Data Analysis For Dialysis Patients

A Thesis
Submitted in partial fulfillment of
the requirements for the degree of
Master of Technology
by

Manjunath S. Vhatkar
(Roll no: 173190026)

Supervisors:
Prof. Narayan Rangaraj
and
Prof. P. Balamurugan

Industrial Engineering and Operations Research
Indian Institute of Technology Bombay
Mumbai 400076 (India)
26 June 2019

Acceptance Certificate

Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay

The thesis entitled "Data Analysis For Dialysis Patients" submitted by Manjunath S. Vhatkar (Roll no: 173190026) may be accepted for being evaluated.

Date: 25 June 2019

Prof. Narayan Rangaraj

P. Balamurugan

Prof. P. Balamurugan

Approval Sheet

This thesis entitled "Data Analysis For Dialysis Patients" by Manjunath S. Vhatkar is approved for the degree of Master of Technology.

25/6/2019

A Kavitha / 25/6/2019

Examiners

P. Balamurugan

25/6/2019

Supervisor (s)

Chairman

Date: 25/6/2019

Place: IIT, Bombay

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this report. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

MVhatkar

Manjunath S. Vhatkar
(Roll no: 173190026)

Date: 25 June 2019

Acknowledgements

I feel great pleasure and privilege to express my deep sense of gratitude and thankfulness towards my supervisors, **Prof. Narayan Rangaraj** and **Prof. P. Balamurugan** for valuable guidance, supervision, and continuous encouragement throughout the project. I am forever grateful for their kindness and contributions, not only towards my project but towards my professional growth as well.

I am very thankful to all the faculties of the Industrial Engineering and Operations Research program who helped me to explore more about IEOR.

I also want to acknowledge Dr. Vishwanath Billa, Dr. Deepa, and others supporting staff of Apex Kidney Care for providing us data of dialysis patients, spending many hours in last two semesters regarding this projects and thanks for coming to IIT Bombay for meetings.

I also acknowledge my colleagues and the research scholars especially Mr. Sufiyan Adhikari, Mr. Jaswant Singh, Mr. Ashutosh Kushwaha and Mr. Gaurav Amrutkar for their valuable suggestions and helpful discussions.

I would also like to extend my regards to my family and friends for always trusting me with my capabilities, helping me with my self-confidence to achieve my goals and going out of the ways to help me.

At last, I would like to thank everyone who helped me directly or indirectly in the completion of the work.

Manjunath S. Vhatkar
IIT Bombay
27 June 2019

Abstract

The people with damaged or failed kidney may have difficulty in eliminating the toxic waste and unwanted water waste like urea, creatinine, etc, from the human blood. The artificial way of carrying out this process is called Dialysis. The process of removing the toxic wastes and unwanted water waste by circulating blood to an external filter by taking out the blood from human body is called "Hemodialysis" and if this toxic wastes and unwanted water waste are removed from the blood inside the body using the peritoneum in an abdominal cavity called "Peritoneal Dialysis".

In the hemodialysis process, the blood from the body is taken outside and passed through the external filter called dialyzer, where the blood impurities like urea, creatinine, etc, are removed, and the goal 65 percent clearance of urea are set. So to meet the of 65 percent of clearance of urea the parameter like BSP, heart rate (Hrate), blood flow(BpFlow), dialysate flow (Dflow), etc are taken in consideration at the time dialysis.

Hemodialysis patients have a higher risk of mortality than the general population due to fluctuations in the BSP, heart rate (Hrate),etc. which may lead to cardiovascular morbidity and mortality in patients after dialysis. Hence by using some Statistical technique and Machine learning technique on these parameters of the hemodialysis process, we analyze in this report the trend of these parameters over time and propose to use this analysis to predict the health of patients after dialysis.

The practical application of this techniques will include a continuous monitoring process for alerting clinicians when there is deterioration in a patient's status that signals some clinical outcome, thus providing an opportunity for intervention and proper care of the patients are taken. We believe that this would lead to better care for the patients after dialysis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This project aims to develop some data analysis and forecasting tools to help clinical technicians and doctors who conduct dialysis procedures on patients who suffers from renal problems, i.e where the functioning of kidney is not normal.

1.1 What is Dialysis ?

Dialysis is the process of removing excess unwanted water waste, toxins, solutes, etc, from the blood in patients whose native kidneys are damaged or which failed to perform these functions in natural ways [11].
Following are the two kinds of dialysis processess:

- Hemodialysis - Process of removing toxic wastes and unwanted water waste by circulating blood to an external filter by taking out the blood from human body [5].
- Peritoneal Dialysis - The toxic wastes and unwanted water waste are removed from the blood inside the body using the peritoneum in an abdominal cavity [5].

1.2 Effects of dialysis on patients

Hemodialysis patients have 10 to 20 times higher mortality risk than that of the general population. The role of BSP in cardiovascular morbidity and mortality in hemodialysis patients has not been fully explained. There appears to be an increased risk for individuals with either abnormally high or low BSP. [17].

In the dialysis process clinicians are not able to puncture the arteries, only veins are available so they convert veins to perform the function of arteries physically and the process of dialysis is carried out. Arterial end carries blood for cleaning from body to the dialyzer. Venous end carries cleaned blood from dialyzer to body. In the process of dialysis, the filter is run for 4 hours and blood is purified and sent back to the body through the venous end. During this 4-hour purification of blood, the urea level in blood before dialysis and after dialysis is measured and the goal is set to 65 percent of clearance of urea from the blood and water clearance depends on the patient's weight and water weight.

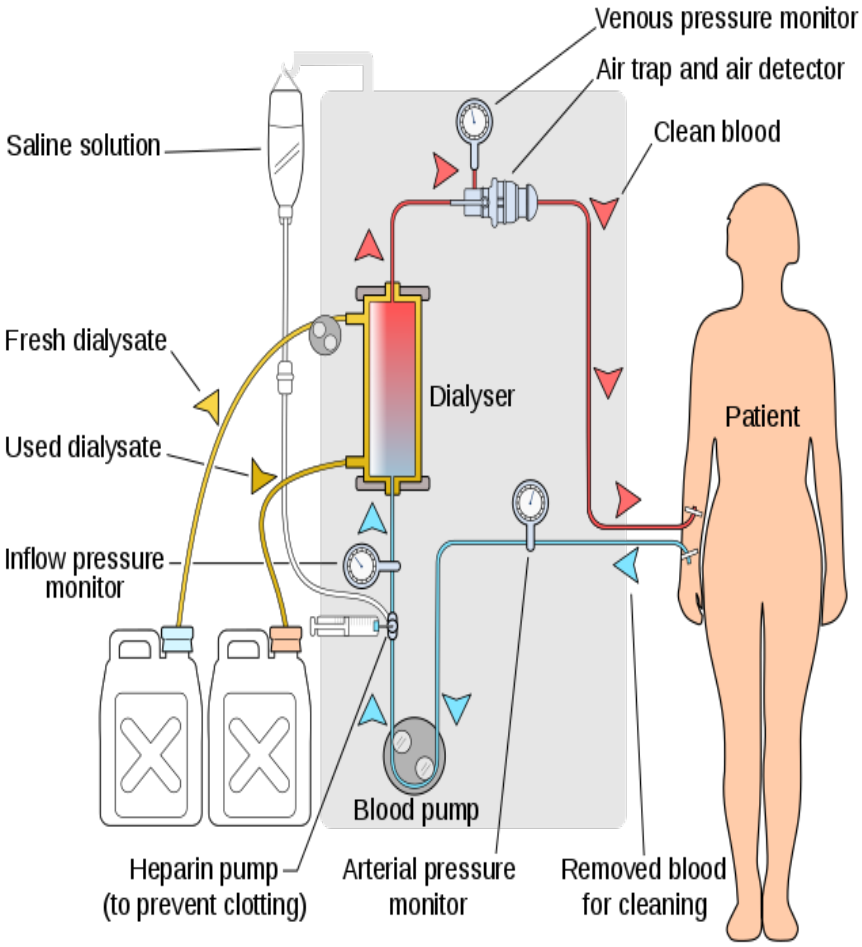


Figure 1.1: Outline of Dialysis process [21]

In figure (1.1), we show the outline of dialysis process. Impure blood from the patient's body is removed for cleaning through the arterial end and passed through the blood pump to increase the pressure and after that blood is sent to the dialyzer to remove impurities like urea, creatinine, etc. In dialyzer there are dialysate solution which are used to remove the blood impurities from the blood and the goal is set to 65 percent of clearance of urea from the blood. After the filtration of blood in dialyzer, the purified blood is again sent to the patient's body through the venous end. So, to meet the target of 65 percent of clearance of urea from the blood, parameters like BSP, heart rate (Hrate), blood flow(BpFlow), dialysate flow (Dflow), etc are taken in consideration at the time of dialysis [17].

The fluctuations in the BSP, heart rate, and other parameters measured during dialysis may lead to health deterioration which may futher cause mortality in patients. So we believe that parameters like BSP, heart rate, blood flow, etc. measured during dialysis may be used as predictors of the mortality in hemodialysis. From doctors we got to know that BSP is measure parameter to cause of mortality in patients. So this parameter called "BSP" is taken for further analysis .

1.3 Objectives of the thesis

By using time series forecasting on these parameters of the dialysis process, we will analyze the trends over time on the various parameters. We propose to predict the health of the patients after dialysis and by clustering analysis we are able to cluster the sessions of the patients based on their increasing BSP and decreasing BSP to get an idea of the impact of different drugs on the increase/decrease of BSP during dialysis.

The practical application of these techniques which we have applied for the trend analysis of the BSP will include a continuous monitoring process for alerting clinicians when there is drastic changes the parameters like BSP, heart rate, etc, thus providing an opportunity for intervention and proper care of the patients. We believe that this would lead to better care for the patients after dialysis.

Chapter 2

Statistical Time Series Forecasting

2.1 Time series forecasting

Time Series Points is the set of observation that the random variables takes at various time points. this type of data may collected at fixed time intervals such as hourly, weekly, daily,etc. Time series forecasting is a statistical technique that deals with time series or trend analysis of parameters [14].
The data for Time series forecasting can belong to the following categories:

- **Uni-variate data**: A uni-variate data points refer to a time series that consists of one observation recorded over a regular time interval.
  Example: Monthly returns data of stock.

- **Cross Sectional data**: Such data are collected by examine many cases (such as stock,demand, individual, or regions) at the same points of time or during the same time period.

- **Pooled data**: A combination of cross-sectional data and uni-variate time-series data.

Time series forecasting can be used in many applications such as sales forecast, economic forecast, stock market forecast, budgetary analysis, weather forecast, census forecast and inventory forecast.

2.2 Components of Time-Series forecasting

The model in time-series analysis is classify into seasonality, cyclic,random, trend, etc components [14].

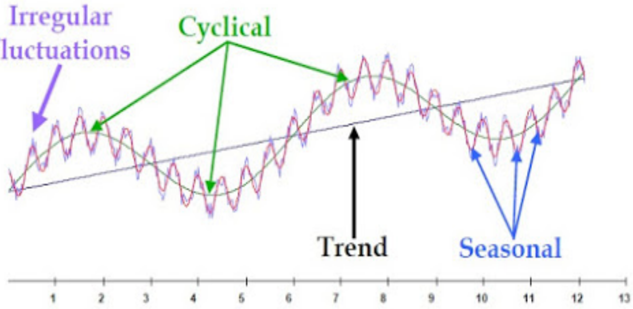


Figure 2.1: Components of Time Series Forecasting [18]

- Trend: The trend is definite movement of time-series for a large time period without much random effects. Example of trends includes population growth, demand growth, and general trending economic changes. See figure 2.2 which shows a growth trend over time.

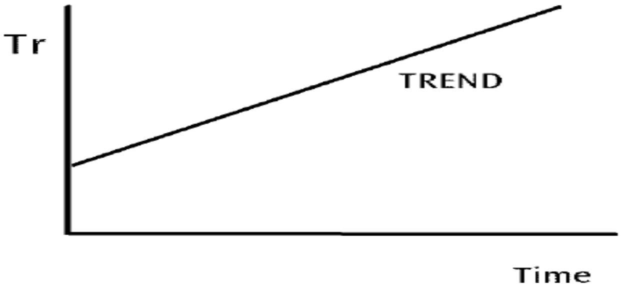


Figure 2.2: Trend [4]

- Seasonality: The seasonality is the characteristic of a time-series data in that the data experience predictable and regular change that repeats every year. Any pattern or predictable change in a time-series that repeats over a one year time period can be said to be seasonal.

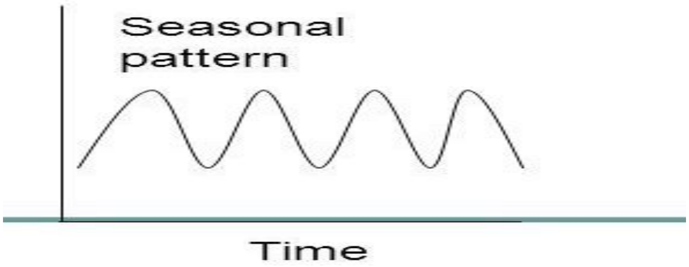


Figure 2.3: Seasonality [4]

- Cyclic: This type of pattern exists when time-series has ups and downs which are not of any fixed time period.

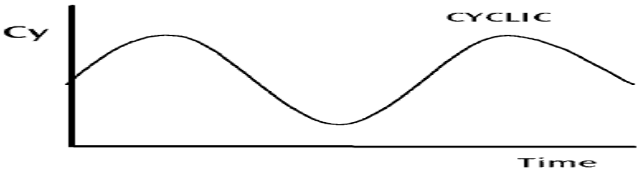


Figure 2.4: Cyclic [4]

- Random: The random component results when time series data has short term fluctuations which are neither systematic nor predictable.

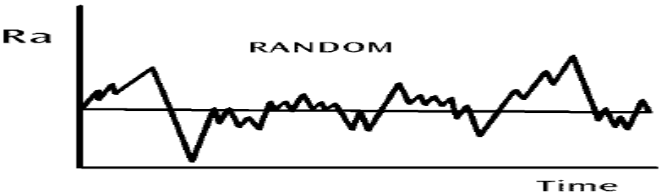


Figure 2.5: Random [4]

2.3 Stationarity

Stationarity of time-series : A stationarity is a process that has constant mean, constant variance, constant auto-correlation structure across time period. The series which looks flat,without random effects, without any trend, without any seasonality, constant variance over time, a constant auto-correlation formation across time and no periodic variations over time [22].

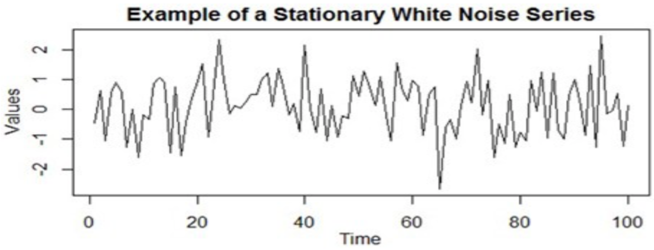


Figure 2.6: Stationarity Time Series [15]

Non-Stationary Time Series: The time-series in which data show random effects, seasonality effects, periodic variations over time. Mean, auto-correlation and variance change across time in non-stationary time series [22].

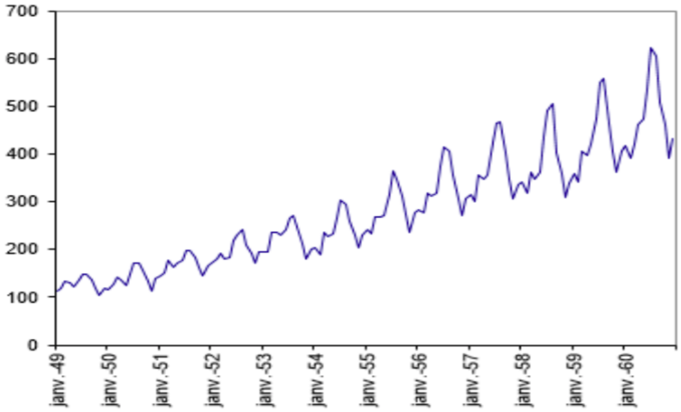


Figure 2.7: Non-Stationary Time Series [10]

2.3.1 Transforming Models

Most of the time, in real-life the time-series sets is non-stationary. We have to make data stationary in order to get any useful forecasting predictions from it. A model can sometimes be stationarised by using some mathematical transformation which makes the model relatively easy to forecast. The mathematical transformations are then reversed so that new model forecast the performance of the original time-series models. Transformations might include [22]:

- Difference the data: Differencing the data that has one less point than the original data. Example: given a series X_t you can generate a new series $\nabla Y_t = X_t - X_{t-1}$
- Take the log or square root (usually works when time series has non-constant variance)

2.3.2 Check Stationarity

Before going to further analysis of the time series we must have to perform the test to check the Stationarity. However, the classical methods are not yet well developed such that it can take cases where unknown nonlinear relationships to be determined. Hence before applying the time-series models like AR/MA/ARMA/ARIMA for forecasting, the time-series must be stationary and checks for stationarity is given as below [22].

- **Look at Plots**: For stationarity, we check the plot and checks if they do not must show any kind of trend and seasonality effects
- **Statistical Tests** : We use this type of tests to check if expectation and variance of stationarary time-series are met or violated.

Dickey-Fuller Test: Dickey-Fuller test for the unit root ($\phi = 1$) is present in the time-series.
Let us consider a time series

$$Y_t = \delta + \phi Y_{t-1} + \epsilon_t \tag{2.1}$$

where

$$\epsilon_i \sim N(0, \sigma^2) \qquad \text{and} \tag{2.2}$$

$$cov(\epsilon_i, \epsilon_j) = 0 \qquad \text{for} \qquad i \neq j \tag{2.3}$$

If $\delta = 0$, then the time-series is said to be without drift, while if $\delta \neq 0$, then the time-series is with drift.

Regression test for the Random Walk:

$$H_0 : \phi = 1 \tag{2.4}$$

$$H_1 : \phi < 1 \tag{2.5}$$

If $\phi = 1$ shows that the process is the random walk and if $\phi < 1$ show that the process is not a random walk.

By subtracting Y_{t-1} from equation (3.1) on both the side we get

$$Y_t - Y_{t-1} = \delta + \phi Y_{t-1} - Y_{t-1} + \epsilon_t \tag{2.6}$$

$$\Delta Y_t = \delta - (1 - \phi) Y_{t-1} + \epsilon_t \tag{2.7}$$

Let $(\phi - 1) = \beta$

$$\Delta Y_t = \delta + \beta Y_{t-1} + \epsilon_t \tag{2.8}$$

If $\beta = 0$ the system has the Unit Root, i.e. $\phi < 1$. Hence we can modify the hypothesis as below:

$$H_0 : \beta = 0 \tag{2.9}$$

$$H_1 : \beta < 0. \tag{2.10}$$

This Hypothesis testing is called the Dickey-Fuller Test of Stationarity and if we add more lagged changed on the right hand side its Augmented Dickey-Fuller test

- **Null Hypothesis** (H_0): If Null Hypothesis is accepted, then we say that the time-series has a unit root ($\phi = 1$) and the process is non-stationary.
- **Alternate Hypothesis** (H_1): If Null hypothesis is rejected then,we can say that the time-series does not have any unit root ($\phi \neq 1$) and that means the process is stationary.

2.4 Autocorrelation Function

Co-variance: Co-variance measures the linear dependence between two random variables.

Let X and Y be two random variables and co-variance between these two variables is given by:

$$Cov[X, Y] = E[(X - \mu_x)(Y - \mu_y)] \tag{2.11}$$

Where μ_x and μ_y are the means of random variables X and Y respectively [14].

Auto-covariance: The function auto-covariance gives the covariance of the process with itself at pairs of time points and it is denoted by γ_k and k is the number of lags.

Let collection of random variables $Y = Y_1, Y_2, Y_3, ...$ and $Y_t \sim$ distribution(μ_t, σ_t^2)

Thus we have

$$\gamma(s, t) = cov(Y_s, Y_t) \tag{2.12}$$

$$= E[(Y_s - \mu_s)(Y_t - \mu_t)] \tag{2.13}$$

and

$$\gamma(t, t) = cov(Y_t, Y_t) \tag{2.14}$$

$$= E[(Y_t - \mu_t)^2] \tag{2.15}$$

We thus have

$$\gamma_k = \gamma(t, t + k) = cov(Y_t, Y_{t+k}) \tag{2.16}$$

$$= E[(Y_t - \mu_t)(Y_{t+k} - \mu_{t+k})] \tag{2.17}$$

In stationary time series with zero means,

$$\gamma_k = E[Y_t.Y_{t+k}] \tag{2.18}$$

The **Auto-covariance Coefficients** C_k at lag k is given as

$$C_k = \frac{\sum_{t=1}^{N-k}(Y_t - \hat{y})(Y_{t+k} - \hat{y})}{N} \tag{2.19}$$

where

$$\hat{y} = \frac{\sum_{t=1}^{N}(Y_t)}{N} \tag{2.20}$$

Auto-correlation: The auto-correlation measures the correlation between the time points of a series and other time points from the same series separated by a given interval.

The **Auto-correlation function** (ACF) at lag k, denoted ρ_k, of a stationary process, is defined as $\rho_k = \gamma_k/\gamma_0$ where $\gamma_k = cov(Y_t, Y_{t+k})$ for any t [14].
The value of the auto-correlation function(ρ_k) varies from -1 to 1.

The **Auto-correlation Coefficient** r_k at lag k is given as

$$r_k = \frac{C_k}{C_0} \tag{2.21}$$

$$= \frac{\sum_{t=1}^{N-k}(Y_t - \hat{y})(Y_{t+k} - \hat{y})}{\sum_{t=1}^{N-k}(Y_t - \hat{y})^2} \tag{2.22}$$

It always starts at 1 since $r_0 = C_0/C_0 = 1$ and the plots of the auto-correlation coefficient at different lags is called correlogram [14].

2.5 Partial Auto-correlation Function

Partial auto-correlations gives the relationship between one time point on another after removing the effect of other time points in between.
For example, the partial auto-correlation of order 2 measures the linear dependence effects of Y_{t-2} on Y_t after removing the effect of Y_{t-1} on both Y_t and Y_{t-2}.
For a time-series, the partial auto-correlation between Y_t and Y_{t-p} is defined as the conditional correlation between Y_t and Y_{t-p} , conditional on $Y_{t-p+1}, ..., Y_{t-1}$, the set of information's that come between the time point t and $t - p$.
The first order partial auto-correlation will be defined to equal the first order auto-correlation.
The second order (lag) partial auto-correlation is

$$PACF_2 = \frac{\text{cov}(Y_t, Y_{t-2}|Y_{t-1})}{\sqrt{\text{var}(Y_t|Y_{t-1}).\text{var}(Y_{t-2}|Y_{t-1})}} \tag{2.23}$$

AR processes have Auto-correlation Function (ACF) values that converge to zero as the lag increases, while MA processes have Partial Auto-correlation Function (PACF) values that converge to zero as the lag increases. The order of the process may not be obvious using this approach. AR(p) processes have Partial Auto-correlation Function (PACF) values that are small (ie almost zero) for lags $> p$, while MA(q) processes have Auto-correlation Function (ACF) values that are small for lags $> q$. If the ACF and PACF values do not converge to zero, then differencing may be needed [22].

2.6 White Noise

White Noise indicates a set of a random variable with constant mean, constant variance and zero autocorrelation at all lags.

- If this set of random variables are independent and identically distributed and has mean equal to zero and constant variance then the process is called as White Noise.

Let ϵ_t denote such a time series, then we have

$$E[\epsilon_t] = 0 \tag{2.24}$$

And

$$V[\epsilon_t] = \sigma^2 \tag{2.25}$$

In time-series forecasting this white noise concept is very much important for two main reasons:

- Predictability: Time-series is white noise then, it is random effect. So we cannot able to model it and make useful predictions from it.

The white noise is a **Stationary time-series** since the mean and variance of the series is constant and zero has autocorrelation at all lags [22].
Let

$$Y_t \sim N(0, \sigma^2) \tag{2.26}$$

Then

$$\gamma(t_1, t_2) = cov[Y_{t_1}, Y_{t_2}] \tag{2.27}$$

where

$$\gamma(t_1, t_2) = 0 \qquad \text{for} \qquad t_1 \neq t_2 \tag{2.28}$$

$$\gamma(t_1, t_2) = \sigma^2 \qquad \text{for} \qquad t_1 = t_2 \tag{2.29}$$

2.7 Random Walk

A random walk time series takes the form

$$Y_t = \delta + Y_{t-1} + \epsilon_t \tag{2.30}$$

where

$$\epsilon_i \sim N(0, \sigma^2) \tag{2.31}$$

If $\delta = 0$, then the random walk is said to be without drift, while if $\delta \neq 0$, then the random walk is with drift (i.e. with drift equal to δ) [22].
To find the Expectation and the variance of the random walk without drift i.e $\delta = 0$.
We have

$$Y_t = Y_{t-1} + \epsilon_t \tag{2.32}$$

Let $Y_0 = 0$

$$\begin{align} Y_1 &= 0 + \epsilon_1 \tag{2.33}\\ Y_2 &= Y_1 + \epsilon_2 = \epsilon_1 + \epsilon_2 \tag{2.34}\\ Y_3 &= Y_2 + \epsilon_3 = \epsilon_1 + \epsilon_2 + \epsilon_3 \tag{2.35}\\ &.... \tag{2.36}\\ Y_t &= \sum_{i=1}^{t} \epsilon_t \tag{2.37} \end{align}$$

The Expectation and the variance of the random walk is given as

$$\begin{align} E[Y_t] &= E\Big[\sum_{i=1}^{t} \epsilon_t\Big] \tag{2.38}\\ &= t\mu \tag{2.39}\\ Var[Y_t] &= Var\Big[\sum_{i=1}^{t} \epsilon_t\Big] \tag{2.40}\\ &= t\sigma^2 \tag{2.41} \end{align}$$

Since above equation (2.39) and (2.41) shows that the expectation (Mean) and the variance of the Random walk is not constant it varies with the time hence the Random walk is non-stationary time series

2.8 Statistical Models

2.8.1 Auto-regressive process of order p (AR(p))

An Auto-regressive (AR) model is one where Y_t at time t depends only on its own past values $Y_{t-1}, Y_{t-2}, ...$ etc [22].

$$Y_t = f(Y_{t-1}, Y_{t-2}, Y_{t-3}, Y_{t-4}, ...) \tag{2.42}$$

An AR model of order is given as:

$$Y_t = \mu + \epsilon_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3}, ... + \phi_p Y_{t-p} \tag{2.43}$$

An AR model of order 1 is given as:

$$Y_t = \mu + \epsilon_t + \phi_1 Y_{t-1} \tag{2.44}$$

Since eq (2.44) contains only one lagged value on right-hand side this is called AR model of order 1 or AR(1) and ϕ_1 is the parameter of AR(1), if $\phi_1 = 1$ then series becomes a random walk and if $\phi_1 = 0$ then series becomes white noise.
Let us discuss some more properties of Auto-regressive model.

- The mean of the Y_t in a stationary AR(p) process is computed as follows:
  Since the process is stationary, for any k, $E[Y_t] = E[Y_{t-k}]$, a value which we will denote μ. Since $E[\epsilon_t] = 0, E[\phi_0] = \phi_0$ and

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \phi_3 Y_{t-3}, ... + \phi_p Y_{t-p} + \epsilon_t \tag{2.45}$$

  it follows that

$$\begin{aligned}
\mu &= E[Y_t] = E[\phi_0] + \phi_1 E[Y_{t-1}] + ... + \phi_p E[Y_{t-p}] + E[\epsilon_t] \\
\mu &= \phi_0 + \phi_1\mu + \phi_2\mu + \phi_3\mu + ... + \phi_p\mu \\
\mu &= \frac{\phi_0}{1 - \sum_{i=1}^{p} \phi_i}
\end{aligned} \tag{2.46}$$

- The Variance of the Y_t in a stationary AR(1) process is computed as follows
  Since the Y_t and ϵ_t are independent, by basic properties of variance,

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \epsilon_t \tag{2.47}$$

  it follows that

$$\begin{aligned}
var[Y_t] &= var[\phi_0] + \phi_1^2 var[Y_{t-1}] + 2\phi_1 cov[\phi_0 Y_{t-1}] + 2cov[\phi_0 \epsilon_t] + 2\phi_1 cov(\epsilon_t Y_{t-1}) + var[\epsilon_t] \\
var[Y_t] &= 0 + \phi_1^2 var(Y_{t-1}) + \sigma^2 + 0 + 0 + 0 \\
var[Y_t] &= \phi_1^2 var(Y_{t-1}) + \sigma^2
\end{aligned} \tag{2.48}$$

Since the process is stationary, $var[Y_t] = var[Y_{t-1}]$, and so

$$var[Y_t] = \phi_1^2 var(Y_t) + \sigma^2 \tag{2.49}$$

$$var[Y_t] = \frac{\sigma^2}{1 - \phi_1^2} \tag{2.50}$$

- The lag k auto-correlation in a stationary AR(1) process is computed below If we assume that $\phi_0 = 0$, an AR(1) process and using for $var(Y_t) = cov(Y_t, Y_t)$, we have

$$\begin{aligned}\gamma_k &= cov(Y_t, Y_{t-k})\\ &= E[Y_t.Y_{t-k}]\\ &= E[(\phi_1 Y_{t-1} + \epsilon_t)Y_{t-k}]\\ &= \phi_1 E[Y_{t-1}Y_{t-k}] + E[\epsilon_t Y_{t-k}]\\ &= \phi_1 \gamma_{k-1} \end{aligned} \tag{2.51}$$

By induction on k it is easy to show that

$$\gamma_k = \phi_1^k \gamma_0 \tag{2.52}$$

And hence

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \phi_1^k. \tag{2.53}$$

Estimation of σ_ϵ^2:

Consider AR(2) process,

$$Y_t = \epsilon_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} \tag{2.54}$$

Taking variance on both sides, we get

$$var[Y_t] = \sigma_\epsilon^2 + \phi_1^2 var[Y_{t-1}] + \phi_2^2 var[Y_{t-2}] + 2\phi_1\phi_2 cov[Y_{t-1}Y_{t-2}] \tag{2.55}$$

$$\sigma_\epsilon^2 = var[Y_t] - \phi_1^2 var[Y_{t-1}] - \phi_2^2 var[Y_{t-2}] - 2\phi_1\phi_2 cov[Y_{t-1}Y_{t-2}] \tag{2.56}$$

$$= \sigma^2 - \sigma^2\phi_1^2 - \phi_2^2\sigma^2 - 2\phi_1\phi_2\gamma_1 \tag{2.57}$$

Since $\gamma_0 = \sigma^2$

$$\sigma_\epsilon^2 = \gamma_0(1 - \phi^2 - \phi_2^2 - 2\phi_1\phi_2\frac{\gamma_1}{\gamma_0}) \tag{2.58}$$

$$= \gamma_0(1 - \phi^2 - \phi_2^2 - 2\phi_1\phi_2\rho_1) \tag{2.59}$$

Since $\rho_1 = \phi_1 + \rho_1\phi_2$ and $\rho_2 = \phi_1\rho_1 + \phi_2$
Therefore

$$(1 - \phi^2 - \phi_2^2 - 2\phi_1\phi_2\rho_1) = 1 - \phi_1(\phi_1 + \rho_1\phi_2) - \phi_2(\phi_1\rho_1 + \phi_2) \tag{2.60}$$

$$= 1 - \phi_1\rho_1 - \phi_2\rho_2 \tag{2.61}$$

Hence

$$\sigma_\epsilon^2 = \gamma_0(1 - \phi_1\rho_1 - \phi_2\rho_2) \tag{2.62}$$

2.8.2 Moving Average process of order q (MA(q))

The moving average model (MA(q)) is one when Y_t depends only on the random error terms which follows the white noise [22].

$$Y_t = f(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, ...)$$

Where

$$\epsilon_t \sim iid(0, \sigma^2) \tag{2.63}$$

The moving average MA(q) process of the time series is given as

$$Y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \theta_3\epsilon_{t-3}, ... + \theta_q\epsilon_{t-q} \tag{2.64}$$

In MA(q) model q denotes the lagged error term on right-hand side. If $q = 1$ then MA model of order 1 and θ is the parameter of MA(q) model, If $\theta = 0$ then MA process is White noise.
Using the back shift operator (BSO), we can express a zero mean ($\mu = 0$) MA(q) process as

$$Y_t = \theta(B)\epsilon_t \tag{2.65}$$

where

$$\theta(B) = 1 + \theta_1(B) + \theta_2(B^2)..... + \theta_q(B^q) \tag{2.66}$$

Let us discuss some properties of moving average process of time series

- The mean of an MA(q) process is μ
- The variance of an MA(q) process is

$$var(Y_t) = \sigma^2(1 + \theta_1^2 + \theta_2^2 + ... + \theta_q^2) \tag{2.67}$$

- The auto-covariance of an MA(q) process with a zero mean ($\mu = 0$) at different lag k is given as

$$\gamma_k = cov[Y_t, Y_{t+k}] = E[Y_t.Y_{t+k}] - E[Y_t]E[Y_{t+k}] \tag{2.68}$$

Since $E[Y_t] = 0$ and $E[Y_{t+k}] = 0$

$$\gamma_k = cov[Y_t, Y_{t+k}] = E[Y_t.Y_{t+k}] \tag{2.69}$$

$$cov[Y_t, Y_{t+k}] = E\Big[[\epsilon_t + \theta_1\epsilon_{t-1}, ... + \theta_q\epsilon_{t-q}].[\epsilon_{t+k} + \theta_1\epsilon_{t+k-1} .... + \theta_q\epsilon_{t+k-q}]\Big] \tag{2.70}$$

$$cov[Y_t, Y_{t+k}] = E[\epsilon_t\epsilon_{t+k} + \epsilon_t\theta_1\epsilon_{t+k-1} + .. + \epsilon_t\theta_q\epsilon_{t+k-q} + \theta_1\epsilon_{t-1}\epsilon_{t+k} + \theta_1\epsilon_{t-1}\theta_1\epsilon_{t+k-1}...] \tag{2.71}$$

For $k = 0$

$$cov[Y_t, Y_t] = E[\epsilon_t^2 + \epsilon_t\theta_1\epsilon_{t-1} + .. + \epsilon_t\theta_q\epsilon_{t-q} + \theta_1\epsilon_{t-1}\epsilon_t + \theta_1^2\epsilon_{t-1}^2...] \tag{2.72}$$

When

$$Y_t \sim N(0, \sigma^2) \tag{2.73}$$

$$\gamma(t_1, t_2) = cov[Y_{t_1}, Y_{t_2}] \tag{2.74}$$

$$\gamma(t_1, t_2) = 0 \implies t_1 \neq t_2 \tag{2.75}$$

$$\gamma(t_1, t_2) = \sigma^2 \implies t_1 = t_2 \tag{2.76}$$

Hence

$$cov[Y_t, Y_t] = [\sigma^2 + \theta_1^2\sigma^2 + \theta_2^2\sigma^2 + ... + \theta_q^2\sigma^2] \tag{2.77}$$

$$= \sigma^2[1 + \theta_1^2 + \theta_2^2 + ... + \theta_q^2] \tag{2.78}$$

Similarly for $k = 1$

$$cov[Y_t, Y_{t+1}] = \sigma^2[\theta_1 + \theta_1\theta_2 + \theta_2\theta_3 + ... + \theta_{q-1}\theta_q] \tag{2.79}$$

for $k = q$

$$cov[Y_t, Y_{t+1}] = \sigma^2\theta_q \tag{2.80}$$

for $k < q$

$$\gamma_k = \sigma^2[\theta_k + \theta_1\theta_{1+k} + \theta_2\theta_{2+k} + ... + \theta_{q-k}\theta_q] \tag{2.81}$$

- The autocorrelation function of an MA(q) process is
for $q = 1$

$$\rho_1 = \frac{\gamma_1}{\gamma_0} \tag{2.82}$$

$$= \frac{\theta_1}{(1 + \theta_1^2)} \tag{2.83}$$

for $q = 2$

$$\rho_2 = \frac{\gamma_2}{\gamma_0} \tag{2.84}$$

$$= \frac{\theta_2}{(1 + \theta_1^2 + \theta_2^2)} \tag{2.85}$$

The properties discussed show that the Moving Average time-series has a constant mean and constant variance. Hence this time-series is stationary.

2.8.3 Auto-regressive Moving Average (ARMA(p, q))

The process where the time-series may be created from the mixture of two process like AR(p) and MA(q) then model referred to ARMA(p, q) [22].

The general form of such a time series model, which depends on p of its own past value and q of past value of error term is called as auto-regressive moving average (ARMA(p, q)).

$$Y_t = f(\epsilon_1, \epsilon_2...., Y_{t-1}, Y_{t-2}........) \tag{2.86}$$

$$Y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + ... + \theta_q\epsilon_{t-q} + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} \tag{2.87}$$

The above eq (2.87) is called general ARMA(p, q) with p order of AR model and q order of MA model.

For $\mu = 0$ we can write above equation ARMA(p, q).

$$Y_t - \phi_1 Y_{t-1} - .... - \phi_p Y_{t-p} = \epsilon_t + \theta_1\epsilon_{t-1} + ... + \theta_q\epsilon_{t-q} \tag{2.88}$$

$$Y_t - \phi_1 Y_t B - .... - \phi_p Y_t B^p = \epsilon_t + \theta_1\epsilon_t B + ... + \theta_q\epsilon_t B^q \tag{2.89}$$

where B is the Back Shift Operator (BSO)

Thus,

$$(1 - \phi_1 B - .... - \phi_p B^p)Y_t = (1 + \theta_1 B + ... + \theta_q B^q)\epsilon_t \tag{2.90}$$

$$\phi(B)Y_t = \theta(B)\epsilon_t \tag{2.91}$$

Therefore,

$$Y_t = \frac{\theta(B)}{\phi(B)}\epsilon_t = \psi(B)\epsilon_t \tag{2.92}$$

where $\psi(B) = \frac{\theta(B)}{\phi(B)}$

$$\epsilon_t = \frac{\phi(B)}{\theta(B)}Y_t = \pi(B)Y_t \tag{2.93}$$

where $\pi(B) = \frac{\phi(B)}{\theta(B)}$

2.8.4 Autoregressive Integrated Moving Average (ARIMA(p, d, q))

ARIMA model is a generalization of an Auto-regressive moving average (ARMA) process. Both the model is fitted to time-series data either to better understand or to predict the future points in the series (Forecasting) [22].

ARIMA Models are applied when there non-stationary time-series, where an initial differencing step (corresponding to the "integrated " part of the model) can be applied one or more times to eliminate non-stationarity.

An ARIMA(p, d, q) process with parameter p = order of AR model, q = order of MA model and d = degree of differencing.

Consider an ARMA process.

$$Y_t = \frac{\theta(B)}{\phi(B)}\epsilon_t \tag{2.94}$$

where

$$\phi(B) = 1 - \phi_1 B' - \phi_2 B^2 ...... - \phi_p B^p \tag{2.95}$$

$$\theta(B) = 1 + \theta_1 B' + \theta_2 B^2 ...... + \theta_q B^q \tag{2.96}$$

The real-life datasets are non-stationary and there may be some systematic change in the Trends, Seasonality, etc. So before going to the analysis we have to remove the trend, seasonality by difference operator $\nabla = (1 - B)$

Hence, the ARIMA process is given as:

$$\phi(B)\nabla^d Y_t = \theta(B)\epsilon_t \tag{2.97}$$

or

$$\phi(B)(1 - B)^d Y_t = \theta(B)\epsilon_t \tag{2.98}$$

2.8.5 Comparing ARIMA models

So which model of $ARIMA(p, d, q)$ performs better and gives us the best fit to the time-series forecast and predicts the future value. There are information criteria which can be used to choose the best working model. These information criteria adjust goodness of the fit based on the number of parameters.
The most popular information criteria are:

1. **Akaike Information Criteria**: This information criteria matches the status of a set of AR/MA/ARMA/ARIMA models to each other. The Akaike Information Criteria will take each model and put them in order of greatest to worst. The best AR/MA/ARMA/ARIMA models will be the one best model with lowest AIC that does not under-fits and over-fits [1].
   Akaike Information Criterion is usually calculated with the software. The basic formula is defined as:

   $$AIC = -2(\text{Log-Likelihood}) + 2K \tag{2.99}$$

   Where
   K = Model parameters in AR/MA/ARMA/ARIMA models.
   Log-Likelihood = A measure of model fit. The higher the log-likelihood number,the best the fit.

2. **Bayesain Information Criteria**: The Bayesian information criterion (BIC) is a criterion for model selection among a all set of AR/MA/ARMA/ARIMA models. It is based, on the likelihood function and it is exactly related to AIC [1].
   The formula for BIC is

   $$BIC = -2 * log_e(L) + K * log_e(n) \tag{2.100}$$

   Where
   K = Model parameters in AR/MA/ARMA/ARIMA models.
   n = The data points in observed data
   L = the maximized value of the likelihood function for the estimated model

Generally, the process with the lower AIC or BIC value should be selected as the best fit to the time series forecasting and predict the best future value.
So, in this chapter we have studied the have the what are the statistical model and how to applied them on time series data and further we are will be discussing the how machine learning models are applied to time series and what are the models applied for time series forecasting.

Chapter 3

Machine Learning Models

3.1 Recurrent Neural Network(RNN)

RNN is a kind of neural network. That enables the modeling of time-dependent and sequential data. Tasks such as stock market prediction, machine translation and text generation are possible using RNNs [8].

3.1.1 Working of RNN

RNN has a memory which obtains knowledge about whatever has been estimated so far. RNNs can make use of this knowledge in arbitrarily long sequence but, in practice they limited to looking back only a few steps.

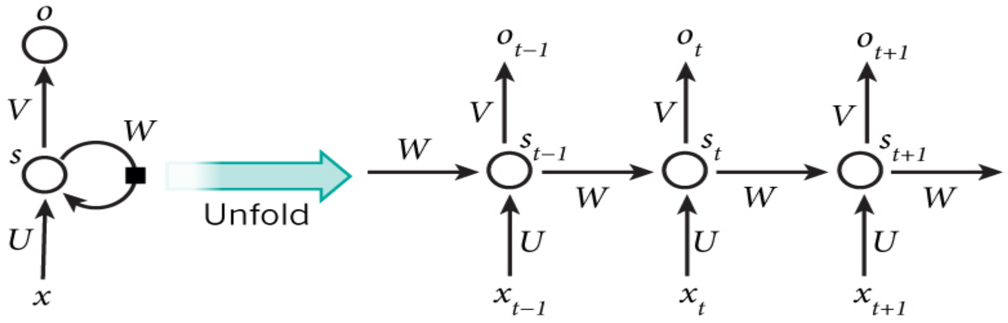


Figure 3.1: Recurrent Neural Network [6]

Typical RNN is shown in figure 3.1 wHere we have, X_t = the input at time step t, s_t = the hidden state at time step t. It is usually called the memory of the network. s_t is calculated based on the previous hidden state and the input at the current step:

$$s_t = f(U * X_t + W * s_{t-1}). \tag{3.1}$$

The function f usually is non-linearity such as tanh or relu.
s_{-1} which is required to calculate the 1st hidden state is typically initialized to zero.
o_t = the output at step t, given by:

$$o_t = softmax(V * s_t). \tag{3.2}$$

3.1.2 Back-propagation through time in RNN

The RNN output the predictions vector o_t at time step t. We compute the prediction error E_k for time K and use the Back Propagation Through Time (BPTT) algorithm to compute the gradient of the error. We now define our loss, or cumulative error as follows [9]:

$$E(o, \hat{o}) = \sum_{i=1}^{k} E_k(o_k, \hat{o}_k). \tag{3.3}$$

In equation 3.3 we have:
o_k = the actual value at time step k,
$\hat{o}_k$ = prediction made by RNN at time step K,.
$o = (o_1, \ldots, o_k)$, $\hat{o} = (\hat{o}_1, \ldots, \hat{o}_k)$.
We typically treat the full sequence as one training example, so the total error is just the sum of the errors at each time step.

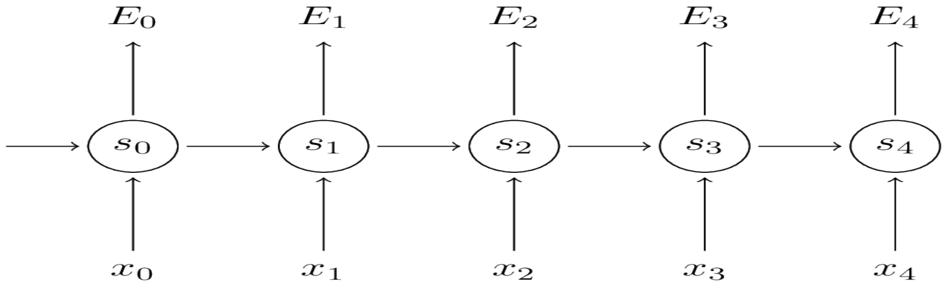


Figure 3.2: Back-propagation through time in RNN [7]

Our goal is to calculate the gradients of the error with respect to our parameters U, V and W and then learn good parameters using Gradient Descent. As we sum up the errors, we also sum up the gradients at each time step for one training example

$$\frac{\partial E}{\partial W} = \sum_{i=1}^{k} \frac{\partial E_k}{\partial W}. \tag{3.4}$$

The gradient is used to update the model parameters by:

$$W = W - \alpha * \frac{\partial E}{\partial W} \tag{3.5}$$

where,

α is the learning rate.

To calculate these gradients we use the chain rule of differentiation. The back-propagation algorithm is applied backwards starting from the final time step.

Let us compute the gradients used to update the network parameters for a learning task that includes k time steps.

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial s_k} * \frac{\partial s_k}{\partial s_{k-1}} .... \frac{\partial s_2}{\partial s_1} * \frac{\partial s_1}{\partial W} \tag{3.6}$$

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial s_k} * \left(\prod_{t=2}^{k} \frac{\partial s_t}{\partial s_{t-1}} \right) * \frac{\partial s_1}{\partial W} \tag{3.7}$$

Let's compute the E_3 as an example, just to have concrete numbers to work with. In figure3.3, the gradient of E_3 with respect to w is given by

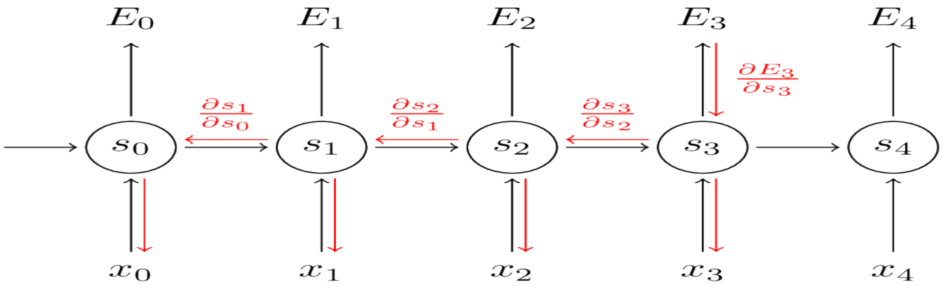

E_0
E_1
E_2
E_3
E_4
$\frac{\partial E_3}{\partial s_3}$
$\frac{\partial s_1}{\partial s_0}$
$\frac{\partial s_2}{\partial s_1}$
$\frac{\partial s_3}{\partial s_2}$
s_0
s_1
s_2
s_3
s_4
x_0
x_1
x_2
x_3
x_4

Figure 3.3: Back-propagation through time in RNN for E_3 [7]

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial o_3} * \frac{\partial o_3}{\partial s_3} * \frac{\partial s_3}{\partial s_2} * \frac{\partial s_2}{\partial s_1} * \frac{\partial s_1}{\partial W}. \tag{3.8}$$

Now, note that $s_3 = f(Ux_t + Ws_2)$ depends on s_2, which depends on W and s_1, and so on. So if we take the derivative with respect to W we can not simply treat s_2 as a constant. We need to apply the chain rule again.

3.1.3 Problems with RNN

RNNs suffer from the problem of vanishing gradients, which hampers learning of long data sequence. The gradients carry information used in the RNN parameters used in the update of RNN parameters. When the gradients become smaller and smaller, the parameter updates become insignificant which means no real learning is done.
Let us consider the gradients used to update the network parameters for a learning task that includes k time steps [9].

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial s_k} * \frac{\partial s_k}{\partial s_{k-1}} .... \frac{\partial s_2}{\partial s_1} * \frac{\partial s_1}{\partial W} \tag{3.9}$$

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial s_k} * \left(\prod_{t=2}^{k} \frac{\partial s_t}{\partial s_{t-1}} \right) * \frac{\partial s_1}{\partial W} \tag{3.10}$$

But, s_t can be written as,

$$s_t = f(Ux_t + Ws_{t-1}) \tag{3.11}$$

Computing the derivative of s_t we get,

$$\frac{\partial s_t}{\partial s_{t-1}} = f'(Ux_t + Ws_{t-1}) * \frac{\partial}{\partial s_{t-1}}(Ux_t + Ws_{t-1}) \tag{3.12}$$

$$\frac{\partial s_t}{\partial s_{t-1}} = f' * (Ux_t + Ws_{t-1}) * W \tag{3.13}$$

Plugging equation 3.13 into equation 3.10 we have,

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial s_k} * \left(\prod_{t=2}^{k} f'(Ux_t + Ws_{t-1}) * W \right) * \frac{\partial s_1}{\partial W}. \tag{3.14}$$

Here in above equation the gradient tends to vanish when k is large due to the derivative of the activation function "tanh" which is smaller or equal to 1 in such cases we have

$$\left(\prod_{t=2}^{k} f'(Ux_t + Ws_{t-1}) * W \right) \approx 0. \tag{3.15}$$

So,

$$\frac{\partial E}{\partial W} \approx 0 \tag{3.16}$$

So, there are some techniques like Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) that help us in solving this type of problem by having the capacity to learn long term dependencies.

3.2 Long Short Term Memory (LSTM)

LSTMs are types of RNN that are capable of learning long term dependencies and remembering information for long periods. Some of text is taken from [19].

3.2.1 Working of LSTMs

The key to LSTM is the cell state. The LSTM has the ability to remove or add information to cell state by taking the use of regulated structure called gates.
An LSTM network has an input vector $[X_t, H_{t-1}]$ at time step t. The network cell state is denoted by C_t. The output vector passed through the network between consecutive time steps t, $t+1$ are denoted by H_t [19].

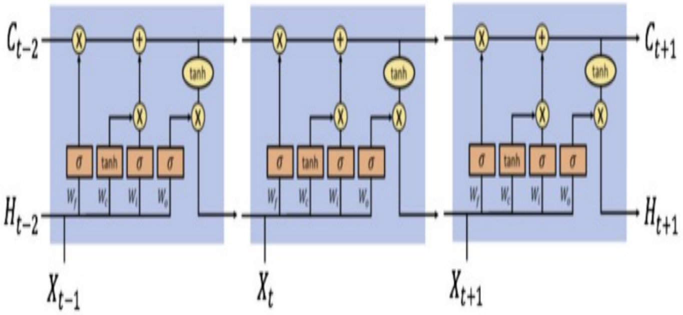


Figure 3.4: Long Short Term Memory [3]

The LSTM has three gates that update and control the cell states, these are the forget gate, input gate and output gate. The gates use sigmoid fuctions and hyperbolic tangent activation functions.

- Forget Gate: This gate is used to decide how much of the past it should remember. It is decided by the sigmoid function. It looks at the previous state (H_{t-1}) and the current input x_t and computes the function

$$f_t = \sigma(W_f * H_{t-1} + W_f * X_t)) \tag{3.17}$$

  where,
  f_t= forget gate activation.

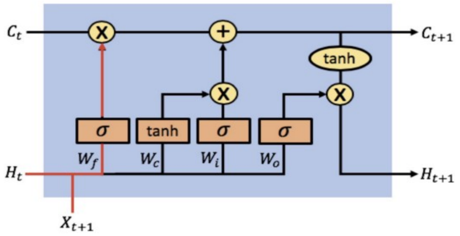


Figure 3.5: The LSTM forget gate update of the cell state C_t [3]

- Input Gate: This gate is used to decide how much should this unit add to the current state. the following sigmoid action function used at input gate:

$$i_t = \sigma(W_i * H_{t-1} + W_i * X_t)) \tag{3.18}$$

- Cell state update:The cell state update is carried out using

$$C_t = C_{t-1} * f_t + \tilde{c}_t * i_t \tag{3.19}$$

where,

$$\tilde{c}_t = tanh(W_c * H_{t-1} + W_c * X_t)) \tag{3.20}$$

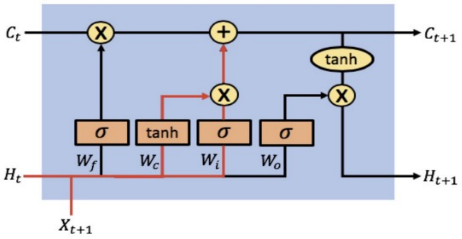


Figure 3.6: The LSTM input gate and cell state update [3].

Output Gate: This gate is used to decide what part of the current cell state makes it to the output. The sigmoid layer is used to decide what parts of the cell state make it to the output. Then we put the cell state through tanh to get the value between -1 and 1 and multiply it by the output of the sigmoid gate the activation at output gate is gievn as:

$$o_t = \sigma(W_o * H_{t-1} + W_o * X_t)) \tag{3.21}$$

The output of LSTM cell is given by:

$$h_t = o_t.(tanh(C_t)) \tag{3.22}$$

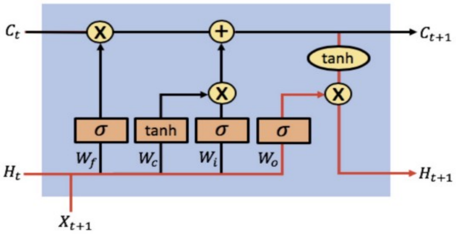


Figure 3.7: The LSTM output gate's action on cell state C_t [3].

3.2.2 Back-propagation through time in LSTM

As in our RNN model, we assume that our LSTM network outputs a single prediction vector H_k on the final kth time step. The knowledge encoded in the state vectors C_t captures long-term dependencies and relationship existing in the sequential data [3]. The length of the data sequences can be hundreds and even thousands of time steps, making it extremely hard to learn using a basic RNN. We compute the gradient we would use to update the network parameters, the computation is done over k time steps.

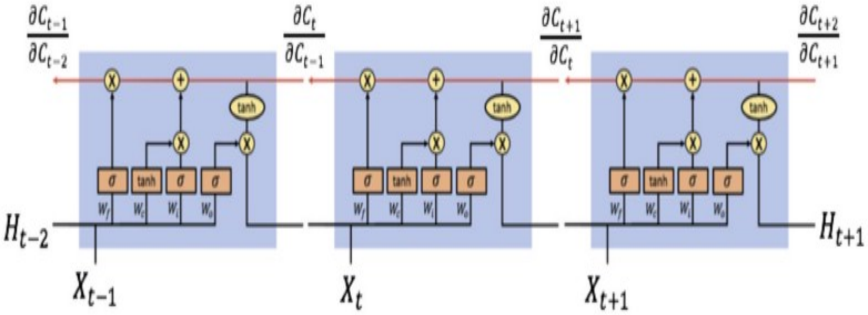

$\frac{\partial C_{t-1}}{\partial C_{t-2}}$
$\frac{\partial C_t}{\partial C_{t-1}}$
$\frac{\partial C_{t+1}}{\partial C_t}$
$\frac{\partial C_{t+2}}{\partial C_{t+1}}$
H_{t-2}
H_{t+1}
X_{t-1}
X_t
X_{t+1}

Figure 3.8: Backpropagating through time for gradient computation [3].

For a learning task with k time steps, as in RNNs, the gradient has form:

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial H_k} * \frac{\partial H_k}{\partial C_k} * \frac{\partial C_k}{\partial C_{k-1}} .... \frac{\partial C_2}{\partial C_1} * \frac{\partial C_1}{\partial W} \tag{3.23}$$

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial H_k} * \frac{\partial H_k}{\partial C_k} * \left(\prod_{t=2}^{k} \frac{\partial C_t}{\partial C_{t-1}} \right) * \frac{\partial C_1}{\partial W}. \tag{3.24}$$

In an LSTM, the state vector C_t, has the form

$$C_t = C_{t-1} * \sigma(W_f * H_{t-1} + W_f * X_t)) + tanh(W_c * H_{t-1} + W_c * X_t) * \sigma(W_i * H_{t-1} + W_i * X_t) \tag{3.25}$$

By computing the derivative of C_t we get,

$$\frac{\partial C_t}{\partial C_{t-1}} = \sigma(W_f * H_{t-1} + W_f * X_t]) + \frac{\partial}{\partial C_{t-1}} (tanh(W_c * H_{t-1} + W_c * X_t]) * \sigma(W_i * H_{t-1} + W_i * X_t)) \tag{3.26}$$

For simplicity, we leave out the computation of:

$$\frac{\partial}{\partial C_{t-1}} (tanh(W_c * H_{t-1} + W_c * X_t]) * \sigma(W_i * H_{t-1} + W_i * X_t)) \tag{3.27}$$

This is of little importance to our proof, as we will see that for the gradients not to vanish, it is enough that the activation of the forget gate are greater than 0.

So we just take,

$$\frac{\partial C_t}{\partial C_{t-1}} = \sigma(W_f * H_{t-1} + W_f * X_t]) \tag{3.28}$$

And we put this equation in equation no(3.24),

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial H_k} * \frac{\partial H_k}{\partial C_k} * \left(\prod_{t=2}^{k} \sigma(W_f * H_{t-1} + W_f X_t]) \right) * \frac{\partial C_1}{\partial W} \tag{3.29}$$

The gradient behaves similarly to the forget gate, and if the forget gate decides that a certain piece of information should be remembered, it will be open and have values closer to 1 to allow for information flow. Hence for simplicity, we can think of the forget gate's action as:

$$\sigma(W_f * H_{t-1} + W_f * X_t)) \approx 1 \tag{3.30}$$

So we get,

$$\frac{\partial C_t}{\partial C_{t-1}} \approx 1 \tag{3.31}$$

Finally,

$$\frac{\partial E}{\partial W} = \frac{\partial E_k}{\partial H_k} * \frac{\partial H_k}{\partial C_k} * \frac{\partial C_1}{\partial W}. \tag{3.32}$$

And, hence the gradients do not vanish.

3.3 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRUs) is also special kind of recurrent neural network which have a gating mechanism. The GRU performs just like LSTM with forget gate but has lesser parameter than LSTM as it does have an output gate. GRU performance on certain tasks of music modeling and speech signal modeling was found to be similar to that of LSTM. GRUs are improved version of standard recurrent neural network to solve the vanishing gradient problem of a standard RNN. GRU uses gates called update gate and reset gate.

3.3.1 Working of GRU

In GRU there are two input vectors which decide what information should be passed to the output. The special thing about gated recurrent units is that they can be trained to keep information from long period, without removing information which is irrelevant to the prediction. Instead of the input, forget, and output gates in the LSTM cell, the GRU cell has two gates, an update gate z, and a reset gate r. The update gate decides how much previous memory to keep around by using the sigmoid function and the reset gate defines how to combine the new input with the previous memory [16].

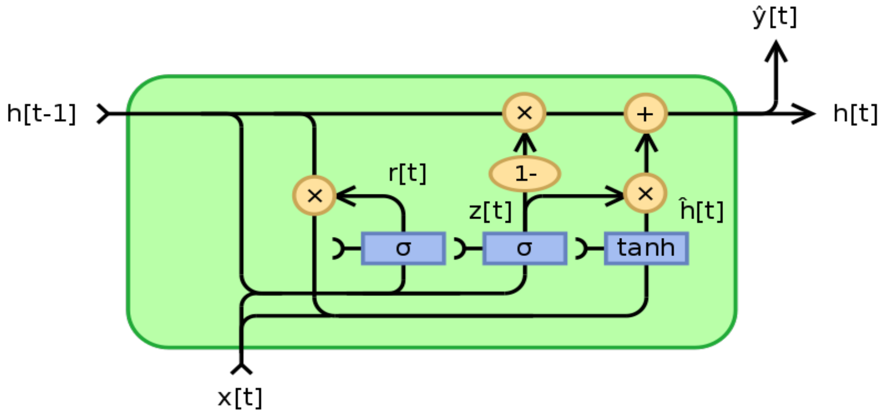


Figure 3.9: Gated Recurrent Unit [2].

The following equations define the gates in a GRU:

- Update Gate: Update gate helps to determine how much of the past information needs to be passed for the future.

$$z_t = \sigma(W_z * h_{t-1} + W_z * x_t)) \tag{3.33}$$

- Reset Gate: This gate is used to decided how much of the past information to forget.

$$r_t = \sigma(W_r * h_{t-1} + W_r * x_t)) \tag{3.34}$$

$$\tilde{h}_t = tanh(W_c * (r_t * h_{t-1} + W_c * x_t)) \tag{3.35}$$

$$h_t = z_t * \tilde{h}_t + (1 - z_t) * h_{t-1} \tag{3.36}$$

The BPTT for Gated Recurrent Unit works similar to that of Long Short Term Memory (LSTM).

So, In this chapter we have studied the machine learning model which can be used for the time series forecasting, as far we learn three machine learning models which are RNN, LSTM and GRU can used for the sequential data like Time Series. In further chapter we are going to discuss about how to apply this statistical models, Machine learning model on the time series data and how this can used for the trend analysis of different parameter and which models perform better on data.

Chapter 4

Experiments & Results

4.1 Trend Analysis on BSP

4.1.1 Data and Data Preprocessing

The data of dialysis patients is from Apex Kidney care. This dialysis data is from 11th March 2019 to 16th March 2019. There are 15 to 18 patients visits for dialysis on a single day. For the one session of dialysis, the external filter runs for 3 hours, so all parameters recording is taken with a fixed interval of time in 3 hours. The time for a filter to run also depends on the goal of 65 percent of urea clearance, if the goal of 65 percent clearance of urea of not meet in 3 hours run then filter running time increase and vice versa. In 3 hours running time of filter, we get around about 4000 to 6000 reading by the dialyzer machine for single patients.

The dialysis machine measure all the parameter like BSP, heart rate, blood flow, dialysate flow, treated blood volume, conductivity, dialysate inlet side pressure, ultrafiltration speed, ultrafiltration remaining, arterial end pressure, venous end pressure, time of dialysis process, current Kt/V, syringe pump speed, intravenous syringe pump, temperature, transmembrane pressure, etc. Since the patient's kidney not working so again after dialysis these wastes in the human body increase. So patients have to come again for the dialysis on the clinician call.

The data of patients are randomly selected on a random day for the time series analysis of dialysis parameters. For the analysis of we have taken the parameter called BSP and applied some statistical models like AR/MA/ARMA/ARIMA and Machine learning models like RNN, LSTM, GRU to analysis the trend of this parameter and compare this trend analysis with some patient having the heart disease.

So, we randomly selected 16th March 2019 data for trend analysis of the parameter "BSP". Around about 15 patient are going through dialysis on that day and we group all these patients with a unique ID called "Patient Number". From the 15 we have randomly selected the patient with ID 41295 to the further trend analysis.
In one session of dialysis of a patient (41295) goes for the dialysis of 3 hours and every two seconds of the interval the data collected by dialyser and we get around about 5057 entries. After grouping all patients data is not arranged according to time series, we arranged it according to time by using the parameter 'LastUpdateTimeStamp' and for verification, we check parameter called "Treated Blood Volume"

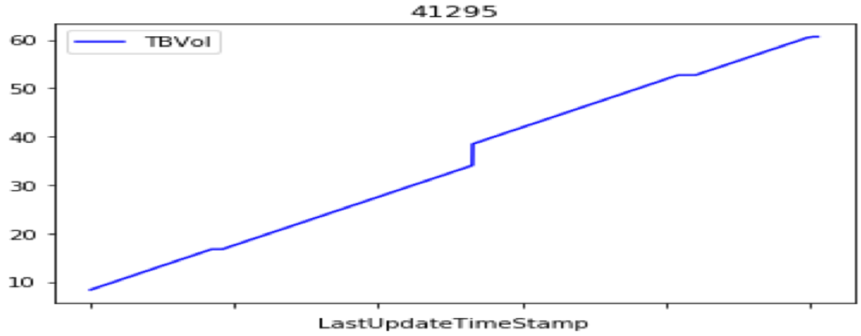


Figure 4.1: Treated Blood Volume for patient (41295)

The above plot is of treated blood volume for the patient with patient number 41295. From plot we say that as the process of dialysis goes on the blood treated by the dialyser machine should also go on increasing. The range of the treated blood volume goes from 0 to 60 and also this value of treated volume varies with patient and day of returning to the dialysis.

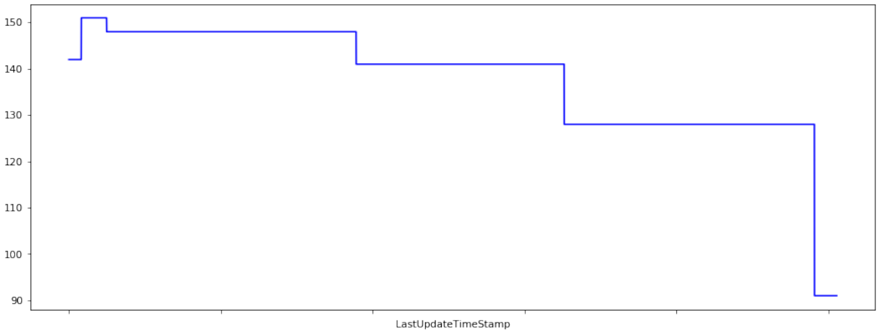


Figure 4.2: BSP for patient(41295)

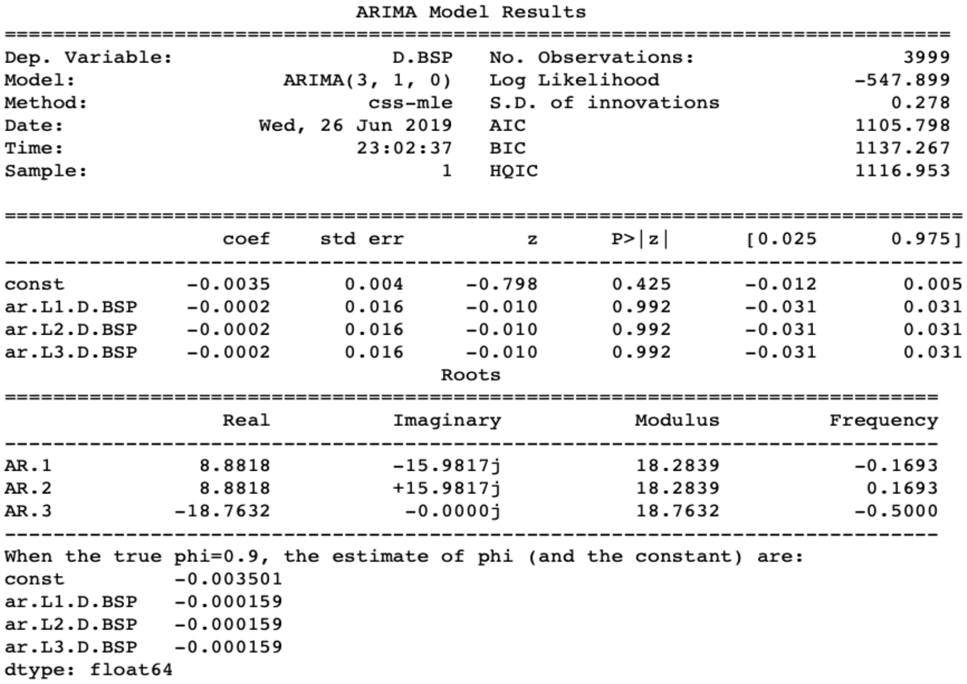
The above plot is of BSP for the patient with a patient number (41295) on which we would be performing the trend analysis by using the statistical model and machine learning models. So to perform the trend analysis, we split the data of the patient (41295) into a train set with 3500 entries and test set with 1557 entries we train this statistical and machine learning models on this data and found the trend of this parameter whether it is increasing or decreasing.

4.1.2 Results for Statistical Models

The following are the steps performed in trend analysis for the statistical models.

- Data collection
- Plot Data
- Check for Stationarity
- Take first difference (if data is non stationary).
- Compute ACF and PACF.
- Fit few AR/MA/ARMA/ARIMA models.
- Use information criteria to choose best model.
  - Akaike information criteria.
  - Bayesian information criteria.
- Forecast the data for a required period.

So we get the data of a patient (41295) for the trend analysis, and we plot the data after plotting the data we did the visual trend analysis and found that the trend of the BSP is going down. Hence we can say that the data is non-stationary data. By taking the first difference of the series data, we convert data into stationary data, and after that we compute the plots of ACF and PACF. Then tried to fit the multiple models with multiple order and found that the ARIMA with order (3,1,0) perform better with lowest AIC and BIC for the given training data and summary given of ARIMA (3,1,0) below in the figure.



```
                              ARIMA Model Results
==============================================================================
Dep. Variable:                  D.BSP   No. Observations:                 3999
Model:                 ARIMA(3, 1, 0)   Log Likelihood                -547.899
Method:                       css-mle   S.D. of innovations              0.278
Date:                Wed, 26 Jun 2019   AIC                           1105.798
Time:                        23:02:37   BIC                           1137.267
Sample:                             1   HQIC                          1116.953

===============================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
const          -0.0035      0.004     -0.798      0.425      -0.012       0.005
ar.L1.D.BSP    -0.0002      0.016     -0.010      0.992      -0.031       0.031
ar.L2.D.BSP    -0.0002      0.016     -0.010      0.992      -0.031       0.031
ar.L3.D.BSP    -0.0002      0.016     -0.010      0.992      -0.031       0.031
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            8.8818         -15.9817j           18.2839           -0.1693
AR.2            8.8818         +15.9817j           18.2839            0.1693
AR.3          -18.7632          -0.0000j           18.7632           -0.5000
-----------------------------------------------------------------------------
When the true phi=0.9, the estimate of phi (and the constant) are:
const          -0.003501
ar.L1.D.BSP    -0.000159
ar.L2.D.BSP    -0.000159
ar.L3.D.BSP    -0.000159
dtype: float64
```

Figure 4.3: Summary for ARIMA (3,1,0) for patient(41295)

As we said earlier the model with lowest AIC and BIC is selected as the best model here we have the ARIMA model which gives us the AIC and BIC 1101.79 and 1120.67 respectively and this RMSE of 12.630

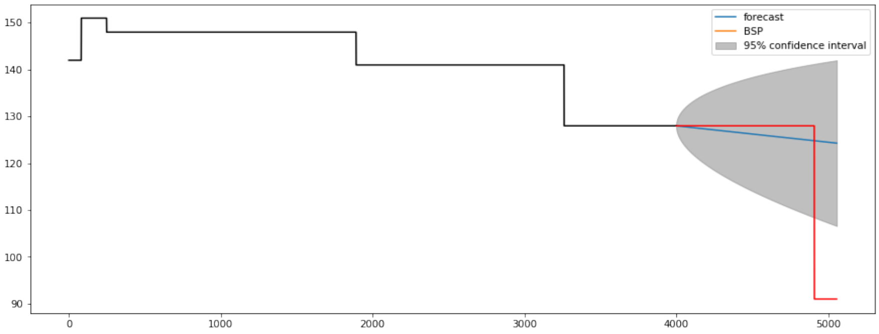


Figure 4.4: ARIMA with order (3,1,0) on BSP

The above plot is of BSP for the patient with patient number (41295) on which we have performed the trend analysis by using the ARIMA model with order (3,1,0) as we can see that the data of 3500 is taken as the training and after fitting the ARIMA with order (3,1,0) we get the trend of the BSP which is dercresing as shown in the plot and by comparing with the trend plot with the test value we get the RMSE value of 12.630.

4.1.3 Results for Machine Learning Models

The following are the steps performed in trend analysis for the Machine learning models.

- Formulate the series for an RNN/LSTM/GRU supervised learning regression problem
- Scale all the series
- Split the series for training and testing
- Reshape the series for RNN/LSTM/GRU implementation
- Define the initial for the RNN/LSTM/GRU model
- Produce the forecast and reverse-scale the forecasted series.
- Forecast the data for the required time.
- Calculate the loss as Root mean squared error(RMSE)

So after getting the series of BSP data for the patient (41295), we formulate the series for supervised learning regression, then feature scaling is done on this supervised learning regression series and for this, we have used the "MinMax Scaler" to scale all the series values. After that, we split the data into training with 4000 entires and testing with 1057 entires, then fitted the models like RNN, LSTM and GRU on this training data and produce the forecast and reverse-scale the forecasted series. We forecast the data for the required time and test this forecasted value with a the real value of BSP and found the RMSE value. The below plot is of original BSP for the patient (41295)

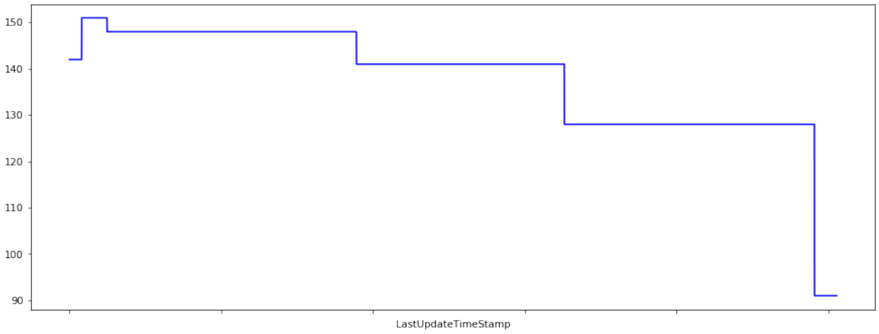


Figure 4.5: BSP for patient(41295)

1. Recurrent Neural Network (RNN)

   Plots of actual and prediction values of BSP by RNN model.

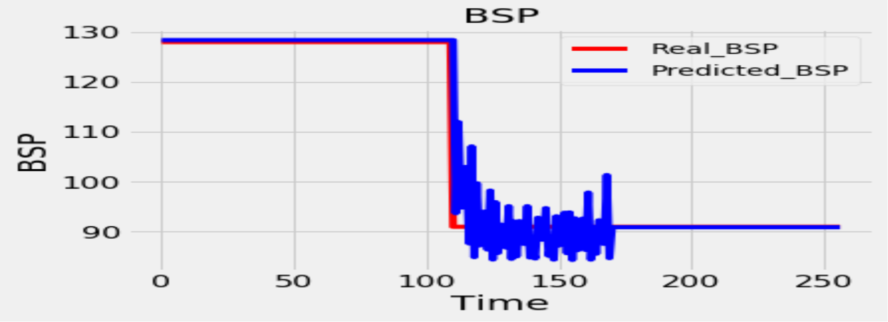


Figure 4.6: Result for RNN on BSP

2. Long Short Term Memory (LSTM)

   Plots of actual and prediction values of BSP by LSTM Model.

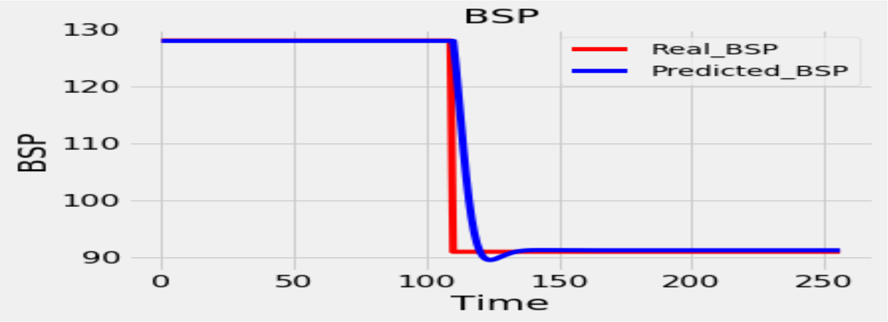


Figure 4.7: Result for LSTM on BSP

3. Gated Recurrent Unit (GRU)

   Plots of actual and prediction values of BSP by GRU Model.

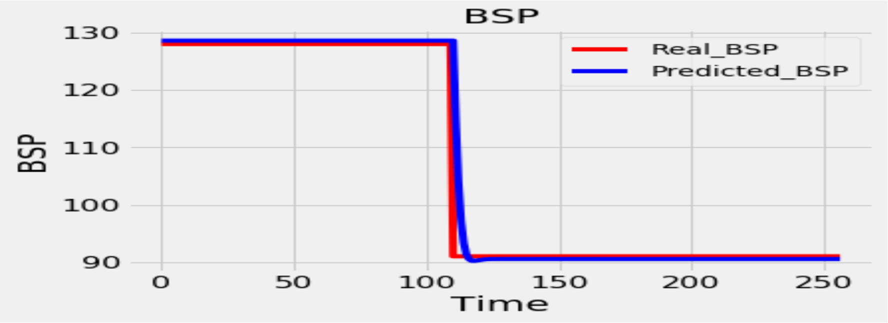


Figure 4.8: Result for GRU on BSP

From the above plot, we observed that the following RMSE values for machine learning models given below in the table below. From the above table we can observe that Gated Recurrent Unit (GRU) performs better on the training data and gives the minimum RMSE. But if we keenly observe the plot of all models RNN, LSTM and GRU we can see that these all models are just following the actual path of the real BSP with some lag as we can see that there is drastic change in between 100 and 150 seconds and our models and also give the exact drastic change with some sufficient lags.

Sr No	Model	RMSE
1	RNN	2.135
2	LSTM	1.967
3	GRU	1.717

Table 4.1: RMSE for Machine Learning Models

4.1.4 Comparison of Statistical models and Machine learning models

So we compare the Machine learning models and Statistical models below.
Statistical Models:

- A linear system of equations
- Applied only on Stationary Time Series.
- Data Prepossessing and Model specification is relatively straight forward.
- Examples: Financial time series, Business time series and other numeric series data.

Machine learning Models:

- Layers of non-linear transformation
- Stationary is not a requirement.
- Data Prepossessing, Model training, and hyperparameter tuning require some effort
- Examples: DNA sequence, Image, Voice sequence, Text and all other numeric series data

4.2 Clustering Analysis on BSP

We used the K-means algorithm for cluster analysis of BSP for various sessions of dialysis. Some of text is taken from [13].

4.2.1 K-Means Clustering

K-means is a type of unsupervised learning algorithm, which can be used when there are no labels available. The goal of this algorithm is to find the number of groups in the data, with the number of groups represented by variable K. The algorithm works in iteration to assign each data point to one of K groups based on the features that are provided [20]. The K-means clustering algorithm in data mining starts with a K randomly selected centroids, which are used as the beginning centroids for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids the objective function to be optimize is given as: [13].

$$J = \sum_{j=1}^{K} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{4.1}$$

where,
$\left\| x_i^{(j)} - c_j \right\|^2$ = Distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.
The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

2. Assign each object to the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although there are proofs of termination of this K-means algorithm, it does not necessarily find the most optimal configuration. K-means is a simple algorithm that has been adapted to many problem domains like Behavioral segmentation, Inventory categorization, Group images, etc [12].

4.2.2 Data and Data Preprocessing

The data of the dialysis patients taken for the clustering analysis is from 30th April 2019 to 28th May 2019. The number of patients (33) goes for multiple session (313) of the dialysis from these 33 patients we have selected the patient whose entries of the dialysis greater than 2500 for the single session. So after applying the filter of 2500 entries we get around about 96 sessions of 33 patients. So, for 96 sessions we sort the data point all according to the time series and look for the parameter called "Treated blood volume (TBVol)" as we know that this parameter should be gradually increase as the process of dialysis goes on. Hence the plot must look as given below.

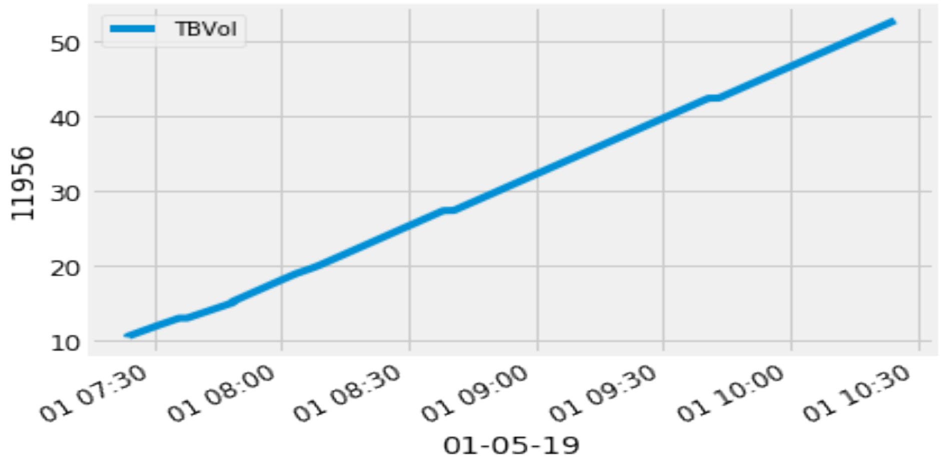


Figure 4.9: Gradual increase in TBVol

So, we plotted the plots of TBVol for all the 96 sessions of different patients and look for the gradual increase in this parameter and after plotting all 96 sessions of all patient.

We get 14 session of different patient are not properly increasing and below plot is just the example of this 14 session, So we neglect this 14 session from this 96 session and clustering analysis is on the this remaining 82 session of different patients.

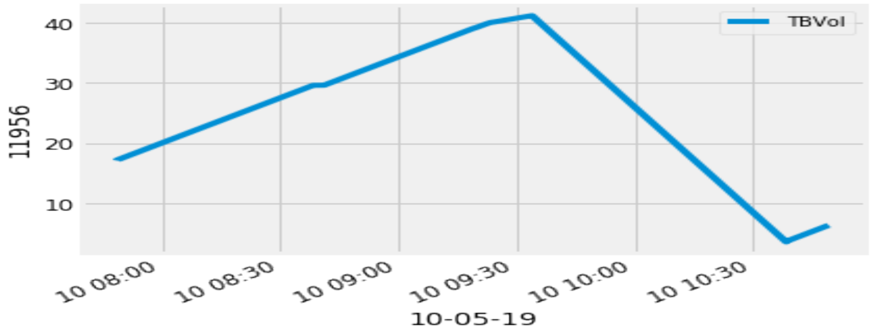


Figure 4.10: Non-Gradual increase in TBVol

After neglecting the 14 session in which the treated blood volume are not gradually increasing, we get 82 sessions of dialysis, But the entries of all this 82 sessions are not same some are having 4000 and some are having above 5000. So, we take the data of first 2500 points for all the 82 sessions and then from each sessions of dialysis we find the number of drastic increase in BSP and number of drastic decrease in BSP for example given below.

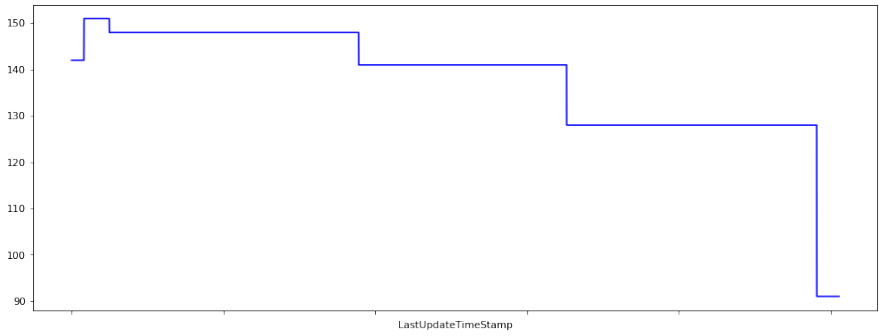


Figure 4.11: BSP for patient(41295)

Here in the above plot of BSP we can observe that there are four drastic decrease and one drastic increase in the BSP. We did this for all 82 sessions and got array 82 by 2, where 2 give number of increase and number of decrease for all 82 sessions and the K-means clustering is applied on this array.

4.2.3 Results for K-means Clustering

We have applied K-means clustering algorithm with $K = 3$ to cluster the behaviour of BSP for 82 session of dialysis of different patient on different day.

1. Cluster 0

Patient Number	Date	Entries	Cluster
14358	10/05/19	2573	0
14358	17/05/19	3918	0
14358	30/04/19	2796	0
17619	04/05/19	3450	0
21415	06/05/19	5619	0
22784	28/05/19	2655	0
25478	30/04/19	3358	0
27150	01/05/19	3947	0
27150	03/05/19	2938	0
27150	17/05/19	3941	0
27150	22/05/19	3701	0
30868	06/05/19	3277	0
31962	06/05/19	2814	0
31962	17/05/19	3862	0
31962	22/05/19	4404	0
32625	03/05/19	3154	0
32625	06/05/19	2822	0
32922	06/05/19	4466	0
39491	01/05/19	4843	0
39708	03/05/19	3324	0
41684	06/05/19	2838	0
41968	28/05/19	2652	0
43091	10/05/19	2578	0
43091	13/05/19	3394	0
8436	02/05/19	2648	0
8436	04/05/19	3710	0
8436	28/05/19	2701	0

Table 4.2: Patients in Cluster 0

The above table is of patients going for the dialysis on different day and clustered in cluster 0 category and hence, out of 82 session 27 session are clustered in cluster 0 category and plot for the BSP for this can be given as below and from plot of "BSP" we can say that all the session of patient of dialysis the BSP for this cluster patients is increasing.

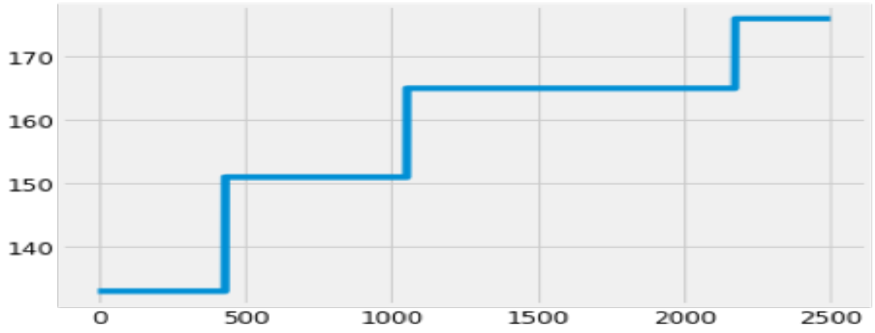


Figure 4.12: Sample of 'BSP' for Cluster 0

2. Cluster 1

Patient Number	Date	Entries	Cluster
11956	06/05/19	2838	1
11956	17/05/19	3805	1
16897	28/05/19	2715	1
17619	02/05/19	2790	1
17619	10/05/19	3509	1
18179	03/05/19	3390	1
18179	22/05/19	4434	1
28991	28/05/19	2671	1
28991	30/04/19	3201	1
31431	30/04/19	3132	1
32625	17/05/19	3826	1
35331	03/05/19	3078	1
35331	06/05/19	2799	1
36306	17/05/19	3924	1
37835	06/05/19	3375	1
37835	10/05/19	3829	1
37835	13/05/19	3489	1
7384	13/05/19	3465	1

Table 4.3: Patients in Cluster 1

From we can see that out of 82 sessions 18 sessions were clustered in cluster 1 category and plot for the BSP for this can be given as below, and from plot of "BSP" we can say that all the session of patient of dialysis the BSP for this cluster patients is decreasing.

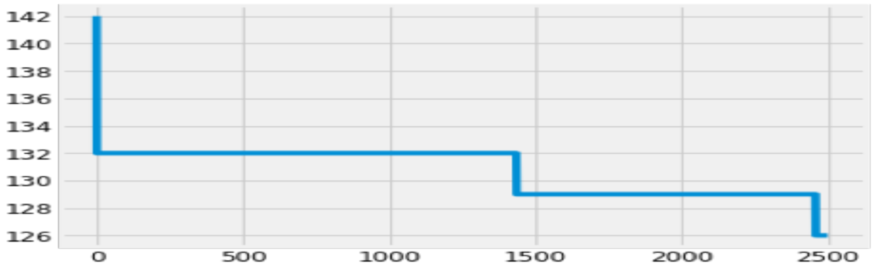


Figure 4.13: Sample of 'BSP' for Cluster 1

3. Cluster 2

Patient Number	Date	Entries	Cluster
11956	01/05/19	4895	2
11956	27/05/19	2611	2
14358	03/05/19	3281	2
16897	02/05/19	2701	2
16897	04/05/19	3709	2
16897	16/05/19	3341	2
17619	16/05/19	5112	2
18179	10/05/19	3457	2
22784	04/05/19	3710	2
26710	01/05/19	4031	2
27150	27/05/19	2578	2
27271	18/05/19	3623	2
27271	30/04/19	3353	2
28991	04/05/19	3711	2
29438	10/05/19	2890	2
30591	01/05/19	3322	2
30591	28/05/19	2748	2
30868	03/05/19	3332	2
31962	01/05/19	4489	2
31962	03/05/19	3430	2
31962	27/05/19	2637	2
32625	10/05/19	3891	2
32625	27/05/19	2664	2
35331	01/05/19	4642	2
35331	10/05/19	3456	2
37425	02/05/19	2667	2
38687	01/05/19	3034	2
39708	13/05/19	3997	2

Table 4.4: Patients in Cluster 2

Patient Number	Date	Entries	Cluster
41684	13/05/19	3459	2
41968	04/05/19	3711	2
41968	16/05/19	3238	2
41968	30/04/19	3600	2
42959	01/05/19	2951	2
43091	01/05/19	4072	2
7384	01/05/19	3543	2
8436	16/05/19	4234	2
8436	30/04/19	3601	2

Table 4.5: Patients in Cluster 2

The above table is of session of dialysis clustered in cluster 2 category and hence, out of 82 session 45 session are clustered in cluster 2nd category and plot for the BSP for this category can be given as below

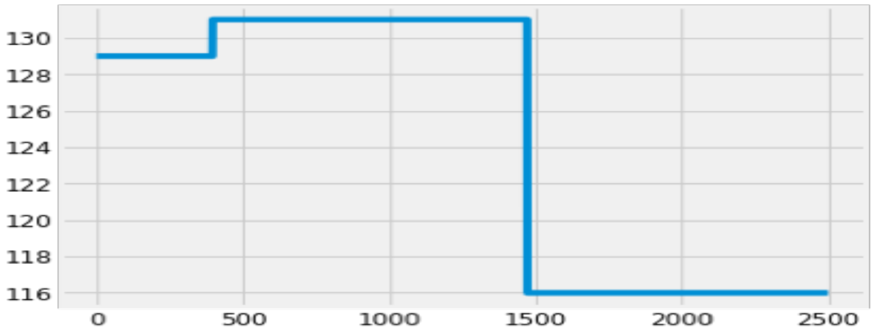


Figure 4.14: Sample of 'BSP' for Cluster 2

From the plot of "BSP" we can say that for all the session of dialysis the BSP for this cluster patients is has a drastic change in middle of the dialysis process.

4.3 Classification Analysis using the parameter Temperature("Temp")

4.3.1 Data Preprocessing

In clustering analysis we have 82 sessions for the different patients on different days, we taken the data for all 82 sessions for the classification analysis. For classification analysis we have taken a variable with name "Temp" which is temperature of the patient on dialysis. The range for this "Temp" varies from 34 to 38 degree Celcius . So we created the class form this parameter as given below.

Sr No	Range of Temperature	Class
1	less than 35.5	0
2	35.5-36.5	1
3	36.5-37.5	2
4	37.5-38.5	3
5	greater than 38.5	4

Table 4.6: Class by using the Temperature

From each 82 sessions, we have taken three series of each of length 60 for the parameter called "BSP" and their corresponding class of the temperature was taken as the label. So we get a data set of 246 data points each having 60 features and label using corresponding temperature class. We divide this dataset into X-train, y-train and X-test, y-test by split criteria of 80-20 percent. So we get around about 184 datapoint in X-train and 62 data points in X-test.

But after taking the three series of 60 units from one session for X-train, we found that for all the values of BSP in X-train the class are either divided in class 1 or in class 2 only in y-train.

This series "BSP" with there class is then passed through the LSTM model and found the results as follows.

4.3.2 Results of Classification

In our case, we have applied the LSTM with 128 units and output with two category which are class 1 and class 2 where class 1 belongs to a temperature between range 35.5 to 36.5 degree Celcius and class 2 belong to 36.5 to 37.5 degree Celcius. Hence, training the LSTM model on X-train and y-train for 1000 epoch, we get training accuracy of 60.87 percent, and then we predicted the class for the X-test data and matched with our y-test and got test accuracy of 64.52 percent, but for all data in X-test output was class 1 which is strange. So, after looking further into it found we that there is no relationship between the BSP value and temperature value which we have categorized.

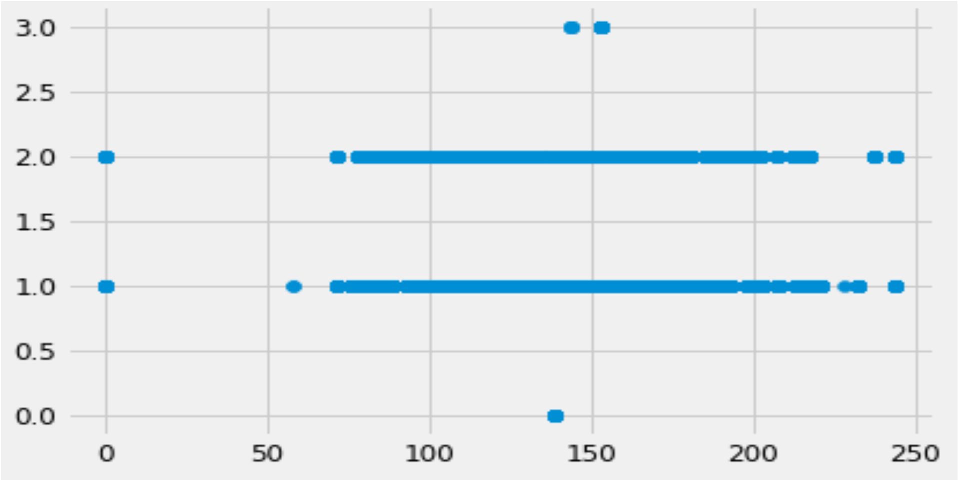


Figure 4.15: BSP vs Class of temp

Hence, from the above plot of BSP vs. class of temperature, we can observe that there are particular dependencies in BSP and classes of Temp. So, our model is not training well on this type of data and giving only class 1 for all values in X-test. So, we can say that the temperature of patients can not be used to classify data based on BSP of patients on dialysis.

Chapter 5

Conclusion

In the hemodialysis patients, irregularities in certain dialysis related parameters may lead to decrease or increase of BSP and other symptoms, which further lead to health deterioration. This may be better analyzed by time series analysis models like Statistical and Machine Learning models. These models are used to find the future value trends over time, which indicate if there may be a decrease or increase in the BSP and other parameters. On doing the time series analysis on the different patients on different days, we found the following conclusions:

- By performing experiments statistical models and machine learning models and looking for root mean square error (RMSE) value, we see that machine learning models perform better.

- Statistical Models, can gives an idea about trends, but they are not suitable for predicting or forecasting sudden changes in values of BSP, which may happen because of parameter exceeding some threshold value.

- Machine learning model are able to capture drastic change in BSP, but after keen observation we find that this not actually predicting this models are just following the path of BSP with some lags..

- From clustering analysis, we are able to group the patients according to their increasing and decreasing trend of parameter "BSP".

- From classification analysis, we can say that the temperature of patients cannot be used to classify data based on BSP of patients.

After looking for the parameter called "treated blood volume", we noticed that there were many patients whose treated blood volume parameter did not gradually increase Since it know that this parameter should gradually increase during the dialysis sessions, we had meeting with doctors after phase one (December 2018) then doctors made some valuable changes in the application to get the clean data and from March 2019 on-wards we started getting the clean data. Thus our initial analysis of Treated blood volume helped to identify irregularities in data collection process
The trend analysis, clustering analysis and classification analysis is done on the BSP by considering that the data of this BSP values are collected every 2 seconds, but again having meeting with the doctors on 26th June 2019, we got to know that this value of BSP is collected every 15 minute during dialysis and this frequency seems to vary for different patient. Doctors have promised to collect the data in fixed frequency for all the patients and give us proper data of BSP. Thus our data analysis hepled to identify discrepancies notice in data collection of BSP parameter.
The practical application of this model will include a continuous monitoring process for alerting clinicians when there is deterioration in a patient's status that signals some clinical outcome, thus providing an opportunity for intervention and proper care of the patients.

Chapter 6

Future Work

The following are the points that can consider as the future work on this project:

- The BSP values are continuously varying with time, the sudden changes in parameter is difficult to predict, but we can use change point detection models which can capture this type of behaviour.

- For prediction of health deterioration and cause of health deterioration in hemodialysis patients we need labels which can tell the status of the patient during the dialysis these label can be more useful in predictive analysis.

- Parameter selection to identify key indicator which improve the health of patients can be a future work. For this task also we need label based on the patients health.

References

[1] A.J. O'Malley, B. N., 2014 oct, "Akaike information criterion," https://www.sciencedirect.com/topics/medicine-and-dentistry/akaike-information-criterion

[2] Antonio Gulli, S. P., 2017, *Deep Learning with Keras* (ISBN 9781787128422 ÂĪ' Packt Publishing Limited. All Rights Reserved).

[3] Arbel, N., 2018 oct, "Long short term memory," https://medium.com/datadriveninvestor/how-do-lstm-networks-\solve-the-problem-of-vanishing-gradients-a6784971a577

[4] australia.gov.au, 2017 oct, "Components of time series," https://www.atap.gov.au/tools-techniques/travel-demand-modelling/6-forecasting-evaluation.aspx

[5] Brian Krans, A. G., 2018 may, "Types of dialysis," https://www.healthline.com/health/dialysis#types-of-dialysis

[6] BRITZ, D., 2015 sep, "Recurrent neural networks," http://www.wildml.com/wp-content/uploads/2015/09/rnn.jpg

[7] BRITZ, D., 2015 oct, "Recurrent neural networks," http://www.wildml.com/wp-content/uploads/2015/10/rnn-bptt-with-gradients.png

[8] BRITZ, D., 2015 sep, "Recurrent neural networks," http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[9] BRITZ, D., 2015 oct, "Recurrent neural networks," http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-\part-3-backpropagation-through-time-and-vanishing-gradients/

[10] Brownlee, J., 2016 dec, "Non-stationarity time series," https://machinelearningmastery.com/time-series-data-stationary-python/

[11] Christopher Thiam Seong Lim, K. J. C. M. A. K. N. Y. L. A. L., Xian Hui Yap, and Goh, B. L., 2015, “Predictor of cardiovascular risks in end stage renal failure patients on maintenance dialysis,”

[12] Dabbura, I., 2018 sep, “Kmeans clustering,” `https://towardsdatascience.com/k-means-clustering-algorithm-\applications-evaluation-methods-and-drawbacks-aa03e644b48a`

[13] Garbade, D. M. J., 2018 sep, “Kmeans clustering,” `https://towardsdatascience.com/understanding\k-means-clustering-in-machine-learning-6a6e67336aa1`

[14] Hyndman, R. J., 2014, *Forecasting: Principles Practice* (University of Western Australia).

[15] Kang, E., 2017 aug, “Stationarity time series,” `https://medium.com/@kangeugine/time-series-check-stationarity-1bee9085da05`

[16] Kostadinov, S., 2017 dec, “Gated recurrent unit,” `https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be`

[17] Lacson, R., 2008, “Predicting hemodialysis mortality utilizing blood pressure trends,” in *AMIA Annual Symposium Proceedings*, Vol. 2008 (American Medical Informatics Association). p. 369.

[18] Sawla, S., 2018 oct, “Components of time series,” `https://medium.com/greyatom/introduction-to-time-series-analysis-431beb02adc4`

[19] SRIVASTAVA, P., 2017 dec, “Long short term memory,” `https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/`

[20] Trevino, A., 2016 dec, “Kmeans clustering,” `https://www.datascience.com/blog/k-means-clustering`

[21] YassineMrabet, 2008, “Simplified hemodialysis circuit,” `https://en.wikipedia.org/wiki/Dialysis#/media/File:Hemodialysis-en.svg`

[22] Zaiontz, C., 2000, “Time series analysis,” `"http://www.real-statistics.com/time-series-analysis"`