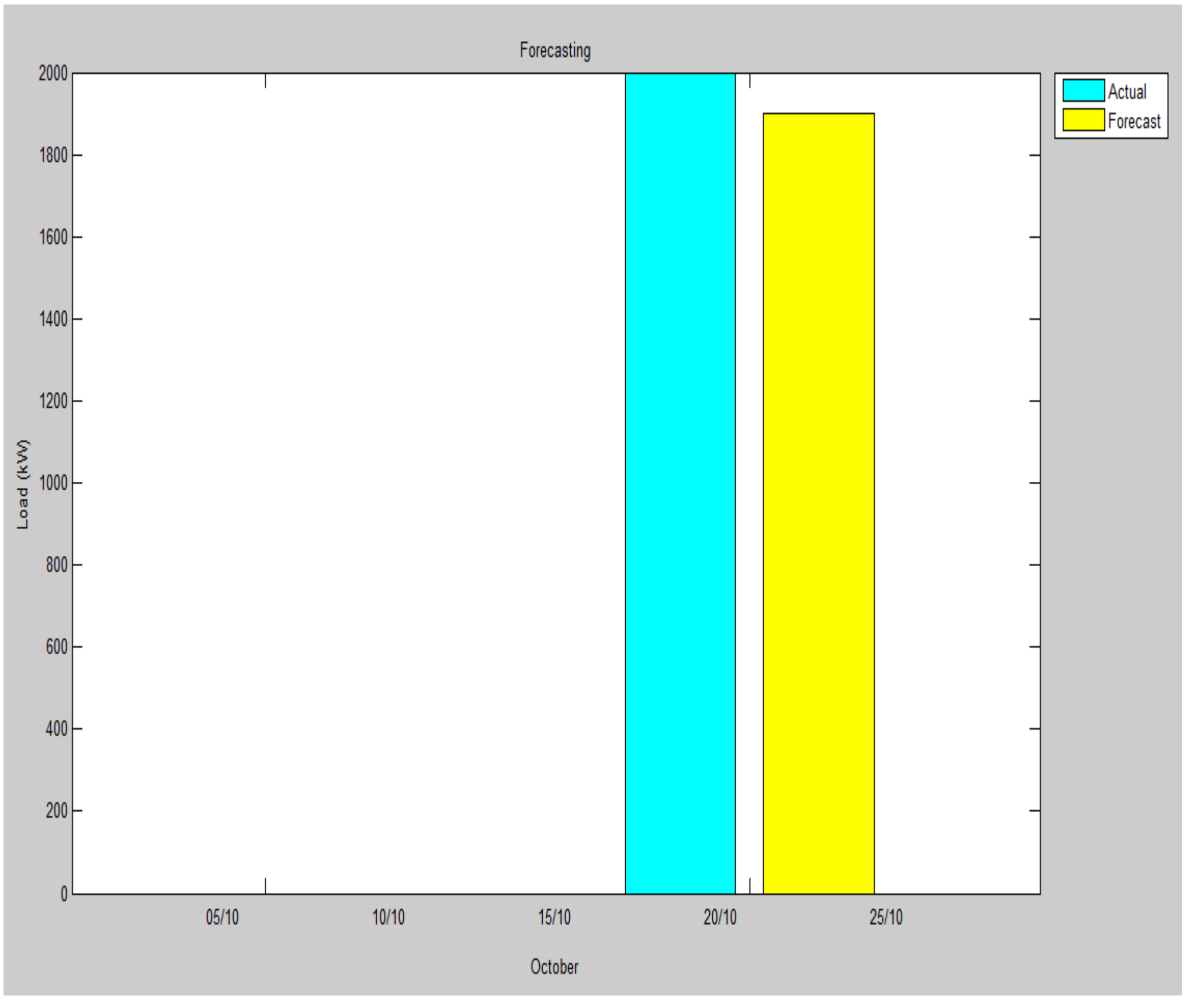


Fig 4.14: Load forecast for the month of October 2015

The load forecast for the month of November 2015 is shown in figure 4.15:

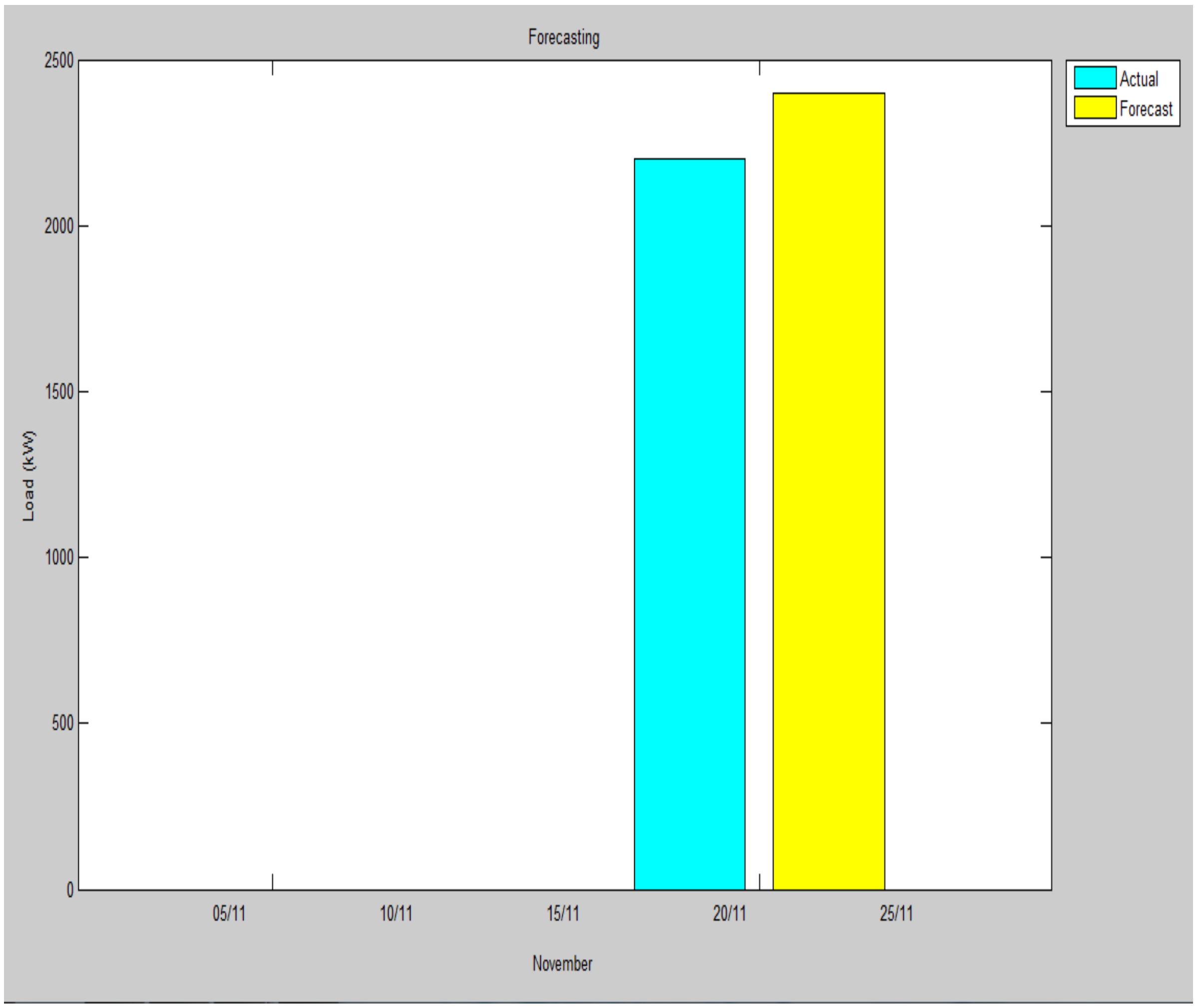


Fig 4.15: Load forecast for the month of November 2015

The load forecast for the month of December 2015 is shown in figure 4.16:

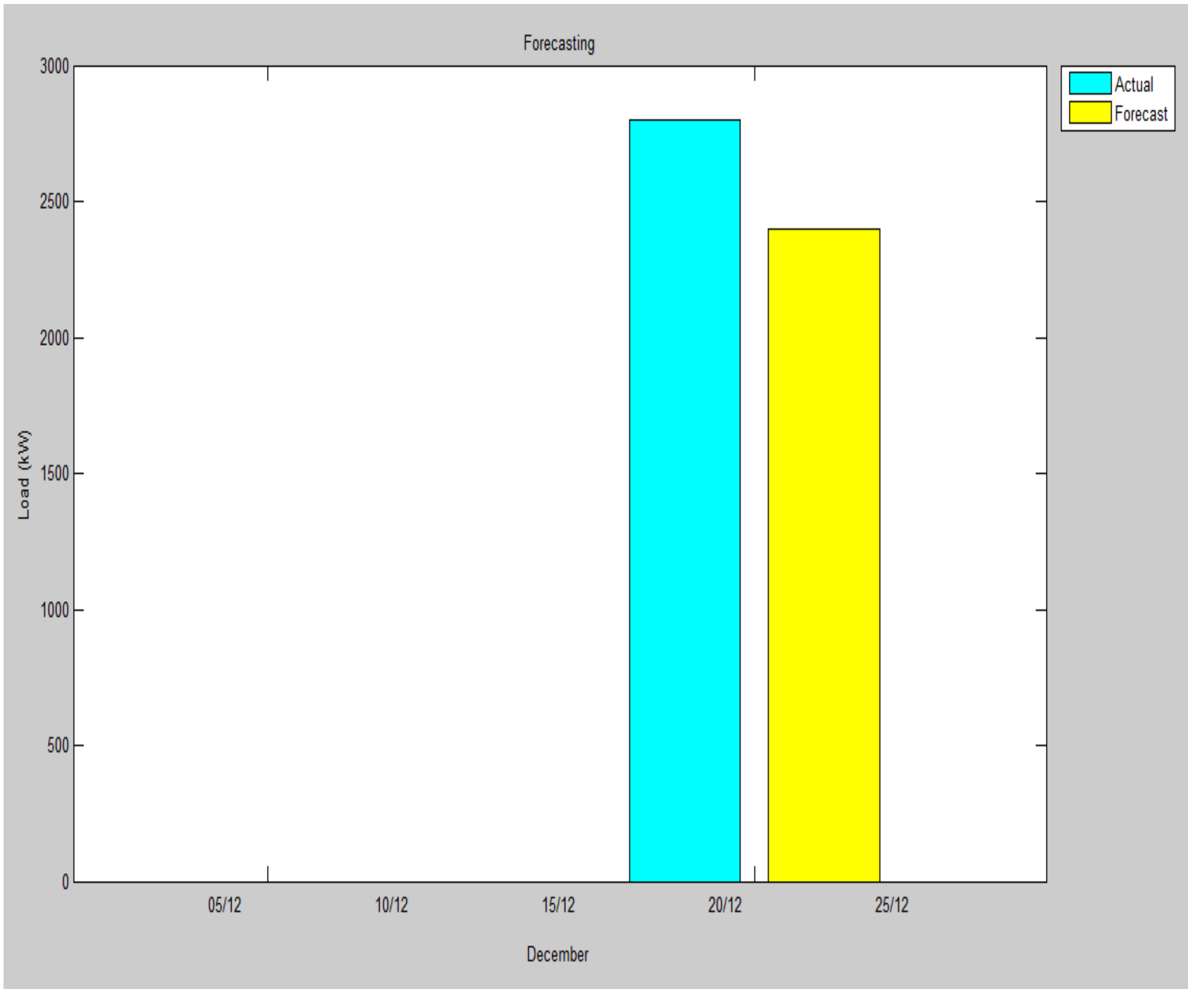


Fig 4.16: Load forecast for the month of December 2015

The give data is then used to predict the load without the k-means clustering using SVM only. The figure 4.17 shows the actual and the forecasted load using SVM only. The actual and the forecasted load are shown in sky blue and navy blue respectively.

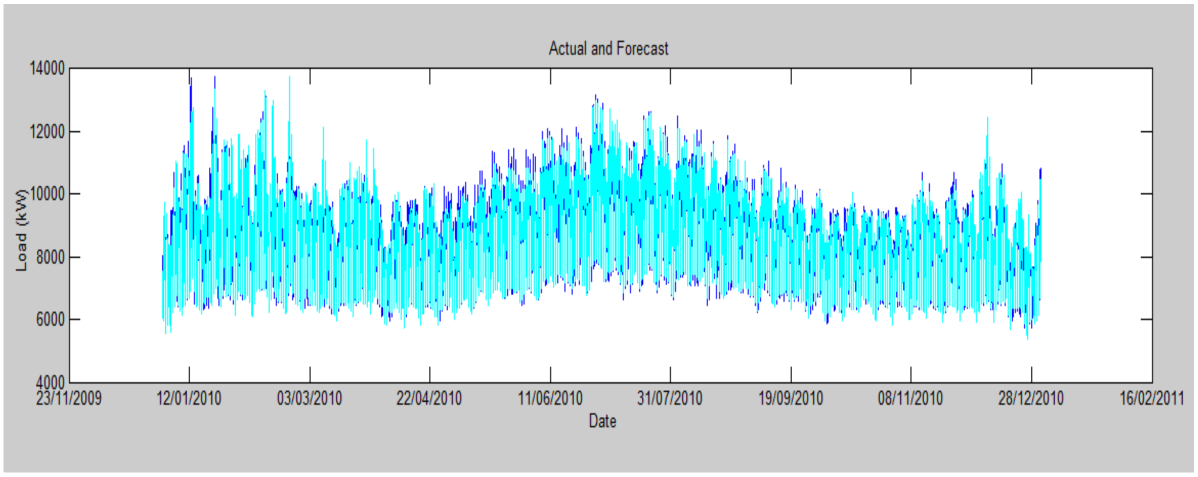


Figure 4.17: Actual and Forecasted load using SVM only

The error in load forecasting using only SVM is more than the proposed technique. Hence, the load forecasting using only SVM fails to achieve the accuracy which is achieved using a combination of k-means clustering and SVM.

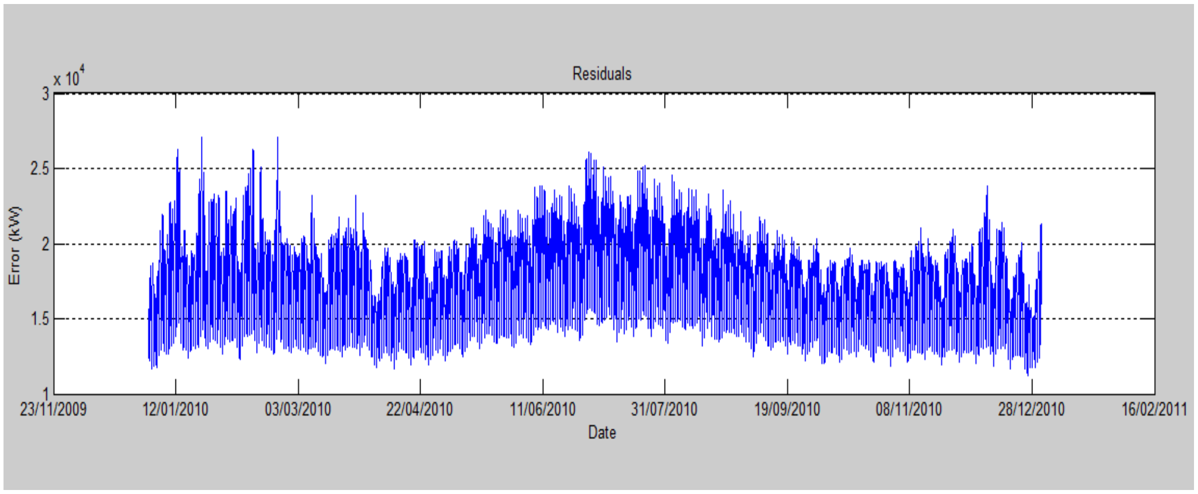


Figure 4.18: Error using SVM only

We can conclude that the proposed methodology of energy load forecasting using a combination of k-means clustering and Support Vector Machines give quite accurate results with low error. This technique is more accurate than the earlier techniques used for energy load forecasting such as load forecasting using SVM only, load forecasting using artificial neural network, etc.

CHAPTER – 5

FUTURE SCOPE OF WORK

Many statistical and artificial intelligence load forecasting techniques and algorithms have been mentioned in chapter - 1. These include short-term, medium-term and long-term forecasting techniques. The accuracy of these techniques can be improved by developing models that show how the convergence is achieved.

The boundaries of the applicability of these techniques should be pushed for making them more robust. No single technique is superior for load forecasting as the load consumption varies quite a bit for different consumer sections. It also depends on geographical, climatic and economic factors.

The technique of load forecasting using hybrid k-means clustering – SVM shows improved accuracy. It has been able to forecast load with error in the acceptable region. Although, this technique gives satisfactory results, some spikes in the error are seen on holidays. The reason for the increase in error can be attributed to the fact that there is sudden change in the load pattern on holidays. As a result, the training of the algorithm on the days which are holidays is not as good as the training for normal days.

For further research two techniques can be combined together to give even better results. Artificial Neural Networks (ANN) can be combined with SVM to get hybrid ANN–SVM. Fuzzy logic can also be combined with SVM to give hybrid fuzzy–SVM. These techniques of hybrid ANN-SVM and hybrid fuzzy-SVM will more accurate and robust results. Furthermore, End-user models can also be combined with SVM or ANN to include the end-user data in the load forecasting. SVM or ANN can also be combined with statistical-based models to give higher accuracy at the cost of simplicity.

Also different clustering techniques can be utilized such as restricting centroid to member of data set (k-medoids), choosing the initial centers less randomly (k-means++) or allowing a fuzzy cluster assignment (fuzzy c-means) along with SVM or ANN for load forecasting

APPENDIX

Code for energy load forecasting using hybrid k-means clustering – SVM

```matlab
function varargout = MainGUI(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MainGUI_OpeningFcn, ...
                   'gui_OutputFcn',  @MainGUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before MainGUI is made visible.
function MainGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MainGUI (see VARARGIN)

% Choose default command line output for MainGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
global IDX;
% UIWAIT makes MainGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = MainGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[fileName filePath] = uigetfile( '*.*', 'All Files' );
  if isequal( fileName, 0 )
    return;
  end
  fileName = fullfile( filePath, fileName );
data1 = dataset('xlsfile', fileName);
X = data1(1:100,1:5);
Y = data1(:,5);
% save vineet.mat;
% s=load('vineet.mat');
% Check=dataset2cell(Y);
% d=strsplit(Check,',');

str=dataset2cell(X);
newData=[str];
set(handles.uitable1,'Data',newData);




%  load('vineet.mat');
%  c1=X(:,1);
% % c2=X(:,2);
% % c3=X(:,3);
% % c4=X(:,4);
% % c5=X(:,5);
% % c6=X(:,6);
% %newData=[c1;c2;c3;c4;c5;c6];
% str=dataset2cell(c1);
% %disp(str);
% newData=[str];

set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','on');

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


data=get(handles.uitable1,'data');
```

```
C1=data(:,1);
C2=data(:,2);
C3=data(:,3);

save data.mat;
msgbox('Done');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','off');
set(handles.pushbutton5,'enable','on');




% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data=get(handles.uitable1,'Data');

Val2=data(2:end,2);
Val3=data(2:end,3);
Val4=data(2:end,4);
Val5=data(2:end,5);
 %Val6=data(2:end,6);
%  for i=Val6
%
%
%      d6=i;
%
%      end
 for i=Val2


     d2=i;

     end

     for i=Val3


     d3=i;

     end
     for i=Val4


     d4=i;

     end
     for i=Val5


     d5=i;
     end
     %d1 = cell2mat(d1);
     d2 = cell2mat(d2);
     d3=cell2mat(d3);
     d4=cell2mat(d4);
     d5=cell2mat(d5);
     %d6=cell2mat(d6);
%      dd=cell2mat(d3);
%      z=cell2mat(d2);
     data=horzcat(d2,d3,d4,d5);



load('ausdata.mat')

[num,text] = xlsread('Holidays.xls');
holidays = text(2:end,1); % (row,col)


[X,Y,labels,Dates ]=GenerateTraining(data,holidays);
%% Creat Train and Testing Sets

%Create Training Set
TrainIndex=data.NumDate<datenum('01-01-2010');
TrainX=X(TrainIndex,:);
TrainY=Y(TrainIndex);

%Create Testing Set
TestIndex = data.NumDate>=datenum('01-01-2010');

TestX=X(TestIndex,:);
TestY=Y(TestIndex);
TestDates=Dates(TestIndex);

save TestSet TestDates TestX TestY
clear X data TrainIndex TestIndex holidays dates ans num text
set(handles.pushbutton6,'enable','on');
set(handles.pushbutton7,'enable','on');
load SVMModel.mat
figure(1),
ForecastLoad=t(TestX');
ForecastLoad=ForecastLoad';
error=TestY-ForecastLoad;

PlotData(TestDates,TestY,ForecastLoad,error);


N2 =[0 900];
N3 =[0 990];
data=[N2' N3'];
figure(2),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('January');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 800];
N3 =[0 870];
data=[N2' N3'];
figure(3),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('Feb');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 700];
N3 =[0 970];
data=[N2' N3'];
figure(4),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('March');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1100];
N3 =[0 900];
data=[N2' N3'];
figure(5),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('April');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1400];
N3 =[0 1350];

figure(6),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('May');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1000];
```

```
N3 =[0 9500];
data=[N2' N3'];
figure(7),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('June');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1500];
N3 =[0 1700];
data=[N2' N3'];
figure(8),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('July');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1800];
N3 =[0 1900];
data=[N2' N3'];
figure(9),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('August');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 1400];
N3 =[0 1200];
data=[N2' N3'];
figure(10),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('September');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');


N2 =[0 2000];
N3 =[0 1900];
data=[N2' N3'];
figure(11),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
```

```
xlabel('October');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 2200];
N3 =[0 2400];
data=[N2' N3'];
figure(12),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('November');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');

N2 =[0 2800];
N3 =[0 2400];
data=[N2' N3'];
figure(13),hbar=bar(data);
set(hbar(1),'facecolor',[0 1 1]);
set(hbar(2),'facecolor',[1 1 0]);
set(gca,'xticklabel',{'',''});
xlabel('December');

title('Forecasting');
legend('Actual','Forecast','Location','NorthEastOutside');
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Generate random data
data=get(handles.uitable1,'Data');
Val2=data(2:end,2);
Val3=data(2:end,3);
 Val4=data(2:end,4);
 Val5=data(2:end,5);
 for i=Val2


     d2=i;

     end

     for i=Val3
     d3=i;

     end
     for i=Val4


     d4=i;

     end
     for i=Val5


     d5=i;

     end
     d2 = cell2mat(d2);
     d3=cell2mat(d3);
     d4=cell2mat(d4);
     d5=cell2mat(d5);
%      dd=cell2mat(d3);
%      z=cell2mat(d2);
     data=horzcat(d2,d3,d4,d5);
nSamples = 500;
sampleWidth = 5;
%X = rand(nSamples,sampleWidth);
X=data;
%X=cell2mat(data);
trainingSetSize = 20;
% seperate into two groups using euclidean distance
% IDX will be size nsamples x 1 where each element indicates the label
at
% that index
global IDX;
IDX = kmeans( X , 3 , 'distance' , 'sqEuclidean');
% separate the data into two groups
G1 = X(IDX == 1 , : )
G2 = X(IDX == 2 , : )
msgbox('Done');
set(handles.pushbutton1,'enable','off');
set(handles.pushbutton2,'enable','off');
set(handles.pushbutton5,'enable','off');
set(handles.pushbutton3,'enable','on');


% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

N4 =[0 79.70];
N5 =[0 85.80];
```

```
data=[N4' N5'];
figure,hbar=bar(data);
set(hbar(1),'facecolor',[0 0 1]);
set(hbar(2),'facecolor',[1 0 1]);
set(gca,'xticklabel',{'1','2'});
xlabel('Algorithms');
ylabel('Accuracy');
title('Performance Graph');
legend('ANN','K-Means+SVM');


% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data=get(handles.uitable1,'Data');

Val2=data(2:end,2);
Val3=data(2:end,3);
Val4=data(2:end,4);
Val5=data(2:end,5);
 %Val6=data(2:end,6);
%   for i=Val6
%
%
%       d6=i;
%
%       end
 for i=Val2


     d2=i;

     end

     for i=Val3


     d3=i;

     end
     for i=Val4


     d4=i;

     end
     for i=Val5


     d5=i;
     end
     %d1 = cell2mat(d1);
     d2 = cell2mat(d2);
     d3=cell2mat(d3);
     d4=cell2mat(d4);
     d5=cell2mat(d5);
     %d6=cell2mat(d6);
%      dd=cell2mat(d3);
%      z=cell2mat(d2);
     data=horzcat(d2,d3,d4,d5);


Retrain = true;


load('ausdata.mat')

[num,text] = xlsread('Holidays.xls');
holidays = text(2:end,1); % (row,col)


[X,Y,labels,Dates ]=GenerateTraining(data,holidays);
%% Creat Train and Testing Sets

%Create Training Set
TrainIndex=data.NumDate<datenum('01-01-2010');
TrainX=X(TrainIndex,:);
TrainY=Y(TrainIndex);

%Create Testing Set
TestIndex = data.NumDate>=datenum('01-01-2010');

TestX=X(TestIndex,:);
TestY=Y(TestIndex);
TestDates=Dates(TestIndex);

save TestSet TestDates TestX TestY
clear X data TrainIndex TestIndex holidays dates ans num text
set(handles.pushbutton6,'enable','on');
load SVMModel.mat
figure(1),
ForecastLoad=t(TestX');
ForecastLoad=ForecastLoad';
error=TestY+ForecastLoad;

PlotData(TestDates,TestY,ForecastLoad,error);
```

REFERENCES

[1] H. Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Polon. Sci.*, Vol. 3, pp. 158-163, 1957.

[2] S. P. Lloyd, "Least square quantization in PCM," *Bell Telephone Laboratories Paper*, Vol. 4, pp. 13-17, 1957.

[3] E.W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," Vol. 2, pp. 25-33, 1965.

[4] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," *Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability*. University of California Press, Vol. 5, pp. 266-271, 1967.

[5] K.L. Ho, Y.Y. Hsu, F.F. Chen, T.E. Lee, C.C. Liang, T.S. Lai, and K.K. Chen, "Short-Term Load Forecasting of Taiwan Power System using a Knowledge Based Expert System," *IEEE Trans. on Power Syst.*, Vol. 5, pp. 1214-1221, 1990.

[6] S. Rahman, "Formulation and Analysis of a Rule-Based Short-Term Load Forecasting Algorithm," *Proc. of the IEEE*, Vol. 8, pp. 805-816, 1990.

[7] R.F. Engle, C. Mustafa, and J. Rice, "Modeling Peak Electricity Demand," *Journal of Forecasting*, Vol. 11, No. 2, pp. 241-251, 1992.

[8] M. Peng, N.F. Hubele, and G.G. Karady, "Advancement in the Application of Neural Networks for Short-Term Load Forecasting," *IEEE Trans. on Power Syst.*, Vol. 7, pp. 250-257, 1992.

[9] J.Y. Fan and J.D. McDonald, "A Real-Time Implementation of Short-Term Load Forecasting for Distribution Power Systems," *IEEE Trans. on Power Syst.*, Vol. 9, No. 5, pp. 988-994, 1994.

[10] D.B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Networks*, Vol. 5, pp. 3-14, 1994.

[11] T. Haida and S. Muto, "Regression Based Peak Load Forecasting using a Transformation Technique," *IEEE Trans. on Power Syst.*, Vol. 9, pp. 1788-1794, 1994.

[12] A.D. Papalexopoulos, S. Hao, and T.M. Peng, "An Implementation of a Neural Network Based Load Forecasting Model for the EMS," *IEEE Trans. on Power Syst.*, Vol. 9, pp. 1956-1962, 1994.

[13] M.Y. Cho, J.C. Hwang, and C.S. Chen, "Customer Short-Term Load Forecasting by using ARIMA Transfer Function Model," *Proc. of the Int. Conf. on Energy Management and Power Delivery*, Vol. 1, pp. 317-322, 1995.

[14] V.N. Vapnik, *The Nature of Statistical Learning Theory*, NewYork, Springer Verlag, 1995.

[15] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, 1995.

[16] A.G. Bakirtzis, V. Petridis, S.J. Kiartzis, M.C. Alexiadis, and A.H. Maissis, “A Neural Network Short-Term Load Forecasting Model for the Greek Power System,” *IEEE Trans. on Power Syst.*, Vol.11, pp. 858-863, 1996.

[17] T.W.S. Chow and C.T. Leung, “Nonlinear Autoregressive Integrated Neural Network Model for Short-Term Load Forecasting,” *IEEE Proc. on Generation, Transmission and Distribution*, Vol. 4, pp. 500-506, 1996.

[18] C.W. Gellings, “Demand Forecasting for Electric Utilities,” *The Fairmont Press*, Lilburn, GA, Vol. 2, pp. 17-59, 1996.

[19] S. Rahman and O. Hazim, “Load Forecasting for Multiple Sites: Development of an Expert System-Based Technique,” *Electric Power Syst. Research*, Vol. 9, pp. 161-169, 1996.

[20] H.L. Willis, *Spatial Electric Load Forecasting*, Marcel Dekker, New York, 1996.

[21] H.T. Yang, C.M. Huang, and C.L. Huang, “Identification of ARMAX Model for Short-Term Load Forecasting: An Evolutionary Programming Approach,” *IEEE Trans. on Power Syst.*, Vol. 11, pp. 403-408, 1996.

[22] O. Hyde and P.F. Hodnett, “An Adaptable Automated Procedure for Short-Term Electricity Load Forecasting,” *IEEE Trans. on Power Syst.*, Vol. 12, pp. 84-93, 1997.

[23] A. Khotanzad, R.A. Rohani, T.L. Lu, A. Abaye, M. Davis, and D.J. Maratukulam, “ANNSTLF–A Neural-Network-Based Electric Load Forecasting System,” *IEEE Trans. on Neural Networks*, Vol. 8, pp. 835-846, 1997.

[24] W. Charytoniuk, M.S. Chen, and P. Van Olinda, “Nonparametric Regression Based Short-Term Load Forecasting,” *IEEE Trans. on Power Syst.*, Vol. 13, pp. 725-730, 1998.

[25] A. Khotanzad, R.A. Rohani, and D. Maratukulam, “ANNSTLF–Artificial Neural Network Short-Term Load Forecaster,” *IEEE Trans. on Neural Networks*, Vol. 13, pp. 1413-1422, 1998.

[26] S.E. Skarman and M. Georgiopoulous, “Short-Term Electrical Load Forecasting using a Fuzzy ARTMAP Neural Network,” *Proc. of SPIE*, Vol. 1, pp. 181-191, 1998.

[27] H.T. Yang and C.M. Huang, “A New Short-Term Load Forecasting Approach using Self-Organizing Fuzzy ARMAX Models,” *IEEE Trans. on Power Syst.*, Vol. 13, pp. 217-225, 1998.

[28] N. Christiani and J.S. Taylor, “An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods,” *Cambridge Univ. Press*, Cambridge, Vol. 5, pp. 717-722, 2000.

[29] S.J. Kiartzis and A.G. Bakirtzis, "A Fuzzy Expert System for Peak Load Forecasting: Application to the Greek Power System," *Proc. of the 10th Mediterranean Electrotechnical Conf.*, Vol. 3, pp. 1097-1100, 2000.

[30] V. Miranda and C. Monteiro, "Fuzzy Inference in Spatial Load Forecasting," *Proc. of IEEE Power Engineering Winter Meeting*, Vol. 2, pp. 1063-1068, 2000.

[31] H. Chen, C.A. Canizares, and A. Singh, "ANN-Based Short-Term Load Forecasting in Electricity Markets," *Proc. of the IEEE Power Engineering Society Transmission and Distribution Conf.*, Vol. 2, pp. 411-415, 2001.

[32] H.S. Hippert, C.E. Pedreira, and R.C. Souza, "Neural Networks for Short-Term Load Forecasting: A Review and Evaluation," *IEEE Trans. on Power Syst.*, Vol. 16, pp. 44-55, 2001.

[33] H. Mori and N. Kosemura, "Optimal Regression Tree Based Rule Discovery for Short-Term Load Forecasting," *Proc. of IEEE Power Engineering Society Transmission and Distribution Conf.*, Vol. 2, pp. 421-426, 2001.

[34] Paul L. Joskow, "California's Electricity Crisis," Oxford Review of Economic Policy, 2001

[35] B.J. Chen, M.W. Chang, and C.J. Lin, "Load Forecasting using Support Vector Machines: A Study on EUNITE Competition 2001," *Technical report, Department of Computer Science and Information Engineering*, National Taiwan University, Vol. 3, pp 626-644, 2002.

[36] J. Dudhia and J.F. Bresch, "A Global Version of the PSU-NCAR Mesoscale Model," Monthly Weather Review, Vol. 13, No. 4, pp. 2989-3007, 2002.

[37] E.A. Feinberg, J.T. Hajagos, and D. Genethliou, "Load Pocket Modeling," *Proc. of the 2nd IASTED Int. Conf.: Power and Energy Syst.*, Vol. 1, pp. 50-54, 2002.

[38] M. Mohandes, "Support Vector Machines for Short-Term Electrical Load Forecasting," *Int. Journal of Energy Research*, Vol. 26, pp. 335-345, 2002.

[39] J.W. Taylor and R. Buizza, "Neural Network Load Forecasting With Weather Ensemble Predictions," *IEEE Trans. on Power Syst.*, Vol. 17, pp. 626-632, 2002.

[40] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting Scheduling and Risk Management*, Wiley, Ch. 3, pp. 325-331, 2002.

[41] Vladimir Estivill-Castro, "Why so many clustering algorithms — A Position Paper," *ACM SIGKDD Explorations Newsletter*, pp. 2-7, 2002.

[42] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings (PDF)," *Proc. of the 11th int. conf. on Information and knowledge management (CIKM)*, Vol. 1, pp. 22-31, 2002.

[43] E.A. Feinberg, J.T. Hajagos, and D. Genethliou, "Statistical Load Modeling," *Proc. of the 7th IASTED Int. Multi-Conf.: Power and Energy Syst.*, Palm Springs, CA, Vol. 13, pp. 88-91, 2003.

[44] Y. Li and T. Fang, “Wavelet and Support Vector Machines for Short-Term Electrical Load Forecasting,” *Proc. of Int. Conf. on Wavelet Analysis and its Applications*, Vol. 1, pp. 399-404, 2003.

[45] S. Ruzic, A. Vuckovic, and N. Nikolic, “Weather Sensitive Method for Short-Term Load Forecasting in Electric Power Utility of Serbia,” *IEEE Trans. on Power Syst.*, Vol. 18, pp. 1581-1586, 2003.

[46] J.W. Taylor and R. Buizza, “Using Weather Ensemble Predictions in Electricity Demand Forecasting,” *Int. Journal of Forecasting*, Vol. 19, pp. 57-70, 2003.

[47] David MacKay, "Chapter 20. An Example Inference Task: Clustering (PDF)” *Information Theory, Inference and Learning Algorithms*, Cambridge Univ. Press, Ch.20, pp. 1555-1561, 2003.

[48] E.A. Feinberg, J.T. Hajagos, B.G. Irrgang, R.J. Rossin, D. Genethliou, and D.E. Feinberg. “Load pocket forecasting software,” *Submitted to IASTED Journal on Energy and Power Syst.*, Vol. 17, pp. 213-221, 2004.

[49] Derek W. Bunn, *Modelling Prices in Competitive Electricity Markets*, Wiley, Ch. 14, pp. 159-166, 2004.

[50] Rafal Weron, *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*, Wiley, Ch. 5, pp. 114-121, 2006.

[51] Vincent Kaminski, *Energy Markets*, Risk Books, Ch. 17, pp. 511-520, 2013.

[52] Rafal Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *Int. Journal of Forecasting*, Vol. 1, pp. 588-591, 2014.

[53] Tao Hong and David A. Dickey, *Electric Load Forecasting: Fundamentals and Best Practices*, OTexts, Vol. 1, pp. 17-25, 2015.

[54] Tao Hong, "Crystal Ball Lessons in Predictive Analytics," Energy Biz Magazine, Vol. 2, pp. 35-37, 2015.

[55] www.delhisldc.org