ตัวแบบค่าเฉลี่ยเคลื่อนที่ถดถอยอัตโนมัติชนิดหลายจุดเปลี่ยน

นางสาวพิมสิริ ผลทรัพย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

MULTIPLE CHANGE-POINT AUTOREGRESSIVE MOVING AVERAGE MODEL

Miss Pimsiri Ponsap

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Applied Mathematics and Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2012

Thesis Title	MULTIPLE CHANGE-POINT AUTOREGRESSIVE MOVING AVERAGE MODEL
By	Miss Pimsiri Ponsap
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Assistant Professor Krung Sinapiromsaran, Ph.D.
Thesis Co-advisor	Phantipa Thipwiwatpotjana, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

....Chairman

(Associate Professor Paisan Nakmahachalasint, Ph.D.)

....Thesis Advisor

(Assistant Professor Krung Sinapiromsaran, Ph.D.)

....Thesis Co-advisor

(Phantipa Thipwiwatpotjana, Ph.D.)

....Examiner

(Kittipat Wong, Ph.D.)

....External Examiner

(Thidaporn Supapakorn, Ph.D.)

พิมสิริ ผลทรัพย์ : ตัวแบบค่าเฉลี่ยเคลื่อนที่ถดถอยอัตโนมัติชนิดหลายจุดเปลี่ยน. (MULTIPLE CHANGE-POINT AUTOREGRESSIVE MOVING AVERAGE MODEL) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. กรุง สินอภิรมย์สราญ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ดร. พันทิพา ทิพย์วิวัฒน์พจนา, 57 หน้า.

ตัวแบบค่าเฉลี่ยเคลื่อนที่ถดถอยอัตโนมัติชนิดหลายจุดเปลี่ยน ใช้สำหรับทำนายค่าในอนาคตบนอนุกรมเวลาที่ไม่คงที่ โดยใช้กลยุทธการลดตัวอย่างเพื่อกำหนดจุดเปลี่ยนบนอนุกรมเวลา กลยุทธดังกล่าว ทดสอบสมมติฐานการกระจายอย่างเป็นปกติของเศษเหลือจากการทำนายด้วยตัวแบบถดถอยอัตโนมัติในการหาจุดเปลี่ยน ถ้าเศษเหลือจากการทำนายมีการกระจายตัวไม่ปกติ ข้อมูลทางซ้ายมือของอนุกรมเวลาจะถูกนำออกทีละจุด จนเศษเหลือจากการทำนายมีการกระจายตัวปกติหรือจุดตัวอย่างเหลือไม่เพียงพอ ซึ่งกลุ่มข้อมูลที่มีเศษเหลือกระจายตัวปกตินี้จะถูกจัดเก็บไว้และกลุ่มข้อมูลสุดท้ายที่อยู่ใกล้จุดที่ต้องการทำนายมากที่สุด จะถูกใช้เป็นข้อมูลสำหรับการสร้างโมเดลการทำนาย ตัวแบบที่นำเสนอนี้ได้ทดสอบประสิทธิภาพการทำนาย ของตัวแบบค่าเฉลี่ยเคลื่อนที่ถดถอยอัตโนมัติชนิดหลายจุดเปลี่ยน กับการทำนายข้อมูล Sunspot ในปี 1920 ถึง 2008 นอกจากนี้ยังทดลองทำนายราคาปิดหุ้นล่วงหน้าหนึ่งวันเป็นเวลาทั้งสิ้น 26 วัน จากหุ้นทั้งหมดสิบตัว ได้แก่ หุ้น ADVANC, AOT, BANPU, CPALL, DTAC, JAS, KBANK, LOXLEY, PTT และ STANLY โดยใช้ข้อมูลราคาในปี 2010 ถึง 2012 ซึ่งได้ผลการทำนายที่ได้ดีกว่าตัวแบบอื่น ได้แก่ ตัวแบบ threshold autoregressive, ตัวแบบ autoregressive integrated moving average และ ตัวแบบ generalized autoregressive conditional heteroskedasticity โดยรายงานตัววัดการทำนายด้วย mean absolute percent error, mean absolute error, root mean square error และ mean square error

ภาควิชา คณิตศาสตร์ประยุกต์และวิทยาการคณนา ลายมือชื่อนิสิต ....

สาขาวิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ....

ปีการศึกษา 2555 ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม ....

5373884123 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : AUTOREGRESSIVE MODEL / NORMALITY

PIMSIRI PONSAP : MULTIPLE CHANGE-POINT AUTOREGRESSIVE MOVING AVERAGE MODEL. ADVISOR : ASST. PROF. KRUNG SINAPIROMSARAN, Ph.D., CO-ADVISOR : PHANTIPA THIPWIWATPOTJANA, Ph.D., 57 pp.

This thesis presents a new method to obtain multiple change-points. These change-points are used in a Multiple Change-point AutoRegressive Moving Average (MARMA) model for a fluctuated time series prediction. A change-point is captured by using a sample reduction strategy. The statistical residual normality test is used to validate the change-point detection in our strategy. If the residual series is not normally distributed, the initial point will be removed until the residual series has normality or not enough sample is left. The in-sample dataset is selected from the dataset in the last cluster. In this framework, we examine one-step ahead prediction of an annual sunspot data between the year 1920 and 2008 and ten specific daily closing prices of Thai Stock Exchange, during the year of 2010 to 2012. These ten indices are ADVANC, AOT, BANPU, CPALL, DTAC, JAS, KBANK, LOXLEY, PTT and STANLY. The result of the multiple change-point autoregressive moving average models obtains a better performance than the threshold autoregressive models; the autoregressive integrated moving average models and the generalized autoregressive conditional heteroskedasticity models using R programming. The prediction accuracy is reported by the mean absolute error, root mean square error, mean absolute percentage error and mean square error.

Department : Mathematics and Computer Science Student's Signature..............................

Field of Study : Applied Mathematics and Advisor's Signature..............................

Computational Science Co-advisor's Signature..............................

Academic Year : 2012

ACKNOWLEDGEMENTS

Firstly, I am very thankful to Assistant Professor Dr. Krung Sinapiromsaran and Dr. Phantipa Thipwiwatpotjana, my advisor and co-advisor. Without their suggestion, this thesis would never been completed. In addition, I would like to thank my friends in the Applied Mathematics and Computational Science laboratory for their friendships during my study at Chulalongkorn University. And the last but not least, I feel very grateful to my family for their compassion and encouragement throughout my life.

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

1.1 Motivation and Literature Surveys

An AutoRegressive (AR) process is applied in a time series forecasting for many years, for instance, the stock price prediction. Javier, et al. [1] used AutoRegressive Integrated Moving Average (ARIMA) models to forecast the next-day electricity price in Spanish Market and Californian Market. Nochai et al. [2] predicted the Thai oil prices by using ARIMA models and suggested the order of the ARIMA model that yielded the small values of the Mean Absolute Percentage Error (MAPE).

The AR model is a probabilistic model (Williams [3]). Its prediction capability comes from a stationary time series assumption. A Threshold AutoRegressive (TAR) model, exhibited by Tong [4], extends the capability of an AR model in a time series prediction. More specifically, the TAR model has an advantage to predict a shifted time series. Tong proved that the TAR process has flexibility in building the different AR models of each shifted regime. An initial process of the TAR model starts with identifying a threshold variable; however, Tong did not present an exact experimental procedure for the threshold identification. Tsay [5] introduced an effective method to identify a threshold variable by using a simple linear regression technique. Bermejo et al. [6] proposed a method to identify the threshold values and compared their results with the results that were proposed by Tong [4] and Tsay [5]. Their experiments were examined on the same dataset, the annual sunspot series and logged lynx data which were reported with lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values than Tong [4] and Tsay [5].

Furthermore, TAR models were used to investigate a stock market movement. Dijk et al. [7] proposed the discussion on the self-exciting TAR model for the Standard & Poor's 500 Index (S&P) prediction. Narayan [8] proposed the acceptance of a unit root in the TAR method to predict the behavior of stock markets in USA.

However, TAR models encounter the difficulty to determine threshold variables (Tsay [5]) and it has a limited prediction capability for some characteristics of a time series (Gibson [9]). Hence, a method that combines two or more techniques was proposed as an alternative model for predicting a volatile time series. This combined method is called a hybrid model. For instance, Pai and Lin [10] presented their hybrid model, which was constructed from an ARIMA model and a Support Vector Machines (SVMs) model, to forecast Taiwan Stock Exchange. Merh, Saxena and Pardasani [11] simulated the daily open, close, high and low prices of Indian stock market by using a hybrid model which includes three layers back propagation Artificial Neural Network (ANN) and the ARIMA model. The conclusion of their research was that their hybrid model outperformed the ARIMA model or ANN model alone. Areekul et al. [12] proposed a short-term price forecasting in the deregulated market by using the model that was constructed from the ARIMA and ANN model. Its result evaluated by the statistical measurement had a better performance than the pure ARIMA or ANN model.

In Economics, the heteroskedasticity happens when a sub-population of a time series has different variability from the others. The behavior of the heroskedasticity shows that the variance of the prediction errors in an AR process is not a constant, but the ordinary least square (OLS) test still holds for the parameter estimation. Consequently, the use of this AR model leads to the wrong inferences according to the heteroskedasticity. An AutoRegressive Conditional Heteroskedasticity (ARCH) model, popularized by Engle [13], was proposed for estimating the heteroskedasticity. The studies of the ARCH were widely used in the stock market prediction, for instance, Mahajan and Singh [14] analyzed the Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) model capability in prediction of the stock returns in India. Liu et al. [15] investigated that the return distribution of the stock market had an influence on the forecast accuracy by using two GARCH models. Even though the various methods of reducing the prediction errors were introduced as mentioned above, the AR process still plays an important role in the time series predictions. The ARIMA models, based on the AR process, are commonly used in comparison with other models, see Table 1.1.

This thesis proposes a Multiple change-point AutoRegressive Moving Average (MARMA) model. The method of the MARMA model requires a simple reduction strategy and statistical technique. The sample reduction strategy is used to capture a change-point by the residual normality test. If the residual series is not normally distributed, the initial point in the time series will be removed until the residual series has normality or the number of samples is too low. We define the first point that causes the residual distribution depart from normality, as the change-point. The MARMA model can be used for predicting a volatile time series, such that the time series shifted up or down by some unknown factors such as stock market circumstances. This research compares the prediction accuracy of the ARIMA, TAR, GARCH and MARMA models. The MARMA models are built for the stock prices of the specific ten indexes in Thai Stock Exchange, in the year 2012 and then the prediction accuracy is compared to ARIMA, TAR and GARCH models. The ten indices are ADVANC, AOT, BANPU, CPALL, DTAC, JAS, KBANK, LOXLEY, PTT and STANLY. The MARMA model is also used for forecasting the sunspot data. We compare the sunspot data result by MARMA model to the one by the TAR model, proposed by Bermejo et al., [6]. This result is reported in Appendix A.

1.2 Research Objective

The goals of this research are to predict the next day closed prices of the ten indices in Thai Stock Exchange by using the MARMA model and to compare the MARMA performance with the ARIMA, TAR and Generalized ARCH.

1.3 Overview

We organize the thesis as follows. After this introduction chapter, in Chapter II, we provide the background knowledge and methodologies used in this thesis. In Chapter III, we present the results of using MARMA models. In chapter IV, we present the conclusion. The detail on sunspot data prediction result is in the Appendix A, and the Appendix B shows R code implementing in the MARMA model.

Table1.1**:** The comparison of ARIMA models and other prediction models.

Publication	**Data**	**Model**	**Criterion**	**Criterion Value**
(a) Pai and Lin [10]	Taiwan Stock Exchange (2002)	SVM	MAPE	1.1433
		Hybrid model		0.7593
		ARIMA		1.1494
(b) Hassan Md.R., et al. [16]	The daily stock price of Apple Computer Inc.(2003)	ANN-GA-HMM-Interpolation	MAPE	2.16429
		ARIMA		1.8009
(c) Yeh C., et al. [17]	DS-V indexes in TAIEX (2004)	SKSVR	RMSE	45.686
		MKSVR		45.634
		ARIMA		45.421
(d) P. Areekul., et al. [12]	Australian national electricity market (2006)	ARIMA-ANN	MAPE	15.62946
		Seasonal ARIMA		16.06611

From Table 1.1, SVM, MAPE, GA, HMM, SKSVR, RMSE and MKSVR stand for Support Vector Machine, Mean Average Percent Error, Genetic Algorithm, Hidden Markov Model, Single-Kernel Support Vector Regression, Root Mean Square Errors and Multiple-Kernel Support Vector Regression, respectively.

CHAPTER II

BACKGROUND KNOWLEDGE AND METHODOLOGY

This chapter provides the background knowledge and methodologies that are used in this study. It consists of four main sections. First, we introduce a method to reduce an in-sample series to a stationary time series. The second section, we describe an estimation of AR parameters by using Yule-Walker Equation. The method of an AR order determination is also included. The third section, we describe a normality test for a residual series and the sample reduction strategy. The last section, the prediction procedure of the MARMA model is explained.

2.1 Data Transformation and Stationary Validation

Lo and Mackinlay [18] proved that the stock market was predictable in some degree. Their theory showed that the prices in the stock market were shifted by **trends**. Hence, some past pattern series can be used for the future prediction. Their proposed equation is called the simple volatility-based specification.

$$X\left(t\right)=\mu+X\left(t-1\right)+\varepsilon_t, \tag{2.1}$$

where $X\left(t\right)$ is the price of a stock index at time t, μ is an arbitrary drifted trend parameter, and ε_t is a random disturbance term.

Definition 2.1.1 A time series is said to be **strictly stationary** for $t_1,t_2,...,t_n$ if the joint distribution of $X\left(t_1\right),...,X\left(t_n\right)$ is the same as the joint distribution of $X\left(t_1+\tau\right),...,X\left(t_n+\tau\right)$ for integer $\boldsymbol{\tau}$.

We denote $\{x_t\}_{t=1}^{\infty}$ as a time series. In Equation (2.1), it implies that the stock market time series $\{x_t\}_{t=1}^{\infty}$ is not stationary. The mean and variance of $\{x_t\}_{t=1}^{\infty}$ is changed by the trend μ hence, the time series $\{x_t\}_{t=1}^{\infty}$ is transformed into the stationarity by decomposing μ. For non-seasonal time series, first-order differencing is usually sufficient to attain apparent stationarity [19]. The first-order differencing formula is given as

$$\nabla x_t = x_t - x_{t-1}, \tag{2.2}$$

where x_t is the value of the time series at particular time t. Box and Jenkins [20] introduced to difference a given non-seasonal time series until it becomes stationary. The ***d***-order differencing is required using the operation ∇^d [19], where

$$\nabla^d x_{t+d} = \nabla^{d-1} x_{t+d} - \nabla^{d-1} x_{t+d-1}. \tag{2.3}$$

Definition 2.1.2 The process $\{x_t\}_{t=1}^{\infty}$ is the **differencing stationary process** of order d, if it satisfies $\nabla^d x_t = \Pi(\mathrm{t})$ for all t where $\{\Pi(t)\}_{t=1}^{\infty}$ is a stationary process and ***d*** is a positive integer.

In the case of a time series has a **seasonality component**; Chatfield [19] introduced three seasonal models.

$$\begin{aligned} &\text{Model A} \qquad X(t) = m_t + S(t) + \varepsilon_t \\ &\text{Model B} \qquad X(t) = m_t S(t) + \varepsilon_t \\ &\text{Model C} \qquad X(t) = m_t S(t) \varepsilon_t \end{aligned} \tag{2.4}$$

where m_t is the deseasonalized mean level at time t, $S(t)$ is the seasonal effect at time t, and ε_t is the random error. The model A is an additive case that the variation of $S(t)$ appears to be roughly constant in size. The models B and C are multiplicative cases that the variations of $S(t)$

increase amplitude over time. In our experiment, the multiplicative or additive seasonal time series are observed by a time series plot. Figure 2.1 shows the examples of the non-seasonal time series, the additive seasonal time series [2] and the multiplicative seasonal time series, respectively. The seasonal components are estimated by using the “decompose” function in R. The function manages the seasonal figures by averaging, for each time unit, over all periods. The period is determined by the time series plot. We remove the seasonal components from the time series by subtracting for the additive case or dividing for the multiplicative case.

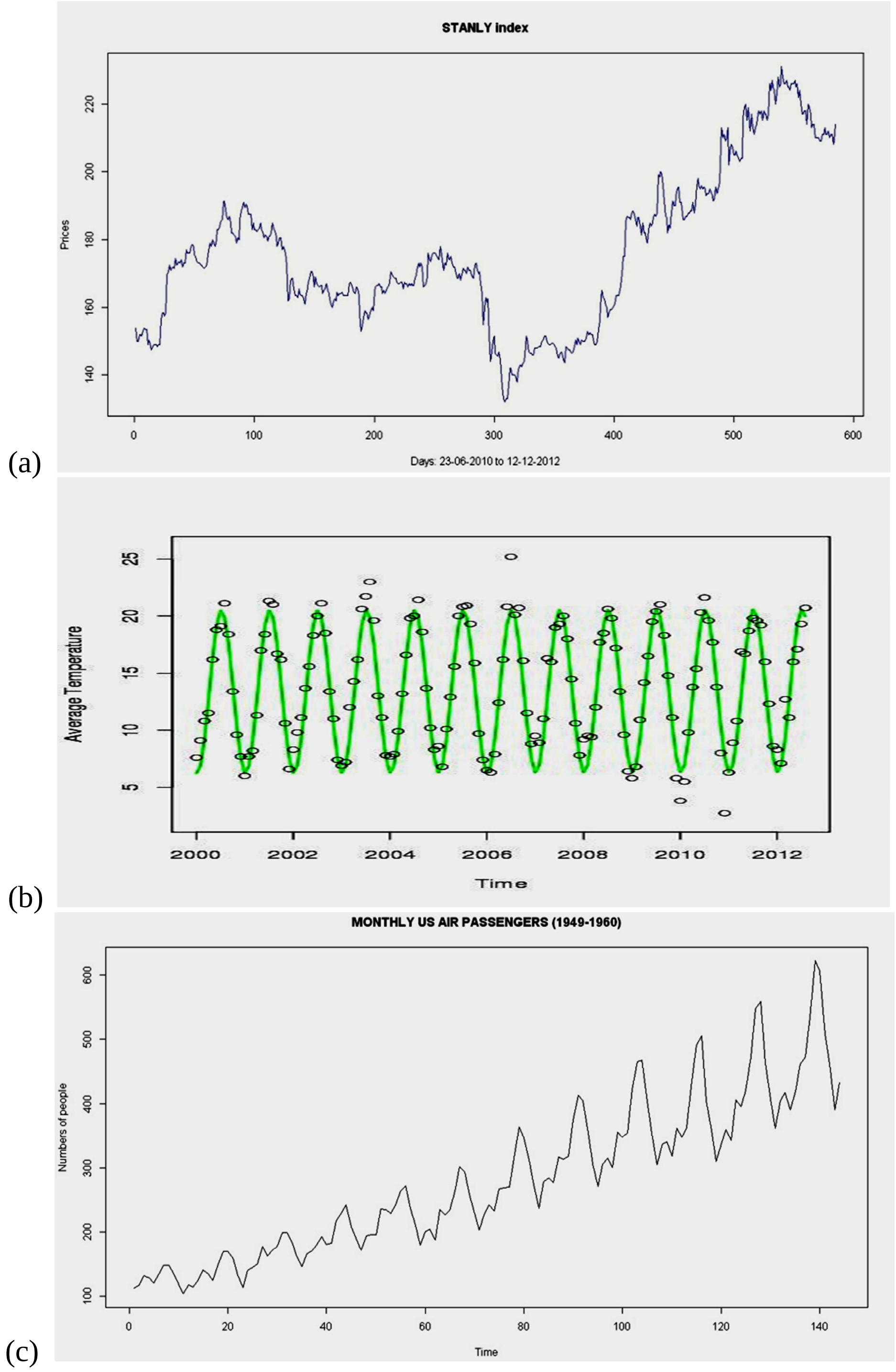


Figure 2.1: The respective time series plots of (a) the non-seasonal case, (b) the additive case and (c) the multiplicative case.

We identify the stationary time series by using the statistical unit root test.

Definition 2.1.3 The process $\{x_t\}_{t=1}^{\infty}$ is **a unit root process** if it satisfies $\nabla x_t = \Pi(t)$, where $\{\Pi(t)\}_{t=1}^{\infty}$ is a stationary process.

For a given time series, the null hypothesis states that "the time series is a unit root process" against the alternative hypothesis as "the time series is not a unit root process". The unit root test is performed by the method of the Augmented Dickey Fuller (ADF) test. The 'adf.test' function on R programming is used to calculate the test statistic [22] of the ADF test. The null hypothesis is rejected if the test statistic is less than the critical value at $\alpha = 0.05$ significant level and then the given time series is accepted the stationarity.

2.2 Parameter Estimation

Definition 2.2.1 A process $\{y_t\}_{t=1}^{\infty}$ is said to be an **AutoRegressive (AR) process** of order ***p*** if

$$y_t = \psi_1 y_{t-1} + \psi_2 y_{t-2} + ... + \psi_p y_{t-p} + z_t, \tag{2.5}$$

where $\psi_1, \psi_2, ..., \psi_p$ is the set of the parameters and $\{z_t\}_{t=1}^{\infty}$ is a **purely random process**.

Definition 2.2.2 A time series, $\{z_t\}_{t=1}^{\infty}$, is said to be a **purely random process** if it satisfies the following properties:

(a) $E[z_t] = 0,$

(b) σ_z^2 is a constant and

(c) $Cov(z_t, z_{t+k}) = 0$, where $t > k,\ k = \pm 1, \pm 2, ...$.

The residual is the difference between the real data y_t at time t and the predicted value $\hat{y}_t$; i.e.,

$$z_t = y_t - \hat{y}_t. \tag{2.6}$$

Definition 2.2.3 Suppose that $\{z_t\}_{t=1}^{\infty}$ is a purely random process with a mean zero and constant variance σ_z^2. A process $\{\Omega_t\}_{t=1}^{\infty}$ is said to be a **moving average process** of order q if

$$\Omega_t = z_t + \beta_1 z_{t-1} + ... + \beta_q z_{t-q}. \tag{2.7}$$

The Yule-Walker Equation in [19] is used for the parameter estimations of the in-sample series, $\{y_t\}_{t=1}^{n}$. The equation is

$$R\psi = r , \tag{2.8}$$

where $R = \begin{pmatrix} 1 & r_1 & \cdots & r_{p-1} \\ r_1 & 1 & \cdots & r_{p-2} \\ \cdots & r_1 & \cdots & \cdots \\ r_{p-1} & r_{p-2} & \cdots & 1 \end{pmatrix}$, $\psi^T = (\psi_1, \psi_2, ..., \psi_p)$, and $r^T = (r_1, r_2, ..., r_p)$.

The values of r_i, $i \in \{1, ..., p\}$, is calculated by

$$r_i = \frac{c_i}{c_0}, \tag{2.9}$$

where $c_i = \sum_{t=1}^{n-i} \frac{(y_t - \bar{y})(y_{t+i} - \bar{y})}{n}$, and $c_0 = \sum_{t=1}^{n} (y_t - \bar{y})^2$. The order of an AR model is determined by Akaike Information Criterion (AIC) value, [19]. We can automatically calculate from the AIC function providing in R program. The order of an AR model is selected from the value $\boldsymbol{p}$ that gives the smallest value of AIC.

In the AR process, a very small number of time series data is used to build an AR model may not effective because the AR parameter calculation is based on the mean of data, see Equation (2.9). We set the smallest number of in-sample series for the AR process as $\boldsymbol{u}$ =20.

2.3 Residual Normality Test and Sample Reduction Strategy

From Definition 2.2.1, the prediction errors are from the AR process, whose residual values have a constant mean and variance. The residual distribution is taken into the consideration by using the statistical normality test.

Definition 2.3.1 Let $\{z_t\}_{t=1}^{\infty}$ be a time series random variable. Then, $\{z_t\}_{t=1}^{\infty}$ has a **normality** if it is normally distributed.

Suppose that $\hat{y}_t$ is the predicted values of y_t by using the AR model of the order p and $z_{p+1}, z_{p+2}, ..., z_n$ are the in-sample prediction errors, where $z_t = y_t - \hat{y}_t$ for all $t \in \{p+1, p+2, ..., n\}$ .We use the Shapiro-Wilk test [23] (the statistical normality test) to detect the normal distribution of a residual series, $\{z_t\}_{t=1}^{\infty}$ using its sample $\{z_t\}_{t=p+1}^{n}$. The test statistics of the Shapiro-Wilk test is calculated by

$$\omega = \frac{\beta^2}{\zeta^2}. \tag{2.10}$$

δ_i is defined as the order statistics of z_i, $\delta_i = z_{[i]}$, where $i \in \{1, 2, ..., n-p\}$ so that $\delta_1 \le \delta_2 ... \le \delta_{n-p}$. The value of β is estimated by

$$\beta = \sum_{i=1}^{k} a_{\eta-i+1}\left(\delta_{\eta-i+1} - \delta_i\right), \tag{2.11}$$

where η is the number of residuals, $\eta = n - p$. The value of k is calculated by $k = \frac{\eta}{2}$ when the value of η is the even number, or $k = \frac{\eta - 1}{2}$ when the value of η is the odd number. The values of $\{a_{\eta-i+1}\}_{i=1}^{k}$ are the normalized coefficients that were proposed by Sarhan and Greenberg [24] for $\eta \le 20$ and Royston, [25] and [26], for $\eta > 20$. The value of ζ is calculated by the formula,

$$\zeta = \sum\nolimits_{t=p+1}^{n} \left(z_t - \overline{z}\right)^2 , \qquad (2.12)$$

where $\overline{\boldsymbol{z}}$ is the mean of $\{z_t\}_{t=p+1}^{n}$.

To show the calculation of the test statistic, suppose a residual series $\{z_t\}_{t=p+1}^{p+7}$ is $\{7, 2, 1, 5, 3, 6, 9\}$, namely $z_{p+1} = 7, z_{p+2} = 2, z_{p+3} = 1, z_{p+4} = 5, z_{p+5} = 3, z_{p+6} = 6$ and $z_{p+7} = 9$. The test statistic is provided by

(i) ordering $\{z_t\}_{t=p+1}^{p+7}$ to obtain $\delta_1 = 1,\ \delta_2 = 2,\ \delta_3 = 3,\ \delta_4 = 5,\ \delta_5 = 6,\ \delta_6 = 7$ and $\delta_7 = 9,$

(ii) calculating β , from [23] $a_7 = 0.6233, a_6 = 0.3031, a_5 = 0.1401, a_4 = 0.000$, thus $\beta = 0.6233(9-1) + 0.3031(7-2) + 0.1401(6-3) + 0 = 6.9222$, and

(iii) estimating ζ , $\zeta = \sum\nolimits_{t=p+1}^{p+7} \left(z_t - \overline{z}\right)^2 = 49.4285$, and then $\omega = \dfrac{6.9222^2}{49.4285} = 0.1400$.

The null hypothesis of the Shapiro-Wilk test is that the prediction residuals are normally distributed. We determine the null hypothesis acceptance at 0.05 significance level. In the case when the null hypothesis is rejected, we propose a new method called **the sample reduction strategy**. The sample reduction strategy operates on the time series to detect the series with normality residual.

Definition 2.3.2 Let $\{y_t\}_{t=1}^{\infty}$ be a time series and $\hat{y}_t$ be the predicted value that is estimated by AR model of the order p , $\hat{y}_t = \psi_1 y_{t-1} + ... + \psi_p y_{t-p}$, where $\psi_1, \psi_2, ..., \psi_p$ are the parameters; $t \in \{p+1, p+2, ...\}$. z_t is the residual value at time t , where $z_t = y_t - \hat{y}_t$. $\{y_t\}_{t=d}^{k}$ is said to be a **cluster** in the time series $\{y_t\}_{t=1}^{\infty}$ if $\{z_t\}_{t=d+p}^{k}$ is normally distributed and the series $\{z_t\}_{t=d+p-f}^{k+g}$ is not normally distributed for all positive integers g and f that $g + f > 0$.

We perform a sample reduction strategy to search for a cluster in time series, see Figure 2.2-2.5 which we repeatedly remove an initial point of the time series until the cluster is found or not enough sample points to test. The AR model is rebuilt for every recursion; i.e. we keep removing the first point of a current set of the time series whose residuals fail the Shapiro-Wilk test, until we obtain the first cluster whose residuals are normality or not enough samples is left.

In the case that we follow the method of removing the first point recursively, until we have only the last point but the Shapiro-Wilk test still fail to detect normality; we define this in-sample series to be in a **border**.

After identifying a border, we need to remove the last point of the in-sample series. It forces us to have a new considered interval which is one datum smaller than the old one. We redo the process of reduction strategy again; until we can obtain the next cluster, see Figure 2.6-2.8.

We cluster time series by using the property of the residual normality, and then we consider the time series in any two consecutive clusters having no border in order to indicate the point that is called **a change-point**. Tsay [5] proposed the method for testing the nonlinearity in time series as in [27]. His proposed method is used to identify whether a linear AR model or a nonlinear model is better in describing the time series. We use the method of Tsay [5] to examine our two consecutive clusters that there is no border between both clusters. If the time series from both clusters cannot be modeled by AR model while the time series from each cluster can modeled separately by two AR models, and then the last point in the time series of the previous cluster is called a change-point. We exhibit step by step to identify an AR(p) model in the given time series in the example 2.3.1.

The Algorithm 1 presents the procedure of the sample reduction strategy, clustering time series and change-point identification in a finite time series $\{y_t\}_{t=1}^{n}$.

Algorithm 1:

```
$n :=$ the number of all in-sample data.
$d := 1,\ i := 0,\ j := 0$, and $k := n$.
$D := \{y_t\}_{t=d}^{k}$
$u := 20;$
# Define $o_i$ as the clusters of the order $i$,
                    # $cp_i$ as the time index of the change-point of order $i$
                    # $b_j$ as the time index of the border of order $j$ .
                    # $o_i$ as the time series in the cluster $i$ .
                    # $t$ as the time indices, where
                    # $\psi_1, \psi_2, ..., \psi_p$ as the parameters of an AR model that are generated by
                    the Yule-Walker equation. $p$ as the order of an AR model that is
                    evaluated by using AIC value.
                    # $D$ as an in-sample time series
                    # $z_t$ as a residual.
$O := \{\},\ CP := \{\},\ B := \{\},\ T := \{\},$
# Step 1:      (AR Model Construction)
               $D := \{y_t\}_{t=d}^{k}$
               # Build the AR( $p$ ) model from D, generate $\psi_1, \psi_2, ..., \psi_p$ and $p$
                    for all $t \in \{d+p, d+p+1, ..., k\}$, $\hat{y}_t := \psi_1 y_{t-1} + \psi_2 y_{t-2} + ... + \psi_p y_{t-p}$
                              and $z_t := y_t - \hat{y}_t$ .
# Step 2:      (Residual Normality Test and Stationarity Test)
               # Examine $\{z_t\}_{t=d+p}^{k}$ using the Shapiro-Wilk test and examine $\{y_t\}_{t=d}^{k}$
               # using the ADF test at 0.05 significance level. There are two possible cases.
               # Case I: (New Cluster Found)
                    If
```

```
        {the Shapiro-Wilk test accepts the null hypothesis and the ADF test
            rejects the null hypothesis}
        then
        #update i
        {$i := i+1$, $o_i := \{y_d, y_{d+1}, ..., y_k\}$ and $O := O \cup \{o_i\}$
            If $d = 1$ then stop the algorithm.
            If $d \neq 1$ then
                #add the elements to the set of the change-points,
                $cp_i := d-1$; $CP := CP \cup \{cp_i\}$
                #and update k,
                $k := d-1$,
                #and update d,
                $d := 1$.
                    If $k-d < u$ then go to Step 4.
                    If $k-d \geq u$ then recur Step 1.}
        Else,
    # Case II: (Non-normality and/or Non-stationarity, and then Removing Initial
    Point)
        {$d := d+1$.
            If $k-d \geq u$ , then the algorithm recurs Step 1.
            If{ $k-d < u$ },
            then
            #(Border Identification)
            { $j := j+1$, $b_j := k$, $B := B \cup \{b_j\}$, $d = 1$ and $k := k-1$,
                If $k-d < u$ then go to Step 4.
                If $k-d \geq u$ then go to Step 1.
                }
            }
#Step4:     # After all clusters in the finite time series, $\{y_t\}_{t=1}^{n}$ are determined, we identify
            the change-point by examining the time series in two consecutive clusters
            whether the series is able to model by AR model.
            Set $H_0$: time series is able to be modeled by AR model.
            # r:=running time of 'for' looping function
            for (r in 1:i) {
                    if { $cp_r \in o_1 \vee ... \vee cp_r \in o_i$ }
                     then {
                    [a]:=examine $H_0$ in the time series $o_{r+1}$,
                    [b]:= examine $H_0$ in the time series $o_r$
                    [c]:=examine $H_0$ in the gathering time series $O_{r-1}$ and $O_r$.
                    If [a] accepts $H_0$ and [b] accepts $H_0$ but [c] rejects $H_0$, we do
                    not delete $cp_r$ from $CP$.
                    Otherwise, delete $cp_r$ from $CP$.
                    }
            } End.
```

Remarks:

(a) If $\{1, 2, ..., n\} \in O$, $\{y_t\}_{t=1}^{n}$ will be used in the prediction process,

(b) if $O = \phi$, the MARMA model is unable to use,

(c) if $n \in B$, the MARMA model is unable to use.

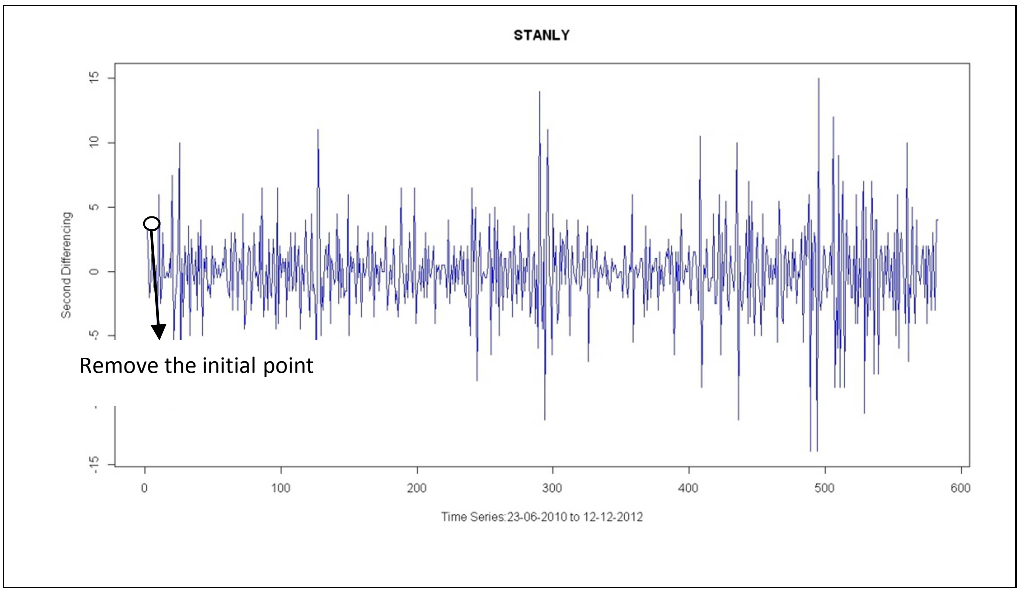


Figure 2.2: Removing the first point when the residual is not normally distributed.

Figure 2.2 shows the sample reduction strategy to reduce the in-sample series of STANLY index after the series is decomposed a trend by differencing method. We remove points for achieving the first cluster (see Figure 2.2), which is the same as our new considered interval in Figure 2.3. We continue using Shapiro-Wilk test to remove points until we got our second cluster as in Figure 2.4.

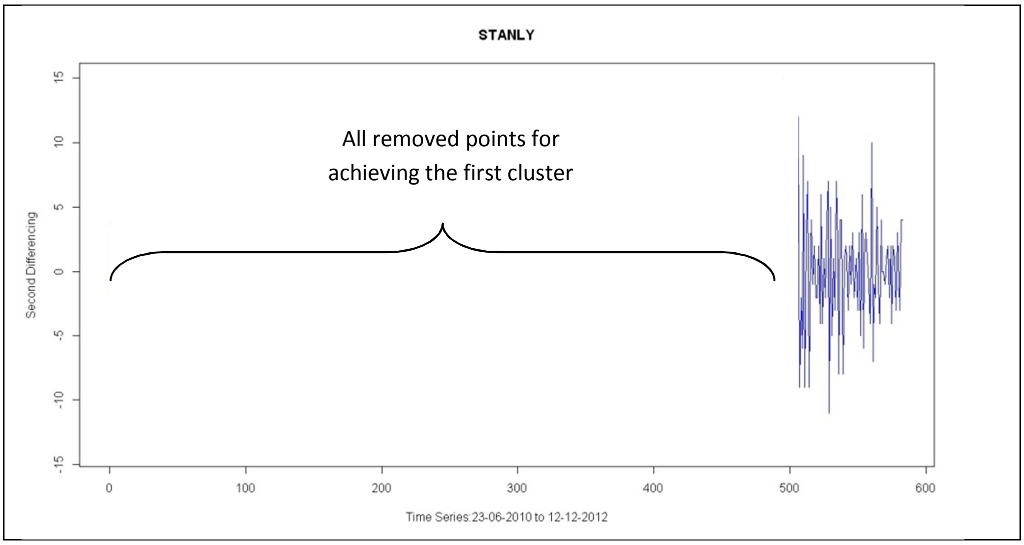


Figure 2.3: The first (normal distributed residuals) cluster after performing the recursion process.

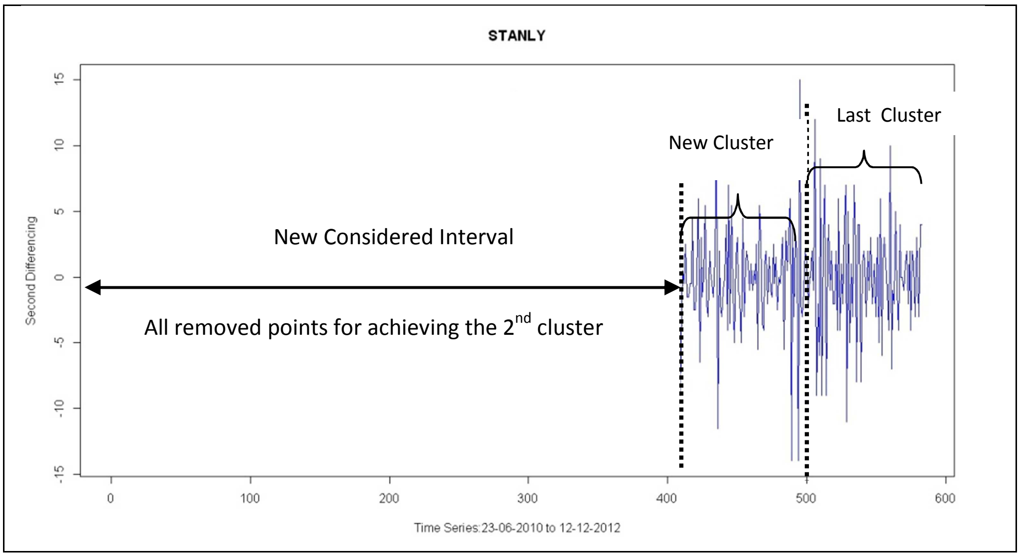


Figure 2.4: The example of searching the clusters in time series.

We continue the process until we can divide our in-sample series into clusters as in Figure 2.5. The time series of STANLY index has five clusters that each consecutive cluster has a successive time index. Even though we always obtain the 1st cluster at the last partition of time index in the in-sample data, we rename the clusters by running the cluster number from the smallest to the largest time index, as in Figure 2.5.

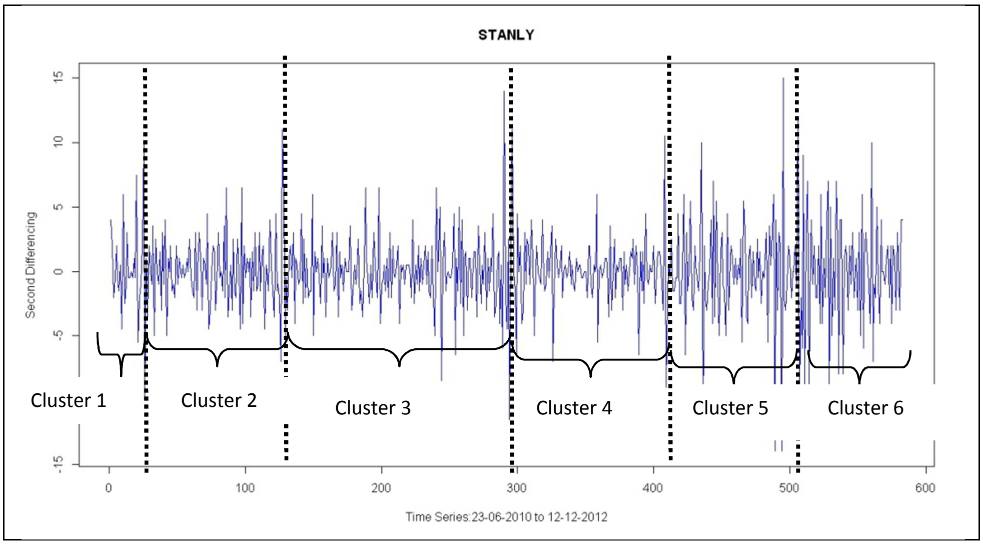


Figure 2.5: Clusters for STANLY in-sample data.

We use the in-sample time series of daily closing prices of DTAC index to demonstrate the border detection. The DTAC in-sample dataset (after we decompose a trend) is provided in Figure 2.6. We apply the same procedure as the one for STANLY index to obtain the 1st cluster of DTAC data series, as in Figure 2.7. Continue the process with the new considered interval series, it turns out that all points are removed after testing by the Shapiro-Wilk test.

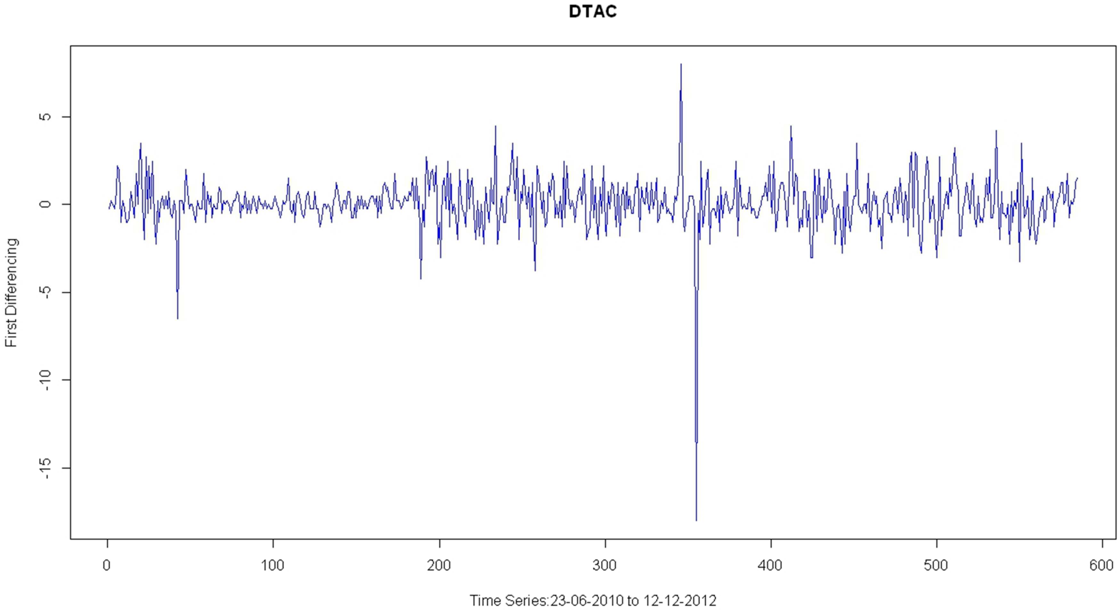


Figure 2.6: The DTAC in-sample time series (after decompose a trend).

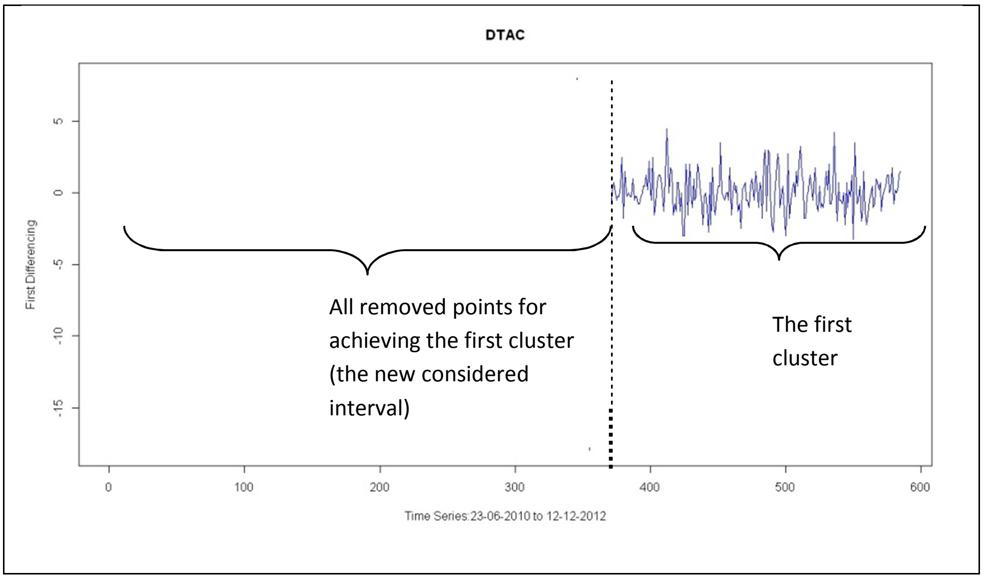


Figure 2.7: The first cluster of DTAC index prices.

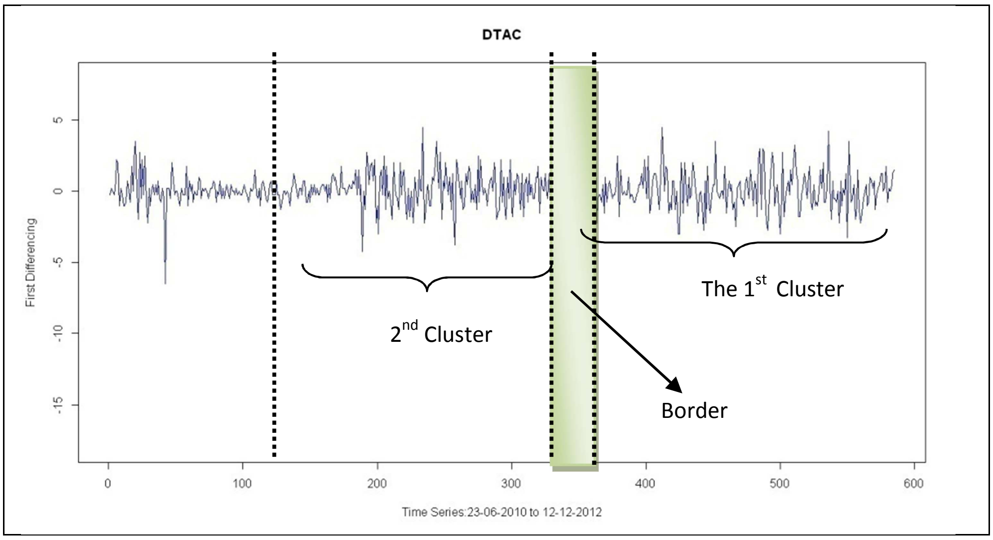


Figure 2.8: The second cluster obtained after detecting the border.

After identifying a border, we redo the process of the reduction strategy again; until we can obtain the next cluster, see Figure 2.8.

We show the example to examine the time series whether the series can be modeled by AR model in the example 2.3.1.

Suppose that $\{y_t\}_{t=d}^{cp}$ and $\{y_t\}_{t=cp+1}^{k}$ are samples of the time series in two consecutive clusters where $1 \le d \le cp \le k$. We examine $\{y_t\}_{t=d}^{cp}$, $\{y_t\}_{t=cp+1}^{k}$ and $\{y_t\}_{t=d}^{k}$ on the nonlinearity by using Tsay [5] method. The concept is based on the ordered AR [27]. If the distribution in $\{y_t\}_{t=d}^{cp}$ and $\{y_t\}_{t=cp+1}^{k}$ can be modeled by AR model but $\{y_t\}_{t=d}^{k}$ cannot be modeled by AR model, then the change-point is indicated at the point, y_{cp}.

Example 2.3.1 Let $\{y_t\}_{t=1}^{13} = \{1, 0.5, 2.5, 7.5, 12.5, 3.5, 1.2, 4.5, 5.6, 6.7, 9.0, 10.0, 11.0\}$ be the time series in a cluster. Suppose that its order of the maximum partial autocorrelation function value is 3; i.e., $p = 3$.

Tsay method starts with reordering the given time series into a non-decreasing order as follows

$$\{0.5, 1, 1.2, 2.5, 3.5, 4.5, 5.6, 6.7, 7.5, 9.0, 10.0, 11.0, 12.5\}.$$

Let $[t]$ be the original time index of the reordered series. Table 2.1 shows the non-decreasing order of $\{1, 0.5, 2.5, 7.5, 12.5, 3.5, 1.2, 4.5, 5.6, 6.7, 9.0, 10.0, 11.0\}$ and its corresponding time index $[t]$.

Table 2.1: Non-decreasing order of the time series of the example 2.3.1 at its original time index $[t]$.

t	y_t	$[t]$	**Non-decreasing order**
1	**1**	**2**	**0.5**
2	**0.5**	**1**	**1**
3	**2.5**	**7**	**1.2**
4	**7.5**	**3**	**2.5**
5	**12.5**	**6**	**3.5**
6	**3.5**	**8**	**4.5**
7	**1.2**	**9**	**5.6**
8	**4.5**	**10**	**6.7**
9	**5.6**	**4**	**7.5**
10	**6.7**	**11**	**9.0**
11	**9.0**	**12**	**10.0**
12	**10.0**	**13**	**11.0**
13	**11.0**	**5**	**12.5**

Next, the matrix of regressors are built with the order $p = 3$, (the details are in the shaded columns of Table 2.2).

Table 2.2: The regressors of the order $p=3$ for the time series of Example 2.3.1.

t	y_t	$[t]$	$y_{[t]}$	**Regressors**		
				$y_{[t]}$	$y_{[t]\text{-}1}$	$y_{[t]\text{-}2}$
3	**2.5**	**7**	**1.2**	**1.2**	**3.5**	**12.5**
4	**7.5**	**3**	**2.5**	**2.5**	**0.5**	**1**
5	**12.5**	**6**	**3.5**	**3.5**	**12.5**	**7.5**
6	**3.5**	**8**	**4.5**	**4.5**	**1.2**	**3.5**
7	**1.2**	**9**	**5.6**	**5.6**	**4.5**	**1.2**
8	**4.5**	**10**	**6.7**	**6.7**	**5.6**	**4.5**
9	**5.6**	**4**	**7.5**	**7.5**	**2.5**	**0.5**
10	**6.7**	**11**	**9.0**	**9.0**	**6.7**	**5.6**
11	**9.0**	**12**	**10.0**	**10.0**	**9.0**	**6.7**
12	**10.0**	**5**	**12.5**	**12.5**	**7.5**	**2.5**

The parameters, ψ_1, ψ_2 and ψ_3, are generated from the ordinary least squares method that is provided by the "lm" function on R programming. The predicted values, $\hat{y}_{[t]+1}$, is calculated by

$\hat{y}_{[t]+1} = \psi_1 y_{[t]} + \psi_2 y_{[t]-1} + \psi_3 y_{[t]-2}$ for all $[t]$.

Then, the prediction errors, $e_{[t]}$, are calculated by $e_{[t]} = y_{[t]} - \hat{y}_{[t]}$ for all $[t]$. The test is performed on the prediction errors. We consider the regression on Equation (2.13).

$$e = Y\Theta + \varepsilon, \tag{2.13}$$

where e is the vector of the prediction errors, $\boldsymbol{Y}$ is the matrix of regressors, Θ is the parameters vector, where $\Theta = [\psi_1, \psi_2, \psi_3]^T$, and ε is the error vector. The null hypothesis is accepted when the all values of Θ are close to zero. We use the F test in R programming for the hypothesis testing. If the null hypothesis is accepted, the prediction errors are independently identically distributed and orthogonality to the regressors, and then a linear model will not handle the time series.

We apply Algorithm 1 to DTAC data. The results on clusters, borders and change-points are presented in figure 2.9.

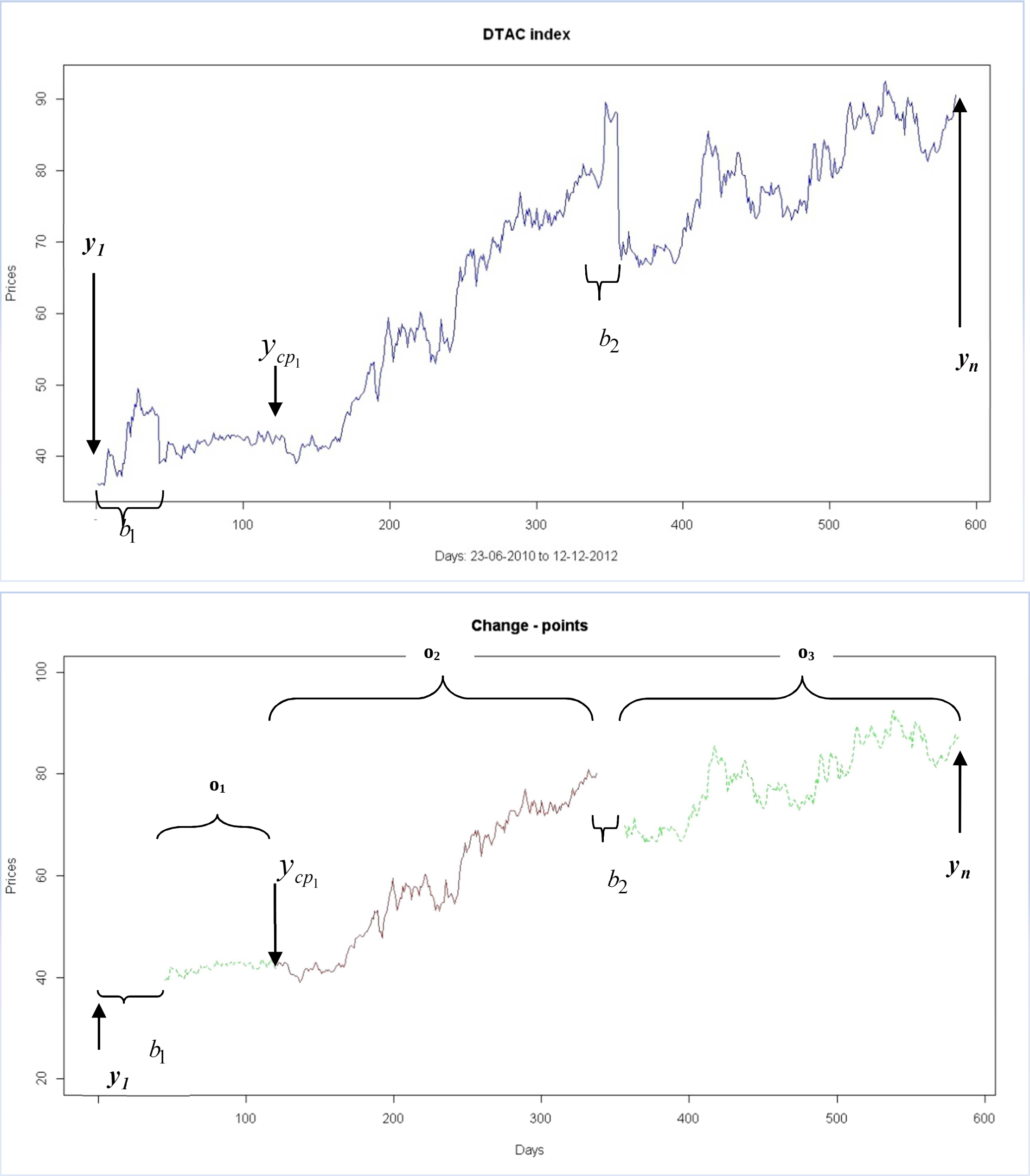


Figure 2.9: Change-points, borders, and clusters for DTAC index prices from 23/06/2010 to 01/11/2012.

From Figure 2.9, the first graph shows that the raw data with the locations of the change-points and the borders. The second graph shows the change-points, borders and clusters of in-sample time series. The voids between lines indicate the borders.

2.4 Prediction Procedure

In the MARMA model, the series from the last cluster that is generated by the method in Section 2.3 is used for the in-sample time series. Then, the MARMA equation is defined as

$$\hat{y}_{n+1} = \psi_1 y_n + \psi_2 y_{n-1} + ... + \psi_p y_{n-p+1}, \tag{2.14}$$

where $\psi_1, \psi_2, ..., \psi_p$ are AR(p) parameters, generated by Yule-Walker Equation. The order $\boldsymbol{p}$ is determined by AIC values. In order to estimate the error part, z_{n+1}, we use the strategy suggested by Elder [28] and the procedure of Zero Lag Exponential Moving Average (ZLEMA) suggested by Chen et al. [29]. Elder suggested that the moving average will be use to smooth three nearest closing prices, and the ZLEMA method is a weighted smoothing method that takes more weight to the current data. The method of ZLEMA is adapted from Exponential Moving Average, that the procedure is shown by Algorithm 2.

Algorithm 2

```
Define:         P :=Numeric Series;  # P[t] is a price at time t
                Period:=NumericSimple;
                t=1; # t is the order in series P
                ZLEMA:=Variable;
                F=2/(Period+1); lag=(Period-1)/2
while(t<=length(P)){
        If(t==1){
                ZLEMA[t] =P[t];
        }else{
                ZLEMA[t]=F*(2*P[t]-P[lag])+(1-F)*ZLEMA[t-1];
        }
t=t+1
}
End.
```

A MARMA model for prediction is defined by the equation,

$$\hat{y}_{n+1} = \psi_1 y_n + ... + \psi_p y_{n-p+1} + \Omega_{n+1}, \tag{2.15}$$

where Ω_{n+1} is the moving average process of the order $q = 3$.

$$\Omega_{n+1} = \beta_0 z_{n+1} + \beta_1 z_n + \beta_2 z_{n-1} + \beta_3 z_{n-2}, \tag{2.16}$$

where β_0, $\beta_1, ..., \beta_3$ are the parameters, that $\beta_0 = 1,$ the values of β_1, β_2 and β_3 are estimated by the Algorithm 2.

CHAPTER III

EXPERIMENTS AND RESULTS OF USING MARMA MODELS

3.1 Dataset

The closing ten index prices of Thai Stock Exchange used as the in-sample and out-of-sample dataset are provided in Table 3.1.

Table 3.1: The names and companies of the ten indices.

Index Name	**company**
ADVANC	ADVANC Info Service Public Company Limited
AOT	Airports of Thailand Public Company Limited
BANPU	BANPU Public Company Limited
CPALL	CP All public company Limited
DTAC	Total Access Communication Public Company Limited
JAS	Jasmine International Public Company Limited
KBANK	Kasikorn Bank Public Company Limited
LOXLEY	LOXLEY Public Company Limited
PTT	PTT Public Company Limited
STANLY	Thai Stanley Eletric Public Company Limited

Table 3.2: The number of the in-sample and out-of-sample dataset of the ten indices.

Names of Indexes	No. of In-sample Dataset	No. of Out-of-sample Dataset
AOT	560	26
BANPU	560	26
DTAC	560	26
ADVANC	562	26
JAS	560	26
STANLY	559	26
CPALL	559	26
KBANK	560	26
PTT	559	26
LOXLEY	560	26

The in-sample dataset of each index is collected between June 23, 2010 and November 1, 2012. The out-of-sample dataset is extracted from November 5, 2012 to December 12, 2012. Table 3.2 lists the exact number of in-sample and out-of-sample data of these specific ten indexes. Note that these ten time series dataset was skipped the dates with no trading activities, for instance, national holidays.

3.2 Tools

This thesis uses R programming version i386 2.15.1, which is an open source, that can be able to download from http://cran.r-project.org. The R programming builds an AR model with the function, ‘ar()’ and generates the residuals by ‘ar()$resid’. The function ‘shapiro.test()’ is used for detecting the residual normality. The stationarity is detected by the function ‘adf.test()’. The prediction error is smooth by the moving average function, ‘ZLEMA()’.

3.3 Procedures and Results

We first provide the clusters of the in-sample time series of ten indices using the Algorithm 1 in Chapter II. We predict the next day of index prices by using the selected in-sample dataset from the last cluster. The code of the MARMA model is shown in Appendix B.

The prediction accuracy of the next day price of ten indices for 26 days is examined by the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Square Error (MSE). We provide the ARIMA, TAR and Generalized ARCH model as the MARMA model's competitors. ARIMA, TAR and GARCH models are generated by using the same in-sample and out-of-sample time series as in the MARMA model. The functions, 'auto.arima()', 'setar()' and 'garchFit()' are used to generate the ARIMA, TAR and GARCH model, respectively. We input the order *m* from *m=1, 2, 3, 4* and *5* of TAR model then select the order that yields the best fit TAR model. The GARCH model is provided in R by using function 'garchFit()' without any external input.

The out-of-sample prediction errors of the MARMA model are compared to the prediction errors of ARIMA, TAR and GARCH models that are measured by using MAE, RMSE, MAPE and MSE, see more details in Table 3.3 (a) – (j).

Table 3.3 : The prediction errors of four models, MARMA, ARIMA, TAR and GARCH in the out-of-sample series using MAE, RMSE, MAPE and MSE in ten indices, (a) AOT, (b) BANPU, (c) DTAC, (d) ADVANC, (e) JAS, (f) STANLY, (g) CPALL, (h) KBANK, (i) PTT and (j) LOXLEY.

(a)

	Measurements	**MARMA**	**ARIMA**	**TAR**	**GARCH**
AOT	**MAE**	0.8556	1.1160	1.0656	1.3997
	RMSE	1.0242	1.4699	1.4387	1.6885
	MAPE	0.9614	1.2558	1.2029	1.5841
	MSE	1.0911	2.2471	2.1502	2.9653

(b)

	Measurements	**MARMA**	**ARIMA**	**TAR**	**GARCH**
BANPU	**MAE**	3.6819	4.1769	4.1376	4.2030
	RMSE	5.6179	6.5844	6.6700	6.6343
	MAPE	0.9703	1.1051	1.0933	1.1104
	MSE	32.8239	45.0890	46.2697	45.774

(c)

	Measurements	**MARMA**	**ARIMA**	**TAR**	**GARCH**
DTAC	**MAE**	0.5283	0.7669	0.7896	0.7738
	RMSE	0.6980	0.9592	0.9604	0.9650
	MAPE	0.6261	0.9019	0.9243	0.9119
	MSE	0.5067	0.9569	0.9593	0.9685

(d)

ADVANC	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	2.3380	3.1128	2.7908	2.9513
	RMSE	2.8557	3.8434	3.8742	3.7760
	MAPE	1.1545	1.5428	1.3860	1.4674
	MSE	8.4813	15.3629	15.6100	14.8287

(e)

JAS	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	0.0608	0.0781	0.0722	0.0819
	RMSE	0.0859	0.1081	0.1043	0.1113
	MAPE	1.2226	1.5578	1.4483	1.6372
	MSE	0.0076	0.0121	0.0113	0.0128

(f)

STANLEY	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	1.0954	1.7877	1.5177	1.6682
	RMSE	1.2509	2.4042	2.2647	2.3685
	MAPE	0.5160	0.8376	0.7122	0.7824
	MSE	1.6273	6.0114	5.3342	5.8345

(g)

CPALL	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	0.4087	0.4774	0.4503	0.4709
	RMSE	0.4922	0.5888	0.5630	0.5849
	MAPE	1.0148	1.1843	1.1177	1.1667
	MSE	0.2519	0.3606	0.3296	0.3558

(h)

KBANK	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	1.6089	1.7813	1.7951	1.8662
	RMSE	2.1336	2.4807	2.4094	2.4232
	MAPE	0.8715	0.9632	0.9721	1.0104
	MSE	4.7344	6.4005	6.0378	6.1070

(i)

PTT	Measurements	MARMA	ARIMA	TAR	GARCH
	MAE	1.3045	1.8469	1.7392	1.8717
	RMSE	1.7218	2.2465	2.2403	2.2695
	MAPE	0.4089	0.5783	0.5461	0.5867
	MSE	3.0833	5.2486	5.2197	5.3568

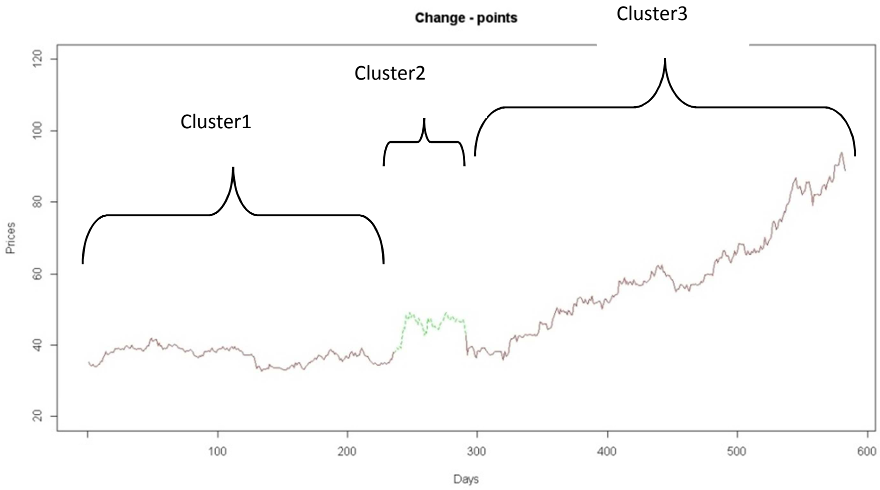
(j)

	Measurements	MARMA	ARIMA	TAR	GARCH
LOXLEY	**MAE**	0.0956	0.1124	0.1135	0.1152
	RMSE	0.1152	0.1472	0.1513	0.1468
	MAPE	1.8853	2.2049	2.2166	2.2546
	MSE	0.0138	0.0225	0.0238	0.0224

Next, we provide the figures of the cluster regions of each index in the time series plots. The bars in the time series plots represent the borders.

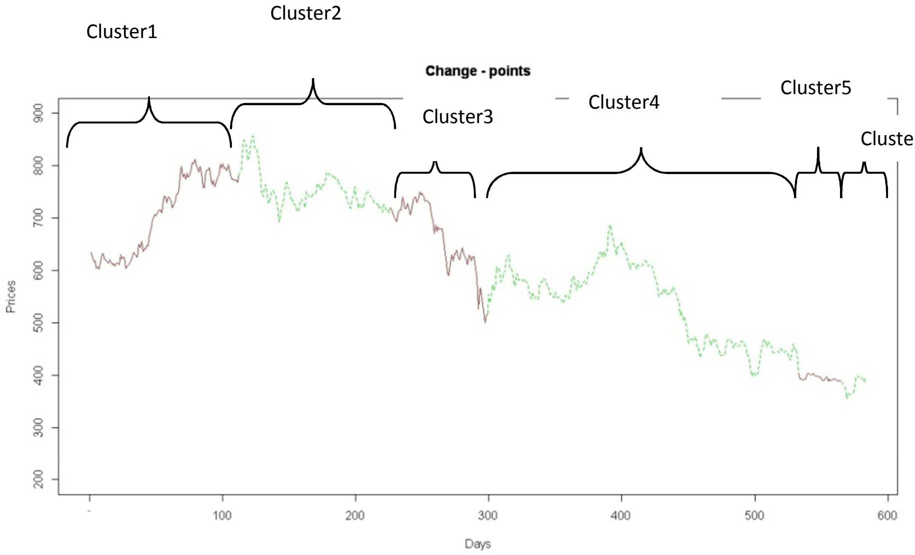
(a) Index Name: AOT

Number of Clusters: 3; 1st cluster=$\{y_1,\ldots,y_{236}\}$**, 2nd cluster=**$\{y_{237},\ldots,y_{291}\}$**, 3rd cluster =** $\{y_{292},\ldots,y_{586}\}$



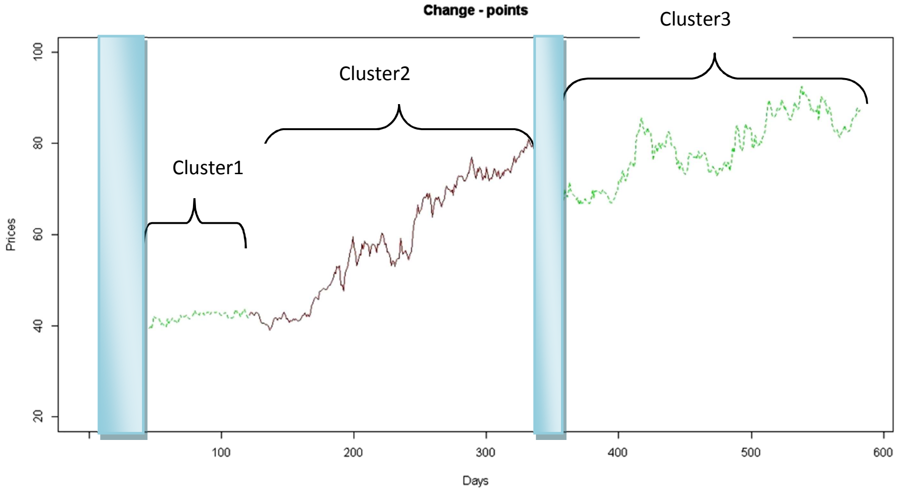
(b) Index Name: BANPU

Number of Clusters: 6; 1st cluster=$\{y_1,\ldots,y_{112}\}$**, 2nd cluster=**$\{y_{113},\ldots,y_{226}\}$**, 3rd cluster =** $\{y_{227},\ldots,y_{299}\}$**, 4st cluster =** $\{y_{300},\ldots,y_{533}\}$**, 5st cluster =** $\{y_{534},\ldots,y_{565}\}$**, 6st cluster =** $\{y_{566},\ldots,y_{586}\}$



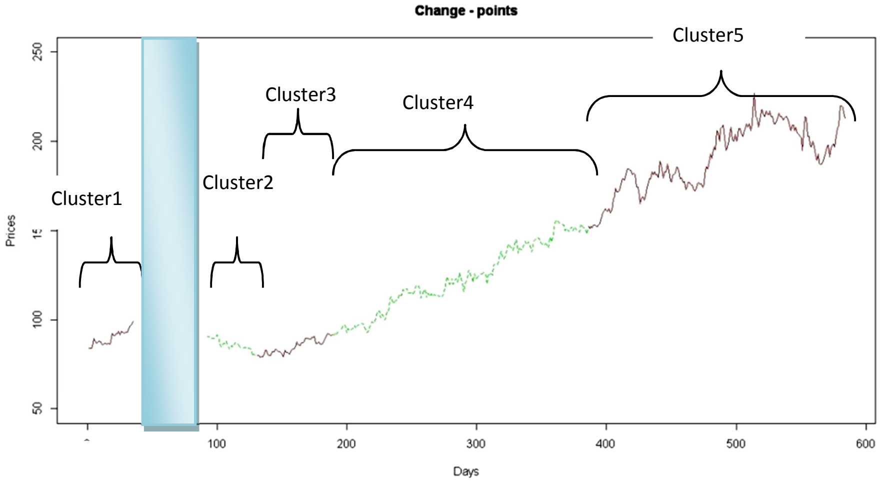
(c) Index Name: DTAC

Number of Clusters: 3; 1st cluster=$\{y_{45},...,y_{121}\}$**, 2nd cluster=**$\{y_{122},...,y_{337}\}$**, 3rd cluster =** $\{y_{356},...,y_{586}\}$



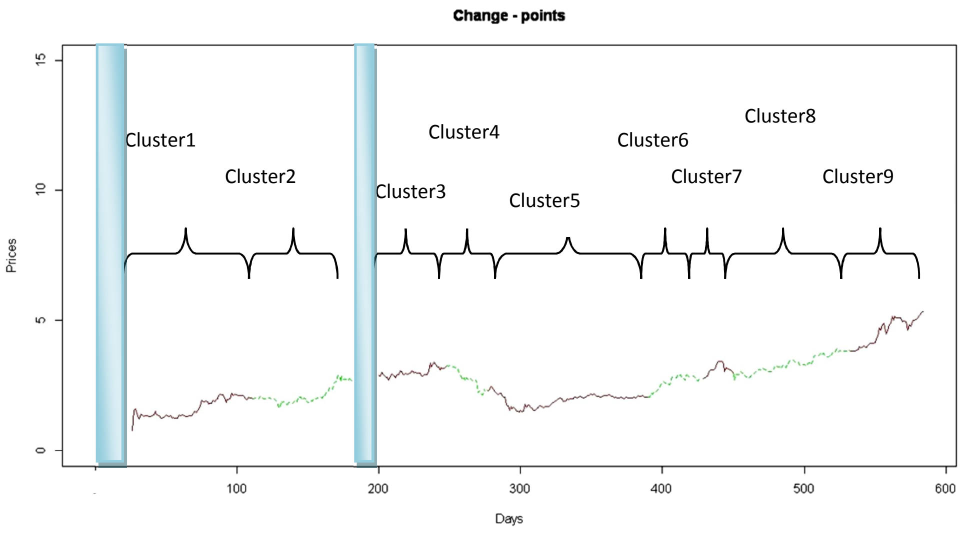
(d) Index Name: ADVANC

Number of Clusters: 5; 1st cluster=$\{y_{1},...,y_{35}\}$**, 2nd cluster=**$\{y_{92},...,y_{131}\}$**, 3rd cluster =** $\{y_{132},...,y_{189}\}$**, 4st cluster =** $\{y_{190},...,y_{386}\}$**, 5st cluster =** $\{y_{387},...,y_{588}\}$



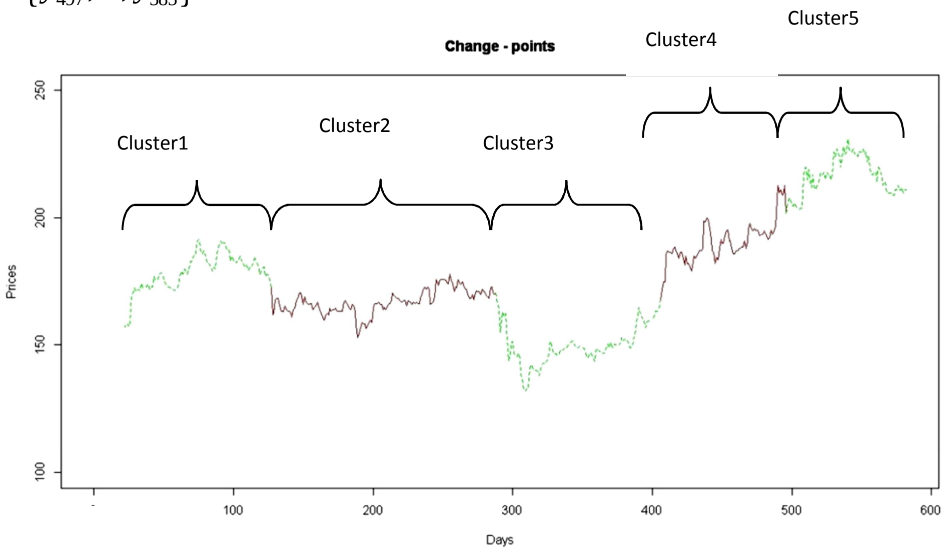
(e) Index Name: JAS

Number of Clusters: 9; 1st cluster=$\{y_{26},\ldots,y_{111}\}$**, 2nd cluster=**$\{y_{112},\ldots,y_{181}\}$**, 3rd cluster =** $\{y_{200},\ldots,y_{248}\}$**, 4st cluster =** $\{y_{249},\ldots,y_{277}\}$**, 5st cluster =** $\{y_{278},\ldots,y_{390}\}$**, 6st cluster =** $\{y_{391},\ldots,y_{429}\}$**, 7st cluster =**$\{y_{230},\ldots,y_{250}\}$**, 8st cluster =** $\{y_{350},\ldots,y_{533}\}$**, 9st cluster =**$\{y_{534},\ldots,y_{586}\}$



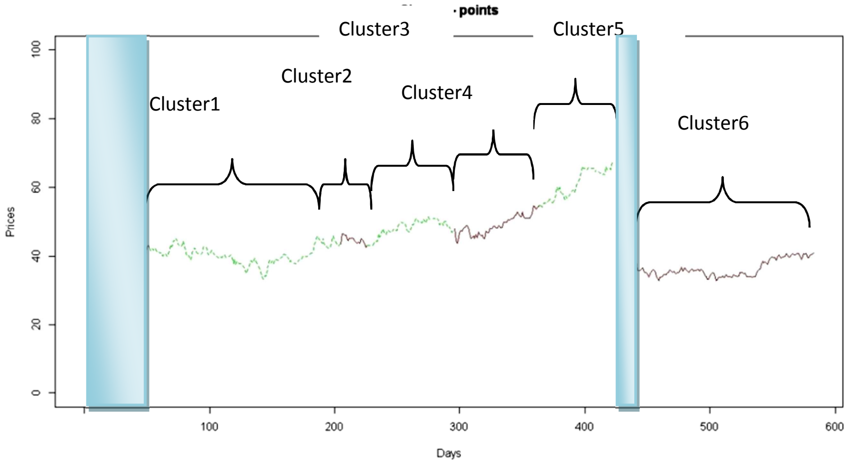
(f) Index Name: STANLY

Number of Clusters: 6; 1st cluster=$\{y_{1},\ldots,y_{22}\}$**, 2nd cluster=**$\{y_{23},\ldots,y_{127}\}$**, 3rd cluster =** $\{y_{128},\ldots,y_{288}\}$**, 4st cluster =** $\{y_{289},\ldots,y_{406}\}$**, 5st cluster =** $\{y_{407},\ldots,y_{496}\}$**, 6st cluster =** $\{y_{497},\ldots,y_{585}\}$



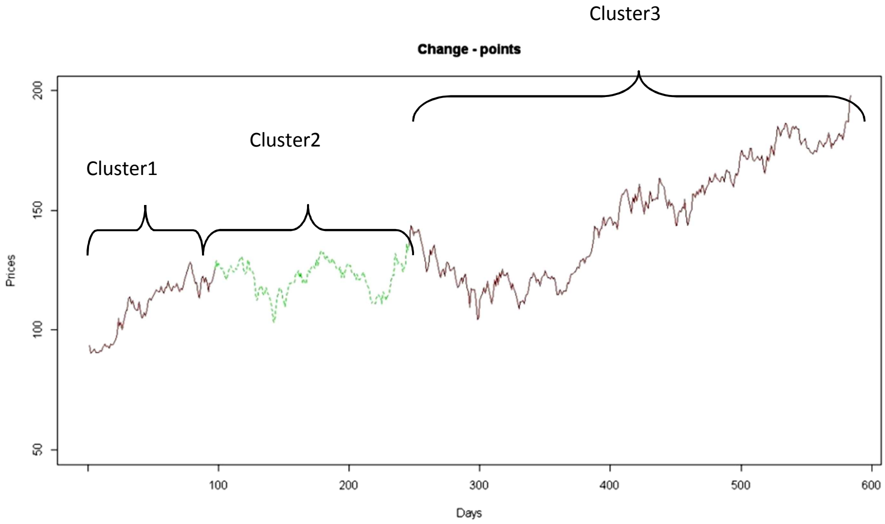
(g) Index Name: CPALL

Number of Clusters: 6; 1st cluster=$\{y_{30},\dots,y_{52}\}$, **2nd cluster=**$\{y_{53},\dots,y_{205}\}$, **3rd cluster =** $\{y_{206},\dots,y_{226}\}$, **4st cluster =** $\{y_{227},\dots,y_{296}\}$, **5st cluster =** $\{y_{297},\dots,y_{364}\}$, **6st cluster =** $\{y_{441},\dots,y_{585}\}$



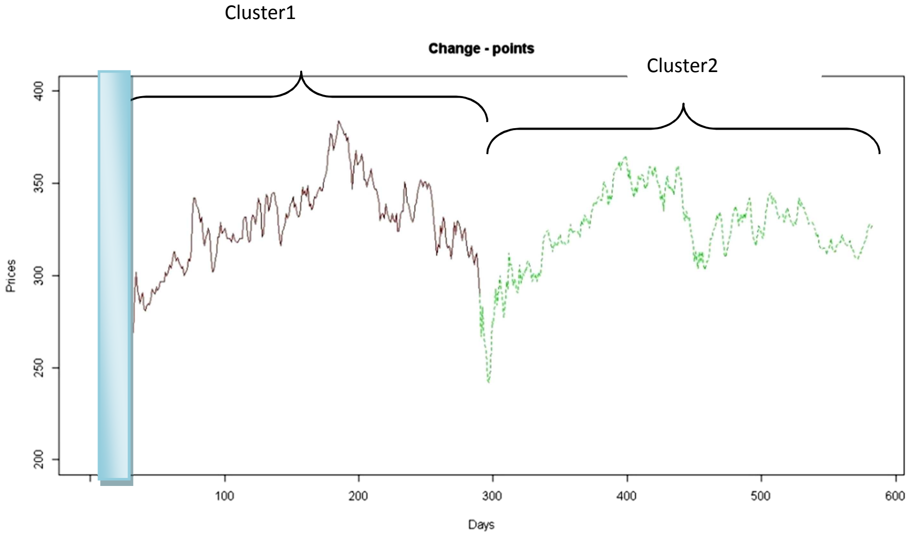
(h) Index Name: KBANK

Number of Clusters: 3; 1st cluster=$\{y_{1},\dots,y_{97}\}$, **2nd cluster=**$\{y_{98},\dots,y_{246}\}$, **3rd cluster =** $\{y_{247},\dots,y_{586}\}$



(i) Index Name: PTT

Number of Clusters: 2; 1st cluster=$\{y_{32},\dots,y_{290}\}$, 2nd cluster=$\{y_{291},\dots,y_{585}\}$



(j) Index Name: LOXLEY

Number of Clusters: 7; 1st cluster=$\{y_{85},\dots,y_{144}\}$, 2nd cluster=$\{y_{145},\dots,y_{189}\}$, 3rd cluster = $\{y_{190},\dots,y_{307}\}$, 4st cluster = $\{y_{308},\dots,y_{396}\}$, 5st cluster = $\{y_{397},\dots,y_{444}\}$, 6st cluster = $\{y_{445},\dots,y_{515}\}$, 7st cluster = $\{y_{516},\dots,y_{586}\}$

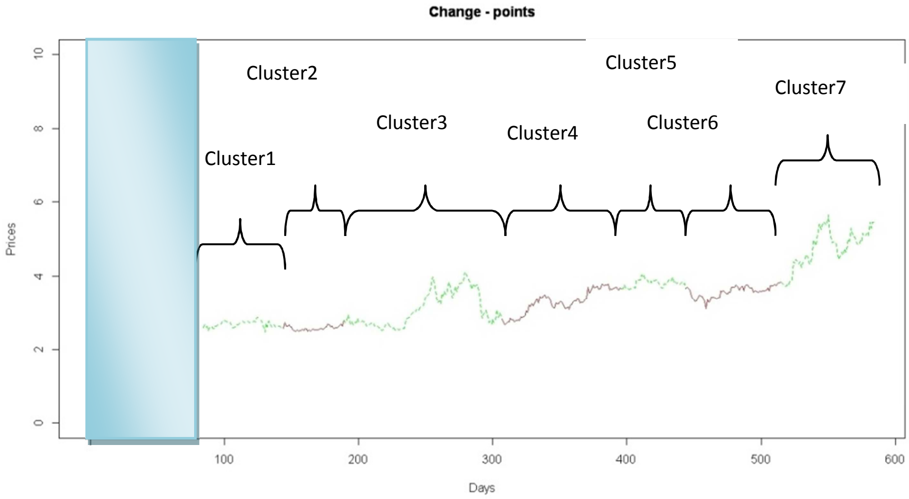
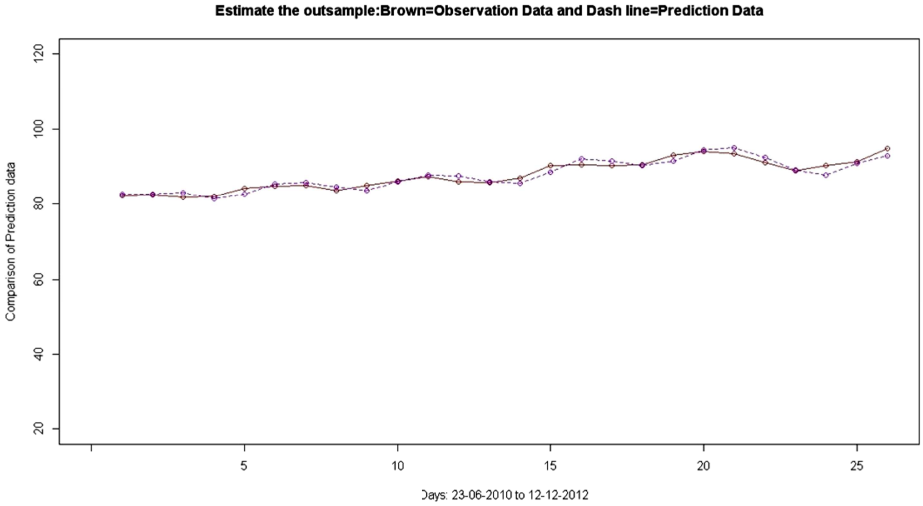
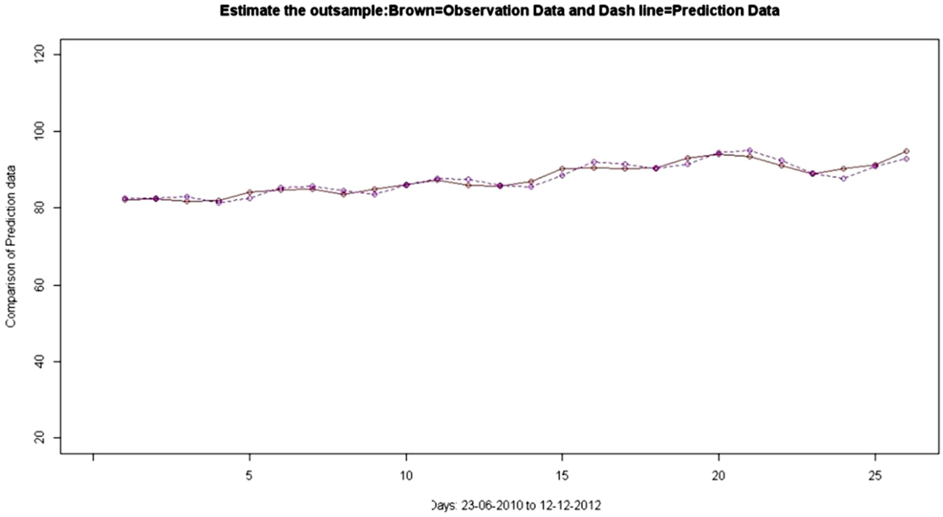


Figure 3.1: The cluster region plots of the specific ten indices (a)-(j).

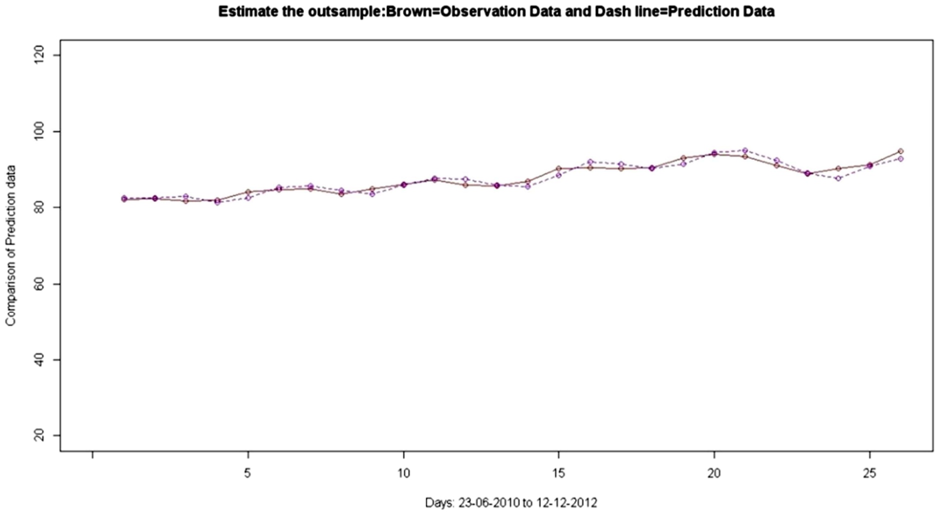
(a) **Index Name: AOT**



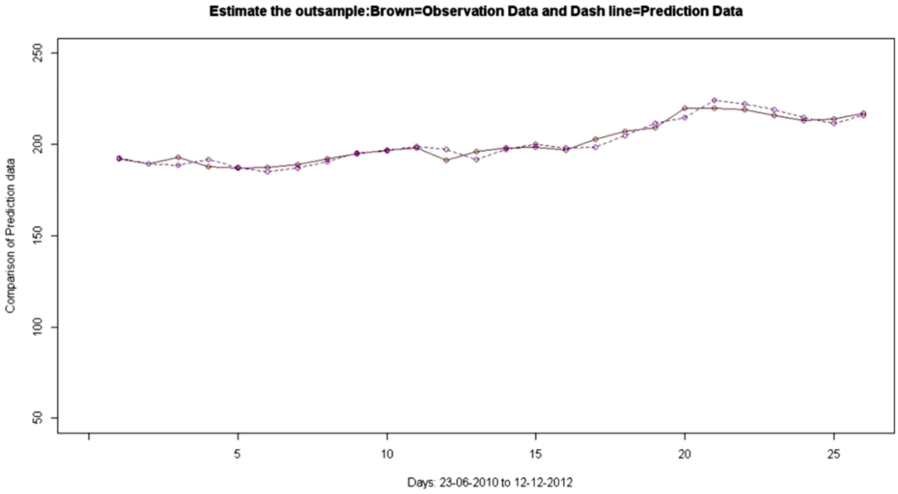
(b) **Index Names: BANPU**



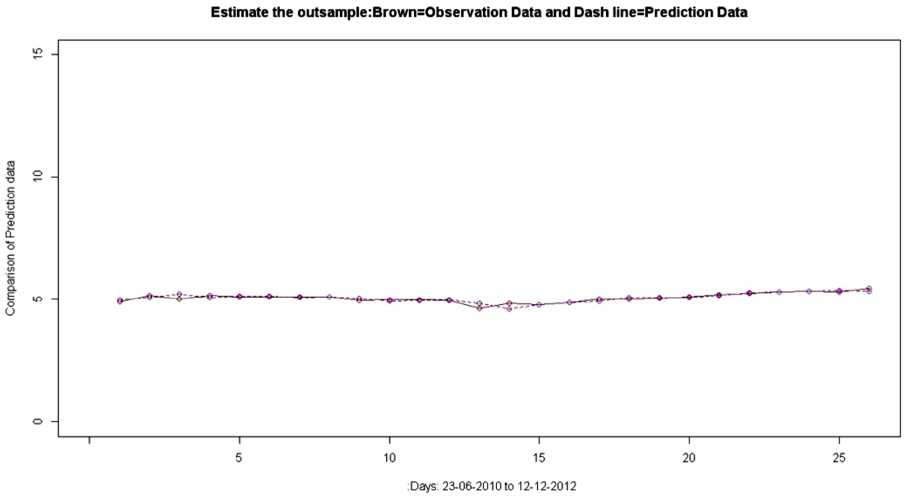
(c) Index Name: DTAC



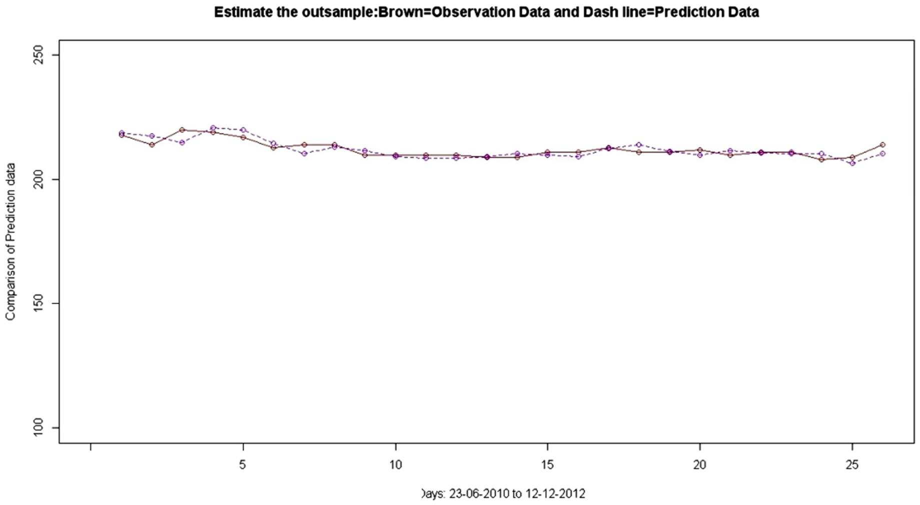
(d) Index Name: ADVANC



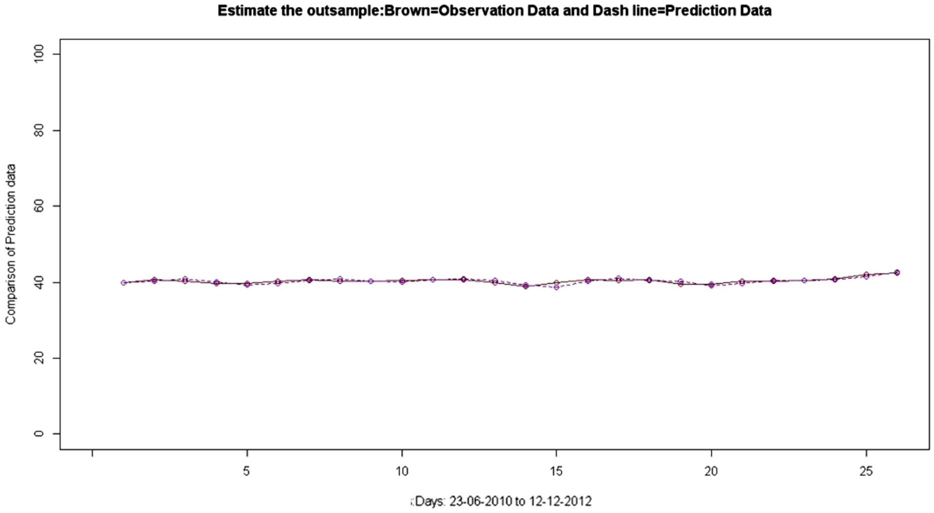
(e) Index Name: JAS



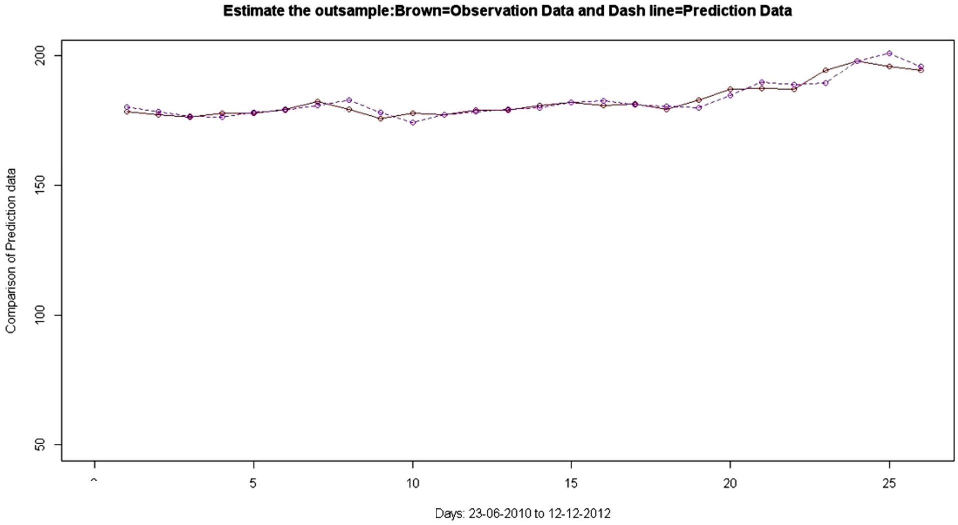
(f) Index Name: STANLY



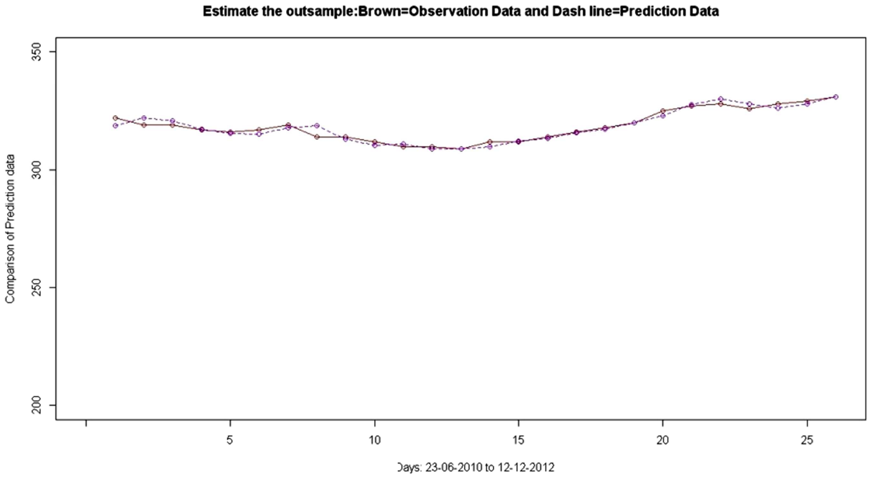
(g) Index Name: CPALL



(h) Index Name: KBANK



(i) Index Name: PTT



(j) Index Name: LOXLEY

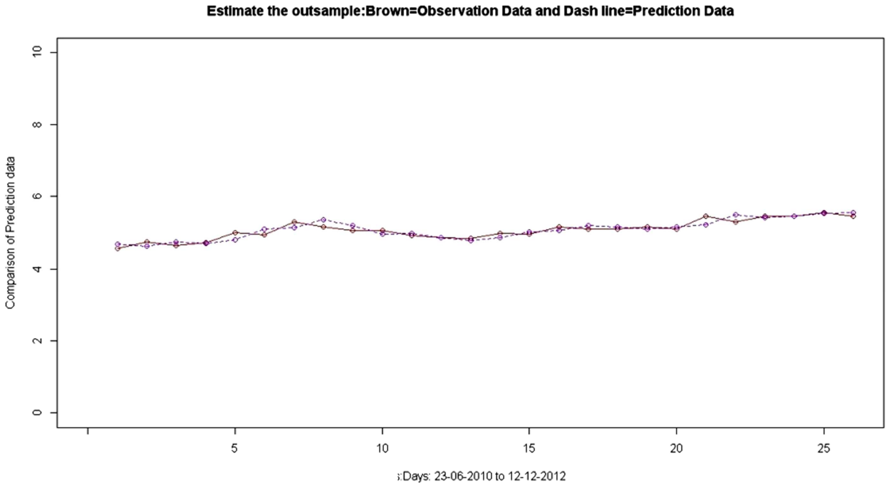


Figure 3.2: The predicted values of the specific ten indexes, (a) to (j), by the MARMA model, ---: predicted time series and — : observed time series.

CHAPTER IV

CONCLUSION

A Multiple change-point AutoRegressive Moving Average model is a prediction model based on a linear AutoRegressive model but the MARMA procedure is more flexible than an AR model. The AR process requires a normality of residuals throughout an in-sample time series. However, a MARMA model uses a part of whole in-sample series, that the residual has a normal distribution to predict the out-of-sample values. Moreover, in comparison with other models, a MARMA model is more realistic in a various kind of time series. For instance, the GARCH procedure has a misleading estimation when a time series is affected by an unknown short period factor which is required the use of the exogenous variable while the MARMA procedure does not require any external variable.

The comparison of the MARMA, ARIMA, TAR and GARCH models in Chapter III shows that the MARMA model consistently outperformed the other competitor models in the prediction of the specific ten indexes. In addition, we investigate to use the MARMA model in the prediction of a periodic time series. The sunspot data has a seasonal period in every 11 years. We predict the sunspot data during the years 1920 to 2008. We obtain that the prediction accuracy measurement showed a smaller Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values than the TAR model that was presented by Bermejo, Pena and Scanchez [6], more detail in Appendix A.

However, the MARMA model is unable to use when the out-of-sample series is not successive to the in-sample series. Hence, if the border is indicated at the previous point of the out-of-sample dataset in time series, the MARMA model will not be applicable.

REFERENCES

[1] Javier, C.; Espinola, R.; Nogales, F.J.; and Conejo, A.J. ARIMA models to predict next day electricity prices. **IEEE Transaction on Power Systems** 3 (2003): 1014-1020.

[2] Nochai, R.; and Nochai T. ARIMA model for forecasting oil palm price. **2nd Indonesia–Malaysia – Thailand Growth Triangle (IMT-GT) Regional Conference on Mathematics, Statistics, and Applications** 3 (2006): 16-21.

[3] Williams, C. **Lecture Note: Probabilistic Modeling and Reasoning** [Online]. Available from: www. inf.ed.ac.uk/teaching/courses/pmr/docs/arma.pdf [2012, June 15].

[4] Tong, H. **Threshold Models in Nonlinear Time Series Analysis**. Springer-Verlag, 1989.

[5] Tsay, R.S. Testing and modeling threshold autoregressive processes. **Journal of the American Statistical Association** 84 (1989): 231-240.

[6] Bermejo, M.; Pena, D.; and Scanchez, I. Identification of TAR models using recursive estimation. **Journal of Forecasting** 30 (2011): 31-50.

[7] Dijk, D.V.; Franses, P.H.; Clements, M.P.; and Smith, J. On SETAR non-linearity and forecasting. **Journal of Forecasting** 22 (2003): 359-375.

[8] Narayan, P.K. The behavior of US stock prices: evidence from a threshold autoregressive model. **Journal of Mathematics and Computers in Simulation** 71 (2006): 103-108.

[9] Gibson, D. Threshold autoregressive models in finance: a comparison approach. **Proceeding of the fourth Annual ASEARC Conference** (2011): 72-75.

[10] Pai, P.F.; and Lin, C.F. A hybrid ARIMA and support vector machines models in stock price forecasting**. Omega the International Journal of Management Science** 33 (2005): 497-505.

[11] Merh, N.; Saxena, V.P.; and Pardasani, K.R. A comparison between hybrid approaches of ANN and ARIMA for Indian stock trend forecasting**. Business Intelligence Journal** 3 (2010): 23-43.

[12] Areekul, P.; Senjyu, T.; Toyama, H.; and Yona, A. A hybrid ARIMA and neural network model for short-term price forecasting in deregulated market**. IEEE Transactions on Power Systems** 25 (2010): 524-530.

[13] Engle, R.F. Autoregressive conditional heteroskedasticity with estimates of variance of United Kingdom inflation. **Econometrica** 5 (1982): 987-1008.

[14] Mahajan, S.; and Singh, B. Return-volume dynamics in India Stock Market. **SAGE** 5 (2009): 63-70.

[15] Liu, H.C.; Lee, Y.H.; and Lee, M.H. Forecasting China Stock Markets volatility via GARCH models under Skewed-GED distribution. **Journal of Money Investment and Banking Euro** 7 (2009).

[16] Hussan, Md.R.;Nath, B.; and Kirley, M. A fusion model of HMM, ANN and GA for stock market forecasting. **Expert Systems with Applications** 33 (2007): 171-180.

[17] Yeh, C.Y.; Huang, C.W.; and Lee, S.J. A multiple-kernel support vector regression approach for stock market price forecasting**. Expert Systems with Applicatio**ns 38 (2011): 2177-2186.

[18] Lo, A.; Mackinlay, A.C. **A non-random walk down Wall Street**. Princeton University Press, 1989.

[19] Chatfield, C. **The analysis of time series an introduction**. New York: Chapman & Hall, 1992.

[20] Box, G.E.P.; and Jenkins, G.M. **Time series analysis, forecasting and control**. Holden-Day, 1970.

[21] Mcculloch, A. **Looking at time-series using waves**[Online]. Available from: www.significancemagazine.org [2013, January 10].

[22] Dickey, D.A.; and Fuller, W.A. Distribution of the estimators for autoregressive time series with a unit root. **Journal of the AMERICAN Statistical Association** 74 (1979): 427-431.

[23] Shapiro, S.S.; and Wilk, M.B. An analysis of variance test for normality (complete samples). **Biometrika** 52 (1965): 591-611.

[24] Sarhan, A.E.; and Greenberg, B.G. Estimation of location and scale parameters by order statistics from singly and doubly censored samples. **Ann. Math. Statist.** 27 (1956): 427-45.

[25] Royston, P. An extension of Shapiro and Wilk's test for normality to large samples. **Applied Statistics** 31 (1982): 115-124.

[26] Royston, P. A remark algorithm AS181. **Applied Statistics** 44 (1995): 547-551.

[27] Petruccelli, J.; and Davies, N. A Portmanteau test for Self-Exciting Threshold Autoregressive-Type nonlinearity in time series. **Biometrika** 73 (1986): 687-694.

[28] Elder, A. **Come into My Trading Room A Complete Guide to Trading**. New York: Wiley, 2002.

[29] Chen, D.H.; Lo, P.; and Swan, W.P. **Zero-lag exponential moving average for real-time control and noisy data processing** [Online]. Available from: www.hydrocarbonprocessing.com [2010, October 15].

APPENDICES

APPENDIX A: SUNSPOT DATA PREDICTION

We use the MARMA model to predict the annual sunspot data in the year 1920-2008 by using the in-sample series in the year 1700-1919, and then we compare the results with Tong, Tsay and Miguel [2] that they predicted the sunspot data by using the same time series data, see more details in Table 5.1.

Measurements	**MARMA**	**Tong (1983)**		**Tsay (1989)**		**Miguel (2011)**	
		[1]**_h_=1**	**_h_=2**	**_h_=1**	**_h_=2**	**_h_=1**	**_h_=2**
MAE	09.25	12.31	12.42	11.74	11.84	11.37	11.47
RMSE	12.16	16.59	16.69	15.52	15.61	15.33	15.42

Table 5.1: The prediction errors of the MARMA model and the TAR models that are proposed by Tong, Tsay and Miguel, respectively.

In Table 5.1, ***h*** stands for the number of horizontal lines for the regime regions in TAR model.

APPENDIX B: MARMA MODEL CODE

```
#-----------------------------------Functions and Variables-------------------------------

#          Code: MARMA model for ADVANC index prices
#          insample Year 2010 to 2012
library("nlme")
library("Rcmdr")
library("lmtest")
library("tseries")
require(graphics)
library("nortest")
library("e1071")
library("rpart")
library("Hmisc")
library("fBasics")
library("TTR")
library("fUnitRoots")
library("tsDyn")
library("FinTS")
data<-seq()
data<-adv[,-1]
data<-rev(data)
data<-data[-c(1:2)]
plot.ts(data,col="blue",main="ADVANCE index",type="l",xlab="Days: 23-06-2010 to
12-12-2012",ylab="Prices")
windows()
tdat<-diff(data)
realdat<-data
plot(tdat,col="blue",main="ADVANCE",type="l",xlab="Time Series:23-06-2010 to 12-
12-2012",ylab="First Differencing")
sam<-559
outsam<-26
#--------------------------------------------------
keep.s<-seq()
run=1
pre.end<-seq()
p.va<-seq()
res.end<-seq()
mod.end<-seq()
cor.end<-seq()
cp.all<-list()
pre.all<-list()
res.all<-list()
best.mod<-seq()
cor.all<-list()
min.res<-seq()
cor<-list()
marma.lag<-seq()
s=3
while(run<=outsam){
datcheck<-tdat[1:(sam+run-1)]
aa=1
bb=length(datcheck)
u=1
cpf<-seq()
cpb<-seq()
        while(length(datcheck)>=20){
                dat<-seq()
                dat<-ar(datcheck)$resid[-c(1:ar(datcheck)$order)]
```

```
#dat<-datcheck
aa<-1
g=1
kptrend<-kpss.test(datcheck,null="Trend")
kplevel<-kpss.test(datcheck,null="Level")
adfuller<-adf.test(datcheck)
ander<-ad.test(datcheck)
shap<-shapiro.test(dat)
    test.st<-seq()
if(adfuller$p.value[1]<0.05){
    test.st[g]<-1
    g=g+1
}
if(shap[[2]]>0.05){
    test.st[g]<-1
    g=g+1
}
        if(sum(test.st)!=2){
                ao<-aa
            while(length(ao:bb)>20){
                aa=aa+1
                ao<-aa
                datcheck<-seq()
                datcheck<-tdat[aa:bb]
dat<-ar(datcheck)$resid[-c(1:ar(datcheck)$order)]
                #dat<-seq()
        #dat<-datcheck
            g=1
    kptrend<-kpss.test(datcheck,null="Trend")
    kplevel<-kpss.test(datcheck,null="Level")
```

```
            adfuller<-adf.test(datcheck)
                ander<-ad.test(datcheck)
                        shap<-shapiro.test(dat)
                            test.st<-seq()
                        if(adfuller$p.value[1]<0.05){
                                test.st[g]<-1
                                g=g+1
                                }
                            if(shap[[2]]>0.05){
                                test.st[g]<-1
                                g=g+1
                            if(sum(test.st)==2){

cpf[u]<-aa
cpb[u]<-bb
u=u+1
                            ao<-bb
                                }
                        }
                        if(length(aa:bb)<=20){
                                aa<-bb-1
                        }
                    }else{
                        cpf[u]<-aa
                        cpb[u]<-bb
                        u=u+1
                    }
        bb<-aa
        datcheck<-tdat[1:bb]
    }
```

```
cpf<-rev(cpf)
cpb<-rev(cpb)
#kd<-seq()
#bb<-seq()
#m=1
#for(i in 1:length(cpf)){
#        kd[i]<-cpb[i]-cpf[i]+1
#                if(kd[i]<10){
#                        bb[m]<-i
#                        m=m+1
#                }
#}
#cpf<-cpf[-c(bb)]
#cpb<-cpb[-c(bb)]
cp.all[[run]]<-c(cpf,cpb)
#if(((220+run-1)-cpb[length(cpb)])>30){
#cpf[(length(cpf)+1)]<-cpf[length(cpf)]+1
#cpb[(length(cpb)+1)]<-(220+run-1-s)
#}
keep.s[run]<-s
#---------------------------------T R A I N I N G..... S E T-----------------------
datcheck<-tdat[1:(sam+run-s)]
pred.train<-list()
res.train<-list()
data.train<-list()
datalast<-seq()
datalast<-tdat[(cpf[(length(cpf))]):(sam+run-1)]
#-----------------------------------------------------Moving Average ----------------
res.ma<-seq()
pre.ma<-seq()
```

```
parameter<-list()
m<-ar(datalast)
marma.lag[run]<-m$order
for(i in 1:s){
        l<-m$order
        parameter[[i]]<-rev(m$ar)
        pre.ma[i]<-realdat[(sam+run-s+i)]+sum((parameter[[1]])*(tdat[(sam+run-s+i-
1-l+1):(sam+run-s+i-1)]))
        res.ma[i]<-pre.ma[i]-realdat[(sam+run-s+i+1)]
}
par.end<-seq()
moddy<-ar(datalast)
ord<-moddy$order
par.end<-rev(moddy$ar)
#pre.end[run]<-(realdat[(sam+run)]+sum((par.end)*(tdat[(sam+run-1-
(ord)+1):(sam+run-1)])))-mean(res.ma)
pre.end[run]<-(realdat[(sam+run)]+sum((par.end)*(tdat[(sam+run-1-
(ord)+1):(sam+run-1)])))-ZLEMA(res.ma,n=3)[length(res.ma)]
res.end[run]<-pre.end[run]-realdat[(sam+run+1)]
#-----------------------------------------------------compare with all model
plot.ts(realdat[(sam+2):(sam+run+1)],type="o",col="brown",main="Estimate the
outsample:Brown=Observation Data and Dash line=Prediction
Data",ylim=c(50,250),xlim=c(0,run),xlab="Days:Days: 23-06-2010 to 12-12-
2012",ylab="Comparison of Prediction data")
par(new=TRUE)
plot.ts(pre.end[1:run],col="purple",lty=2,type="o",main="Estimate the
outsample:Brown=Observation Data and Dash line=Prediction
Data",ylim=c(50,250),xlim=c(0,run),xlab="Days:Days: 23-06-2010 to 12-12-
2012",ylab="Comparison of Prediction data")
run=run+1
```

```
}
#-----------------------------------------------------
sum(abs(res.end))/length(res.end)
#root mean square error
print("RMSE=")
sqrt((1/outsam)*sum(res.end^2))
#MAPE
(100/outsam)*(sum(abs(res.end)/abs(realdat[(sam+2):length(realdat)])))
#MSE
(sum(res.end^2))/(outsam-1)
windows()
#############plot change.point
inter.cp<-cp.all[[length(cp.all)]]
en<-length(inter.cp)/2
a<-inter.cp[1:en]
b<-inter.cp[(en+1):length(inter.cp)]
mab<-rbind(a,b)
mab
i=1
          while(i<=en){
                    plot(c(a[i]:b[i]),realdat[a[i]:b[i]],type="l",col="brown",main="Change
- points",ylim=c(50,250),xlim=c(0,b[length(b)]),xlab="Days",ylab="Prices")
                    par(new=TRUE)
                              if(i!=en){
          plot(c(a[(i+1)]:b[(i+1)]),realdat[a[(i+1)]:b[(i+1)]],lty=2,type="l",col="green",
main="Change -
points",ylim=c(50,250),xlim=c(0,b[length(b)]),xlab="Days",ylab="Prices")
                              }
                    par(new=TRUE)
                    i=i+2
```

```
        }
matrix(marma.lag)
#### Running Time
ptm <- proc.time()
for (i in 1:50) mad(stats::runif(500))
proc.time() - ptm
```

End.

BIOGRAPHY

Name:	Pimsiri Ponsap
Date of Birth:	December 2, 1980
Place of Birth:	Chumphon Province, Thailand
Education:	Chulalongkorn University