Thesis-SS14 2501

A HYBRID TRANSFER FUNCTION AND ARTIFICIAL NEURAL NETWORK MODEL FOR TIME SERIES FORECASTING

FAHMI

(1314201210)

Supervisors

1. Dr. BRODJOL SUTIJO S.U., M.Si.

2. Dr. SUHARTONO

Master Program

Department of Statistics

Faculty of Mathematics and Natural Sciences

Institut Teknologi Sepuluh Nopember

Surabaya 2016

A HYBRID TRANSFER FUNCTION AND ARTIFICIAL NEURAL NETWORK MODEL FOR TIME SERIES FORECASTING

The Thesis is submitted in partial fulfillment of the requirements for the degree of Megister Sains (M.Si.)

at
Institut Teknologi Sepuluh Nopember

by

FAHMI
NRP: 1314201210

Examination Date : January 28, 2016
Date of Graduation : March, 2016

Approved by:

1. **Dr. Brodjol Sutijo S.U., M.Si.** **(Supervisor)**
   NIP: 19660125 199002 1 001
2. **Dr. Suhartono** **(Co-Supervisor)**
   NIP: 19710929 199512 1 001
3. **Dr. Kartika Fithriasari, M.Si** **(Internal Examiner)**
   NIP: 19691212 199303 2 002
4. **Dr. rer. Pol. Dedy Dwi Prastyo, M.Si.** **(Internal Examiner)**
   NIP: 19831204 200812 1 002

Director of Magister Program,

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
PROGRAM PASCASARJANA

Prof.Ir. Djauhar Manfaat, M.Sc, Ph.D.
NIP: 19601202 198701 1 001

Contents

List of Tables

This page is intentionally left blank

List of Figures

This page is intentionally left blank

List of Abbreviations

ARIMA	Autoregressive Integrated Moving Average
ACF	Autocorrelation Function
CPI	Consumer Price Index
CCF	Cross Correlation Function
HP	Hybrid Prediction
MAD	Mean Average Deviation
PACF	Partial Autocorrelation Function
RBF	Radial Basis Function
RMSE	Root Mean Square Error

This page is intentionally left blank

A HYBRID TRANSFER FUNCTION AND ARTIFICIAL NEURAL NETWORK MODEL FOR TIME SERIES FORECASTING

By	:	FAHMI
Student Identity Number	:	1314201210
Supervisors	:	1. Dr. BRODJOL SUTIJO S.U., M.Si.
	:	2. Dr. SUHARTONO

ABSTRACT

In finance and banking, the ability to accurately predict the future cash requirement is fundamental to many decision activities. In this research, we study time series forecasting of cash inflow and outflow requirements as the output series of Indonesian Central Bank (BI) at the representative offices in Aceh Province, Indonesia. We use a Consumer Price Index (CPI) as the leading indicator of the input series to predict the output series. A CPI measures change in the price level of a market basket of consumer goods and services purchased by households. CPI has been used in time series modeling as a good predictor for response. In this study, we propose a hybrid approach to forecast the cash inflow and outflow by combining linear and non linear models. This methodology combines both Transfer Function and Radial Basis Function (RBF) Neural Network models. The idea behind this model is the time series are rarely pure linear or nonlinear parts in practical situations. The RBF neural network is used to develop a prediction model of the residual from Transfer Function model. The RBF Neural Networks model is trained by Gaussian activation function in hidden layer. The main concept of the proposed hybrid model approach is to let the Transfer Function forms the linear component and let the neural network forms the nonlinear component and then combine the results from both linear and nonlinear models. This combination model provides a better forecast accuracy than the individual linear or non linear model.

Keywords: Inflow, Outflow, Consumer Price Index, Transfer Function, Artificial Neural Network, Radial Basis Function, Gaussian Function, Stationary, ARIMA, Time Series Forecasting, Hybrid Model.

This page is intentionally left blank

CHAPTER 1
INTRODUCTION

1.1 Background

Forecasting is an important part of decision making at all levels. In finance and banking, the ability to accurately predict the future cash requirement is fundamental to many decision activities. In this research, we study the time series forecasting of cash inflow and outflow as the output series of Indonesian Central Bank (BI) at representative offices in Aceh Province, Indonesia. Cash inflow is the transfer of money (in IDR) from other parties to Indonesian Central Bank (BI) in a given period of time through the deposit of commercial banks and the public transaction activities. Conversely, cash outflow is the money (in IDR) coming out from Indonesian Central Bank (BI) to other parties through the withdrawal process by commercial banks and public transaction activities. For the input series, we use a Consumer Price Index (CPI) as the leading indicator to predict the response series. A CPI measures change in the price level of a market basket of consumer goods and services purchased by households. CPI has been used in time series modeling as a good predictor for response variables.

Badan Pusat Statistics (BPS) of Indonesia calculated CPI based on monthly consumer price surveys on several cities in Indonesia (BPS, 2013). One of the factors that affect the change of price is the amount of money circulation. The quantity theory of economy states that increasing the amount of money in the economy will eventually lead to an equal percentage rise in the prices of products and services, if it is not followed by the growth of the real sector, which is called the inflation (Mankiw, 2000). Bank Indonesia (BI) as the central bank, has the task to regulate the amount of money circulation of the cash deposits by commercial banks to the Bank Indonesia (inflow) and withdrawals by banks from Bank Indonesia (outflow) in order to control inflation (Bank Indonesia, 2014). CPI is necessary for Bank Indonesia in making the monetary policy system, while for the government, especially local government, CPI is used as the basis for determining the amount of local government budget, budget planning, and other fiscal policies (Bank Indonesia, 2014). Fisher, Liu and Zhou (2002) stated that one of the important things for making monetary decisions is the forecasting of inflation. So far, BPS of Indonesia uses CPI as an indicator to measure the level of inflation in Indonesia. Therefore, the forecasting of inflation can be approached by modeling the forecasting of CPI.

The appropriate time series forecasting methodology plays an important role to meet the accurate future values. Time series modeling approach is the major techniques widely used in practice. Time series methods, on the other hand, use past or lagged values of the same criterion variable and model the relationship between these past observations and the future value. Autoregressive Integrated Moving Average (ARIMA) is one of the most important and popular linear models in time series forecasting a few decades ago. This model was popularized by Box and Jenkins (1976). Due to its statistical properties, ARIMA is also called the Box-Jenkins model particularly in the model building process. Although ARIMA models are the effective method to study time series with one time series in systems and widely used in time series forecasting, their major limitation is unable to express well the relationships in multivariate time series among variables in system (Zhang, 2003). So it is necessary to model a time series with multivariable models (Fan, Shan, Cao, and Li, 2009). Many different mathematical presentations of Transfer Function models can be found in the literature (Pektas and Cigizoglu, 2013). Transfer Function models have found extensive practical application. In Box and Jenkins methodology, ARIMA and Transfer Function model building follow the standard procedures.

Nowadays, Artificial Neural Network (ANN) is a promising tool in time series forecasting. The interest in neural networks is evident from the growth in the number of papers published in journals of diverse scientific disciplines (Zhang, 2004). The capability of neural networks have been widely used for a variety of assignments in different fields. In financial and business for example, Forecasting Foreign Exchange Rates, Equity Markets, Market Response, Stock Returns, and Market Indexes, have been studied and confirmed by Yao and Tan, (2002); Malliaris and Salchenberger, (2002); Parsons and Dixit, (2004); Thawornwong and Enke, (2004); Walczak, (2004) respectively. Likewise in robotics to control Spacecraft Formation by Wang, Min, Sun, and Zhang, (2010) and many other real world applications like Analysis of Groundwater by Ding, Liu, and Zhao, (2010), Wind Energy Forecasting by Campbell, Ahmed, Fathulla, and Jaffar, (2010), RBF by Mohammadi, Ghomi, Zeinali (2014), Traffic volume forecasting by Zhu, Cao, and Zhu (2014). Furthermore, Du, Leung, and Kwong (2014) shown that the distribution of the ANN in multi objective evolutionary algorithm produced the best forecast results.

In this study, we propose a hybrid approach to forecast the cash *inflow* and *outflow* using both Transfer Function and Radial Basis Function (RBF) Neural Network models. Numerous efforts have been introduced for developing and improving time series forecasting methods. In practical situations, time series are rarely pure linear or nonlinear components. However, the linear nature of the transfer function model is inappropriate to capture nonlinear structure of time series data, where the residuals of linear model contain information about the nonlinearity. Since it is difficult to comprehensively know the data characteristics whether a time series is generated from a linear or nonlinear process in a real world, hybrid methodology is proposed in practice (Zhang, 2003).

Makridakis and Hibon (2000) concluded in their paper of *M3-competitions: results, conclusions and implications* point 3 that combination is more accurate than the individual methods being combined for practically all forecasting horizons. Six years later, De Gooijer and Hyndman (2006) concluded in their paper of *25th years of time series forecasting* that combination method for linear and nonlinear models is an important practical issue that will no doubt receive further research attention in the future. Therefore, combining different models become so popular in practice due to the ability of capturing different patterns in the data and nice forecasting performance. Combining model in time series forecasting can significantly improve the forecasting performance over each individual model used separately (Zhang, 2003; Zhang, 2004; Robles *et al.*, 2008; Shukur and Lee, 2015; Khandelwal, Adhikari, and Verma 2015)

The basic idea of the proposed hybrid model approach is to let the transfer function forms the linear component and let the neural network forms the non linear component and then combine the results from both linear and non linear models. By combining Transfer Function with Radial Basis Function (RBF) Neural Network models, we can accurately model the complex autocorrelation structures in the data. Based on Zhang's hybrid model (Zhang, 2003 and 2004), the methodology of the hybrid system consists of two steps. In the first step, an Transfer Function is used to analyze the linear part of the problem. In the second step, an RBF Neural Network model is developed to model the residuals from the Transfer Function. Since the Transfer Function model cannot capture the non linear structure of the data, the residuals of linear model will contain information about the non linearity. The results from RBF Neural Network can be used as predictions of the error terms for the Transfer Function model.

This research paper contains five chapters, which are organized as follows: Chapter 2 provides a related literature review on the basic concepts of time series analysis like stationarity, differencing, the theory of Transfer Function, Neural Network, Hybrid model, Forecasting, etc. are also stated in this chapter. In chapter 3 we present the location of research area, data, variables, and research methods. Section 4 displays and discusses the empirical results and the conclusions are provided in the last section.

1.2 Problem Statement

Based on the background described above, we want to figure out the following problem formulations:

1. How to obtain the appropriate Transfer Function-noise model to forecast the cash inflow and outflow of Indonesian Central Bank (BI) at representative offices in Aceh Province by using input variable Consumer Price Index (CPI).

2. How to obtain the appropriate RBF Neural Network for forecasting residual of Transfer Function

3. How to formulate the Hybrid forecast (combination of linear and non linear parts) and how to analyze the forecasts accuracy among the individual Transfer Function, Neural Network, and Hybrid models.

1.3 Purpose of Study

The objective of this research is to forecast cash inflow and outflow of Indonesian Central Bank using the hybrid Transfer Function and Radial Basis Function Neural Network time series model. This approach is valuable because time series are rarely pure linear or nonlinear components in practical situations.

1.4 Limitation of Study

Although this research was carefully prepared, we still aware of its limitations and shortcomings:

1. Numerous methods for selecting forecast combination have been proposed. For instance, the simple linear combination, non linear combination, and the mixture of linear and non linear combination. In this research we only restrict to the combination of linear (Transfer Function) and non linear (Radial Basis Function Neural Network) methods.

2. In Transfer Function model, the input series play an important role to predict the output series. There are a plenty of input variables can be used and tested in Transfer Function modeling. However, due to time and resources constrains, we restrict to only use Consumer Price Index (CPI) as an independent input series.

This page is intentionally left blank

CHAPTER 2
LITERATURE REVIEW

The primary objective of time series analysis is to develop mathematical models that provide plausible descriptions for sample data. In this research, a time series $y_1, \ldots, y_T$ are assumed to be generated by a collection of random variables with probability structure (stochastic processes) $\{y_t\}_{t\in\mathtt{T}}$, where $\mathtt{T}$ is an index set containing the subset $1, \ldots, T$, the subscripts t are usually thought of as representing time or time periods (Lütkepohl and Krätzig, 2004).

The mathematical model which is used in this research is the Hybrid system of Transfer Function and Radial Basis Function Neural Network models. For the purpose of forecasting, the basic concepts of Autoregressive Integrated Moving Average (ARIMA) model is required due to its statistical properties as well as the well known Box and Jenkins methodology in the model building process. The other necessary concepts considered in this research are White Noise, Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), and Cross Correlation Function (CCF).

2.1 Autoregressive Integrated Moving Average (ARIMA)

Box and Jenkins (1976) and Wei (2006) define the autoregressive or self-regressive process of order p (AR(p)) as

$$\dot{y}_t = \varphi_1\dot{y}_{t-1} + \varphi_2\dot{y}_{t-2} + \cdots + \varphi_p\dot{y}_{t-p} + \varepsilon_t \tag{2.1}$$

or by using the lag operator

$$\varphi_p(L)\dot{y} = \varepsilon_t, \tag{2.2}$$

where $\varphi_p(L) = (1 - \varphi_1 L - \varphi_2 L^2 - \cdots - \varphi_p L^p)$, and $\dot{y}_t = y_t - \mu$. The analogy to the moving average process of order q, denoted as MA(q) can be written

$$\dot{y}_t = \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \cdots - \theta_q\varepsilon_{t-q} \tag{2.3}$$

or by using the lag operator

$$\dot{y}_t = \theta(L)\varepsilon_t, \tag{2.4}$$

where $\theta(L) = 1 - \theta_1 L - \theta_2 L^2 - \cdots - \theta_q L^q$.

A mixed ARMA process y_t with AR order p and MA order q (ARMA(p, q)) has the representation

$$\dot{y}_t = \varphi_1 \dot{y}_{t-1} + \varphi_2 \dot{y}_{t-2} + \cdots + \varphi_p \dot{y}_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{2.5}$$

or by using the lag operator can be written

$$\varphi_p(L)\dot{y} = \theta_q(L)\varepsilon_t, \tag{2.6}$$

A stochastic process y_t is called an ARIMA(p,d,q) process ($y_t \sim$ ARIMA(p,d,q)) if it is I(d) and the d times difference process has an ARMA(p,q) representation, that is $\Delta^d y_t \sim$ ARMA(p,q) (Lütkepohl and Krätzig, 2004). For practical purposes, d is usually equal to 1 or at most 2. It is important that the number of difference in ARIMA is written differently even though referring to the same model. Time series y_t is said to follow an integrated autoregressive moving average model if the dth difference $Wt = \Delta^d y_t$ is a stationary ARMA process (Cryer and Chan, 2008). For example, ARIMA(*1,1,0*) of the original series can be written as ARMA(*1,0,0*) of the differenced series.

2.2 Stationarity and Differencing

2.2.1 Stationarity

The basis for any time series analysis is stationary time series. A stochastic process y_t is called stationary if it has time-invariant first and second moments (Lütkepohl and Krätzig, 2004). In other words, y_t is stationary if

$E[y_t] = \mu_y$ for all $t \in T$ and

$E[(y_t - \mu_y)(y_{t-h} - \mu_y)] = \gamma_h$ for all $t \in T$ and all integers h such that t-$h \in T$.

We can roughly say, stationary time series describes no long-term trend, has constant mean and variance. More specifically, there are two definitions of stationary stochastic processes that are sometimes used in the literature: strictly stationary and weakly stationary.

A process Y_t is said to be *strictly stationary* if the joint distribution of $y_{t_1}, y_{t_2}, \ldots, y_{t_n}$ is the same as the joint distribution of $y_{t_1-k}, y_{t_2-k}, \ldots, y_{t_n-k}$ for all choices of time points $t_1, t_2, \ldots t_n$ and all choices of time lag k (Hamilton, 1994). A less restrictive requirement of stochastic process y_t is considered by Kirchgässner and Wolters (2007) since the concept of strict stationary is difficult to apply in practice. This process is called *weakly stationary* or *covariance stationary*. A stochastic process y_t having finite mean and variance is covariance stationary if for all t and t-s,

$$E[y_t] = E[y_{t-s}] = \mu_y$$

$$E[(y_t - \mu_y)^2] = E[(y_{t-s} - \mu_y)^2] = var(y_t) = var(y_{t-s}) = \sigma_y^2$$

$$\begin{aligned} E[(y_t - \mu_y)(y_{t-s} - \mu_y)] &= E[(y_{t-j} - \mu_y)(y_{t-j-s} - \mu_y)] \\ &= Cov(y_t, y_{t-s}) \\ &= Cov(y_{t-j}, y_{t-j-s}) = \gamma_s, \end{aligned}$$

where μ_y, σ_y^2, and all γ_s are constants (Box and Jenkins, 1976; Hamilton, 1994; and Enders, 2009).

For example, Hamilton (1994) shown that since the mean and autocovariances are not function of time, MA(*1*) process is weakly stationary regardless of the values of θ. In the case when $|\varphi| < 1$, AR(*1*) is also weakly stationary. In this research, we will use the term stationary as a weakly stationary where it is mean and covariance stationary.

2.2.2 Differencing

In practice, many time series are non-stationary and so we cannot apply stationary AR, MA or ARMA processes directly. One possible way of handling non-stationary series is to apply differencing so as to make them stationary (Chatfield, 2000). Differencing method governs the series to be lagged 1 step and subtracted from original series. For instance, $(y_t - y_{t-1}) = (1 - L)y_t$ may themselves be differenced to give second differences, and so on. The dth differences may be written as $(1 - L)^d y_t$.

The difference operator, *del*, is symbolized by the Δ. The first difference of y_t is given by the following expression: $\Delta y_t = (1 - L)y_t = y_t - y_{t-1}$. The second difference is expressed by $\Delta^2 y_t = \Delta(\Delta y_t) = (1-L)(1-L)y_t = (1-2L+L^2)y_t = (y_t - 2y_{t-1} + y_{t-2})$ (Yafee and McGee, 1999).

2.3 White Noise

A stationary process, where all autocorrelations are zero is called white noise or a white noise process. Hamilton (1994) and Wei (2006) said a process ε_t is a white noise process if it is a sequence of uncorrelated random variable from fixed distribution follows: $E(\varepsilon_t) = 0$, $E(\varepsilon_t^2) = \sigma^2$, and for which the $\varepsilon's$ are uncorrelated across time, $E(\varepsilon_t \varepsilon_\tau) = 0$ for all $t \neq \tau$. Since ε_t and ε_τ are independent (uncorrelated) for all $t \neq \tau$, then $\varepsilon_t \sim N(0, \sigma^2)$ which is also called Gaussian white noise process. The white noise process ε_t may be regarded as a series of shocks which drive the system (Box and Jenkins, 1976)

2.4 Correlations

2.4.1 Autocorrelations

Autocovariances and autocorrelations are addiction measurements between variables in a time series. Suppose that $y_1, \ldots, y_T$ are square integrable random variables, then Falk (2006) and Montgomery, Jennings, and Kulahci (2008) defined the **autocovariance** is the covariance between y_t and its value at another time period y_{t-k}

$$\gamma_k = Cov(y_t, y_{t-k}) = E[(y_t - \mu_y)(y_{t-k} - \mu_y)], \quad k = 0, 1, 2, \ldots, \tag{2.7}$$

with the property that the covariance of observations with lag k does not depend on t. The collection of the values of γ_k , k = 0, 1, 2, ... is called the **autocovariance function**. The autocovariance at lag $k = 0$ is just the variance of the time series; that is, $\gamma_0 = \sigma_y^2$. The autocorrelation between y_t and its another time period y_{t-k} can be defined as

$$\rho_k = Corr(y_t, y_{t-k}) \tag{2.8}$$

Thus, the autocorrelation coefficient at lag k is

$$\rho_k = \frac{E[(y_t - \mu_y)(y_{t-k} - \mu_y)]}{\sqrt{E[(y_t - \mu_y)^2]E[(y_{t-k} - \mu_y)^2]}} = \frac{Cov(y_t, y_{t-k})}{Var(y_t)} = \frac{\gamma_k}{\gamma_0}. \tag{2.9}$$

The collection of the values of $\rho_k, k = 0, 1, 2, \ldots$ is called the **autocorrelation function (ACF)**. In practice, the autocorrelation function from a time series of finite length $y_1, y_2, \ldots, y_T$ is estimated by the sample autocorrelation function (or sample ACF) r_k. By using $\overline{y} = \frac{1}{T}\sum_{t=1}^{T} = \mu_y$ as the sample mean, Lütkepohl and Krätzig (2004) compute the sample autocorrelations as follows

$$\widetilde{\rho}_k = \frac{\widetilde{\gamma}_k}{\widetilde{\gamma}_0}, \tag{2.10}$$

where

$$\widetilde{\gamma}_k = \frac{1}{T}\sum_{t=k+1}^{T}(y_t - \overline{y})(y_{t-k} - \overline{y}) \tag{2.11}$$

If the sample autocorrelations are mutually independent and identically distributed (*i.i.d.*) so that y_t and y_{t-k} are stochastically independent for $k \neq 0$, then the normalized estimated autocorrelations are asymptotically standard normally distributed, $\sqrt{T}\widetilde{\rho}_k \xrightarrow{d} N(0,1)$, and thus $\widetilde{\rho}_k \approx N(0, 1/T)$. Hence, $[-1,96/\sqrt{1/T}; 1,96/\sqrt{1/T}]$ is an approximate 95% confidence interval around zero, where the variance of the sample autocorrelation coefficient is $Var(\widetilde{\rho}_k) \cong 1/T$ and the standard error is $SE(\widetilde{\rho}_k) \cong 1/\sqrt{T}$. For detail properties of autocovariance and autocorrelation see (Wei, 2006).

2.4.2 Partial Autocorrelations

Lütkepohl and Krätzig (2004) define the partial correlation between y_t and y_{t-k} is the conditional autocorrelation given $y_{t-1}, y_{t-2}, \ldots, y_{t-k+1}$ as follows

$$\phi_{kk} = Corr[y_t, y_{t-k} | y_{t-1}, y_{t-2}, \ldots, y_{t-k+1}] \tag{2.12}$$

that is, the autocorrelation conditional on the in-between values of the time series. For an AR(p) model the partial autocorrelation function between y_t and y_{t-k} for $k > p$

should be equal to zero. A more formal definition can be found below (Box and Jenkins, 1976; Wei, 2006; Kirchgässner and Wolters, 2007; and Montgomery *et al.*, 2008). The Yule-Walker equations for the ACF of an AR(p) process for any fixed value of k is

$$\rho(j) = \sum_{i=1}^{k} \phi_{ik}\rho(j-i), j = 1, 2, \cdots, k \tag{2.13}$$

or

$$\begin{aligned}
\rho(1) &= \phi_{1k} + \phi_{2k}\rho(1) + \cdots + \phi_{kk}\rho(k-1) \\
\rho(2) &= \phi_{1k}\rho(1) + \phi_{2k} + \cdots + \phi_{kk}\rho(k-2) \\
&\vdots \\
\rho(k) &= \phi_{1k}\rho(k-1) + \phi_{2k}\rho(k-2) + \cdots + \phi_{kk}.
\end{aligned}$$

We can write the corresponding linear equation systems above for $k = 1, 2, \ldots$, in the matrix notation as

$$\begin{pmatrix} 1 & \rho(1) & \rho(2) & \cdots & \rho(k-1) \\ \rho(1) & 1 & \rho(1) & \cdots & \rho(k-2) \\ \rho(2) & \rho(1) & 1 & \cdots & \rho(k-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \cdots & 1 \end{pmatrix} \begin{pmatrix} \phi_{1k} \\ \phi_{2k} \\ \phi_{3k} \\ \vdots \\ \phi_{kk} \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \rho(3) \\ \vdots \\ \rho(k) \end{pmatrix} \tag{2.14}$$

or

$$\mathbf{P}_k \phi_k = \rho(k) \tag{2.15}$$

where

$$\mathbf{P}_k = \begin{pmatrix} 1 & \rho(1) & \rho(2) & \cdots & \rho(k-1) \\ \rho(1) & 1 & \rho(1) & \cdots & \rho(k-2) \\ \rho(2) & \rho(1) & 1 & \cdots & \rho(k-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \cdots & 1 \end{pmatrix}, \tag{2.16}$$

$$\phi_k = \begin{pmatrix} \phi_{1k} \\ \phi_{2k} \\ \phi_{3k} \\ \vdots \\ \phi_{kk} \end{pmatrix} \quad \text{and} \quad \rho(k) = \begin{pmatrix} \rho(1) \\ \rho(2) \\ \rho(3) \\ \vdots \\ \rho(k) \end{pmatrix}, \tag{2.17}$$

with Cramer's rule we have

$$\phi_{kk} = \frac{\begin{vmatrix} 1 & \rho(1) & \rho(2) & \cdots & \rho(k-2) & \rho(1) \\ \rho(1) & 1 & \rho(1) & \cdots & \rho(k-3) & \rho(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \cdots & \rho(1) & \rho(k) \end{vmatrix}}{\begin{vmatrix} 1 & \rho(1) & \rho(2) & \cdots & \rho(k-2) & \rho(k-1) \\ \rho(1) & 1 & \rho(1) & \cdots & \rho(k-3) & \rho(k-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \cdots & \rho(1) & 1 \end{vmatrix}} \tag{2.18}$$

For any given k, $k = 1, 2, \ldots,$ the last coefficient ϕ_{kk} is called the **partial autocorrelation** of the process at lag k. The PACF cuts off after lag p for an AR(p) since ϕ_{kk} = 0 for $k > p$. Sample partial autocorrelation function, $\widehat{\phi}_{kk}$ or some authors called the sample estimate of $\widehat{\phi}_{kk}$ is estimated as

$$\widehat{\phi}_{kk} = \frac{\widehat{\rho}(k) - \sum_{j=1}^{k-1} \widehat{\phi}_{k-1,j}\widehat{\rho}_{k-j}}{1 - \sum_{j=1}^{k-1} \widehat{\phi}_{k-1,j}\widehat{\rho}_j} \tag{2.19}$$

and

$$\widehat{\phi}_{k,j} = \widehat{\phi}_{k-1,j} - \widehat{\phi}_{kk}\widehat{\phi}_{k-1,k-j} \quad \texttt{for} \quad j = 1, 2, \cdots, k-1 \tag{2.20}$$

Furthermore, in a sample of T observations from an AR(p) process, $\widehat{\phi}_{kk}$, for $k > p$ is approximately normally distributed with

$$E[\widehat{\phi}_{kk}] \approx 0 \quad \text{and} \quad Var(\widehat{\phi}_{kk}) \approx \tfrac{1}{T}$$

Hence the 95% critical limits to test whether any $\widehat{\phi}_{kk}$ is statistically significant different from zero are given by $\pm 1,96/\sqrt{T}$. For further detail and example could be seen in Wei (2006).

2.4.3 Cross-Correlation Function

For the bivariate time series, x_t, y_t, the cross-correlation function measures the strength and the direction of correlation between two random variables. In Transfer Function model, the shape of the cross-correlation between those input and output series reveals the pattern of (r, s, and b) parameters of the transfer function (Box and Jenkins, 1976). The two stationary stochastic processes x_t and y_t, for $t = 0, \pm 1, \pm 2, \ldots$, have the following cross-covariance function

$$Cov(x_t, y_t) = \gamma_{xy}(k) = E\left[(x_t - \mu_x)(y_{t+k} - \mu_y)\right] \tag{2.21}$$

for $k = 0, \pm 1, \pm 2, \ldots$, where $\mu_x = E[x_t]$ and $\mu_y = E[y_t]$. Hence Wei (2006) defined the **cross-correlation function (CCF)** of the bivariate process as

$$\rho_{xy}(k) = \frac{\gamma_{xy}(k)}{\sigma_x \sigma_y} \tag{2.22}$$

for $k = 0, \pm 1, \pm 2, \ldots$, where σ_x and σ_y are the standard deviations of x_t and y_t. For a given sample of T observations of the bivariate process, the cross correlation function, $\rho_{xy}(k)$, is estimated by sample cross correlation

$$\widehat{\rho}_{xy}(k) = \frac{\widehat{\gamma}_{xy}(k)}{S_x S_y}, \tag{2.23}$$

where

$$\widehat{\gamma}_{xy}(k) = \begin{cases} \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \overline{x})(y_{t+k} - \overline{y}), & k \geq 0 \\ \frac{1}{n}\sum_{t=1-k}^{n}(x_t - \overline{x})(y_{t+k} - \overline{y}), & k < 0 \end{cases}, \tag{2.24}$$

where $n = T - d$ pairs of values $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, $S_x = \sqrt{\widehat{\gamma}_{xx}(0)}$ and $S_y = \sqrt{\widehat{\gamma}_{yy}(0)}$ available after differencing, and $\overline{x}$ and $\overline{y}$ are the sample mean of x_t and y_t respectively.

2.5 Transfer Function Model

The relationship between the input series, X_t, and the output series, Y_t, is a functional process. The response may be delayed from one level to another and distributed over a period of time. Such relationships are called dynamic transfer functions. A dynamic system may exist where an input series seems related to an output series. Box and Jenkins (1976) illustrated a dynamic system of input and output series with impulse response function as shown in Figure 2.1. Both the input and output series are time series and the output series is a function of the input series that is driving it.

Generally the transfer function models are formulated as $Y_t = v(L)X_t + n_t$. This model, which will be focused on this research, have two components. They are the Transfer Function component and the noise model component. The Transfer Function component consists of a response regressed on lagged autoregressive output variables and lagged input variables. ARIMA(p,d,q) is usually the underlying noise model for the Transfer Function. Then the residual of Transfer Function-noise model is modeled by using Radial Basis Function Neural Network model.

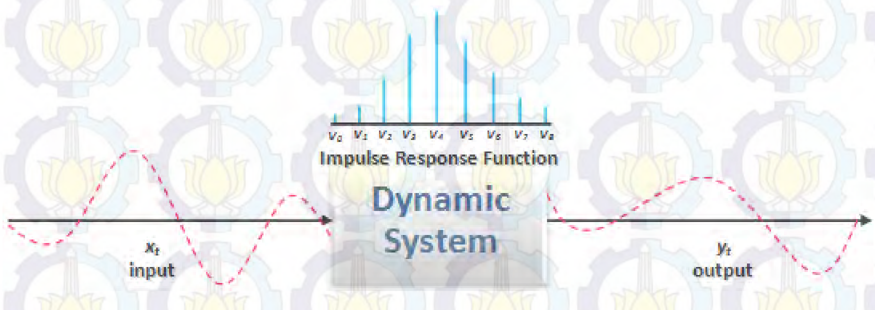


Figure 2.1: Input to, and Output from, a Dynamic System by Box and Jenkins

2.5.1 The Assumptions of the Single Input Model

The Transfer Function model, consisting of a response series, Y_t, a single explanatory input series, X_t, and an impulse response function $v(L)$, is established on the basic assumptions. The input series and the output series are stochastic, both series are assumed stationary (Box and Jenkins, 1976; Wei, 2006). Therefore centered and differenced (if necessary) is required to attain a stationarity condition.

For the input modeling, an error term which may be autocorrelated, is usually assumed to be white noise (Bisgaard and Kulahci, 2011). The represent of the unobservable noise process N_t, is assumed to be independent of the dynamic relationship between the output Y_t and the input X_t (Montgomery *et al.*, 2008). Moreover, Yafee and McGee (1999) presumed that this relationship is unidirectional with the direction of flow from the input to the output series. Therefore X_t in a Transfer Function must be input and the Transfer Function is assumed to be stable.

2.5.2 Prewhitening

Prewhitening is to make the input look like white noise. When the original input follows some other stochastic process, simplification is possible by prewhitening (Box-Jenkins, 1976). In other words, we remove the autocorrelation in the input series that caused the spurious cross correlation effect (Bisgaard and Kulahci, 2011). The first step in prewhitening involves identifying and fitting a time series model to the input x_t. The steps of prewhitening are outlined by Bisgaard and Kulahci in Figure 2.2.

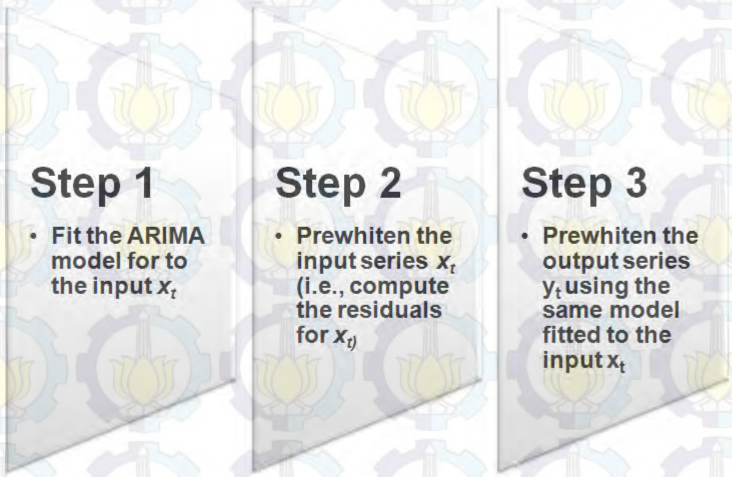


Figure 2.2: Prewhitening Step-by-Step

Suppose that x_t follows an ARIMA model, the filter of Transfer Function model when we applied to x_t generates a white noise time series, hence the name is prewhitening

(Montgomery *et al.*, 2008). For example, suppose that the sample of **ACF** and **PACF** indicate that the original input series X_t is non-stationary, but its differences $x_t = (1 - L)X_t$ are stationary. Then suppose the appropriate model might be a second order autoregressive AR(2) model

$$(1 - \varphi_1 L + \varphi_2 L^2)x_t = \varphi(L)x_t = \alpha_t \tag{2.25}$$

where $x_t = (1 - L)X_t$ and the α_t are the white noise process. After prewhitening the input, the next step is to prewhiten the output Y_t. The prewhitening filter for x_t will be used to filter the differences output series $y_t = (1 - L)Y_t$. This involves filtering the output data through the same model with the same coefficient estimates that we fitted to the input data. The prewhitened output is

$$(1 - \varphi_1 L + \varphi_2 L^2)y_t = \varphi(L)y_t = \beta_t \tag{2.26}$$

The process of transforming X_t to α_t and from Y_t to β_t via the filter $1 - \varphi_1 L + \varphi_2 L^2 = \varphi(L)$ is known as **whitening** or **prewhitening** (Cryer and Chan, 2008).

2.5.3 Identification of Transfer Function

The dynamic relationship between the output Y_t and the input X_t using the Transfer Function-Noise model for linear system is

$$Y_t = v(L)X_t + N_t, \tag{2.27}$$

where $v(L) = \sum_{j=0}^{\infty} v_j L^j$ is Transfer Function filter and N_t is the noise series that is assumed to be independent of X_t and generated usually by an ARIMA process. However in our case N_t will be generated by non linear neural network process. If the series exhibits non-stationarity, an appropriate differencing is required to obtain stationary x_t, y_t, and n_t. Hence, the Transfer Function-Noise model becomes

$$y_t = v(L)x_t + n_t \tag{2.28}$$

and

$$v(L) \approx \frac{\omega(L)}{\delta(L)} L^b \tag{2.29}$$

This imply that (2.27) can be written as

$$\begin{aligned} y_t &= v(L)x_t + n_t \\ &\approx \frac{\omega(L)L^b}{\delta(L)} x_t + n_t \\ &\approx \frac{\omega(L)}{\delta(L)} x_{t-b} + n_t, \end{aligned} \tag{2.30}$$

where $\omega(L) = \omega_0 - \omega_1 L - \cdots - \omega_s L^s$ and $\delta(L) = 1 - \delta_1 L - \cdots - \delta_r L^r$. The denominator $\delta(L)$ in (2.30) determines the structure of the infinitely order, therefore the stability of $v(L)$ depends on the coefficients in $\delta(L)$. As an infinite series, $v(L)$ converges as $|L| \leq 1$ (Yafee and McGee, 1999; Mongomery *et al.*, 2008). $v(L)$ is said to be stable under this condition, if all the roots of $m^r - \delta_1 m^{r1} - \cdots - \delta_r$ are less than 1 in absolute value. When the transfer function is absolutely summable, it converges and is considered to be stable, then

$$\sum_{L=0}^{\infty} |v(L)| < \infty \tag{2.31}$$

In the identification procedure, Wei (2006) derived the following simple steps to obtain the transfer function $v(L)$:

1. Prewhiten the input series, $\varphi_x(L)x_t = \theta_x(L)\alpha_t$

   $$i.e., \quad \alpha_t = \frac{\varphi_x(L)}{\theta_x(L)} x_t \tag{2.32}$$

   where α_t is a white noise series with mean zero and variance σ_α^2

2. Calculate the filtered output series. That is, transform the output series y_t using the above prewhitening model to generate the series

   $$\beta_t = \frac{\varphi_x(L)}{\theta_x(L)} y_t \tag{2.33}$$

3. Calculate the sample CCF, $\widehat{\rho}_{\alpha,\beta}(k)$, between α and β.The significant of the CCF can be tested by comparing it with its standard error $(n-k)^{-1/2}$

4. Identify $b, r, s, \omega(L), \delta(L)$ by matching the pattern of CCF. Once b, r, s are chosen, preliminary estimates $\widehat{\omega}_j$ and $\widehat{\delta}_j$ can be found from their relationships with v_k. Thus we have a preliminary estimates of the transfer function $v(L)$ as

$$\widehat{v}(L) = \frac{\widehat{\omega}(L)}{\widehat{\delta}(L)} L^b. \tag{2.34}$$

The noise series of the transfer function can be estimated by

$$\begin{aligned} \widehat{n}_t &= y_t - \widehat{v}(L)x_t \\ &= y_t - \frac{\widehat{\omega}(L)}{\widehat{\delta}(L)} L^b x_t. \end{aligned} \tag{2.35}$$

By examining the ACF and the PACF of $\widehat{n}_t$, we have

$$\varphi(L)n_t = \theta(L)a_t \tag{2.36}$$

By combining (2.34) and (2.36), the general form of transfer function-noise model is

$$y_t = \frac{\omega(L)}{\delta(L)} x_{t-b} + \frac{\theta(L)}{\varphi(L)} a_t. \tag{2.37}$$

2.5.4 The Structure of the Transfer Function

The impulse response weights v_j consist of a ratio of a set of s regression weights to a set of r decay rate weights, plus a lag level, b, associated with the input series, and may be expressed with parameters designated with r, s, and b subscripts, respectively. Box and Jenkins (1976) and Wei (2006) found the orders of r, s, and b and their relationships to the impulse response weight v_j by equating the equations of L^b in both sides of the equation

$$\delta(L)v(L) = \omega(L)L^b \tag{2.38}$$

and we obtain the identity

$$[1 - \delta_1 L - \cdots - \delta_r L^r][v_0 + v_1 L + v_2 L^2 + \cdots] = [\omega_0 - \omega_1 L - \cdots - \omega_s L^s]L^b \tag{2.39}$$

Thus we have

$$\begin{aligned}
v_j &= 0 & j &< b, \\
v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} + \omega_0 & j &= b, \\
v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} - \omega_{j-b} & j &= b+1, b+2, \cdots, b+s, \\
v_j &= \delta_1 v_{j-1} + \delta_2 v_{j-2} + \cdots + \delta_r v_{j-r} & j &> b+s.
\end{aligned}$$

The weights $v_{b+s}, v_{b+s-1}, \ldots, v_{b+s-r+1}$ supply r impulse response starting values for the difference equation

$$\delta(L)v_j = 0, \qquad j > b + s \tag{2.40}$$

In general, the impulse response v_j consist of

1. b zero values $v_0, v_1, \ldots, v_{b-1}$
2. $s - r + 1$ weights $v_b, v_{b+1}, \ldots, v_{b+s-r}$ that follows no fixed pattern
3. r starting impulse response weights $v_{b+s-r+1}, v_{b+s-r+2}, \ldots$, and v_{b+s}
4. v_j for $j > b + s$ follows the pattern given in (2.39)

Yafee and McGee (1999) define the order b, r, s as follows: The delay time b, sometimes referred to as dead time, determines the pause before the input begins to have an effect on the response variable $(L)^b X_t = X_{t-b}$. The order of the regression s designates the number of lags for unpatterned spikes in the transfer function. The number of unpatterned spikes is $s + 1$. The order of decay r represents the patterned changes in the slope of the function. The order of this parameter signifies the number of lags of autocorrelation in the transfer function. The denominator of the transfer function ratio consists of decay weights, δ_r from time $= 1$ to r. If there is more than one decay rate, the rate of contemplation may fluctuate.

2.5.5 Estimation of Transfer Function Model

Estimating a Transfer Function model involves judgment on the part of the researcher. Experienced econometricians would agree that the the procedure is a blend of skill, art, and perseverance that is developed through practice (Enders, 2009). If the Transfer Function follows the linear ARIMA noise model, by assuming the residual

a_t is $N(0,\sigma_a^2)$, it can be estimated by conditional least squares, unconditional least squares, or maximum likelihood.

In this research, the parameter for single input Transfer Function-noise model is estimated by using conditional least square. For the linear model part, the δ_i and ω_i can be recursively estimated by using $\widehat{\delta}(L)\widehat{v}(L) = \widehat{\omega}(L)$ from (2.41) (Montgomery *et al.*, 2008).

$$v_j - \delta_1 v_{j-1} - \delta_2 v_{j-2} - \cdots - \delta_r v_{j-r} = \begin{cases} -\omega_{j-b}, & j = b+1, \cdots, b+s \\ 0, & j > b+s, \end{cases} \tag{2.41}$$

with $v_b = \omega_0$ and $v_j = 0$ for $j < b$. Hence we have

$$\begin{aligned} \widehat{v}_1 &= \widehat{\omega}_0 \\ \widehat{\omega}_2 - \widehat{\delta}_2 \widehat{v}_1 &= 0 \\ &\vdots \end{aligned} \tag{2.42}$$

2.5.6 Diagnostic Checking of Transfer Function Model

To test the adequacy of a model, the estimated parameters should be significant. If the parameters are not significant, we drop them out from the model. In diagnostic checking step, Montgomery *et al.* (2008) checked the validity of two assumptions in the fitted model. These assumptions are: the noise a_t is white noise by examining the residual $\widehat{a}_t$ through ACF and PACF pattern and the independence between $\widehat{a}_t$ and x_t by observing the sample cross-correlation function between the residual α_t from the prewhitened input and x_t. To see whether these assumptions hold, Wei (2006) examined the residual $\widehat{a}_t$ from the noise model as well as the residual α_t:

1. *Cross correlation check.* For an adequate model, the sample CCF, $\widehat{\rho}_{\alpha,\widehat{a}}(k)$, between $\widehat{a}$ and α should show no patterns and lie within their two standard errors $2(n-k)^{-1/2}$. The Portmanteau test Q_0 which approximately follows a χ^2 distribution with $(K+1)-M$ degree of freedom can be used to check this assumptions.

$$Q_0 = m(m+2)\sum_{j=0}^{K}(m-j)^{-1}\widehat{\rho}^2_{\alpha,\widehat{a}}(j), \tag{2.43}$$

where $m = n - t_0 + 1$, $t_0 = max\{p + r + 1, b + p + s + 1\}$, M is the number of parameters δ_i and ω_j, $j = 0, 1, 2, \ldots, K$, and n is the number of residuals a_t.

2. *Autocorrelation check.* For an adequate model, both the sample ACF and PACF should not show any patterns. The Portmanteau test Q_1 which approximately follows a χ^2 distribution with $K - p - q$ degree of freedom can be used to check this assumptions.

$$Q_1 = m(m+2)\sum_{j=1}^{K}(m-j)^{-1}\hat{\rho}_{\hat{a}}^2(j) \tag{2.44}$$

2.5.7 The General Modeling Strategies for Transfer Function Model

The procedure for modelling strategies of the transfer function model according to (Wei, 2006; Bisgaard and Kulahci, 2011; and Yafee and McGee, 1999) are as the following steps:

1. *Plot the data.* By graphing or plotting the input and output series, we will gain a preliminary overview of the general behavior of these variables. This step is also called preprocessing step. In this step, transformation of both input and output series into stationary is required. After the preprocessing, an ARMA model is applied for the input series which may still contain the cross-correlation between the input and output series. Analogously, the similar filter is then applied to output series.

2. *Prewhitening.* The prewhitening filter is established from the ARMA model. This filter turns the input series into white noise. Prewhitening is needed if there is autocorrelation within the input series. Model building can be possibly done without prewhitening if there is no autocorrelation within the input series.

3. *Identification of the impulse response function and Transfer Function.* After the input and output series are prewhitened, the impulse response function and the Transfer Function are identified by examining the cross-correlation function. ln practice, examination of the cross-correlation will suggest several plausible Transfer Functions. The shape of the cross-correlation between those two prewhitened

series reveals the pattern of (r, s, and b) parameters of the Transfer Function (Box and Jenkins, 1976).

4. *Fitting a model for the noise.* Generally, the noise of Transfer Function model is expected to be autocorrelated. The noise is modeled by using ARIMA (p,d,q)

5. *Estimation of the Transfer Function.* To estimate the full model equation, we combine the results of steps 3 and 4. This combination provide us with the model specifications and the initial estimates of the parameters in the Transfer Function-Noise model. The Transfer Function-noise model can be estimated by conditional least squares.

6. *Diagnostics checks.* Incorrectly specified the Transfer Function is indicated by still much information left in the residuals explained by the input series. Therefore, we should first check the cross-correlation between the residuals and the prewhitened input series, then we check whether there is any autocorrelation left in the residuals, that is, to see whether the noise is modeled properly. ACF, PACF and Box - Ljung Q test can be used to diagnose the residuals. Any violations observed in the diagnostic checks will suggest the reevaluation of the noise model and/or Transfer Function model.

2.6 Neural Network

Artificial Neural Networks (ANN) is a mathematical model which is used to solve a complex systems of non-linear elements and to compute models for information processing and pattern identification. The major advantage of neural networks is that they are data driven and do not require restrictive assumptions about the form of the underlying model (Zhang, 2004). The neural network involves one or more hidden layers, in which the input variables are squashed or transformed by a special function, i.e. Gaussian function. In this research we will use Radial Basis Function Neural Network to model the residual from the Transfer Function-noise model

2.6.1 Neural Network Architecture

The structures of neural network are mainly based on the number of neurons in each layer and the type of activation functions. The number of input and output neurons can be easily determined by the number of input and output variables.

2.6.1.1 Neurons

A bounded function $y = f(x_1, x_2, \ldots, x_R; w_1, w_2, \ldots, w_R)$ where the x_i are the variables and the w_j are the parameters (or weights) is called as a neuron (Dreyfus, 2005). The variables of the neuron are often called inputs of the neuron and its value is its output. A neuron has a few entries provided by the outputs of other neurons. A group of connected neurons makes a layer. The entries are summed up and the state of the neuron is determined by the value of the resulting signal with respect to a certain threshold: if the signal is larger than the threshold the neuron is active; otherwise it is silent (Peretto, 1992).

Each neuron sends impulses to many other neurons (divergence), and receives impulses from many neurons (convergence). This simple idea appears to be the foundation for all activity in the central nervous system, and forms the basis for most neural network models (Freeman and Skapura, 1991). Let $x_1, x_2, \ldots, x_R$ are the individual element inputs and $w_{i,1}, w_{i,2}, \ldots, w_{i,R}$ are weights, where $i = 1, 2, \ldots,$ then the individual element inputs are multiplied by weights and the weighted values are fed to the summing junction. Their sum is simply $W * x$, the dot product of the (single row) matrix $\mathbf{W}$ and the vector $\mathbf{x}$ (Demuth and Beale, 2002).

The neuron has a bias b, and the sum of the weighted inputs and the bias forms the net input n, proceeds into an activation function f, and produces the scalar neuron output (Hagan, Demuth, Beale, and Jesus, 1996). The inputs to a neuron include its bias and the sum of its weighted inputs (using the inner product). The output of a neuron depends on the neurons inputs and on its activation function. This expression can be written as

$$a = f(W * x + b) \tag{2.45}$$

By considering the biological neuron, the weight w corresponds to synapse strength, the summation represents the cell body and the activation function, a basic multiple-input artificial neuron model is shown by Hanrahan (2011) in Figure 2.3.

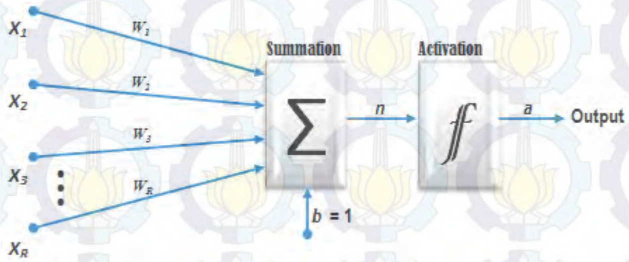


Figure 2.3: A Basic Multiple Input Artificial Neuron Model

2.6.1.2 Perceptron

The perceptron is a mathematical model of a biological neuron. A neural network which is made of R input units and one output unit is called a perceptron (Peretto, 1992). Multiplying each input value by a value called the weight is also modeled in the perceptron. The perceptron learning model can be represented as a processing node that receives a number of inputs, forms their weighted sum, and gives an output that is a function of this sum (Dunne, 2007). Perceptrons are useful as classifiers. They can classify linearly separable input vectors very well.

2.6.1.3 Layer

Artificial Neural Network (ANN) may consist of input, hidden and output layers. The net input renders to the activation function by calculating the output signal of the ANNs, which is called the layered network (Ali, 2014). The layer includes the weight matrix, the summers, the bias vector, the activation function boxes and the output vector. Some authors said that the inputs vector is another layer. A layer whose output is the network output is called an output layer, but the other layers are called hidden layers. Determining the number of hidden nodes within the hidden layer is best done by trial and error (Kitchens, Johnson and Gupta, 2002). A good starting point is the average of the number of input and output nodes, then greater and fewer numbers of hidden nodes are tested until an optimal architecture is obtained.

2.6.2 Radial Basis Function (RBF) Neural Network

Radial functions are simply a class of functions. In principles they could be employed in any sort of model linear or nonlinear and any sort of network single layer or multilayer. The Radial Basis Function Neural Network has local generalization abilities and fast convergence speed (Zhu *et al.*, 2014). Radial basis function neural network consists of input, hidden layer and output. The input contains units of signal source and the output which reacts to input model. The number of units on the hidden layer is determined by K-mean cluster. The movement from input to hidden layer is nonlinear and that from hidden layer to output is linear. Activation function of the units in hidden layer is Gaussian function.

2.6.2.1 *K*-Mean Cluster

RBF centers may be obtained with a clustering procedure such as the K-means algorithm. The real positive numbers (spread) parameters can be computed from the sample covariance of of each cluster. The K-means clustering algorithm starts by picking the number K of centers and randomly assigning the data points to K subsets. It then uses a simple re-estimation procedure to end up with a partition of the data points into K disjoint subsets or clusters. These clusters can be used as nodes in Radial Basis Function neural network. The following are K-means algorithm (Johnson and Winchern, 2007)

1. Initialize the clusters center
2. Finding the nearest mean to each data point, reassigning the data points to the associated clusters and then recomputing the new cluster means
3. Repeat step 2 until the old value of old cluster center equal to the value of new cluster center.

2.6.2.2 Radial Functions

Radial functions are a special class of function. Their characteristic feature is that their response decreases or increases monotonically with distance from a central point. In this study, the radial basis function applied was the Gaussian function (2.46) and the neural network output was then evaluated as the weighted linear summation of the radial basis functions (2.47) (Kagoda *et al.*, 2010)

$$\Omega_{j,t}(\mu_j, \sigma_j) = e^{-\left(\frac{\left\|x_{i,t}-\mu_j\right\|^2}{2\sigma_j^2}\right)}, \quad t = 1, 2, 3, \ldots, T. \tag{2.46}$$

where $x_{i,t}$ is the m dimensional input vector at time-step t, μ_j is the mean (center) of the jth node and, σ_j is the standard deviation (spread) of the jth node, T is the length of each input series.

$$a_t = \omega_b + \sum_{j=1}^{n} \omega_j.\Omega_{j,t}(\mu_j, \sigma_j), \quad t = 1, 2, 3, \ldots, T. \tag{2.47}$$

where n is the number of hidden nodes, y_t is the ith neural network output, ω_j is the weight vector of the connection between the jth node and the output node, ω_b is the bias term.

There are two operating modes named training and testing in the RBF neural network. Training the RBF network involves determining the number of RBF units, the width of RBF units and the output weight values. The criterion is to minimize the sum of squared errors (SSE). The architecture of Radial Basis Function Neural Networks is shown in Figure 2.4.

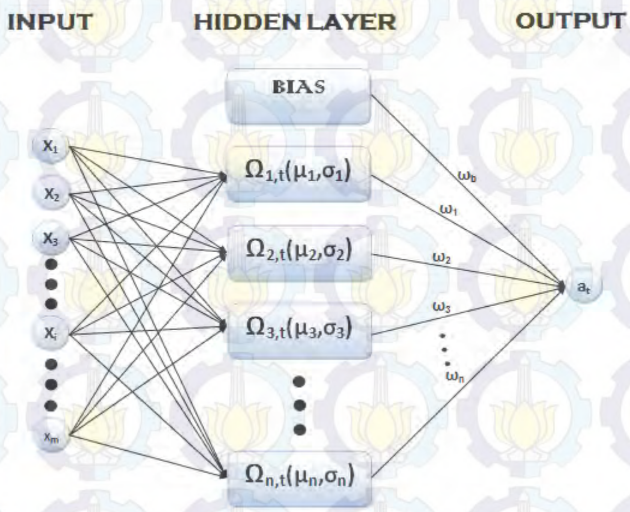


Figure 2.4: Radial Basis Function Neural Networks Architecture

The algorithm for Radial Basis Function Neural Network can be derived as follow:

1. Determine the number of clusters by using K-means cluster
2. Compute the values of μ_j and σ_j for each input variables within each cluster
3. Compute the values of $\Omega_{j,t}(\mu_j, \sigma_j)$ in hidden layer unit, e.g. $\Omega_{1,1}(\mu_1, \sigma_1)$ is the value for the first input in the first node
4. Estimate the weights as the following way:

   Consider the network output

$$a(x_i) \approx \omega_1 \Omega_{1,t}(\mu_1, \sigma_1) + \cdots + \omega_n \Omega_{n,t}(\mu_n, \sigma_n),$$

   we want to find $d_i = a(x_i)$ for each observation, so that

$$d_i \approx \omega_1 \Omega_{1,t}(\mu_1, \sigma_1) + \cdots + \omega_n \Omega_{n,t}(\mu_n, \sigma_n),$$

   in the matrix form, it can be written as

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_t \end{pmatrix} = \begin{pmatrix} \Omega_{1,1}(\mu_1, \sigma_1), & \ldots, & \Omega_{n,1}(\mu_n, \sigma_n) \\ \Omega_{1,2}(\mu_2, \sigma_2), & \ldots, & \Omega_{n,2}(\mu_n, \sigma_n) \\ \vdots & & \vdots \\ \Omega_{1,t}(\mu_1, \sigma_1), & \ldots, & \Omega_{n,t}(\mu_n, \sigma_n) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{pmatrix}$$

   or

$$d = \Phi \omega,$$

   where

$$d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_t \end{pmatrix}, \quad \Phi = \begin{pmatrix} \Omega_{1,1}(\mu_1, \sigma_1), & \ldots, & \Omega_{n,1}(\mu_n, \sigma_n) \\ \Omega_{1,2}(\mu_2, \sigma_2), & \ldots, & \Omega_{n,2}(\mu_n, \sigma_n) \\ \vdots & & \vdots \\ \Omega_{1,t}(\mu_1, \sigma_1), & \ldots, & \Omega_{n,t}(\mu_n, \sigma_n) \end{pmatrix}, \quad \omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{pmatrix}.$$

Finally, the weights can be computed by using the following formula

$$\begin{aligned}
d &= \Phi\omega \\
\Phi' d &= \Phi'\Phi\omega \\
(\Phi'\Phi)^{-1}\Phi' d &= (\Phi'\Phi)^{-1}(\Phi'\Phi)\omega \\
(\Phi'\Phi)^{-1}\Phi' d &= I\omega \\
(\Phi'\Phi)^{-1}\Phi' \left[d_1, d_2, \ldots, d_t\right]^T &= \left[\omega_1, \omega_2, \ldots, \omega_n\right]^T
\end{aligned}$$

For example, we will compute RBF neural network by using the residual data from Transfer Function-noise of outflow Lhokseumawe. Suppose our RBF neural network architecture has two inputs a_{t-1} and a_{t-2}, one output a_t, and 2 hidden nodes. These two hidden nodes are computed by using K-means cluster, the means and standard deviations for every inputs a_{t-1} and a_{t-2} within each node (cluster). Thus by minimizing sum square error, we obtain the weights $\omega_1 = -30.98$, $\omega_2 = 82.75$, and the bias $\omega_b = -35.16$. Suppose we only compute the forecast for in-sample in this example, then we formulate the RBF neural network model as follow

$$\begin{aligned}
a_t &= \omega_1\Omega_{1,t}(\mu_1, \sigma_1) + \omega_2\Omega_{2,t}(\mu_2, \sigma_2) + \omega_b, \\
&= -30.98\Omega_{1,t}(\mu_1, \sigma_1) + 82.75\Omega_{2,t}(\mu_2, \sigma_2) - 35.16, \\
&= -30.98 \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1} - \mu_{1_1}}{\sigma_{1_1}}\right)^2 + \left(\frac{a_{t-2} - \mu_{1_2}}{\sigma_{1_2}}\right)^2\right\}\right] + \\
&\quad + 82.75 \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1} - \mu_{2_1}}{\sigma_{2_1}}\right)^2 + \left(\frac{a_{t-2} - \mu_{2_2}}{\sigma_{2_2}}\right)^2\right\}\right] - 35.16,
\end{aligned} \tag{2.48}$$

for each nodes, we compute the values of $\mu_{1_1} = 227.36$, $\mu_{1_2} = -115.26$, $\sigma_{1_1} = 84$ and $\sigma_{1_2} = 71$ from the input series a_{t-1}, and the values of $\mu_{2_1} = 17.48$, $\mu_{2_2} = 95.29$, $\sigma_{2_1} = 63$, and $\sigma_{2_2} = 89$ from the input series a_{t-2}, then (2.47) can be written as

$$\begin{aligned}
a_t = -30.98 \exp&\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1} - 227.36}{84}\right)^2 + \left(\frac{a_{t-2} + 115.26}{71}\right)^2\right\}\right] + \\
+ 82.75 \exp&\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1} - 17.48}{63}\right)^2 + \left(\frac{a_{t-2} - 95.29}{89}\right)^2\right\}\right] - 35.16,
\end{aligned} \tag{2.49}$$

the one step ahead forecast can be written as

$$a_t(1) = -30.98 \exp\left[-\frac{1}{2}\left\{\left(\frac{a_t - 227.36}{84}\right)^2 + \left(\frac{a_{t-1} + 115.26}{71}\right)^2\right\}\right] + \\ + 82.75 \exp\left[-\frac{1}{2}\left\{\left(\frac{a_t - 17.48}{63}\right)^2 + \left(\frac{a_{t-1} - 95.29}{89}\right)^2\right\}\right] - 35.16 \tag{2.50}$$

and if the last two observations we seen from the actual series are $a_t = 4.87$ and $a_{t-1} = 14.51$, the one step ahead forecast output from (2.49) is

$$a_t(1) = -30.98 \exp\left[-\frac{1}{2}\left\{\left(\frac{4.87 - 227.36}{84}\right)^2 + \left(\frac{14.51 + 115.26}{71}\right)^2\right\}\right] + \\ + 82.75 \exp\left[-\frac{1}{2}\left\{\left(\frac{4.87 - 17.48}{63}\right)^2 + \left(\frac{14.51 - 95.29}{89}\right)^2\right\}\right] - 35.16 \\ = 18.23 \tag{2.51}$$

2.7 Hybrid Transfer Function - Neural Network Model

Transfer Function and Radial Basis Function Neural Network models have the advantages in their own linear or non linear patterns. However, not all time series types are universally suitable for each of them separately. The reason is, the phenomena of both linear and non linear correlation structures are often occurred among the observations in real world time series. Hence, it is not appropriate to apply Transfer Function and Neural Network models separately to any type of time series data. Therefore a number of efforts are developed to improve time series forecasting methods.

One approach to overcome the limitation of a pure linear or non linear time series modeling is the Hybrid model. Combining Transfer Function and Neural Network models in one Hybrid model will be more useful for improving forecasting accuracy. In this study, the Hybrid methodology follows the similar concept of linear ARIMA and non linear ANN models approach for time series forecasting developed by (Zhang, 2003 and Zhang, 2004). According to Zhang's hybrid model, time series data is based on the assumption that a sum of two components, linear and non linear. These two components can be expressed in the following

$$y_t = L_t + N_t, \tag{2.52}$$

where, y_t is the observation at time t and L_t, N_t denote linear and non linear components, respectively. Firstly, Transfer Function is fitted to the linear component and the corresponding forecast $\widehat{L}_t$ at time t is obtained. Then the residual at time t is given by $e_t = y_t - \widehat{L}_t$.

Adopting for Zhang's hybrid model, the residuals e_t after fitting Transfer Function contains only non linear component and can be modeled through Radial Basis Function Neural Network. Using p input nodes, the ANN model for residuals is

$$e_t = f(e_{t-1}, e_{t-2}, \ldots, e_{t-p}) + \varepsilon_t, \tag{2.53}$$

where f is a non linear function determined by the neural network and ε_t is the white noise. If $\widehat{N}_t$ is the forecast of neural network, the hybrid forecast at time t will be

$$\widehat{y}_t = \widehat{L}_t + \widehat{N}_t. \tag{2.54}$$

Through the previous empirical analysis, the forecast combinations were able to improve the overall modeling and forecasting performance. Such a model is illustrated in Figure 2.4.

2.8 Forecasting

Forecasting using Transfer Function models similar to forecasting using ARIMA models, with the exception of the additional input variable in the model. The Hybrid model is obtained through two steps: First, we build Transfer Function-noise model to estimate the linear component. Second, we apply Radial Basis Function Neural Network to the residuals of linear part to estimate the non linear component. Finally, we combine these two parts together to make a forecast.

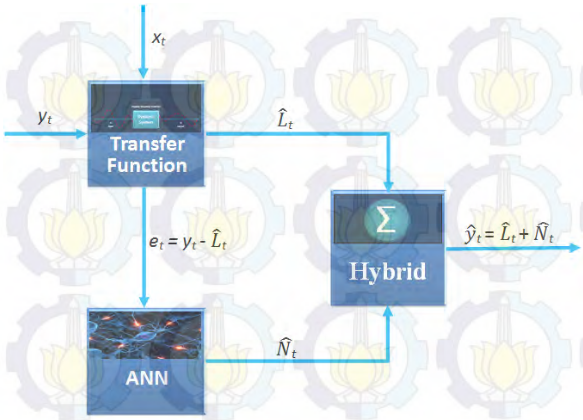


Figure 2.5: Hybrid Model

To show how to make forecasts of linear part for Transfer Function model, we refer to Box and Jenkins (1976), Wei (2006), and Montgomery *et al.*, (2008). Suppose, we use the notation for Transfer Function-noise model as in (2.36)

$$y_t = \frac{\omega(L)}{\delta(L)} L^b x_t + \frac{\theta(L)}{\varphi(L)} a_t \tag{2.55}$$

and

$$\varphi_x(L) x_t = \theta_x(L) \alpha_t. \tag{2.56}$$

Let α_t is independent zero mean white noise series, with variance σ^2_α, we have

$$u(L) = \frac{\varphi_x(L) L^b \theta_x(L)}{\delta(L) \varphi_x(L)} = u_0 + u_1 L + u_2 L^2 + \cdots \tag{2.57}$$

and

$$\varphi(L) = \frac{\theta(L)}{\phi(L)} = 1 + \varphi_1 L + \varphi_2 L^2 + \cdots \tag{2.58}$$

we can write (2.54) as

$$Y_t = u(L)\alpha_t + \varphi(L) a_t = \sum_{j=0}^{\infty} u_j \alpha_{t-j} + \sum_{j=0}^{\infty} \varphi_j a_{t-j}, \tag{2.59}$$

where $\varphi_0 = 1$. Thus, $Y_{t+l} = \sum_{j=0}^{\infty} u_j \alpha_{t+l-j} + \sum_{j=0}^{\infty} \varphi_j a_{t+l-j}$. Let $\widehat{Y}_t(l)$ of Y_{t+l} made at origin t be the l-step ahead optimal forecast, which is written in the form

$$\widehat{Y}_t(l) = \sum_{j=0}^{\infty} u^*_{l+j}\alpha_{t-j} + \sum_{j=0}^{\infty} \varphi^*_{l+j} a_{t-j} \tag{2.60}$$

Then the forecast error is

$$\begin{aligned} Y_{t+l} - \widehat{Y}_t(l) &= \sum_{j=0}^{l-1}[u_j\alpha_{t+l-j} + \varphi_j a_{t+l-j}] \\ &\quad - \sum_{j=0}^{\infty}[u^*_{l+j} - u_{l+j}]\alpha_{t-j} - \sum_{j=0}^{\infty}(\varphi^*_{l+j} - \varphi_{l+j})a_{t-j} \end{aligned} \tag{2.61}$$

The mean square forecast error $E[Y_{t+l} - \widehat{Y}_t(l)]^2$ is given by

$$\begin{aligned} E[Y_{t+l} - \widehat{Y}_t(l)]^2 &= \sum_{j=0}^{l-1}(\sigma^2_\alpha u^2_j + \sigma^2_\alpha \varphi^2_j) \\ &\quad + \sum_{j=0}^{\infty}\sigma^2_\alpha(u^*_{l+j} - u_{l+j})^2 + \sum_{j=0}^{\infty}\sigma^2_\alpha(\varphi^*_{l+j} - \varphi_{l+j})^2 \end{aligned} \tag{2.62}$$

which is minimized if $u^*_{l+j} - u_{l+j}$ and $\varphi^*_{l+j} - \varphi_{l+j}$. Therefore the mean square error forecast $\widehat{Y}_t(l)$ of Y_{t+l} at origin t is given by the conditional expectation of Y_{t+l} at time t. Since $E[Y_{t+l} - \widehat{Y}_t(l)] = 0$ then the forecast is unbiased. The variance of the lead-l forecast is given by

$$V(l) = E[Y_{t+l} - \widehat{Y}_t(l)]^2 = \sigma^2_\alpha \sum_{j=0}^{l-1} u^2_j + \sigma^2_\alpha \sum_{j=0}^{l-1} \varphi^2_j \tag{2.63}$$

To make the forecast of non linear part, the residuals from the Transfer Function-noise model is modeled by Radial Basis Function Neural Network. Suppose this prediction is obtained as follow

$$\widehat{N}_t = f(e_{t-1}, e_{t-2}, \ldots, e_{t-p}) + \varepsilon_t \tag{2.64}$$

Then the final predictions based on Zhang's hybrid model are obtained by summing the Transfer Function predictions in (2.59) and Radial Basis Function Neural predictions in (2.63) which is stated as Hybrid Prediction (HP) in (2.64). This model is suitable for both one step ahead and multi step ahead predictions

$$HP = \sum_{j=0}^{\infty} u_{l+j}^{*} \alpha_{t-j} + \sum_{j=0}^{\infty} \varphi_{l+j}^{*} a_{t-j} + f(e_{t_1}, e_{t_2}, \ldots, e_{t_n}) + \varepsilon_t \tag{2.65}$$

2.9 Performance Evaluation

There are a number of error statistics are used in evaluating forecasting performance. Here in this research, we only use Root Mean Squared Error (RMSE) and Mean Absolute Deviation (MAD). These methods provide enough information to evaluate the forecasting performance.

1. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T} e(t)^2} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (y_t - \widehat{y}_t)^2} \tag{2.66}$$

where the error is calculated as the difference between the target output y_t and the network output $\widehat{y}_t$.

2. Mean Absolute Deviation (MAD)

$$MAD = \frac{1}{T}\sum_{t=1}^{T} |y_t - \widehat{y}_t| \tag{2.67}$$

2.10 Testing Nonlinearity

Linear models have been the focus of theoretical and applied econometrics. Under the stimulus of the economic theory, the relationships between variables of nonlinear models are frequently suggested. Consequently, the interest in testing whether or not a single economic series or group of series may be generated by a nonlinear model against the alternative that they were linearly related instead.

The tests for nonlinearity are influenced by specific hypothesis under which they have been conceived (Bisaglia and Gerolimetto, 2014). A wide variety of nonlinearity tests have been developed to ascertain the incidence of nonlinearity in economic data.

But here in this study we only use White and Terasvirta tests to detect whether or not a time series data is generated by nonlinear process.

Theorem: Let the Reduced Model is defined as

$$Y_t = f\left(X_t, \widehat{w}_n^{(R)} + \varepsilon_t^R\right), \tag{2.68}$$

where I_R is the number of parameters to be estimated. And, let the Full Model that is more complex than the Reduced Model is defined as

$$Y_t = f\left(X_t, \widehat{w}_n^{(F)} + \varepsilon_t^F\right), \tag{2.69}$$

where I_F is the number of parameters in the Full Model, $I_F > I_R$. Then, under the hypothesis testing for an additional parameters in the Full Model equal to zero, the F statistic can be constructed, i.e.

$$\frac{SSE_{(R)} - SSE_{(F)}/(I_F - I_R)}{SSE_{(F)}/(n - I_F)} \sim F_{(v_1=[I_F-I_R],v_2=[n-I_F])}. \tag{2.70}$$

F test in (2.66) can be written as

$$F = \frac{SSE_{(R)} - SSE_{(F)}/(df_{(R)} - df_{(F)})}{SSE_{(F)}/df_{(F)}}, \tag{2.71}$$

or

$$F = \frac{R^2_{incremental}/(df_{(R)} - df_{(R)})}{(1 - R^2_{(F)})/df_{(F)}}, \tag{2.72}$$

where $R^2_{incremental} = R^2_{(F)} - R^2_{(R)}$, $df_{(R)} = n - I_R$ is degree of freedom at Reduced Model, and $df_{(F)} = n - I_F$ is degree of freedom at Full Model.

Generally, Suhartono (2008) define the hypothesis tests which is used in nonliearity test as follow:

$$H_0 : f(x) \text{ is linear function of } x$$
$$H_1 : f(x) \text{ is nonlinear function of } x$$

By using F test in (2.71), if the p-value less than α=5%, then the appropriate model to describe the relationship between x and *f(x)* is nonlinear model.

2.11 Calendar Variation Model

One of the effectiveness model for estimating trading day and moving holiday effects in economic time series is calender variation model. Monthly time series data are frequently subject to calendar variation. In most Islamic countries for example, monthly time series data subject to Ramadhan and Idul Fitri Day effects. A number of calendar variation effect on time series data have been studied by many researchers. Liu (1980) for instance, he studied the effect of holiday variation on the identification and estimation of ARIMA models. Cleveland and Devlin (1980) proposed two sets of diagnostic methods for detecting calendar effects in monthly time series, i.e. by using spectrum analysis and time domain graphical displays. Suhartono, Lee, and Hamzah (2010) developed a calendar variation model based on time series regression method for forecasting time series data with Ramadhan effect. The results show that modeling some real data with calendar variation effect provided better forecast.

Data with calendar variation can also be modeled by using regression (Suhartono, Lee, and Hamzah 2010). Linear regression model for data with calendar variation can be expressed as:

$$y_t = \beta_0 + \beta_1 V_{1,t} + \beta_2 V_{2,t} + \cdots + \beta_p V_{p,t} + w_t \tag{2.73}$$

where $V_{p,t}$ is dummy variable for p-th calendar variation effect, and w_t is the error component, usually assumed independently and identically distributed as normal with mean 0 and variance σ_w^2. This method can be applied to the identification of intervention and transfer function models which may also be subject to calendar variation.

The procedure of finding the Hybrid model of Transfer Function subjected to calender variation with outlier detection and RBF neural network is similar to the procedure of finding Hybrid Transfer Function-noise and RBF neural network. The firs step is to compute the forecast of linear part, then the residuals from the linear part are computed by RBF neural network to obtain the forecasts.

CHAPTER 3
METHODOLOGY

In this chapter, we discuss data gathering method and data variables. Indonesian central bank (BI) of Aceh's representative offices are located in Banda Aceh and Lhokseumawe. The map of these locations are shown in Figure 3.1.

Figure 3.1: Representative Offices of Indonesian Central Bank (BI) in Aceh Province

3.1 Data and Variables

The data set that we use in this analysis is based on monthly data of cash inflow and outflow of Indonesian central bank (BI) at representative offices in Banda Aceh and Lhokseumawe, province of Aceh, Indonesia. We analyzed this monthly data for the periods of time from January 2003 to December 2014. The structure of data consist of inflow, outflow and Consumer Price Index (CPI) with 144 observations and the starting date is at 01.01.2003. Figures 3.2, 3.3, 3.4 show data Inflow, outflow, and Consumer Price Index (CPI) respectively.

For evaluating the performance of our analysis, we divided these data into in-sample data as a training data and out-sample data as a testing data. The training data (92%

of total observation) is started from January 2003 to December 2013 (equal to 132 observations). While the testing data (8% of total observation) is started from January 2014 to December 2014 (equal to 12 observations).

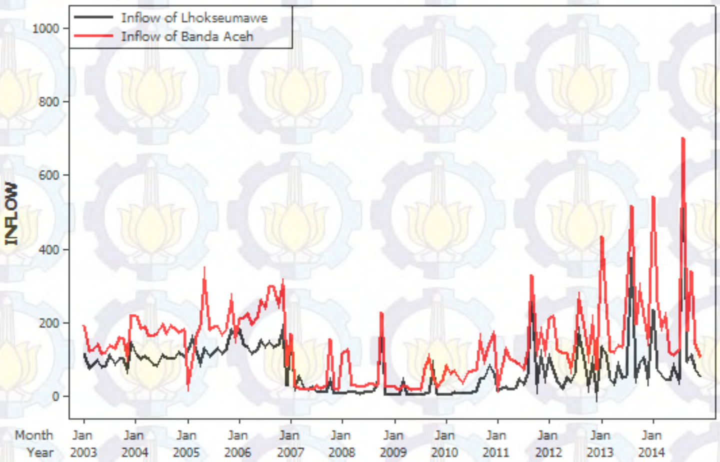


Figure 3.2: Inflow (in Billion IDR) - Lhokseumawe and Banda Aceh

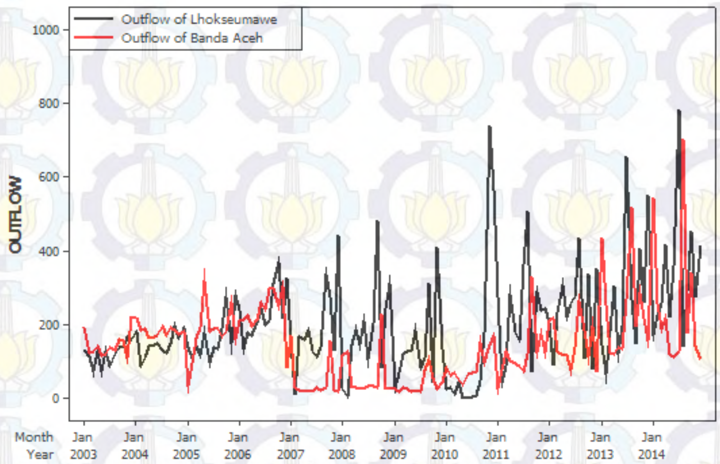


Figure 3.3: Outflow (in Billion IDR) - Lhokseumawe and Banda Aceh

We use two variables in this analysis: the output variable and the input variable. In this study, we analyze the empirical data for Lhokseumawe and Banda Aceh separately. Table 3.1 shows a brief description of the variables that we use in this study.

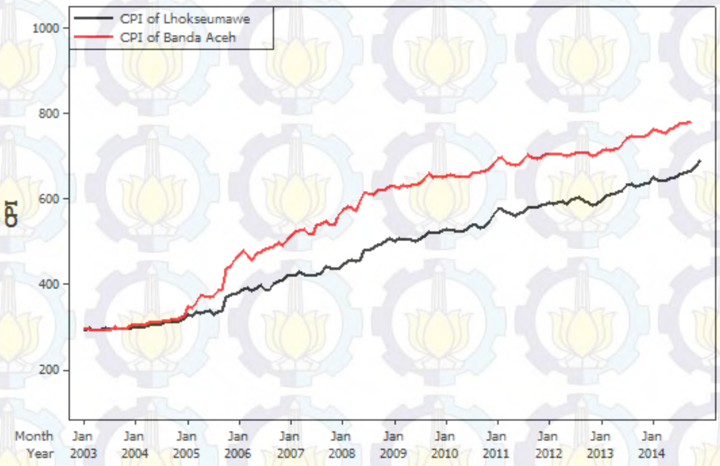


Figure 3.4: Consumer Price Index (CPI) - Lhokseumawe and Banda Aceh

3.2 Steps for Data Analysis

The following are steps for building Hybrid Transfer Function-noise and Neural Network models for Lhokseumawe and Banda Aceh time series data:

1. Model Identification

   (a) Plot the inflow and outflow data for Lhokseumawe and Banda Aceh representative offices of Indonesian Central Bank (BI) . This time series plots will give us a preliminary overview of the general behavior of these variables.

   (b) Transform both input and output series (for inflow, outflow, and Consumer Price Index) into stationary by applying differencing

   (c) Prewhiten the input and output series from the exising ARIMA model to obtain the white noise series.

   (d) Examine cross correlation function and autocorrelation for each the prewhitened input and output series

   (e) Identify the orde b, s, r of Transfer Function model

   (f) Preliminary estimate Transfer Function model based on determined order.

2. Parameter Estimation

   Estimate the parameter of single input Transfer Function by using conditional least square.

Table 3.1: Summary of Name, Notation, and Definition of Variables

Variable	Notation	Description
Outflow of Lhokseumawe	$Y_{1,t}$	The money (in IDR) coming out from Indonesian Central Bank (BI) of Lhokseumawe representative office to other parties through the withdrawal process by commercial banks and public transaction activities
Inflow of Lhokseumawe	$Y_{2,t}$	The transfer of money (in IDR) from other parties to Indonesian Central Bank (BI) of Lhokseumawe representative office in a given period of time through the deposit of commercial banks and the public transaction activities
Outflow of Banda Aceh	$Y_{3,t}$	The money (in IDR) coming out from Indonesian Central Bank (BI) of Banda Aceh representative office to other parties through the withdrawal process by commercial banks and public transaction activities
Inflow of Banda Aceh	$Y_{4,t}$	The transfer of money (in IDR) from other parties to Indonesian Central Bank (BI) of Banda Aceh representative office in a given period of time through the deposit of commercial banks and the public transaction activities
Consumer Price Index Lhokseumawe	$X_{1,t}$	A Consumer Price Index of Lhokseumawe measures change in the price level of a market basket of consumer goods and services purchased by households
Consumer Price Index Banda Aceh	$X_{2,t}$	A Consumer Price Index of Banda Aceh measures change in the price level of a market basket of consumer goods and services purchased by households

3. Diagnostic Checking

   (a) Check whether the residual are white noise, independent of the input series and hence the independent of the prewhitened input series, and normally distributed with zero mean and variance σ^2. If this assumptions hold, then the model is adequate. The residuals can be diagnosed by their ACF and PACF along with use of the Box - Ljung Q test. However, if the residuals are not independent, it can be modeled by the ARIMA (p,d,q).

(b) After obtaining the predictions from Transfer Function-noise model, the residual series is used to model the Radial Basis Function Neural Network.

4. Forecasting

   (a) Compute the residual from Transfer Function-noise model $e_t = y_t - \widehat{L}_t$

   (b) Forecast the pure linear part of Transfer Function model

   (c) Forecast the non linear part of Neural Network model

   (d) Compute the Hybrid forecast by combining the both linear and non linear forecasts.

This page is intentionally left blank

CHAPTER 4
RESULTS AND ANALYSIS

Six data sets are used in this research. These six sets of time series data are shown in Figure 3.2, Figure 3.3, and Figure 3.4. We see that the series Inflow, Outflow, and consumer price index (CPI) in Banda Aceh and Lhokseumawe are clearly non-stationary series, they are trending up over the given time period. Therefore, differencing is required to obtain stationarity. In this study we use a single input and single output models.

The general steps of Transfer Function-noise model building process is applied to obtained the appropriate linear model. After obtaining the predictions of Transfer Function-noise model, its residual series is used to form non linear model by using the Radial Basis Function (RBF) Neural Network. We estimated several different RBF Neural network models with different nodes in training data (in-sample) in order to obtain the optimum networks. It is aimed to better capture the underlying behavior of the time series movement of Inflow and Outflow for both Lhokseumawe and Banda Aceh representative offices.

We use single layer network with 2 input and 1 output for model building process of the entire series in this research. 5 node is the most optimum network in comparison to the other tested nodes of the residuals of Lhokseumawe Outflow and 10 node is for the residual of Lhokseumawe Inflow series. While the optimum network of residual series of Outflow Banda Aceh is 8 node. The network of residual Inflow Banda Aceh residual is optimized in 25 nodes. Nodes are computed by using K-means cluster. Means and standard deviations are computed for each input variables within each nodes. In hidden layer, bias term is employed and the Gaussian function is used as the activation function for each nodes.

To assess the forecasting performance of different models, each data set of both Transfer Function-noise model and Hybrid model is divided into two samples of in-sample and out-sample. We use in-sample data for model selection and estimation. The rest of out-sample data is not used in the model building process and hence it represents a set of true "unseen future" observations for testing the proposed model effectiveness as well as for model comparisons.

Prior to apply the Neural Network and Hybrid model, we first test the residual of Transfer Function-noise model by using the White and Terasvirta tests. Table 4.1 show that all data have the nonlinear relationship between residual and its first and second lags, except for residual of Inflow Banda Aceh, where both p-values of White test and Terasvirta test are larger than α=5%. In this study, we use several software packages

Table 4.1: Nonliearity Test

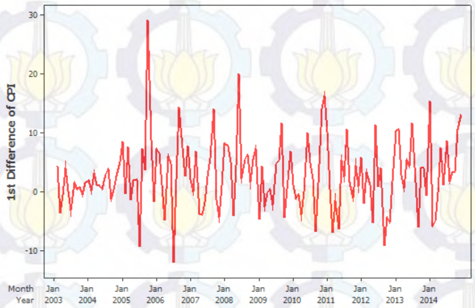
Residual	P-value of White Test	P-value of Terasvirta Test
Outflow Lhokseumawe	0.081	0.036
Inflow Lhokseumawe	0.002	0.002
Outflow Banda Aceh	9.004e-14	2.442e-15
Inflow Banda Aceh	0.305	0.385

to run the data. All Transfer Function-noise modeling is implemented via SAS system while neural network models are built using Matlab. Non linearity test is performed by R. All figures are plotted by Minitab. The Root Mean Squared Error (MSE) and Mean Absolute Deviation (MAD) are selected to be the forecasting accuracy measures.

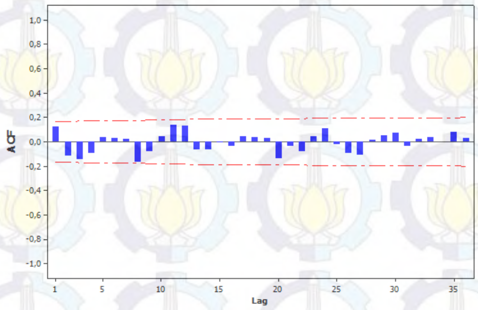
4.1 Lhokseumawe Representative Office

4.1.1 Outflow of Lhokseumawe

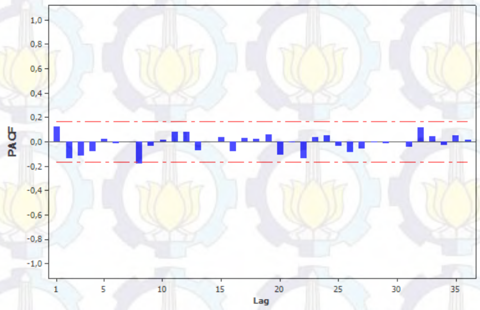
In the prewhitening step, we first fit an ARIMA model to the Consumer Price Index (CPI) of Lhokseumawe. Since the CPI of Lhokseumawe is not stationary, we require first difference in order to obtain stationarity. Figure 4.1(a) shows that the process is changing around a constant mean and has a constant variance, and hence we assume that it is stationary. Sample ACF and PACF plots in Figure 4.1(b) and Figure 4.1(c) suggest that the certain pattern of ARIMA(p,d,q) can be formulated. However, since the residual of CPI series is already white noise, then the subset ARIMA(0,1,0) model can be used as the prewhitening filter for the input series. Now with the same filter in the output series, we get the prewhitening for the output series. We then compute the sample cross correlation of the prewhitened input α_t and the prewhitened output β_t. The crosscorrelation function (CCF) between α_t and β_t is given in Appendix A.3. The



(a) 1^{st} Differencing



(b) ACF



(c) PACF

Figure 4.1: Time Series plots of Differenced CPI, ACF, and PACF of Lhokseumawe

CCF output indicate that there is a delay of 0 lag in the system, then we specify $b = 0$. The specifications of the orders s and r are based on the number of spikes in CCF after 0 lag. There are several spikes appear after lag 0, but only lag 1 are significant, then we specify the order of s is subset 1 and we set the denominator as zero, that is, r=0. Now we fit the noise which incorporate into the overall model. From the output in Appendix A.4 and Appendix A.5, the subset ARIMA([1,2,12],0,[3,23]) is the most parsimony model for noise term. The appropriate coefficients estimation are $\omega_0 = 5.55$, $\omega_1 = -4.47$, $\theta_3 = 0.55$, $\theta_{23} = -0.23$, $\varphi_1 = -0.70$, $\varphi_2 = -0.60$ and $\varphi_{12} = 0.47$. All these coefficients are less than $\alpha = 10\%$, then these coefficients are all significant (Appendix A.2). Thus the Transfer Function-noise model for outflow series of Lhokseumawe is given as follow:

$$
\begin{aligned}
y_t &= (\omega_0 - \omega_1 L^1)x_t + \frac{1-\theta_3 L^3 - \theta_{23}L^{23}}{1-\varphi_1 L - \varphi_2 L^2 - \varphi_{12}L^{12}} a_t \\
&= (5.55 - 4.47L)x_t + \frac{1-0.55L^3 + 0.23L^{23}}{1+0.70L+0.60L^2-0.47L^{12}} a_t
\end{aligned} \tag{4.1}
$$

Next step is to forecast the Transfer Function-noise model in (4.1). The residual a_t is analyzed by using Radial Basis Function Neural Network. The nonlinearity test in Table 4.1 indicate a non linear relationship between the residual of Outflow Lhokseumawe series and its first and second lags. The optimal estimated weights of this model are $\omega_1 = -30.98$, $\omega_2 = 82.75$, $\omega_3 = -339.45$, $\omega_4 = 38.23$, $\omega_5 = -12.43$, and the bias weight $\omega_b = 51.16$. Thus, RBF neural network model for $a(t)$ is as follow

$$
\begin{aligned}
a_t &= \sum_{j=1}^{5} \omega_j \Omega_{j,t}(\mu_j, \sigma_j) + \omega_b \\
&= -30.98\Omega_{1,t}(\mu_1,\sigma_1) + 82.75\Omega_{2,t}(\mu_2,\sigma_2) + \cdots + \\
&\quad - 12.43\Omega_{5,t}(\mu_5,\sigma_5) + 51.16,
\end{aligned} \tag{4.2}
$$

where

$$
\begin{aligned}
\Omega_{1,t}(\mu_1,\sigma_1) &= \exp\left[-\tfrac{1}{2}\left\{\left(\tfrac{a_{t-1}-\mu_{1_1}}{\sigma_{1_1}}\right)^2 + \left(\tfrac{a_{t-2}-\mu_{1_2}}{\sigma_{1_2}}\right)^2\right\}\right], \\
\Omega_{2,t}(\mu_2,\sigma_2) &= \exp\left[-\tfrac{1}{2}\left\{\left(\tfrac{a_{t-1}-\mu_{2_1}}{\sigma_{2_1}}\right)^2 + \left(\tfrac{a_{t-2}-\mu_{2_2}}{\sigma_{2_2}}\right)^2\right\}\right], \\
&\vdots \\
\Omega_{5,t}(\mu_5,\sigma_5) &= \exp\left[-\tfrac{1}{2}\left\{\left(\tfrac{a_{t-1}-\mu_{5_1}}{\sigma_{5_1}}\right)^2 + \left(\tfrac{a_{t-2}-\mu_{5_2}}{\sigma_{5_2}}\right)^2\right\}\right],
\end{aligned}
$$

and μ, σ for each node (cluster) are availiable in Appendix C.1.

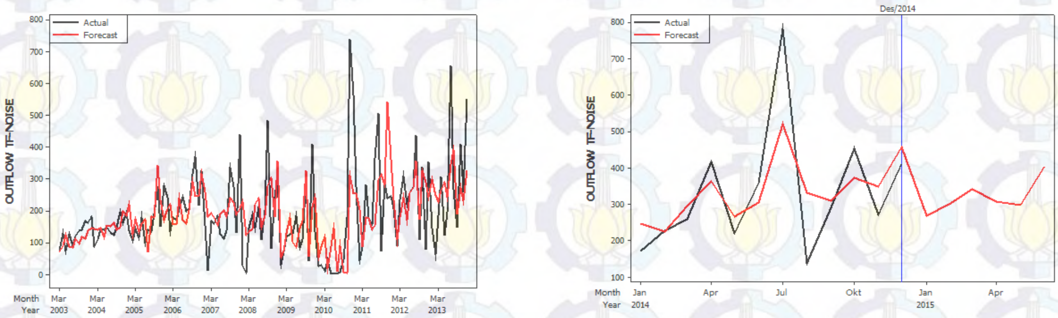
The last step is to perform the forecast combination. Two forecast horizons of in-sample and out-sample monthly periods are used. The results the hybrid model show that by combining forecast together the overall forecasting errors can be reduced in training data but can't in testing data. Table 4.2 shows that the percentage improvements of Root Mean Square Error (RMSE) of the hybrid model over the Transfer Function-noise model for in-sample data. RMSE decrease to 3.1% in in-sample and MAD goes down by 0.6% in in-sample as well. However, as time extend to 12 months ahead, the forecast errors are getting worse. For out-sample, 5% and 4.5% increase in

RMSE and MAD respectively

Table 4.2: Measurement of Performance Test for Outflow Series of Lhokseumawe

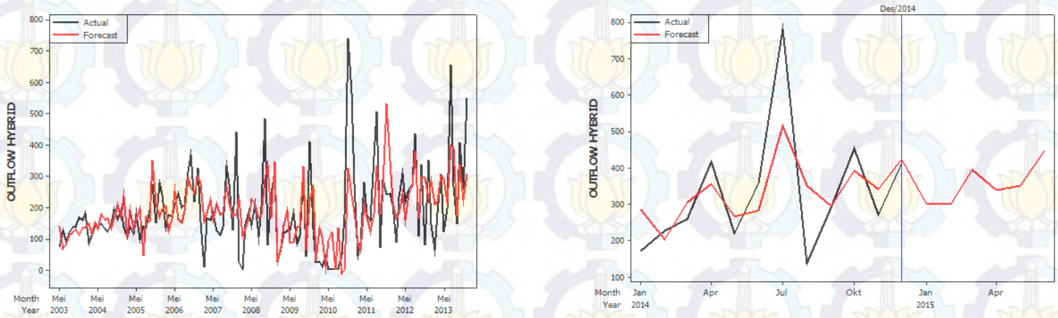
Model	In-Sample		Out-Sample	
	RMSE	MAD	RMSE	MAD
TF-ARIMA NOISE	106.19	73.42	106.76	79.14
HYBRID	102.91	72.97	112.53	82.70

Figures 4.2 and Figures 4.3 show the comparison between actual observations and the forecasts from the in-sample and out-sample horizons of Transfer Function-noise model and Hybrid model of Lhokseumawe Outflow series respectively.



(a) In-sample (b) Out-sample

Figure 4.2: Actual vs Transfer Function-Noise prediction of Outflow Lhokseumawe



(a) In-sample (b) Out-sample

Figure 4.3: Actual vs Hybrid prediction of Outflow Lhokseumawe

4.1.2 Inflow of Lhokseumawe

In a similar fashion to the Outflow series, we first fit an ARIMA model to the Consumer Price Index (CPI) of Lhokseumawe. The first differencing of this series is shown in Figure 4.1(a). The ACF and PACF plots in Figure 4.1(b) and Figure 4.1(c) indicate that the subset ARIMA([1,2,8],1,0) is appropriate to model Consumer Price Index (CPI) series. This model is used as a filter to prewhiten the input and the output series.

The CCF output (Appendix A.9) indicate that there is a delay of 1 lag in the system, then we specify $b = 1$. The specifications of the orders s and r are based on the number of spikes in CCF after 1 lag. There are several spikes appear after lag 1. In identification step, we found that there are two possible candidate Transfer Function-noise model for Inflow series.

The first model, we identify the orders of b=1, the subset s=[1,5,6,10,11] and r=0. The noise model is identified to be ARIMA(0,0,0). The appropriate coefficients estimation are $\omega_0 = 1.91$, $\omega_1 = 2.44$, $\omega_5 = 1.69$, $\omega_6 = -2.03$, $\omega_{10} = 1.81$, $\omega_{11} = -1.83$. All these parameters are significant. Thus the Transfer Function-noise model is given as follow

$$\begin{aligned} y_t &= (\omega_0 - \omega_1 L - \omega_5 L^5 - \omega_6 L^6 - \omega_{10} L^{10} - \omega_{11} L^{11}) L^1 x_t + a_t \\ &= (1.91 - 2.44L - 1.69L^5 + 2.03L^6 - 1.81L^{10} + 1.83L^{11}) x_{t-1} + a_t. \end{aligned} \tag{4.3}$$

The second model, the orders of b, s, and r are identified to be equal to zero. While the noise model is fit to follow the subset ARIMA(12,0,[1,22,23,24]). The appropriate coefficients estimation are $\omega_0 = 0.78$, $\theta_1 = 0.73$, $\theta_{22} = 0.29$, $\theta_{23} = -0.64$, $\theta_{24} = 0.31$, $\varphi_{12} = 0.48$. Thus the Transfer Function-noise model is given as follow

$$\begin{aligned} y_t &= \omega_0 x_t + \frac{1 - \theta_1 L - \theta_{22} L^{22} - \theta_{23} L^{23} - \theta_{24} L^{24}}{1 - \varphi_{12} L^{12}} a_t \\ &= 0.78 x_t + \frac{1 - 0.73L - 0.29L^{22} + 0.64L^{23} - 0.31L^{24}}{1 - 0.48L^{12}} a_t \end{aligned} \tag{4.4}$$

All parameters of model in (4.4) are significant. Although both model 1 and model 2 have significant parameters, in term of model selection, model 1 has smaller AIC = 133.24 and BIC = 1347.92 than AIC = 1351.68 and BIC = 1368.93 in model 2.

Base on this reason, we choose model 1 to proceed to Neural Network forecasting. The parameters estimation of model 1 is given in Appendix A.8. Since all coefficients are less than $\alpha = 10\%$, these parameters are significant.

Next step is to forecast the Transfer Function-noise of model 1 (4.3). The residual a_t is analyzed by using Radial Basis Function Neural Network. Table 4.1 indicate a non linear relationship between the residual of Inflow Lhokseumawe series and its first and second lags. The optimal estimated weights of Inflow series model are $\omega_1 = -32.56$, $\omega_2 = 91.42$, $\omega_3 = 102.12$, $\omega_4 = -377.96$, $\omega_5 = 139.86$, $\omega_6 = -133.99$, $\omega_7 = 50.37$, $\omega_8 = 5.43$, $\omega_9 = 7.35$, $\omega_{10} = -3.32$, and the bias weigh $\omega_b = 24.29$. Thus, RBF neural network model for $a(t)$ is as follow

$$\begin{aligned} a(t) &= \sum_{j=1}^{10} \omega_j \Omega_{j,t}(\mu_j, \sigma_j) + \omega_b \\ &= -32.56\Omega_{1,t}(\mu_1, \sigma_1) + 91.42\Omega_{2,t}(\mu_2, \sigma_2) + \cdots + \\ &\quad - 3.32\Omega_{10,t}(\mu_{10}, \sigma_{10}) + 24.29, \end{aligned} \tag{4.5}$$

where

$\Omega_{1,t}(\mu_1, \sigma_1) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{1_1}}{\sigma_{1_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{1_2}}{\sigma_{1_2}}\right)^2\right\}\right]$,

$\Omega_{2,t}(\mu_2, \sigma_2) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{2_1}}{\sigma_{2_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{2_2}}{\sigma_{2_2}}\right)^2\right\}\right]$,

$\vdots$

$\Omega_{10,t}(\mu_{10}, \sigma_{10}) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{10_1}}{\sigma_{10_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{10_2}}{\sigma_{10_2}}\right)^2\right\}\right]$,

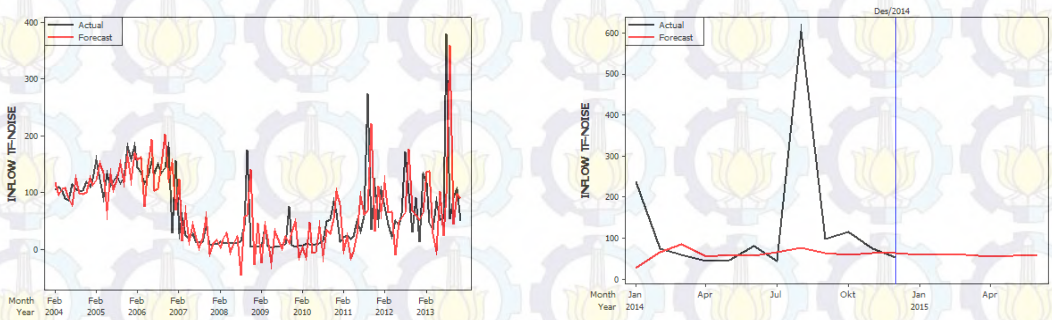
and μ, σ for each node (cluster) are availiable in Appendix C.2.

In the hybrid stage, the forecast of Transfer Function-noise model is combined with the Radial Basis Function model. Table 4.3 shows that the Root Mean Square Error (RMSE) and the Mean Absolute Deviation (MAD) of the hybrid model decrease 18.28% and 16.62% respectively over the Transfer Function-noise model for in-sample data. However if the time horizon extend to 12 months ahead, the MSE increase slightly 2% and MAD goes up sharply 24,51%.

Table 4.3: Measurement of Performance Test for Inflow Series of Lhokseumawe

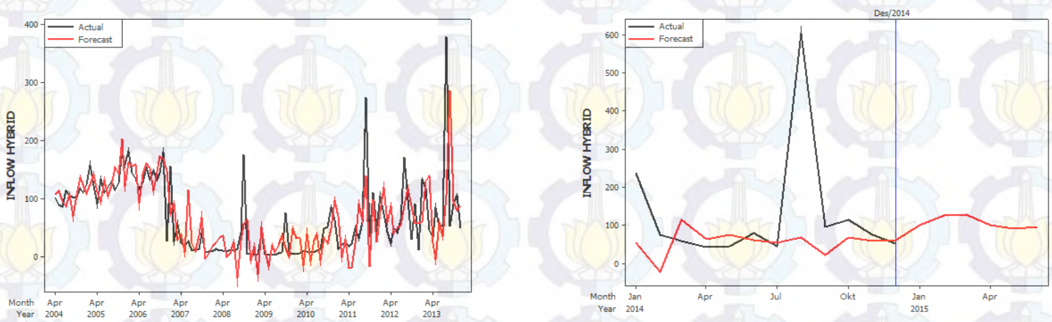
Model	In-Sample		Out-Sample	
	RMSE	MAD	RMSE	MAD
TF-ARIMA NOISE	61.81	39.17	165.47	73.47
HYBRID	50.51	32.66	168.94	91.48

Figures 4.4 and Figures 4.5 show the comparison between actual observations and the forecasts from the in-sample and out-sample horizons of Transfer Function-noise model 1 and Hybrid model 1 of Lhokseumawe Inflow series respectively.



(a) In-sample (b) Out-sample

Figure 4.4: Actual vs Transfer Function-Noise prediction of Inflow Lhokseumawe-Model 1



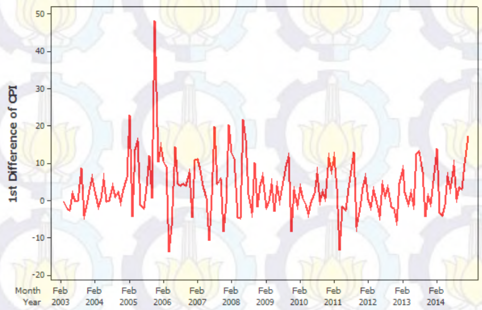
(a) In-sample (b) Out-sample

Figure 4.5: Actual vs Hybrid prediction of Inflow Lhokseumawe-Model 1

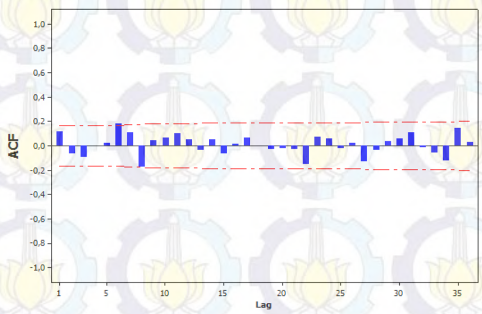
4.2 Banda Aceh Representative Office

4.2.1 Outflow of Banda Aceh

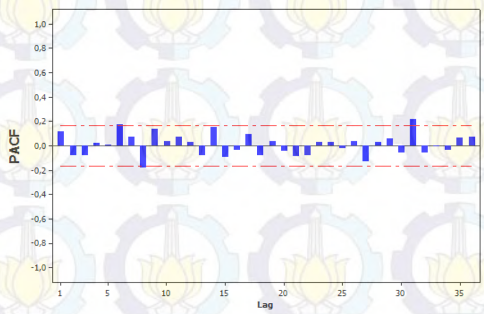
We first prewhiten the Consumer Price Index (CPI) series of Banda Aceh. The original series of Banda Aceh CPI series is not stationary, but its difference is assumed to be stationary (Figure 4.6.a). By examining the sample ACF and PACF in Figure 4.6(b) and Figure 4.6(c), the time series plot suggest that the subset ARIMA([6],1,0) is appropriate to prewhiten input series as well as the output series of Banda Aceh.



(a) 1^{st} Differencing



(b) ACF



(c) PACF

Figure 4.6: Time Series plots of Differenced CPI, ACF, and PACF of Banda Aceh

The sample CCF output (Appendix B.3) suggest the order of b=1, r=0, and s=0. The noise model from overall model (Appendix B.4 and Appendix B.5) is identified to follow the ARIMA(0,0,0). The significant estimated coefficients (all coefficients are less than $\alpha = 10\%$) of this model is $\omega_0 = -4.89$ (Appendix B.2). Thus the model is given as follow

$$\begin{aligned} y_t &= \omega_0 L^1 x_t + a_t \\ &= -4.89 X_{t-1} + a_t \end{aligned} \tag{4.6}$$

Next step is to forecast the Transfer Function-noise model in (4.6). The residual a_t is modeled by using Radial Basis Function Neural Network. This analysis is conducted due to non linear relationship between residual of Outflow Banda Aceh and its first and second lags as indicated in Table 4.1. The optimal estimated weights of Inflow series model are $\omega_1 = 70.59$, $\omega_2 = 0.0001$, $\omega_3 = 108.39$, $\omega_4 = -128.19$, $\omega_5 = -4.50$, $\omega_6 = 162.31$, $\omega_7 = 63.41$, $\omega_8 = 4.93$, and the bias weight $\omega_b = -4.29$. Thus, RBF neural network model for $a(t)$ is as follow

$$\begin{aligned} a(t) &= \sum_{j=1}^{10} \omega_j \Omega_{j,t}(\mu_j, \sigma_j) + \omega_b \\ &= 70.59\Omega_{1,t}(\mu_1, \sigma_1) + 0.0001\Omega_{2,t}(\mu_2, \sigma_2) + \cdots + \\ &\quad + 4.93\Omega_{8,t}(\mu_8, \sigma_8) - 4.29, \end{aligned} \tag{4.7}$$

where

$$\Omega_{1,t}(\mu_1, \sigma_1) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{1_1}}{\sigma_{1_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{1_2}}{\sigma_{1_2}}\right)^2\right\}\right],$$

$$\Omega_{2,t}(\mu_2, \sigma_2) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{2_1}}{\sigma_{2_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{2_2}}{\sigma_{2_2}}\right)^2\right\}\right],$$

$$\vdots$$

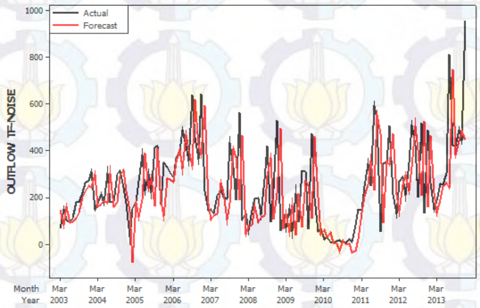
$$\Omega_{8,t}(\mu_8, \sigma_8) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{8_1}}{\sigma_{8_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{8_2}}{\sigma_{8_2}}\right)^2\right\}\right],$$

and μ, σ for each node (cluster) are available in Appendix C.3.

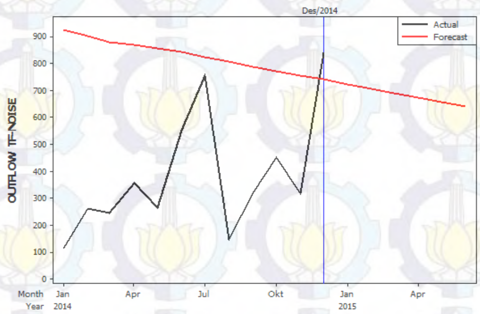
In building the hybrid model, we combine the forecast of Transfer Function-noise model and Radial Basis Function Neural Network model. Table 4.4 shows that the percentage improvements of Root Mean Square Error (RMSE) and Mean Absolute Deviation (MAD) of Hybrid model for in-sample data. The slight reduction of RMSE and MAD are 5.97% and 4.39% respectively. However, the hybrid model can't reduce the error as time horizon extend to 12 months ahead. The RMSE and MAD go up slightly 0.88% and 0.93% respectively. Figures 4.7 and Figures 4.8 show the comparison between actual observations and the forecasts from the in-sample and out-sample horizons of Transfer Function-noise model and Hybrid model of Banda Aceh Outflow series respectively.

Table 4.4: Measurement of Performance Test for Outflow Series of Banda Aceh

Model	In-Sample		Out-Sample	
	RMSE	MAD	RMSE	MAD
TF-ARIMA NOISE	170.30	120.38	510.24	460.80
HYBRID	160.14	115.09	514.71	465.07

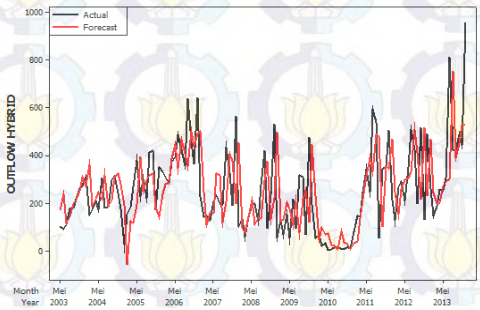


(a) In-sample

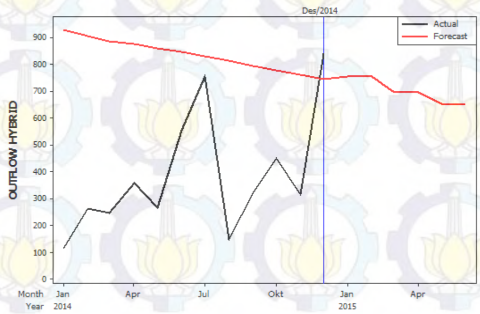


(b) Out-sample

Figure 4.7: Actual vs Transfer Function-Noise prediction of Outflow Banda Aceh



(a) In-sample



(b) Out-sample

Figure 4.8: Actual vs Hybrid prediction of Outflow Banda Aceh

4.2.2 Inflow of Banda Aceh

Similar manner to Banda Aceh Outflow series, an ARIMA model is fitted to the Consumer Price Index (CPI) of Lhokseumawe and its first difference is stationary series (Figure 4.6.a). Sample ACF and PACF plots in Figure 4.6(b) and Figure 4.6(c) suggest that the subset ARIMA([6],1,0) model should be used to fit the Consumer Price Index (CPI) of Lhokseumawe data. Then this model is used as a filter to prewhiten the input and output series. The CCF between α_t and β_t is given in Appendix B.6.

The CCF output indicate that there is a delay of 1 lag in the system (Appendix B.9), then we specify $b = 1$. There are several spikes appear after lag 1, we specify the order of s are subset 2 and we set the denominator as zero, that is, r=0. From the output in Appendix B.10 and Appendix B.11, the subset ARIMA(0,0,[1,12,13]) is the most parsimony model for noise term. The appropriate significant coefficients estimation are $\omega_0 = 1.13$, $\omega_2 = 1.03$, $\theta_1 = 0.51$, $\theta_{12} = -0.46$, and $\theta_{13} = 0.49$ (Appendix B.8). These coefficients are significant since they are less than $\alpha = 10\%$. Thus the Transfer Function-noise model for Inflow series of Banda Aceh is given as follow:

$$\begin{aligned} y_t &= (\omega_0 - \omega_2 L^2)L^1 x_t + (1 - \theta_1 L - \theta_{12} L^{12} - \theta_{13} L^{13})a_t \\ &= (1.13 - 1.03L^2)x_{t-1} + (1 - 0.51L + 0.46L^{12} - 0.49L^{13})a_t \end{aligned} \tag{4.8}$$

Next step is to forecast the Transfer Function-noise model in (4.8). The relationship of residual of Inflow Banda Aceh with its first and second lags is not nonlinear (Table 4.1). The residual a_t is modeled by using Radial Basis Function Neural Network. The optimal estimated weights of Inflow Banda Aceh series model are $\omega_1 = 11.64$, $\omega_2 = -35.39$, $\omega_3 = -18.24$, $\omega_4 = 1.73$, $\omega_5 = -13.51$, $\omega_6 = -2.65$, $\omega_7 = -12.99$, $\omega_8 = 10.89$, $\omega_9 = -0.06$, $\omega_{10} = -3.64$, $\omega_{11} = 22.32$, $\omega_{12} = 7.45$, $\omega_{13} = -2.52$, $\omega_{14} = -0.42$, $\omega_{15} = 12.58$, $\omega_{16} = 6.60$, $\omega_{17} = -3.55$, $\omega_{18} = 26.26$, $\omega_{19} = -16.48$, $\omega_{20} = -2.22$, $\omega_{21} = 49.49$, $\omega_{22} = -0.31$, $\omega_{23} = -29.55$, $\omega_{24} = -10.12$, $\omega_{25} = -0.01$, and the bias weight $\omega_b = -7,89$. Thus, RBF neural network model for $a(t)$ is as follow

$$\begin{aligned} a(t) &= \sum_{j=1}^{10} \omega_j \Omega_{j,t}(\mu_j, \sigma_j) + \omega_b \\ &= 11.64\Omega_{1,t}(\mu_1, \sigma_1) - 35.39\Omega_{2,t}(\mu_2, \sigma_2) + \cdots + \\ &\quad - 0.01\Omega_{25,t}(\mu_{25}, \sigma_{25}) - 7,89 \end{aligned} \tag{4.9}$$

where

$\Omega_{1,t}(\mu_1, \sigma_1) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{1_1}}{\sigma_{1_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{1_2}}{\sigma_{1_2}}\right)^2\right\}\right]$,

$\Omega_{2,t}(\mu_2, \sigma_2) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{2_1}}{\sigma_{2_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{2_2}}{\sigma_{2_2}}\right)^2\right\}\right]$,

$\vdots$

$$\Omega_{25,t}(\mu_{25}, \sigma_{25}) = \exp\left[-\frac{1}{2}\left\{\left(\frac{a_{t-1}-\mu_{25_1}}{\sigma_{25_1}}\right)^2 + \left(\frac{a_{t-2}-\mu_{25_2}}{\sigma_{25_2}}\right)^2\right\}\right],$$

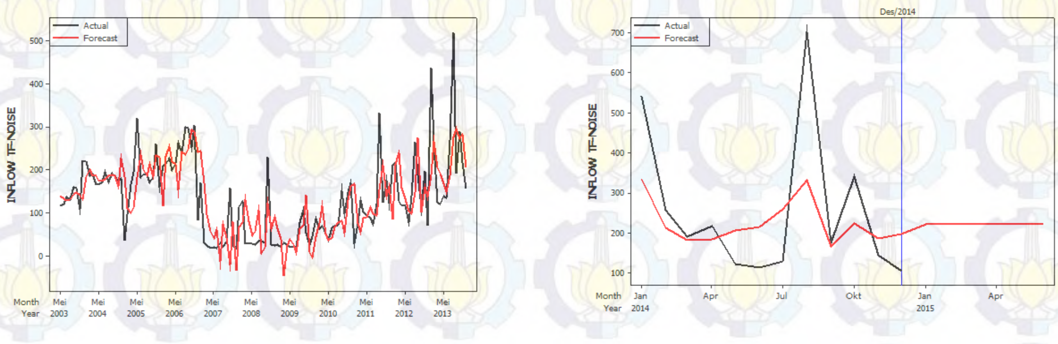
and μ, σ for each node (cluster) are availiable in Appendix C.4.

Table 4.5 shows that the Root Mean Square Error (RMSE) and the Mean Absolute Deviation of in-sample hybrid model can be slightly reduced by 4.19% and 3.71% respectively. However, RMSE and MAD slightly rise by 1.34% and 0.79% respectively. It is no surprise since the relationship of Inflow Banda Aceh residual and its first and second lags doesn't indicate the nonlinear relationship. Figures 4.9 and Figures 4.10

Table 4.5: Measurement of Performance Test for Outflow Series of Banda Aceh

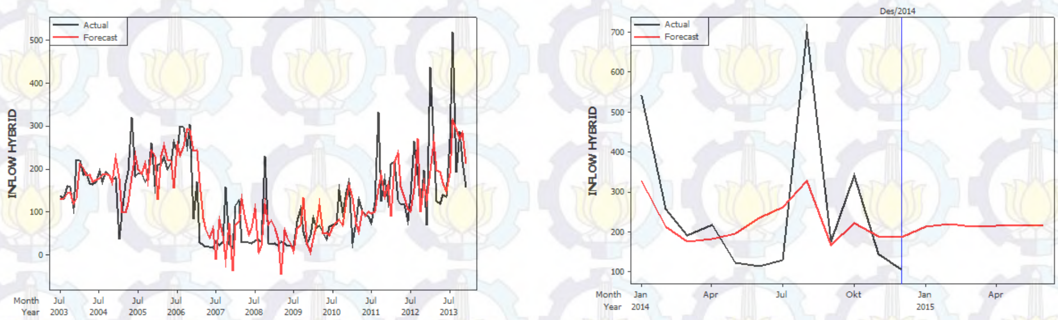
Model	**In-Sample**		**Out-Sample**	
	RMSE	MAD	RMSE	MAD
TF-ARIMA NOISE	66.75	47.13	142.13	104.15
HYBRID	65.95	45.38	144.04	104.97

show the comparison between actual observations and the forecasts from the in-sample and out-sample horizons of Transfer Function-noise model and Hybrid model of Banda Aceh Inflow series respectively.



(a) In-sample (b) Out-sample

Figure 4.9: Actual vs Transfer Function-Noise prediction of Outflow Banda Aceh



(a) In-sample (b) Out-sample

Figure 4.10: Actual vs Hybrid prediction of Inflow Banda Aceh

4.2.3 Calendar Variation

The main objective of outflow and inflow modeling is ultimately to predict the direction and level of change in a given series. However, this financial series are quantitative measures of given behavior of underlying factors (holiday, consumers, businesses, traders, etc.). Given that the behavior of economic factors, the four hybrid model we have developed do not show the superior models in out-sample. This poor forecasts may probably affected by local and national condition in Aceh and Indonesia in a certain months, particularly during June to August of 2014. For instance, the Idul Fitri day fell in July in the year of 2014. In this period, the money coming out/in from/to Bank Indonesia is irregularly different from the normal daily activities.

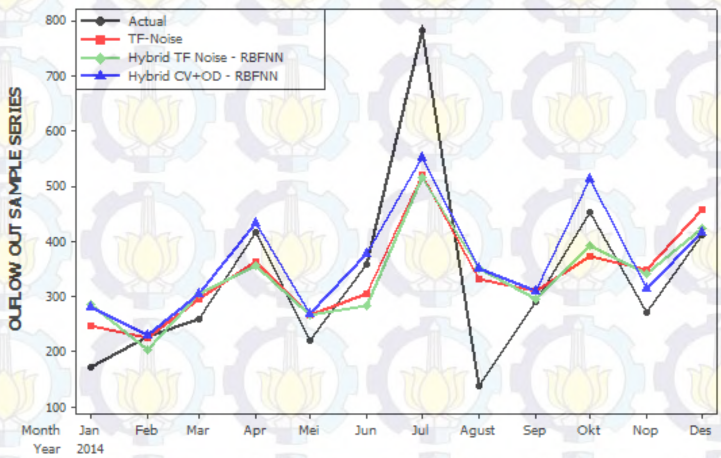


Figure 4.11: Out-Sample Forecasts of Outflow Lhokseumawe

Such behavior is clearly tied to the calendar. Therefore, in order to produce better forecast in out-sample, the Transfer Function-noise model should be subjected to calender variation and outlier detection. For example, we built Transfer Function with subject to calender variation and outlier detection for outflow series of Lhokseumawe. The result shows the model which subject to Calender Variation and Outlier Detection provide better accuracy in comparison with the individual Transfer Function-noise and Hybrid models.

Table 4.6: Performance Test of Out-Sample of Lhokseumawe Outflow Series

Model	**RMSE**	**MAD**
TF-ARIMA NOISE	106.76	79.14
HYBRID TF-NOISE RBFNN	112.53	82.70
HYBRID CV+OD RBFNN	100.15	66.76

Figure 4.11 shows that the forecast of Hybrid Calender Variation with Outlier Detection and RBFNN (Hybrid CV+OD - RBFNN) model follows the pattern of actual out-sample series slightly. The RMSE and MAD of these out-sample forecast can be seen in Table 4.5. From this table, we see that the Hybrid CV+OD - RBFNN has the smallest RMSE and MAD in comparison to the Transfer Function noise and Hybrid Transfer Function noise-RBFNN models. It is because some of the outlier points are overcame in calender variation and outlier detection. Based on the better forecast result of the Hybrid Calender Variation with Outlier Detection - RBFNN, we believe that the other four series of Inflow Lhokseumawe, Outflow Banda Aceh, and Inflow Banda Aceh will produce the better forecast accuracy performance.

This page is intentionally left blank

CHAPTER 5
CONCLUSIONS

The research for improving the effectiveness of forecasting models is continuously conducted. In this study, we have obtained the appropriate Transfer Function-noise models, Radial Basis Function Neural Network models, and Hybrid models for the series of outflow and inflow of Lhokseumawe and Banda Aceh representative offices by using input variable, i.e. Consumer Price Index (CPI).

This study demonstrates that by combining both linear and nonlinear models, forecasting performance can be improved over each individual model, particularly in training data (in-sample). Although both Transfer Function-noise and Neural Network models have the flexibility in modeling a various problems, none of them is the commonly best model that can be used blindly in every forecasting situation. The appropriate model classes should be tested before implementing the statistical analysis. All Hybrid models that are constructed in this study do not outperform well in out-sample data. Many factors can affected these findings, among other are, there are one high peak data point in out-sample period (July 2014). This solely value ruins the constructed model, since it is considered as the outlier.

Based on the given results in chapter 4, we investigate that the poor performance in out-sample data of three Hybrid models are probably affected by at least four reasons: The *first*, the length of time horizon for out-sample data. In M-3 Competition, the performance of forecasting methods depend on the length of forecast horizons. The combination model should be more useful the longer the forecast horizon (Amstrong, 2001). The *second*, it is because the accuracy measurement. RMSE and MAD probably do not appropriate for the type of the given time series data. Makridakis and Hibon (2003) explained in M3-Competition that the performance of the various methods although they can better fit a statistical model to depends upon the length of the forecasting the available historical data. The *third*, the Radial Basis Function perform poorly in noisy data (Bullinaria, 2004).) Since the series that we modeled in Neural Network is the residual of Transfer Function-noise data, this series is probably contaminated by huge noisy data. The *fourth*, The poor performance of out sample data from three series above is probably affected by peak points during July and August 2014. This conditions maybe affected by Idul Fitri day and can be overcame by applying calendar variation prior to doing neural network. We have shown that, the Hybrid model by subjecting to calender variation and outlier detection, performs better in out-sample in comparison the individual Transfer Function-noise model and Hybrid model.

Future research may be considered to choose the most appropriate or effective neural network model and the accuracy measurements for noisy data. In addition, Variation Calendar with outlier detection should be applied in the linear model part before analyzing the residual of non linear term by using neural network.

Appendix A
LHOKSEUMAWE REPRESENTATIVE OFFICE

A.1. TF-Noise Code of Outflow-Lhokseumawe Series with SAS

```
data Outflow_Lhok;
   input x y;
   y1=y/1000000000;
datalines;
293 131895758550
297 112359898601
.   .
.   .
635 549531837300
;
proc arima data= Outflow_Lhok;

   /*--- Look at the input process -----------------*/
   identify var=x(1);
   run;

   /*--- Fit a model for the input -----------------*/
   estimate p=0 q=0 method=cls;
   run;

   /*--- Crosscorrelation of prewhitened series ----*/
   identify var=y1(1) crosscorr=(x(1));
   run;

    /*- Fit a simple transfer function-look at
    residuals-*
   estimate input=( 0$ (1,13) / (0) x) plot
   method=cls;
   run;

   /*--- Final Model - look at residuals ---*/
    estimate p=(1,2)(12) q=(3,23) input=( 0$ (1) / (0)
    x)      noconstant  plot method=cls;
   run;

   /*--- Forecasting step ---*/
   forecast lead=18 out=out2 printall;
   run;

   /** Normality Test **/
   proc univariate data=out2 normal;
   var residual;
   run;

  /** export to file **/
   proc export data=out2
   outfile="D:\FORECAST OUTFLOW LHOK.xls"
   dbms=excel
   replace;
   run;
```

A.2. Parameter Estimation of Outflow Lhokseumawe

Conditional Least Squares Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
MA1,1	0.54914	0.10233	5.37	<.0001	3	y4	0
MA1,2	-0.22840	0.10127	-2.26	0.0259	23	y4	0
AR1,1	-0.69617	0.08734	-7.97	<.0001	1	y4	0
AR1,2	-0.59973	0.09989	-6.00	<.0001	2	y4	0
AR2,1	0.47345	0.09627	4.92	<.0001	12	y4	0
NUM1	5.55409	1.56670	3.55	0.0006	0	x	0
NUM1,1	4.47319	1.59257	2.81	0.0058	1	x	0

Variance Estimate	11918.14
Std Error Estimate	109.1702
AIC	1595.885
SBC	1615.957
Number of Residuals	130

* AIC and SBC do not include log determinant.

A.3. CCF of Outflow Lhokseumawe

```
                              Crosscorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0      161.447       0.17815   |                    .  |****           |
  1     -266.684       -.29427   |                ******|  .             |
  2    28.729331       0.03170   |                    .  |* .            |
  3   -35.940320       -.03966   |                    . *|  .            |
  4    18.464131       0.02037   |                    .  |  .            |
  5   -38.969185       -.04300   |                    . *|  .            |
  6    87.166128       0.09618   |                    .  |**.            |
  7   -36.115708       -.03985   |                    . *|  .            |
  8   -76.771783       -.08471   |                    .**|  .            |
  9      128.653       0.14196   |                    .  |***            |
 10     -103.831       -.11457   |                    .**|  .            |
 11      133.895       0.14775   |                    .  |***            |
 12    71.656828       0.07907   |                    .  |**.            |
 13     -165.150       -.18224   |                  ****|  .             |
 14     9.959267       0.01099   |                    .  |  .            |
 15   -36.374121       -.04014   |                    . *|  .            |
 16    -0.807161       -.00089   |                    .  |  .            |
 17     -124.274       -.13713   |                   ***|  .             |
 18      106.534       0.11756   |                    .  |**.            |
 19   -10.586527       -.01168   |                    .  |  .            |
 20   -15.592818       -.01721   |                    .  |  .            |
 21    27.113698       0.02992   |                    .  |* .            |
 22   -41.451786       -.04574   |                    . *|  .            |
 23      135.247       0.14924   |                    .  |***            |
 24   -53.567101       -.05911   |                    . *|  .            |
```

A.4. ACF of Outflow Lhokseumawe

```
                              Autocorrelations

Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1      Std Error

  0     24669.562        1.00000    |                    |********************|            0
  1     -9868.418        -.40002    |            ********|  .                 |     0.087370
  2     -2805.785        -.11373    |                . **|   .                |     0.100382
  3      2863.218        0.11606    |                .   |**  .               |     0.101361
  4     -2312.834        -.09375    |                . **|   .                |     0.102371
  5      -281.289        -.01140    |                .   |   .                |     0.103024
  6       163.667        0.00663    |                .   |   .                |     0.103034
  7      -425.354        -.01724    |                .   |   .                |     0.103037
  8      -850.544        -.03448    |                .  *|   .                |     0.103059
  9      6020.066        0.24403    |                .   |*****               |     0.103147
 10     -8397.874        -.34041    |             *******|   .                |     0.107464
 11      -619.942        -.02513    |               .   *|    .               |     0.115402
 12     11419.920        0.46292    |               .    |*********           |     0.115444
 13     -8730.918        -.35391    |             *******|    .               |     0.128837
 14      3584.444        0.14530    |               .    |***  .              |     0.136056
 15     -1612.274        -.06535    |               .   *|    .               |     0.137235
 16      -341.366        -.01384    |               .    |    .               |     0.137473
 17       122.438        0.00496    |               .    |    .               |     0.137483
 18       309.538        0.01255    |               .    |    .               |     0.137485
 19     -2175.214        -.08817    |               .  **|    .               |     0.137493
 20      1967.329        0.07975    |              .     |**    .             |     0.137924
 21      4335.497        0.17574    |              .     |****  .             |     0.138276
 22    -11282.074        -.45733    |           *********|     .              |     0.139971
 23      8160.407        0.33079    |              .     |*******             |     0.150947
 24      1295.583        0.05252    |              .     |*     .             |     0.156382
```

A.5. PACF of Outflow Lhokseumawe

```
                     Partial Autocorrelations

Lag    Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1       -0.40002   |            ********|  .                 |
  2       -0.32590   |             *******|  .                 |
  3       -0.09941   |                 .**|  .                 |
  4       -0.14968   |                 ***|  .                 |
  5       -0.13183   |                 ***|  .                 |
  6       -0.13123   |                 ***|  .                 |
  7       -0.12468   |                 .**|  .                 |
  8       -0.17009   |                 ***|  .                 |
  9        0.17835   |                 .  |****                |
 10       -0.23595   |               *****|  .                 |
 11       -0.33300   |             *******|  .                 |
 12        0.27891   |                 .  |******              |
 13       -0.05140   |                 . *|  .                 |
 14        0.13910   |                 .  |***                 |
 15       -0.06549   |                 . *|  .                 |
 16        0.00646   |                 .  |  .                 |
 17        0.00104   |                 .  |  .                 |
 18       -0.00233   |                 .  |  .                 |
 19       -0.02176   |                 .  |  .                 |
 20       -0.01495   |                 .  |  .                 |
 21        0.04864   |                 .  |* .                 |
 22       -0.24660   |               *****|  .                 |
 23        0.08858   |                 .  |**.                 |
 24       -0.00964   |                 .  |  .                 |
```

A.6. NN Code for Outflow-Lhok. Series with MATLAB

```
function RBFTS
% Radial Basis Function Time Series Model
clc;
data=dlmread('Data_Outflow_Lhok.txt');
[n,p]=size(data);
kluster=dlmread('5_Cluster_Outflow_Lhok.txt');

mean=kluster(:,1:5);
SD=kluster(:,6:10);

data1 = data(1:128,:); %data training
[n1,p1] = size(data1);
data2 = data(129:n,:);%data testing
[n2,p2] = size(data2);

p1 = transpose(data1(:,2:3));
p2 = transpose(data2(:,2:3));
t1 = (data1(:,1));
t2 = (data2(:,1));

 %---ProsesTraining---
H=rbfDesign(p1,mean,SD,'b');
lambda=globalRidge(H,t1,0.05);
w=inv(transpose(H)*H+lambda*eye(6))*transpose(H)*t1
yhat1=H*w;

 for i = 1:n1
    se(i) = (t1(i)-yhat1(i))^2;
    ys(i) = t1(i)^2;
    pe(i)=abs(t1(i)-yhat1(i));
end

sse = sum(se);
sst = (sum(ys)-(n1*((sum(t1))/n1)^2));
Rsquared1 = 1-(sse/sst);
RMSE1=sqrt((1/n1)*(sum(se)))
MAD = (1/n1)*(sum(pe))

%---ProsesTesting---
Ht=rbfDesign(p2,mean,SD,'b');
yhat2=Ht*w;

for i = 1:n2
    se(i) = (t2(i)-yhat2(i))^2;
    ys(i) = t2(i)^2;
end

sse = sum(se);
sst = (sum(ys)-(n2*((sum(t2))/n1)^2));
Rsquared2 = 1-(sse/sst);
Rsquared1 = 1-(sse/sst);
yhat1
yhat2
end
```

A.7. TF-Noise Code For Inflow-Lhokseumawe Series with SAS

```
data Inflow_Lhok;
   input x y;
   y2=y/1000000000;
datalines;
293 117183476500
297 77737708268
.   .
.   .
635 48822350850
;
proc arima data=Inflow_Lhok;

   /*--- Look at the input process -----------------*/
   identify var=x(1);
   run;

   /*--- Fit a model for the input -----------------*/
   estimate p=(1,2,8) q=(0) method=cls;
   run;

   /*--- Crosscorrelation of prewhitened series ----*/
   identify var=y2(1) crosscorr=(x(1));
   run;

    /*- Fit a simple transfer function-look at
    residuals-*
   estimate input=( 1$ (0) / (1) x) plot method=cls;
   run; **/

   /*--- Final Model - look at residuals ---*/
    estimate p=(0) q=(0) input=( 1$ (1,5,6,10,11)/(0)x)
    noconstant plot method=cls;
   run;

   /*--- Forecasting step ---*/
    forecast lead=13 out=out1 printall;
    run;

   /** Normality Test **/
     proc univariate data=out1 normal;
     var residual;
     run;

   /** export to file **/
    proc export data=out1
    outfile="D:\FORECAST INFLOW LHOK.csv"
    dbms=excel
    replace;
    run;
```

A.8. Parameter Estimation of Inflow Lhokseumawe

Conditional Least Squares Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag	Variable	Shift
NUM1	1.91715	0.96292	1.99	0.0489	0	x	1
NUM1,1	2.44395	0.94171	2.60	0.0107	1	x	1
NUM1,2	1.68930	0.96119	1.76	0.0815	5	x	1
NUM1,3	-2.03124	0.96166	-2.11	0.0369	6	x	1
NUM1,4	1.80971	0.96091	1.88	0.0622	10	x	1
NUM1,5	-1.82979	0.98954	-1.85	0.0671	11	x	1

Variance Estimate	4023.565
Std Error Estimate	63.43158
AIC	1331.242
SBC	1347.916
Number of Residuals	119

* AIC and SBC do not include log determinant.

A.9. CCF of Inflow Lhokseumawe

```
                         Crosscorrelations

Lag   Covariance    Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0    30.118202      0.07520     |                  .  |**.                |
  1    50.234833      0.12542     |                  .  |***                |
  2   -68.137913      -.17012     |                ***|  .                  |
  3    20.486235      0.05115     |                  .  |* .                |
  4   -35.355021      -.08827     |                 .**|  .                 |
  5    29.202134      0.07291     |                  .  |* .                |
  6   -53.881628      -.13452     |                ***|  .                  |
  7    65.796580      0.16427     |                  .  |***                |
  8   -29.530818      -.07373     |                  . *|  .                |
  9     1.967368      0.00491     |                  .  |  .                |
 10    11.824148      0.02952     |                  .  |* .                |
 11   -56.063127      -.13997     |                ***|  .                  |
 12    56.740153      0.14166     |                  .  |***                |
 13    -6.424888      -.01604     |                  .  |  .                |
 14    -7.486244      -.01869     |                  .  |  .                |
 15   -21.755858      -.05432     |                  . *|  .                |
 16   -31.885002      -.07961     |                 .**|  .                 |
 17    36.718610      0.09167     |                  .  |**.                |
 18   -56.746721      -.14168     |                ***|  .                  |
 19    63.066987      0.15746     |                  .  |***                |
 20   -35.548329      -.08875     |                 .**|  .                 |
 21    30.622369      0.07645     |                  .  |**.                |
 22   -19.830109      -.04951     |                  . *|  .                |
 23   -32.123876      -.08020     |                 .**|  .                 |
 24    89.955435      0.22459     |                  .  |****               |
```

A.10. ACF of Inflow Lhokseumawe

```
                          Autocorrelations

Lag   Covariance    Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1       Std Error

  0     4174.555       1.00000    |                    |********************|              0
  1    -2256.610       -.54056    |          ***********|  .                 |       0.087370
  2      233.599       0.05596    |                  .  |*  .                |       0.109976
  3      321.072       0.07691    |                  .  |**  .               |       0.110193
  4     -473.681       -.11347    |                  . **|  .                |       0.110602
  5    33.954745       0.00813    |                  .  |   .                |       0.111488
  6      177.068       0.04242    |                  .  |*  .                |       0.111492
  7      231.783       0.05552    |                  .  |*  .                |       0.111615
  8     -673.830       -.16141    |                  .***|  .                |       0.111826
  9      583.104       0.13968    |                 .   |***  .              |       0.113590
 10     -511.158       -.12245    |                 .  **|   .               |       0.114894
 11     3.450720       0.00083    |                 .   |    .              |       0.115886
 12     1027.839       0.24622    |                 .   |*****              |       0.115886
 13     -858.188       -.20558    |                 .****|   .               |       0.119813
 14      394.215       0.09443    |                 .   |**   .             |       0.122476
 15     -290.825       -.06967    |                 .   *|    .             |       0.123030
 16     9.617802       0.00230    |                 .   |    .              |       0.123331
 17    81.325600       0.01948    |                 .   |    .              |       0.123331
 18     -103.147       -.02471    |                 .   |    .              |       0.123355
 19   -49.322856       -.01182    |                 .   |    .              |       0.123393
 20     -126.682       -.03035    |                 .   *|    .             |       0.123401
 21      543.607       0.13022    |                 .   |***  .             |       0.123458
 22    -1290.688       -.30918    |               ******|    .              |       0.124502
 23     1742.412       0.41739    |                 .   |********           |       0.130232
 24     -812.742       -.19469    |               . ****|     .             |       0.140071
```

A.11. PACF of Inflow Lhokseumawe

```
                 Partial Autocorrelations

Lag   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1      -0.54056   |          ***********|  .                 |
  2      -0.33379   |              *******|  .                 |
  3      -0.10045   |                  .**|  .                 |
  4      -0.13975   |                  ***|  .                 |
  5      -0.17608   |                 ****|  .                 |
  6      -0.11068   |                  .**|  .                 |
  7       0.06576   |                  .  |*  .                |
  8      -0.12253   |                  .**|  .                 |
  9      -0.05033   |                  . *|  .                 |
 10      -0.15281   |                  ***|  .                 |
 11      -0.19240   |                 ****|  .                 |
 12       0.18098   |                  .  |****                |
 13       0.11120   |                  .  |**.                 |
 14       0.12393   |                  .  |**.                 |
 15       0.00425   |                  .  |  .                 |
 16      -0.00598   |                  .  |  .                 |
 17       0.05325   |                  .  |*  .                |
 18      -0.04337   |                  . *|  .                 |
 19      -0.13939   |                  ***|  .                 |
 20      -0.13709   |                  ***|  .                 |
 21       0.06979   |                  .  |*  .                |
 22      -0.28466   |               ******|  .                 |
 23       0.11037   |                  .  |**.                 |
 24       0.02827   |                  .  |*  .                |
```

A.12. NN Code for Inflow-Lhok. Series with MATLAB

```
function RBFTS
% Radial Basis Function Time Series Model
clc;
data=dlmread('Data_Inflow_Lhok.txt');
[n,p]=size(data);
kluster=dlmread('10_Cluster_Inflow_Lhok.txt');

mean=kluster(:,1:10);
SD=kluster(:,11:20);

data1 = data(1:117,:); %data training
[n1,p1] = size(data1);
data2 = data(118:n,:);%data testing
[n2,p2] = size(data2);

p1 = transpose(data1(:,2:3));
p2 = transpose(data2(:,2:3));
t1 = (data1(:,1));
t2 = (data2(:,1));

 %---ProsesTraining---
H=rbfDesign(p1,mean,SD,'b');
lambda=globalRidge(H,t1,0.05);
w=inv(transpose(H)*H+lambda*eye(11))*transpose(H)*t1
yhat1=H*w;

 for i = 1:n1
     se(i) = (t1(i)-yhat1(i))^2;
     ys(i) = t1(i)^2;
     pe(i)=abs(t1(i)-yhat1(i));
end

sse = sum(se);
sst = (sum(ys)-(n1*((sum(t1))/n1)^2));
Rsquared1 = 1-(sse/sst);
RMSE1=sqrt((1/n1)*(sum(se)))
MAD = (1/n1)*(sum(pe))

%---ProsesTesting---
Ht=rbfDesign(p2,mean,SD,'b');
yhat2=Ht*w;

for i = 1:n2
    se(i) = (t2(i)-yhat2(i))^2;
    ys(i) = t2(i)^2;
end

sse = sum(se);
sst = (sum(ys)-(n2*((sum(t2))/n1)^2));
Rsquared2 = 1-(sse/sst);
Rsquared1 = 1-(sse/sst);
yhat1
yhat2
end
```

Appendix B
BANDA ACEH REPRESENTATIVE OFFICE

B.1. TF-Noise Code For Outflow-Banda Aceh Series with SAS

```
data Outflow_Banda;
   input x y;
   y3=y/1000000000;
datalines;
296 194516945190
296 145005816220
.   .
.   .
749 954696414190
;
proc arima data=Outflow_Banda;

   /*--- Look at the input process -----------------*/
   identify var=x(1);
   run;

   /*--- Fit a model for the input -----------------*/
   estimate p=(6) q=(0) method=cls;
   run;

   /*--- Crosscorrelation of prewhitened series ----*/
   identify var=y3(1) crosscorr=(x(1));
   run;

    /*- Fit a simple transfer function-look at
    residuals-*
   estimate input=( 1$ (23) / (0) x) plot method=cls;
   run;  **/

   /*--- Final Model - look at residuals ---*/
   estimate p=(0) q=(0) input=( 1$ (0) / (0) x)
   noconstant plot method=cls;
   run;

   /*--- Forecasting step ---*/
   forecast lead=13 out=out1 printall;
   run;

   /** Normality Test **/
    proc univariate data=out1 normal;
    var residual;
    run;

   /** export to file **/
    proc export data=out1
      outfile="D:\FORECAST OUTFLOW Banda Aceh.xls"
      dbms=excel
      replace;
    run;
```

B.2. Parameter Estimation of Outflow Banda Aceh

```
                          Conditional Least Squares Estimation

                            Standard                 Approx
Parameter      Estimate        Error    t Value    Pr > |t|     Lag    Variable    Shift

NUM1           -4.89665      1.75703      -2.79      0.0061       0    x               1


                            Variance Estimate      29225.98
                            Std Error Estimate     170.9561
                            AIC                    1706.686
                            SBC                    1709.553
                            Number of Residuals         130
                        * AIC and SBC do not include log determinant.
```

B.3. CCF of Outflow Banda Aceh

```
                               Crosscorrelations

Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0       157.755        0.11577    |                 .  |**.                |
  1      -371.763        -.27283    |               *****|  .                |
  2     73.964991        0.05428    |                 .  |* .                |
  3     60.087365        0.04410    |                 .  |* .                |
  4     37.971945        0.02787    |                 .  |* .                |
  5    -54.763182        -.04019    |                 . *|  .                |
  6       145.291        0.10663    |                 .  |**.                |
  7     13.795015        0.01012    |                 .  |  .                |
  8    -96.938816        -.07114    |                 . *|  .                |
  9     64.017897        0.04698    |                 .  |* .                |
 10    -28.150483        -.02066    |                 .  |  .                |
 11       198.889        0.14596    |                 .  |***                |
 12    -61.814946        -.04536    |                 . *|  .                |
 13      -229.810        -.16865    |                 ***|  .                |
 14     63.155605        0.04635    |                 .  |* .                |
 15      -266.695        -.19572    |                ****|  .                |
 16     63.304485        0.04646    |                 .  |* .                |
 17    -11.463650        -.00841    |                 .  |  .                |
 18      4.460335        0.00327    |                 .  |  .                |
 19    -38.141364        -.02799    |                 . *|  .                |
 20     62.476374        0.04585    |                 .  |* .                |
 21     69.463711        0.05098    |                 .  |* .                |
 22      -137.188        -.10068    |                 .**|  .                |
 23       329.661        0.24193    |                 .  |*****              |
 24      -263.397        -.19330    |                ****|  .                |
```

B.4. ACF of Outflow Banda Aceh

```
                                    Autocorrelations

 Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1      Std Error

   0     30497.590        1.00000    |                    |********************|             0
   1    -13572.249        -.44503    |           *********|  .                 |      0.087370
   2     -1817.355        -.05959    |                .  *|   .                |      0.103234
   3      5514.063        0.18080    |                .   |****                |      0.103496
   4     -4535.557        -.14872    |                .***|   .                |      0.105880
   5      2056.601        0.06743    |                .   |*  .                |      0.107463
   6      -891.084        -.02922    |                .  *|   .                |      0.107785
   7       329.165        0.01079    |                .   |   .                |      0.107846
   8     -2582.622        -.08468    |                . **|   .                |      0.107854
   9      5565.434        0.18249    |                .   |****                |      0.108360
  10     -7370.749        -.24168    |               *****|   .                |      0.110681
  11     -1284.497        -.04212    |               .   *|    .               |      0.114639
  12     13559.499        0.44461    |               .    |*********           |      0.114757
  13    -11848.311        -.38850    |            ********|    .               |      0.127229
  14      4299.737        0.14099    |               .    |*** .               |      0.135983
  15      1184.444        0.03884    |               .    |*   .               |      0.137095
  16     -3209.619        -.10524    |               .  **|    .               |      0.137179
  17      1106.003        0.03627    |              .     |*    .              |      0.137794
  18     19.546315        0.00064    |              .     |     .              |      0.137866
  19     -1922.682        -.06304    |              .    *|     .              |      0.137866
  20      2188.984        0.07178    |              .     |*    .              |      0.138086
  21      3418.125        0.11208    |              .     |**   .              |      0.138371
  22    -10083.176        -.33062    |             *******|     .              |      0.139062
  23      5039.734        0.16525    |              .     |***  .              |      0.144938
  24      4739.551        0.15541    |              .     |***  .              |      0.146370
```

B.5. PACF of Outflow Banda Aceh

```
                         Partial Autocorrelations

 Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

   1       -0.44503    |          **********|  .                 |
   2       -0.32127    |              ******|  .                 |
   3        0.00388    |                 .  |  .                 |
   4       -0.08331    |                 .**|  .                 |
   5        0.00412    |                 .  |  .                 |
   6       -0.05064    |                 . *|  .                 |
   7        0.00584    |                 .  |  .                 |
   8       -0.13488    |                 ***|  .                 |
   9        0.13036    |                 .  |***                 |
  10       -0.17885    |                ****|  .                 |
  11       -0.24782    |               *****|  .                 |
  12        0.32181    |                 .  |******              |
  13        0.00875    |                 .  |  .                 |
  14        0.05516    |                 .  |* .                 |
  15       -0.00109    |                 .  |  .                 |
  16       -0.00505    |                 .  |  .                 |
  17       -0.09248    |                 .**|  .                 |
  18       -0.02808    |                 . *|  .                 |
  19       -0.09738    |                 .**|  .                 |
  20        0.05669    |                 .  |* .                 |
  21        0.10284    |                 .  |**.                 |
  22       -0.11355    |                 .**|  .                 |
  23       -0.09807    |                 .**|  .                 |
  24        0.00049    |                 .  |  .                 |
```

B.6. NN Code for Outflow-Banda Aceh Series with MATLAB

```
function RBFTS
% Radial Basis Function Time Series Model
clc;
data=dlmread('Data_Outflow_Banda.txt');
[n,p]=size(data);
kluster=dlmread('8_Cluster_Outflow_Banda.txt');

mean=kluster(:,1:8);
SD=kluster(:,9:16);

data1 = data(1:128,:); %data training
[n1,p1] = size(data1);
data2 = data(129:n,:);%data testing
[n2,p2] = size(data2);

p1 = transpose(data1(:,2:3));
p2 = transpose(data2(:,2:3));
t1 = (data1(:,1));
t2 = (data2(:,1));

 %---ProsesTraining---
H=rbfDesign(p1,mean,SD,'b');
lambda=globalRidge(H,t1,0.05);
w=inv(transpose(H)*H+lambda*eye(9))*transpose(H)*t1
yhat1=H*w;

 for i = 1:n1
     se(i) = (t1(i)-yhat1(i))^2;
     ys(i) = t1(i)^2;
     pe(i)=abs(t1(i)-yhat1(i));
end

sse = sum(se);
sst = (sum(ys)-(n1*((sum(t1))/n1)^2));
Rsquared1 = 1-(sse/sst);
RMSE1=sqrt((1/n1)*(sum(se)))
MAD = (1/n1)*(sum(pe))

%---ProsesTesting---
Ht=rbfDesign(p2,mean,SD,'b');
yhat2=Ht*w;

for i = 1:n2
     se(i) = (t2(i)-yhat2(i))^2;
     ys(i) = t2(i)^2;
end

sse = sum(se);
sst = (sum(ys)-(n2*((sum(t2))/n1)^2));
Rsquared2 = 1-(sse/sst);
Rsquared1 = 1-(sse/sst);
yhat1
yhat2
end
```

B.7. TF-Noise Code For Inflow-Banda Aceh Series with SAS

```
data Inflow_Banda;
   input x y;
   y4=y/1000000000;
datalines;
296 192016301650
296 122757197327
.   .
.   .
749 156646070000
;
proc arima data= Inflow_Banda;

   /*--- Look at the input process -----------------*/
   identify var=x(1);
   run;

   /*--- Fit a model for the input -----------------*/
   estimate p=(6) q=(0) method=cls;
   run;

   /*--- Crosscorrelation of prewhitened series ----*/
   identify var=y4(1) crosscorr=(x(1));
   run;

    /*- Fit a simple transfer function-look at
    residuals-*
   estimate input=( 1$ (2,24) / (0) x) plot
   method=cls;
   run;   **/

   /*--- Final Model - look at residuals ---*/
   estimate p=(0) q=(1,12,13) input=(1$ (2) / (0) x)
   noconstant plot method=cls;
   run;

   /*--- Forecasting step ---*/
    forecast lead=13 out=out1 printall;
    run;

   /** Normality Test **/
    proc univariate data=out1 normal;
    var residual;
    run;

   /** export to file **/
    proc export data=out1
      outfile="D:\FORECAST INFLOW Banda Aceh.xls"
      dbms=excel
      replace;
run;
```

B.8. Parameter Estimation of Inflow Banda Aceh

```
                           Conditional  Least  Squares  Estimation

                                Standard                 Approx
Parameter       Estimate           Error    t Value    Pr > |t|      Lag     Variable      Shift

MA1,1            0.50587         0.07423       6.82      <.0001        1     y1                0
MA1,2           -0.45781         0.08371      -5.47      <.0001       12     y1                0
MA1,3            0.49218         0.08673       5.68      <.0001       13     y1                0
NUM1             1.13032         0.51026       2.22      0.0286        0     x                 1
NUM1,1           1.03218         0.51015       2.02      0.0452        2     x                 1


                              Variance  Estimate      4637.173
                              Std Error  Estimate     68.09679
                              AIC                     1448.706
                              SBC                     1462.966
                              Number  of Residuals         128
                       * AIC and  SBC  do  not  include  log  determinant.
```

B.9. CCF of Inflow Banda Aceh

```
                                  Crosscorrelations

 Lag     Covariance     Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

   0      24.755723       0.03828     |                 .  |* .                 |
   1        144.363       0.22325     |                 .  |****                |
   2       -125.905       -.19471     |                ****|  .                 |
   3     -14.814889       -.02291     |                 .  |  .                 |
   4      13.048256       0.02018     |                 .  |  .                 |
   5      -5.400557       -.00835     |                 .  |  .                 |
   6     -44.747655       -.06920     |                 . *|  .                 |
   7      21.064965       0.03258     |                 .  |* .                 |
   8      49.431983       0.07644     |                 .  |**.                 |
   9     -41.603574       -.06434     |                 . *|  .                 |
  10      89.068900       0.13774     |                 .  |***                 |
  11     -77.650295       -.12008     |                 .**|  .                 |
  12      33.741757       0.05218     |                 .  |* .                 |
  13      31.975947       0.04945     |                 .  |* .                 |
  14     -92.240510       -.14265     |                 ***|  .                 |
  15      27.456729       0.04246     |                 .  |* .                 |
  16       -115.873       -.17919     |                ****|  .                 |
  17       2.743035       0.00424     |                 .  |  .                 |
  18     -29.404237       -.04547     |                 . *|  .                 |
  19      73.286081       0.11333     |                 .  |**.                 |
  20      17.541894       0.02713     |                 .  |* .                 |
  21      -8.484187       -.01312     |                 .  |  .                 |
  22     -60.948698       -.09425     |                 .**|  .                 |
  23     -56.683863       -.08766     |                 .**|  .                 |
  24        211.638       0.32729     |                 .  |*******             |
```

B.10. ACF of Inflow Banda Aceh

```
                                   Autocorrelations

Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1      Std Error

  0      7175.719        1.00000    |                    |********************|             0
  1     -3024.337        -.42147    |            ********|   .                |      0.087370
  2    -72.454471        -.01010    |                .   |   .                |      0.101713
  3       190.074        0.02649    |                .   |*  .                |      0.101721
  4      -968.705        -.13500    |                .***|   .                |      0.101774
  5       158.790        0.02213    |                .   |   .                |      0.103131
  6       683.055        0.09519    |                .   |** .                |      0.103168
  7       198.792        0.02770    |                .   |*  .                |      0.103836
  8     -1339.885        -.18672    |                ****|   .                |      0.103892
  9      1322.515        0.18430    |                .   |****                |      0.106423
 10     -1405.386        -.19585    |                ****|   .                |      0.108833
 11       507.550        0.07073    |                .   |*  .                |      0.111491
 12      1445.222        0.20140    |                .   |****                |      0.111833
 13     -1721.487        -.23990    |               *****|    .               |      0.114568
 14       758.017        0.10564    |               .    |**  .               |      0.118341
 15      -425.706        -.05933    |               .   *|    .               |      0.119058
 16       413.883        0.05768    |               .    |*   .               |      0.119284
 17     31.472689        0.00439    |               .    |    .               |      0.119496
 18       119.023        0.01659    |               .    |    .               |      0.119498
 19      -614.992        -.08570    |               .  **|    .               |      0.119515
 20       317.099        0.04419    |               .    |*   .               |      0.119983
 21    -12.051625        -.00168    |               .    |    .               |      0.120108
 22     -1365.994        -.19036    |               .****|    .               |      0.120108
 23      2590.895        0.36106    |               .    |*******             |      0.122389
 24     -1674.882        -.23341    |               *****|    .               |      0.130267
```

B.11. PACF of Inflow Banda Aceh

```
                      Partial Autocorrelations

Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1       -0.42147    |            ********|   .                |
  2       -0.22828    |               *****|   .                |
  3       -0.09615    |                 .**|   .                |
  4       -0.21240    |                ****|   .                |
  5       -0.18468    |                ****|   .                |
  6       -0.02262    |                 .  |   .                |
  7        0.07443    |                 .  |*  .                |
  8       -0.18958    |                ****|   .                |
  9        0.01687    |                 .  |   .                |
 10       -0.15966    |                 ***|   .                |
 11       -0.09350    |                 .**|   .                |
 12        0.15290    |                 .  |***                 |
 13       -0.09930    |                 .**|   .                |
 14       -0.01833    |                 .  |   .                |
 15       -0.06627    |                 . *|   .                |
 16        0.05753    |                 .  |*  .                |
 17        0.07325    |                 .  |*  .                |
 18       -0.03074    |                 . *|   .                |
 19       -0.06403    |                 . *|   .                |
 20        0.04794    |                 .  |*  .                |
 21       -0.04259    |                 . *|   .                |
 22       -0.23008    |               *****|   .                |
 23        0.14508    |                 .  |***                 |
 24       -0.06598    |                 . *|   .                |
```

B.12. NN Code for Inflow-Banda Aceh Series with MATLAB

```
function RBFTS
% Radial Basis Function Time Series Model
clc;
data=dlmread('Data_Inflow_Banda.txt');
[n,p]=size(data);
kluster=dlmread('25_Cluster_Inflow_Banda.txt');

mean=kluster(:,1:25);
SD=kluster(:,26:50);

data1 = data(1:126,:); %data training
[n1,p1] = size(data1);
data2 = data(127:n,:);%data testing
[n2,p2] = size(data2);

p1 = transpose(data1(:,2:3));
p2 = transpose(data2(:,2:3));
t1 = (data1(:,1));
t2 = (data2(:,1));

 %---ProsesTraining---
H=rbfDesign(p1,mean,SD,'b');
lambda=globalRidge(H,t1,0.05);
w=inv(transpose(H)*H+lambda*eye(26))*transpose(H)*t1
yhat1=H*w;

 for i = 1:n1
    se(i) = (t1(i)-yhat1(i))^2;
    ys(i) = t1(i)^2;
    pe(i)=abs(t1(i)-yhat1(i));
end

sse = sum(se);
sst = (sum(ys)-(n1*((sum(t1))/n1)^2));
Rsquared1 = 1-(sse/sst);
RMSE1=sqrt((1/n1)*(sum(se)))
MAD = (1/n1)*(sum(pe))

%---ProsesTesting---
Ht=rbfDesign(p2,mean,SD,'b');
yhat2=Ht*w;

for i = 1:n2
    se(i) = (t2(i)-yhat2(i))^2;
    ys(i) = t2(i)^2;
end

sse = sum(se);
sst = (sum(ys)-(n2*((sum(t2))/n1)^2));
Rsquared2 = 1-(sse/sst);
Rsquared1 = 1-(sse/sst);
yhat1
yhat2
end
```

Appendix C
MEAN AND STANDARD DEVIATION

C.1. Mean and Std Dev. of Outflow-Lhokseumawe Series

Node	Mean		Standard Deviation	
	a(t-1)	a(t-2)	a(t-1)	a(t-2)
1	227,3572	17,4772	84	63
2	-115,2643	95,2942	71	89
3	-45,4201	-139,197	68	63
4	181,2471	315,2544	144	109
5	15,7368	-4,6818	41	46

C.2. Mean and Std Dev. of Inflow-Lhokseumawe Series

Node	Mean		Standard Deviation	
	a(t-1)	a(t-2)	a(t-1)	a(t-2)
1	50,5073	-42,6517	24,03	20,77
2	-20,291	42,0123	17,16	15,08
3	-46,599	-9,1506	18,1	18,69
4	-0,7912	-50,1903	18,78	20,32
5	305,8038	-7,9928	151,6	31,1
6	-152,9477	337,8396	172,8	168,3
7	63,1766	-188,4451	25,7	83,1
8	2,068	-1,4059	13,13	12,91
9	44,3492	15,9422	25,71	14,37
10	-76,4921	106,4629	49,1	49,3

C.3. Mean and Std Dev. of Outflow-Banda Aceh Series

Node	Mean		Standard Deviation	
	a(t-1)	a(t-2)	a(t-1)	a(t-2)
1	118,0106	-377,0197	109,2	56,5
2	-385,2278	260,5599	155,7	176,8
3	-486,9374	-579,2334	128,8	140,6
4	-60,7462	-145,6787	77,7	76,8
5	308,5449	-66,7885	94,6	96,4
6	9,23	213,2161	81,5	71
7	-72,8808	76,1905	55,09	47,2
8	60,4051	6,7678	53,24	42,34

C.4. Mean and Std Dev. of Inflow-Banda Aceh Series

Node	Mean		Standard Deviation	
	a(t-1)	a(t-2)	a(t-1)	a(t-2)
1	-51,8621	-60,4684	18,28	19,23
2	40,5345	-37,1016	9,71	15,46
3	8,7752	18,6184	4,23	5,64
4	43,8105	11,089	12,54	7,15
5	16,4574	66,3908	11,78	20,71
6	-42,3573	3,5787	6,47	11,29
7	275,1982	-83,2961	84	44,2
8	-33,5201	115,263	31,4	13,36
9	-43,5496	43,2351	5,97	6,65
10	0,9162	-15,2145	5,87	4,9
11	-21,051	4,1794	3,65	1,88
12	-6,1926	-46,7131	11,68	4,81
13	-24,0184	-25,6437	9	7,41
14	21,1042	-1,4286	4,66	1,115
15	-14,0169	31,0756	1,97	12,34
16	22,0389	-87,5543	10,32	10,88
17	-2,4439	4,5981	3,32	3,67
18	12,0366	-2,9363	0,8	2,35
19	-83,6193	23,7438	16,22	19,73
20	-137,4037	-47,1264	19,68	48,8
21	-41,5338	-133,7259	30,3	22,64
22	92,1074	-15,7259	20,3	23,8
23	61,6631	55,3633	28,1	10,81
24	183,1395	45,2853	43,8	45,1
25	-34,023	237,0684	57,9	71,6

REFERENCES

Armstrong, J. S. (2001). 'Combining forecasts', Principles of Forecasting: *A Handbook for Researchers and Practioners*. Armstrong, J. S. (ed.), Norwell, MA: Kluwer Academic Publishers.

Ali, F. A. (2014). Feed Forward Neural Network For Sine Function With Symmetric Table Addition Method Using Labview And Matlab Code. *Intl. Journal on Computational Sciences & Applications (IJCSA)* Vol.4, No.2

Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis-Forecasting and Control*. Holden Day, Inc., Oakland, California.

Bisgaard, S. and Kulahci, M. (2011). *Time Series Analysis and Forecasting by Example*. John Wiley Sons, Inc., New Jersey.

Badan Pusat Statistik (2013). *Diagram Timbang Indeks Harga Konsumen Hasil Survei Biaya Hidup Tahun 2012*. Badan Pusat Satistik, Jakarta.

Bank Indonesia (2014). *Buku Petunjuk Tim Pengendali Inflasi Daerah*, Bank Indonesia, Jakarta.

Cleveland, W.S. Devlin, S.J. (1980). Calendar Effects in Monthly Time Series: Detection by Spectrum Analysis and Graphical Methods. *Journal of the American Statistical Association*, 75(371), 487-496.

Chatfield, C. (2000). *Time Series Forecasting*. Chapman Hall/CRC, London, UK

Campbell, P. R. J., Ahmed, F., Fathulla, H., and Jaffar, A. D. (2010). A Neural Network Based Approach to Wind Energy Yield Forecasting. *Advances in Neural Network Research and Application*, LNEE 67, ISBN 978-3-642-12989-6, pp.917924, Springer-Verlag Berlin Heidelberg.

Cryer, J. D., Chan, K. S. (2008). *Time Series Analysis With Applications in R*. Springer Science+Business Media, LLC, New York.

Dennis, J. E. and More, J. J. (1977). Quasi-Newton Method, Motivation and Theory. *Society for Industrial and Applied Maths.*. SIAM Review 19 (1977), pp. 46-89.

Dai, Y. H. (2002). Convergence Properties of the BFGS Algorithm. *Society for Industrial and Applied Maths.* Vol.13, No.3, pp.693-701, SIAM J. OPTIM.

Demuth, H. and Beale, M. (2002). *Neural Network Toolbox: For Use with MATLAB.* The MathWorks, Inc

Dreyfus, G. (2005). *Neural Networks Methodology and Applications.* Springer-Verlag Berlin Heidelberg.

De Gooijer, J. G. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting* 22, 443 473, Elsevier B.V.

Dunne, R. A. (2007). *A Statistical to Neural Networks for Pattern Recognition.* John Wiley Sons, Inc., New Jersey.

Ding, H., Liu, D., and Zhao, F. (2010). Variation Trend Analysis of Groundwater Depth in Area of Well Irrigation in Sanjiang Plain Based on Wavelet Neural Network. *Advances in Neural Network Research and Application*, LNEE 67, ISBN 978-3-642-12989-6, pp. 829836, Springer-Verlag Berlin Heidelberg.

Du, W., Leung, S. Y. S., Kwong, C. K. (2014). Time series forecasting by neural networks: A knee point-based multiobjective evolutionary algorithm approach. *Expert Systems with Applications* 41,80498061, Elsevier Science B.V.

De Mulder, W., Bethard, S., Moens, M. F., (2015). A survey on the application of recurrent neural networksto statistical language modeling. *Computer Speech and Language* 30, 6198, Elsevier Science B.V.

Enders, W. (2009). *Applied Econometric Time Series.* John wiley sons Inc., Iowa.

Freeman, J. A., Skapura, D. M. (1991). *Computation and Neural Systems Series.* Addison-Wesley Publishing Company, Inc., USA

Fine, T. L. (1999). *Feedforward Neural Network Methodology.* Springer-Verlag, New York Inc., USA

Fisher, J.D.M, Liu, C.T., dan Zhou, R. (2002), *"'When can we forecast inflation?"'*, Economic Perpective, page. 230-42.

Falk, M. (2006). *A First Course on Time Series Analysis, Chair of Statistics.* University of Wrzburg.

Fan, J., Shan, R., Cao, X. and Li, P. (2009). The Analysis to Tertiary-industry with ARIMAX Model. *Journal of Mathematics Research*, Vol. 1, No. 2.

Greene, W. H. (2008). *Econometric Analysis, Pearson Education.* Inc., New Jersey

Hu, M.Y., Shanker, M., and Ming, S. H. (2004). Predicting Consumer Situational Choice with Neural Networks. *Neural Networks in Business: Techniques and Applications*, ISBN 1-59140-176-3, p.172-194, Idea Group Inc

Hamilton, J. D. (1994). *Time Series Analysis.* Princeton University Press, New Jersey, UK.

Hagan, M.T., Demuth, H.B., Beale, M.H., and De Jesus, O. (1996). *Neural Network Design.* PWS Publishing Company: Boston.

Hanrahan, G. (2011). *Artificial Neural Networks in Biological and Environmental Analysis.* CRC Press- Taylor and Francis Group, LLC, New York

Kitchens, F. L., Johnson, J. D, and Gupta, J. N. D. (2004). Predicting Automobile Insurance Losses Using Artificial Neural Networks. *Neural Networks in Business: Techniques and Applications*, ISBN 1-930708-31-9, pp.167-187, Idea Group Publishing, USA.

Kirchgässner, G., Wolters, J. (2007). *Introduction to Modern Time Series Analysis.* Springer-Verlag Berlin Heidelberg.

Kagoda P.A., Ndiritu, J., Ntuli, C., Mwaka, B. (2010). Application of radial basis function neural networks to short-term streamflow forecasting. *Physics and Chemistry of the Earth*,35, 571581, Elsevier Science B.V.

Khandelwal, I., Adhikari, R. and Verma, G. (2015). Time Series Forecasting using Hybrid ARIMA and ANN Models based on DWT Decomposition. *Procedia Computer Science* 48, 173 179, Elsevier B.V.

Liu, L.M. (1980). Analysis of Time Series with Calendar Effects. *Management Science*, 2, 106-112.

Lajbcygier, P. (2002). Comparing Conventional and Artificial Neural Network Models for the Pricing of Options. *Neural Networks in Business: Techniques and Applications*, ISBN 1-930708-31-9, p.220 235, Idea Group Publishing.

Lütkepohl, H. and Krätzig, M. (2004). *Applied Time Series Econometrics.* Cambridge University Press, UK.

Li, L. K., Pang, W.K., Yu, W. T. and Troutt, M. D. (2004). Forecasting Short-Term Exchange Rates: A Recurrent Neural Network Approach. *Neural Networks in Business: Techniques and Applications*, ISBN 1-930708-31-9, pp.195-212, Idea Group Publishing, USA.

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis.* Springer-Verlag Berlin Heidelberg .

Lewis, Adrian, S., Overton, Michael (2009). Nonsmooth optimization via BFGS. *Society for Industrial and Applied Mathematics*, SIAM J. Optimiz.

Mankiw, N.G. (2000). *Teori Makro Ekonomi*, Erlangga, Jakarta.

Makridakis, S. and Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, 16, 451476. Elsevier Science B.V.

McCullough, B.D. (2001). Software reviews. *International Institute of Forecasters*, 17, 305322. Elsevier Science B.V.

Mary, E. M. and Salchenberger, L. (2002). Using Neural Networks to Discover Patterns in International Equity Markets: A Case Study. *Neural Networks in Business: Techniques and Applications*, ISBN 1-930708-31-9, p.205 219, Idea Group Publishing.

McNelis, P. D. (2005). *Neural Networks in Finance-Gaining Predictive Edge in the Market*. Elsevier Inc., USA.

Montgomery, D. C., Jennungs, C. and Kulahci, M. (2008). *Introduction to Time Series Analysis and Forecasting*. Willey Interscience, New Jersey.

Mohammadi,R., Ghomi, F., Zeinali, F. (2014). A New Hybrid Evolutionary Based RBF Networks Method for Forecasting Time Series: A case study of forecasting emergency supply demand time series. *Engineering Applications of Artificial Intelligence*,36, 204214, Elsevier Science B.V.

Peretto, P. (1992). *An Introduction to the Modelling of Neural Networks*. Cambridge University Press, Inc., San Diego.

Parsons, L. J. and Dixit, A. (2004). Using Artificial Neural Networks to Forecast Market Response. *Neural Networks in Business Forecasting*, ISBN 1-59140-176-3, p. 23 46, Idea Group Inc.

Pektas, A.O. and Cigizoglu, H. K. (2013). ANN hybrid model versus ARIMA and ARIMAX models of runoff coefficient. *Journal of Hydrology* 500 (2013) 2136, Elsevier B.V.

Robles, L. A. D., Ortega, J. C., Fu, J. S., Reed, G. D., Chow, J. C., Watson, J. G. and Herrera, M. J. A. (2008). A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco-Chile. *Atmospheric Environment* 42, 83318340, Elsevier B.V.

Suhartono (2008). *Analisis Data Statistik Dengan R*. Lab. Statistik Komputasi, ITS, Surabaya

Suhartono, Lee, M. H., Hamzah, N., A. (2010). Calendar variation model based on Time Series Regression for sales forecasts: The Ramadhan effects. *Proceedings of the Regional Conference on Statistical Sciences*, Malaysia.

Shumway, R. H., and Stoffer D. S. (2011). *Time Series Analysis and Its Application*. Springer Science+Business Media, LLC, London

Shukur, O. B. and Lee, M. H. (2015). Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA. *Renewable Energy* 76, 637-647, Elsevier B.V.

Walczak, S.(2004). Forecasting Emerging Market Indexes with Neural Networks. *Neural Networks in Business: Techniques and Applications*, ISBN 1-59140-176-3, p. 80 101, Idea Group Inc.

Wei, W. W. S. (2006). *Time Series Analysis: Univariate and Multivariate Method.* Pearson Education, Inc., New York.

Wang, S., Min, H., Sun, F., and Zhang, J. (2010). Neural Network Control of Spacecraft Formation Using RISE Feedback. *Advances in Neural Network Research and Application*, LNEE 67, ISBN 978-3-642-12989-6, pp. 521528, Springer-Verlag Berlin Heidelberg

Wang, J. and Wang, J. (2015). Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. *Neurocomputing* 156, 6878, Elsevier B.V.

Yaffee, R. A. and McGee, M. (1999). *Introduction to Time Series Analysis and Forecasting.* Academic Press, INC., New York.

Yao, J. T. and Tan, C. L. (2002). Neural Networks for Technical Forecasting of Foreign Exchange Rates. *Neural Networks in Business: Techniques and Applications*, ISBN 1-930708-31-9, p.189 204. Idea Group Publishing

Yu, L., Wang, S. and Lai, K. K. (2007). *Foreign Exchange Rate Forecasting with Artificial Neural Networks.* Springer Science+Business Media, LLC, New York.

Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 3562. Elsevier B.V.

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

Zhang, G. P. (2004). *Business Forecasting with Artificial Neural Networks: An Overview.* Idea Group Inc., USA

Zhang, G. P. (2004), A Combined ARIMA and Neural Network Approach for Time Series Forecasting. *Neural Networks in Business: Techniques and Applications*, ISBN 1-59140-176-3, p.213-225, Idea Group Inc.

Zhu, J., Z., Cao, J., X., Zhu, J. (2014). Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. *Transportation Research Part C*, www.elsevier.com/locate/trc, Elsevier Science B.V.

This page is intentionally left blank

BIOGRAPHY

I was born in Aceh, Matang Glumpang Dua, 1978. I completed bachelor study at Syiah Kuala University in 2003. As of 2005, I joined Save the Children (International Non-profit Organization) to help Achenese people in rehabilitation and reconstruction program. I have five years experienced with INGO in implementing programs, communicating with partners and stakeholders, arranging and sensitizing programs of organization policies to public community and government. I implemented a comprehensive M&E system that meets information for program requirement, internal and external reporting.

At the beginning of my career as a lecturer in 2008, I was joined State Polytechnic of Lhokseumawe, Aceh, Indonesia. As a lecturer, I taught Mathematics and Statistics to the related undergraduate students. In addition, I also taught the similar courses to the other students at the other universities as a visiting lecturer.

For the purpose of my career, I continued my study to master level at Humbold Universität, Berlin-Germany in 2012. I majored in statistics focusing on time series and econometric. Due to several reasons, I decided to choose university transfer to Institut Teknologi Sepuluh Nopember Surabaya (ITS), Indonesia in 2015. Here in ITS, I wrote my master thesis in line with the specialization I have chosen in my previous study, time series.

My future career will be still focused on time series and econometric. I hope I can develop more advanced topics in time series and econometrics in order to solve the related practical issues in my country. The Ph.D level in the similar field is required to support my career. Any further communication with me can be contacted via the following email address: fahmi_mfj@yahoo.com

Surabaya, Februari 2016

Fahmi