1-1-2015

Developing An Automated Forecasting Framework For Predicting Operation Room Block Time

Azad Sadr Haghighi
Wayne State University,

DEVELOPING AN AUTOMATED FORECASTING FRAMEWORK FOR PREDICTING OPERATION ROOM BLOCK TIMES

by

AZAD SADR HAGHIGHI

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2015

MAJOR: INDUSTRIAL ENGINEERING

Approved By:

_____

Advisor Date

DEDICATION

. . . Dedicated to my parents

In love and gratitude

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Background

Operating rooms are biggest source of the revenue as well as the largest cost center in every hospital, so naturally the surgical case scheduling process is the key concept in the functioning of the Operating Rooms (Denton, Viapiano, & Vogl, 2007). In most hospitals with standard OR scheduling process, total allocated capacity of the ORs are distributed into the set of fixed blocks. Every block is a time interval in single or multiple OR that is assigned to a specific surgeon (surgeon block), group of surgeons (group blocks) or surgical service area (service blocks) under the master schedule, typically on a weekly basis. Unlike to open booking system, it is more efficient, but its effectiveness is dependent upon whether the scheduled block accurately reflects the actual patterns of usage and whether mechanisms are in place to release unreserved blocks in a timely manner (Milewski). So from the system level perspective, case scheduling process is very critical issue in hospitals. In large medical centers with greater number of the operating room and surgical demands, the scheduling process becomes more complicated and difficult to manage. Employing block scheduling approach can results in the significant improvement in scheduling process that ultimately lead to patient, surgeon and staff satisfaction and cost reduction.

1.2 Research Motivations

Surgical case demand fluctuate widely day by day due to the many involved factors. The scheduling of the surgeries becomes a very challenging and critical issue in the hospitals, since overestimating lead to wasting resources and on the other hand underestimation causes the overtime staffing and maybe surgery cancellations (Denton,

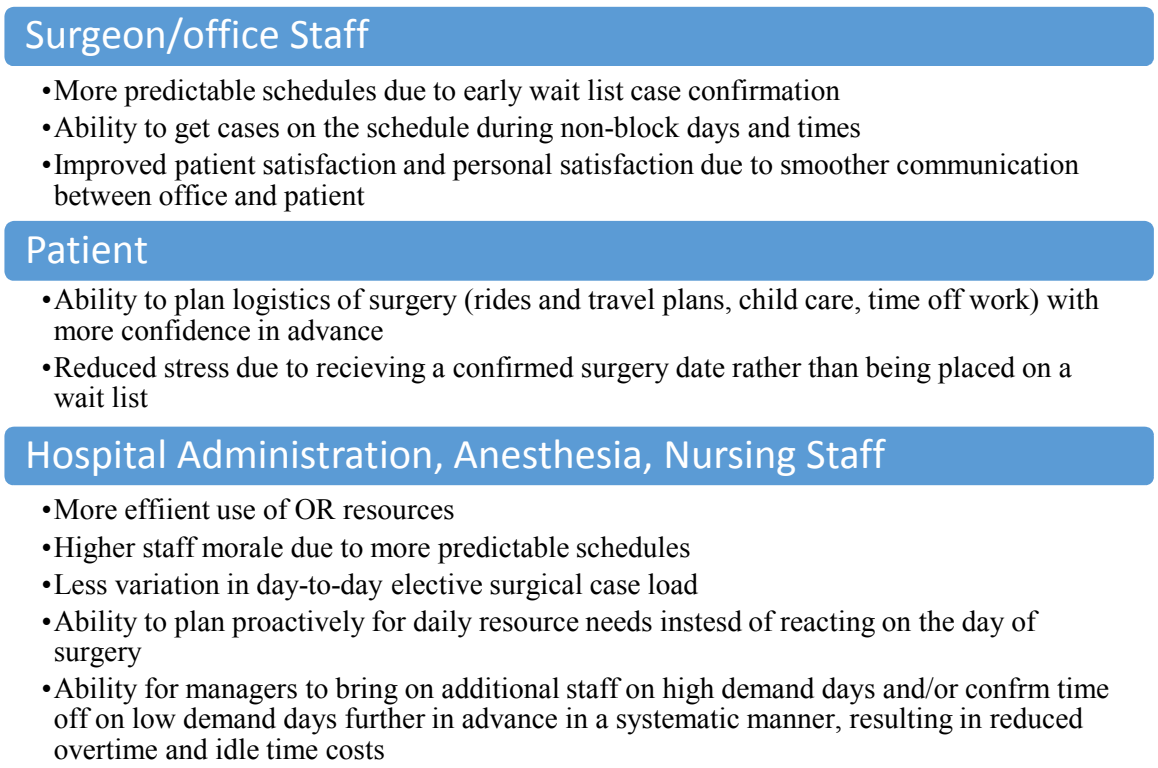
Viapiano, & Vogl, 2007). As a result, to solve this problem we need to estimate the case load of the surgeries in future in an accurate way. Unfortunately, unlike the many other industries like manufacturing, transportation and …, which by using different data analysis and decision support systems try to improve their industry and outpace their competition, most of the medical centers have not equipped with the technologies to reduce the uncertainty in surgery demands. Instead, the scheduling system of the ORs has not been changed over the past 30 years and typically it's performed by the human intervention, intuition and experience rather than the systematic framework and decision support system. Large uncertainty and Wide fluctuation in daily surgical demand limits opportunities for dynamic resource allocation and adaptive staff planning, preventing optimal resource utilization and obstructing desired financial results. Over-allocation and under-allocation of the block times lead to the significant amounts of unused OR time or overtime staff scheduling. In both situations there are too much loss of the resource.

OR scheduling also faces some challenges which you cannot poses in other scheduling problems. In addition to restrictions related to block time allocation, each case must be scheduled in a way that all required resources are available during the duration of the case. At minimum, these resources include the surgeon, anesthesia team, nursing staff, equipment and physical room, and the case will be canceled or delayed if one or more of these resources are unavailable. Also the perioperative environment differs from manufacturing that companies can store the unsold product in inventory and sell them later but in hospitals unused OR time cannot be recovered.

Moreover US health care system will serve estimated thirty-two million Americans in addition, since they will be covered by insurance by 2019. US medical system must serve at least 15 percent higher demand of the patients. Research has demonstrated that

because of high cost of medical system extensions, most efficient way to overcome this issue, can be investing in streamlining patient flow and developing more efficient scheduling framework instead of adding resources, staffs and beds (Litvak & Bisognano, 2011). The ability to forecast the future demand provides numerous benefits to all stakeholder groups as shown in below.

Figure 1: Benefits of efficient scheduling



Also, the automatic forecasting of univariate time series are often needed in business. In implementing the forecasting models, there may be nobody suitably trained in using and implementing time series models to produce the forecasted values. In these circumstances, an automatic forecasting algorithm is an essential tool.

1.3 Thesis Overview

The primary goal of this thesis is increasing the accuracy of the forecasted block time, which will be allocated to surgeon, group of surgeon or surgical service area, and perform the forecasting process without the human intervention. In designing the forecasting framework, it's important that any such system is smart enough to keep efficiency regardless of the current demand state. The proposed framework in this thesis uses the historical data of the surgical demands, then captures the pattern and propose the best estimate of the block time that minimize the underutilization and overutilization of the resource and staffs and system shocks, which resulted in significant cost saving. This framework employs the machine learning and traditional statistical models e.g. ARMA, exponential smoothing, neural network and hybrid methodology. The proposed forecasting framework in this thesis incorporate the forecasting algorithms in order to determine an appropriate structure for time series model, estimate the parameters and compute the forecasts values. They must be robust to unusual time series patterns, and applicable to large numbers of series without any user intervention (Hyndman & Khandakar, 2008). And finally by applying the multi-criteria decision modeling tools, best forecasting method will be selected.

The results from this forecasting framework, gives the schedulers very good heads up to schedule the block time for future that minimize the uncertainty effects of the future demands and help them to have more confidence on their decisions for the future schedules. This adaptive decisions help medical centers to recognize the best utilization of the OR resources and also reduction in cost associated with idle ORs and staff overtime.

CHAPTER 2 PREPROCESSING

2.1 Time Series

Time series is a random variable which is the chronology of observations, Therefore, it is a stochastic process. Examples include the monthly demand for a product, weekly surgical demands and so on. Forecasting the future value of the time series data is pretty helpful in the field of operations research since these forecasts often provide the required information for developing decision making models. Time series analysis consists of approaches and methods which analyze the time series data in order to extract behavior and statistical characteristics of the data. These models then can be applied to forecast the future values of variable of interest.

2.2 Stationarity

A very important type of time series is a stationary time series. A time series is said to be strictly stationary if its properties are not affected by a change in the time origin. That is, if the joint probability distribution of the observations $y_t, y_{t+1}, \dots, y_{t+n}$ is exactly the same as the joint probability distribution of the observations $y_{t+k}, y_{t+k+1}, \dots, y_{t+k+n}$, then the time series is strictly stationary. The Stationarity property of a time series is related to its statistical properties in time. In other hand, a stationary time series exhibits similar "statistical behavior" in time and this is often characterized as a constant probability distribution in time. We can define stationarity (or weak stationarity) as follows:

- The expected value of the time series does not depend on time
- The autocovariance function defined as $Cov(y_t, y_{t+k})$ for any lag k is only a function of k and not time: that is $\gamma_y(k) = Cov(y_t, y_{t+k})$

In a crude way, the Stationarity of a time series can be determined by taking arbitrary "snapshots" of the process at different points in time and observing the general behavior of the time series. If it exhibits "similar" behavior, one can then proceed with the modeling efforts under the assumption of stationarity. The stationarity or non-stationarity of a time series can significantly influence its behavior and properties. We can recognize the stationarity of the time series by observing the behavior of the autocorrelation function. Better and more methodological tests of stationarity also exist to test the stationarity of the time series. Mostly the time series with Stationary property have the constant mean and variance. We can use sample mean and sample variance to estimate these parameters. If the observations in time series are $y_1, y_2, \dots, y_T$ then for sample mean we have

$$\bar{y} = \hat{\mu}_y = \frac{1}{T}\sum_{t=1}^{T} y_t$$

And for sample variance we can use

$$S^2 = \hat{\sigma}_y^2 = \frac{1}{T}\sum_{t=1}^{T} (y_t - \bar{y})^2$$

2.3 Autocovariance and Autocorrelation Functions

If a time series is stationary this means that the joint probability distribution of any two observations, say, y_t, and y_{t+k}, is the same for any two time periods t and $t + k$ that are separated by the same interval k. Useful information about this joint distribution and hence about the nature of the time series, can be obtained by plotting a scatter diagram of all of the data pairs y_t, y_{t+k} that are separated by the same interval k. The interval k is

called the lag. The covariance between y_t and its value at another time period, say, y_{t+k} is called the autocovariance at lag k, defined by

$$\gamma_k = Cov(y_t, y_{t+k}) = E[(y_t - \mu)(y_{t+k} - \mu)]$$

The collection of the values of $\gamma_k, k = 0,1,2, \ldots$ is called the autocovariance function. Note that the autocovariance at lag , $k = 0$ is just the variance of the time series. That is; $\gamma_k = \sigma_y^2$.

The autocorrelation coefficient at the lag k is

$$\rho_k = \frac{E[(y_t - \mu)(y_{t+k} - \mu)]}{\sqrt{E[(y_t - \mu)^2]E[(y_{t+k} - \mu)^2]}} = \frac{Cov(y_t, y_{t+k})}{Var(y_t)} = \frac{\gamma_k}{\gamma_0}$$

The collection of the values of $\rho_k, , k = 0,1,2, \ldots$ is called the autocorrelation function (ACF). Note that by definition $\rho_0 = 1$. Also, the ACF doesn't have a scale of measurement for time series data, so it is a quantity without dimension. Furthermore $\rho_k = \rho_{-k}$; that is, the autocorrelation function is symmetric around zero, so it is only necessary to compute the positive (or negative) half.

It is necessary to estimate the autocovariance and autocorrelation functions from a time series of finite length, say, $y_1, y_2, \ldots, y_T$. The usual estimate of the autocovariance function is

$$c_k = \hat{\gamma}_k = \frac{1}{T}\sum_{t=1}^{T-k}(y_t - \bar{y})(y_{t+k} - \bar{y}), k = 0,1,2, \ldots, K$$

And the autocorrelation function is estimated by the sample autocorrelation function.

$$r_k = \hat{\rho}_k = \frac{c_k}{c_0}, \qquad k = 0,1,2, \ldots, K$$

The individual sample autocorrelations should be calculated up to lag k. where k is about $T/4$.

2.3.1 Autocorrelation function and stationarity

The sample ACF of stationary time series are cutting off or tailing off near zero after a few lags, But for non-stationary time series this sample ACF is very persistent; that is, it decays gradually and the value of the sample autocorrelations even at the long lags are large. In other hand a strong and slowly dying ACF will suggest deviations from stationarity.

2.4 Automated Stationarity Test

Since using the autocorrelation function for stationarity test need visual inspection, it cannot be useful for automated forecasting framework. In order to eliminate the visual inspection, the proposed solution is KPSS test.

2.4.1 Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

In KPSS test the null hypothesis assumes that process is stationary and alternative hypothesis assumes that series is non-stationary due to presence of the unit root. In KPSS test series of observations is presented as a combinations of the three components: deterministic trend, random walk and error term:

$$y_t = \xi t + r_t + \varepsilon_t$$

And for random walk we have

$$r_t = r_{t-1} + u_t$$

Which we have $u_t \sim iid(0, \sigma_u^2)$

The initial value r_0 is a fixed value and considered as an intercept. The stationarity hypothesis is $\sigma_u^2 = 0$, in this situation time series are stationary around the trend. There is another condition that we have $\xi = 0$ which under null hypothesis, time series is level stationary. Test statistics is the LM test for hypothesis that random walk component has zero variance. Statistics we will use is one sided LM statistics, for the $\sigma_u^2 = 0$, as a null hypothesis. There are also some stronger assumption that the u_t is normal and the ε_t is $iid\ N(0, \sigma_\varepsilon^2)$. The reason we are interested in one sided LM test not the two sided, is because parameter value determined in null hypothesis is in the boundary of the parameter space

$$H_0: \sigma_u^2 = 0$$

$$H_1: \sigma_u^2 \neq 0$$

Let $e_t, t = 1, \dots, T$ be the residuals from the regression of y on an intercept and time trend. Let $\hat{\sigma}_\varepsilon^2$ be the estimate of the error variance from this regression. We have partial sum of the residuals as:

$$S_t = \sum_{t=1}^{t} e_i\,, for\ t = 1, \dots, T$$

For LM statistic we have

$$\sum_{t=1}^{T} S_t^2 / \sigma_\varepsilon^2$$

In addition if we want to test the null hypothesis of the level stationarity instead of the trend stationarity, we need just to change the e_t to the residuals from the regression of y on an

intercept only, ($e_t = y_t - \bar{y}$) and rest of the process in unchanged (Mahadeva & Robinson, 2004) (Kwiatkowski, Phillips, Schmidt, & Shin, 1992).

We can decide about the stationarity of the time series with referring to the tables 1 and 2. For each critical level there is corresponding critical value. If the value of the statistics surpass the critical value of the corresponding level then null hypothesis is rejected and time series data are non-stationary.

Table 1: Level Stationarity

Critical level	0.10	0.05	0.025	0.01
Critical value	0.347	0.463	0.574	0.739

Table 2: Trend Stationarity

Critical level	0.10	0.05	0.025	0.01
Critical value	0.119	0.146	0.176	0.216

2.5 Differencing

One of the most applicable approaches to make the non-stationarity data, stationary, is differencing. With applying the difference operator to the original time series we can obtain a new time series. We will have

$$x_t = y_t - y_{t-1} = \nabla y_t$$

Where ∇ is the (backward) difference operator. Another way to write the differencing operation is in terms of a backshift operator B, defined as $By_t = y_{t-1}$, so

$$x_t = (1 - B)y_t = \nabla y_t = y_t - y_{t-1}$$

With $\nabla = (1 - B)$. Differencing can be performed successively if necessary, until the data becomes stationary. In general, powers of the backshift operator and the backward difference operator are defined as

$$B^d y_t = y_{t-d}$$

$$\nabla^d = (1 - B)^d$$

Another way to eliminate the trend is to fit a regression model describing the trend component to the data and then subtracting it out of the original observations, leaving a set of residuals that are free of trend. But differencing has two advantages relative to fitting a trend model to the data. First, it does not require estimation of any parameters, so it is a more parsimonious approach; and second, model fitting assumes that the trend is fixed throughout the time series history and will remain so in the (at least immediate) future. In other words, the trend component, once estimated, is assumed to be deterministic but differencing can allow the trend component to change through time. The first difference accounts for a trend that impacts the change in the mean of the time series. The second difference accounts for changes in the slope of the time series, and so forth. Usually, one or two differences are all that is required in practice to remove an underlying trend in the data.

2.5.1 Convert differenced forecast

When the original data represent the non-stationary behavior, we perform the differencing until we reach stationary time series data. Then we uses the difference data to forecast the future values. Since the forecasted values are based on differenced data we need to convert them to the original scale. The usual approach for conversion is to cumulatively add the differenced forecasts to the last observation. We can do this for also

more than one step ahead forecast values. If z are the differenced data and y are the original, then we will have.

$$\hat{y}_{t+1} = y_t + \hat{z}_{t+1},$$

$$\hat{y}_{t+2} = \hat{y}_{t+1} + \hat{z}_{t+2}$$

CHAPTER 3 EXPONENTIAL SMOOTHING

3.1 Intro

We can often think of a data set as consisting of two distinct components: signal and noise. Signal represents any pattern caused by the intrinsic dynamics of the process from which the data is collected. These patterns can take various forms from a simple constant process to a more complicated structure that cannot be extracted visually or with any basic statistical tools. The constant process is represented as $y_t = \mu + \varepsilon_t$ where μ represent the underlying constant level of system response and ε_t is the noise at time t. the ε_t are often assumed to be uncorrelated with mean 0 and constant variance σ_ε^2. Exponential smoothing can be used to separate the signal and the noise of the time series, so the smoother is considered as a filter to estimate the signal. The smoothers achieve this by relating the current observation to the previous ones. The concept of the exponential smoothing is that earlier data carry less information about the change in the process, so we can discount the older data, Instead of giving all the observations equal weights, since they don't have the same amount of influence on process. Thus if the weights of each observation are changed so that earlier observations are weighted less, more precise underlying pattern of the data can be obtained. To do this one way is to give geometrically decreasing weights to the previous observations. Hence an exponentially weighted smoother is obtained by introducing a discount factor θ as

$$\sum_{t=0}^{T-1} \theta^t y_{T-t} = y_T + \theta y_{T-1} + \theta^2 y_{T-2} + \cdots + \theta^{T-1} y_1 \quad (4.3)$$

And if the previous observations are to be discounted in a geometrically decreasing manner, then we should have $|\theta| < 1$.

Sum of the weights (θ) is

$$\sum_{t=0}^{T-1} \theta^t = \frac{1-\theta^T}{1-\theta}$$

We can adjust the smoother by multiplying it by $(1-\theta)/(1-\theta^T)$. However, for large T values, θ^T goes to zero and so the exponentially weighted average will have the following form

$$\tilde{y}_T = (1-\theta)(y_T + \theta y_{T-1} + \theta^2 y_{T-2} + \cdots + \theta^{T-1} y_1)$$

This is called a simple or first-order exponential smoother. An alternate expression in a recursive form for simple exponential smoothing is given by

$$\tilde{y}_T = (1-\theta) y_T + \theta \tilde{y}_{T-1}$$

The recursive form shows that first-order exponential smoothing can also be seen as the linear combination of the current observation and the smoothed observation at the previous time unit. The simple exponential smoother is often represented in a different form by setting $\lambda = 1-\theta$.

$$\tilde{y}_T = \lambda y_T + (1-\lambda)\tilde{y}_{T-1}$$

In this representation the discount factor, λ, represents the weight put on the last observation and $(1-\lambda)$ represents the weight put on the smoothed value of the previous observations.

3.2 The initial value

Since $\tilde{y}_0$ is needed in the recursive calculations that start with $\tilde{y}_1 = \lambda y_1 + (1-\lambda)\tilde{y}_0$, its value needs to be estimated. But we have

$$\tilde{y}_T = \lambda(y_T + (1-\lambda)y_{T-1} + \cdots + (1-\lambda)^{T-1}y_1) + (1-\lambda)^T\tilde{y}_0$$

Which means that as T gets large and hence $(1-\lambda)^T$ gets small, the contribution of $\tilde{y}_0$ to $\tilde{y}_T$ becomes negligible. Thus for large data sets, the estimation of $\tilde{y}_0$ has little relevance. Nevertheless, two commonly used estimates for $\tilde{y}_0$ are the following.

1. Set $\tilde{y}_0 = y_1$ If the changes in the process are expected to occur early and fast. This choice for the starting value for $\tilde{y}_T$ is reasonable.
2. Take the average of the available data or a subset of the available data, $\bar{y}$, and set $\tilde{y}_0 = \bar{y}$. If the process is at least at the beginning locally constant, this starting value may be preferred.

3.3 Choice of Discount Factor

In general, as λ gets closer to 1, more emphasis is put on the last observation, the smoothed values will approach the original observations. Two extreme cases will be when $\lambda = 0$ and $\lambda = 1$. In the former, the smoothed values will all be equal to a constant, namely $\tilde{y}_0$. We can think of the constant line as the "smoothest" version of whatever pattern the actual time series follows. For $\lambda = 1$, we have $\tilde{y}_T = y_T$ and this will represent the "least" smoothed (or unsmoothed) version of the original time series. We can accordingly expect the variance of the simple exponential smoother to vary between 0 and the variance of the original time series based on the choice of λ. Mostly, values between 0.1 and 0.4 are

often recommended and do indeed perform well in practice. To achieve more accurate value, We can define the sum of the squared errors as

$$SS_E(\lambda) = \sum_{t=1}^{T} e_{t-1}^2(1)$$

For a given historic data, we can in general calculate SS_E values for various values of λ and pick the value of λ that gives the smallest sum of the squared forecast errors.

3.4 Modeling Time Series Data

The constant process can be represented as

$$y_t = \beta_0 + \varepsilon_t$$

The smoothing techniques are effective in illustrating the underlying pattern in the time series data. We can find another use for the exponential smoothers: model estimation. Indeed for the constant process, we can see the simple exponential smoother as the estimate of the process level, or in an estimate of β_0. To show this in greater detail we need to introduce the sum of weighted squared errors for the constant process. The sum of squared errors for the constant process is given by

$$SS_E = \sum_{t=1}^{T} (y_t - \mu)^2$$

If we argue that not all observations should have equal influence on the sum and decide to introduce a string of weights that are geometrically decreasing in time, the sum of squared errors becomes

$$SS_E^* = \sum_{t=0}^{T-1} \theta^t (y_{T-t} - \beta_0)^2$$

Where $|\theta| < 1$. To find the least squares estimate for β_0 we take the derivative of equation above with respect to β_0 and set it to zero:

$$\frac{dSS_E^*}{d\beta_0} = -2\sum_{t=0}^{T-1} \theta^t \left(y_{T-t} - \hat{\beta}_0\right) = 0$$

Then we then have

$$\hat{\beta}_0 = (1-\theta)\sum_{t=0}^{T-1} \theta^t y_{T-t}$$

We can see that $\hat{\beta}_0 = \tilde{y}_T$. Thus the simple exponential smoothing procedure does in fact provide a weighted least squares estimate of $\hat{\beta}_0$ in the constant process with weights that are exponentially decreasing in time.

3.5 Forecasting

We have so far considered exponential smoothing techniques as either visual aids to point out the underlying patterns in the time series data or to estimate the model parameters. The latter brings up yet another use of exponential smoothing-forecasting future observations. At time T, we may wish to forecast the observation in the next time unit, $T+1$, or further into the future. For that, we will denote the $\tau - step - ahead$ forecast made at time T as $\hat{y}_{T+\tau}(T)$. Since the constant model consists of two parts (β_0 that can be estimated by the first-order exponential smoother and the random error that cannot be predicted) our forecast for the future observation is simply equal to the current value of the exponential smoother.

$$\hat{y}_{T+\tau}(T) = \tilde{y}_T$$

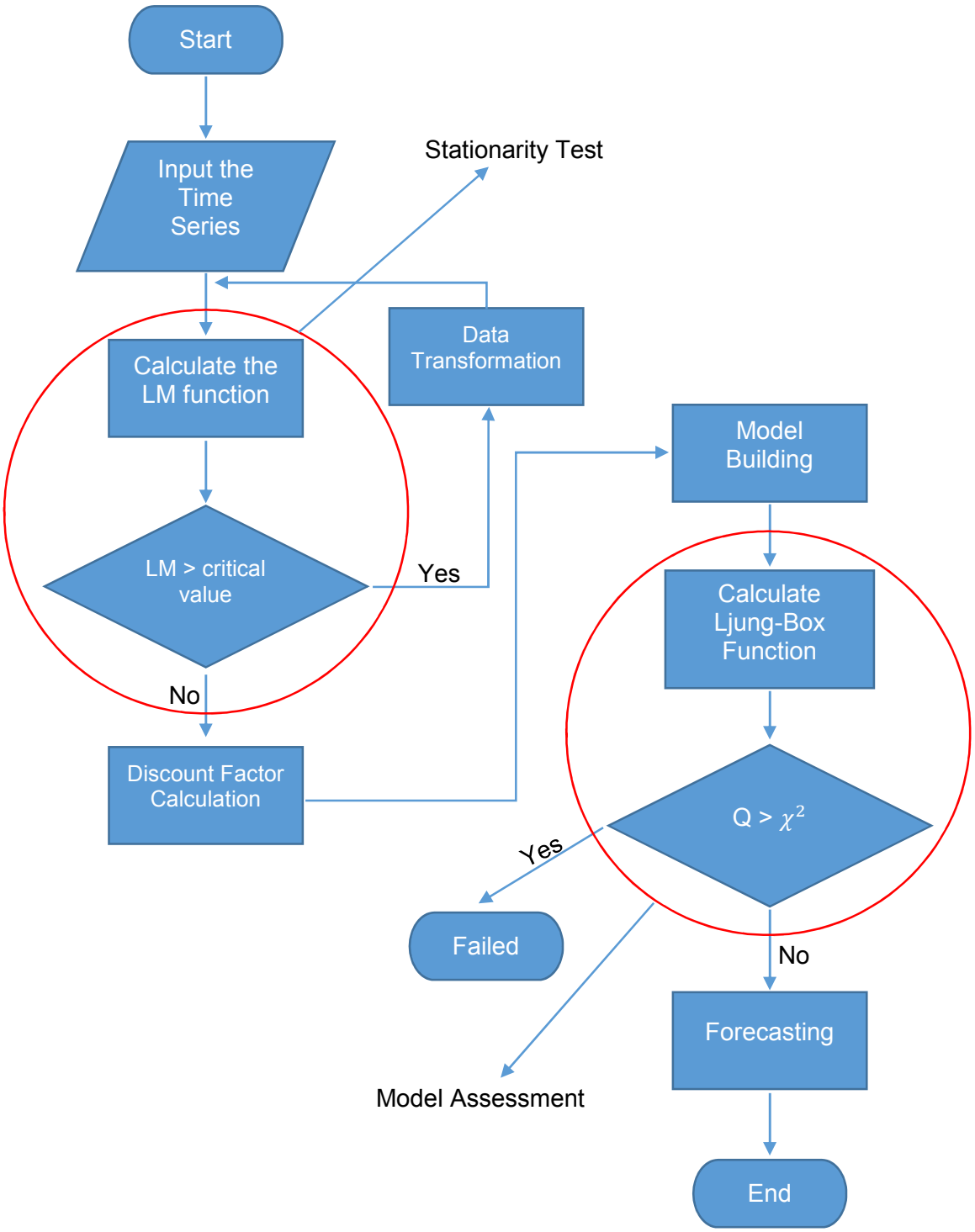
We should consider that, for the constant process, the forecast is the same for all future values. Since there may be changes in the level of the constant process, forecasting all future observations with the same value will most likely be misleading. However, as we start accumulating more observations, we can update our forecast.

Also we can have

$$\hat{y}_{T+\tau} = \hat{y}_T + \lambda e_T$$

Where $e_T = y_T - \hat{y}_T$ is called the one-step-ahead forecast or prediction error. The interpretation of the forecasting formula, makes it easier to understand the forecasting process using exponential smoothing: our forecast for the next observation is simply our previous forecast for the current observation plus a fraction of the forecast error we made in forecasting the current observation. The fraction in this summation is determined by λ. Hence how fast our forecast will react to the forecast error depends on the discount factor. A large discount factor will lead to fast reaction to the forecast error but it may also make our forecast react fast to random fluctuations.

Figure 2: Exponential Smoothing Forecasting Algorithm



CHAPTER 4 THE ARMA MODEL

4.1 Intro

In exponential smoothing method, the time series can be represented as a combination of two component: deterministic and stochastic. The deterministic part is a function of time whereas in stochastic component we assumed that some random noise is added to the deterministic signal that totally represent the stochastic behavior of the time series. In practice, the random noise assumption is often violated and usually successive observations show serial dependence. Under these circumstances, forecasting methods based on exponential smoothing may be inefficient and sometimes inappropriate because they do not take advantage of the serial dependence in the observations in the most effective way. To formally incorporate this dependent structure, we will explore a general class of models called autoregressive moving average or ARMA models.

4.2 ARMA Process

In an autoregressive moving average model, the future value of a variable is a linear function of some past observations plus the random errors. The process has the following form

$$y_t = \delta + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$$

Where y_t denotes the actual value and ε_t denotes the random error at time t. ; φ_i $(i = 1,2,..,p)$ and θ_j $(j = 1,2,..,q)$ are parameters of the model. p and q referred to orders of the model and are integers. Random errors, ε_t, are assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2.

There are several special cases of the ARMA models. If $q = 0$, then it becomes an AR model of order p. When $p = 0$, the model reduces to an MA model of order q. Using back shift operators we can have

$$\Phi(B)y_t = \delta + \Theta(B)\varepsilon_t$$

Where, we have ε_t as a white noise process. The important issue in constructing the ARMA model is to recognize the appropriate order of the model (p, q).

4.2.1 Modified Box-Jenkins Methodology

The Box–Jenkins methodology, which proposed by George Box and Gwilym Jenkins, is the set of steps for time series analysis and forecasting which applies the autoregressive moving average (ARMA) to find the best model of a time series. A three-step iterative procedure is used to build an ARIMA model. First, a tentative model of the ARIMA class is identified through analysis of historical data. Second, the unknown parameters of the model are estimated. Third, through residual analysis, diagnostic checks are performed to determine the adequacy of the model, or to indicate potential improvements. Based on goal of this thesis, the modified Box-Jenkins methodology has been proposed to fulfill the automated framework requirements. We will discuss them in more detail.

4.2.2 Model Identification

Model identification efforts should start with preliminary efforts in understanding the type of process from which the data is coming and how it is collected. The process's perceived characteristics and sampling frequency often provide valuable information in this preliminary stage of model identification. In today's data rich environments, it is often expected that the practitioners would be presented with "enough" data to be able to

generate reliable models. Simple time series plots should be used as the preliminary assessment tool for stationarity. The sample ACF and PACF of the time series of the original time series should be obtained. Depending on the nature of the autocorrelation, the first 20-25 sample autocorrelations and partial autocorrelations should be sufficient with the $\pm 2/\sqrt{N}$ limits can be used as a guide for identifying AR or MA models.

Table 3: Model Identification Based on ACF and PACF

Model	ACF	PACF
AR(P)	Dies down (decays)	Cuts off (zero for h > p)
MA(q)	Cuts off (zero for h > q)	Dies down (decays)
ARMA(p, q)	Dies down (decays)	Dies down (decays)

4.2.2.1 Step-wise algorithm for model selection

Using autocorrelation function and partial autocorrelation function is a good approach for determining the order of the model, but it needs visual inspection by the human. That's why it cannot be used in automated forecasting framework. The best solution for determining the order of the model is based on the step-wise search algorithm. Based on this algorithm we start with some initial value for model parameters then compare their adequacy and select the best model. In the next step by changing the parameters, we create new set of the models include the best model from last set and new set models and again compare their adequacy and select the best model among the current set of models. We continue this approach until reach the best model which with changing the parameters no better model can obtained.

4.2.2.2 Model selection Criteria

In order to compare the models in step-wise search algorithm we need a criteria. The Akaike information criterion is one of the most strong and popular method for comparing the adequacy of multiple, possibly models. The value of the AIC by itself, doesn't represent any meaning. It can be useful when it is compared to the AIC of a series of models, then the model with the lowest AIC could be considered as the best model among all models specified for the data. If we consider poor models, the AIC will select the best of the poor models. So it's so important to dedicate time to recognize as much as possible bigger set of candidate models based on previous investigations, as well as judgement and a knowledge of the system under study. After having specified the set of potential models based on the AIC we can choose the best model for forecasting.

$$AIC = 2k + n\ln(RSS/n)$$

RSS: Residual sum of squares

k: Number of the parameters in model

n: Number of observations

4.2.3 Parameter Estimation

When the appropriate ARMA model determined, next step is to estimate the φ, θ and δ. There are several methods such as methods of moments, maximum likelihood, and least squares that can be employed to estimate the parameters in the tentatively identified model. In this thesis the least squared method has been applied to estimate the model parameters.

4.2.4 Diagnostic Checking

After a tentative model has been fit to the data, we must examine its adequacy and, if necessary, suggest potential improvements. This is done through residual analysis. The residuals for an ARMA(p, q) process can be obtained from

$$\hat{\varepsilon}_t = y_t - \left(\hat{\delta} + \sum_{i=1}^{p} \hat{\varphi}_i y_{t-i} - \sum_{i=1}^{q} \hat{\theta}_i \varepsilon_{t-i} \right)$$

If the specified model is adequate and hence the appropriate orders p and q are identified, it should transform the observations to a white noise process. Thus the residuals should behave like white noise. Let the sample autocorrelation function of the residuals be denoted by $\{r_e(k)\}$. If the model is appropriate, then the residual sample autocorrelation function should have no structure to identify. That is, the autocorrelation should not differ significantly from zero for all lags greater than one. But again like model identification this approach which is proposed by Box-Jenkins methodology needs visual inspection. So we need an alternative that can be used in automated forecasting framework.

4.2.4.1 Ljung-Box Test

The Ljung-Box test is applied to the residuals of a time series after fitting an ARMA(p,q) model to the time series data. The test examines autocorrelations of the residuals. If the autocorrelations are very small, we conclude that the model does not exhibit significant lack of fit. The Ljung–Box test statistic can be defined as follows

$$Q = n(n+2) \sum_{k=1}^{h} (n-k) r_e^2(k)$$

Where n is the length of the time series, $r_e(k)$ is the kth autocorrelation coefficient of the residuals, and h is the number of lags to test. Large values of Q indicate that there are significant autocorrelations in the residual series. We should reject the hypothesis of model adequacy if Q exceeds an approximate small upper tail point of the chi-square distribution with m degrees of freedom and significant level of α, where $m = h - p - q$.

$$Q > \chi^2_{1-\alpha,m}$$

Null hypothesis H_0 : adequate fitted model

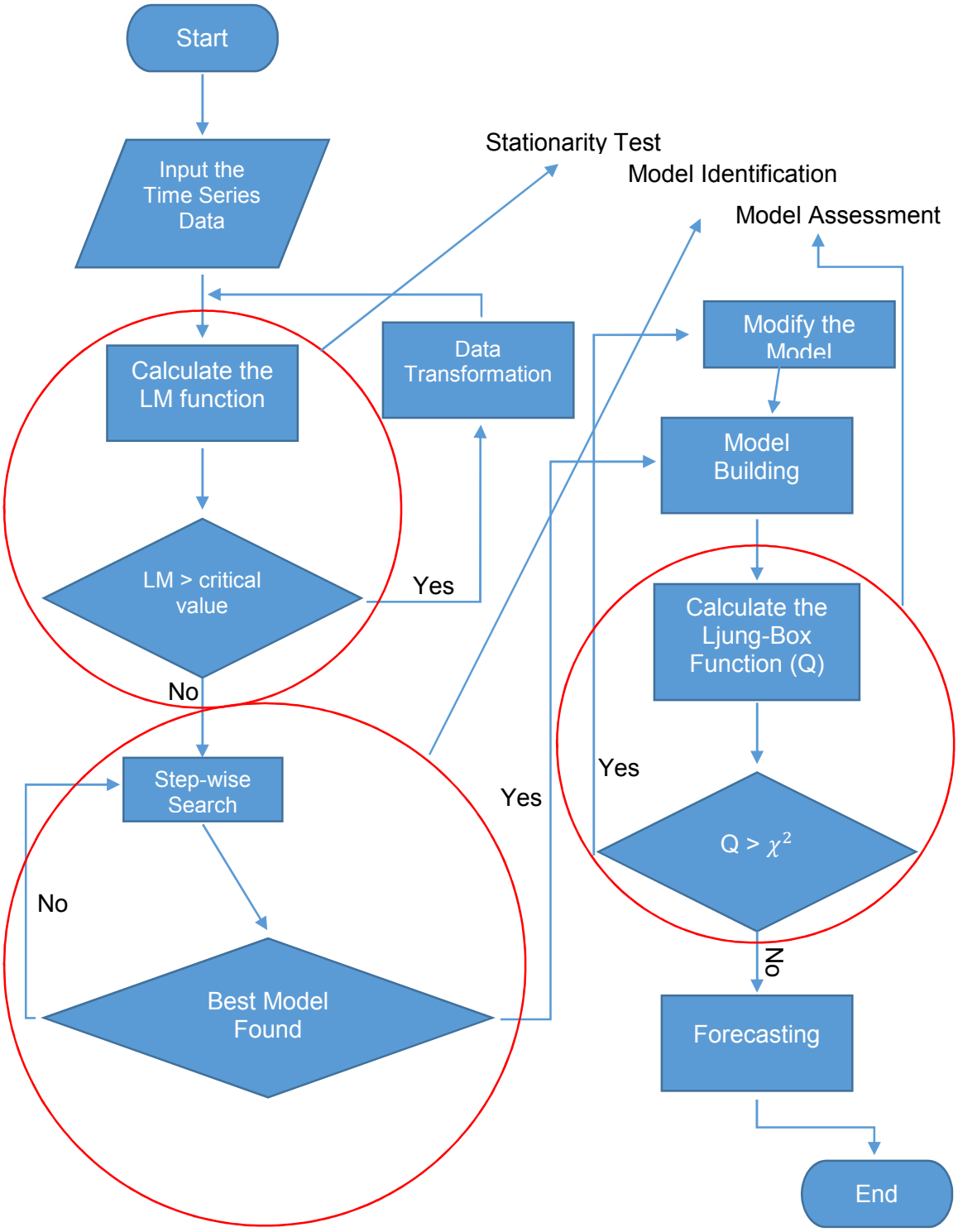
Alternative Hypothesis: H_1: not adequate fitted model

4.3 ARMA Process Forecasting

The main purpose of modeling a time series is to make forecasts which are then are used directly for making decisions. Once an appropriate time series model has been fit, it may be used to generate forecasts of future observations. If we denote the current time by T, the forecast for $y_{T+\tau}$ is called the τ period-ahead forecast and denoted by $\hat{y}_{T+\tau}(T)$. For prediction of the future values we can use conditional expectation of $y_{T+\tau}$ given current and previous observations, that is, $y_T, y_{T-1}, \ldots$.

$$\hat{y}_{T+\tau}(T) = E[y_{T+\tau}|y_T, y_{T-1}, \ldots]$$

Figure 3: ARMA Forecasting Algorithm



CHAPTER 5 ARTIFICIAL NEURAL NETWORK

5.1 What is the neural network?

Working on artificial neural networks, which commonly referred to as neural networks, has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer. Typically, brain neurons are slower than silicon logic gates. Process speed in a silicon chip is in nanosecond range, whereas neural process speed is in the millisecond range. Brain compensate the process slowness by constructing interconnections between neurons and consequently making the networks of the neurons. It is estimated that human cortex includes order of 10 billion neurons, and 60 trillion synapses or connections. These network of the neurons and interconnection between them significantly improve the efficiency of the brain structure. Also, the brain is very efficient in energy consumption in comparison to the best computers (Hajek, 2005). The brain working procedure is nonlinear, parallel and highly complex. It has the capability of performing certain computations many times faster than the fastest digital computer. Consider, for example, human vision system. It gets data from environment, then processes the data, and finally supply the information we need for interaction with the environment. To be specific, the brain continuously perform the recognition tasks (e.g. recognizing a familiar face embedded in an unfamiliar scene) in of the order of 100-200 ms (Hajek, 2005).

5.2 How neural network helps us?

Neural networks use the massive parallel distributed structure and have the capability of learning and Generalization. These capabilities make it possible for neural networks to solve complex problems that are difficult to solve by traditional tools. It is

important to know, however, we are far away from building a computer architecture that exactly mimics a human brain but neural networks offer the variety of tools to solve many intractable problems (Hajek, 2005). The neural networks have the following useful properties

5.2.1 Nonlinearity

A neuron is actually considered as a nonlinear tool. As a result, a neural network that is set of neurons and interconnection between them, is itself nonlinear. This property helps to capture the nonlinear patterns of the time series data, where many statistical tools are unable to do that.

5.2.2 Input-output mapping

In supervised learning, training algorithm is based on modifying synaptic weights of a neural network by using set of training sample data. Each sample include an input signal and the corresponding output signal. In each iteration, training algorithm modifies the weights in a way that minimize the difference between the desired output and the actual output of the neural network. The training algorithm iterates until the network reaches a stable state, where the synaptic weights changes is not significant.

5.2.3 Adaptivity

Neural networks have an inherent capability to adjust their synaptic weights regarding the changes happen in their environment. Moreover, when data are nonstationary a neural network can modify synaptic weights in real time to compensate the non-stationarity issue.

5.2.4 Contextual information

Every neuron in the network potentially affected by the global activity of all other neurons in the network. Consequently, contextual information approach is an inherent property of the neural network.

5.2.5 Fault tolerance

A neural network structure design has an inherent ability to be fault tolerant in the sense that its performance is degraded gracefully, rather than experiencing catastrophic failure.

5.2.6 Uniformity of analysis and design

Neural networks have a great universality capability as information processors. It can be applied to the great variety of the field without any prior assumptions.

5.3 Artificial Neuron

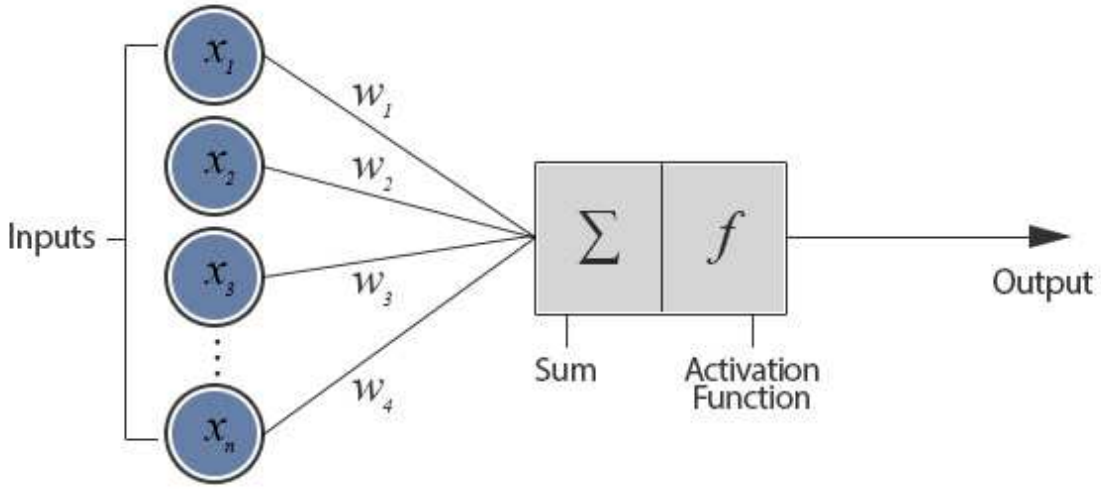


Figure 4: Artificial Neuron

A neuron is a fundamental unit for operation of the neural network. It includes the four basic elements:

1. **Synapses**: each synapse is identified by a weight. Input x_i connected to neuron j is multiplied by the corresponding synaptic weight w_{ji}. It is important to care the order of the subscripts of the synaptic weight w_{ji}. The first subscript refers to the neuron and the second subscript refers to the input node.
2. **Adder**: make summation of input signals, which are multiplied by the corresponding weights.
3. **Activation function**: limits the permissible amplitude of the output of a neuron to the finite value.
4. **Bias (threshold):** known as w_{k0} has the effect on the net input of the activation function. It causes decreasing or increasing of the net input.

5.4 The ANN approach in time series modeling

Transferring from linear forecasting methods to nonlinear methods, gives us plenty of forecasting models that can be used for time series analysis and forecasting. But one of the most applied models is artificial neural networks. ANN is a flexible computational framework for modeling a large range of nonlinear problems. The important advantage of the Artificial Neural Networks models in comparison to other classes of nonlinear model is that Artificial Neural Networks are universal model that can be applied to broad range of data types in term of forecasting with a high level of accuracy. The strength of the artificial neural network is under the parallel distributed information processing capability.

Basically the main characteristics of the ANN model is determined by the characteristics of the data (Zhang G. P., 2003).

5.5 Topology of the Artificial Neural Network

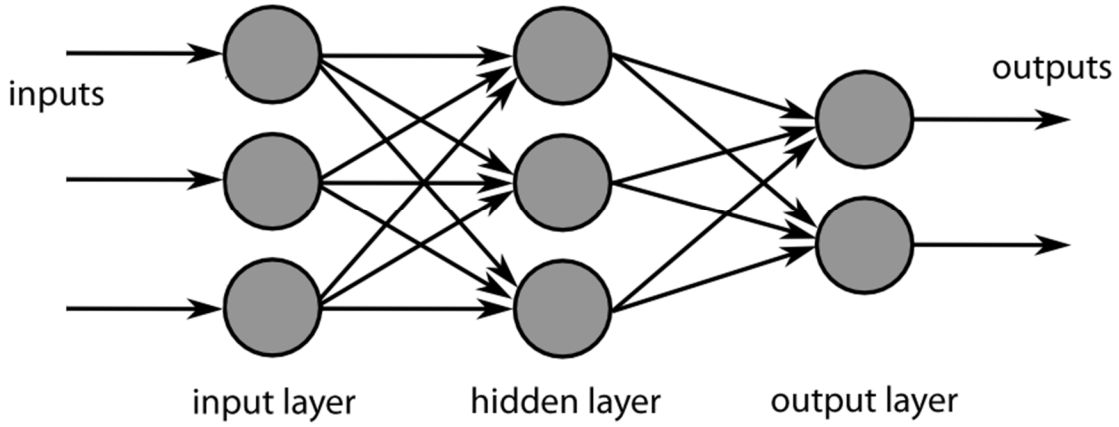


Figure 5: Single Hidden Layer Architecture

Input layer of every artificial neural network consists of source nodes (input vector) which supply the next layer (computation nodes). The output signals of the second layer are used as inputs to the third layer, and so on for the rest of the network. Typically, all the neurons in neural network architecture have their own inputs and output signals. The set of output signals of the output layer in artificial neural networks constitutes the overall response of the network from activation function supplied by the source nodes in the input layer. The network shown in figure 5 is referred to as a 3-3-2 network, it means this network has 3 source nodes, 3 neurons in hidden layer, and 2 output neurons. Generally, a feedforward network with p source nodes, h neurons in hidden layer, and q neurons in the output layer is referred to as $p - h - q$ network. If every node in each layer of the

artificial neural network is connected to every node in adjacent forward layer, it is called fully connected, otherwise it's called partially connected.

For time series modeling and forecasting, the mostly used model is Single hidden layer feedforward network (Zhang, Patuwo, & Hu, 1998). This model consists of the three layers which are connected by acyclic links. We can represent the relationship between the output (y_t) and the inputs $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$ by the following mathematical equation:

$$y_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right) + \varepsilon_t,$$

Where α_j $(j = 0,1,2, \dots, q)$ and β_{ij} $(i = 0,1,2, \dots, p; j = 1,2, \dots, q)$ are the connection weights of the model (parameters); p is the number of input nodes and q is the number of hidden nodes.

5.5.1 Network Parameters Selection

The single hidden layer network is powerful model which can perform accurately if the number of the neurons (q) in hidden layer has been selected properly (Hornik, 1989). In implementing artificial neural network even with the small number of hidden neuron, mostly the results for out of sample forecasting is good. This result can be due to the overfitting effect typically found in artificial neural network modeling. An overfitted model gives accurate results over the sample used for training but its performance for out of sample data is poor. For choosing q there is no any systematic rule and since its data dependent. In designing the artificial neural network for time series forecasting, In addition to selecting an appropriate number of hidden neurons, another important issue is the choosing the appropriate number of lagged observations, p (dimension of the input vector). p is the most important parameter in constructing an Artificial neural network since

it plays a prominent role in determining the (nonlinear) autocorrelation structure of the time series. Like number of the hidden neurons, there is no any systematic rule that can be used for choosing of p. It's also data dependent and mostly experiments are the only way to find the appropriate value of the p. Choosing the number of the nodes in output layer unlike to p and q is pretty straight forward, just based on the number of the output of the interest, we can determine the total number of the neurons in output layer. We will need just one output for our dataset, so naturally in network design there will be one neuron in output layer.

In this thesis stepwise search algorithm is proposed to select the number of lagged observations and hidden neurons. We start with some initial value for p and q to make a set of potential models, then evaluate them and select the best model. In next step we change the parameters and make the new models set plus the best model from last set and again compare them to choose the best one. We will continue this steps until we cannot select the better model. The criteria for model evaluation we can use the AIC, therefore the models with smallest value of the AIC, will be considered as the best model among the other models of the evaluation set.

5.6 Activation function

The sigmoid function is the most widely used function in time series modeling and forecasting in artificial neural networks (hidden layer transfer function), that is

$$f(x) = \frac{1}{1 + e^{-x}}$$

Hence, the Artificial neural network in model, in fact performs a nonlinear functional mapping from the past observations $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$ to the future value y_t, i.e.,

$$y_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-p}, w) + \varepsilon_t,$$

Where w is a vector of synaptic weights (parameters) and f is a nonlinear function, which map the input data to output data.

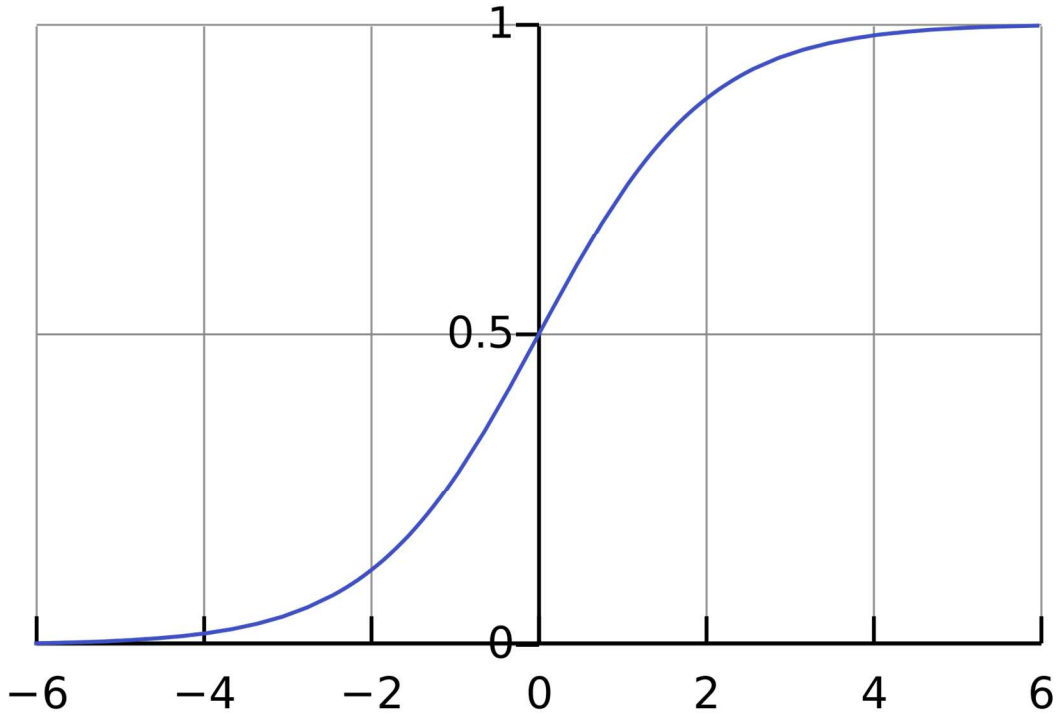


Figure 6: Sigmoid Function Curve

This big interest in using sigmoid function is its nonlinearity characteristics and the computational simplicity of its derivative, which is an important feature in construction of the artificial neural network. The derivative of the sigmoid function is as below.

$$f(x) = \frac{1}{1 + e^{-x}}$$

$$\frac{\partial y}{\partial x} = f(x)\big(1 - f(x)\big)$$

5.7 Network Learning

5.7.1 Supervised Learning

In supervised learning, the artificial neural network take advantage of existing external teacher, which is able to supply the artificial neural network with a desired response. The synaptic weight of the network are adjusted under the influence of the error signal. This adjustment is performed iteratively in a step-by-step fashion until getting to the desired results. Neural network has the capability to learn from the environment and continuously improve the performance. Neural network do this task by adjusting the synaptic weights based on the errors through the iterative process.

5.7.2 Error Correction Learning Rule

When network architecture (p, q) is determined, we can start the network training. In error correction learning rule, the parameters are estimated such that an overall accuracy criterion such as the mean squared error is minimized.

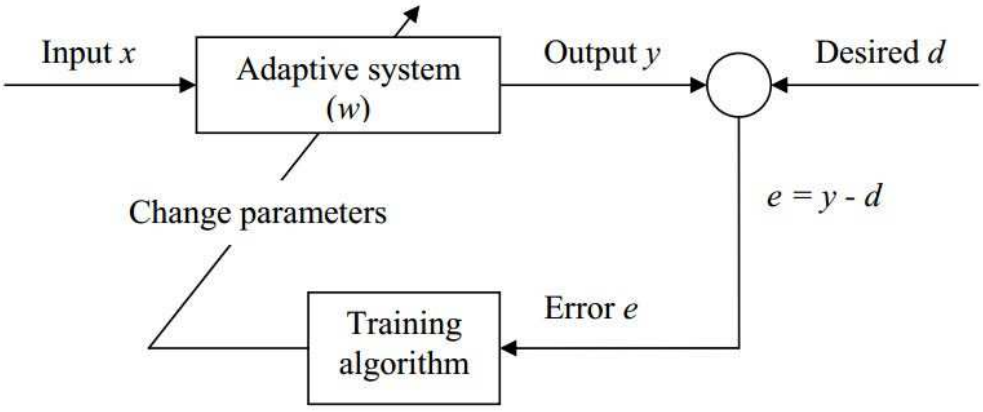


Figure 7: Error Correction Learning Rule

5.7.3 Backpropagation Algorithm with Gradient Descent method

Set of steps to adjust the synaptic weights of artificial neural network is called a learning algorithm. Basically there is no unique learning algorithm for designing of neural networks. There are plenty of algorithms, each one offers advantages of its own. Their difference lays in the way they adjust the synaptic weight. One of the most applied and effective algorithms is Backpropagation algorithm. Let $d_k(n)$ represent the desired response for neuron k at time n. The corresponding value of the response (output) of this neuron at time n be denoted by $y_k(n)$. The output $y_k(n)$ is the value, calculated by the activation function included in neuron k. Typically, $y_k(n)$, as a output of the neuron k is different from the desired response $d_k(n)$. So we can define the error signal as follows

$$e_k(n) = y_k(n) - d_k(n)$$

The goal of the error-correction learning algorithm is to minimize the error signal $e_k(n)$. A common criteria function that is usually used for this purpose is

$$E_j(\bar{x},\, \bar{w},\, d) = \frac{1}{2}\left(O_j(\bar{x},\, \bar{w}) - d_j\right)^2$$

We can call it error function, Where, $\bar{x}$ is the input vector, $\bar{w}$ is vector of the weights and d is the desired value. The network will be continuously optimized by minimizing E_j with respect to the synaptic weights of the network.

We have

$$O_j(\bar{x},\, \bar{w}) = \frac{1}{1 + e^{A_j(\bar{x},\, \bar{w})}}$$

As the output of the node j and

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji}$$

As the input signal to the activation function which Is the summation of the all input signals after multiplying in their corresponding weights. Let $w_{ji}(n)$ denote the value of the synaptic weight w_{ji} at time n and x_i denotes the ith input signal. The threshold input and its corresponding weights are also included in this formula as x_0 and w_0. In matrix form we have

$$A_j = [w_{j0}\, w_{j1} \dots\, w_{jp}] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_p \end{bmatrix} = w_j^T x$$

Based on backpropagation algorithm, we can adjust the synaptic weight by

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1 - O_j)x_i$$

And the value of the synaptic weights at time $n + 1$ is calculated as following.

$$w_{ji}(n + 1) = w_{ji}(n) + \Delta w_{ji}(n)$$

We continue to calculate the new weights until we obtain the desired output of the network.

CHAPTER 6 THE HYBRID METHODOLOGY

6.1 Intro

The ARIMA and Artificial neural networks models performs pretty well in their own linear and nonlinear domain in time series forecasting. But, both of them are not that much universal that can be used for all types of the data. The ARMA model faces problems in forecasting nonlinear problems. Also the artificial neural networks performance in dealing with linear problems lead to the mixed results and its performance fluctuate in comparison to the regular statistical models since it depends on the criteria like sample size and noise level (Markham & Rakes , 1998). So it's not recommended to use the artificial neural networks blindly to any type of data and in other hand it's not easy to completely recognize the characteristics of the data. The solution can be the hybrid methodology that uses the unique features of the each model, in other words, hybrid method advantages from the strength aspect of the each model. So the mixture of the linear modeling of the ARMA method and nonlinear modeling of the artificial neural network can be a great strategy for practical modeling of the time series. By combining ARMA and ANN, different aspects of the underlying patterns may be captured (Zhang G. P., 2003). We can split the time series to the linear autocorrelation structure and a nonlinear component.

$$y_t = L_t + N_t,$$

Where L_t denotes the linear component and N_t denotes the nonlinear component. These two components can be estimated from the original time series data. First, by using the ARMA method, we can model the linear component (L_t), After constructing the ARMA model, in diagnosis section, model adequacy is evaluated based on the residuals. Basically in appropriate ARMA model, Residuals doesn't show any linear correlation

structures, but, residual analysis is not able to detect any nonlinear patterns in the data. Generally there is no any statistical test to capture the nonlinearity of the data. So, even if the constructed model passed diagnosis test, it may still not be appropriate since the nonlinear relationships may not be modeled appropriately. Existing of the any nonlinear pattern in the residuals will not be considered by the ARMA model and it's one of the ARMA model limitations. Then we can calculate the residuals of the ARMA model. As we said above, If appropriate model was applied, then, the residuals from the ARMA model will contain only the nonlinear relationship. We can Let e_t denote the residual of the ARMA model at time t, that is

$$e_t = y_t - \hat{L}_t,$$

Where y_t is the actual value and $\hat{L}_t$ is the fitted value of ARMA model, at time t. In this case, modeling the residuals of the ARMA model, using artificial neural network, will help the modeler to capture the nonlinear pattern of the time series data. With n input nodes, the ANN model for the residuals will be

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t,$$

Where f is the function determined by the artificial neural network and ε_t is the white noise. If we denote the forecast by the artificial neural network as $\hat{N}_t$, the final forecast will be combination of the ARMA and ANN forecast and we will have

$$\hat{y}_t = \hat{L}_t + \hat{N}_t,$$

Implementation of the Hybrid methodology consists of two steps. First, an ARMA model is used to capture the linear part of the problem. Next, by implementing the artificial neural network we can model the residuals from the ARIMA model. Because of the inherent

characteristics of the ARIMA model, it cannot capture the nonlinear structure of the data, as a result, the residuals from linear model fitted by ARMA model will have information about the nonlinearity. The results from implementing the artificial neural network are used as the predictions of the error terms in ARMA model. The hybrid methodology employs strength of ARMA model as well as Artificial Neural Network in capturing underlying patterns of the data. It is a good approach to capture the linear and nonlinear patterns of the data separately, using different methods and then sum up the forecasted values from two different model in order to improve the overall performance of the time series forecasting.

CHAPTER 7 Method Selection

7.1 Intro

After implementing the different forecasting methods, the next important issue is selecting the method which satisfies the schedulers more than others. In choosing the best forecasting method, there are some criteria, which schedulers are interested in them, and according to them, best forecasting method must be selected. In this thesis, these measures are categorized into MSE, Error Variance, Over-utilized time and Under-utilized time.

Mean squared error (MSE)

MSE is one of the most common measures to evaluate the performance of the forecasting models. It is very important to decrease the total deviation from allocated block time in order to minimize the total idle time and over time in scheduling process. We can define the MSE by the average squared difference between the forecasted and the actual values.

Error Variance

In OR scheduling, only minimizing the total deviation from allocated block time is not enough. Having more constant deviation also is very important. The forecasting model, which absorbs the system shocks and decrease the unusual deviations from scheduled times is more desirable for OR schedulers.

Over-utilized Time

Another important measure is over-utilized time. Sometimes the schedulers are more interested in less over-utilized time rather than under-utilized time, based on higher cost of it.

Under-utilized Time

In contrast it's possible in some hospitals that minimizing the under-utilized time is more important than the minimizing over-utilized time, since they can decrease the wasted resources. In these cases, schedulers prefer the forecasting models with least under-utilized time.

7.2 Multi-Criteria Decision Modeling

The MCDM is a tool which help decision makers to make best decision with considering and evaluating multiple criteria in decision-making environments. MCDM is concerned with constructing and making decision over the problems involving multiple criteria. Steps of implementing MCDM is as follows.

7.2.1 Rescale the Scores

Since different measures are in different scale, first we need to rescale data to common unit, so all the measure can be comparable. We can do it by following formula.

$$Scaled\ Score = \frac{\left|Score - least\ preferred\ score\right|}{\left|most\ preferred\ score - least\ preferred\ score\right|}$$

Where the score is the value of interest, and least preferred score and most preferred score are worst and best value among all existing scores respectively.

7.2.2 Defining the Weights

In determining the best forecasting method, we should consider that measures are not equally important and some of them are more important than the others. For involving these issue in decision making process, weights should be assigned to the measures based on the preference. These weights reflect the relative importance of the measures. Choosing appropriate weights in this thesis for each criteria is based on the policy and characteristics of the health center. For calculating the weights first we rank the measures based on their importance, then assign the point to each measure in 0 to 100 scale. The most important measure will be assigned 100 points and others will be assigned less than 100 according to their priority. After all, each point should be divided by total summation of all points. The results will be the weights of each measure.

7.2.3 Score calculation

After rescaling the scores and calculating the corresponding weights, total score of the each method can be calculated as summation of the scores multiplied by corresponding weights. The earned values can be used to compare the different methods, in a way that the method with highest score, is considered as the most desirable method.

CHAPTER 8 EMPIRICAL RESULTS

8.1 Dataset

In John D. Dingell VA medical center which was studied in this Thesis, block times schedule change every 3 months. During this 3 months period each specialty has a base level of OR availability, which providing flexibility for upcoming surgical demand fluctuations. Additional fine-tuning occurs in the weekly block schedule when specialties already know ahead the list of surgeries they need to schedule or cancel. From the surgical activity routine at Detroit VAMC, we collected the surgery data from October 2009 to July 2010 for 43 consecutive weeks. The data set provided us with the following information:

- Specialty
- Date
- Regular Usage
- Over time Usage

The surgeons in hospital are grouped into 14 different groups (Table 4) based on their surgical service area. The total usage time of the OR by each group of specialty in a weekly basis, constitute our time series data. Data are split to two unequal parts. The first part which is used for model building includes the 70 percent of the data and the second part includes 30 percent which will be used as the test data. Using four different methods (exponential smoothing, ARMA, Artificial neural network and hybrid methodology), we want to forecast the amount of time that we need to allocate to each specialty each week. Based on the John D. Dingell VAMC policy, block time schedules are fixed for 3 months period. Since data points are in weekly basis, we will forecast for 12 consecutive weeks

and then will take the average of these values as the final amount of the block time during the next 3 months.

Table 4: Specialty Groups

	Specialty
1	Anesthesiology
2	Cardiac Surgery
3	General
4	Gynecology
5	Neurosurgery
6	Ophthalmology
7	Oral Surgery (Dental)
8	Orthopedics
9	Otorhinolaryngology
10	Peripheral Vascular
11	Plastic Surgery
12	Podiatry
13	Thoracic Surgery
14	Urology

8.2 Results

After implementing the four different methods for all 14 specialty blocks, the forecasting model structures which include discount factor of exponential smoothing method, orders of the ARMA model, number of input layer and hidden layer nodes of ANN and Hybrid models, presented as below.

Table 5: Models Structure (Anesthesia)

	Model Parameters
Exponential Smoothing	Lambda = 0.18
ARMA	p (AR) = 1
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 1
	No. of hidden layer nodes = 2
Hybrid	p (AR) = 1
	q (MA) = 1
	No. of input layer nodes = 1
	No. of hidden layer nodes = 3

Table 6: Models Structure (Cardiac Surgery)

	Model Parameters
Exponential Smoothing	Lambda = 0.36
ARMA	p (AR) = 5
	q (MA) = 5
Artificial Neural Network	No. of input layer nodes = 1
	No. of hidden layer nodes = 3
Hybrid	p (AR) = 5
	q (MA) = 5
	No. of input layer nodes = 2
	No. of hidden layer nodes = 8

Table 7: Models Structure (General)

	Model Parameters
Exponential Smoothing	Lambda = 0.52
ARMA	p (AR) = 1
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 1
	No. of hidden layer nodes = 4
Hybrid	p (AR) = 1
	q (MA) = 1
	No. of input layer nodes = 2
	No. of hidden layer nodes = 3

Table 8: Models Structure (Gynecology)

	Model Parameters
Exponential Smoothing	Lambda = 0.41
ARMA	p (AR) = 1
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 7
	No. of hidden layer nodes = 6
Hybrid	p (AR) = 1
	q (MA) = 1
	No. of input layer nodes = 6
	No. of hidden layer nodes = 8

Table 9: Models Structure (Neurosurgery)

	Model Parameters
Exponential Smoothing	Lambda = 0.10
ARMA	p (AR) = 3
	q (MA) = 2
Artificial Neural Network	No. of input layer nodes = 2
	No. of hidden layer nodes = 4
Hybrid	p (AR) = 3
	q (MA) = 2
	No. of input layer nodes = 2
	No. of hidden layer nodes = 5

Table 10: Models Structure (Ophthalmology)

	Model Parameters
Exponential Smoothing	Lambda = 0.10
ARMA	p (AR) = 0
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 2
	No. of hidden layer nodes = 5
Hybrid	p (AR) = 0
	q (MA) = 1
	No. of input layer nodes = 3
	No. of hidden layer nodes = 7

Table 11: Models Structure (Oral Surgery)

	Model Parameters
Exponential Smoothing	Lambda = 0.12
ARMA	p (AR) = 1
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 6
	No. of hidden layer nodes = 3
Hybrid	p (AR) = 1
	q (MA) = 1
	No. of input layer nodes = 6
	No. of hidden layer nodes = 4

Table 12: Models Structure (Orthopedics)

	Model Parameters
Exponential Smoothing	Lambda = 0.40
ARMA	p (AR) = 2
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 10
	No. of hidden layer nodes = 4
Hybrid	p (AR) = 2
	q (MA) = 1
	No. of input layer nodes = 9
	No. of hidden layer nodes = 2

Table 13: Models Structure (Otorhinolaryngology)

	Model Parameters
Exponential Smoothing	Lambda = 0.0887
ARMA	p (AR) = 1
	q (MA) = 5
Artificial Neural Network	No. of input layer nodes = 1
	No. of hidden layer nodes = 4
Hybrid	p (AR) = 1
	q (MA) = 5
	No. of input layer nodes = 2
	No. of hidden layer nodes = 2

Table 14: Models Structure (Peripheral Vascular)

	Model Parameters
Exponential Smoothing	Lambda = 0.0393
ARMA	p (AR) = 2
	q (MA) = 2
Artificial Neural Network	No. of input layer nodes = 10
	No. of hidden layer nodes = 9
Hybrid	p (AR) = 2
	q (MA) = 2
	No. of input layer nodes = 5
	No. of hidden layer nodes = 7

Table 15: Models Structure (Plastic Surgery)

	Model Parameters
Exponential Smoothing	Lambda = 0.24
ARMA	p (AR) = 3
	q (MA) = 2
Artificial Neural Network	No. of input layer nodes = 6
	No. of hidden layer nodes = 2
Hybrid	p (AR) = 3
	q (MA) = 2
	No. of input layer nodes = 1
	No. of hidden layer nodes = 6

Table 15: Models Structure (Podiatry)

	Model Parameters
Exponential Smoothing	Lambda = 0.16
ARMA	p (AR) = 5
	q (MA) = 4
Artificial Neural Network	No. of input layer nodes = 5
	No. of hidden layer nodes = 5
Hybrid	p (AR) = 5
	q (MA) = 4
	No. of input layer nodes = 3
	No. of hidden layer nodes = 4

Table 17: Models Structure (Thoracic Surgery)

	Model Parameters
Exponential Smoothing	Lambda = 0.24
ARMA	p (AR) = 1
	q (MA) = 1
Artificial Neural Network	No. of input layer nodes = 3
	No. of hidden layer nodes = 5
Hybrid	p (AR) = 1
	q (MA) = 1
	No. of input layer nodes = 1
	No. of hidden layer nodes = 2

Table 18: Models Structure (Urology)

	Model Parameters
Exponential Smoothing	Lambda = 0.10
ARMA	p (AR) = 1
	q (MA) = 2
Artificial Neural Network	No. of input layer nodes = 2
	No. of hidden layer nodes = 2
Hybrid	p (AR) = 1
	q (MA) = 2
	No. of input layer nodes = 2
	No. of hidden layer nodes = 5

As we mentioned in chapter 7, four different criteria (MSE, Error Variance, Over-utilized time and Under-utilized time) has been defined for making comparison between different methods in order to choose the best method, which is most desirable for OR schedulers. These measures are provided for all 14 specialty blocks as below.

Table 19: Forecasting Performance (Anesthesiology)

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	57.32	61.36	27.81	40.28
ARMA	58.21	58.32	25.26	42.1
ANN	60.49	61.36	21.14	45.86
Hybrid	58.40	63.02	24.79	42.43

Table 20: Forecasting Performance (Cardiac Surgery)

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	185.43	152.81	27.91	108.72
ARMA	179.27	140.26	30.28	105.4
ANN	140.62	150.54	59.05	67.95
Hybrid	225.82	153.32	131.41	20.29

Table 21: Forecasting Performance (General)

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	216.76	239.82	67.39	70.61
ARMA	217.63	244.15	64.93	79.49
ANN	216.45	241.81	75.22	68.77
Hybrid	214.26	235.24	70.08	73.91

Table 22: Forecasting Performance (Gynecology)

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	1.73	1.88	6.26	5.22
ARMA	1.72	2.87	5.87	6.38
ANN	2.29	2.35	8.07	1.20
Hybrid	1.9	1.2	7.29	2.13

Table 23: Forecasting Performance (Neurosurgery [5])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	66.32	51.30	67.12	13.96
ARMA	63.25	52.93	63.88	15.03
ANN	54.74	48.50	53.71	19.64
Hybrid	51.93	50.93	50.23	22.76

Table 24: Forecasting Performance (Ophthalmology [6])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	45.57	46.25	24.77	40.23
ARMA	45.71	47.10	24.43	40.45
ANN	45.69	45.90	24.49	40.5
Hybrid	47.13	42.60	21.72	43.27

Table 25: Forecasting Performance (Oral Surgery [7])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	5	6.69	9.29	8.13
ARMA	5	5.45	9.21	8.35
ANN	10.12	5.30	23.17	4.10
Hybrid	7.22	6.95	16.13	2.25

Table 26: Forecasting Performance (Orthopedics [8])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	235.40	226.30	45.08	101.85
ARMA	236.90	232.28	44.45	103.09
ANN	213.47	236.56	71.12	63.01
Hybrid	215.54	230.20	76.04	56.95

Table 27: Forecasting Performance (Otorhinolaryngology [9])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	87.42	82.78	17.89	70.21
ARMA	83.09	73.45	20	65.98
ANN	75.32	74.66	26.71	58.28
Hybrid	73.72	71.63	28.66	56.33

Table 28: Forecasting Performance (Peripheral Vascular [10])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	171.11	116.63	13.81	109.96
ARMA	174.58	94.05	14.38	112.09
ANN	148.88	105.65	19.31	96.05
Hybrid	118.60	112.78	29.48	70.51

Table 29: Forecasting Performance (Plastic Surgery [11])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	41.12	42.08	14.41	33.56
ARMA	41.99	41.11	12.65	34.82
ANN	54.44	42.95	3.30	51.09
Hybrid	42.1	43.65	12.44	34.96

Table 30: Forecasting Performance (Podiatry [12])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	5.04	5.33	9.67	14.45
ARMA	4.91	5.45	10.92	12.7
ANN	7.18	4.06	19.24	1.05
Hybrid	4.88	4.29	11.56	11.81

Table 31: Forecasting Performance (Thoracic Surgery [13])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	96.29	104.08	49.47	38.23
ARMA	95.88	110.45	47.75	39.46
ANN	95.75	106.09	40.31	47.36
Hybrid	95.41	107.55	44.07	42.92

Table 32: Forecasting Performance (Urology [14])

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	78.23	85.33	43.6	44.86
ARMA	80.22	83.45	55	37.99
ANN	79.32	83.93	37.68	50.31
Hybrid	76.34	84.65	41.92	46.07

8.3 Example for Peripheral Vascular

We will use the forecasting results of the peripheral vascular specialty as an example for MCDM application. Rescaled scores of the peripheral vascular block is as below.

Table 33: Rescaled Scores of Peripheral Vascular

	MSE	Error Variance	Over-utilized	Under-utilized
Exponential Smoothing	0.07	0	1	0.05
ARMA	0	1	0.93	0
ANN	0.46	0.48	0.62	0.39
Hybrid	1	0.17	0	1

The points based on the priority has been given as below table

Table 34: measures weights

	Rank Order	Points (0-100)	Weight (points/sum)
MSE	2	90	0.26
Error Variance	1	100	0.30
Over-utilized	3	85	0.25
Under-utilized	4	65	0.19
	Sum	340	1

And Total scores, which is summation of the scaled scores multiplied by the corresponding weights are as below.

Table 35: Total Scores for Peripheral Vascular

Methods	Total Score
Exponential Smoothing	0.28
ARMA	0.53
Artificial Neural Network	0.49
Hybrid	0.50

We can see that based on the weights we gave to the criteria, ARMA model lead to the better performance in comparison to other methods.

8.4 Conclusion

The MCDM methodology can be performed to choose the best model with desired results for every specialty blocks. However, based on some criteria's, we can conclude which method is stronger, but, it's so important to realize that the time series analysis and

forecasting is a very complicated field and just based on some historical data we cannot conclude that one specific method is best in all circumstance. Generally the time series forecasting is highly data dependent. So it's important to keep track of the results continuously. This thesis has given an overview of applying different forecasting methods in order to predict optimized time intervals to be allocated to corresponding surgery blocks, and besides, it developed the framework for each of the methods to be performed without human intervention. Ultimately, with examples of the real data, we have shown the value of using automated forecasting framework, which statistical ideas can be applied to healthcare area in order to increasing the efficiency and utilizations.

7.5 Future Research

For future research opportunities, I suggest to develop the framework in order to release the unused amount of the allocated block time a few days in advance. Whenever a block is released, it means, that block is available to other services or surgeons. Most elective surgery blocks are released three to five days in advance of the day of surgery. Block time releasing can be very useful idea in block time scheduling, since there is huge uncertainty in the surgery cases demand.

REFERENCES

Denton, B., Viapiano, J., & Vogl, A. (2007). Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science*, 13-24.

Hajek, M. (2005). *Neural Networks.*

Hornik, K. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks, 2*, 359-366.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software, 27*(3), 18-39.

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics, 54*(1-3), 159-178.

Litvak, E., & Bisognano, M. (2011). More patients, less payment: increasing hospital efficiency in the aftermath of health reform. *Health Affairs, 30*(1), 76-80.

Mahadeva, L., & Robinson, P. (2004). *Unit Root Testing to Help Model Building.* London: Unit Root Testing to Help Model Building.

Markham, I. S., & Rakes , T. R. (1998). The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression. *Computers and Operations Research, 25*(4), 251-263.

Milewski, F. (n.d.). *Operating Room Utilization and Perioperative Process Flow.*

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159-175.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35-62.

ABSTRACT

DEVELOPING AN AUTIMATED FORECASTING FRAMEWORK FOR PREDICTING OPERATION ROOM BLOCK TIMES

by

AZAD SADR HAGHIGHI

May 2015

Advisor: Dr. Alper Murat

Major: Industrial Engineering

Degree: Master of Science

Operating rooms are the most important part of the hospitals, since they have highest influence on financial state of the hospital. Because of high uncertainty in surgery cases demands and their durations, the scheduling of the surgeries becomes a very challenging and critical issue in hospitals. One of the most common approaches to overcome this uncertainty is applying block times which is the time intervals allocated to surgery groups in the hospital. Assigning sufficient amount of the time to each block, is very important, since overestimating lead to wasting resources and on the other hand underestimation causes the overtime staffing and probably surgery cancellation. The objective of this study is developing an automatic forecasting framework with applying a high performance forecasting methods to predict the future block time intervals for surgical groups. The main property of proposed forecasting framework is elimination of the human intervention which means the system follows the certain algorithms to perform the forecasting. In this framework we have applied four different methods include exponential

smoothing, ARMA, artificial neural network and hybrid ANN-ARMA methodology, then by applying multi-criteria decision analysis, the most effective method can be selected. The accurate forecasting can result in reductions in total waiting time, idle time, and overtime costs. We illustrate this with results of a case study which conducted by real world data at John D. Dingell Detroit VA Medical Center.

AUTOBIOGRAPHICAL STATEMENT

AZAD SADR HAGHIGHI

EDUCATION

MASTER OF SCIENCE	**Industrial Engineering**
Wayne State University, Detroit, MI	2012-2015
BACHELOR OF SCIENCE	**Industrial Engineering**
Kar University, Qazvin, Iran	2005-2009

EXPERIENCE

JOHN D. DINGELL VA MEDICAL CENTER	Detroit, MI
Research Assistant/Student Trainee	Oct 2013 – Jan 2015