CHAPTER 1

INTRODUCTION

Chapter 1

INTRODUCTION

This Chapter introduces primarily, the basic concepts of time series data (TSD). Significance of predictive data mining on TSD is presented. Various prediction models for TSD and their modeling procedures are outlined. Further, decomposition technique based TSD prediction models are illustrated. Where ever necessary, examples are used to illustrate the concepts.

The chapter is organized as follows. In section 1.1, the basics of TSD and nature of TSD are explained. In section 1.2, the concepts of data mining, and in particular the predictive data mining methodology is introduced. Further, the necessity of prediction, prediction model and related terminologies are explained. Various TSD prediction models are introduced in section 1.3. In section 1.4, decomposition technique based prediction models are illustrated. In section 1.5, the motivation and scope of the thesis work are presented. In section 1.6, the complete organization of rest of the thesis chapters is outlined.

1.1 Time Series Data (TSD)

A TSD is a sequence of values which vary with time. It can be represented as a table which comprises of time instant and value of TSD at that time instant. Many a time, a useful representation would be a rectangular plot, where $X-$axis indicates time and $Y-$axis indicates value of TSD. Some examples of TSD origination are:

1. The internet traffic for each second is different. So it is a TSD which shows number of bytes used as a function of seconds.
2. The price of gold on each day is different. So gold price is a TSD which shows price of gold as a function of days.
3. The cost of a commodity as a function of day is yet another TSD.
4. The rainfall in a given place as a function of month is another TSD.
5. The growth of a child can be shown as height vs. age, which represents a TSD.
6. The number of children born every minute is also another TSD.
7. The number of trees planted as a function of day is a TSD.

Other than the above many other TSD like electricity price, mobile calls, global temperature and stock market data. All these TSD do not have same origination. Depending on the application, the nature of originated TSD varies. Some typical classifications of TSD based on one or more attributes, with examples in each case are illustrated below.

1. **Linear vs. Non-linear** Based on the nature of the approximate curve fit on the data, the TSD are either linear or nonlinear.

   (a) **Linear:** Some of the TSD have linear nature. For example, the height of a child as a function of age is a linear TSD. The height vs. age of a male child from 1 to 10 years is plotted as shown in Figure (1.1). In the same figure, an approximate linear curve is also plotted, which is almost similar to the height vs. age TSD.

   (b) **Nonlinear:** Not all the TSD are linear in nature. In fact many available TSD can be roughly approximated to linear over a very short duration with a significant error. However, they can be categorized as non-linear TSD. For example global temperature TSD is shown in Figure (1.2). In the same figure, an approximate fourth order polynomial curve is fit. Thus it exhibits non-linear nature.

2. **Periodic vs. Non-periodic** Based on whether the values of TSD vary in regular intervals or not, the TSD are either periodic of non-periodic.

   (a) **Periodic:** Some TSD are periodic in nature. For example, the sunspot numbers repeat at regular intervals with a finite periodicity as shown in Figure (1.3). These type of TSD are said to exhibit seasonal dependencies. Some other examples are road traffic is high at some particular times in a day, climate

variations repeat according to the seasons.

(b) **Non-periodic:** The TSD which do not have periodic variations are called as non-periodic TSD. The TSD shown in Figure (1.1) and Figure (1.2) are non-periodic. Other examples include financial and stock market data.

3. **Gaussian and Non-Gaussian**: If the TSD values have a Gaussian or normal distribution, they are Gaussian TSD, else Non-gaussian in nature.

4. **Low Volatile and Highly volatile**: If the conditional variance changes slowly as a function of time, the TSD is termed as low volatile, else it is highly volatile TSD. These type of data are shown in Figure (1.4) and Figure (1.5) respectively, which are financial data.

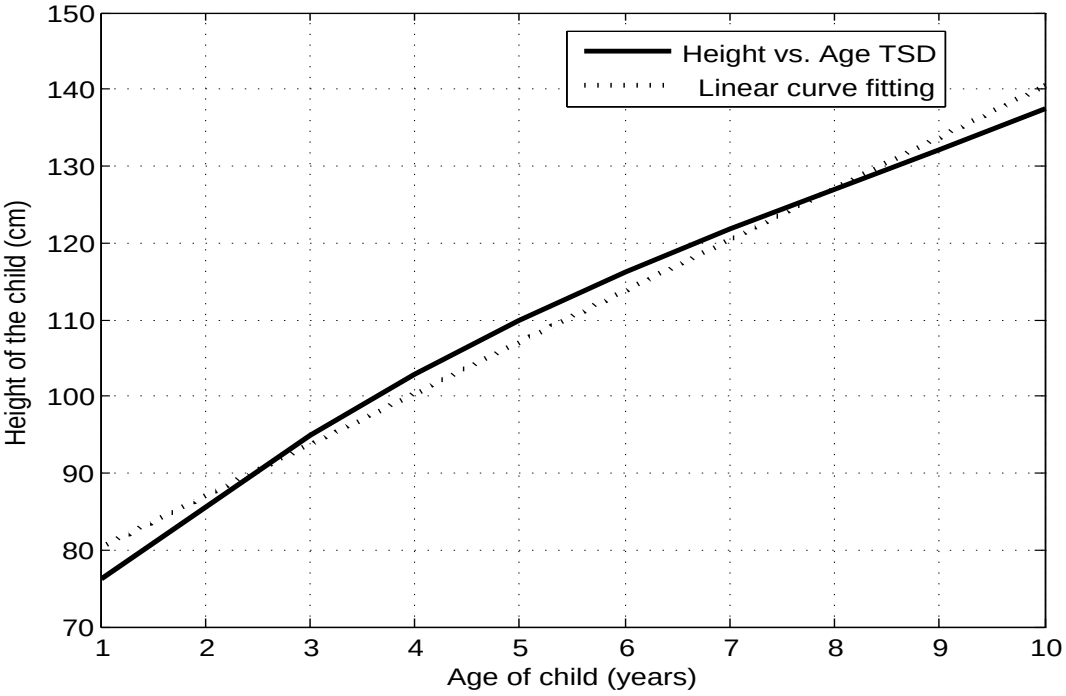

Figure 1.1: Linear TSD: Height of a child

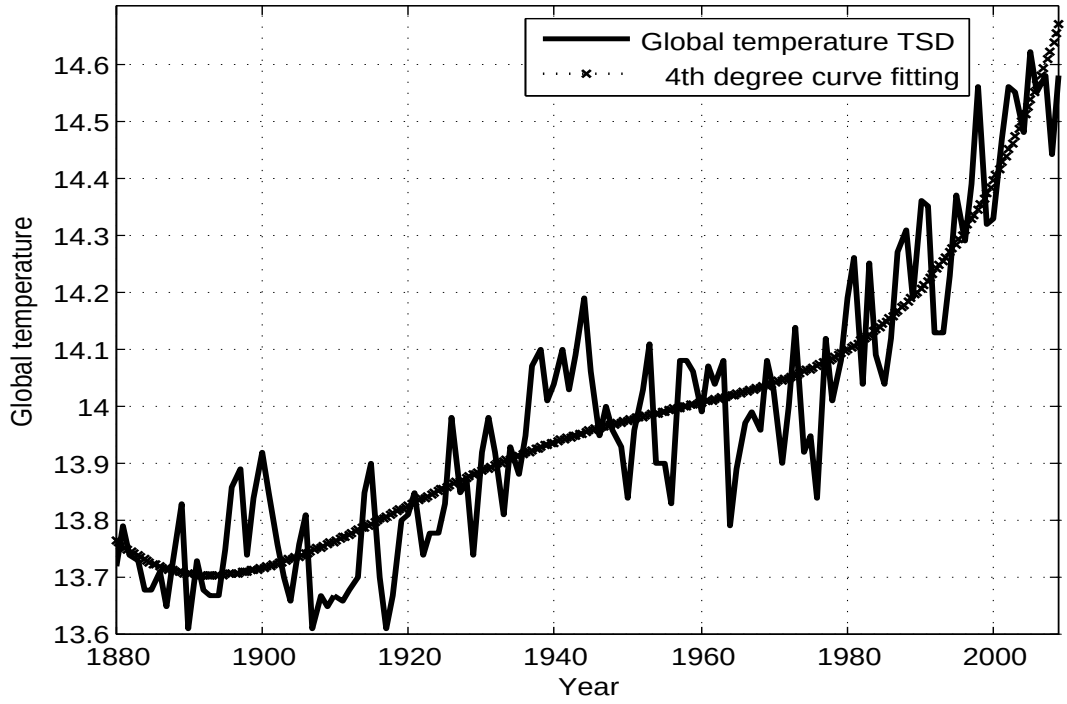

Figure 1.2: Exponential TSD: Average global temperature data

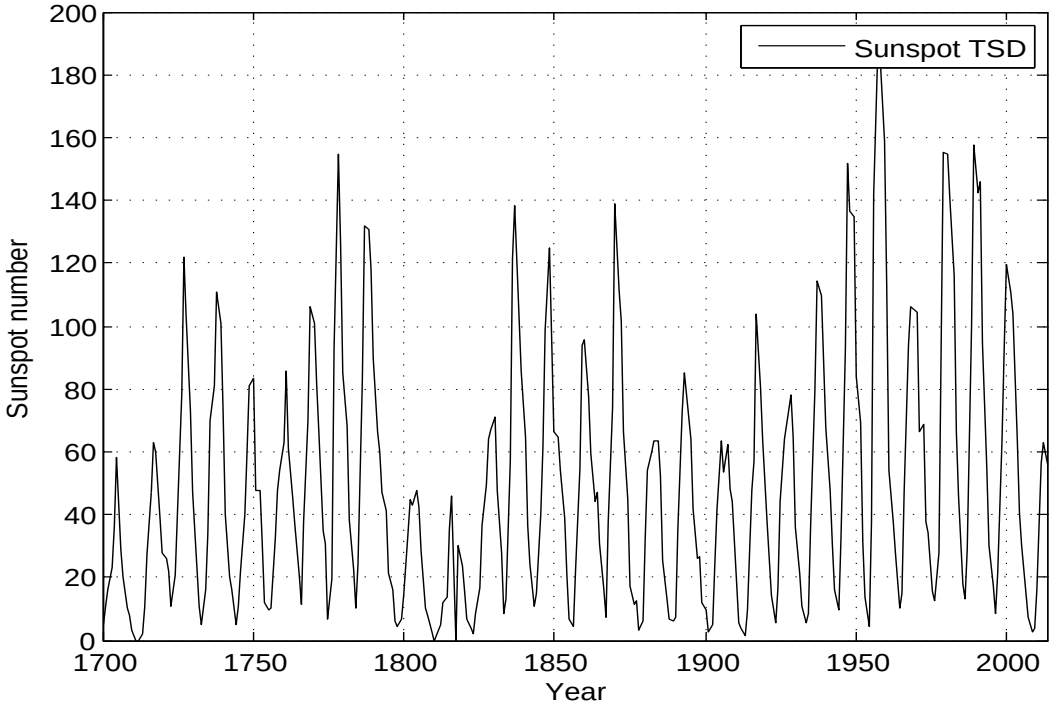

Figure 1.3: Periodic TSD: Sunspots number every year

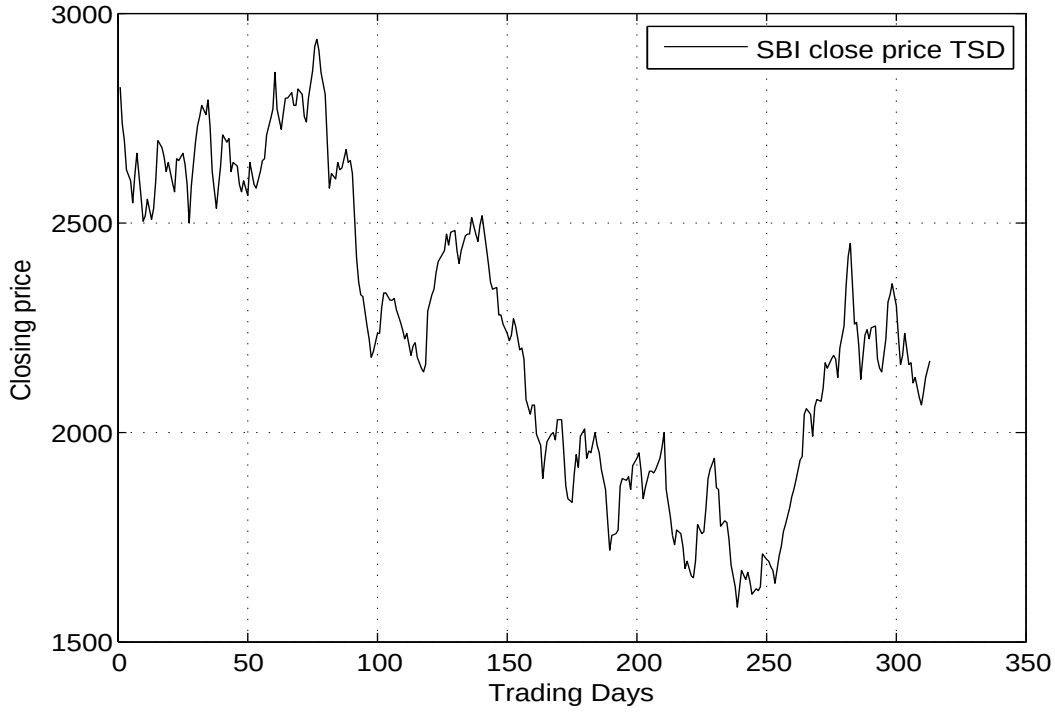

Figure 1.4: Daily closing price of SBI stock TSD

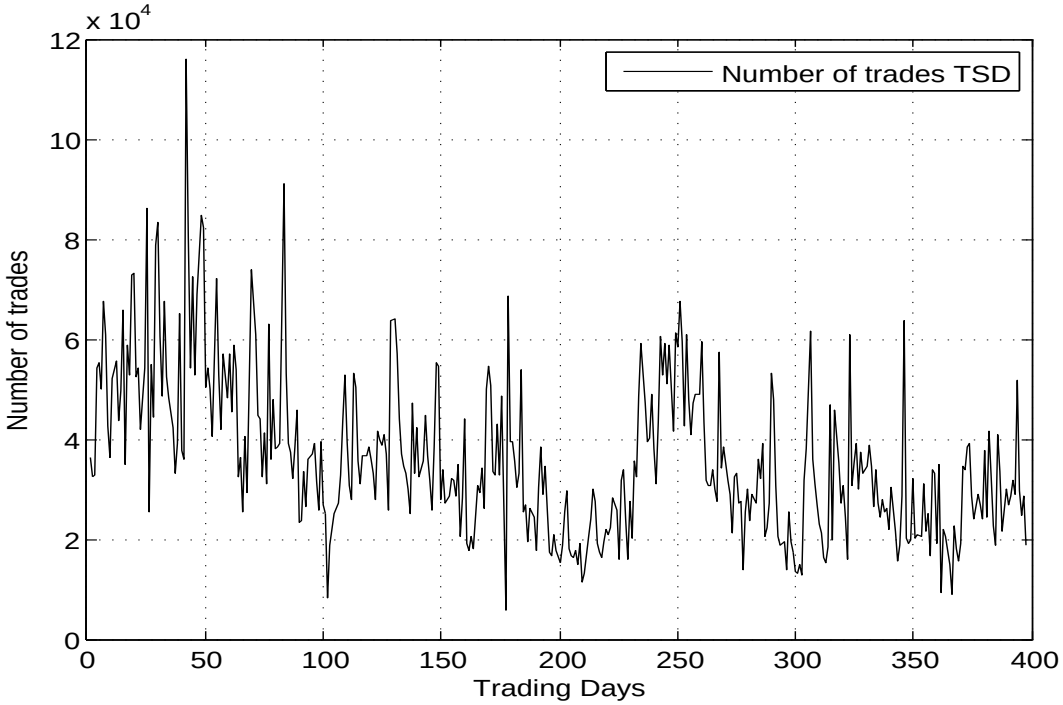

Figure 1.5: Number of trades of Tata Steel TSD

1.2 Predictive Time Series data mining

Data mining is an important branch of computer science which aims to extract and transform the data originating from various fields for the purpose of using it in various applications. Predictive and Descriptive data mining are the two branches of data mining. The descriptive data mining aims to describe general or special features of a set of data in a concise manner in order to extract knowledge hidden in the available data. The descriptive tasks do not predict a target value, but focus more on the intrinsic structure, relations, interconnectedness, etc. of the data to extract the knowledge. On the other hand predictive data mining's primary focus is to provide the future of one or more attributes of the data, making use of the presently available data.

The primary focus of the thesis is predictive mining of TSD. Generally, the TSD contains very valuable information in it. This information has to be extracted using data mining techniques, then it has to be transformed into knowledge to use it for the purpose of making critical decisions. Predictive data mining is one of the technique applied on TSD to transform the available information in the TSD into useful knowledge. Time series forecasting is now a very important research area, owing to the importance of prediction in various applications. Some applications and use of prediction are discussed below.

1. Forecasting network activity helps the service providers to control the available bandwidth and offer better services to customers.

2. Forecasting climate change helps the agricultural sector. For example accurate rainfall prediction helps the farmers to track their farming.

3. Forecasting disasters like earthquakes, cyclones, Tsunami etc., aids in taking necessary precautions and helps mankind to be prepared.

4. Based on the future trends of the market, investors can choose when to invest, and how much to invest.

5. Forecasting also helps in controlling traffic, assessing future economic conditions of a country and also in health care diagnosis.

Predictive data mining has three stages namely,

- initial exploration stage
- model building stage
- deployment stage

In the first stage, data is explored for various attributes which may help in prediction. A detailed list of parameters which effect the given TSD is prepared to be used by the next stage. This information is incorporated while developing the model at a suitable stage so that the model best fits the given TSD. Then the model is validated and deployed for making predictions.

Consider a TSD represented as $y_t, t = \{0, 1, 2, \ldots\}$, where t indicates the time point and y_t indicates value of TSD at time point t. Generally, in all practical applications only finite number of TSD values are available.

Prediction or forecasting of TSD implies finding the future TSD values using the available TSD values. The horizon or interval over which prediction takes place is termed as prediction horizon. Different types of prediction may be required based on the application at hand. These are:

1. **One-step ahead prediction:** In this type, a next single value must be predicted from the available data. This process must be repeated over the complete prediction horizon. For example assume a prediction horizon of 5 days. Using the TSD values from day 1 to day 10 prices we predict day 11 price. Again using day 2 to day 11 price, we predict day 12 price. Continuing further, finally using day 5 to day 14 price values, we predict day 15 price. Then a total of 5 prices have been predicted from day 11 to day 15. This is the process of one-step prediction over the prediction horizon of 5 steps.

2. **N-step ahead prediction:** Here the next $N, N > 1$ values must be predicted from the available data unlike one-step prediction case. This is also called as multi-step ahead prediction, which should be repeated over the complete prediction horizon. It is further divided into two types:

   (a) **Direct forecast:** Let $N = 2$ and the prediction horizon be 10. Then using day 1 to day 10 TSD values, day $11, 12$ are directly predicted. Using day 3 to day 12 TSD values, day $13, 14$ TSD values are directly predicted. This process continues and finally using day 9 to day 18 TSD values, day $19, 20$ TSD values are

directly predicted. This type of forecasting is termed as direct two-step ahead forecasting.

(b) **Iterative forecast:** Let $N = 2$ and the prediction horizon be 10. Then using day 1 to day 10 TSD values, first day 11 TSD value is predicted. Then using day 2 to day 10 TSD values and the predicted day 11 value, the day 12 TSD value is predicted. This means unlike the direct case, here to predict day 12 TSD value, day 11 TSD value is not used. Next, using day 3 to day 12 TSD values, day 13 TSD value is predicted. Then using day 4 to day 13 TSD values and the predicted day 13 TSD value, the day 14 TSD is predicted. It continues all over the prediction horizon till we obtain day 20 forecast. This is termed as iterative two-step ahead forecasting.

In this thesis, we focus on iterative N-step ahead forecast, and one-step ahead forecasts. Whether it is one-step ahead or multi-step ahead prediction, based on the prediction horizon size, two types of prediction exist.

1. **Short-term prediction:** Here prediction horizon is only a small fraction of the total number of available TSD values.

2. **Long-term prediction:** Here prediction horizon is significantly large fraction of the total number of available TSD values.

1.3 TSD Prediction models

To predict or forecast any given TSD, prediction models are necessary.

Some of the available TSD prediction models are ARIMA, GARCH, ANN, Fuzzy model, spectral model, Markov model etc. In this thesis, ARIMA, ANN and GARCH and their variants are discussed.

1.3.1 ARIMA model

Auto Regressive Integrated Moving Average (ARIMA) is a linear modeling technique. It is a combination of AR, I and MA components. The steps for modeling and prediction are:

- **Stationarity check:** If the given data is already stationary, it is passed to the next step directly. Otherwise, a differencing operation is performed and checked if stationary. If the data is still non-stationary, differencing is again performed until the data is finally stationary. If the differencing is performed d times, the integration order of the ARIMA method is said to be d.

- **ARMA modeling:** The stationary data is modeled as ARMA time series as follows. The data value at any given time t, say y_t, is considered as a function of the previous p data values, say $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$, *and the errors at times* $t, t-1, \ldots, t-q$, *say* $n_t, n_{t-1},$ $\ldots, n_{t-q}$. *The corresponding ARMA equation is shown in (1.1).*

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots a_p y_{t-p} + n_t + b_1 n_{t-1} + \ldots b_q n_{t-q} \tag{1.1}$$

In (1.1), a_1 to a_p are the autoregressive (AR) coefficients and b_1 to b_q are the MA coefficients. Thus the time series model is denoted as ARIMA(p, d, q). The ARMA model assumes that the error sequence n_t is white noise and is Gaussian distributed, so the variance of this error series is also a model parameter. Thus ARIMA modeling procedure has following steps:

- Identifying the model orders p, q: It is done using correlation analysis [1] using the nature of the Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) given in 1.2 and 1.3 respectively which are functions of lag or delay k. If the ACF shows a sinusoidal or geometrical decay and simultaneously, PACF becomes zero after a lag p, it is a pure AR process of order p. If the ACF becomes zero after a lag q, and PACF has sinusoidal decay, then it becomes an MA process of order q. If both ACF and PACF have sinusoidal decay, and become zero after lags q and p respectively, then it becomes an ARMA process of order p, q and correspondingly ARIMA process of order (p, d, q).

$$r_k = \frac{1}{N}\left(\frac{\sum_{t=1}^{N-k}(y_t-m)(y_{t+k}-m)}{\sigma^2}\right), \quad \sigma^2 = \frac{1}{N}\sum_{t=1}^{N}(y_t-m)^2 \tag{1.2}$$

$$p_k = \frac{r_k - \sum_{j=1}^{k-1} p_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} p_{k-1,j} r_{k-j}}, \quad p_{k,j} = p_{k-1,j} - p_k p_{k-1,k-j}, \quad p_1 = r_1 \tag{1.3}$$

- Estimating the model coefficients: Using the Box-Jenkins method [1], the model coefficients can be computed. Of the various estimation approaches other than this nonlinear maximum likelihood approach, which is computationally more complex, Gaussian maximum likelihood estimation (GMLE) approaches are generally used for estimation of the ARIMA model parameters. After the model is estimated or fit on a time series data, it must be validated. This diagnosis check is based on the analysis of error series. By analyzing ACF of the this error series data, and checking if they are within the 99% confidence intervals or not, the model can be validated. Some other tests can be performed to validate the model without using the residual ACF. One such test is Ljung and Box test. Other criteria for checking model validity also exist like Akakine Information Criterion (AIC) and Bayesian Information Criterion (BIC). The best-suited ARIMA model has the minimum value of the AIC or BIC.

- **Forecasting the data:** Once the model is validated, it is used to forecast the TSD. Using all the estimated model coefficients and the available TSD values, the next values of the TSD are predicted. The differenced data must be integrated to get back the raw data predictions. So, this model is termed as Auto Regressive "Integrated" Moving Average (ARIMA) model. ARIMA models predict linear TSD with very good accuracy, as they are linear models.

1.3.2 GARCH Model

One of the required facts in ARIMA model is the data must be stationary. But in TSD originating from financial applications, the data exhibits volatile nature, where the conditional variance unlike for ARIMA modeled TSD, varies continually with time. This type of TSD are said to exhibit risk, where risk is measured in terms of volatility. If the TSD is highly volatile, ARIMA modeling is not a suitable model because the error series obtained from the model, will no longer be a Gaussian time series. This fact is shown in [2].

In such cases, if the conditional variance of the time series data at a time t, is modeled in terms of previous variances, such a modeling on conditional variances may be used to predict the time series data. This type of modeling is the basic principle behind ARCH models. ARCH models are introduced by Engle in 1982. Later, Bollerslev proposed a Generalized ARCH (GARCH) model, which is an equivalent version of higher order ARCH model. The GARCH modeling steps are illustrated below.

- **GARCH Modeling:** According to Bollerslev's GARCH model, the present conditional variance is modeled as a function of past variances and past values of squared innovations or residuals. The GARCH modeled time series data, is shown in (1.4), where c stands for a constant term, and ε_t stands for residual time series, which

is Gaussian distributed with zero mean and variance σ_t^2.

$$y_t = c + \varepsilon_t \tag{1.4}$$

In (1.4), $\varepsilon_t = \sigma_t z_t$ where $z_t = N(0,1)$ is the Gaussian random sequence with independent identically distributed (i.i.d) random variables of zero mean and unit variance. It is also called the innovations process. The variance σ_t^2 is modeled as in (1.5), for a GARCH (P,Q) process.

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{P} \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^{Q} \beta_i \varepsilon_{t-j}^2 \tag{1.5}$$

From (1.5), it can be seen that present conditional variance is represented as a function of past variances and past values of squared innovations. If P and Q are taken as 1, the model is the simplest one and is GARCH (1,1). If $P > 1$, $Q > 1$, that means the present variance at time t, is a function of previous variances at time $t-1, t-2$...,$t-P$ and also a function of Q previous squared residual values.

- **Parameter estimation and model validation:** The values of GARCH model parameters and model coefficients are estimated using Maximum Likelihood Estimation. In this estimation, the model parameters are computed by maximizing a Quasi Likelihood function.Using these model parameters, the GARCH model predictions

can be obtained. If the given time series data is not having Gaussian distribution, to account for the fatter tails, the student-T distribution is used with GARCH model. The estimation process slightly changes and is detailed in [3]. Once the model is obtained, the model needs to be validated. Validation tests like AIC, Schwartz Bayesian Criterion (SBC), LM Arch test, Ljung and Box test, Box and Pierce test are performed. The GARCH model can also be validated by using the ACF and PACF of the residual errors obtained during model parameter estimation.

- **Prediction:** After validation, the model can be used in the prediction of future volatility values. In ARIMA, once the model parameters are obtained, the model is ready to be used for the prediction of future TSD values. On the other hand, in GARCH after the parameter estimation step is complete, only the future conditional variances can be predicted but not the future values. To obtain the future forecasts, a gaussian i.i.d with zero mean and unit variance is simulated and correspondingly its variance is made σ_t^2 using $\varepsilon_t = \sigma_t z_t$. σ_t^2 is obtained by substituting the estimated parameters in (1.5). Using the parameter c and the simulated time series ε_t, the GARCH predictions y_t are obtained. But y_t can be any value from a possible set of values. So using Monte Carlo simulations the prediction performance accuracy can be computed unlike the ARIMA based methods.

1.3.3 ANN

Artificial neural networks (ANN) are a nonlinear prediction modeling technique which is suitable for modeling TSD originated form a very wide range of applications. It is more flexible in terms of architecture. The neural-network architecture bears a high similarity to the neurons in the brain, hence the name artificial neural network. The neurons are processing units which are acyclically linked.

Network architecture

In ANN architecture, there may be two or more layers. Three-layer ANNs are widely used in forecasting. A typical three-layer ANN has three layers:

- Input layer: Number of neurons in this layer corresponds to the number of inputs to the ANN. This layer consists of nodes, which do not involve in the actual signal modification; the nodes just transfer the signal to the next layer.
- Hidden layer: This layer has flexible number of layers with arbitrary number of nodes or neurons. The nodes in this layer take part in the signal modification, and are therefore called active nodes. Each active node takes an active part in the modeling.
- Output layer: The number of neurons or nodes in the output layer corresponds to the number of the output values of the ANN, which is generally one for TSD prediction applications. The nodes in this

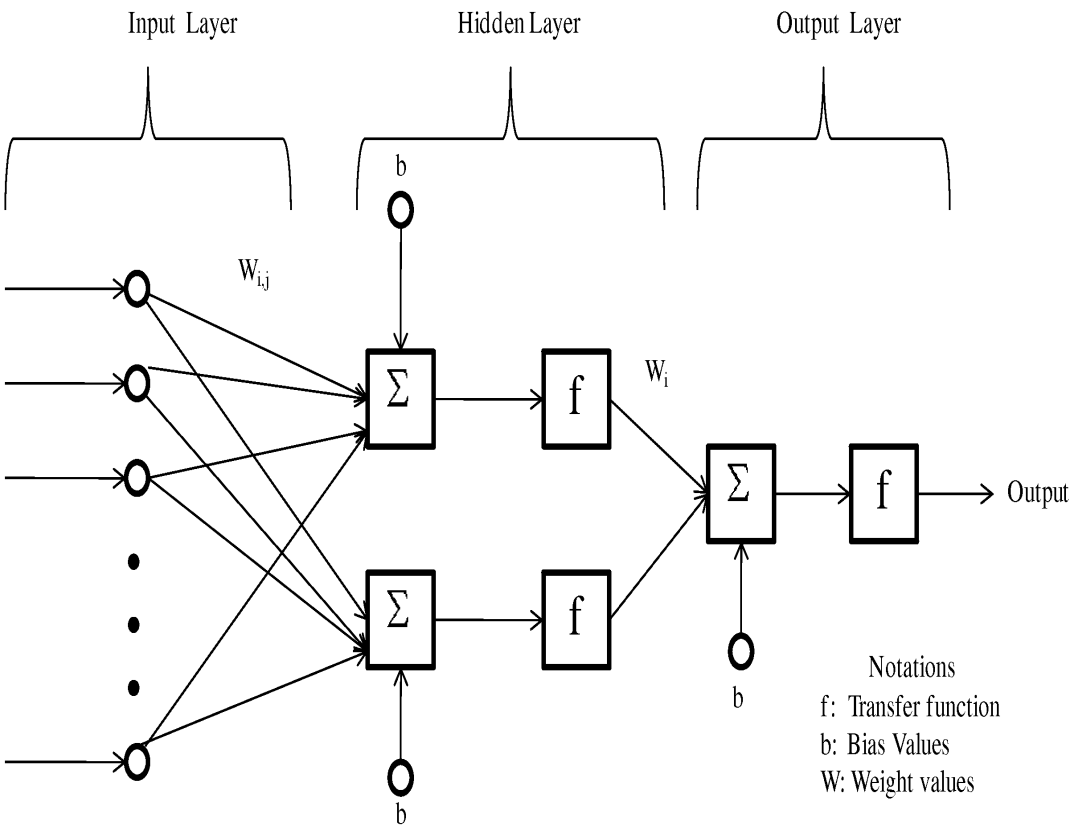

Figure 1.6: Three-layer ANN architecture

layer are also active nodes like those in hidden layer.

A typical three-layer ANN, with its nomenclature, is shown in Figure 1.6. ANN^{i-h-o} represents an ANN architecture with i inputs, h hidden layer neurons, and o outputs. Each layer is characterized by a unique transfer function f shown in the figure.

A transfer function is the key element to mathematically show the non-linear relationship between the inputs and the output. Many types of transfer functions exist, out of which linear, sigmoid and tan sigmoid transfer functions are commonly used.

TSD modeling using ANN

To model a given TSD using such a network architecture, consider the sequence y_t. It's a nonlinear function of previous TSD values, $y_{t-1}, \ldots, y_{t-N}$. The corresponding model equation is given in (1.6). In (1.6), the function g is a nonlinear function, and v_t is a noise or error term.

$$y_t = g(y_{t-1}, y_{t-2}, \ldots, y_{t-N}) + v_t \tag{1.6}$$

The ANN model output can be represented in terms of input and hidden-layer weight parameters. The transfer function of the hidden layer is generally a sigmoid function and that of the output layer has a linear transfer function in TSD prediction applications. The model parameters in an ANN are the weights of each link and the bias values, as shown in Figure 1.6. Weight represents the strength of connection between the two linked neurons. Bias is an optional value which can be added to scale the strength of the summer output in Figure 1.6.

Model parameter estimation: To estimate the model parameters, a specified optimization and computation of the parameters does not exist in ANN, as it involves non-linear equations to be solved. So, the

parameter estimation is carried out using training algorithms. Consider a known data sequence or a training data given as input to the ANN. The ANN is told what should be the corresponding output in the prediction application. Correspondingly, the ANN is trained by considering the minimization of the multivariate global error function formed by the weight values. Either the absolute error or the squared error can be minimized. The weights and the bias values keep changing through out the training period, as the TSD values keep changing. Finally the weights and the bias values converge at the end of the training period. These form the final model parameters. Many training algorithms are available in the literature [4]. For example, in [5], a reduced gradient-based algorithm was used for training the ANN. In [6] and [7], a scaled conjugate gradient algorithm and a Levenberg-Marquardt (LM) training algorithm, respectively, were incorporated.

Model validation and prediction Training of the ANN is carried out in iterations. As the iteration number increases, the sum of the squared errors should decrease. The validation set errors also should follow a similar trend. Once the model is validated, it is used for TSD prediction.

1.3.4 Performance Measures

To compare any two models, some error performance measures most commonly used are:

1. Mean Absolute Percentage Error (MAPE): Computing the absolute error in prediction as a fraction of actual TSD value, and averaging

such error fractions over the prediction horizon, we obtain MAPE (1.7). In (1.7), $pf - pi + 1$ is called prediction horizon, $y_{i,\text{actual}}$ is the actual value of the TSD and $y_{i,\text{predicted}}$ is the predicted value of the TSD.

$$MAPE = \frac{1}{pf - pi + 1}\left(\sum_{i=pi}^{pf}\left|\frac{y_{i,\text{actual}} - y_{i,\text{predicted}}}{y_{i,\text{actual}}}\right|\right) * 100 \tag{1.7}$$

2. Maximum Absolute Percentage Error (MaxAPE): Computing the absolute error in prediction as a fraction of actual TSD value, and identifying the maximum of all such values over the prediction horizon, we obtain MaxAPE (1.8).

$$MaxAPE = \max\left(\left|\frac{y_{i,\text{actual}} - y_{i,\text{predicted}}}{y_{i,\text{actual}}}\right|\right) * 100, i \in [pi, pf] \tag{1.8}$$

3. Mean Squared Error (MSE): The average of all the squared errors over the prediction horizon is called MSE given in (1.9).

$$MSE = \frac{1}{pf - pi + 1}\left(\sum_{i=pi}^{pf}|y_{i,\text{ actual}} - y_{i,\text{ predicted}}|^2\right) \tag{1.9}$$

4. Root Mean Square Error (RMSE): It is the square root of the MSE and is given in (1.10).

$$RMSE = \sqrt{\frac{1}{pf - pi + 1}\left(\sum_{i=pi}^{pf}\left(y_{i,actual} - y_{i,predicted}\right)^2\right)} \tag{1.10}$$

5. Mean Absolute Error (MAE): It is the average of all the absolute errors over the prediction horizon and is given in (1.11).

$$MAE = \frac{1}{pf - pi + 1}\left(\sum_{i=pi}^{pf}|y_{i,\text{ actual}} - y_{i,\text{ predicted}}|\right) \tag{1.11}$$

In many cases, the MAPE between two different models may be nearly same, but MSE and MAE would show a drastic change. To evaluate the performance of a prediction model, the prediction horizon must be sufficiently long. If it is too small, the expectation, i.e., the mean or average value will not be accurate enough, and if it is very long, the time taken for processing will be high. So, the prediction horizon must be chosen appropriately. Any prediction model primarily targets better prediction accuracy. However, in applications like finance and economic conditions of country, the relationship of the data with its past and the nature of dependency, i.e. the data trend or dynamics over a finite horizon (short or long) need to be devised.

1.4 Decomposition based prediction models

Many a time, for the prediction of TSD, a pre-processing technique may help in improving prediction accuracy. Many types of pre-processing methods are available out of which decomposition technique is chosen in this work. Decomposition is the process of splitting the original or given TSD into one or more components. The strategy for the split determines the nature of each decomposition. One particular strategy for decomposition is filtering. A moving-average (MA) filter based decomposition and wavelet based decomposition are considered in this thesis work, which are described below.

MA-filter based decomposition: Consider a given non-seasonal

TSD, on which MA filter based decomposition is performed. This decomposition results in two components; averaged or smoothened component called trend and a residual or noise component, also termed as detrended component. This decomposition is shown in Figure 1.7. Given a TSD y_t, the trend component is obtained as given in (1.12). The noise component is obtained using (1.13). The length of the MA filter, m can be chosen to meet a specific decomposition criterion, for a given application.

$$\bar{y}_t = \frac{1}{m} \sum_{k=t-m+1}^{t} y_k \tag{1.12}$$

$$y_{noi,t} = y_t - \bar{y}_t \tag{1.13}$$

The MA filter based decomposition is applied on the SBI close price TSD, and the resulting decompositions for various filter length values are shown in Figure (1.8), Figure (1.9) and Figure (1.10). In Figure (1.8), the length of the filter is chosen as 5. Tn this case the trend is not much different from the given TSD. As the filter length increases to 10 in Figure (1.9), the trend becomes more smooth, and when the filter length still increases to 20, the trend is shown in Figure (1.10). Corresponding residual component in these three cases is given by (1.13). The above discussed basics provide sufficient introduction to the thesis.

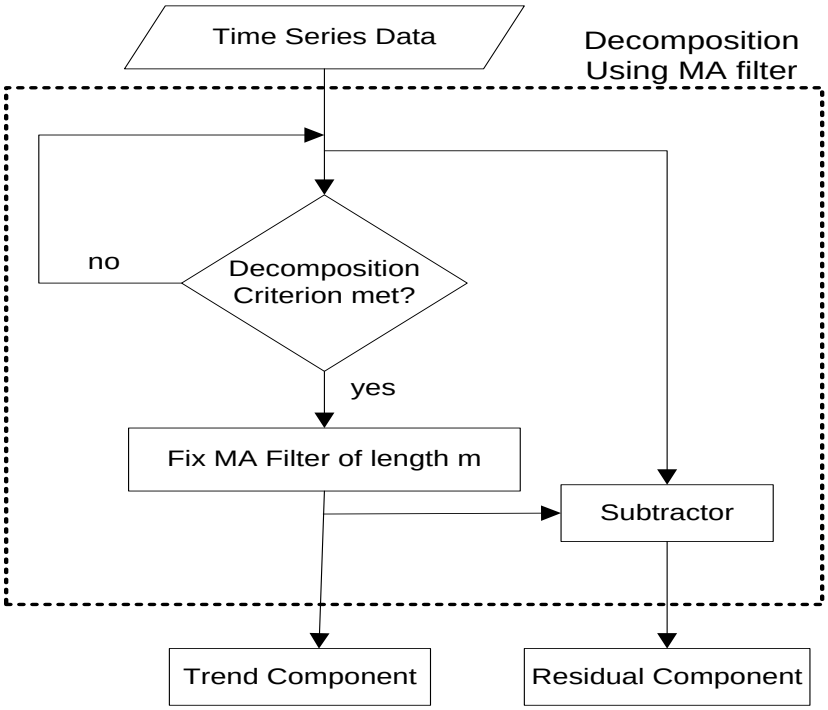

Figure 1.7: Time Series decomposition using MA Filter

1.4.1 Trend-ARIMA model

This prediction model is a composite technique where, MA-filter based decomposition is used as a pre-processing step before fitting the ARIMA model on the data. The method is outlined as follows:

- **Decomposition:** The MA filter based decomposition is applied on the raw data as introduced in 1.4. Correspondingly the trend component s_t, and the noise component r_t are obtained. Note that $y_t = s_t + r_t$ is the original TSD.

- **ARIMA modeling:** On each of the obtained decompositions trend and noise from step 1, ARIMA model is fit as outlined in section 1.3.1. The ARIMA model on the trend component need not be same as that on the noise component.

- **Predictions:** By using the ARIMA model fit on the trend component, the predictions on trend component are obtained. Let them be represented as $s_{t,pre}$. Using the ARIMA model fit on the noise component, the predictions on the noise component are obtained. Let them be represented as $r_{t,pre}$. The final data predictions are obtained by summing up the trend predictions and the noise predictions as in (1.14). The thus obtained predictions have greater accuracy compared to predictions from basic-ARIMA.

$$y_{t,pre} = s_{t,pre} + r_{t,pre} \tag{1.14}$$

1.4.2 Wavelet-ARIMA model

This is another composite prediction model, where wavelet based decomposition is applied as a pre-processing step prior to fitting ARIMA model and obtaining the predictions. Instead of moving average filters, using wavelet filters, such as Haar, db1, db2, db3, db4 or db5, the data is decomposed into approximate and detail components. This decomposition can be of multiple levels as illustrated in Figure 1.11. In the first level, the time series data is filtered using any one of the wavelet filters. Correspondingly the low frequency content is present in the approximate component $ya1$ and the high frequency content is present in the detailed component, $yd1$. The approximate component is further filtered and the next level decompositions are $ya2$ and $yd2$. $ya2$ comprises of the low frequency content of $ya1$, and $yd2$ comprises of the high frequency content

in $ya1$. After these two levels of decompositions, the final components are $ya2$, $yd1$ and $yd2$. The $ya2$ time series represents the final approximate component, while $yd1$ and $yd2$ are the final detailed components. This decomposition can be further continued to a suitable number of levels based on type of the time series data. Based on this decomposition, the wavelet-ARIMA model is illustrated as follows:

- **Decomposition:** Using the wavelet decomposition illustrated before, the raw data is decomposed into suitable number of decompositions based on the nature of given TSD. Let the decompositions after a two level decomposition be $ya1, yd1, yd2$.

- **ARIMA modeling:** On each of the decompositions $ya1, yd1, yd2$, the ARIMA model is fit individually and the best model on each of the decomposed TSD, $ya1, yd1, yd2$ are found out using the steps illustrated in section 1.3.1.

- **Prediction:** The original data predictions are then computed by summing the predictions of all the decomposed components as in equation 1.15. The performance of this method is better than the ARIMA model [8], [9],[10].

$$y_{t,pred} = ya_{2,pred} + yd_{2,pred} + yd_{1,pred} \tag{1.15}$$

This method is an extension of the trend-based ARIMA. Trend and noise data are like decompositions of raw data. As trend is smooth version, it is similar to the low frequency component of raw data and the noise

data comprises of the high frequency component of raw data. Trend and residual data are like first level decompositions of the raw data. Trend can be further decomposed and the method can be extended. The difference between this method and the Trend-based ARIMA method is that the filter used here is not a simple moving average filter but a db5 filter. So this method has a greater accuracy in predicting than trend-based ARIMA. Nevertheless, trend-based ARIMA has a better accuracy than basic ARIMA.

1.5 Motivation and scope of the work

TSD prediction has various applications as already discussed. Owing to the importance of prediction, there is a necessity to develop accurate prediction models, which suit various applications. Many-a-times a single model may not be accurate in all applications. The model accuracies for one-step ahead and multi-step ahead predictions are different. Similarly, a model which is accurate for short term prediction need not retain same accuracy as the forecast horizon becomes long. Other than prediction accuracy, if target is a multi-step prediction with large horizon, the data trend or dynamics need to be maintained over the complete period. In this case both prediction accuracy as well as preserving data trend are equally important. In any case, development of accurate and apt prediction models is an important research area for diversified applications.

The scope of this thesis work is directed to develop accurate and apt

prediction models incorporating ARIMA, ANN and GARCH techniques, for the purpose of one-step ahead and multi-step ahead predictions. Correspondingly some variants of ARIMA, GARCH and ANN models are developed. The first variant involves ARIMA and ANN models, which is a non-linear hybrid model. It targets both one-step and multi-step ahead predictions and suits many TSD prediction applications. The second variant involves ARIMA and GARCH models, which is a linear hybrid model and targets multi-step ahead prediction. Hence, the model should render higher prediction accuracy and also maintain data trend over the complete forecast horizon. The developed variants and some of the existing models are applied on TSD originating from many applications and these studies are all reported with relevant results in this thesis work. The models developed and studied in this thesis are limited to univariate models, owing to the higher complexities involved in the development and usage of multivariate models.

1.6 Organization of the thesis

Chapter 1 introduced the meaning and graphical representation of TSD. Some of the data mining concepts useful in predictive mining of TSD, are elucidated. Various steps in the application of prediction models used in this research, namely ARIMA, GARCH and ANN detailed. Variants of ARIMA model like Trend-ARIMA and Wavelet-ARIMA are described in detail. Various validation tests used in ARIMA and GARCH models are also discussed in detail. The significance of decomposing time series

data along with MA filter based decomposition is explained and demonstrated.

Chapter 2 presents a detailed literature survey on prediction of TSD. TSD originating from different applications, various prediction models used on them are discussed in detail. Comparison of different prediction models on a given TSD and the results obtained there-by are highlighted in this chapter. Starting from the applications of prediction models like ARIMA, till GARCH and the popular ANN are all surveyed in this chapter. Other than these models, existing hybrid models are also discussed along with their applications. A brief mention of models other than ARIMA, GARCH and ANN is also provided in this chapter. However, the thesis work focuses ARIMA, GARCH and ANN models, along with their hybrids and variants.

Chapter 3 explains proposed hybrid model-1: A MA filter based hybrid ARIMA-ANN model for forecasting TSD. The steps involved in modeling and forecasting TSD using this model is discussed in detail. The nature of volatility is explored using a moving-average filter, and then an ARIMA and an ANN model were suitably applied. This model is applied on a simulated data set and experimental data sets such as sunspot data, electricity price data, and stock market data and evaluated using performance measures discussed in chapter 1, for both one-step and multi-step ahead cases. Some of the recently proposed hybrid ARIMA-ANN models of the literature are also described. The advantages of the proposed model over these existing ones are also discussed.

Chapter 4 compares and evaluates four different hybrid ARIMA-ANN models along with individual ARIMA and ANN on internet traffic data. Modeling and prediction of internet traffic is a crucial task, as the data is notorious for its volatility nature. On such data, an investigation on which of the existing hybrid ARIMA-ANN models gives best predictions, is carried out. Out of the compared models, namely our proposed MA filter based hybrid ARIMA-ANN model (2014), multiplicative model by Wang et.al (2013), Khashei and Bijari's hybrid ARIMA-ANN model (2011) and Zhang's hybrid ARIMA-ANN model (2003). The error performance measures showed that our proposed MA filter based hybrid ARIMA-ANN model gives best accuracy of all the four models compared and indeed is very suitable for predicting internet traffic for both one-step and multi-step ahead cases.

Chapter 5 presents a hybrid linear prediction model based on ARIMA and GARCH (proposed model -2). For multi-step ahead forecasting, model should preserve the data dynamics and render least model error. Such a model is devised using a hybrid of ARIMA and GARCH models with the help of a unique partitioning and interpolation technique. Detailed description of the proposed model is presented. The advantages and limitations of the proposed model are also explained. A qualitative analysis of this model is presented explaining why this model is more efficient compared to other individual models. On Indian stock market data, this model is evaluated along with some other existing models and least error measures are obtained with the proposed model. Also, the

proposed model preserved the data trend much better than the other models, hence being suitable for multi-step prediction.

Chapter 6 discusses the application of ARIMA and variants of ARIMA models on some selected TSD. For the prediction of average global temperature, the application of ARIMA, trend-based ARIMA and wavlet-based ARIMA are explored. For all the three models the error performance results are compared. On rainfall data, the results of applying ARIMA, GARCH and ANN are presented.

Chapter 7 presents the conclusions and future scope of this research work. The thesis contributions namely the two proposed models, MA filter based hybrid ARIMA-ANN model and the hybrid ARIMA-GARCH models are summarized. The various findings of the thesis with regard to prediction of TSD originating from various applications are also mentioned.

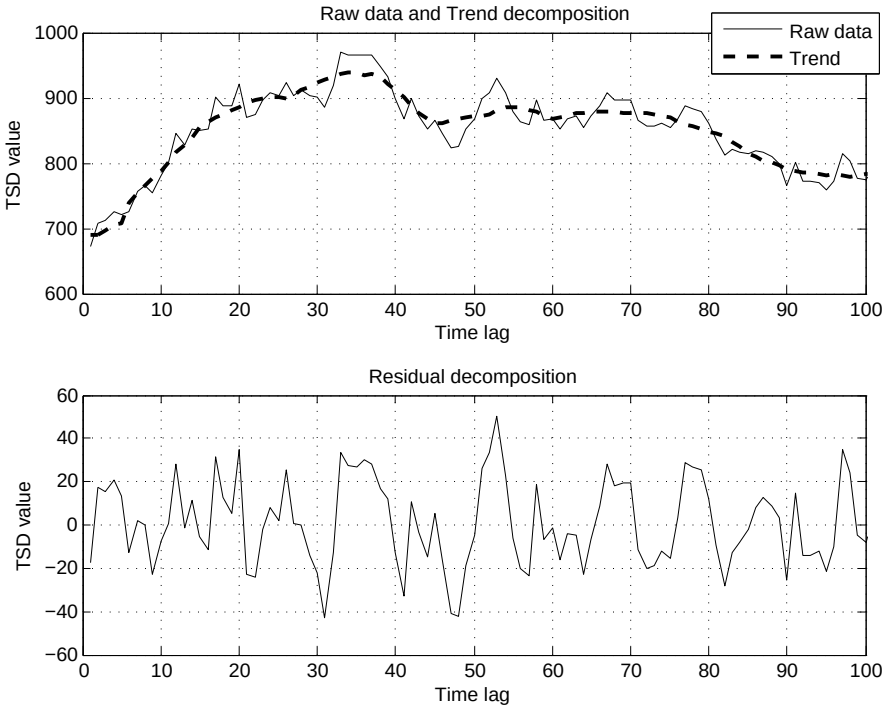

Figure 1.8: Decomposed SBI close price TSD with $m = 5$

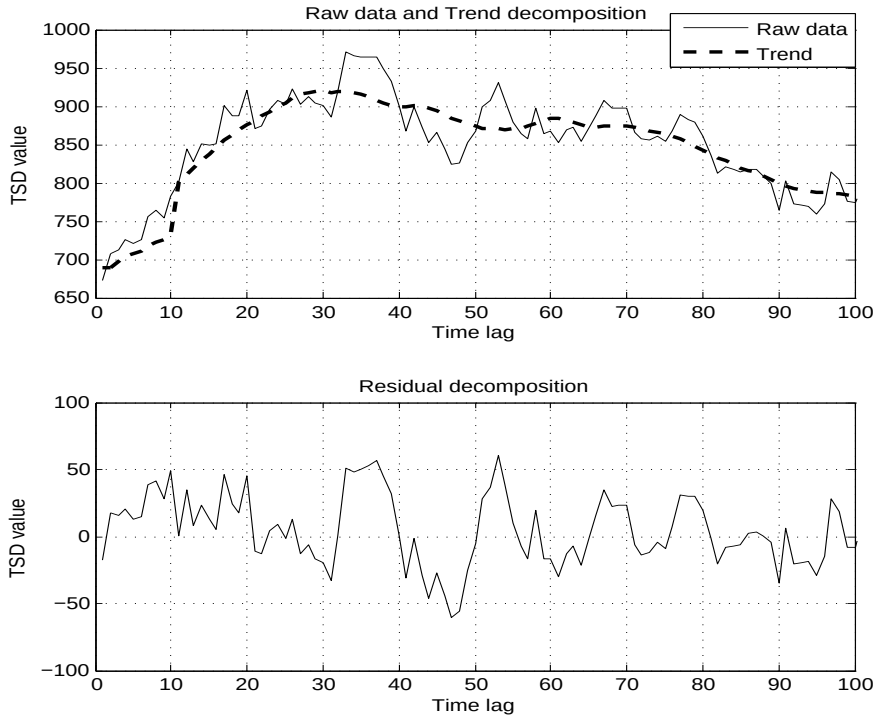

Figure 1.9: Decomposed SBI close price TSD with $m = 10$

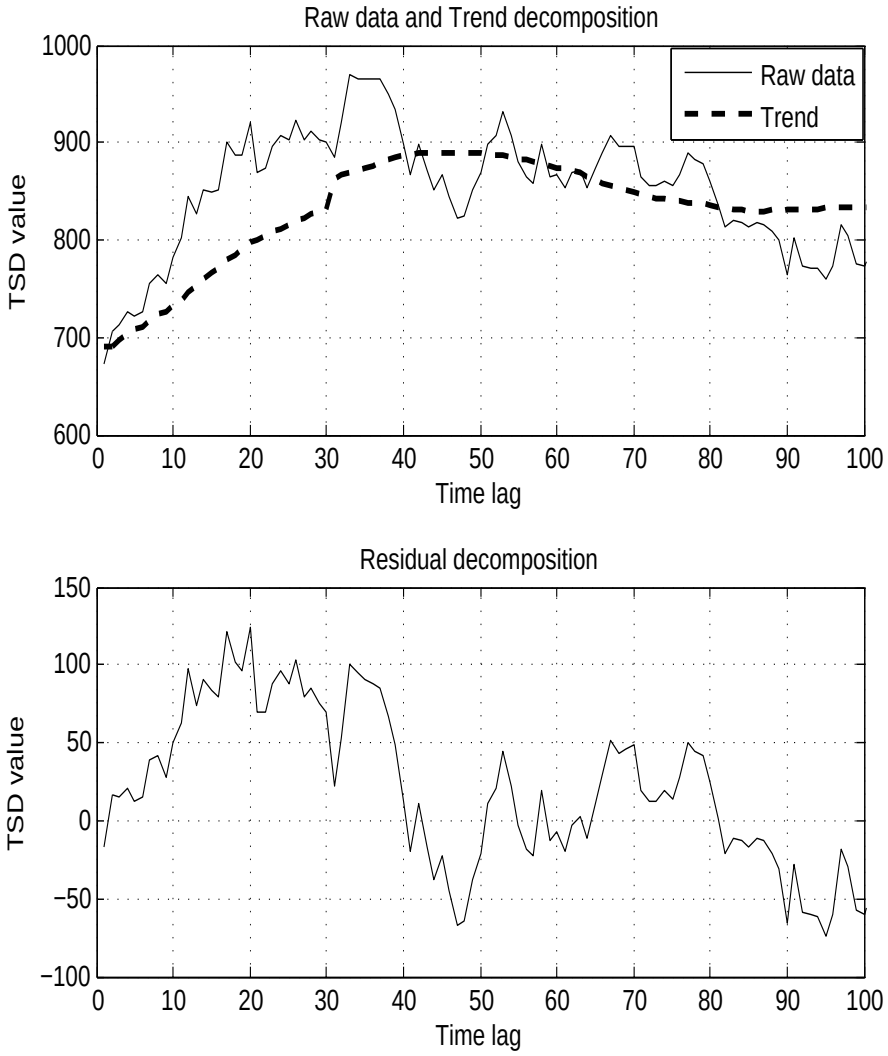

Figure 1.10: Decomposed SBI close price TSD with $m = 20$

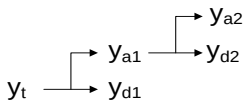

Figure 1.11: Wavelet Decomposition