Comparative Analysis of Model fitting using Time Series Models and Neuro-Fuzzy based ASuPFuNIS model
Course : SYDE 631

Instructor : Prof. Keith W. Hipel
by

Apurva Narayan
Supervisor : Prof. K. Ponnambalam

A project report
presented to the University of Waterloo
in fulfillment of the
course requirement for the degree of
Master of Science in
Systems Design Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this report. This is a true copy of the project report, including any required final revisions, as accepted by my instructor.

Abstract

This project report summarizes comprehensively analsyis of time series models fitting to real world data such as electric prices for a period of 7 years. It compares the time series model with a Neuro-Fuzzy based model called AsuPFuNIS[10] for forecasting the time series data for one step ahead forecasts and subsequently further forecasts. We use extensively the exploratory data analysis for establishing a suitable match between data and the time series models. Ensuring effective exploratory data analysis we use confirmatory data anaylysis tools to finally get the most accurate model to fit the data. Three stages of model construction as identified in [4] have been adhered.

Later on, we compare the accuracy of the forecasts made using a time series model to fit the data with the forecasts made using the Neuro-Fuzzy Inference model. At the end comparative results are shown.

Contents

List of Tables

List of Figures

Chapter 1

Scope and Background of Time Series Modelling Applications

A time series is a sequence of data points, measured typically at successive times spaced at uniform time intervals. Examples of time series are precipitation in a localized area, another example of a time series can be stock prices. Hence Time Series can comprise of any data which can be analysed over a given period of time. Time series analysis comprises methods for analyzing data in order to extract meaningful statistics and other characteristics of the data. *Time series forecasting* is the use of a model to forecast future events based on known past events i.e. to predict data points before they are measured. An example of time series forecasting in econometrics is predicting the opening price of a stock based on its past performance, another very good example of time series forecasting can be predicting the pollution levels in a river based on previous trends, generally time series analysis and forecasting finds immense application in *Hydrology*. Time series are usually plotted via line charts.

Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis techniques, in which there is no natural ordering of the observations. Time series analysis is also distinct from spatial data analysis where the observations typically relate to geographical locations. A time series model will generally reflect the fact that observations close together in time will be more closely related than observations further apart. In addition, time series models will often make use of the natural one-way ordering of time so that values for a given period will be expressed as deriving in some way from past values, rather than from future values. Methods for time series analysis may be divided into two classes: frequency-domain methods and time-domain methods. The former include spectral analysis and recently wavelet analysis; the latter include auto-correlation and cross-correlation analysis.

Here in this report we will focus on time-domain methods. We will analyze a data series of electricity prices. We will focus on various kind of time series models : Nonseasonal Stationary models, Nonseasonal NonStationary models, Seasonal Models, Parameteric Models. We are also going to focus on general model construction technique.

- Model Identification
- Parameter Estimation
- Diagnostic Checking

We are going to study in details the the two major techniques

- Exploratory Data Analysis
- Confirmatory Data Analysis

Later in the report we will compare results of a time series model fit to real electricity prices to a Neuro-Fuzzy based model used for time series prediction.

1.1 Basic Statistical Concepts

Natural phenomenon are usually difficult to predict deterministically what will happen in the future. Hence we realize such phenomenon as something which could possibly happen in the future. Example : Precipitation is an example of a statistical phenomenon that evolves in time based on probabilistic laws. *A mathematical expression which describes the probability structure of the time series that was observed due to the phenomenon is referred to as a Stochastic Process.* The historical observations are a sample realization of the stochastic process that produced them[4].

Now fitting a time series or stochastic model to the time series for use in application is called *Time Series Modelling*. One main idea of time series analysis is to make the inferences regarding the basic features of the stochastic process from the information contained in the historical time series. This is accomplished by developing mathematical model which posses same statiscal charactersitics as the generating mechanism of the stochastic process. Henceforth the fitted model can be used for various purposes as forecasting and simulation[4].

1. Stationarity

   Stationarity of a stochastic process can be qualitatively interpreted as a form of statistical equilibrium. Hence more generally speaking it means that the statistical properties are not a function of time. It is analogous to the concept of *isotropy* in physics[4].

   - Strong Stationarity
   - Weak Stationarity
   - Second Order Stationarity or Covariance Stationarity

2. Statistical Definitions Let $z_1, z_2, \ldots, z_N$ be a time series of N values then :

   (a) Mean

   $$\mu = E[z_t]$$
   $$\mu = \bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i$$

   (b) Variance

   $$\hat{\sigma_z^2} = \frac{1}{N}\sum_{i=1}^{N}(z_i - \bar{z})^2$$

   (c) Autocovariance and Autocorrelation : The covariance between z_t and a value z_{t+k} which is k time lags removed from z_t is theoretically defined in terms of the autocovariance at lag k given by the following relation

   $$\gamma_k = cov[z_t, z_{t+k}] = E[(z_t - \mu)(z_{t+k} - \mu)]$$

   When $k = 0$, the autocovariance is the variance and consequently $\gamma_0 = \sigma_z^2$. A normalized quantity that is more convenient to deal with than γ_k, is the theorectical autocorrelation coefficient which is defined at lag k as

   $$\rho_k = \frac{\gamma_k}{\gamma_0}$$

   [1] refers the former as *autocovariance* and the latter as *autocorrelation function.*

3. Ergodicity : As mentioned in [4] *A desirable property of an estimator is that as the sample size increases the estimator converges with probability one to the population parameter being estimated.* An estimator possesing this property is referred to as *consistent estimator.* In order for the estimator to be consistent, the stochastic process must posses what is called *ergodicity.*

Hence to conclude a stationary time series can be described by the above mentioned statistical properties.

Chapter 2

Time Series Models

In general we are going to define here a variety of *Linear Time Series Models* that can be applied to various nonseasonal time series.

2.1 Nonseasonal Models

Certain set of time series which do not follow a pattern which repeats after a given interval are referred to as Nonseasonal time series . Generally speaking if we refer to a time series dealing with temperature throughout the year at a given location, the series may not be seasonal if taken for a year but will be seasonal if looked for a long duration, say a couple of years. If the mean of the time series is not changing with time and the variance is constant with time then such time series is referred to as *Stationary Time Series.*

2.1.1 Stationary Nonseasonal Models

In this subsection we are going to discuss 3 major models for Stationary Nonseasonal Time Series.

- Autoregressive (AR)
- Moving Average (MA)
- Autoregressive and Moving Average (ARMA)

Autoregressive (AR)

As explained in [4] and [1] the AR model describes how an observation directly depends upon one or more previous observations and a additional white noise term. Such a model is intuitive. We will mention here various statiscal properties of the AR model and then also mention a special case of AR model called the *Markov Process*.

An AR process of order p *(AR(p))* is given as below :

- Autoregressive Process of order p

$$(z_t - \mu) = \phi_1(z_{t-1} - \mu) + \phi_2(z_{t-2} - \mu) + \ldots + \phi_p(z_{t-p} - \mu) + a_t$$

In the above expression ϕ_i is the i^{th} *nonseasonal AR parameter*. Using a the backward shift operator B^1, the above equation can be equivalently written as

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(z_t - \mu) = a_t$$
$$\phi(B)(z_t - \mu) = a_t$$

where $(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)$ is referred to as the *nonseasonal AR operator of order p*.

A special case of AR process, AR(1) is commonly called as *Markov Process*. Here the current observation z_t measured at time t depends only upon the time series value at the *t-1* and a random shock or noise. The equation of AR(1) is given below :

$$(z_t - \mu) = \phi_1(z_{t-1} - \mu) + a_t$$

Here μ is the mean level of the process, ϕ_1 is the nonseasonal AR parameter, a_t is the white noise at the time t. It is assumed that a_t's [2] are *identically independently distributed(IID)* with mean 0 and variance σ_z^2.

- Stationarity

The equation $\phi(B) = 0$ is the characteristic equation of an AR process. Hence it has been shown by [1] that a necessary and sufficient condition for the process to have

[1] $Bz_t = z_{t-1}$

[2] $a_t's$ are generally referred to as random shocks, disturbances, innovations or white noise and the estimates of these innovations are referred to as *residuals*

stationarity is that the roots of the characterisitc equation must fall outside the *unit circle*[3].

The stationarity condition automatically ensures that the AR process can be written in terms of the a_t's in what is called a pure MA process ($(z_t - \mu) = (1 - \phi_1 B)^{-1} a_t$). Hence as $|\phi_1| < 1$ due to stationarity condition, this infers that the infinite series $(1 - \phi_1 B)^{-1}$ will converge for $|B| \leq 1$. Henceforth we can clearly state that the dependance of the current observation on the white noise decreased further in the past.

- Autocorrelation Function (ACF)
  The derivation of the ACF has been explained in detail in [4]. We will mention the results for a stationary AR(p) process. The normal procedure for evaluating ACF is to multiply the AR(p) process equation by $(z_{t-k} - \mu)$ and then taking expectation we get the autocovariance function for the AR(p) process and is given in [1] as :

$$\gamma_k = \phi_1 \gamma_{k-1} + \phi_2 \gamma_{k-2} + \ldots + \phi_p \gamma_{k-p}, \text{ where } k > 0$$

  An important point to remember while evaluating γ_k is that the term $E[(z_{t-k} - \mu)a_t]$ is zero for $k > 0$. Since z_{t-k} is only a function of the disturbances upto time $t - k$ and a_t is uncorrelated with these shocks. Now to determine the *Autocorrelation function* for the AR(p) process we divide the above relation by γ_0.

$$\begin{gathered}\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \ldots + \phi_p \rho_{k-p}, \text{ where } k > 0 \\ (1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)\rho_k = \phi(B)\rho_k = 0, k > 0 \\ \text{here B operates on } k \text{ instead of } t.\end{gathered}$$

  In this context, an important tool for evaluating ACF and the parameters are the Yule-Walker equations which are explained in detail in [4] at page 97.

- Partial Autocorrelation Function (PACF)
  Now from the point of view of model identification, the theorectical ACF for an AR process attenuates and does not truncate at a specified lag, it is advantageous to define a function which does cut off for an AR process.

  This is simply defined from the Yule walker equations as, ϕ_{kj} be the j^{th} coefficient in a stationary AR process of order k so that ϕ_{kk} is the last coefficient. The yule walker equation for PACF are given below :

[3] Unit Circle : It is a circle of unit radius centered at origin of a complex number graph where one axis is the real number component and the other forms the imaginary part of the complex number.

$$
\begin{bmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-2} \\ . & . & . & \dots & . \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ . \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ . \\ \rho_k \end{bmatrix} \tag{2.1}
$$

Here the coefficient ϕ_{kk} is a function of lag k and is called the *partial autocorrelation function (PACF)*. Therefore by the definition the theorectical PACF must be zero after lag p for an AR(p) process.

Moving Average Process (MA)

This model describes how an observation depends upon the current white noise term as well as one or more previous innovations.

- Moving Average Process of order 1
  As defined in [1] and [4] the time series value z_t, is dependent only upon the white noise at time t-1 plus the current shock and the relation ship is described as below :

$$z_t - \mu = a_t - \theta_1 a_{t-1}$$

  In the above expression θ_1 is the nonseasonal MA parameter. The above process is termed as MA process of order 1 and is denoted by MA(1). We can here as well write the above equation using the B operator :

$$\begin{aligned} z_t - \mu &= a_t - \theta_1 a_{t-1} \\ &= (1 - \theta_1 B a_t) \\ &= \theta(B) a_t \end{aligned}$$

  where $\theta(B) = 1 - \theta_1 B$ is the nonseasonal MA operator of order one.

- Moving Average Process of order q
  The MA(1) process can be readily extended to the the MA process of order q and the equation can be written as below :

$$z_t - \mu = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}$$

where θ_j is the jth nonseasonal MA parameter. We can also write the above process in a compact form using the B operator.

$$z_t - \mu = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q) a_t$$
$$z_t - \mu = \theta(B) a_t$$

where $\theta(B)$, is the nonseasonal MA operator of order q.

- Stationarity
  Since this time series is composed of a_t's which are assumed to be stationary. And the time series is a linear combination of a_t's, then z_t must be stationary no matter what values do the MA parameters posses.

- Invertibility
  The characterisitc equation of the MA(q) process defined above as $\theta(B) = 0$. In order for the MA(q) process to be invertible, the roots of the characteristic equation must lie outside the unit circle. Inherent advantage of having a invertible MA process is that it can be represented in the form of a pure AR process.

- Autocorrelation Function

  On similar grounds as we derived the equation for the autocorrelation function for the AR(p) process, similarly we can derive the ACF for the MA(q) process.

$$\gamma_k = E[(z_t - \mu)(z_{t-k} - \mu)]$$
$$= E[(a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q})(a_{t-k} - \theta_1 a_{t-k-1} - \theta_2 a_{t-k-2} - \cdots - \theta_q a_{t-k-q})]$$

After multiplication and taking expectation we get the following result :

$$\gamma_k = \begin{cases} (\text{-}\theta_k + \theta_1\theta_{k+1} + \theta_2\theta_{k+2} + \cdots + \theta_{q-k}\theta_q)\sigma_a^2, & \text{if } k \text{ is 1, 2, . . .q} \\ 0, & \text{if } k > \text{q} \end{cases}$$

where $\theta_0 = 1$ and $\theta_{-k} = 0$ for $k \geq 1$, when k = 0 in the equation then the variance is given by

$$\gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)\sigma_a^2$$

We can easily evaluate the theoretical ACF for an MA process by dividing γ_k by γ_0 to get ρ_k.

- Partial Autocorrelation Function
  Any finite invertible MA process can be expressed as an infinite AR process. The PACF is theoretically defined to be zero after lag p for a finite AR(p) process, the PACF must therefore attenuate at increasing lags for a MA process or equivalently an infinite AR process.

Autoregressive Moving Average Process

Now since we have already described the two basic model of AR and MA. A model which comprises of both AR and MA parameters is called an ARMA process.

- General Autoregressive Moving Average processor
  In general, an ARMA process may consist of p AR parameters and q MA parameters. Such a process is known as ARMA(p,q) process and is written as :

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(z_t - \mu) = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)a_t$$
$$\phi(B)(z_t - \mu) = \theta(B)a_t$$

  where $\phi(B)$ and $\theta(B)$ are AR and MA operators of orders p and q respectively.

- Stationarity and Invertibility
  The conditions for stationarity and invertibility hold the same way as they hold for MA and AR processes. Hence in order for the ARMA(p,q) process to be stationary the roots of $\phi(B) = 0$ should lie outside the unit circle and for the process to be invertible the roots of the charecteristic equation $\theta(B) = 0$ must lie outside the unit circle.

- Autocorrelation Function
  As explained above the the ACF of ARMA(p,q) process can be evaluated in a similar way as that done for the AR and MA process by multiplying the entire equation by $(z_{t-k} - \mu)$ and then taking expectation, the difference in evaluation of ACF for an ARMA process is that on the right hand side of the equation we get terms like $\gamma_{za}(k)$. Hence to evaluate the ACF we need more equation to solve for the ACF of the ARMA process.

  We multiply the standard ARMA equation $\phi(B)(z_t - \mu) = \theta(B)a_t$ by a_{t-k} and then follow the same procedure of taking expectation. Therefore we will get a relation between $\gamma_{za}(k)$ and θ_k and hence the final ACF can be evaluated.

- Three formulation of the ARMA processes

- Random Shock Form

$$(z_t - \mu) = \psi(B)a_t$$

- Difference Form

$$\phi(B)(z_t - \mu) = \theta(B)a_t$$

- Inverted Form

$$a_t = \pi(B)(z_t - \mu)$$

where $\psi(B), \pi(B)$ are random shock and inverted form parameters.

2.1.2 Non-Stationary Nonseasonal Models

As mentioned earlier if the ARMA(p,q) process is stationary, all of the roots of the characterisitic equation $\phi(B) = 0$ lie outside the unit circle. If the ARMA(p,q) process is not stationary then atleast one root of the characteristic equation $\phi(B) = 0$ must lie on the unit circle or inside the unit circle. If atleast one root lie inside the unit circle the process is said to posses *explosive non stationarity* and if one root lie on the unit circle then the process is said to posses *homogenous stationarity*.

The two basic models used for modelling Non-stationary Nonseasonal data are :

- Autoregressive Integrated Moving Average Process (ARIMA)
- Integrated Moving Average Process (IMA)

2.2 Seasonal Models

- **Seasonal Autoregressive Integrated Moving Average Models (SARIMA)**
  SARIMA models are ideal for modelling seasonal time series in which the mean and other statistics for a given season are not stationary across years. Some sort of seasonal or nonseasonal differencing is usually invoked to remove the non-stationarity.

  Once the data has been differenced, a seasonal ARMA model is fit to the resulting stationary series. Also any correlation within a season or across the seasons is assumed to be the same for all seasons, SARIMA models are a direct extension to the seasonal level of the nonseasonal ARIMA models mentioned in the previous section. SARIMA models usually require a few model parameters to describe the given time series. They are very well suited to many kinds of socioeconomic time series as well as natural time series that are significanlty affected by man-induced changes.

- **Deseasonalized Models**
  Deseasonlized models are useful for describing time series in which the mean and the variance within each season are stationary across the years. The seasonal component is removed from the series by subtracting from each observation the seasonal mean and perhaps also dividing this by the seasonal standard deviation.

  Subsequently, the most appropriate nonseasonal ARMA model is identified for fitting to the resulting the stationary nonseasonal time series. Hence here the correlation across seasons is assumed to be the same for all seasons. There have been methods suggested for reducing the large number of parameters in the deseasonalized models. These kinds of models are ideal for fitting to many kinds of natural time series for which sufficient data is available.

- **Periodic Models**
  Periodic models are ideal for fitting to a time series in which second order stationarity occurs within each season and the correlation among seasons may be different for each season. There are basically two main types of periodic models :

  - PAR (Periodic Autoregressive)
  - PARMA (Periodic autoregressive moving average)

  In PAR modelling, a separate AR model is designed for each season of the year. In PARMA modelling, a separate ARMA model is designed for each season of the year. There are a few techniques available for reducing the large number of model parameters. These models are ideal for fitting to many types of natural time series for which sufficient data is available.

Chapter 3

Model Construction

Model Constrcution as the title suggest is one of the major themes in Time Series Modelling as mentioned in [1] where one needs to construct a model which fits the time series data such that a most appropriate forecast of the future can be made. Fitting a model basically mean identifying the characteristics of the data and fitting a model such that in future that model can easily predict the happening of the phenomenon which was fit to a model. This is referred to as Time Series Modelling by [1].

As mentioned by [1], any time series modelling involves basically three phases and we will follow all the three phases to fit our data to an appropriate model.

The three major steps in model construction are :

1. Identification
2. Parameter Estimation
3. Diagnostic Checking

Once one is done with them then one can perform two operations with the time series model:

1. Forecast using the identified modeling
   Which mean one can predict the future occurances of the model/phenomenon accurately.

2. Simulation
   Which means one can simulate an artificial series which will have almost the same characteristic properties as the original data.

Here we are going to cover all the above mentioned stage for model construction and then forecast, simulate using a model which we will fit to our electric price data.

3.1 Data Series Background

The basic time series which we are going to model represents electric prices in Ontario, Canada. The electric prices tend to vary hourly at various time of the day. As they are usually low during the off peak hours and rates are comparatively high at peak hours of the day since the electricity consumption is high during this time. These price fluctuations are dependant on various factors as these electric companies would like to make more profit also they would like to manage things in a way that the power grid doesn't break down. It means to say that there ought to be balance between the consumer demand and supply. They want to kind of do some sort of load balancing such that most people only use important household appliances during the day and others at offpeak hours.

Fluctuations in these electric prices are regulated not only by electric companies but also consumer and government agencies. In the new evolving power systems where *smart grid* is playing an important role in the electricity prices where consumer comes into the scenario in regulating electricity prices. Besides government always imposes some regulations which help these electric companies to regulate the electric prices.

A very basic idea also applies in analysis of electric prices, which comes from the cost of generation i.e. cost of raw material required for generation of electricity, if it increases the cost in supposed to go up. As we all are well aware that the raw material for generation of electricity which is oil, gas, coal etc which are currently the non-renewable sources are getting more expensive since we are running out of such resources. Henceforth a direction for researchers to find a more reliable source of renewable source of energy. Based on the above background information let us infer results for our data.

First step in time series modelling is *exploratory data analysis* where we are going to look at various graphs of the data and predict a model which may most appropriately fit our time series.

3.2 Model Identification

Model Identification refers to in general terms to choose one of the models which will fit to our data. Ex. If we want to get a suit stiched we firstly identify the style, colour and material. Thus in a similar way after we know that we want to fit a model to our data we need to identify what kind of model will be best suited to our data.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Attenuates	Cuts off at lag q	Attenuates
PACF	Cuts off at lag p	Attenuates	Attenuates
IACF	Cuts off at lag p	Attenuates	Attenuates
IPACF	Attenuates	Cuts off at lag q	Attenuates

Table 3.1: Model Identification

We use graphical methods to identify what model would fit our data correctly. During the initial identification phase we need to have some criteria based on which we will be able to decide which model is better than the other. Some of these are listed below :

- Graphs
- Automatic Selection Criteria

Also an important thing to remember while constructing a model is that it should follow 2 major characteristics :

- Parsimony
- Good Statistical Fit

Now for analysis of the data using the exploratory tools we use mainly four graphs as our basic tools which help us in model identification, a table 3.1 has been given which enables us to easily identify an approximate model which would fit our data.

3.2.1 Exploratory and Confirmatory Data Analysis

The figure 3.1 below indicates the general statistical properties of the the raw data of the time series having monthly data for seven year of average electricity prices. From the graph 3.1 we can infer that the electric prices have generally been stationary when averaged daily over a month. Since 3.1 is a monthly data for seven years of electric price variations. Although we can see in the year 2004 there seems to be a shoot up of prices which may be due to various reasons. Such as the cost of resources (raw material increasing) which was controlled by government by may be subsidizing the rates which eventually brought down the rates.

Now we can also see a very interesting variation in the electricity prices in around 74th month where we can see a dip in the electric prices. From the background information about the scenario in the electric market around that year it seems that some new renewable sources of power and concepts of smart grids or some government regulation for subsidizing the electric prices could have been incorporated which led the power manufacturers regulate the prices of electricity and bringing them drastically down.

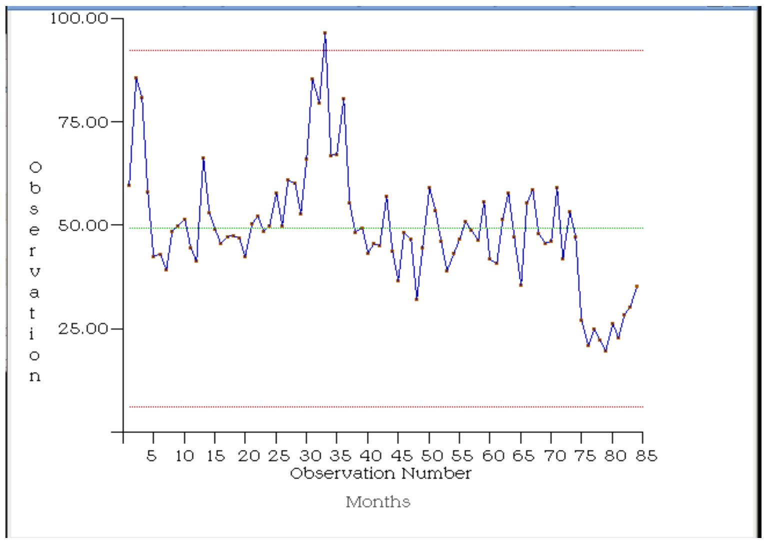

Figure 3.1: This plot is of raw data

Hence we can generally get an overview of how the data looks like by taking a macro level look at the graph of the raw data. It tells us about general things like, the trends, seasonality, interventions etc in our physical phenomenon whose data we are analyzing. We can also perform some experiments, although 3.1 doesn't show any major trends but from physical understanding of electric prices we are made to think that there has to be a low trend in the electric prices increasing. So to just see that we plotted the data with one differencing in 3.2 as mentioned earlier in model identification techniques.

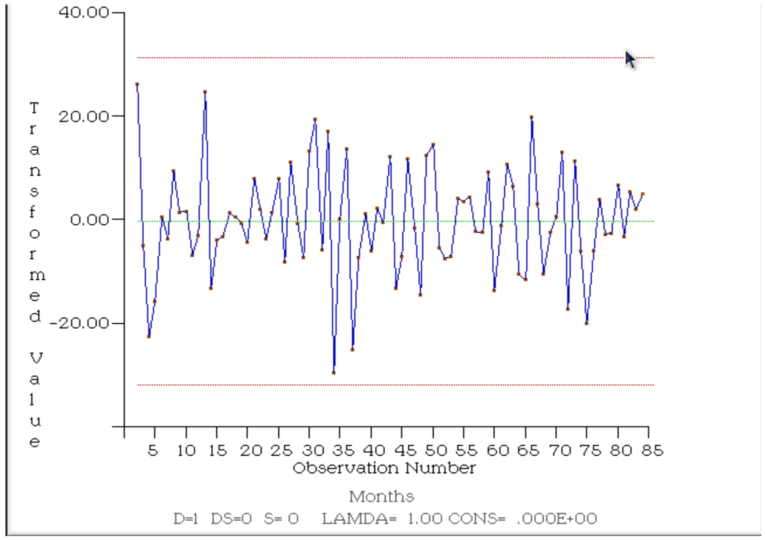

Figure 3.2: This plot is of raw data with once differencing

Statistical Properties

- Auto Correlation

Now analyzing the above two graphs closely will give us an idea about the correlation structure of our data. We can clearly see a linear dependance among the various observations. As can be evidently seen, in the months 25-40 that prices remained above the standard mean level while in the months 70-84, the prices remained substantially below the mean level. But on the other hand in the remaining periods we can see a positive value above the mean follows a positive values above it and similarly on the negative side. This means to say that there is a correlation structure in the data.Though the variations at the above two time periods is a special case, which can be due to some sort of intervention. We can infer from the the market analysis studies that there was a impulsive rise in the prices of electricity during the year 2003 which led to mean shift of electricity prices. This mean shift was eventually normalized by government regulations and subsidizing the electricity prices.

In 3.3 we can see the ACF of the raw data without any differencing being applied to it. Since 3.1 doesnt represent any major trend we do not need any differencing in our data. As mentioned in table 3.1 and visible from 3.3 our ACF dies of which gives us an indication that an AR parameter is needed. Now to more precisely decide on how many AR parameters are needed in our model, we need to look at more graphs

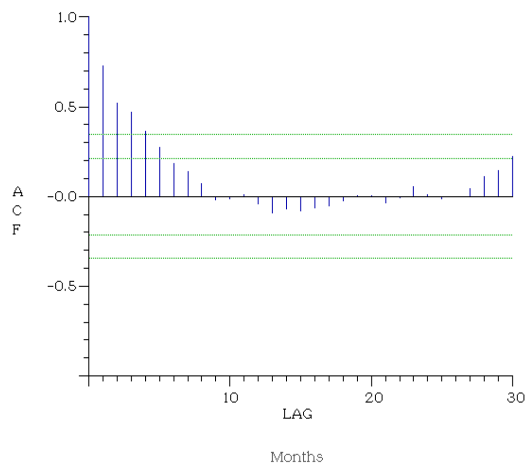

Figure 3.3: Autocorrelation Function

to ensure we make a correct decision. But from table 3.1 we can also see that the ACF also attenutates for an ARMA process, hence we might also need some MA parameters to precisely fit our data to a model.

In 3.4 we can see a theoretical plot of ACF which is very similar to the plot in 3.3. Thus it validates that we need some sort of AR parameters in our model to fit the data.

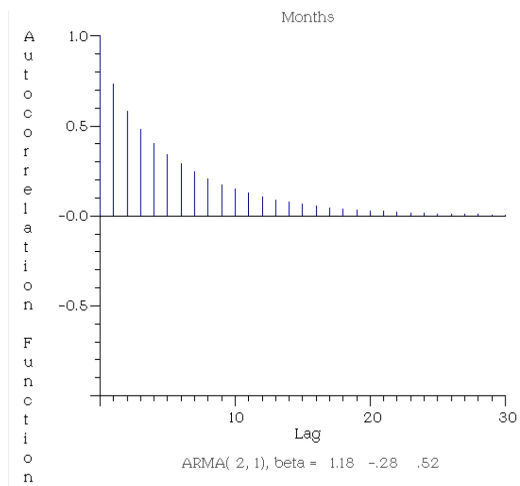

Figure 3.4: Theorectical Autocorrelation Function of the proposed model

- Partial Auto Correlation Function

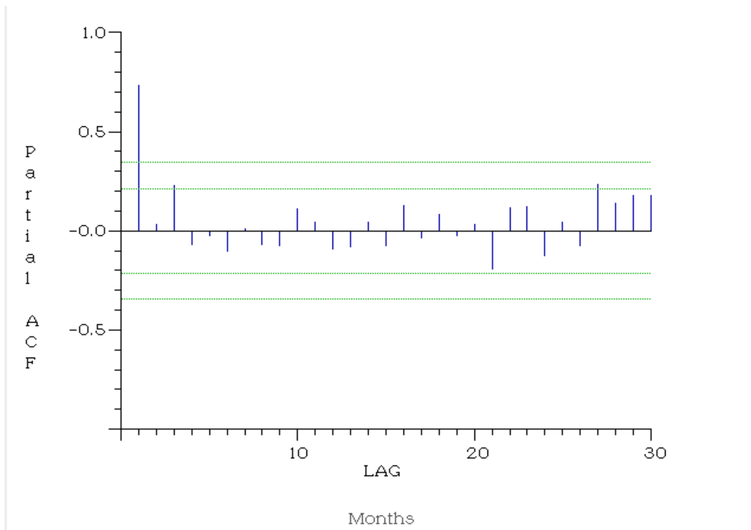

Figure 3.5: Partial Autocorrelation Function

Now this is basically second step where we need to establish firmly on how many parameters are needed to fit a accurate model to our data. As clearly mentioned in table 3.1 that PACF cuts of at lag p for an AR process while it attenutates for a MA process at lag q. It also attenuates for a ARMA process. Now looking at the plots in 3.5 we can see that our PACF for the sample data cuts of at lag 3 and with lag 2 being almost zero. Hence we can kind of think to fit 2 or 3 parameters to our model. We will analyze later to see which one gives us a more accurate statistical fit.

One can also argue that the plot 3.5 dies off and may incorporate a MA parameter as well. We can also compare our inferences with a theorectical plot of PACF for the data as seen in figure 3.6.

Sample ACF and Sample PACF are two important graphs which help us in identification of the number of AR and MA parameters to use in our model and which model to fit. Therefore we can conclude from the above graphs 3.3, 3.4, 3.5 and 3.6, that we need some sort of ARMA model to fit to our data. We still need to figure out how to consider the interventions in our model which occured around lag 24 and lag 70. This gives us an idea that we surely need a more *complex model* to fit to our data to consider such abnormalities. We will discuss later based on more exploratory and confirmatory analysis tools that a *Intervention Model* will be most appropriate to fit to our data.

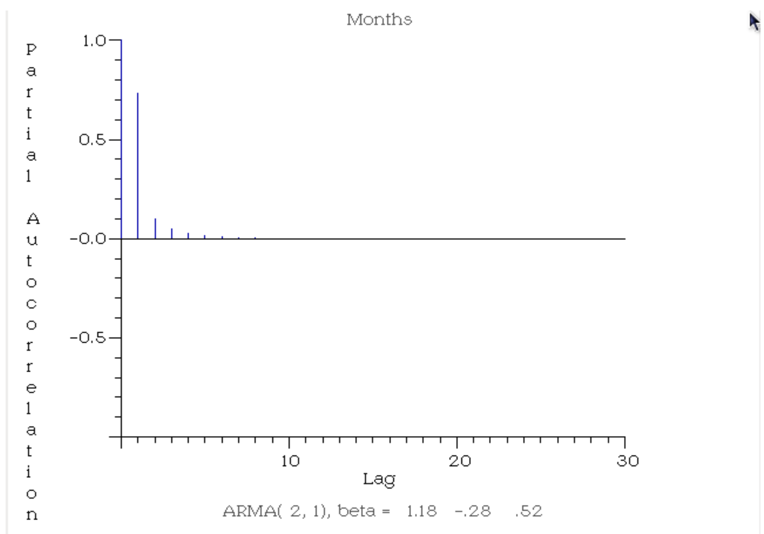

Figure 3.6: Theoretical Partial Autocorrelation Function for the proposed model

- Inverse Auto Correlation Function

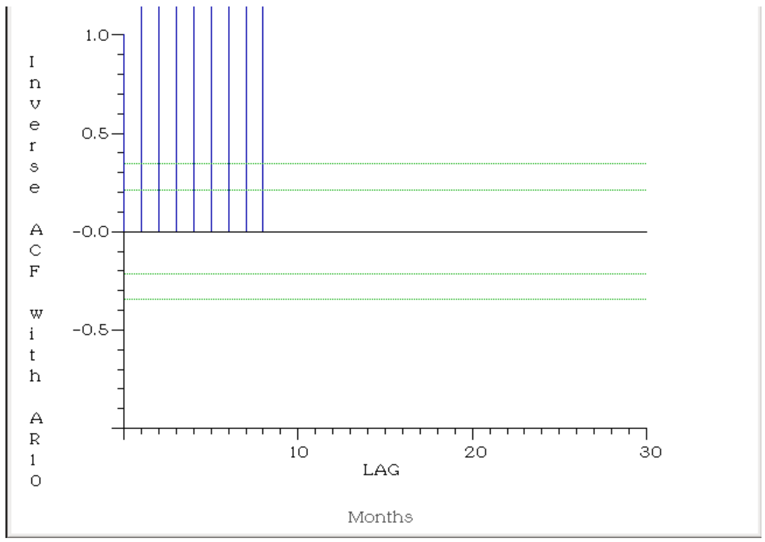

Figure 3.7: Inverse Autocorrelation Function

This is another useful tool in exploratory data analysis where we can see more precisely what kind of model will more accurately fit our data. It is clearly mentioned in table 3.1 about the features of IACF which will tell us what kind of model to fit to our data and how many parameters to use. Although some times the information obtained from ACF and PACF is more than sufficient to identify on a broad scale which model and how many parameters to use.

In figure 3.7 we can see that there are large values at lag 1,2,3,4,5. Hence not much can be inferred from the IACF in this case.

- Inverse Partial Auto Correlation Function

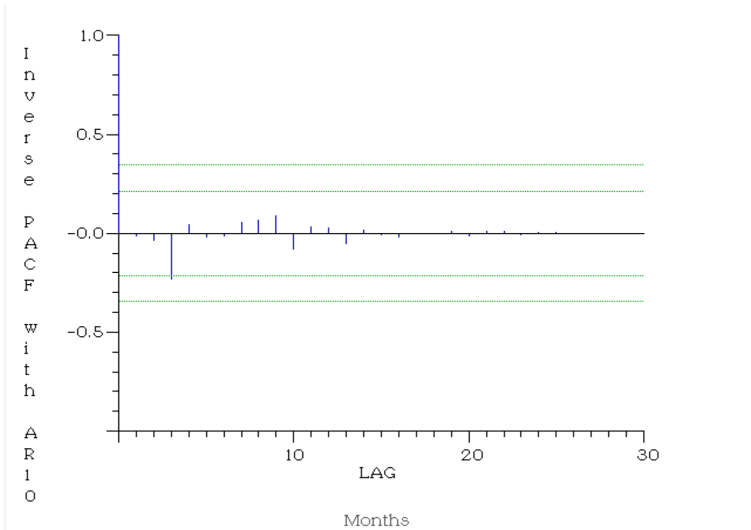

Figure 3.8: Inverse Partial Autocorrelation Function

Similar to IACF we have an inverse counterpart for PACF as IPACF (Inverse Partial Autocorrelation Function). The role of IPACF in model identification has been very clearly specified in table 3.1.

In figure 3.8 we can see that it is not significantly greater than zero at any of the initial lags but has a large value at lag 3. This can be inferred as a need for MA parameter or may be it is just a matter of chance. Usually we take into consideration the effects shown in first 20 lags. We will consider this later when we experiment while fitting our model.

- Seasonality
  It is usually evident from the background information of the data series whether it contains seasonality or not. Here in this case if we look at our data on a *micro-level* we will see that there is a seasonality in the data. Since these are electricity prices which change daily i.e. cheaper during the off-peak hours and costly during the peak hours, similary the electricity prices tend to rise in the summer due increased consumption of electricity and reduce during the winter. But in winter the consumption of the raw material increases and hence some time the prices may remain stable over the seasons but definitely a seasonality behaviour can be seen when looking at the daily behaviour.

  Here in figure 3.1 we can not see a clear seasonality since this is a *macro level* plot of the data, where it has been averaged over month and has been shown for 7 years. We can clearly see from figure 3.1 there is some sort of variance in the data which remains constant over the year. Main reason for the analysis of such a data is to enable to forecast the electricity prices given the historical trend and external interventions.

- Non-Stationarity
  We can see from figure 3.1 that our data is mostly stationary and is *homoscedastic*(Variance is constant with time). Although the mean got shifted at some lags in middle yet the variance remained constant at that mean level.

- Trends
  This is an important statistical property which one needs to look at while trying to fit a model to the data. Trend means, a changing mean level, which is not evident from the plots in figure 3.1. We have some tools which are used to perform test to analyse the presence of trends. Some of them are mentioned here

  - Mann Kendall Test - Non Parametric Test for Trend Detection
  - Hypothesis Test and Significance Tests - Statistical Tests

  In the figure 3.9 we can see that there isnt a very prominent trend, and the data seems to lie around a given mean level. There is peak and trough appearing which is basically an indication of some intervention.

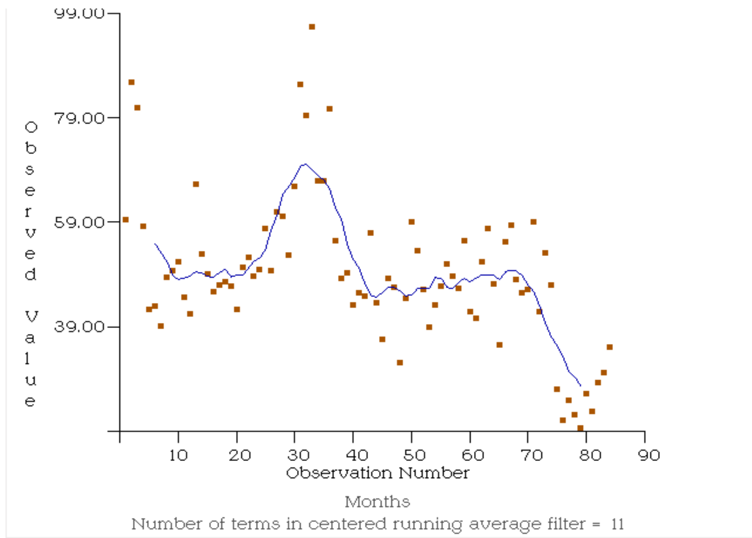

Figure 3.9: Plot of trend test

- Transformation

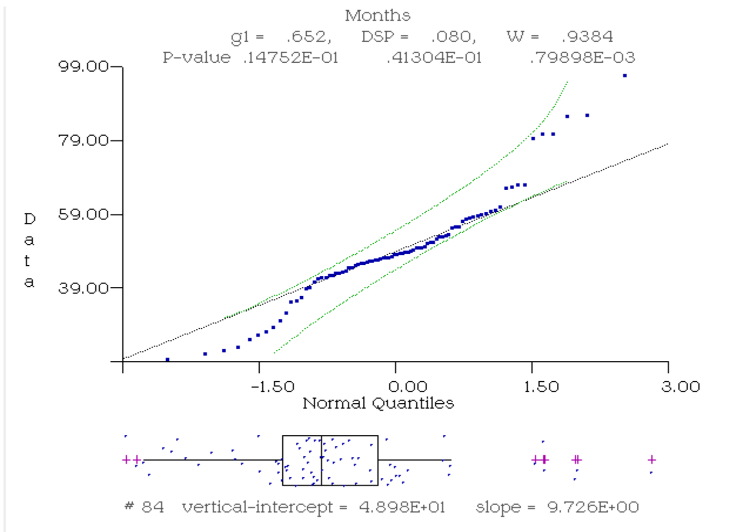

Figure 3.10: Normality

The need for a transformation can some times be identified at initial stage or after analysis of the residuals. Usually a transformation is needed when we have non-normality in the data or the data is hetero-scedastic. In figure 3.10 we can see that our data lies within the 95% confidence limits and hence we do not need any kind of transformation.

If we look at the plots 3.11, 3.12 and 3.13 plots we get an idea about a smoothing of the general trend of the data.

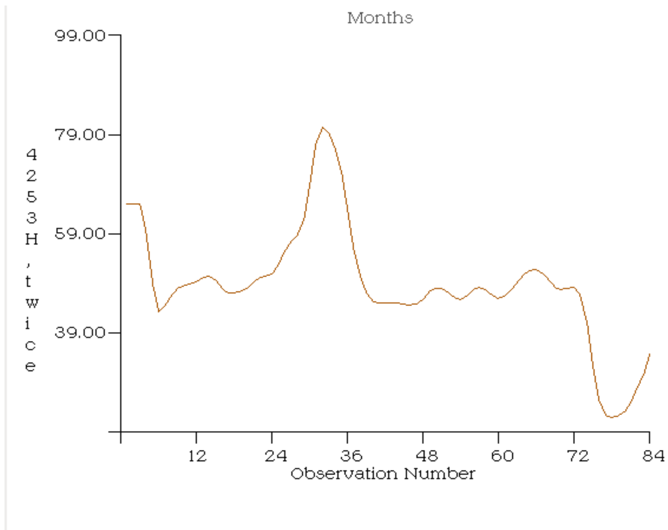

Figure 3.11: Tukey Smoothing

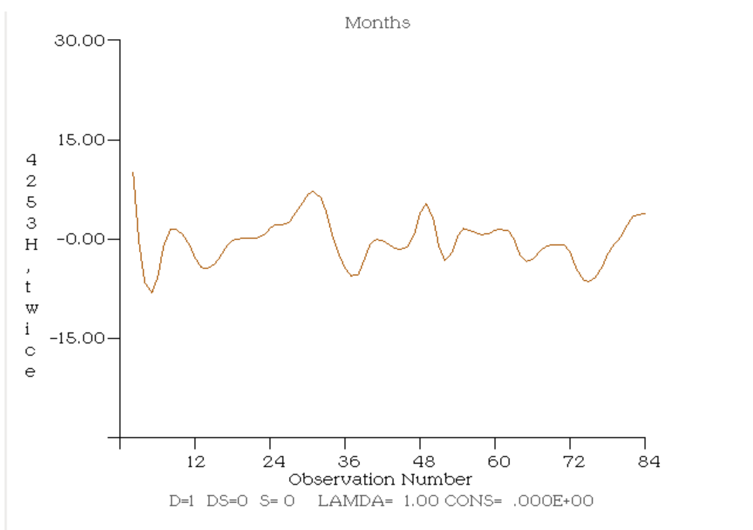

Figure 3.12: Tukey smoothing with one differencing

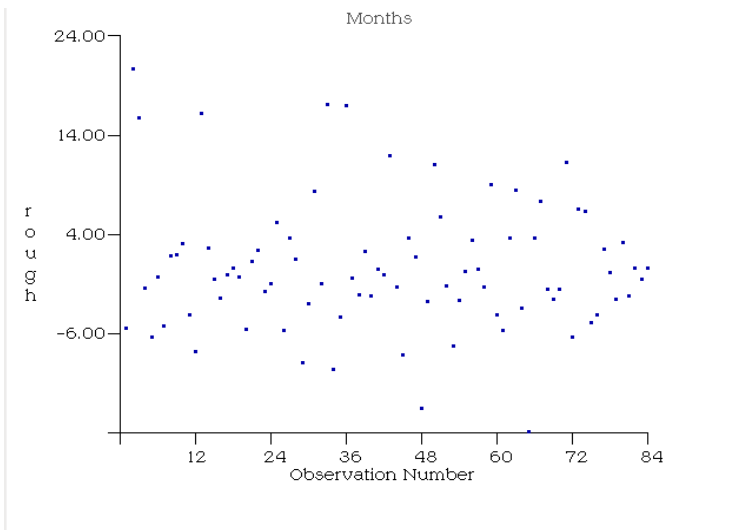

Figure 3.13: Tukey Scatter with no differencing

- Outliers/Extreme Values
  Usually we are able to identify if there are some outliers in the data and extremes from the plot of the data. In our case from figure 3.1 we do not see any such extreme conditions or outliers.

- Interventions
  This is an important feature for our data, some times there are unknown and known interventions which impact the time series. In our data in figure 3.1 we can see that there are most probably 2 interventions where we can see a change in the mean level which again comes back to normal. As the first intervention in figure 3.1 can be seen in around lag 24 and the second intervention can be seen around lag 74th.

3.3 Parameteric Estimation

Once we have identified the model which we would like to fit to our data, we need to estimate the parameters. Given below are a set of parameters which we will need to estimate to complete our model.

- Mean of the series
- AR Parameters
- MA Parameters
- Innovation Series
- Variance of the innovations

A function or statistic that defines how to calculate or estimate the value of a parameter from the data is known as an estimator. We will list out few estimators here :

- Maximum Likelihood Estimator (MLE)
- Moment - (Yule Walker Equations)
- Least Sqaures
- Kalman Filter
- Stein

In our case we will use the Maximum Likelihood Estimator to estimate our model parameters. A detailed explanation of the MLE is given in [1] and also in [4]. We need to evalutae few properties of our estimator before we can rely on the estimates.

1. Consistency of an Estimator
   A consistent estimator converges in probability as the sample size increases to the true value of the parameter or parameters being estimated. Let $\hat{\eta}$ be the estimate of a model parameter η using the sample of size n. The estimator is *consistent* if

$$Lim_{n\to\infty} P_r(\hat{\eta} - \eta > \epsilon) = 0$$

where P_r means the probability and ϵ is any positive number which can be very small.

2. Efficiency
   Let $\hat{\eta_1}$ adn $\hat{\eta_2}$ be two consistent estimators or estimates for a model parameter where the sample size is n. The Asymptotic Relative Efficiency(ARE) of $\hat{\eta_1}$ with respect to $\hat{\eta_2}$ is

$$ARE = Lim_{n\rightarrow}\frac{var\hat{\eta_1}}{var\hat{\eta_2}}$$

The variance of the MLE of a model parameter has minimum asymptotic variance and is asymptotically normally distributed when consistency and other conditions are satisfied. There are various kinds of MLE avaible, Approximate MLEs and Exact MLEs and then we also have algorithms which are used for the optmization of the likelihood function. They have been described in detail in [4].

Now once we have estimated the parameters of our model using MLEs we need to discriminate between the models. We used various criterion to discriminate between the models . Most commonly used is *Akaike Information Criteria (AIC)*

$$AIC = -2lnML + 2k$$

where the ML = Maximum Likelihood, lnML = value of the maximized log likelihood function for a fitted model, k = number of parameters.

AIC reflects two major components :

- Model Parsimony due to 2k terms
- Good Statistical fit due to -2lnML term

We always select the model with the least AIC and this is referred to as Minimum AIC or MAIC.

Model	AIC	Special Characters
SARIMA(2,1,2)	393.90	
SARIMA(2,0,2)	389.77	
SARIMA(2,2,2)	412.35	
ARIMA(2,1,3)	459.56	
ARIMA(3,0,2)	387.87	
ARIMA(1,0,2)	388.90	
ARIMA(1,0,1)	389.85	
ARIMA(1,0,3)	388.92	
ARIMA(2,0,0)	389.99	
ARIMA(0,0,3)	396.32	
ARIMA(2,0,3)	390.35	
PAR Model	34 Parameters, AIC = 521	Assuming 12 periods
Intervention TFN Model	387.7	ARIMA(2,0,1), Intervention 1 : u=2,v=2 at lag 24
		Intervention 2 : u=2,v=2 at lag 74
Intervention TFN Model	392.78	ARIMA(2,0,1), Intervention 1 : u=2,v=1 at lag 24
		Intervention 2 : u=2,v=1 at lag 74
Intervention TFN Model	388.23	ARIMA(2,0,2), Intervention 1 : u=2,v=2 at lag 24
		Intervention 2 : u=2,v=2 at lag 74
Intervention TFN Model	**386.236**	**ARIMA(2,0,1), Intervention 1 : u=1,v=1 at lag 24**
		Intervention 2 : u=1,v=1 at lag 74
Intervention TFN Model	387.23	ARIMA(3,0,1), Intervention 1 : u=1,v=1 at lag 24
		Intervention 2 : u=1,v=1 at lag 74

Table 3.2: Various models fitted to the data and their AICs

Model Fitting
We fitted numerous models to our data and discriminated between them based on the *AIC* value :

Now from the table 3.2 we can see that after exploratory data analysis our job of fitting a more accurate model becomes much easy. As we are partly sure of what parameters to use. In our case if we look at table 3.2 we can see that an *intervention transfer function model* with 2 interventions and u = 1 and v = 1 for both the interventions. We then fit a ARMA(2,1) to the residuals and the final model which we obtain has the least AIC value. Therefore based on our discussion above we are able to select one appropriate model.

In figure 3.14 we can see a plot of the residual of the the series. This is the N_t in the general intervention model. Here we have presented a detailed estimation of all the models mentioned in the table 3.2.

3.4 Diagnostic Checking

Once we have fitted a model to our data and discriminated among all of the other based on various criterion as AIC, BIC etc. Then we need to do some checking on the model thus fitted. In our case we need to check the RACF (Residual Auto Correlation Function).

In figure 3.14, one can see the plot of the residuals of the fitted model. We can also see very cleary from the ACF of the residuals in the plot 3.15 that it lies within the 90% confidence limits and hence we can accept the model which we have fit.

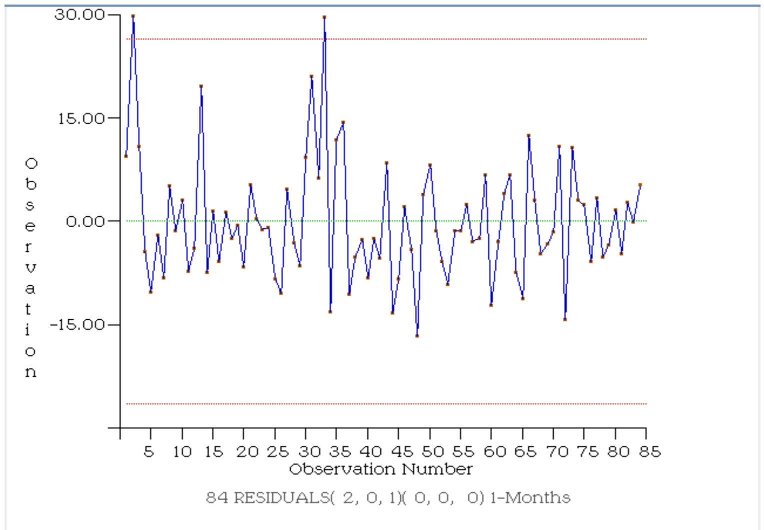

Figure 3.14: Residuals

Beside the residuals and its more comprehensive plots, we can also look at other tools used for diagnostic checking as Cumulative periodogram 3.16which is also lying within the 90% confidence limits which ensures that we have got a good fit for our data.

There are a couple of other graphs which may be useful for diagnostic checking are shown here in figures 3.17 and 3.18. Although the most important check is the analysis of the residuals of the MLEs.

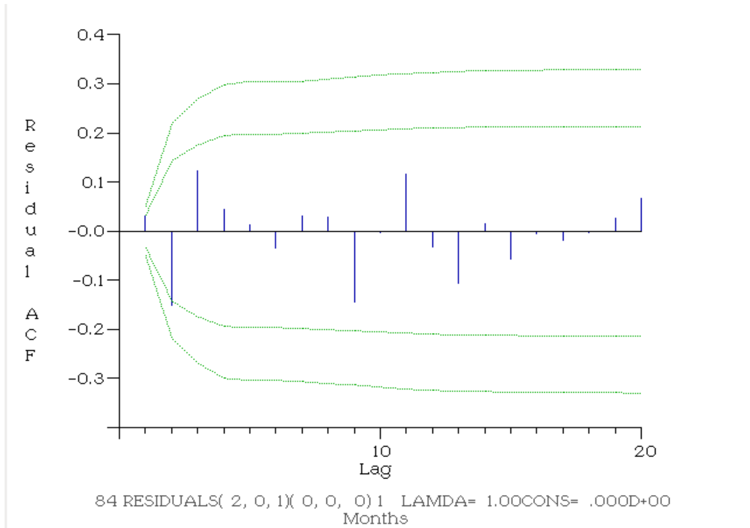

Figure 3.15: Residuals ACF

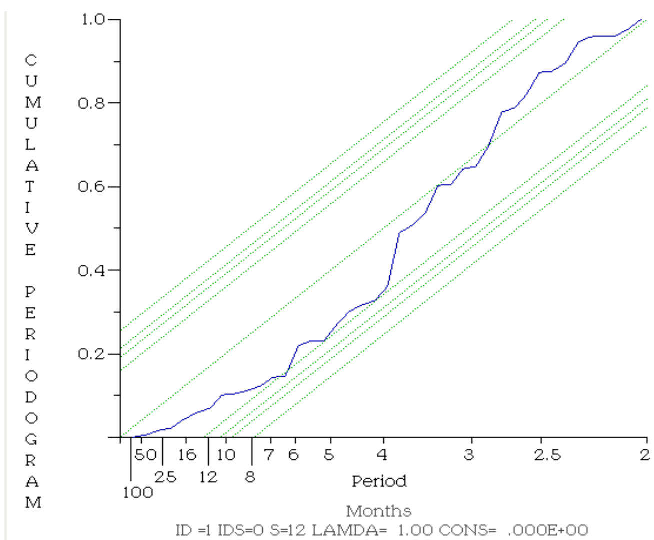

Figure 3.16: Cumulative Periodogram

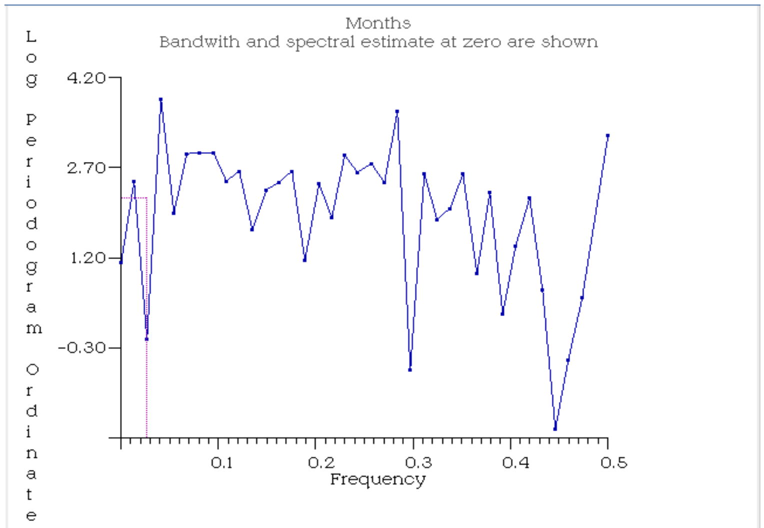

Figure 3.17: Frequency Periodogram

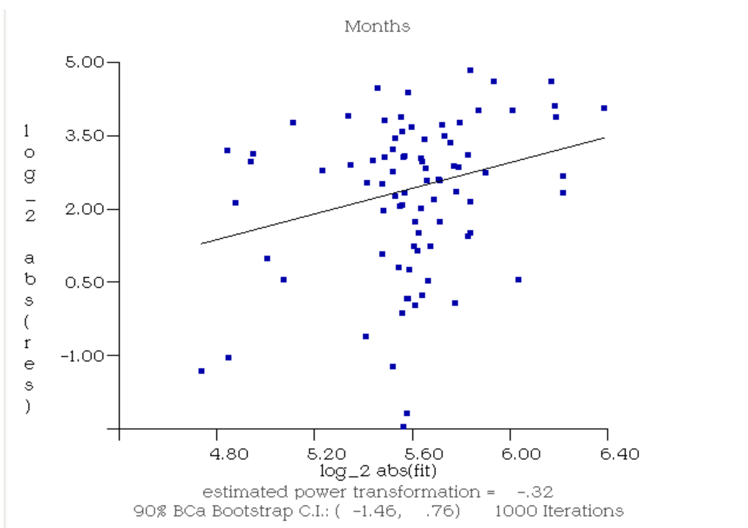

Figure 3.18: Power Transformation

3.5 Intervention Analysis

As per the exploratory data analysis we came to a conclusion that our data undergone 2 major interventions one at around lag 24 and the other one at lag 76. On figuring out the reason behind such interventions, interesting results revealed.

Although a lot of seasonality of the data was loss due to averaging out the hourly data to monthly data over seven years. Hence from the above analysis and following the three stages of model construction from identification, parameter estimation and diagnostic checking, we find that ARMA(2,1) fitted to a 2 intervention based TFN(Transfer Function Noise) model with 1 parameter in the numerator and 1 parameter in the denominator for both the interventions is used.

The general form of 2 intervention model would be like this :

$$(y_t - \mu_y) = \sum_{t=1}^{2} V_i(B) + N_t$$

where N_t is the Noise term with a ARMA model

$$V_i(B) = \frac{\omega(B)}{\delta(B)} = \frac{\omega_{0i} - \omega_{1i}}{(1 - \delta_{1i}B)}$$

The parameter for the above mentioned model have been identified using the *McLeod Hipel Time Series Decision Support System* [4].

3.6 Forecasting/Simulation and Results

Once we have fitted a model to our data and performed diagnostic checks on it now we are ready to use the model for forecasting and simulation which was the basic purpose of modelling so that we will be able to predict the values in the future based on the past. Assuming that the model we design preserves the historical data.

In the figure 3.19 we can cleary see that our model was used to forecast values 10 time steps ahead and we can see that the forecasts lie within th 95% confidence limits. As mentioned in [4] we used the *Minimum Mean Square Error Forecasts* to predict the future values of our time series.

Secondly, another application of a fit model is to simulate artificial series based on the developed model. Figure 3.20 shows how the model was used to simulate another series.

We also plotted the Sample Cross Correlation function between the series and see that there is an instantaneous Causality between the input and the output series which seems correct in our case. 3.21 shows the CCF which clearly shows instantaneous correlation betweent the two series.

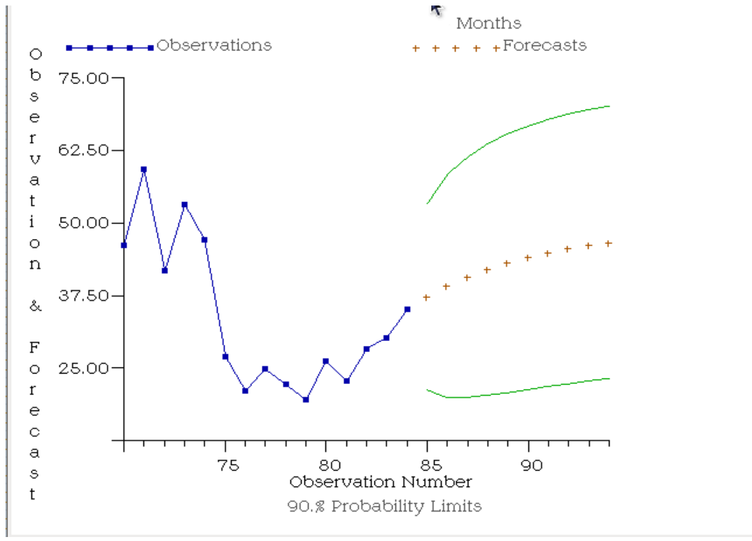

Figure 3.19: Forecast using the Intervention Model

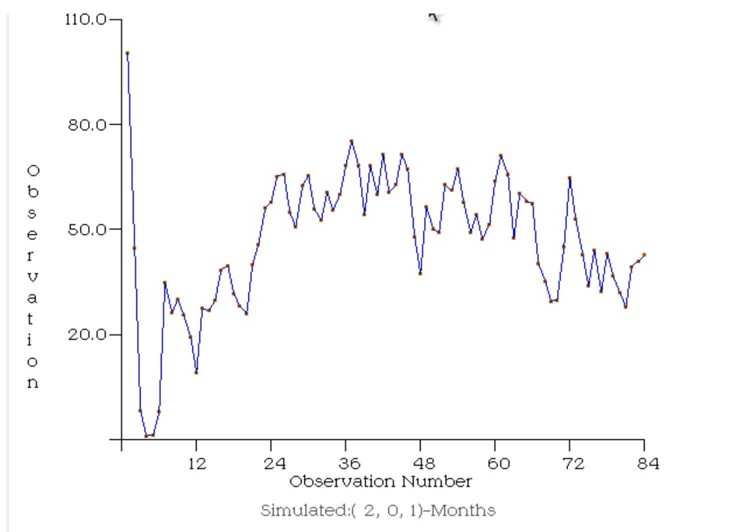

Figure 3.20: Simulation using the Intervention Model

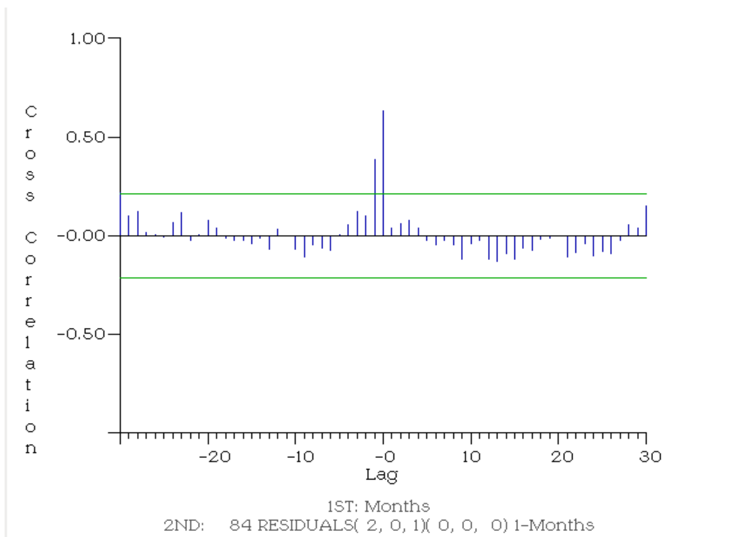

Figure 3.21: Sample Cross Correlation Function

Forecasting

ARIMA(2, 0, 1)

Trend Coefficient Residual Variance

5.041005D+00 9.479612D+01

Generalized Autogressive Coefficients

1: 1.180, 2: -.282,

Moving Average Term

1: .524,

Coefficients in the Moving Average Expansion

1: .656, 2: .491, 3: .395, 4: .327, 5: .275, 6: .232,
7: .196, 8: .166, 9: .140, 10: .119,

Pivotal Values for Forecasting

Forecasting Origin = 84

Observations

Z(81)= 2.289D+01, Z(82)= 2.825D+01, Z(83)= 3.017D+01, Z(84)= 3.519D+01

Disturbances

A(81)=-7.814D+00, A(82)=-4.867D-01, A(83)=-1.997D+00, A(84)= 1.478D+00

Lead Time	MMSE Forecast	90% Probability Interval
1)	37.2719D+00	(21.2571D+00, 53.2867D+00)
2)	39.0868D+00	(19.9367D+00, 58.2369D+00)
3)	40.6407D+00	(19.9362D+00, 61.3452D+00)
4)	41.9620D+00	(20.3131D+00, 63.6108D+00)
5)	43.0825D+00	(20.8083D+00, 65.3568D+00)
6)	44.0319D+00	(21.3272D+00, 66.7365D+00)
7)	44.8358D+00	(21.8297D+00, 67.8419D+00)
8)	45.5164D+00	(22.2972D+00, 68.7356D+00)
9)	46.0927D+00	(22.7221D+00, 69.4633D+00)
10)	46.5806D+00	(23.1022D+00, 70.0590D+00)

Chapter 4

Neuro-Fuzzy model (ASuPFuNIS) for Time Series Prediction

4.1 Inroduction

Numerous examples of synergistic fuzzy neural models that combine the merits of connectionist and fuzzy approaches have been proposed in the literature [11, 19, 2]. These include the fuzzy multilayer perceptron [18]; neural fuzzy systems [6]; and evolvable neuro-fuzzy systems [9]. By virtue of their ability to refine initial domain knowledge, operate and adapt in both numeric as well as linguistic environments, these models are extensively used in a wide range of application domains such as approximate reasoning and inferencing [7]; classification [16]; diagnosis [17]; control [8]; rule extraction and simplification [3].

Most hybrid models embed data-driven or expert derived knowledge, in the form of fuzzy *if-then* rules, into a network architecture to facilitate fast learning [20]. This embedding of knowledge is often done by assuming that antecedent and consequent labels of standard fuzzy *if-then* rules are represented as connection weights of the network as in [5]. The composition of the incoming numeric information with this embedded knowledge is usually done by computing membership values from fuzzy membership functions that represent network weights [16] or by fuzzifying the numeric inputs by modeling them as fuzzy numbers using triangular or Gaussian membership functions and then using the well defined sup-star or mutual subsethood composition mechanisms [20, 13]. In case of symbolic information, a given universe of discourse is generally quantized into pre-specified fuzzy sets. A fuzzy input is then simply one of these pre-specified fuzzy sets [18, 14]. Subsequently a learning algorithm fine tunes these rules based on the available training data that describes the problem. Commonly used learning algorithms employ either supervised gradient descent and their variants [18], unsupervised learning, reinforcement learning [15], or genetic algorithm based search [21, 22, 8].

Asymmetric Subsethood-Product Fuzzy Neural Inference System (ASuPFuNIS) directly extends SuPFuNIS [20] by permitting the signal and weight fuzzy sets to be asymmetric in the sense that the left and right spreads of the signal and weight fuzzy sets can differ. Both signal and weight fuzzy sets are thus defined by three parameters: a center, a left spread and a right spread. This extension is motivated by the fact that asymmetric fuzzy sets lend flexibility and thereby may be fruitful in capturing the non-uniformity of data in complex problems.

4.1.1 Methodology

The ASuPFuNIS network is trained by supervised learning, once again in a way similar to that of SuPFuNIS [20]. This involves repeated presentation of input patterns drawn from the training set and comparing the output of the network with the desired value to obtain the error. Network weights are changed on the basis of an error minimizing criterion. Once the network is trained to the desired level of error, it is tested by presenting unseen test set patterns.

Learning is incorporated into ASuPFuNIS using the standard iterative pattern based gradient descent method. The instantaneous squared error $e(t)$ at iteration t is used as a training performance parameter and is computed in the standard way:

$$e(t) = \frac{1}{2} \sum_{k=1}^{p} \left(d_k(t) - S(y_k(t))\right)^2 \tag{4.1}$$

where $d_k(t)$ is the desired value at output node k, and $e(t)$ is evaluated over all p outputs for a specific input pattern $X(t)$. Notice that for an n-q-p architecture of ASuPFuNIS, the number of connections is $(nq + qp)$. Since in the proposed model, the representation of a fuzzy weight requires three parameters (center, left and right spreads) and an input feature requires two parameters (left and right spreads), the total number of free parameters to be trained will be $3(nq + qp) + 2n$. If the trainable parameters of ASuPFuNIS be represented as a vector $P = (x_i^{\sigma^l}, x_i^{\sigma^r}, w_{ij}^c, w_{ij}^{\sigma^l}, w_{ij}^{\sigma^r}, v_{jk}^c, v_{jk}^{\sigma^l}, v_{jk}^{\sigma^r})^T$, then the iterative gradient descent update equation can be written as

$$P(t+1) = P(t) - \eta \nabla e(t) + \alpha \Delta P(t-1) \tag{4.2}$$

where η is the learning rate, $\nabla e(t) = \left(\frac{\partial e(t)}{\partial x_i^{\sigma^l}}, \frac{\partial e(t)}{\partial x_i^{\sigma^r}}, \frac{\partial e(t)}{\partial w_{ij}^c}, \frac{\partial e(t)}{\partial w_{ij}^{\sigma^l}}, \frac{\partial e(t)}{\partial w_{ij}^{\sigma^r}}, \frac{\partial e(t)}{\partial v_{jk}^c}, \frac{\partial e(t)}{\partial v_{jk}^{\sigma^l}}, \frac{\partial e(t)}{\partial v_{jk}^{\sigma^r}} \right)^T$ is the gradient vector, α is the momentum parameter and $\Delta P(t-1) = P(t) - P(t-1)$.

4.2 Implementation

An indepth explanation of the above algorithm can be seen in [10]. In [10] we can see that the ASuPFuniS model has been used in the prediction of one of the most widely studied time series called the Mackey Glass Time Series. It is found that this neuro-fuzzy model is able to made forecasts l time lags ahead. Here in this report we will compare the result of the forecasting of a time series model fit to the electric price data and make MMSE (Minimum Mean Square Error) forecasts and compare it with the one step ahead forecast made by the neuro-fuzzy model.

The electric price monthly data for a period of seven year had 84 values in the time series. From the point of view of the neuro-fuzzy model the the data was divided into two halves of 42 values each. Each of them was referred to as training and test data sets. Now initially the training data was presented to the network for 10000 epochs. Once the training phase was completed the the test data was presented to the network and based on the test data the network was made to forecast the one step ahead forecast.

An interesting thing about using the one step ahead forecast is that they conserve a lot of information. One step ahead forecast are the one which have least amount of uncertainty and posses most information. If we can make the one step ahead forecast accurately then further forecasts can be made in the same way.

As mentioned in the table 4.1 we ran initially two sets of experiments with the above mentioned set of parameters for 10000 epochs. It was observed that the performance of the network was better when the learning rate was high with a high momentum. This is a particular case, one can iteratively run using various combinations and variations of these learning rates and momentums to get more results for analysis.

One more variation which one could experiment with such a set up is by changing the number of rules to operate. For this problem we considers a set of 5 rules to operate which for a higher precision one can always increase the number of rules and perform experiment with variations in learning rate and momentum. Various combinations of these parameter and how they can be implemented can be easily seen in in [10].

Particulary for this prediction problem, ASuPFuNIS employed a *4-q-1* network architecture : the input layer comprises of four numeric nodes; the output layer comprises a single output node and there are q rule nodes in the hidden layer. Thus, the number of trainable parameter will be $15q + 8$.

Before training the network, the centers of the antecedents and the consequents fuzzy sets were initialized in the range (0,1.5). Both the feature and fuzzy weight spreads were intitalized in the range (0.2, 0.9). The learning rate and momentum values were experiment with were the extremes, 0.2 and 0.02, 0.1 and 0.7 respectively. After the completion of the training phase, the test set was presented to the trained network. As every algorithm

Learning Rate	**0.2**	**0.02**
Momentum	0.1	0.1
Initial Min	Min normalized value of the data	Min normalized value of the data
Initial Max	Max normalized value of the data	Max normalized value of the data
No. of Rules	5	5

Table 4.1: Experiment Set 1

(Learning Rate, Momentum)	(0.2,0.1)	(0.2,0.7)	(0.02,0.1)	(0.02,0.7)
One step ahead Forecast	38	44	49	47

Table 4.2: Experimental Details

has to have a performance criteria, in ASuPFuNIS, the normalized root mean square error (NRMSE) is considered as the performance measure. NRMSE is defined as the ratio of the RMSE divided by the total number of patterns presented to the network.

4.3 Forecasting and Results

The ASuPFuniS was implemented using two sets of parameters. One where we kept the learning rate high and one with a low learning rate. This gave us an idea of how fast a neural network based architecture is able to learn the pattern and make accurate forecasts.

The table 4.2 show the one step ahead forecast for the experiment.

Chapter 5

Comparison and Conclusion

On experimentation by fitting models of time series to some real data, as in our case the electricity prices over the past 7 years, involved rigorous analysis. We need to follow the three key steps involved in model construction (Model Identification, Parameter Estimation and Diagnostic Checking). Various exploratory analysis techniques where one needs to look closely at various graphs and plot of the data play a key role in accurate identification of the model. Some important graphs which were helpful in analyzing the data were Autocorrelation Function, Partial Autocorrelation Function, Normality, Tukey Smoothing; also an important graph is the plot of the raw data at every instant, since it gives us a clear indication of the statistical characteristics of the data.

Once the model identification phase is completed, one moves on to parameter estimation, where we try to estimate the model parameters using the Maximum Likelihood Estimates. We performed numerious experiments using the *MHTS(McLeod Hipel Time Series) Package* [4]. Based on the exploratory data analysis, we tried to fit in a most closest model. The parameter estimates we evaluated. Also once we evaluate parameters for numerous models we need to discriminate among them. We used a selection criterion called the *Akaike Information Criterion* for final selection of the most appropriate model.

Once we have obtained our model we need to do diagnostic checking by evaluating the Residual AutoCorrelation function, if the RACF lies within the 90% limits we assume the model to be very precise and accurate and we can then use it for forecasting and simulation.

Comparison between Time Series Forecast and ASuPFuNIS Time Series Prediction

	TSM	ASuPFuNIS	%age Deviation TS	%age Deviation ASuPFuNIS
Model Parameters			N/A	
Performance Criteria			N/A	
	MMSE	Forecast (LearningRate, Momentum)	Actual Value = 40	Actual Value = 40
One Step Ahead Forecasts	*39*	*39 (0.2,0.1)*	*2.5%*	*2.5%*
	39	44 (0.2,0.7)	10%	2.5%
	39	49 (0.02,0.1)	22%	2.5%
	39	47 (0.02,0.7)	17%	2.5%

Table 5.1: Comparison of Time Series Forecast and ASuPFuNIS Forecasts

In table 5.1 we can clearly see a similarity between the results obtained using a a time series model and the forecast using the ASuPFuNIS based Neuro-Fuzzy model. Although both these models have their advantages and disadvantages. In the time series model it has flexibility to choose more complex models with more number of parameters and more accurately approximate the phenomenon. On the other hand ASuPFuNIS gives us flexibility in terms of tuning only couple of parameters like the learning rate, momentum, number of rules etc to help the network train with the given data. Although the neuro-fuzzy model takes some time during the training phase but the impressive feature of such a model is that it automatically adapts the weights of the network dynamically based on the data. Although from table 5.1 we can see that the results from Time Series Models are fixed while we have flexibility to tune in the parameters still in neuro-fuzzy predictor by changing the learning rate and momentum, and also by varying the number of epochs for the training set. Hence it means that based on our problem as we would follow a three stage procedure to identify a model for Time Series Based technique similary we can develop techniques to choose parameters for our ASuPFuNIS or any Neuro-Fuzzy model based on some pre-analysis of the data.

5.1 Future Work

We would like to explore more techniques used in amalgamation with Time Series Models and Neuro-Fuzzy based model for time series model fitting and forecasting. We would like to explore fuzzy time series models as described by [12] and [23] since fuzzy sets are a more realistic approach to model real world.

Bibliography

[1] George Box, Gwilym M. Jenkins, and Gregory Reinsel. *Time Series Analysis - Forecasting and Control*. Prentice Hall, 1994. 3, 5, 6, 7, 12, 29

[2] J. Buckley and T. Feuring. *Fuzzy and Neural: Interactions and applications*. Studies in Fuzziness and Soft Computing. Physica-Verlag, Heidelberg, Germany, 1999. 39

[3] D. Chakraborty and N. R. Pal. Integrated feature analysis and fuzzy rule-based system identification in a neuro-fuzzy paradigm. *IEEE Trans. Syst. Man, Cybern.*, 31:391–400, 2001. 39

[4] Keith. W. Hipel and A. Ian McLeod. *Time Series Modelling of Water Resources and Environmental Systems*. Elsevier, Amsterdam, The Netherland, 1994. iii, 2, 3, 5, 6, 7, 29, 30, 35, 43

[5] H. Ishibuchi. Neural network that learn from fuzzy if then rules. *IEEE Trans. Fuzzy Syst.*, 1:85–97, May 1993. 39

[6] H. Ishibuchi and Y. Hayashi. A learning algorithm of fuzzy neural networks with triangular fuzzy weights. *Fuzzy Sets Syst.*, 71:277–293, 1995. 39

[7] J. S. R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man, Cybern.*, 23:665–685, May 1993. 39

[8] C. F. Juang. A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. *IEEE Trans. Fuzzy Syst.*, 10:155–170, April 2002. 39, 40

[9] Nikola Kasabov. *Neuro-Fuzzy Techniques for Intelligent Information Processing*, volume 30 of *Studies in Fuzziness and Soft Computing*. Physica Verlag, Heidelberg, Germany, 1999. 39

[10] Satish Kumar and C. Shunmuga Velayutham. *Asymmetric Subsethood-Product Fuzzy Neural Inference System (ASuPFuNIS)*. IEEE Transactions on Neural Networks, 2005. iii, 41

[11] C. T. Lin and C. S. G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice-Hall, Upper Saddle River, NJ, 1996. 39

[12] Hao Tien Liu. *An integrated fuzzy time series forecasting system*. Expert Systems with Applications: An International Journal, 2009. 44

[13] Jerry M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, Upper Saddle River, NJ, 2001. 39

[14] S. Mitra and S. Pal. Fuzzy multi-layer perceptron, inferencing and rule generation. *IEEE Trans. Neural Networks*, 6:51–63, January 1995. 39

[15] S. Mitra and S. Pal. Fuzzy self organization, inferencing and rule generation. *IEEE Trans. Syst. Man, Cybern.*, 26:608–620, 1996. 39

[16] D. Nauck and R. Kruse. A neuro-fuzzy method to learn fuzzy classification rules from data. *Fuzzy Sets Syst.*, 89:277–288, 1997. 39

[17] D. Nauck and R. Kruse. Obtaining interpretable fuzzy classification rules from data. *Artificial Intell. Med.*, 16(2):149–169, 1999. 39

[18] S. K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets and classification. *IEEE Trans. Neural Networks*, 3:683–697, September 1992. 39

[19] S.K. Pal and S. Mitra. *Neuro-fuzzy Pattern Recognition: Methods in Soft Computing*. Wiley, New York, 1999. 39

[20] S. Paul and S. Kumar. Subsethood product fuzzy neural inference system (SuPFuNIS). *IEEE Trans. Neural Networks*, 13(3):578–599, May 2002. 39, 40

[21] M. Russo. FuGeNeSys—A fuzzy genetic neural system for fuzzy modeling. *IEEE Trans. Fuzzy Syst.*, 6:373–388, August 1998. 40

[22] M. Russo. Genetic fuzzy learning. *IEEE Trans. Evol. Comput.*, 4:259–273, September 2000. 40

[23] Hia Jong Teoh, Tai-Liang Chen, and Ching-Hsue Cheng. *Frequency-Weighted Fuzzy Time-Series Based on Fibonacci Sequence for TAIEX Forecasting*. Lecture Notes in Computer Science, Springer-Verlag, 2007. 44