ตัวแบบอนุกรมเวลารวมตามลำดับย่อยของอริมาพหุคูณเพื่อพยากรณ์จำนวนเงินในตู้กดเงินอัตโนมัติ

นายไพสิฐ ขันอาสา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

MULTIPLE ARIMA SUBSEQUENCES AGGREGATE TIME SERIES MODEL TO FORECAST CASH IN ATM

Mr.Paisit Khanarsa

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Applied Mathematics and Computational Sciences
Department of Mathematics and Computer Sciences
Faculty of Science
Chulalongkorn University
Academic Year 2016

Thesis Title	MULTIPLE ARIMA SUBSEQUENCES AGGREGATE TIME SERIES MODEL TO FORECAST CASH IN ATM
By	Mr. Paisit Khanarsa
Field of Study	Applied Mathematics and Computational Sciences
Thesis Advisor	Assistant Professor Krung Sinapiromsaran, 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
(Associate Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

. . . . Chairman
(Assistant Professor Khamron Mekchay, Ph.D.)

. . . . Thesis Advisor
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

. . . . Examiner
(Jiraphan Suntornchost, Ph.D.)

. . . . External Examiner
(Thidaporn Supapakorn, Ph.D.)

ไพสิฐ ขันอาสา : ตัวแบบอนุกรมเวลารวมตามลำดับย่อยของอริมาพหุคูณเพื่อพยากรณ์จำนวนเงินในตู้กดเงินอัตโนมัติ. (MULTIPLE ARIMA SUBSEQUENCES AGGREGATE TIME SERIES MODEL TO FORECAST CASH IN ATM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.กรุง สินอภิรมย์สราญ, 84 หน้า.

ปัจจุบันตู้เอทีเอ็มถูกนำมาใช้เพื่อความสะดวกในการให้บริการลูกค้าในการธนาคารทางการเงิน สำหรับการให้บริการต่างๆ เช่นการโอน, ฝาก และถอน สิ่งที่น่าเป็นห่วงของตู้เอทีเอ็มคือการขาดแคลนของธนบัตรสำหรับการถอนเงิน ดังนั้นหนึ่งในปัญหาที่น่าสนใจของตู้เอทีเอ็มคือการคาดการณ์ความต้องการใช้เงินสดในตู้เอทีเอ็มในแต่ละตู้ ในลักษณะข้อมูลเป็นอนุกรมเวลาโดยปกติพฤติกรรมของการใช้เงินสดในตู้เอทีเอ็มจะแสดงให้เห็นถึงการแกว่ง และมักมีรูปแบบที่ซ่อนอยู่ เนื่องมาจากพฤติกรรมที่หลากหลายของลูกค้า ดังนั้นงานวิจัยนี้จึงได้นำเสนอตัวแบบการพยากรณ์ใหม่สำหรับการทำนายเงินสดในตู้เอทีเอ็ม โดยตัวแบบที่นำเสนอเรียกว่า ตัวแบบอนุกรมเวลาตามลำดับย่อยของอริมาพหุคูณ หรือตัวแบบมาซา ในกระบวนการของขั้นตอนวิธี ข้อมูลเอทีเอ็มจะถูกแบ่งออกเป็นข้อมูลสำหรับการสร้างตัวแบบมาซาและข้อมูลสำหรับการประเมินตัวแบบมาซา ในการสร้างตัวแบบมาซา เทคนิคการรวมจะถูกนำมาประยุกต์ใช้ตามลักษณะของรอบของข้อมูลเพื่อที่จะลดการแกว่งของอนุกรมเวลา จากนั้นข้อมูลจะถูกแบ่งออกเป็นข้อมูลสำหรับสร้างตัวแบบ และข้อมูลสำหรับตรวจสอบตัวแบบ จากนั้นลำดับย่อยต่างๆในอนุกรมเวลาจะถูกสร้างขึ้น และแต่ละลำดับย่อยนั้นจะถูกนำมาสร้างและฟิต โดยใช้ฟังก์ชันออโตอริมาในไลบารีฟอร์แคสของโปรแกรมอาร์ โดยแต่ละตัวแบบอริมานั้นจะถูกนำมาประเมินความถูกต้องโดยใช้ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนสัมบูรณ์สมมาตรกับข้อมูลที่ถูกแบ่งมาเพื่อตรวจสอบตัวแบบเพื่อที่จะพยากรณ์ค่าในเวลาถัดไป ขั้นตอนวิธีจะระบุกลุ่มรวม ณ เวลานั้น และตัวแบบอริมาที่ฟิตดีที่สุดของลำดับย่อยที่บรรจุกลุ่มของเวลานั้นอยู่ จากนั้นในขั้นตอนวิธีจะแยกค่าของกลุ่มการรวมที่พยากรณ์สู่ค่าต่างๆในอนาคต โดยใช้อัตราส่วนของข้อมูลสำหรับการสร้างตัวแบบมาซา ตัวแบบมาซาจะถูกเปรียบเทียบโดยใช้ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนสัมบูรณ์สมมาตรกับตัวแบบซาริมา และตัวแบบการปรับเรียบเอ็กซ์โพเนนเชียล ซึ่งการวิเคราะห์เปรียบเทียบแสดงผลว่าค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนสัมบูรณ์สมมาตรของตัวแบบมาซาดีกว่าทั้งสองตัวแบบ

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต..
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อ อ.ที่ปรึกษาหลัก..............................
ปีการศึกษา	2559	

5872010223: MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES

KEYWORDS: TIME SERIES / MULTIPLE ARIMA / AGGREGATE; SUBSEQUENCES OF TIME SERIES / DISAGGREGATE FORECAST / ATM

PAISIT KHANARSA : MULTIPLE ARIMA SUBSEQUENCES AGGREGATE TIME SERIES MODEL TO FORECAST CASH IN ATM. ADVISOR : Assistant Professor Krung Sinapiromsaran, Ph.D., 84 pp.

Presently, ATM is used to conveniently serve customers in financial banking for multiple services such as transfer, deposit, and withdrawal. The main concern of ATM is the lack of banknotes for customer withdrawing. Hence, one of the interesting ATM issues is to forecast cash demand on the individual ATM as a time series data. Typically, the customer cash usage behavior of ATM shows a sway and it often contains a hidden pattern due to multiple customer behaviors. Consequently, this research proposes a new forecasting model for predicting cash amount in ATM, called the multiple ARIMA subsequences aggregate time series model or the MASA model. In the algorithm process, the ATM data is split into the in-sample data for building the MASA model and the out-sample data for evaluating it. To build the MASA model, the aggregate technique is applied using the period of the in-sample data in order to reduce a sway of the time series which is subdivided into the training and the validating data. Then, subsequence patterns are generated and fitted using auto.arima function by the library forecast of the R programming language. They are evaluated using the symmetric mean absolute percentage error (SMAPE) on the validating data. In order to forecast a value of the next time step, the algorithm identifies the time step aggregate group and the best fitted ARIMA model from a subsequence containing that group. The algorithm then disaggregates the forecasted aggregate group value into individual future values using the ratio from the in-sample data. The MASA model is compared using SMAPE with the SARIMA and exponential smoothing model. The comparative analysis shows that the SMAPE of the MASA model is better than both models.

Department : ......Mathematics and Computer Sciences......................................	Student's Signature ....
Field of Study : ..... Applied Mathematics and Computational Sciences......................	Advisor's Signature ....
Academic Year ....2016............	

Acknowledgements

Firstly, I would like to express my sincere thanks to my thesis advisor, Assistant Professor Krung Sinapiromsaran for his invaluable help and constant encouragement of this research.

I would like to thank Assistant Professor Khamron Mekchay, Jiraphan Suntornchost and Thidaporn Supapakorn, the committee, for my dissertation defense.

In addition, I am grateful to Ms. Wipa Pimpapan, Mr. Chalee Boonprasop, Miss Keerati Kaewrak, Miss Watcharaporn Puengpo and AMCS students for suggestions and all their help. Moreover, I most gratefully acknowledge Science Achievement Scholarship of Thailand (SAST) for funding this research and the Applied Mathematics and Computational Science, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University for supporting the equipment to this work.

Finally, I would like to thank my parents for their increasing encouragement, support and attention throughout my life.

Contents

Chapter Page

List of Tables

List of Figures

Figure Page

CHAPTER I

INTRODUCTION

1.1 Motivation

Currently, the deposit, withdrawal and other services are offered by the financial institutions for customer convenient, especially, via the automatic teller machine or ATM (Handel et al. (1998)). Service quality in ATM composes of seven critical dimensions which are reliability, responsiveness, ease of use, convenience, fulfillment, security, and accuracy (Narteh (2013)). This research will focus on the fulfillment aspect of the service quality in ATM. The fulfillment of service quality shows that the ATM must give quality banknotes, provide enough money during transactions, satisfy most of the customer's banking needs, have a reasonable charge, provide fast services, and give instant money all the time. This research concentrates on the ATM forecasting which is a part of ATM replenishment operations. The ATM replenishment goal is to maintain the qualities of services of ATM having appropriate amount of banknotes for withdraws while guarding an excessive level of cashes to prevent the opportunity loss of investment. Consequently, the bank manager needs to forecast cash demand on each ATM as accurate as possible.

At present, the traditional ATM forecast techniques use an autoregressive or AR model and a moving average or MA model. The improved models are proposed and developed such as ARIMA, SARIMA by Box et al. (2015), ARCH by Engle (1983) and GARCH model by Bollerslev (1986). Recent methods use the machine learning algorithms together with the time series models such as F-ARIMA model by Sadek et al. (2003), the neural networks such as Simutis et al. (2008) to forecast more accurately.

The F-ARIMA was developed by Sadek et al. (2003) on ATM dynamic bandwidth allocation problem comparing the result with AR model. Neural networks and support vector regressive algorithm were used by Simutis et al. (2008) on the daily cash demand for the ATM. The application of both models was slightly better and the results could be achieved using a forecasting method based on the flexible of ANN. The fuzzy neural network was proposed by Darwish (2013) to forecast cash demand for the ATM. An interval type-2 fuzzy neural network had been applied to the simulation results for ATM cash forecasting and showed the effectiveness of the model.

The typical example of time series of daily cash in ATM is shown in Figure 1.1. Evidently, the structure of the time series is sway. As a result, this research aims to reduce the sway that occurs in the time series by applying the aggregate technique based on the observation that if the time series data has large sway then the accuracy of forecasting is low. However, if it has small sway then the fitted model should be more accurate to forecast the future value. Moreover, the customer cash-spending behavior is embedded in the observed time series values. So this research proposes the use of different ARIMA models for each subsequence to capture these hidden patterns. Finally, the future values are forecasted by the best fitted model with a minimum error.

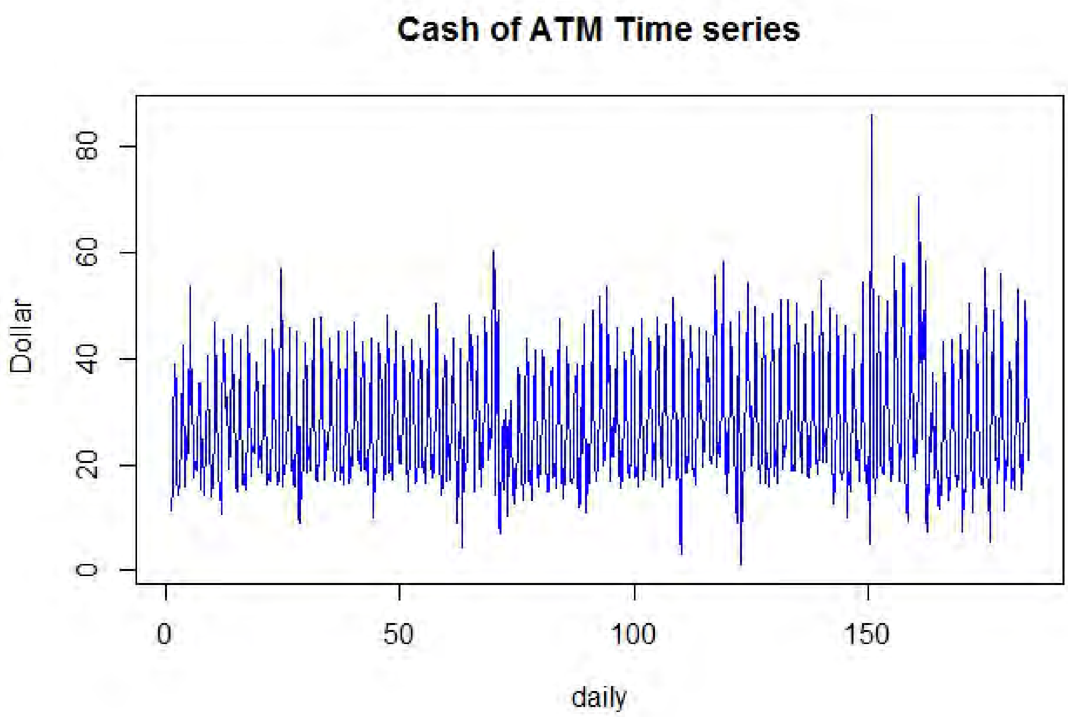


Figure 1.1: Time series plot of ATM

A new forecasting model of time series, called the multiple ARIMA subsequences aggregate time series model or the MASA model is proposed and applied with the ATM data on the daily NN5 datasets which consist of 111 time series. The NN5 dataset is a collection of ATM time series (Crone (2008)) which is applied with the hybrid models by Wichard (2011). They exhibited the best performance from the mixture of the neural network ensemble model, the one-year-cycle model and the nearest trajectory model. In the same year, Andrawis et al. (2011) also forecasted the weekly values and then converted them to daily values using a simple linear interpolation scheme.

The aggregation of the time series was presented by many researchers. Amemiya and Wu (1972) proposed the aggregation effect on the prediction of ARIMA models which were known as the high frequency ARIMA processes aggregate to low frequency ARIMA processes. Pino et al. (1987) applied the aggregation to the ARIMA model together with the linear combinations of time series and special cases of temporal, contemporaneous aggregations and systematic sampling.

This research proposes two algorithms to build the MASA model and to forecast the future values. The first algorithm comprises of the aggregate step, the split step, the subsequence generation step, and the fitted step. Firstly, the aggregate step combines the in-sample data into groups in order to reduce the sway using the time series period. Secondly, the split step subdivides the aggregate groups into the training and the validating time series data. Thirdly, the subsequence generation step produces a provisional number of subsequence patterns. Lastly, the fitted step uses the training time series data to determine the best fitted ARIMA models on each subsequence using the SMAPE.

The second algorithm comprises of the identification step, the forecast step and the disaggregate step. Firstly, the identification step pinpoints the aggregate group and its corresponding subsequence of the particular time. Secondly, the forecast step selects and forecasts the best ARIMA models of all subsequences that contain this aggregate group. Lastly, the disaggregate step applies the empirical ratio of each component of the aggregate to generate the forecasted values. For the result comparisons, SMAPE and Wilcoxon signed-rank test are used with the SARIMA model and the exponential smoothing model of NN5 111 time series data.

1.2 Objectives

The goals of this research are (1) to construct a new time series model utilizing multiple ARIMA models on subsequences of aggregate time series or the MASA model and (2) to compare the results by the accuracy of the proposed method with the SARIMA model and the ETS exponential smoothing model via SMAPE on the ATM dataset.

1.3 Structure of this thesis

This research focuses on the ATM problems dealing with sway and the hidden patterns to forecast the cash demand.

- This research presents the aggregating technique to reduce the sway in the ATM cash time series. Then the disaggregating technique is used to generate the value at particular time.
- This research uses the subsequencing technique to determine the best ARIMA model for each aggregate group.

The remainders of this research are organized as follows. Chapter 2 describes the basic knowledge and the basic model that use in this research. Chapter 3 introduces the data description. Chapter 4 introduces the details of the proposed algorithms, the building MASA algorithm, the forecasting MASA algorithm and the measurement of the model. Chapter 5 shows the empirical results by testing with NN5 dataset and compares it with the SARIMA model and the exponential smoothing model. Chapter 6 contains a conclusion.

CHAPTER II

BACKGROUND KNOWLEDGE

In this chapter, the necessary background knowledge of time series that requires in this research is introduced (S.Prasanna, William (1990)). Firstly, the background knowledge of time series is illustrated in section 2.1 - 2.4. Secondly, The time series models are revealed in section 2.5. Lastly, The automatic forecasting algorithm for R programming is illustrated in section 2.7.

2.1 Stochastic process and time series definition

In this section, the fundamental concepts of time series that are necessary for this research are introduced. It introduces the stochastic process time series definition, autocorrelation and partial autocorrelation function, respectively.

Definition 2.1 *Let $Z(\omega, t)$ be a random variable, where ω be a sample space and t be an index set. For a given ω and $Z(\omega, t)$ be a function of t. $Z(\omega, t)$ can be called a sample function or realization.*

Definition 2.2 *A stochastic process is a family of random variable $Z(\omega, t)$, where ω be a sample space and t be an index set.*

Next, the time series is defined by the definition of the stochastic process.

Definition 2.3 *A time series is a realization from a certain stochastic process and the population that consists of all possible realizations.*

Definition 2.4 *Let the index set be the set of all integers. A finite set of random variables $\{Z_{T_1}, Z_{T_2}, \ldots, Z_{T_n}\}$ is considered from a stochastic process $\{Z(\omega, t){:}t = 0, \pm1, \pm2, \ldots\}$. The n-dimensional distribution function is defined by*

$$F_{Z_{T_1}, Z_{T_2}, \ldots, Z_{T_n}}(x_1, \ldots, x_n) = P\{\omega : Z_{T_1} \leq x_1, \ldots, Z_{T_n} \leq x_n\}$$

where $x_i, i = 1, \ldots, n$ are any real numbers.

Definition 2.5 *A process is called first-order stationary if* $F_{Z_{t_1}}(x_1) = F_{Z_{t_1+k}}(x_1)$ *for any integers* t_1 *and* k.

A process is called second-order stationary if $F_{Z_{t_1},Z_{t_2}}(x_1, x_2) = F_{Z_{t_1+k},Z_{t_2+k}}(x_1, x_2)$ *for any integers* t_1, t_2 *and* k.

For n*th-order stationary,*

$$F_{Z_{t_1},Z_{t_2},\ldots,Z_{t_n}}(x_1, x_2, \ldots, x_n) = F_{Z_{t_1+k},Z_{t_2+k},\ldots,Z_{t_n+k}}(x_1, x_2, \ldots, x_n)$$

for any integers $t_1, t_2, \ldots, t_n$ *and* k.

Definition 2.6 *A process is called strictly stationary if*

$$F_{Z_{t_1},Z_{t_2},\ldots,Z_{t_n}}(x_1, x_2, \ldots, x_n) = F_{Z_{t_1+k},Z_{t_2+k},\ldots,Z_{t_n+k}}(x_1, x_2, \ldots, x_n)$$

is true for any $n = 1, 2, \ldots$.

Definition 2.7 *For a given real-valued process* $\{Z(\omega, t) : t = 0, \pm 1, \pm 2, \ldots\}$*, the mean function of the process is*

$$\mu_t = E(z_t),$$

the variance function of the process is

$$\sigma_t^2 = E(Z_t - \mu_t)^2,$$

the covariance function between Z_{t_1} *and* Z_{t_2} *is*

$$\gamma(t_1, t_2) = E(Z_{t_1} - \mu_{t_1})(Z_{t_2} - \mu_{t_2}),$$

and the correlation function between Z_{t_1} *and* Z_{t_2} *is*

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\sigma_{t_1}^2}\sqrt{\sigma_{t_2}^2}}.$$

Since the distribution function of strictly stationary process is the same for all t*, the mean function* $\mu_t = \mu$ *is a constant and the variance function* $\sigma_t^2 = \sigma^2$ *is a constant.*

The covariance function is

$$\gamma(t_1, t_2) = \gamma(t - k, t) = \gamma(t, t + k) = \gamma_k,$$

and the correlation function is

$$\rho(t_1, t_2) = \rho(t-k, t) = \rho(t, t+k) = \rho_k,$$

where and $t_1 = t - k$ *and* $t_2 = t$.

2.2 The autocovariance and autocorrelation function

Definition 2.8 *For a stationary process* $\{Z_t\}$, *the mean function is* $E(z_t) = \mu$ *and the variance function* $Var(Z_t) = \sigma^2$ *, where* μ *and* σ^2 *are constant. The covariance between* Z_t *and* Z_{t+k} *is*

$$\gamma_k = Cov(Z_t, Z_{t+k}) = E(Z_t - \mu)(Z_{t+k} - \mu)$$

and the correlation between Z_t *and* Z_{t+k} *is*

$$\rho_k = \frac{Cov(Z_t, Z_{t+k})}{\sqrt{Var(Z_t)}\sqrt{Var(Z_{t+k})}} = \frac{\gamma_k}{\gamma_0}$$

ρ_k *is called the autocovariance function and* ρ_k *is called autocorrelation function(ACF).*

Theorem 2.1 *For a given observed time series* $Z_1, Z_2, \ldots, Z_n$, *the sample ACF is defined as*

$$\hat{\rho}_k = \frac{\gamma_k}{\gamma_0} = \frac{\sum_{t=1}^{n-k}(Z_t - \bar{Z})(Z_{t+k} - \bar{Z}_t)}{\sum_{t=1}^{n}(Z_t - \bar{Z})^2},$$

$k = 0, 1, \ldots,$ *where* $\bar{Z}_t = \dfrac{\sum_{t=1}^{n} Z_t}{n}$ *is the sample mean of the series.*

Example 1 *: Let* $Z_1, Z_2, \ldots, Z_{10}$ *be the time series data. It is shown in Table 2.1.*

t	1	2	3	4	5	6	7	8	9	10
Z_t	13	8	15	4	4	12	11	7	14	12

Table 2.1: Time series data $Z_1, Z_2, \ldots, Z_{10}$

The sample mean of the time series data is

$$\bar{Z}_t = \frac{\sum_{t=1}^{n} Z_t}{10} = \frac{13+8+15+4+4+12+11+7+14+12}{10} = 10.$$

Consequently, the ACF of the time series data can be computed as

$$\hat{\rho}_1 = \frac{\sum_{t=1}^{9}(Z_t - \bar{Z})(Z_{t+1} - \bar{Z}_t)}{\sum_{t=1}^{10}(Z_t - \bar{Z})^2}$$

$$= \frac{(13-10)(8-10)+(8-10)(15-10)+\cdots+(14-10)(12-10)}{(13-10)^2+(8-10)^2+\cdots+(12-10)^2} = -0.188$$

$$\hat{\rho}_2 = \frac{\sum_{t=1}^{8}(Z_t - \bar{Z})(Z_{t+2} - \bar{Z}_t)}{\sum_{t=1}^{10}(Z_t - \bar{Z})^2}$$

$$= \frac{(13-10)(15-10)+(8-10)(4-10)+\cdots+(7-10)(12-10)}{(13-10)^2+(8-10)^2+\cdots+(12-10)^2} = -0.201$$

$$\hat{\rho}_3 = \frac{\sum_{t=1}^{7}(Z_t - \bar{Z})(Z_{t+3} - \bar{Z}_t)}{\sum_{t=1}^{10}(Z_t - \bar{Z})^2}$$

$$= \frac{(13-10)(4-10)+(8-10)(4-10)+\cdots+(11-10)(12-10)}{(13-10)^2+(8-10)^2+\cdots+(12-10)^2} = 0.181.$$

Other $\hat{\rho}_k$ can be calculated correspondingly.

2.3 The partial autocorrelation function

Definition 2.9 *For a stationary process $\{Z_t\}$, let the mean function be $\mu = E(z_t)$ and the variance function $Var(Z_t) = \sigma^2$, where μ and σ^2 are constant. The partial autocorrelation function ϕ_k between Z_t and Z_{t+k} is*

$$\phi_k = Corr(Z_t, Z_{t+k} | Z_{t+1}, \ldots, Z_{t+k-1}).$$

Theorem 2.2 *For a given observed time series $Z_1, Z_2, \ldots, Z_n$, the sample PACF for lag k is* *defined as*

$$\phi_k \approx \hat{\phi}_{kk} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k} \hat{\phi}_{kj}\hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k} \hat{\phi}_{kj}\hat{\rho}_j}$$

and

$$\hat{\phi}_{k+1,j} = \hat{\phi}_{kj} - \hat{\phi}_{k+1,k+1}\hat{\phi}_{k,k+1-j},$$

for $j = 1, 2, \ldots$.

From the example 1 in Section 2.2, the PACF of the time series data can be computed as

$$\hat{\phi}_{11} = \hat{\rho}_1 = -0.188$$

$$\hat{\phi}_{22} = \frac{\hat{\rho}_2 - \hat{\rho}_1^2}{1 - \hat{\rho}_1^2} = \frac{-0.201 - (-0.188)^2}{1 - (-0.188)^2} = -0.245$$

$$\hat{\phi}_{21} = \hat{\phi}_{11} - \hat{\phi}_{22}\hat{\phi}_{11} = -0.188 - (-0.245)(-0.188) = -0.234$$

$$\hat{\phi}_{33} = \frac{\hat{\rho}_3 - \hat{\phi}_{21}\hat{\rho}_2 - \hat{\phi}_{22}\hat{\rho}_1}{1 - \hat{\phi}_{21}\hat{\rho}_2 - \hat{\phi}_{22}\hat{\rho}_1} = \frac{0.181 - (-0.234)(-0.201) - (0.245)(-0.188)}{1 - (-0.234)(-0.201) - (0.245)(-0.188)} = 0.097$$

Other $\hat{\phi}_{kk}$ can be calculated correspondingly.

2.4 White noise process

A sequence $\{a_t\}$ is called a white noise process if the process is a sequence of uncorrelated random variables from a distribution.

The constant mean $E(a_t) = 0$, the constant variance $Var(a_t) = \sigma_a^2$ and $\gamma_k = Cov(a_t, a_t + k)$ for all $k \neq 0$.

A white noise process $\{a_t\}$ is stationary with the autocovariance function

$$\gamma_k = \begin{cases} \sigma_a^2 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}$$

the autocorrelation function

$$\rho_k = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}$$

and the partial autocorrelation function

$$\phi_{kk} = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}.$$

A white noise process is Gausian if it is a normal distribution.

2.5 Time series model

In this section, the basic of the time series models and the model that uses in this research are introduced. They are the autoregressive model, the moving average model and the autoregressive moving average model and the model that uses in this research is the autoregressive integrated moving average model. The details are shown as follow.

2.5.1 Autoregressive model

A general autoregressive or AR model of order p representing the time series Z_t can be written as

$$Z_t = \sum_{i=1}^{p} \phi_i Z_{t-i} + a_t$$

where, Z_t represents the time series data at time t, ϕ_i represents a constant for $i = 1, 2, \ldots, p$ and a_t represents the white noise terms at time t.

Example 2 *: To consider an $AR(2)$ model. A series is simulated of 250 values from $Z_t = -Z_{t-1} - 0.5Z_{t-2} + a_t$, with the a_t is the $N(0,1)$ white noise. The time series is plotted in Figure 2.1*

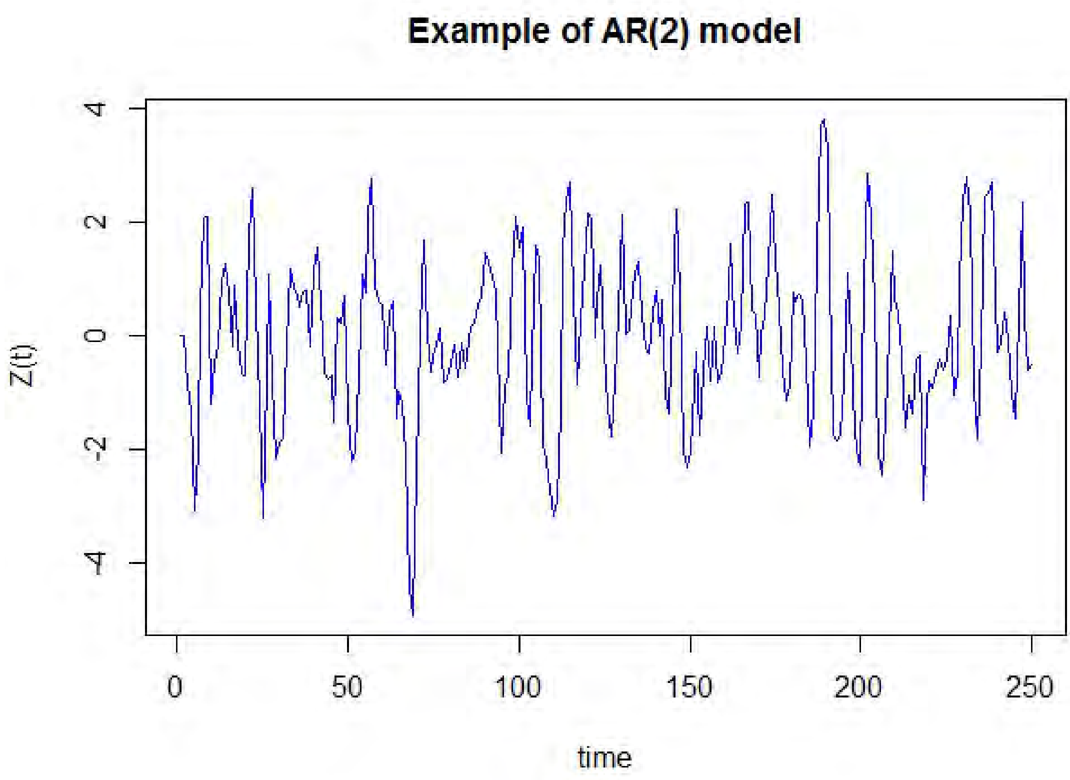


Figure 2.1: Examples of $AR(2)$ model: $Z_t = -Z_{t-1} - 0.5Z_{t-2} + a_t$.

The sample ACF and PACF of the time series are shown in Table 2.2, Figure 2.2 and Figure 2.3.

k	1	2	3	4	5	6	7	8	9	10
$\hat{\rho_k}$	0.626	0.099	-0.208	-0.209	0.000	0.118	0.116	0.020	-0.093	-0.204
$\hat{\phi_{kk}}$	0.626	-0.480	0.001	0.067	0.139	-0.118	0.069	-0.055	-0.051	-0.194

k	11	12	13	14	15	16	17	18	19	20
$\hat{\rho_k}$	-0.262	-0.179	-0.017	0.127	0.170	0.036	-0.110	-0.138	-0.060	0.048
$\hat{\phi_{kk}}$	-0.066	0.039	0.026	0.032	0.058	-0.132	0.021	0.011	-0.017	-0.034

Table 2.2: ACF and PACF of $AR(2)$: $Z_t = -Z_{t-1} - 0.5Z_{t-2} + a_t$

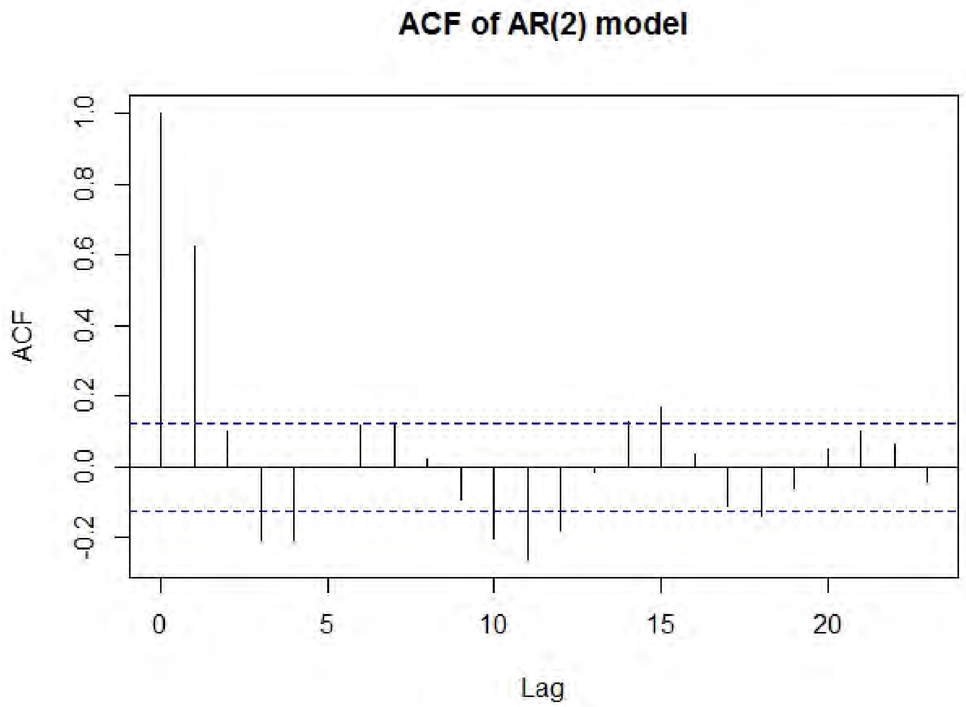


Figure 2.2: ACF of $AR(2)$ model: $Z_t = -Z_{t-1} - 0.5Z_{t-2} + a_t$.

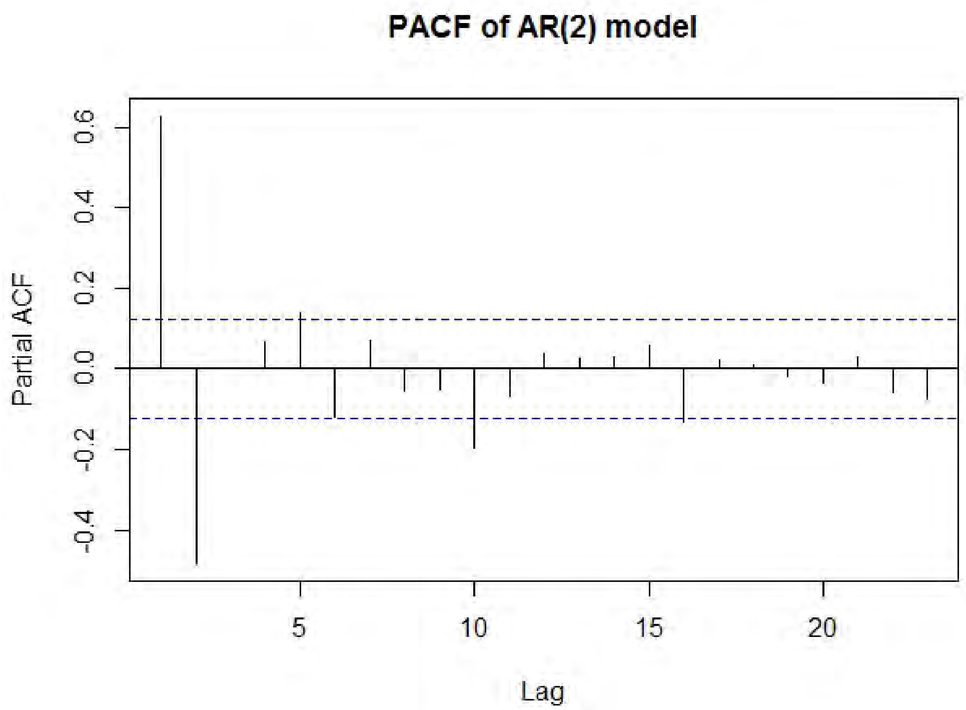


Figure 2.3: PACF of $AR(2)$ model: $Z_t = -Z_{t-1} - 0.5Z_{t-2} + a_t$.

2.5.2 Moving average model

A general moving average or MA model of order q representing the time series Z_t can be written as

$$Z_t = a_t + \sum_{i=1}^{q} \theta_i a_{t-i}$$

where, Z_t represents the time series data at time t, θ_i represents a constant for $i = 1, 2, \dots, q$ and a_t represents the white noise terms at time t.

Example 3 : *To consider an $MA(2)$ model. A series is simulated of 250 values from $Z_t = a_t - 0.65a_{t-1} - 0.24a_{t-2}$, with the a_t is the $N(0,1)$ white noise. The time series is plotted in Figure 2.4.*

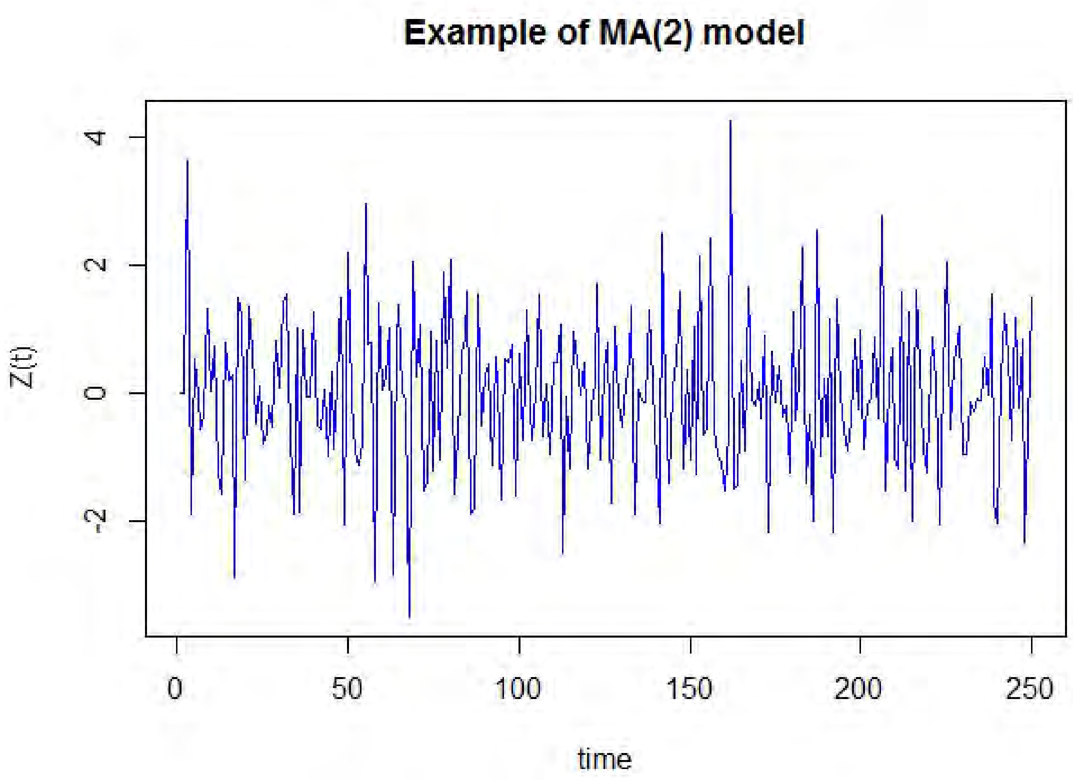


Figure 2.4: Examples of $MA(2)$ model: $Z_t = a_t - 0.65a_{t-1} - 0.24a_{t-2}$.

The sample ACF and PACF of the time series are shown in Table 2.3, Figure 2.5 and Figure 2.6.

k	1	2	3	4	5	6	7	8	9	10
$\hat{\rho_k}$	-0.335	-0.074	-0.138	0.015	0.119	-0.044	-0.056	-0.033	0.118	0.002
$\hat{\phi_{kk}}$	-0.335	-0.210	-0.281	-0.215	-0.038	-0.080	-0.125	-0.131	0.017	0.010
k	11	12	13	14	15	16	17	18	19	20
$\hat{\rho_k}$	-0.018	-0.114	0.024	0.040	-0.018	0.030	-0.070	-0.018	0.118	-0.079
$\hat{\phi_{kk}}$	0.010	-0.093	-0.073	-0.064	-0.109	-0.045	-0.099	-0.166	-0.018	-0.119

Table 2.3: ACF and PACF of $MA(2)$: $Z_t = a_t - 0.65a_{t-1} - 0.24a_{t-2}$

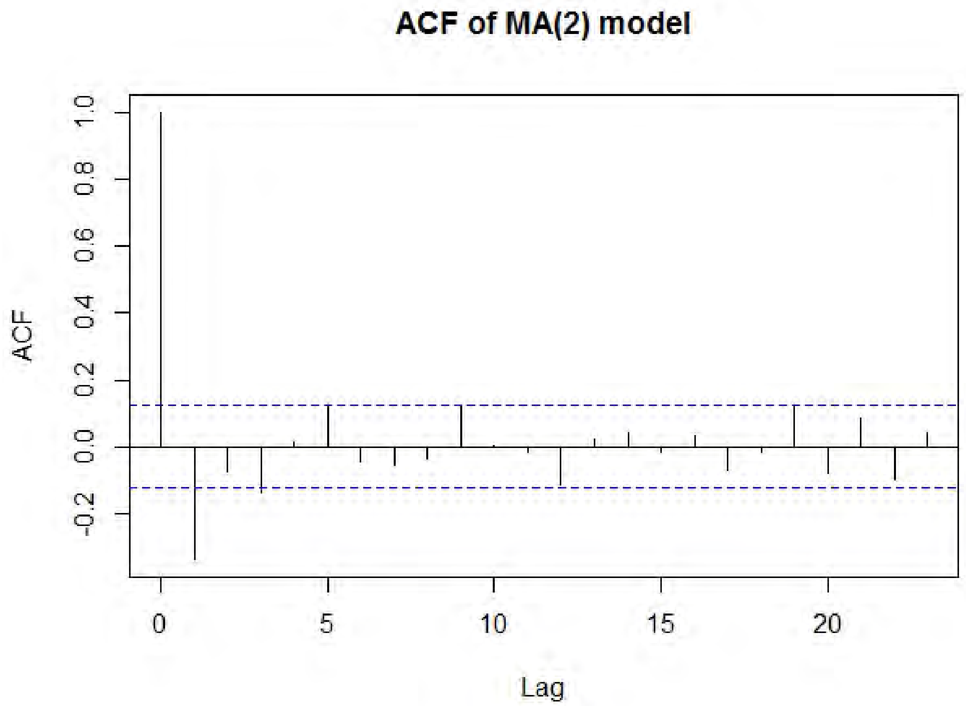


Figure 2.5: ACF of $MA(2)$ model: $Z_t = a_t - 0.65a_{t-1} - 0.24a_{t-2}$.

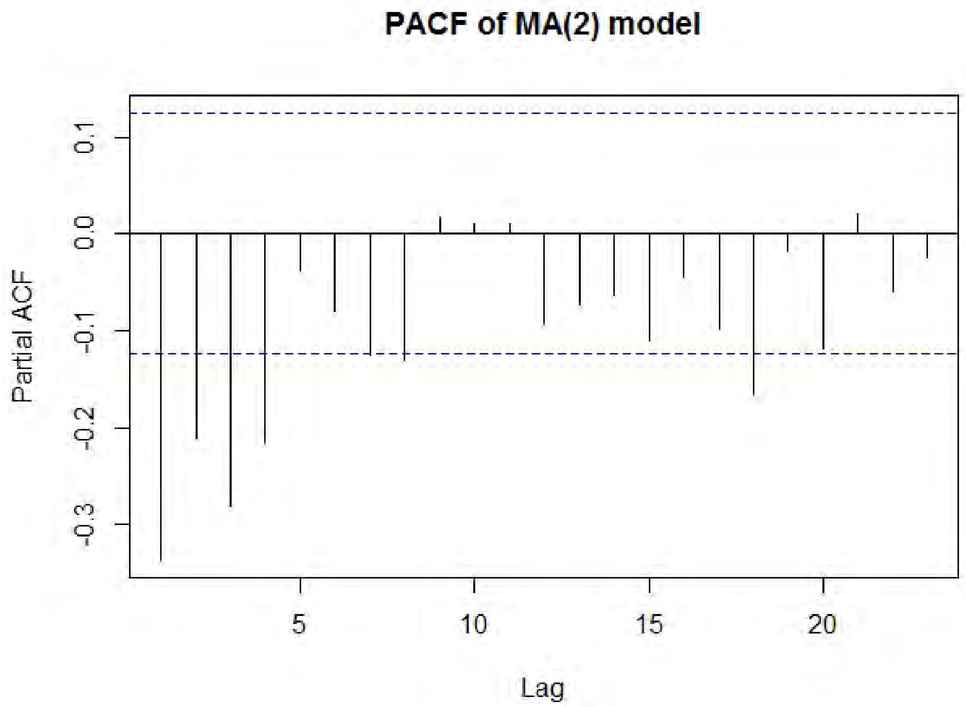


Figure 2.6: PACF of $MA(2)$ model: $Z_t = a_t - 0.65a_{t-1} - 0.24a_{t-2}$.

2.5.3 Autoregressive moving average model

A general autoregressive moving average or ARMA model of order (p, q) representing the time series Z_t can be written as

$$\phi(B)Z_t = \theta(B)a_t$$

where, Z_t represent the time series data at time t and a_t represent the white noise terms at time t. B is a lag operator defined by $BZ_t = Z_{t-1}$.

$\phi(B)$ and $\theta(B)$ are the autoregressive (AR) and the moving averages (MA) operators of order p and q respectively. They are defined as

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$$

$$\theta(B) = 1 + \sum_{j=1}^{q} \theta_j B^j$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients and $\theta_1, \theta_2, \ldots, \theta_q$ are the moving averages coefficients.

Example 4 *: To consider an $ARMA(1, 1)$ model. A series is simulated of 250 values from*

$Z_t = 0.9Z_{t-1} + a_t - 0.5a_{t-1}$, with the a_t is the $N(0,1)$ white noise. The time series is plotted in Figure 2.7.

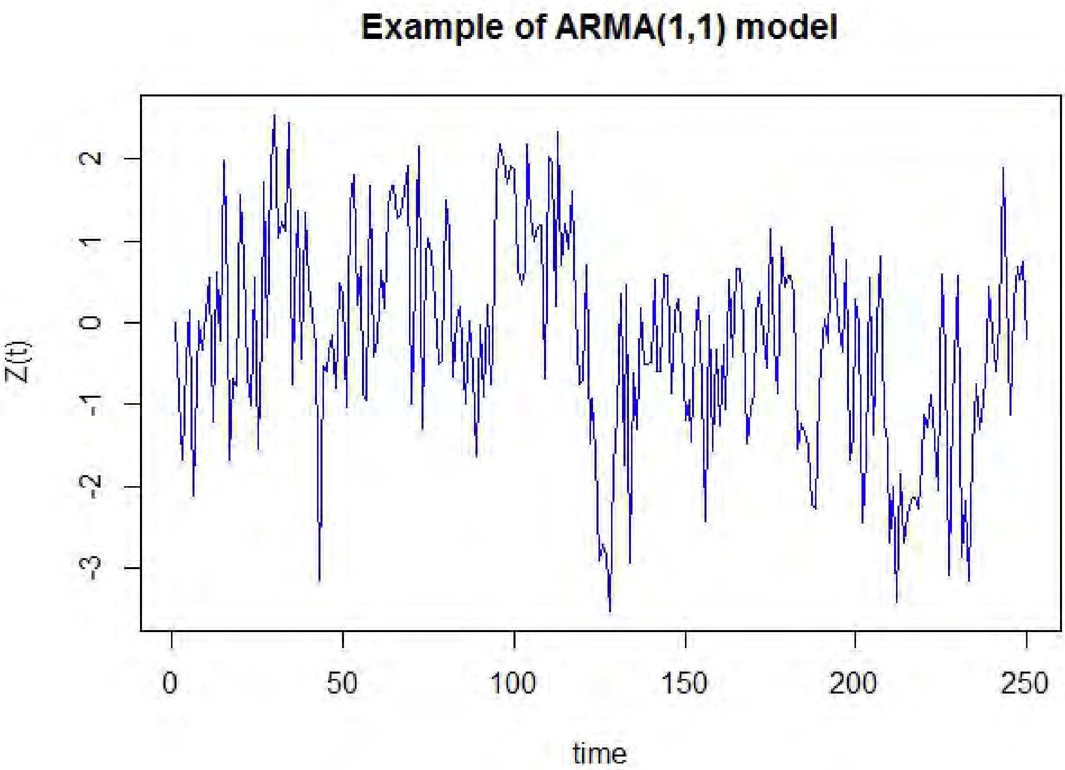


Figure 2.7: Examples of $ARMA(1,1)$ model: $Z_t = 0.9Z_{t-1} + a_t - 0.5a_{t-1}$.

The sample ACF and PACF of the time series are shown in Table 2.4, Figure 2.8 and Figure 2.9.

k	1	2	3	4	5	6	7	8	9	10
$\hat{\rho_k}$	0.502	0.425	0.450	0.386	0.306	0.278	0.213	0.191	0.154	0.155
$\hat{\phi_{kk}}$	0.502	0.232	0.239	0.087	-0.007	0.006	-0.050	0.003	-0.016	0.041

k	11	12	13	14	15	16	17	18	19	20
$\hat{\rho_k}$	0.105	0.108	0.123	0.137	0.087	0.139	0.052	0.106	0.094	-0.001
$\hat{\phi_{kk}}$	-0.027	0.023	0.046	0.062	-0.032	0.068	-0.111	0.065	-0.006	-0.115

Table 2.4: ACF and PACF of $ARMA(1,1) : Z_t = 0.9Z_{t-1} + a_t - 0.5a_{t-1}$

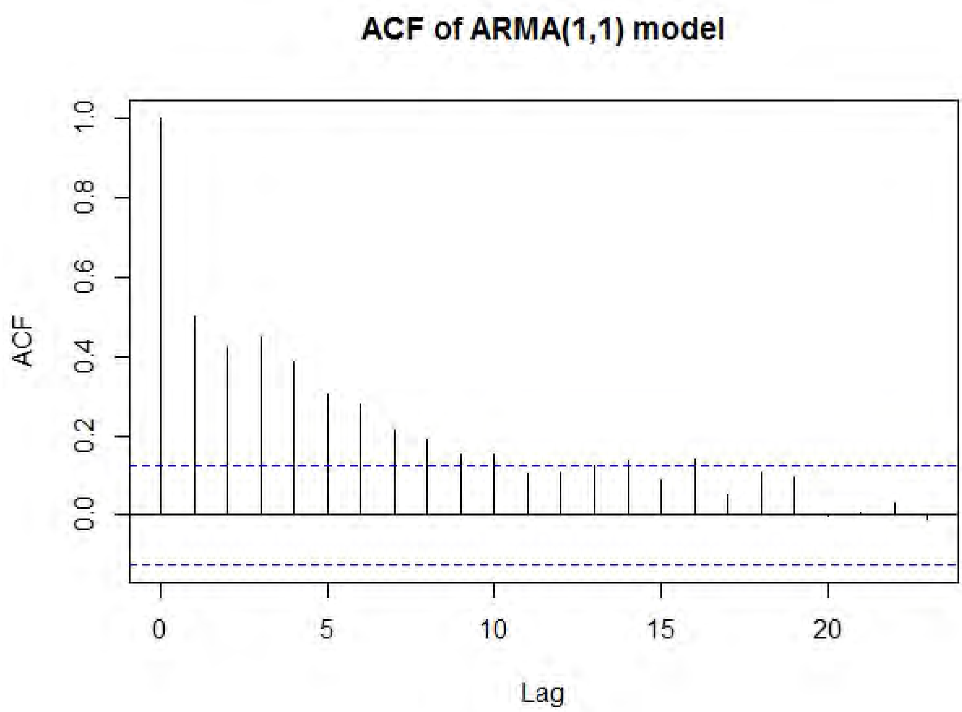


Figure 2.8: ACF of $ARMA(1,1)$ model: $Z_t = 0.9Z_{t-1} + a_t - 0.5a_{t-1}$.

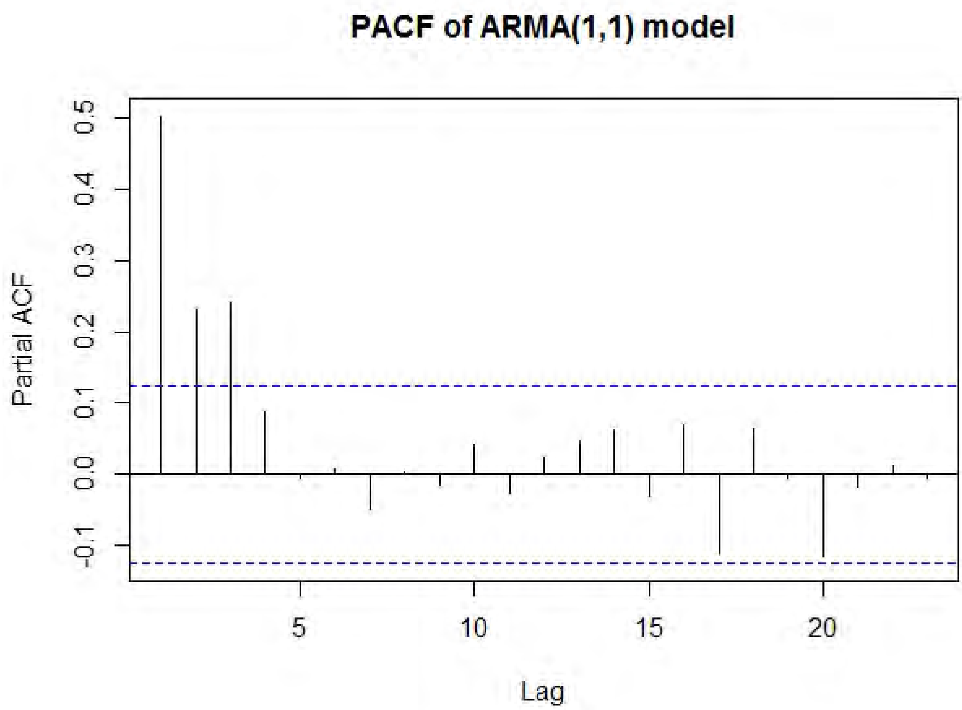


Figure 2.9: PACF of $ARMA(1,1)$ model: $Z_t = 0.9Z_{t-1} + a_t - 0.5a_{t-1}$.

2.5.4 Autoregressive integrated moving average model

A general autoregressive integrated moving average or ARIMA model of order (p, d, q) representing the time series Z_t can be written as

$$\phi(B)\nabla^d Z_t = \theta(B)a_t$$

where, Z_t represents the time series data at time t and a_t represents the white noise terms at time t. B is a lag operator defined by $BZ_t = Z_{t-1}$.

∇ is defined by $\nabla Z_t = Z_t - Z_{t-1} = (1 - B)Z_t$ and $\nabla = (1 - B)$.

$\nabla^d = (1 - B)^d$, d is the order of differencing.

$\phi(B)$ and $\theta(B)$ are the autoregressive (AR) and the moving averages (MA) operators of order p and q, respectively. They are defined as

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$$

$$\theta(B) = 1 + \sum_{j=1}^{q} \theta_j B^j$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients and $\theta_1, \theta_2, \ldots, \theta_q$ are the moving average coefficients.

Example 5 *: To consider an $ARIMA(1, 1, 1)$ model. A series is simulated of 250 values from $(1 - 0.9B)\nabla Z_t = (1 - 0.5B)a_t$, with the a_t is the $N(0, 1)$ white noise. The time series is plotted in Figure 2.10.*

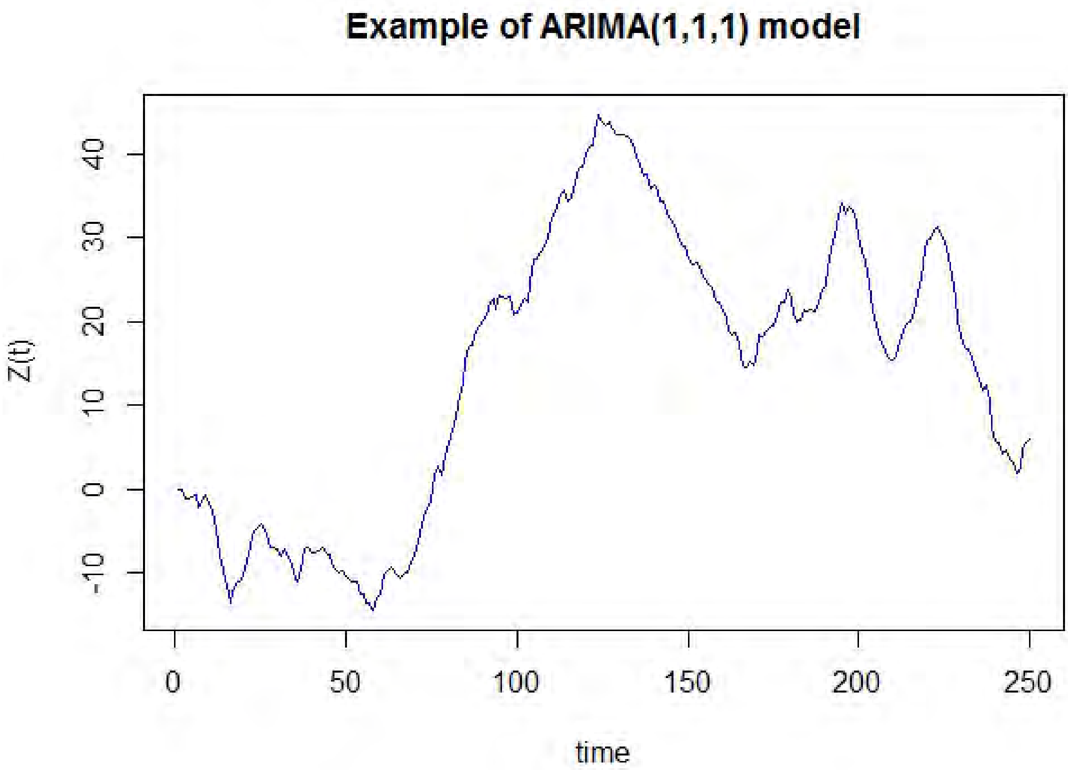


Figure 2.10: Examples of $ARIMA(1,1,1)$ model: $(1-0.9B)\nabla Z_t = (1-0.5B)a_t$.

The sample ACF and PACF of the time series are shown in Table 2.5, Figure 2.11 and Figure 2.12.

k	1	2	3	4	5	6	7	8	9	10
$\hat{\rho}_k$	0.995	0.987	0.977	0.963	0.948	0.932	0.914	0.896	0.878	0.860
$\hat{\phi}_{kk}$	0.995	-0.341	-0.201	-0.172	-0.041	0.026	-0.055	0.047	0.038	0.016

k	11	12	13	14	15	16	17	18	19	20
$\hat{\rho}_k$	0.842	0.824	0.806	0.789	0.771	0.753	0.735	0.718	0.701	0.684
$\hat{\phi}_{kk}$	-0.020	0.006	-0.059	-0.007	-0.031	-0.026	0.053	0.063	-0.036	-0.060

Table 2.5: ACF and PACF of $ARIMA(1,1,1) : (1-0.9B)\nabla Z_t = (1-0.5B)a_t$

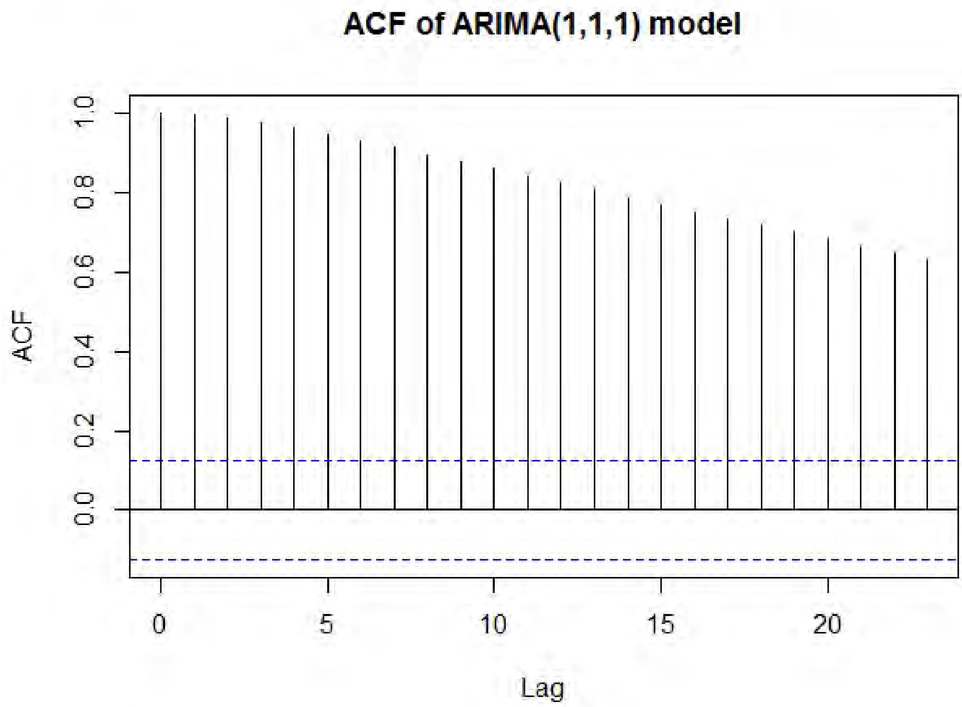


Figure 2.11: ACF of $ARIMA(1,1)$ model: $(1-0.9B)\nabla Z_t = (1-0.5B)a_t$.

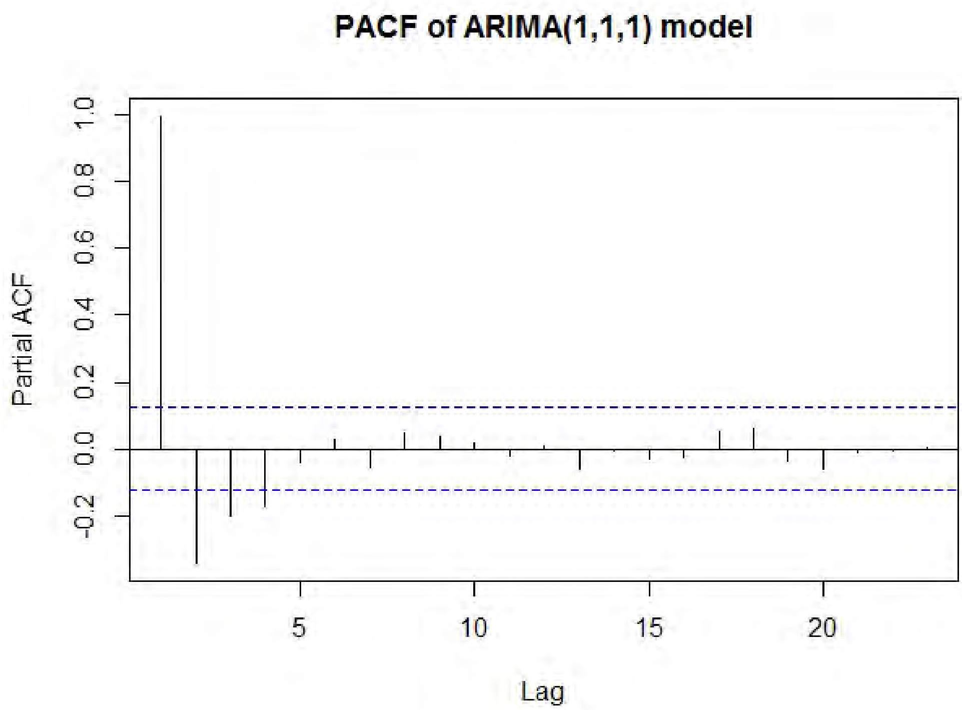


Figure 2.12: PACF of $ARIMA(1,1)$ model: $(1-0.9B)\nabla Z_t = (1-0.5B)a_t$.

2.6 Maximum likelihood method

The maximum likelihood method is used for finding estimator of an unknown parameter. Next, the methods are described below.

Let $z_1, z_2, \ldots, z_t$ be independent random samples from a population Z which belongs to a family of distribution functions governing by the unknown parameter θ of the form $f(z; \theta)$ where $\theta \in \Omega$. The likelihood function, $L(\theta; z_1, z_2, ..., z_t)$ or written shortly as $L(\theta)$, is the distribution of the sample. That is

$$L(\theta) = \prod_{i=1}^{t} f(z_i; \theta)$$

.

The θ is called the maximum likelihood estimator and denoted by $\hat{\theta}$. Hence,

$$\hat{\theta} = \operatorname{Arg} \sup_{\theta \in \Omega} L(\theta)$$

where Ω is the parameter space of θ thus $L(\theta)$ is the joint density of the sample.

Example 6 *Let $z_1, z_2, \ldots, z_t$ be random samples from a normal population with mean (μ) and variance (σ^2). The maximum likelihood estimator of μ and σ^2 can be shown as follow:*

Since $Z \sim N(\mu, \sigma^2)$, the probability density function of Z is given by

$$f(z : \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z-\mu)^2}{\sigma^2}}.$$

The likelihood function of the sample is given by

$$L(\mu, \sigma^2) = \prod_{i=1}^{t} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z_i-\mu)^2}{\sigma^2}}.$$

Hence, the logarithm of this likelihood function is given by

$$\ln L(\mu, \sigma^2) = -\frac{t}{2}\ln(2\pi) - t\ln(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{t}(z_i - \mu)^2.$$

The maximum likelihood function can be calculated by taking the partial derivatives with respect to μ and σ^2, then

$$\frac{\partial}{\partial\mu}\ln L(\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{t}(z_i-\mu)(-2) = \frac{1}{\sigma^2}\sum_{i=1}^{t}(z_i-\mu)$$

and

$$\frac{\partial}{\partial\sigma}\ln L(\mu,\sigma^2) = -\frac{t}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{t}(z_i-\mu)^2$$

.

Setting $\frac{\partial}{\partial\mu}\ln L(\mu,\sigma^2) = 0$ and $\frac{\partial}{\partial\sigma}\ln L(\mu,\sigma^2) = 0$, and solving for the unknown μ and σ^2 , thus

$$0 = \frac{1}{\sigma^2}\sum_{i=1}^{t}(z_i-\mu) = \sum_{i=1}^{t} z_i - t\mu \Rightarrow \mu = \bar{z}$$

Thus the maximum likelihood estimator of μ is

$$\mu = \bar{z}.$$

Similarly, thus

$$0 = -\frac{t}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{t}(z_i-\mu)^2$$

.

implies

$$\sigma^2 = \frac{1}{t}\sum_{i=1}^{t}(z_i-\mu)^2.$$

Then the estimator of μ in the above expression is used for estimating σ^2

$$\hat{\sigma^2} = \frac{1}{t}\sum_{i=1}^{t}(z_i-\bar{z})^2.$$

2.7 The automatic forecasting algorithm for R programming

In this section, the automatic forecasting algorithm for R programming that uses in this research is illustrated. The idea of the algorithm is discussed by Hyndman et al. (2007). The forecasting algorithm is implemented in the forecast package for R using ARIMA models.

2.7.1 Identifying the order (p, d, q) of ARIMA model

A general ARIMA(p, d, q) process is given by

$$(1 - B)^d Z_t = c + \phi(B)y_t + \theta(B)a_t$$

where, Z_t represents the time series data and a_t represents the white noise terms at time t. B is a lag operator. $\phi(B)$ and $\theta(B)$ are the polynomials of order p and q, respectively.

To guarantee the stationary of the time series, it is assumed that $\phi(B)$ and $\theta(B)$ have roots for $|B| > 1$ (William (1990)). If $c \neq 0$, a polynomial of order d is implied in the forecast function.

The concept of automatic ARIMA forecasting is choosing an appropriate order (p, d, q) of the model. If differencing d is known, the order (p, q) are chosen by a criterion AIC:

$$AIC = -2\log(L) + 2(p + q + k)$$

where $k = 1$, if $c \neq 0$ and $k = 0$ if otherwise, and L is the maximum likelihood of the model that fitted with the $(1 - B)^d Z_t$.

Akaike Information Criterion (AIC) is useful in selecting predictors for determining the order of an ARIMA model. The good models are obtained by minimizing the AIC.

For the general ARIMA (p, d, q) model, the differencing d is selected by using successive KPSS unit-root tests (Kwiatkowski et al. (1992)). The data is tested for a unit root, if the test result is significant then the next differenced data is tested for a unit root. When the procedure obtains the first insignificant result, it stops.

2.7.2 A procedure of the automatic forecasting ARIMA algorithm

Suppose the general ARIMA (p, d, q) models are considered, where p and q can be vary from 0 to m_p and m_q, respectively. Note that m_p is the maximum value of p and m_q is the maximum value of q. Consequently, a way of selecting models efficiently is needed to find the model that has the lowest AIC value.

Steps of the algorithm are proposed as follows:

Step 1: Four possible models are considered.

- ARIMA$(1, d, 1)$.

- ARIMA$(0, d, 0)$.

- ARIMA$(1, d, 0)$.

- ARIMA$(0, d, 1)$.

If $d < 1$, these models are fitted with $c \neq 0$. Otherwise, set $c = 0$. From the four possible models, one of these is selected with the smallest AIC values, called the current model.

Step 2: The variations are considered up on the current model.

- One of p and q is allowed to vary by ± 1 from the current model.

- Both of p and q vary by ± 1 from the current model.

- The constant c is added, if the current model has $c = 0$ or removed if the current model has $c \neq 0$.

When a model with lower AIC is found, it becomes the new current model and the procedure is repeated on step 2 to all models for determining the current model with the lowest AIC. The algorithm is confirmed to return a model because the number of the models is finite.

After, the necessary background knowledge and the fundamental concepts of time series are introduced. The research will use this knowledge to build a new model for forecasting cash in ATM. The main model used in this research is autoregressive integrated moving average or ARIMA model. Then, the automatic forecasting algorithm for R programming is used to identify the various parameters and all coefficients in the ARIMA model. The details of the new model are described in the chapter 4.

CHAPTER III

DATA DESCRIPTION

3.1 Competition NN5

The NN5 competition is the international competition to recruit participants to predict the amount of cash in ATMs. The purpose of the forecasting time series data of cash withdrawals from ATMs is to accurate predict cash using the computational intelligence methods. The ATM data includes the amount of cash withdrawals in each day from many ATMs in different locations in England for a period of two years.

The cash machines operate as scaled-down retail outlets that provide cash banknotes to the customers. The ATM data contains a number of time series patterns including multiple seasonality, local trends, structural breaks, outliers, zero and missing values etc. These are driven by a combination of unknown and unobserved causal such as seasonal periods, bank holidays, or special events.

3.2 The description of the NN5 data

The NN5 dataset is comprised of two datasets of different size but similar complexity. The dataset A is a complete dataset of 111 daily time series data from the homogeneous population of the empirical business time series. The dataset B is a sub-sample of 11 time series data from the 111 time series. This research uses all 111 time series from the NN5 dataset. Both datasets can be downloaded from http://www.neural-forecasting-competition.com/NN5.

The time series data is provided from the different cash machines at different, randomly selected locations within England. The time series data is collected staring on March 18, 1996 to March 22, 1998. It contains both observations of 0 withdrawals, which indicate that cash withdrawal does not occur in that day, and missing observations which are blank without any values.

The data includes 112 columns. The first column shows the dates while the other columns show the cash usage of each ATM. The first row shows the name ID of each ATM, the second row shows the starting year, the third row shows the starting date and the fourth row shows the status of the dataset. The examples of NN5 data are shown in Figure 3.1. For the cash withdrawals in

NN5 dataset, the decimal separator uses the period (.), for example, 1.000 dollars means 1,000 dollars.

Time Series #ID	NN5-001	NN5-002	NN5-003	NN5-004	NN5-005	NN5-006	NN5-007	NN5-008	NN5-009
Start Year	1996	1996	1996	1996	1996	1996	1996	1996	1996
Starting day	77	77	77	77	77	77	77	77	77
Dataset	complete	complete	complete	complete	complete	complete	complete	complete	complete
18-มี.ค.-96	13.407	11.550	5.641	13.180	9.779	9.240	14.938	2.891	7.341
19-มี.ค.-96	14.725	13.591	14.399	8.447	10.813	11.635	16.284	12.358	9.155
20-มี.ค.-96	20.564	15.037	24.419	19.515	21.613	12.103	16.667	16.383	10.587
21-มี.ค.-96	34.708	21.570	28.784	28.883	38.520	21.414	23.569	30.159	12.500
22-มี.ค.-96	26.630	19.444	20.621	19.473	24.745	24.674	26.304	31.179	7.157
23-มี.ค.-96	16.610	0.000	13.804	0.000	12.330	5.230	14.895	19.813	5.641
24-มี.ค.-96	15.320	9.722	11.536	7.355	12.996	11.380	16.043	17.489	7.696
25-มี.ค.-96	11.607	12.245	10.743	10.828	11.040	9.552	13.889	8.716	5.726
26-มี.ค.-96	19.884	15.505	14.824	15.618	7.951	15.717	20.394	15.505	5.513
27-มี.ค.-96	23.767	18.934	25.213	21.159	19.515	14.867	19.345	21.485	8.418
28-มี.ค.-96	34.028	26.077	35.133	33.489	33.773	26.885	28.416	35.289	13.308
29-มี.ค.-96	33.787	27.253	24.787	27.905	26.913	30.343	31.349	47.491	7.185
30-มี.ค.-96	18.254	11.706	12.897	13.223	7.993	3.827	2.253	10.062	4.195
31-มี.ค.-96	19.388	14.938	15.122	14.753	6.250	13.648	14.909	18.807	7.200
1-เม.ย.-96	17.262	12.939	11.905	14.300	18.665	13.081	20.026	14.243	7.738
2-เม.ย.-96	23.810	18.169	17.730	10.672	15.873	16.652	20.592	20.266	9.510
3-เม.ย.-96	36.125	24.887	36.139	32.922	28.189	19.459	26.644	28.260	11.069
4-เม.ย.-96	33.603	15.717	22.704	20.337	32.355	16.185	18.155	31.930	6.207
5-เม.ย.-96	32.625	25.213	24.943	22.477	0.170	25.397	29.167	34.850	5.612

Figure 3.1: Nine time series ATM data from the NN5 dataset.

3.3 NN5 summary

In this section, the statistical descriptions of cash withdrawals of all 111 ATM time series are shown in Figure 3.2 and 3.3. The descriptive statistics are computed which comprise of mean and standard deviation appearing in line plot, boxplot and histogram.

The means and the standard deviation of all ATM time series are plotted in Figure 3.2. It shows that the behavior of the means of cash withdrawals on each ATM is different and has high variance which the range of the means of cash withdrawals is approximately between 7,000 dollars and 40,000 dollars. The boxplot and the histogram of the means are shown in Figure 3.3. They show that the behavior of the data is relatively symmetric and has one outlier.

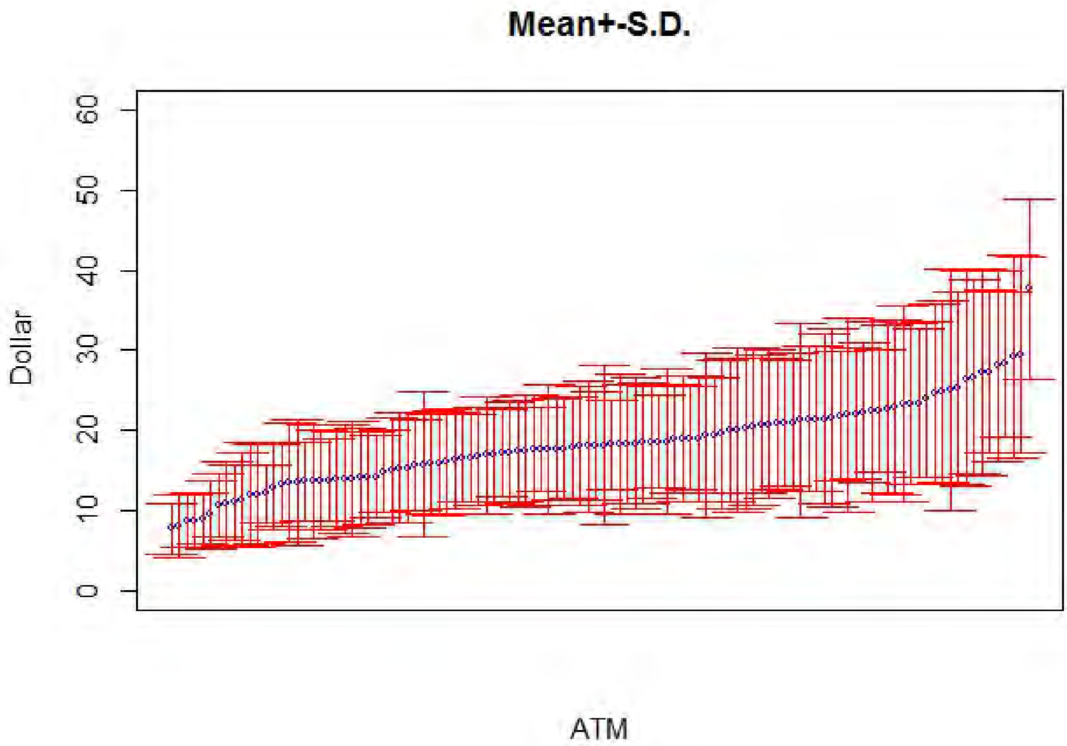


Figure 3.2: Line plot of mean and standard deviation of cash withdrawals

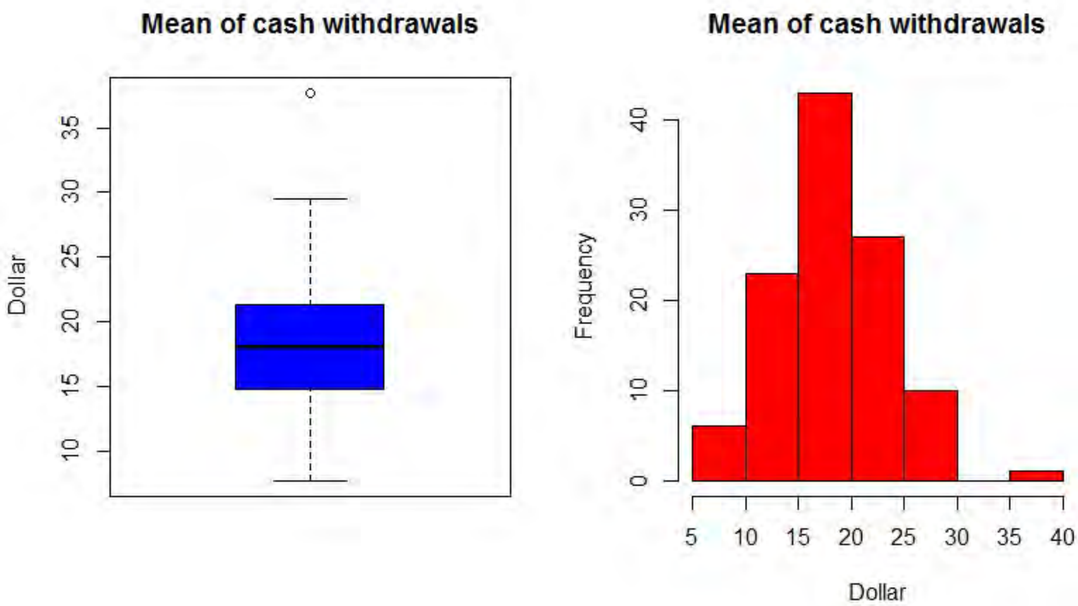


Figure 3.3: Boxplot and histogram of mean of cash withdrawals

CHAPTER IV

METHODOLOGY

In this chapter, the methodology of this research is introduced. This research proposes a new forecasting model to forecast cash in ATM, called the multiple ARIMA subsequences aggregate time series model or the MASA model. The algorithm description is illustrated in section 4.1, called the MASA algorithm.

The concept of the MASA algorithm is illustrated in section 4.2 and the details of the MASA model-building algorithm and the MASA model-forecasting algorithm are illustrated in section 4.3 and 4.4, respectively.

4.1 Overall MASA algorithm

The time series data is divided into the in-sample data and the out-sample data. First, the MASA algorithm builds the model using the in-sample data and then forecast its aggregate future values. Second, the MASA algorithm uses the out-sample data for evaluating the accuracy by the SMAPE. The overall diagram of the MASA algorithm is shown in Figure 4.1.

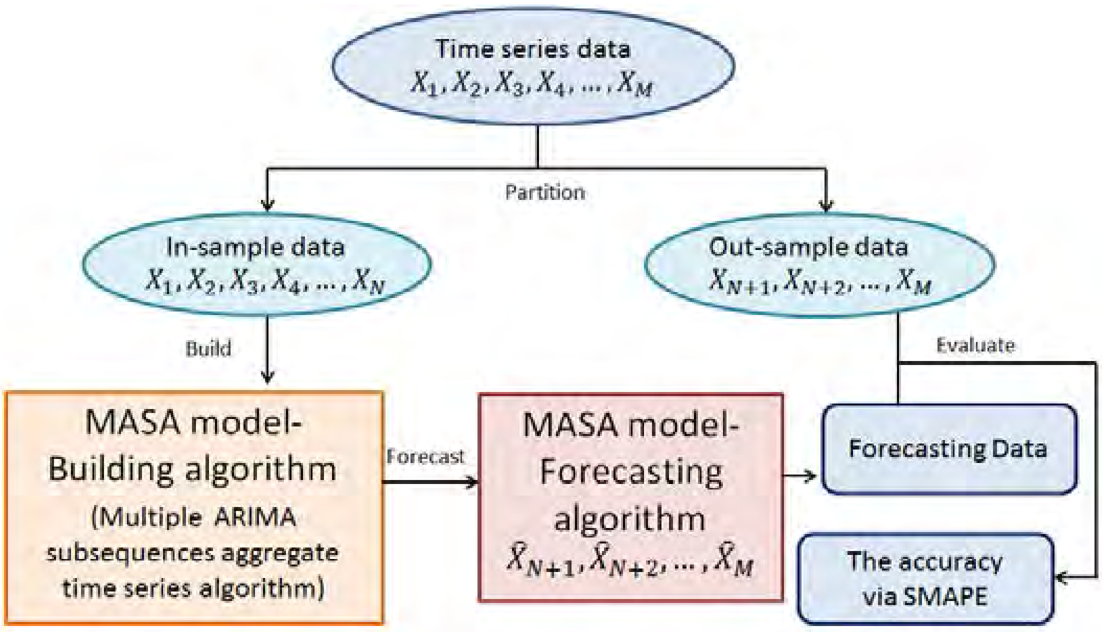


Figure 4.1: The overall diagram of the MASA algorithm

The notation $\hat{X}$ and $\hat{Y}$ are used as forecast values. Let $X_1, X_2, \ldots, X_N$ be the in-sample data and $X_{N+1}, X_{N+2}, \ldots, X_M$ be the out-sample data, where N is the number of the in-sample data that is the multiple of the number of periods and M is the total number of the time series data. Note that, the future values are $\hat{X}_{N+1}, \hat{X}_{N+2}, \ldots, \hat{X}_M$. The details of the MASA algorithm are described in section 4.3 and 4.4.

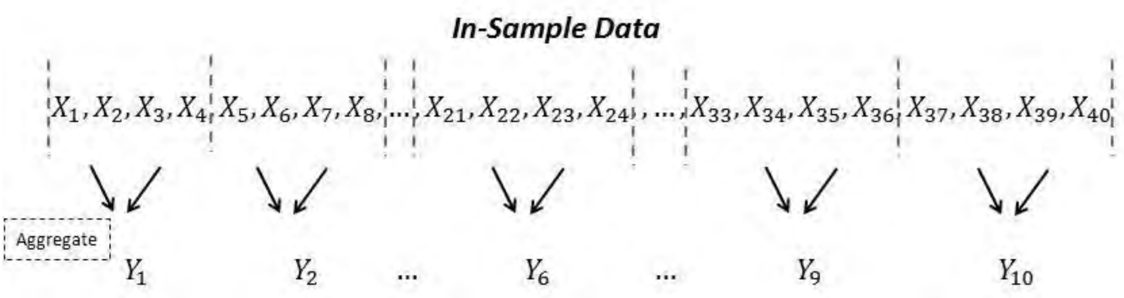
4.2 Concepts of the MASA algorithm

This section will illustrate the concepts of the MASA algorithm by examples.

4.2.1 Concept of the MASA model-building algorithm

Let $X_1, X_2, \ldots, X_{40}$ be the in-sample data.

- **First step**: The in-sample data are aggregated.

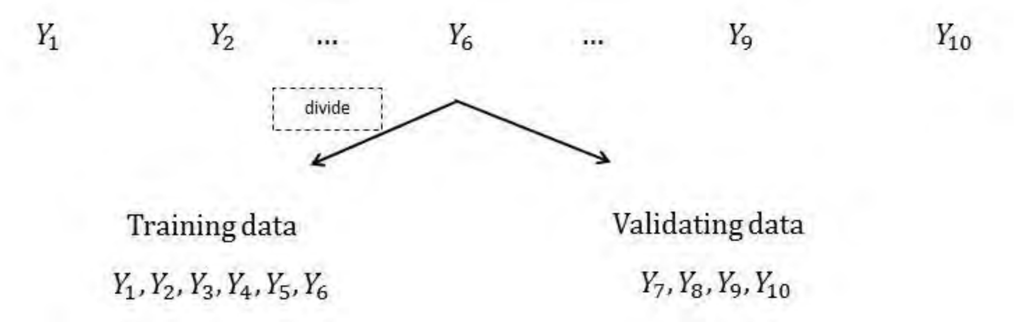


For the example, the period of aggregate is assumed to be 4. Thus, that the aggregate of in-sample data $Y_1, Y_2, \ldots, Y_{10}$ is defined as follow:

$$
\begin{aligned}
Y_1 &= \sum_{i=1}^{4} X_{(0\times 4)+i} \\
Y_2 &= \sum_{i=1}^{4} X_{(1\times 4)+i} \\
Y_3 &= \sum_{i=1}^{4} X_{(2\times 4)+i} \\
&\vdots \\
Y_{10} &= \sum_{i=1}^{4} X_{(9\times 4)+i}
\end{aligned}
$$

- **Second step**: The data is divided into the training data and the validating data.

For the example, The data $Y_1, Y_2, \ldots, Y_{10}$ can be divided into the training data $Y_1, Y_2, \ldots, Y_6$ and the validating data Y_7, Y_8, Y_9, Y_{10}.



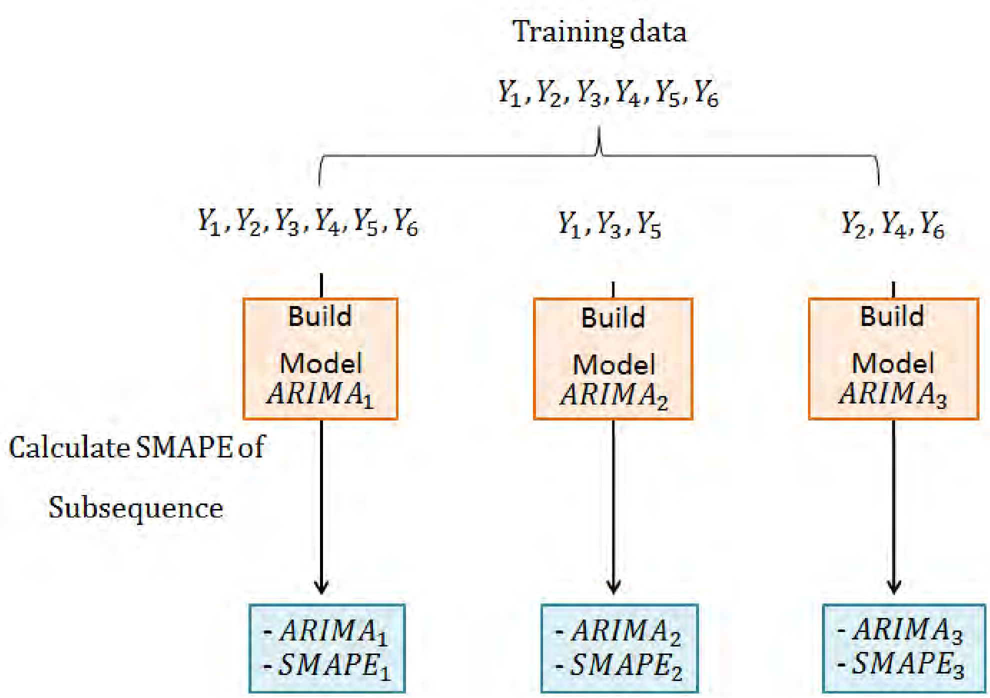
- **Third step**: The subsequences are generated.

For the example, the maximum common difference is assumed to be 2. Thus, the subsequences are generated as follow:

$$\begin{aligned} Y_1 &= \sum_{i=1}^{4} X_{(0\times4)+i} \\ Y_2 &= \sum_{i=1}^{4} X_{(1\times4)+i} \\ &\vdots \\ Y_{10} &= \sum_{i=1}^{4} X_{(9\times4)+i} \end{aligned}$$

- **Fourth step**: The best ARIMA model is fitted and the ratio of each subsequence is generated.

In this step, the ARIMA model is fitted by each subsequences in the training data and then the SMAPE is calculated from each model.



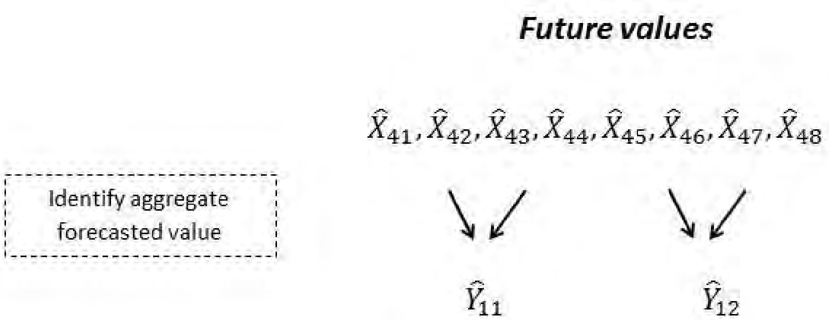
From this step, the MASA model consists of ARIMA model, and SMAPE of each subsequence.

4.2.2 Concept of the MASA model-forecasting algorithm

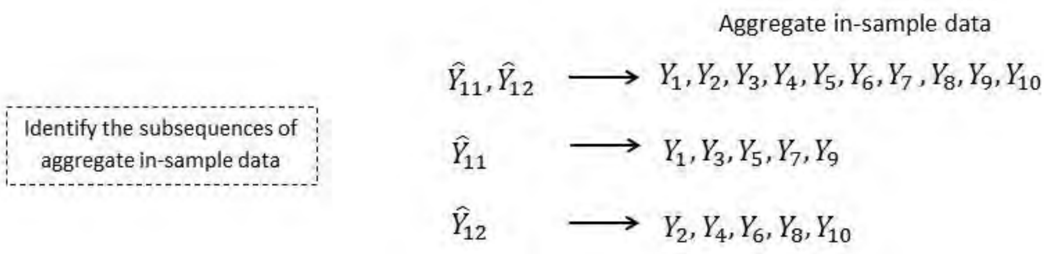
In this example, let the future values be $\hat{X}_{41}, \hat{X}_{42}, \ldots, \hat{X}_{48}$.

- **First step**: The aggregate forecasted value is identified from each future value.

From calculation $\hat{X}_{41}, \hat{X}_{42}, \hat{X}_{43}, \hat{X}_{44}$ are from $\hat{Y}_{11}$ and $\hat{X}_{45}, \hat{X}_{46}, \hat{X}_{47}, \hat{X}_{48}$ are from $\hat{Y}_{12}$.

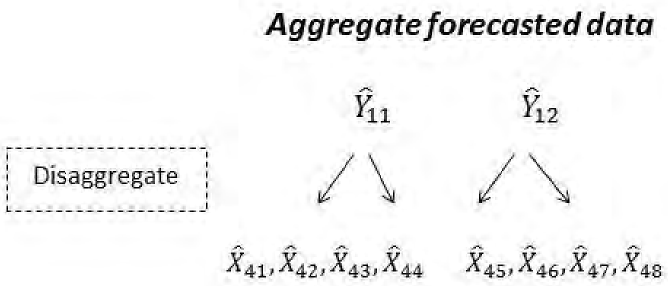


Then all subsequences containing $\hat{Y}_{11}$ and $\hat{Y}_{12}$ are identified. For example, the subsequence $Y_1, Y_2, \ldots, Y_{10}$ can be used to forecast $\hat{Y}_{11}, \hat{Y}_{12}$ and the subsequence $Y_1, Y_3, \ldots, Y_9$ can forecast $\hat{Y}_{11}$, and the subsequence $Y_2, Y_4, \ldots, Y_{10}$ can forecast $\hat{Y}_{12}$ which are shown next.



- **Second step**: The aggregate values are forecasted. From the first step, each of subsequence has its ARIMA model and SMAPE. The forecasting values $\hat{Y}_{11}$ and $\hat{Y}_{12}$ are generated by considering the minimum SMAPE of each subsequence (the details are shown in section 4.4).

- **Third step**: The aggregate forecasted values are disaggregated. After, $\hat{Y}_{11}$ and $\hat{Y}_{12}$ are forecasted, they disaggregate to the original unit by multiplying with their corresponding ratios.



The Figure 4.2 shows the ratio r_1, r_2, r_3, r_4. Example, the forecasting data $\hat{X}_{41}$, $\hat{X}_{42}$, $\hat{X}_{43}$, $\hat{X}_{44}$ can be calculated by applying r_1, r_2, r_3, r_4.

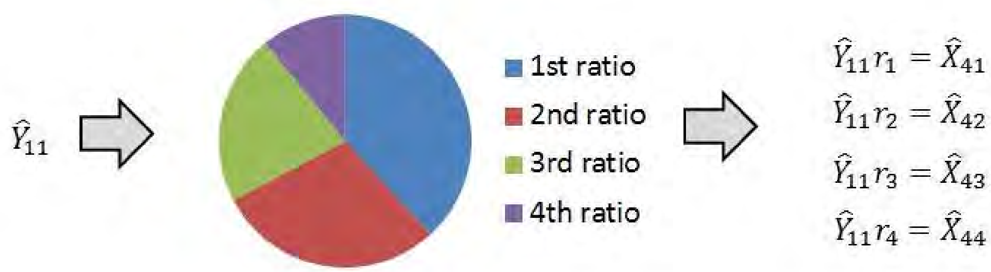


Figure 4.2: The diagram of disaggregate process of $\hat{Y}_{11}$

Resultantly, $\hat{X}_{45}$, $\hat{X}_{46}$, $\hat{X}_{47}$, $\hat{X}_{48}$ can be calculated similarly. The ratios of disaggregation are calculated by the average of the values in the empirical study which the details are shown in section 4.4.

4.3 The MASA model-building algorithm

The MASA model-building algorithm consists of four main parts. The first part is the aggregate of the in-sample data of time series. The second part is to divide the aggregate in-sample data into the training data and the validating data. The third part is to generate the subsequences. The fourth part is to fit the best ARIMA model and calculate SMAPE of each subsequence. The diagram of the MASA model-building algorithm is shown in Figure 4.3.

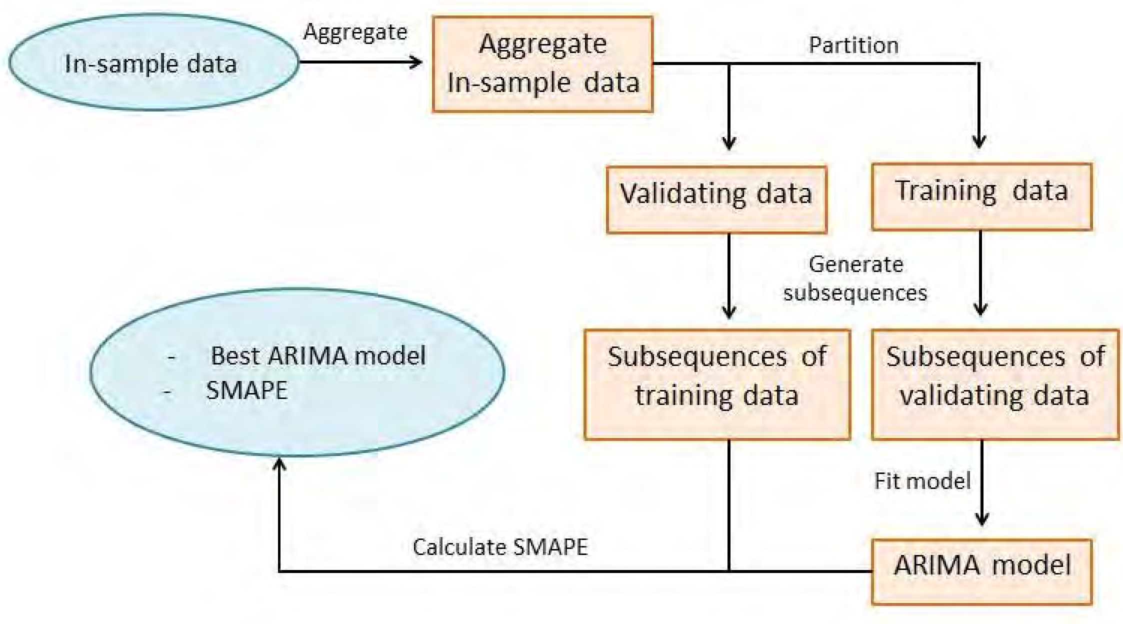


Figure 4.3: The diagram of the MASA model-building algorithm.

The step of the MASA model-building algorithm

- **First step**: The in-sample data is aggregated. Let $Y_1, Y_2, \ldots, Y_T$ be the aggregate in-sample data, l be a period of the time series data and T be a number of aggregate groups on the in-sample data. The aggregation of the time series data is defined as follow:

$$Y_k = \sum_{i=1}^{l} X_{(k-1)l+i}$$

for $k = 1, 2, \ldots, T$ where $N = Tl$.

- **Second step**: The data is partitioned. The aggregate in-sample data is subdivided into the training and the validating time series data.

Let the training time series data be $Y_1, Y_2, \ldots, Y_{T'}$ where T' is the number of the training time series data (T' is less than T) and the validating time series data is $Y_{T'+1}, Y_{T'+2}, \ldots, Y_T$.

- **Third step**: The subsequences of the data are generated. The arithmetic subsequences are generated starting at various initial values.

Let m be the maximum common difference. Thus, a number of arithmetic subsequence patterns is $\sum_{d=1}^{m} d = \frac{m(m+1)}{2}$.

Each subsequence of the aggregate in-sample data is generated as follow:

Common difference d = 1 : the arithmetic subsequence is $(Y_i)_{i=1,2,\ldots,t}^{t \leq T}$.

Common difference d = 2 : the arithmetic subsequences are $(Y_{2i-1})_{i=1,2,\ldots,t}^{2t-1 \leq T}$ and $(Y_{2i})_{i=1,2,\ldots,t}^{2t \leq T}$.

Common difference d = m: there are m arithmetic subsequences with the following form $(Y_{mi-j})_{i=1,2,\ldots,t}^{mt-j \leq T}$ for $j = 1, 2, \ldots, m-1$, respectively.

To generate subsequences of the training time series data, the index of the subsequences start at 1 to T' and the validating time series data starts at $T' + 1$ to T.

- **Fourth step**: The best ARIMA model is fitted. The ARIMA models are fitted from the subsequences of the training time series data and SMAPEs of the validating portion are calculated.

Let $S_{(u,d)}$ be a subsequence of the aggregate in-sample data with the starting index u and the common difference d.

Let $ARIMA_{(u,d)}$ and $SMAPE_{(u,d)}$ be the best fitted ARIMA model of $S_{(u,d)}$ and SMAPE of $S_{(u,d)}$, respectively. This $ARIMA_{(u,d)}$ will be used for forecasting the future values in the second part of the algorithm.

4.4 The MASA model-forecasting algorithm

The MASA model-forecasting algorithm consists of three main parts. The first part is to identify the aggregate group of future time and the subsequence of the aggregate group. The second part is to forecast the aggregate values. The third part is to disaggregate the forecasting values. The diagram of the MASA model-forecasting algorithm is shown in Figure 4.4.

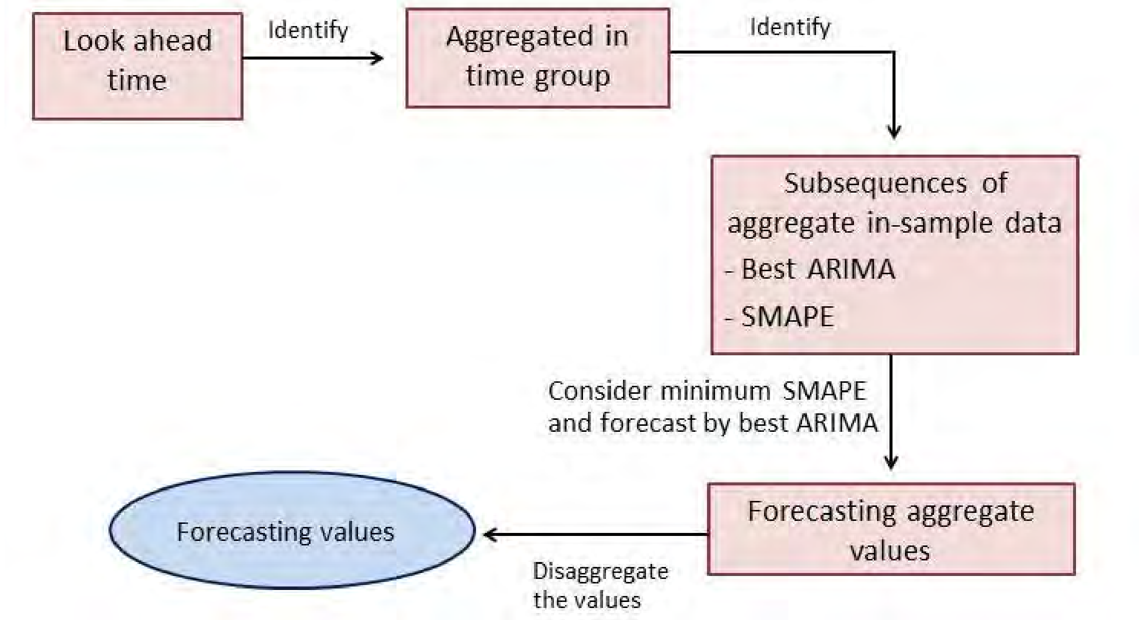


Figure 4.4: The diagram of the MASA model-forecasting algorithm.

The step of the MASA forecasting algorithm

Let $\hat{Y}_{T+1}, \hat{Y}_{T+2}, \ldots, \hat{Y}_{T''}$ be the aggregate forecasted values where T'' is a number of aggregate groups on the time series data and $M = T''l$.

From $N = Tl$, the forecasted future values can be written as $\hat{X}_{Tl+1}$, $\hat{X}_{Tl+2}$, $\ldots$, $\hat{X}_{(T+1)l}$, $\hat{X}_{(T+1)l+1}$, $\ldots$, $\hat{X}_M$.

- **First step**: The aggregate forecasted values are identified from each future value.

$$\begin{array}{ccccc} \hat{X}_{Tl+1}, \cdots, \hat{X}_{(T+1)l} & \hat{X}_{(T+1)l+1}, \cdots, \hat{X}_{(T+2)l} & \cdots & \hat{X}_{(T'-1)l+1}, \cdots, \hat{X}_{(T'l} \\ \text{identify}\Big\Downarrow & \text{identify}\Big\Downarrow & \cdots & \text{identify}\Big\Downarrow \\ \hat{Y}_{T+1} & \hat{Y}_{T+2} & \cdots & \hat{Y}_{T''} \end{array}$$

Then the subsequences of the aggregate in-sample data $S_{(u,d)}$ are identified by $\hat{Y}_{T+1}$, $\hat{Y}_{T+2}$, $\ldots$, $\hat{Y}_{T''}$ follow.

$$
\begin{array}{ccc}
 & \text{Identify to} & \\
\left(\hat{Y}_{i+T}\right)_{i=1,\dots,t}^{t+T\leq T''} & \Longrightarrow & S_{(1,1)} \\
\left(\hat{Y}_{2i-1+T}\right)_{i=1,\dots,t}^{2t-1+T\leq T''} & \Longrightarrow & S_{(1,2)} \\
\left(\hat{Y}_{2i+T}\right)_{i=1,\dots,t}^{2t+T\leq T''} & \Longrightarrow & S_{(2,2)} \\
\vdots & \vdots & \vdots \\
\left(\hat{Y}_{mi-(m-1)+T}\right)_{i=1,\dots,t}^{mt-(m-1)+T\leq T''} & \Longrightarrow & S_{(1,m)} \\
\left(\hat{Y}_{mi-(m-2)+T}\right)_{i=1,\dots,t}^{mt-(m-2)+T\leq T''} & \Longrightarrow & S_{(2,m)} \\
\vdots & \vdots & \vdots \\
\left(\hat{Y}_{mi+T}\right)_{i=1,\dots,t}^{mt+T\leq T''} & \Longrightarrow & S_{(m,m)}
\end{array}
$$

- **Second step**: The aggregate values are forecasted.

The subsequences that contain the index of the aggregate forecasted values are selected. The corresponding ARIMA models and SMAPEs are extracted. The minimum SMAPEs among all ARIMA models of this index is selected to forecast $\hat{Y}_{T+1}$, $\hat{Y}_{T+2}$, $\dots$, $\hat{Y}_{T''}$. The diagram of this step is shown in Figure 4.5.

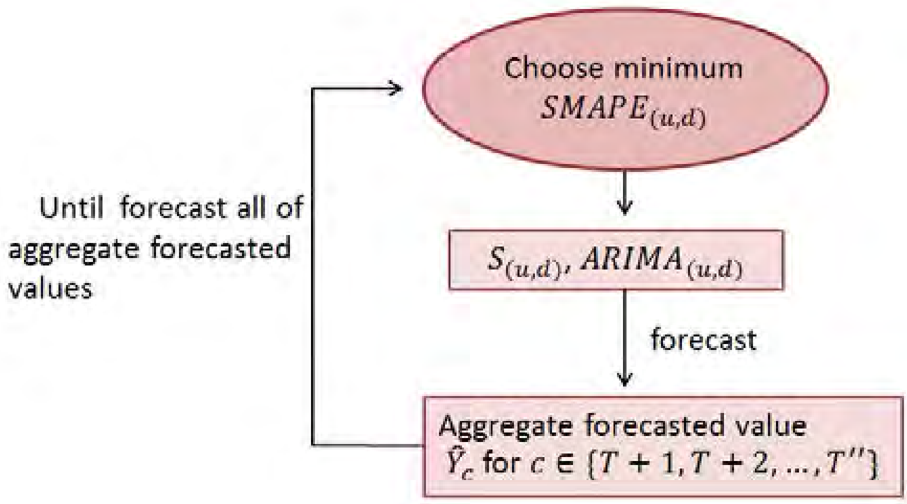


Figure 4.5: The diagram of the forecasting values.

- **Third step**: The aggregate forecasted values are disaggregated.

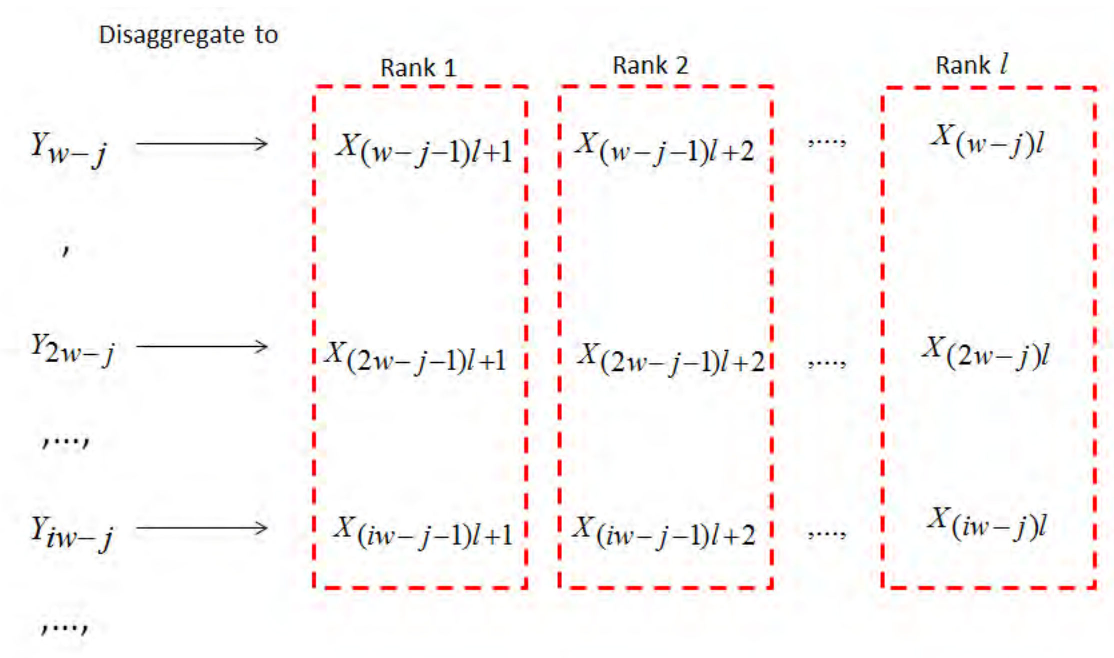
In this step, the aggregate forecasted values $\hat{Y}_{T+1}$, $\hat{Y}_{T+2}$, $\dots$, $\hat{Y}_{T''}$ are disaggregated to $\hat{X}_{N+1}, \hat{X}_{N+2}, \dots, \hat{X}_M$ by the following method.

From $\hat{Y}_c$ for $c \in \{T+1, T+2, \ldots, T''\}$, it is forecasted by the subsequence, $S_{(u,d)} = (Y_{wi-j})_{i=1,2,\ldots,t}^{wt-j \leq T}$ for some $j \in \{0, 1, \ldots, m-1\}$ and $w \in \{1, 2, \ldots, m\}$.

$S_{(u,d)}$ can be disaggregated follow:

$$\begin{array}{ccccc} Y_{w-j} & & Y_{2w-j} & & Y_{iw-j} \\ \Downarrow & & \Downarrow & \cdots & \Downarrow \\ X_{(w-j-1)l+1}, \cdots, X_{(w-j)l} & & X_{(2w-j-1)l+1}, \cdots, X_{(2w-j)l} & & X_{(iw-j-1)l+1}, \cdots, X_{(iw-j)l} \end{array}$$

Next, the ratios that are used to disaggregate the aggregate forecasted values $\hat{Y}_{T+1}$, $\hat{Y}_{T+2}$, $\ldots$, $\hat{Y}_{T''}$ to the original unit are calculated. Firstly, the ratios of disaggregate from each column rank are defined as the average column ranks which are shown in the following figure.



Hence, the average ratio can be written as

$$r_a = \frac{\displaystyle\sum_{i=1}^{\left\lfloor \frac{T+j}{w} \right\rfloor} \frac{X_{(iw-j-1)l+a}}{Y_{iw-j}}}{\left\lfloor \frac{T+j}{w} \right\rfloor}$$

for $a = 1, 2, \ldots, l, j \in \{0, 1, \ldots, m-1\}, w \in \{1, 2, \ldots, m\}$ and $\left\lfloor \dfrac{T+j}{w} \right\rfloor$ is the length of the value X in the rank a.

Finally, the average ratios are weighted by the summation of all average ratios.

Let $\bar{r}_a$ be a weighted average ratio of value of rank a,

$$\bar{r}_a = \frac{r_a}{\sum_{i=1}^{l} r_i}$$

for $a = 1, 2, \ldots, l$

Therefore, $\hat{Y}_c$ can be disaggregated to $\hat{X}_{(c-1)l+1}, \hat{X}_{(c-1)l+2}, \ldots, \hat{X}_{(c-1)l+l}$ using

$$\hat{X}_{(c-1)l+a} = \bar{r}_a \hat{Y}_c$$

for $a = 1, 2, \ldots, l$ and $c \in \{T+1, T+2, \ldots, T''\}$.

4.5 The model measurements

This section introduces the measurement of the model that uses in this research. It comprises of two parts. The first part is the mean absolute percentage error and the symmetric mean absolute percentage error. The second part is the Wilcoxson signed-rank test.

4.5.1 Mean absolute percentage error and symmetric mean absolute percentage error

The general measurement used in the time series is the mean absolute percentage error or MAPE. Let N be the number of forecast data. It is defined by

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|X_i - \hat{X}_i|}{|X_i|}$$

The symmetric mean absolute percentage error or SMAPE, it is defined by

$$SMAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{|X_i - \hat{X}_i|}{\frac{|X_i| + |\hat{X}_i|}{2}}$$

where and X_t, $\hat{X}_t$ are the out-sample data and forecast data, respectively.

This research uses only SMAPE because the data come from the NN5 dataset. The NN5 dataset uses only SMAPE to measure the accuracy.

4.5.2 Wilcoxson signed-rank test

In this research, the Wilcoxon signed-rank test is used for measuring the relation between the data. This section introduces the basic of the Wilcoxon signed-rank test. It is introduced by Wilcoxon (1945).The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test. It is used for comparing two relations of samples or match of samples.

4.5.2.1 Assumption of the Wilcoxon signed-rank test

1. The data is paired and it comes from the same population.
2. The pair of the data is chosen randomly and independently.
3. The data is an ordinal scale.

4.5.2.2 Procedure of Wilcoxon signed-rank test

The hypothesis test can be defined as follow :

H_0: difference between the pairs follows a symmetric distribution around zero.

H_1: difference between the pairs does not follow a symmetric distribution around zero.

Let N be the sample size. Thus, there are a total of $2N$ values. For each pair $i = 1, 2, \ldots N$, let $x_{1,i}$ and $x_{2,i}$ be the data.

Step 1) Calculate $|x_{2,i} - x_{1,i}|$, for $i = 1, 2, \ldots N$, and then calculate $sgn(x_{2,i} - x_{1,i})$, where sgn is the sign function.

Step 2) Remove pairs with $|x_{2,i} - x_{1,i}| = 0$ and let N_r be the sample size that is reduced .

Step 3) Order the N_r pairs from the smallest absolute difference to the largest absolute difference, $|x_{2,i} - x_{1,i}|$.

Step 4) Rank the pairs by starting the smallest as 1 where R_i be the rank.

Step 5) Calculate the test statistic W which is the sum of the signed ranks as

$$W = \sum_{i=1}^{N_r} [sgn(x_{2,i} - x_{1,i}) R_i]$$

Step 6) W is compared to a critical value from a reference table in Lowry (2014). The two-sided test consists in rejecting H_0, if $|W| \leq W_{critical,N_r}$. P-value can also be calculated. If p-value $\leq$ 0.05 then reject H_0 by significant level 0.05.

After, the methodology of this research is introduced. The research uses this methodology to generate the MASA model. The NN5 dataset is applied to the MASA model. Next, the empirical results and the comparative analysis are illustrated in chapter 5.

CHAPTER V

EMPIRICAL RESULTS AND COMPARATIVE ANALYSIS

To measure the performance of this approach, the MASA algorithm will be tested on the NN5 dataset. The NN5 collection consists of 111 daily time series data. R programming language is used for implementing the MASA algorithm. This chapter contains the empirical results and the comparative analysis on train:test ratios as 95:5 and 90:10.

5.1 Empirical results

Figure 5.1 shows the time series plot of cash in ATM ID NN5-030 which is one of 111 time series data. It starts from 1996/03/18 to 1998/03/22 having the number of observations equal to 735. From the Figure 5.1, the structure of the time series is sway thus the time series data is reduced the sway by aggregate technique in the MASA algorithm. In this research, the daily time series data is aggregated to weekly time series data therefore the period of aggregate is set to 7.

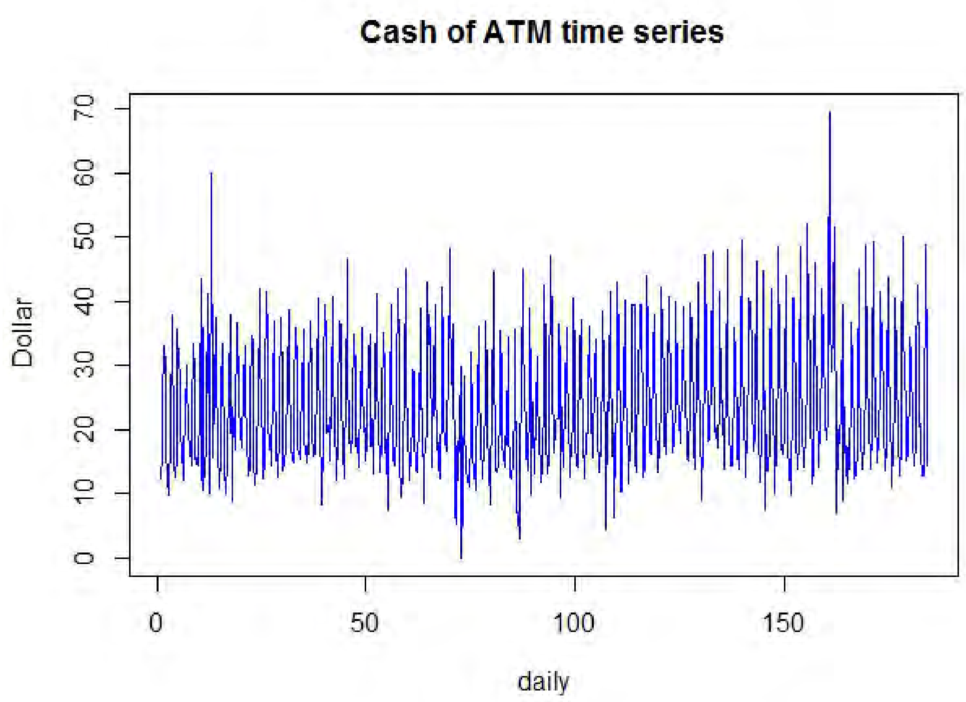


Figure 5.1: Time series plot of ATM ID NN5-030 (1996/03/18 to1998/03/22)

5.1.1 Case I : in-sample data 95% and out-sample data 5%.

This section, the data is set to 95% for the in-sample data and 5% for the out-sample data. A period of aggregate is also set to 7 (weekly). The aggregate time series data from daily to weekly is shown in Figure 5.2. The maximum common difference is set to 4 thus the number of subsequences equals to 10. The time series plots of each aggregate subsequence are shown in Figure 5.3 - 5.12. The parameters and the SMAPE of the MASA model are given in Table 5.1. The subsequence column in Table 5.1 comes from the original sequence Y_i based on the arithmetic formula and the parameter column refers to the ARIMA parameters (p, d, q) of the best ARIMA model to fit the corresponding subsequence where p is the order of autoregressive model, d is the order of differences of the series, and q is the order of the moving average model. The last column keeps the SMAPE of the best ARIMA model on the validating data. The order of SMAPE from the lowest to the highest SMAPEs are 6th, 8th, 2nd, 1st, 4th, 9th, 3rd, 7th, 10th, and 5th which are shown in Table 5.1.

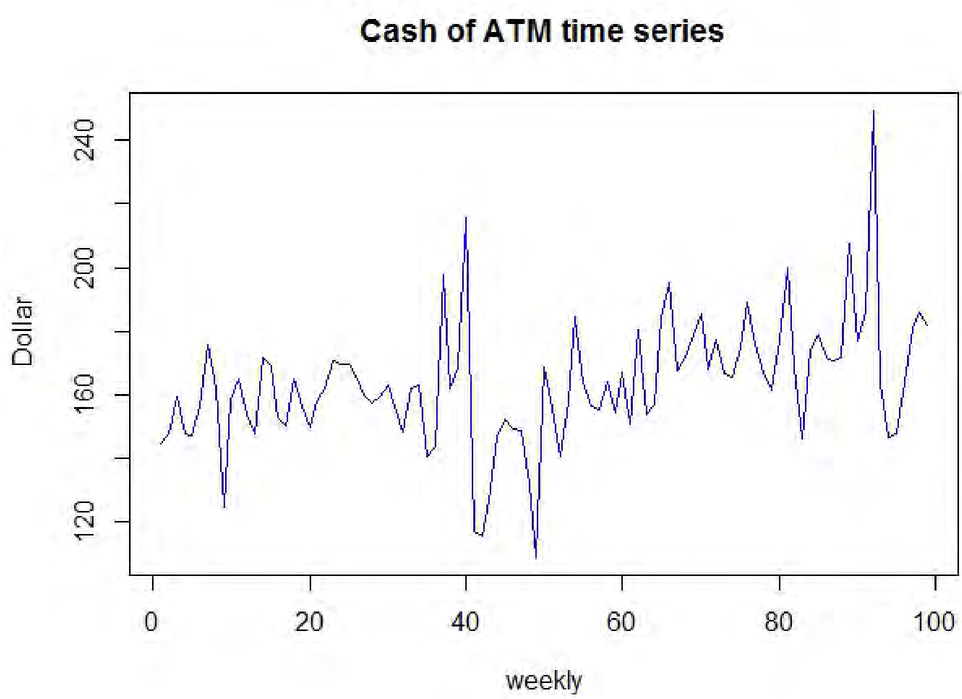


Figure 5.2: Time series plot of aggregate time from daily to weekly for in-sample 95%(ATM ID NN5-030, 1996/03/18 to1998/03/22)

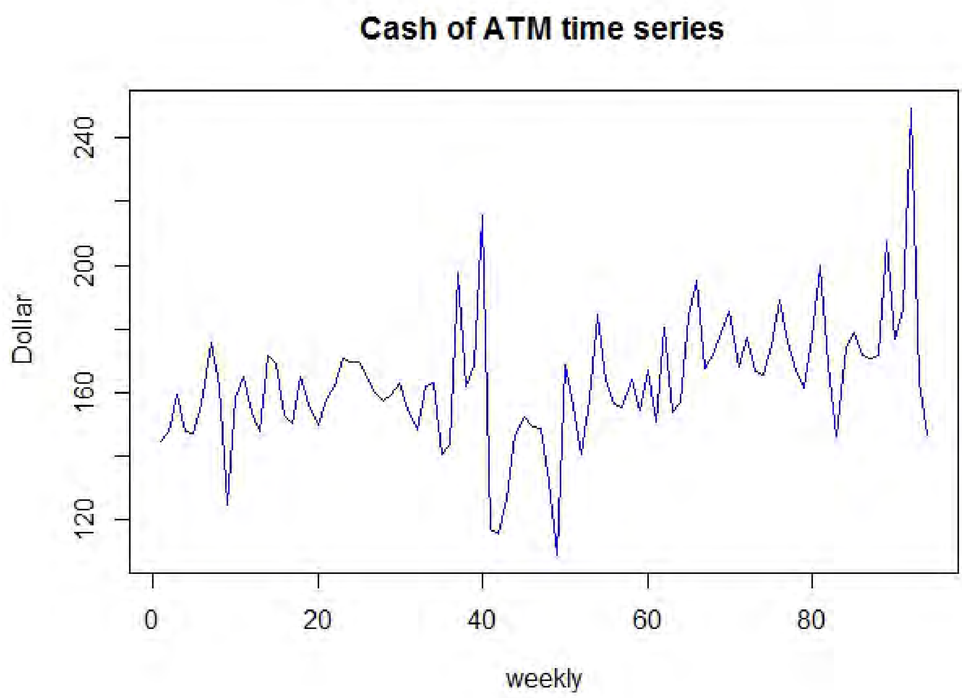


Figure 5.3: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_2, Y_3, $Y_4, \ldots$ (in-sample 95%)

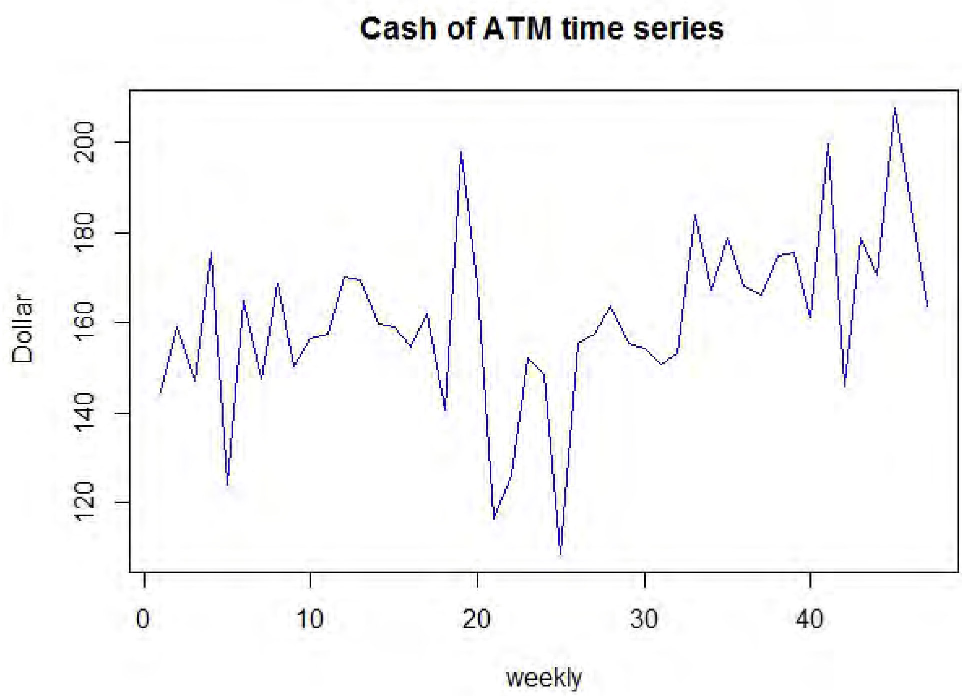


Figure 5.4: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_3, Y_5, $Y_7, \ldots$ (in-sample 95%)

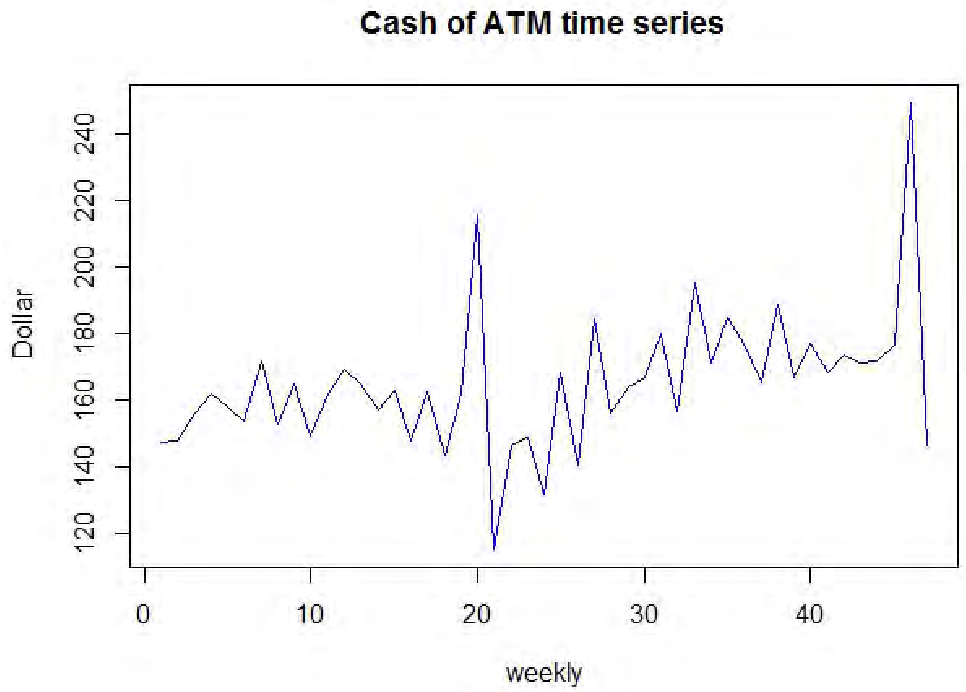


Figure 5.5: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_4, Y_6, $Y_8, \ldots$ (in-sample 95%)

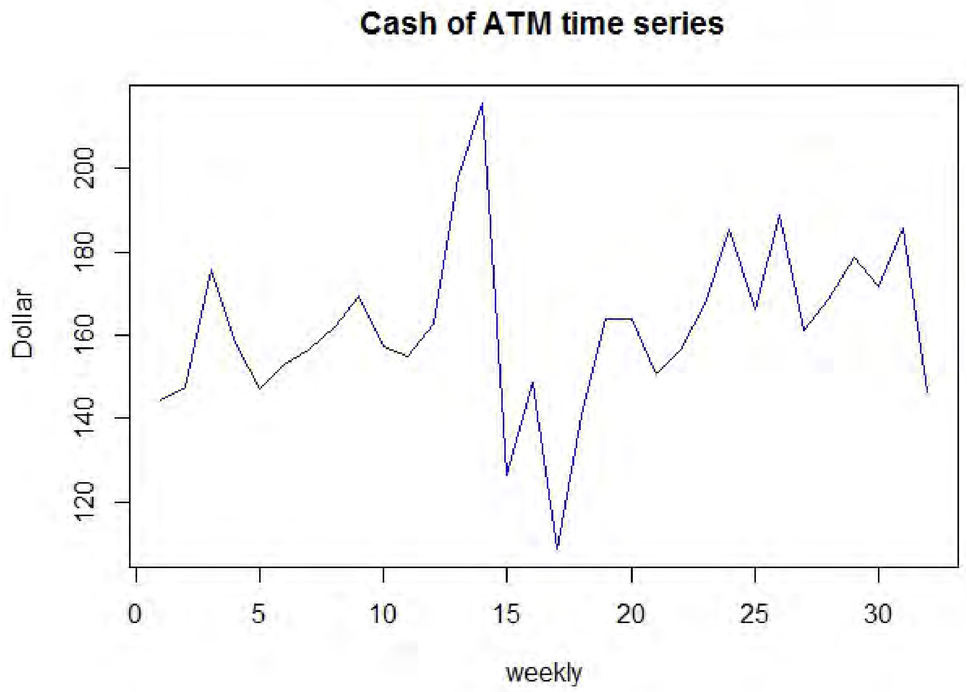


Figure 5.6: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_4, Y_7, $Y_{10}, \ldots$ (in-sample 95%)

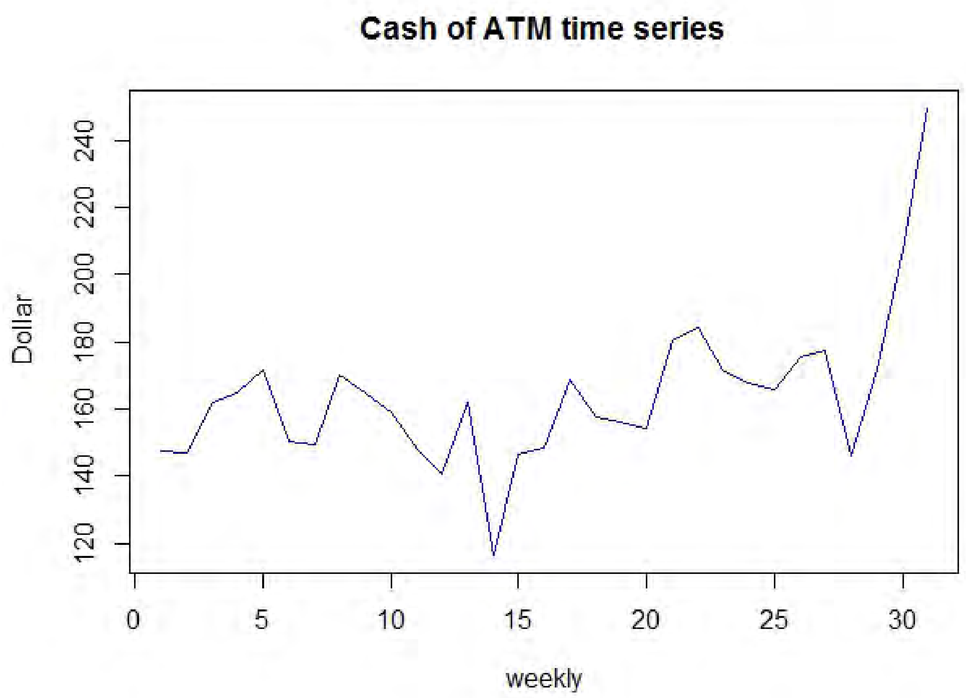


Figure 5.7: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_5, Y_8, $Y_{11}, \ldots$ (in-sample 95%)

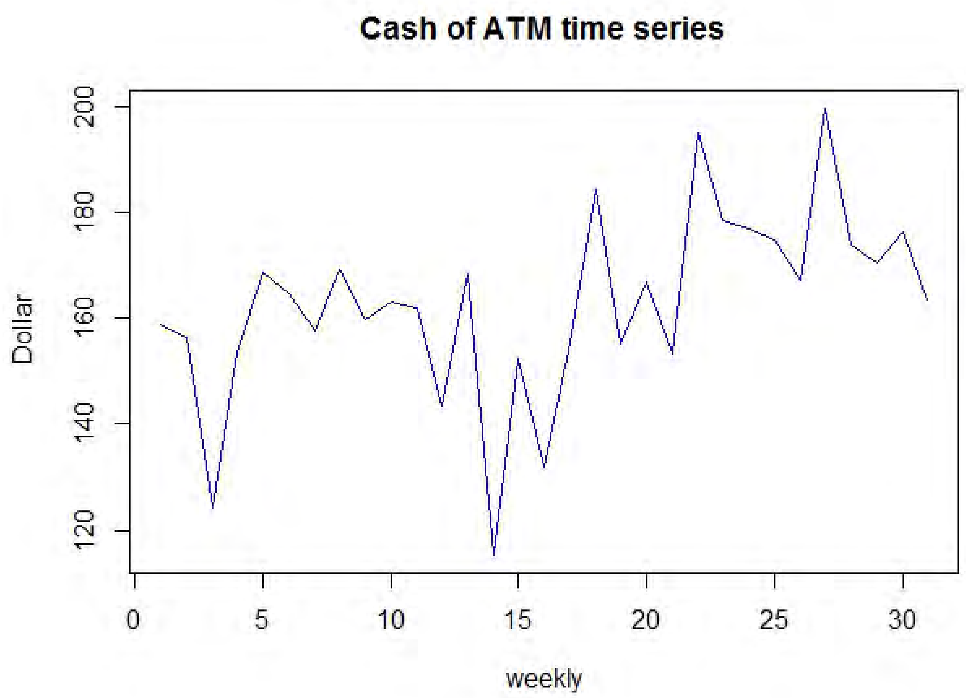


Figure 5.8: Time series plot of aggregate time from daily to weekly for subsequence Y_3, Y_6, Y_9, $Y_{12}, \ldots$ (in-sample 95%)

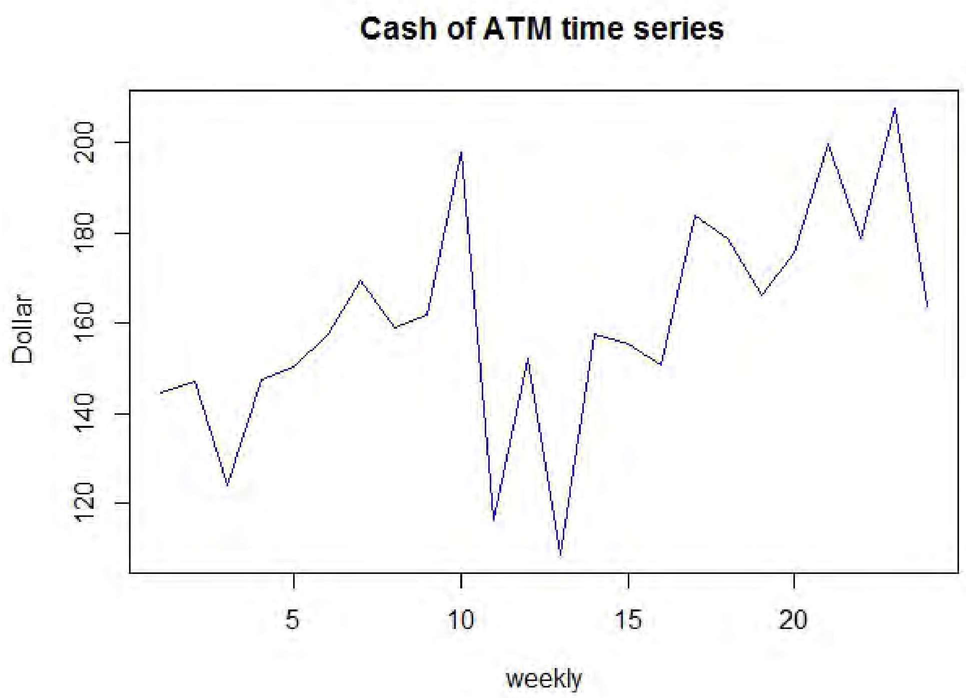


Figure 5.9: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_5, Y_9, $Y_{13}, \ldots$ (in-sample 95%)

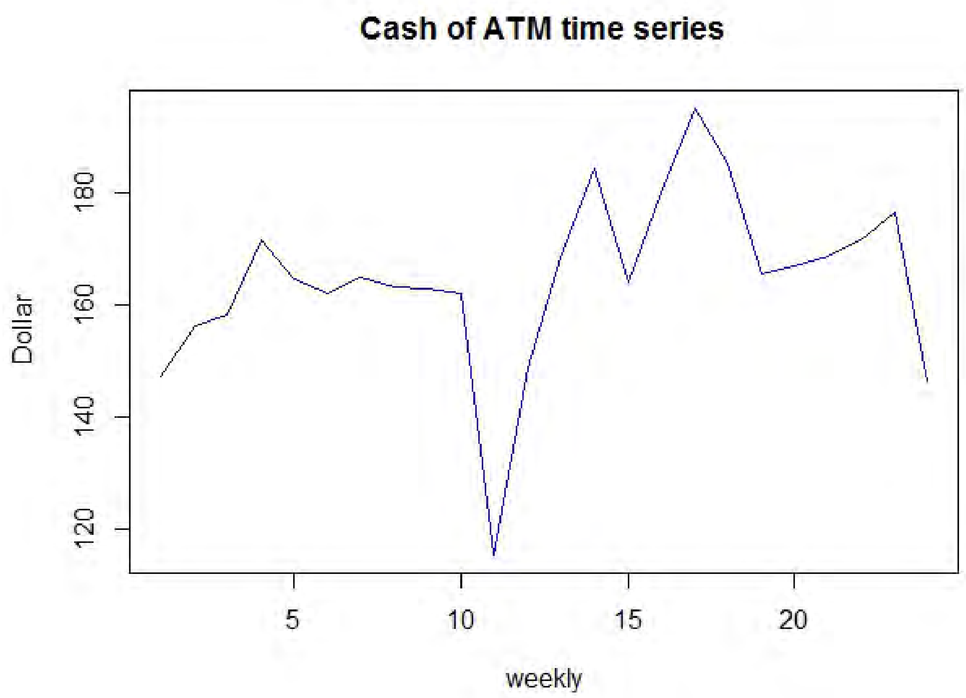


Figure 5.10: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_6, Y_10, $Y_{14}, \ldots$ (in-sample 95%)

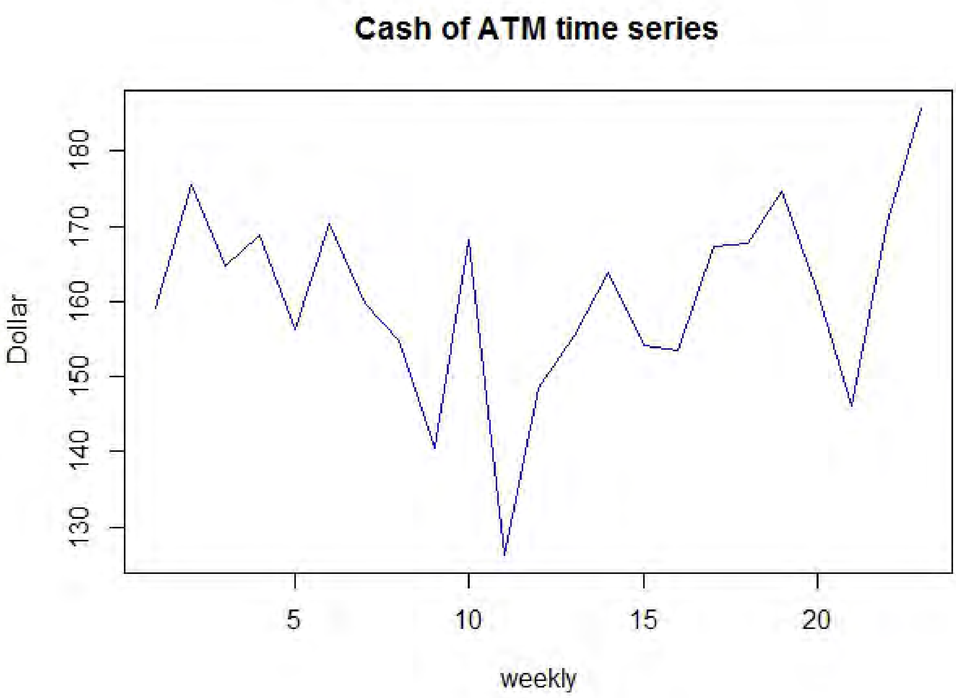


Figure 5.11: Time series plot of aggregate time from daily to weekly for subsequence $Y_3, Y_7, Y_{11}, Y_{15}, \ldots$ (in-sample 95%)

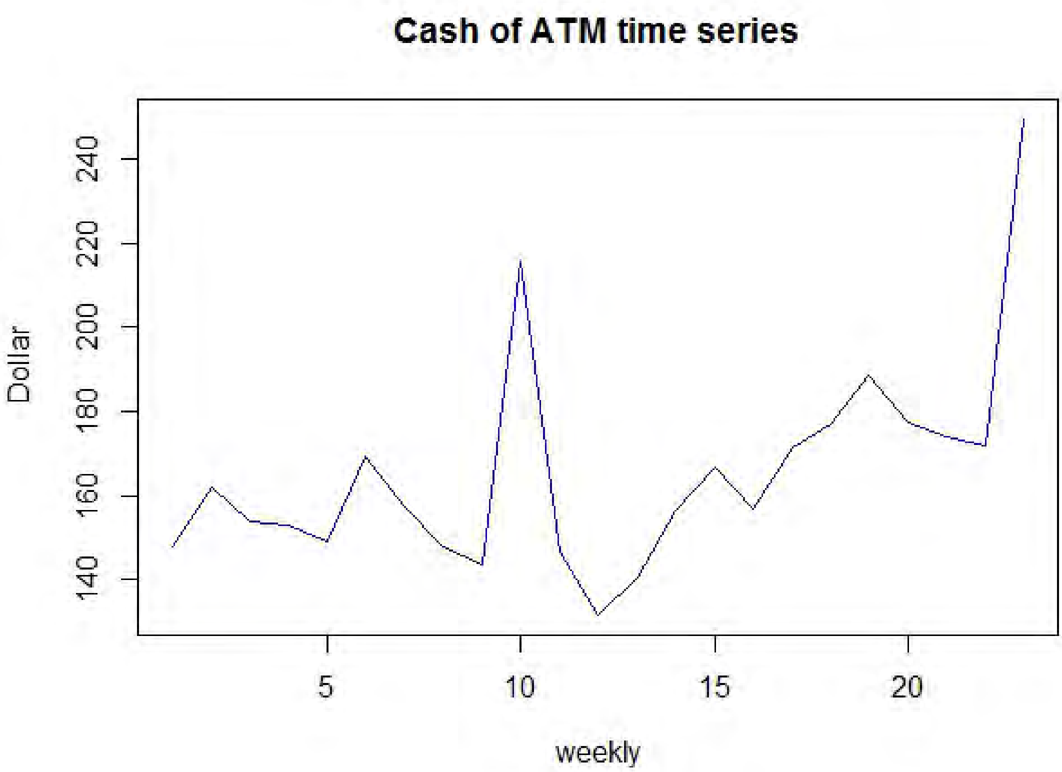


Figure 5.12: Time series plot of aggregate time from daily to weekly for subsequence $Y_4, Y_8, Y_{12}, Y_{16}, \ldots$ (in-sample 95%)

Figure 5.13 shows the empirical results of the MASA model with the out-sample data. There are discrepancies between the results of the MASA model and the out-sample data. The SMAPE of the MASA model from the result is 15.26072.

No.	Subsequence	Parameters(p, d, q)	SMAPE
1	$Y_1, Y_2, Y_3, Y_4, \ldots$	(2,1,3)	7.940902
2	$Y_1, Y_3, Y_5, Y_7, \ldots$	(0,1,1)	7.749427
3	$Y_2, Y_4, Y_6, Y_8, \ldots$	(1,1,0)	11.749961
4	$Y_1, Y_4, Y_7, Y_{10}, \ldots$	(0,0,0)	11.502862
5	$Y_2, Y_5, Y_8, Y_{11}, \ldots$	(0,1,0)	36.407923
6	$Y_3, Y_6, Y_9, Y_{12}, \ldots$	(0,1,1)	4.492906
7	$Y_1, Y_5, Y_9, Y_{13}, \ldots$	(1,1,0)	15.608287
8	$Y_2, Y_6, Y_{10}, Y_{14}, \ldots$	(1,0,0)	5.126374
9	$Y_3, Y_7, Y_{11}, Y_{15}, \ldots$	(0,0,1)	11.727266
10	$Y_4, Y_8, Y_{12}, Y_{16}, \ldots$	(0,1,0)	29.215386

Table 5.1: Output from the MASA algorithm for in-sample 95%

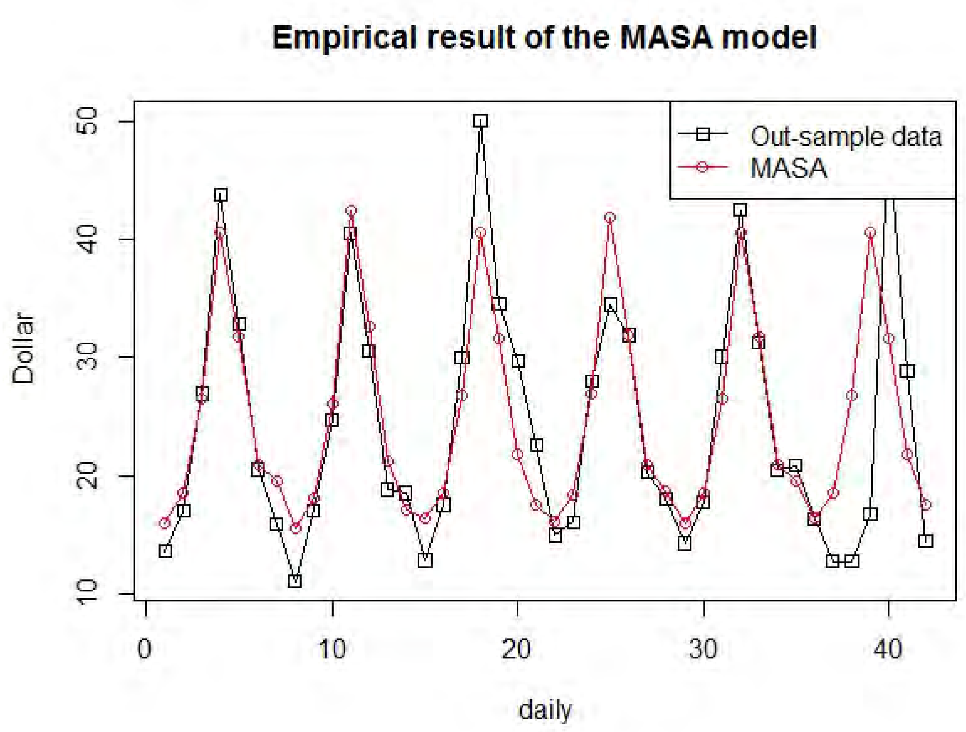


Figure 5.13: The empirical result of the MASA model with the out-sample data (in-sample 95%).

5.1.2 Case II : in-sample data 90% and out-sample data 10%.

This section, the data is set to 90% for the in-sample data and 10% for the out-sample data. A period of aggregate is set to 7 (weekly). The aggregate time series data from daily to weekly is shown in Figure 5.14. The sway of time series is reduced. The maximum common difference is set to 4 thus the number of subsequences equals to 10. The time series plots of each aggregate subsequence are shown in Figure 5.15 - 5.24. The parameters and the SMAPE of the MASA model are given in Table 5.2. The subsequence column in Table 5.2 comes from the original sequence Y_i based on the arithmetic formula and the parameter column refers to the ARIMA parameters (p, d, q) of the best ARIMA model to fit the corresponding subsequence where p is the order of autoregressive model, d is the order of differences of the series, and q is the order of the moving average model. The last column keeps the SMAPE of the best ARIMA model on the validating data. The order from the lowest SMAPE to the highest SMAPE are 8th, 6th, 1st, 4th, 3rd, 9th, 2nd, 7th, 10th and 5th which are shown in Table 5.2.

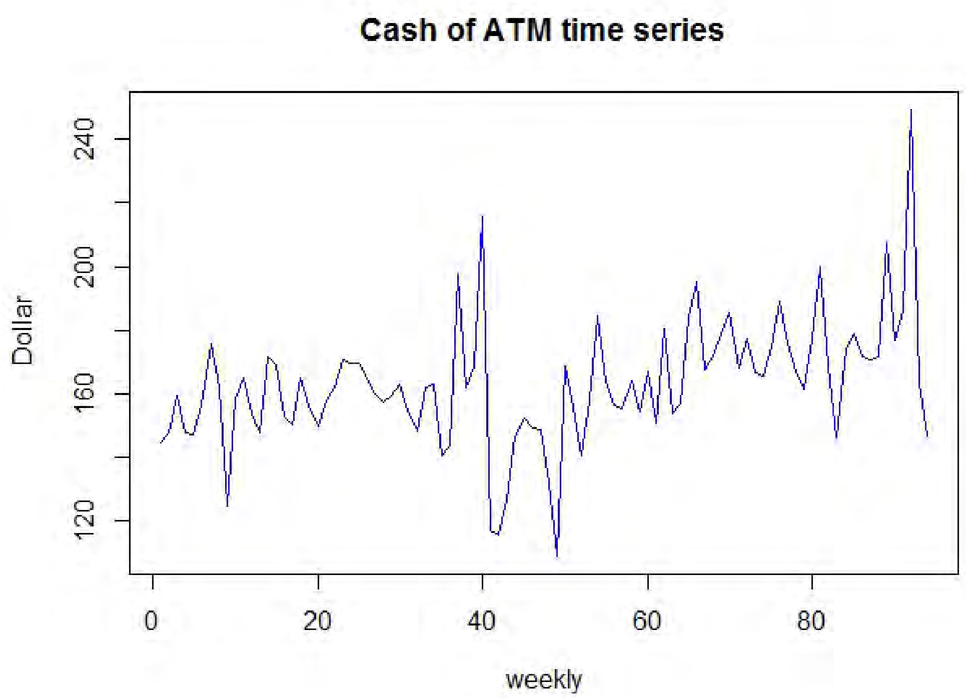


Figure 5.14: Time series plot of aggregate time from daily to weekly for in-sample 90%(ATM ID NN5-030, 1996/03/18 to1998/03/22)

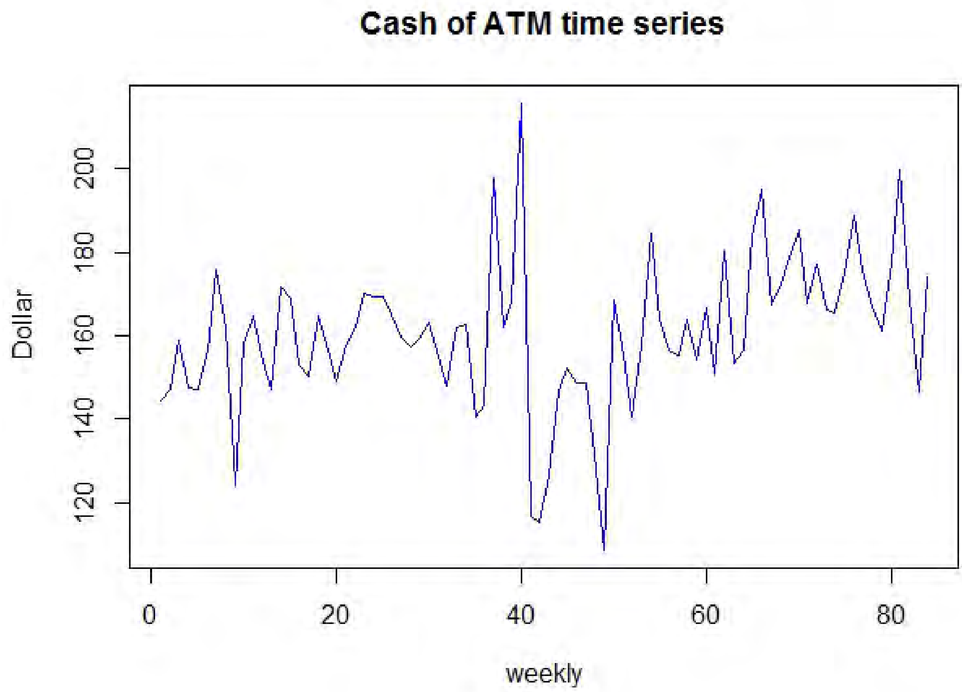


Figure 5.15: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_2, Y_3, $Y_4, \ldots$ (in-sample 90%)

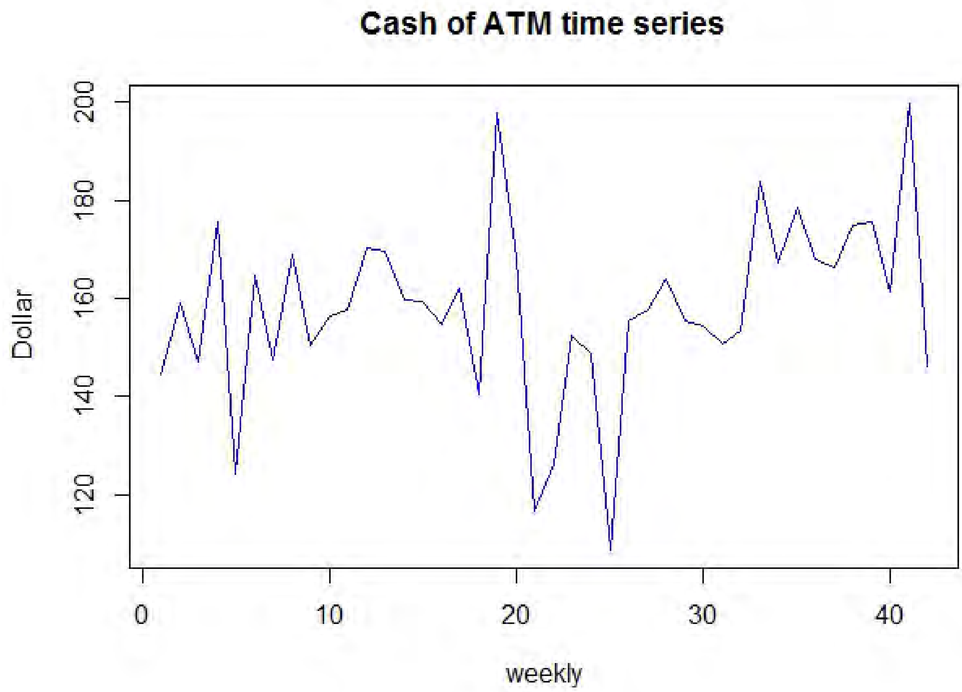


Figure 5.16: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_3, Y_5, $Y_7, \ldots$ (in-sample 90%)

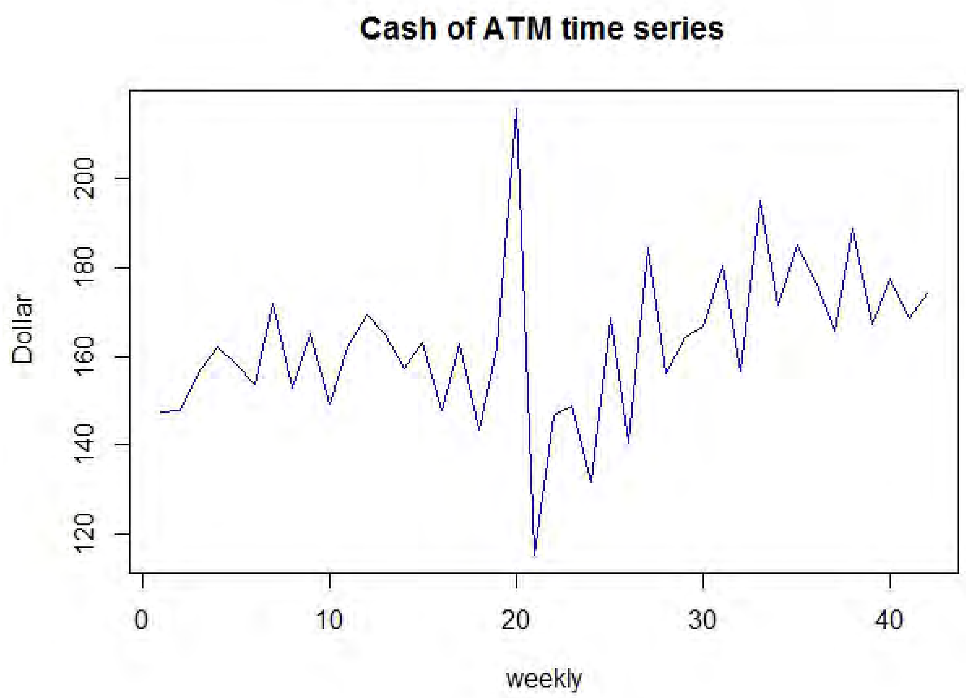


Figure 5.17: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_4, Y_6, $Y_8, \ldots$ (in-sample 90%)

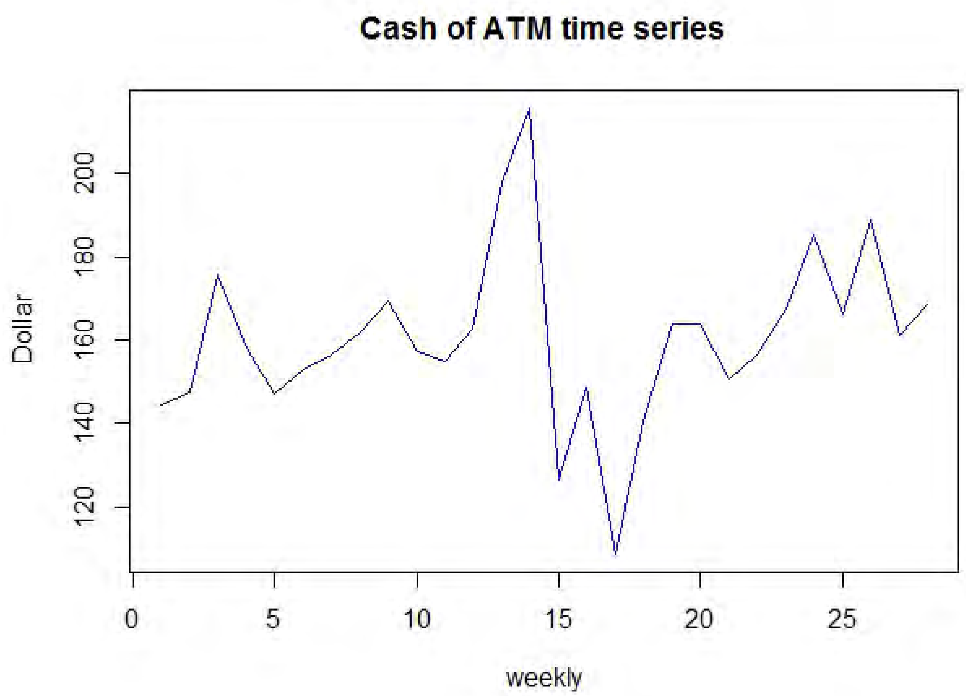


Figure 5.18: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_4, Y_7, $Y_{10}, \ldots$ (in-sample 90%)

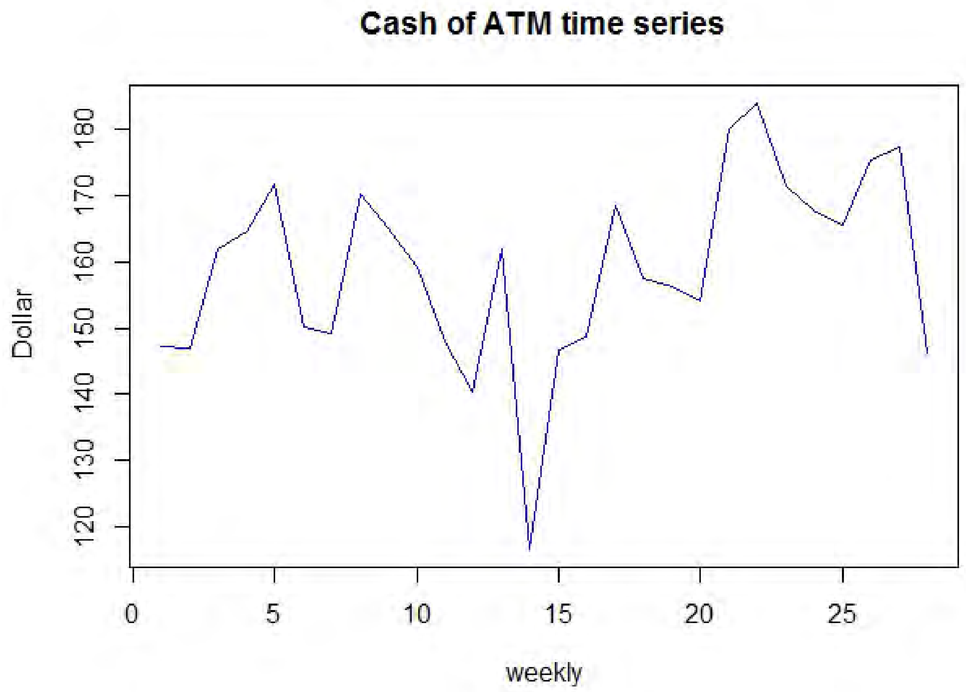


Figure 5.19: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_5, Y_8, $Y_{11}, \ldots$ (in-sample 90%)

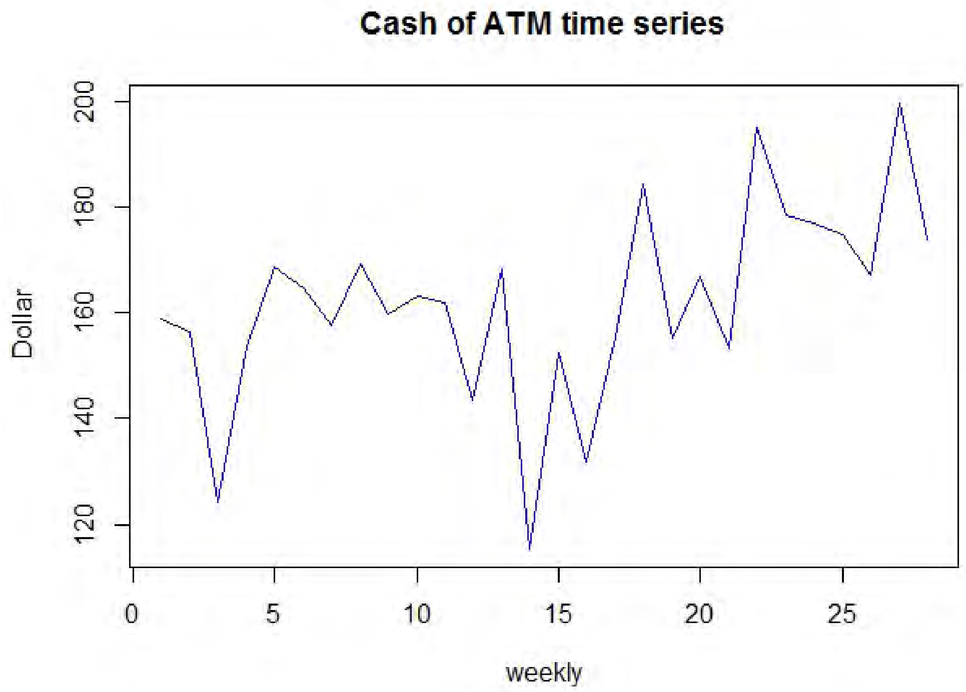


Figure 5.20: Time series plot of aggregate time from daily to weekly for subsequence Y_3, Y_6, Y_9, $Y_{12}, \ldots$ (in-sample 90%)

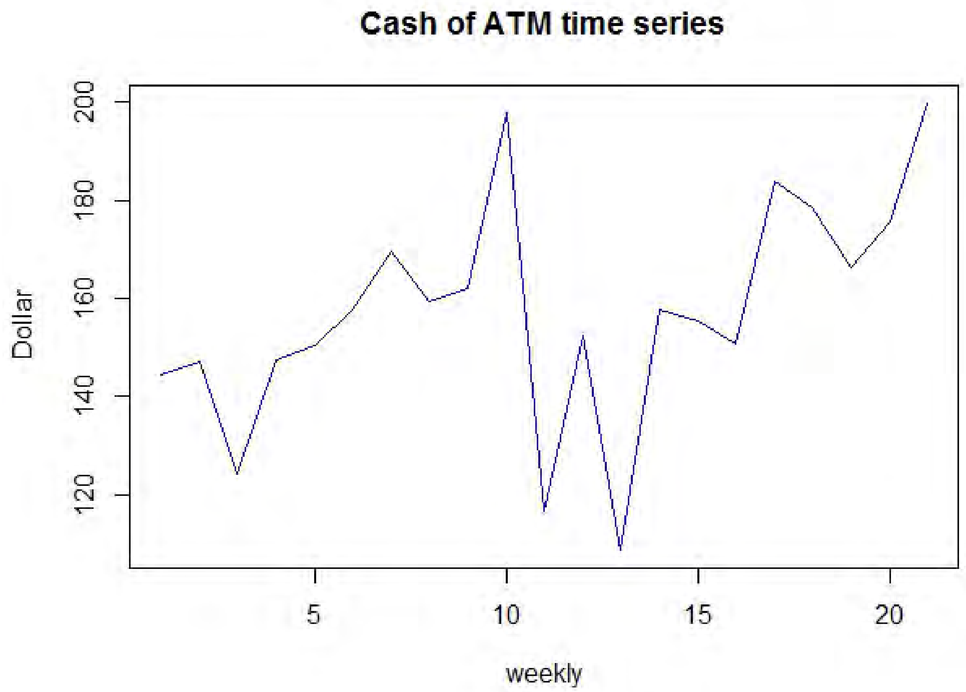


Figure 5.21: Time series plot of aggregate time from daily to weekly for subsequence Y_1, Y_5, Y_9, $Y_{13}, \ldots$ (in-sample 90%)

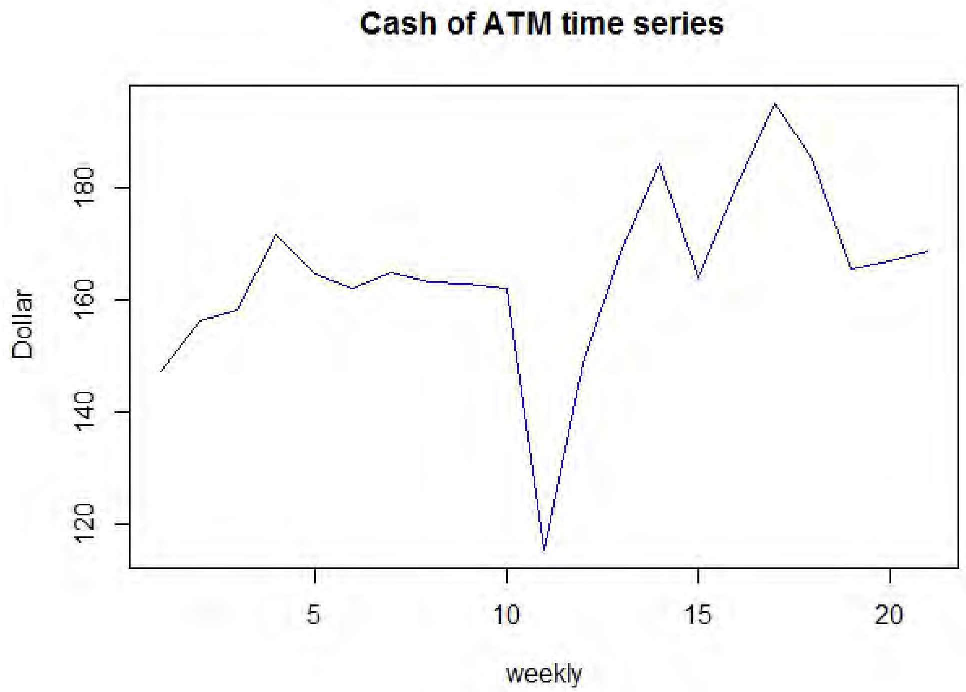


Figure 5.22: Time series plot of aggregate time from daily to weekly for subsequence Y_2, Y_6, Y_10, $Y_{14}, \ldots$ (in-sample 90%)

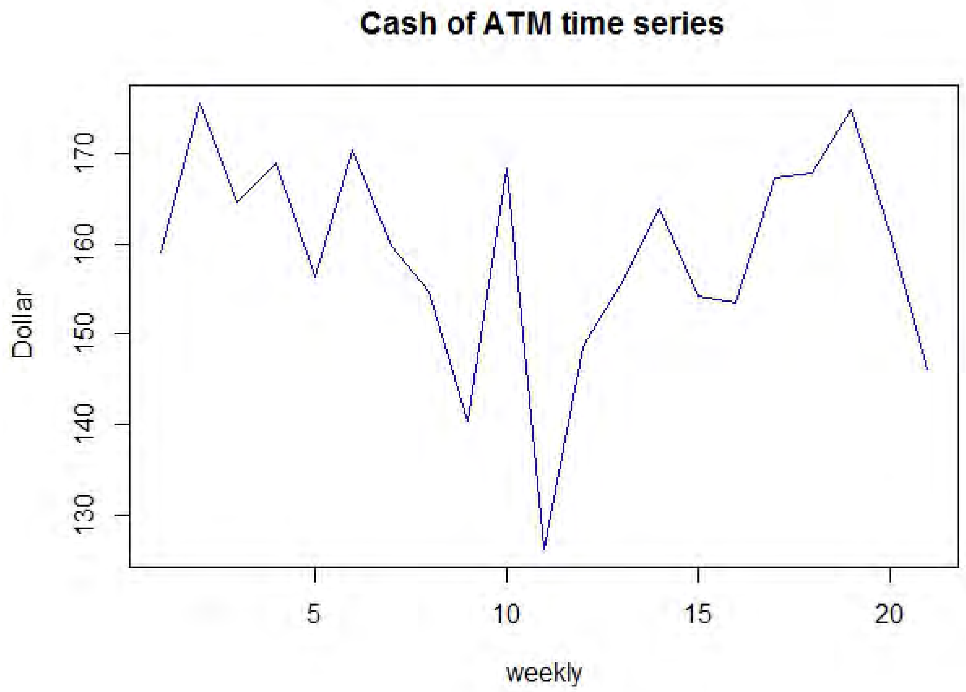


Figure 5.23: Time series plot of aggregate time from daily to weekly for subsequence Y_3, Y_7, Y_{11}, $Y_{15}, \ldots$ (in-sample 90%)

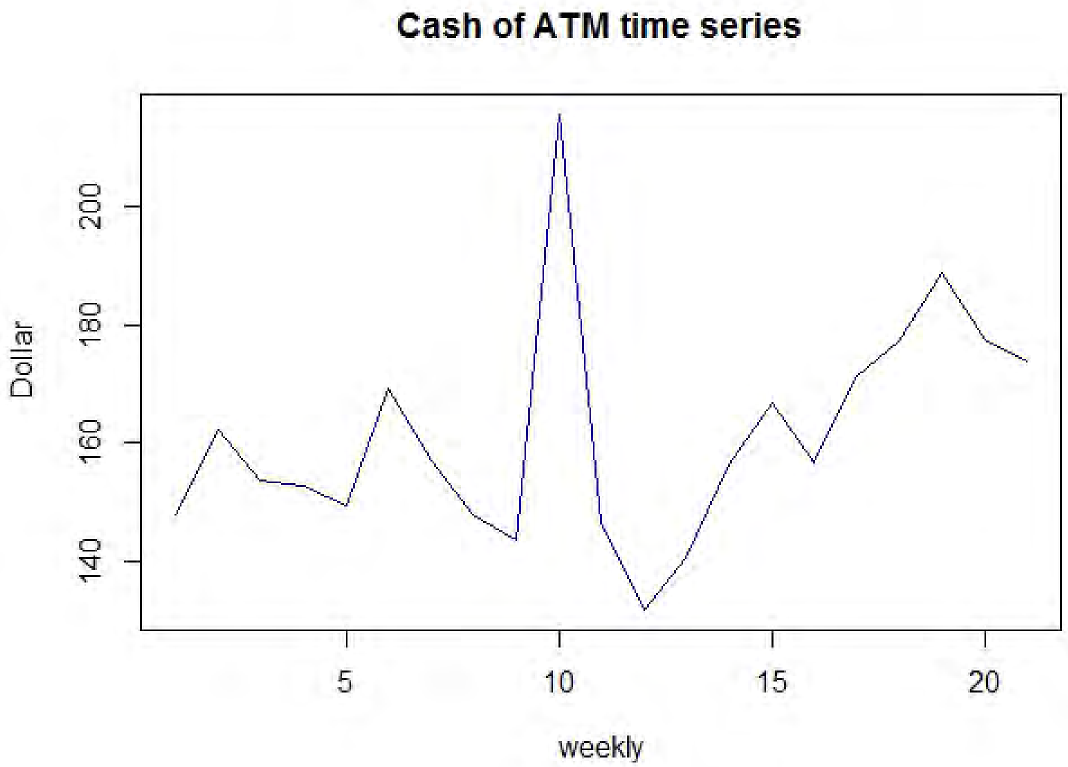


Figure 5.24: Time series plot of aggregate time from daily to weekly for subsequence Y_4, Y_8, Y_{12}, $Y_{16}, \ldots$ (in-sample 90%)

Figure 5.25 shows the empirical results of the MASA model with the out-sample data having the similar distribution. The SMAPE of the MASA model from the result is 14.93942.

No.	Subsequences	Parameters (p, d, q)	SMAPE
1	$Y_1, Y_2, Y_3, Y_4, \ldots$	(1,1,4)	8.896227
2	$Y_1, Y_3, Y_5, Y_7, \ldots$	(0,0,0)	13.240900
3	$Y_2, Y_4, Y_6, Y_8, \ldots$	(2,1,0)	11.116658
4	$Y_1, Y_4, Y_7, Y_{10}, \ldots$	(0,0,0)	10.229235
5	$Y_2, Y_5, Y_8, Y_{11}, \ldots$	(1,0,0)	27.547931
6	$Y_3, Y_6, Y_9, Y_{12}, \ldots$	(1,1,0)	8.401077
7	$Y_1, Y_5, Y_9, Y_{13}, \ldots$	(0,0,0)	14.687281
8	$Y_2, Y_6, Y_{10}, Y_{14}, \ldots$	(1,0,0)	7.195037
9	$Y_3, Y_7, Y_{11}, Y_{15}, \ldots$	(0,0,0)	11.313391
10	$Y_4, Y_8, Y_{12}, Y_{16}, \ldots$	(0,0,0)	24.584056

Table 5.2: Output from the MASA algorithm for in-sample 90%

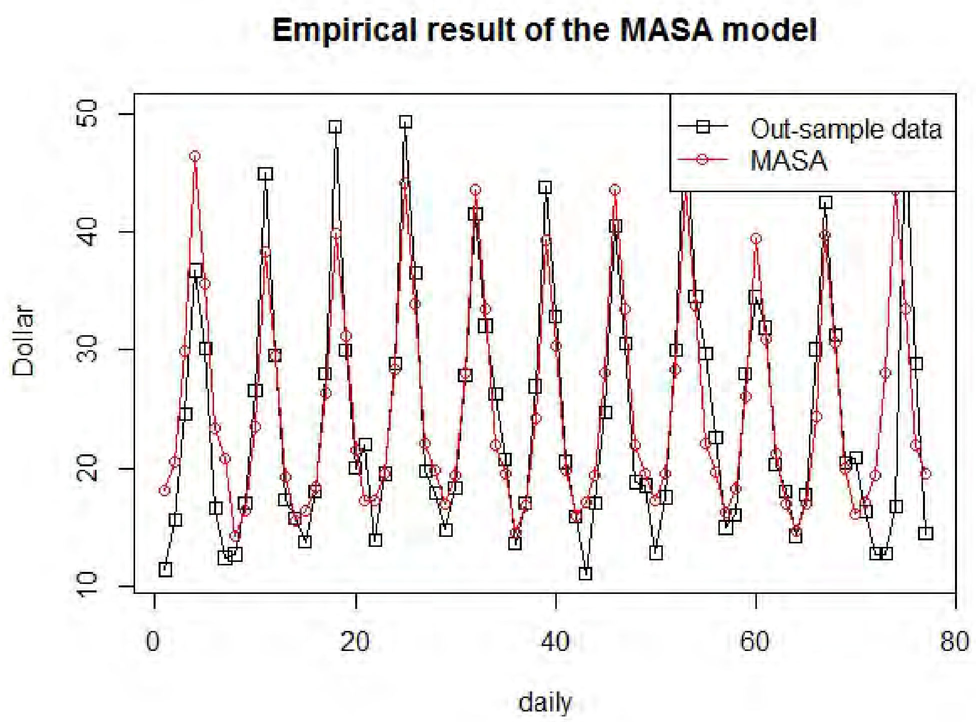


Figure 5.25: The empirical result of the MASA model with the out-sample data (in-sample 90%).

5.2 Comparative analysis

This research focuses on the accuracy of the forecasting model with respect to the out-sample data by using the SMAPE. The performance of this approach is demonstrated by comparing it with the SARIMA (Seasonal Autoregressive Integrated Moving Average) model and the exponential smoothing model.

5.2.1 Case I : in-sample data 95% and out-sample data 5%.

Figure 5.26 and 5.27 show the comparative models of the ATM ID NN5-030 time series data. The forcasted data of the SARIMA model is not much different from that of the MASA model. Moreover, the forecasted values of the SARIMA model converge to a single value. However, the behavior of the forecast data from the MASA model and the exponential smoothing model is quite similar.

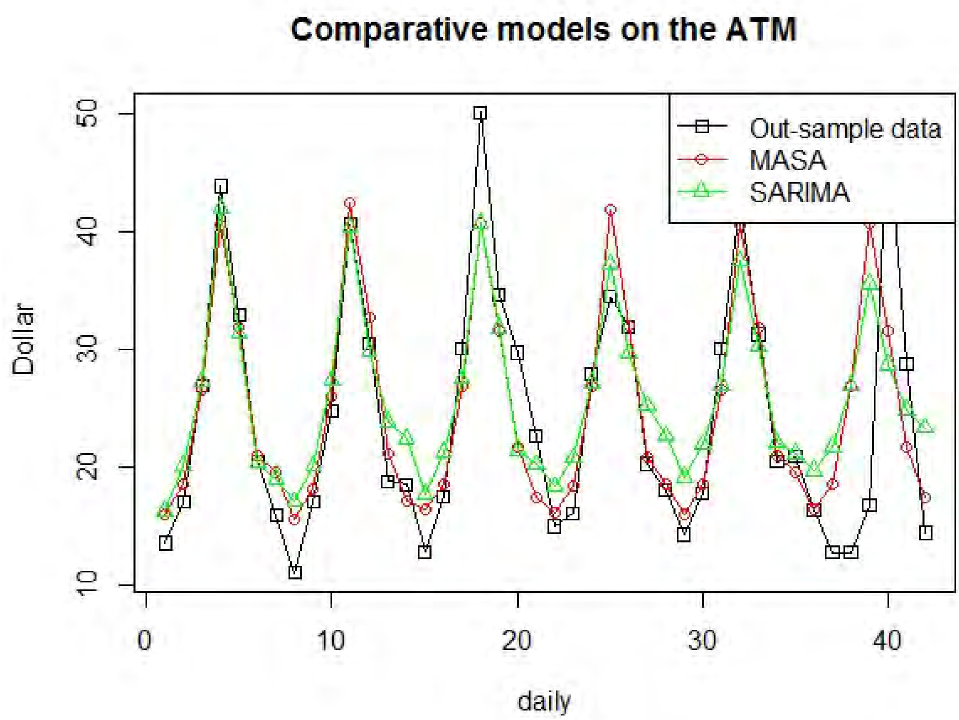


Figure 5.26: Comparative results of the MASA model and the SARIMA model for in-sample 95% (ATM ID NN5-030)

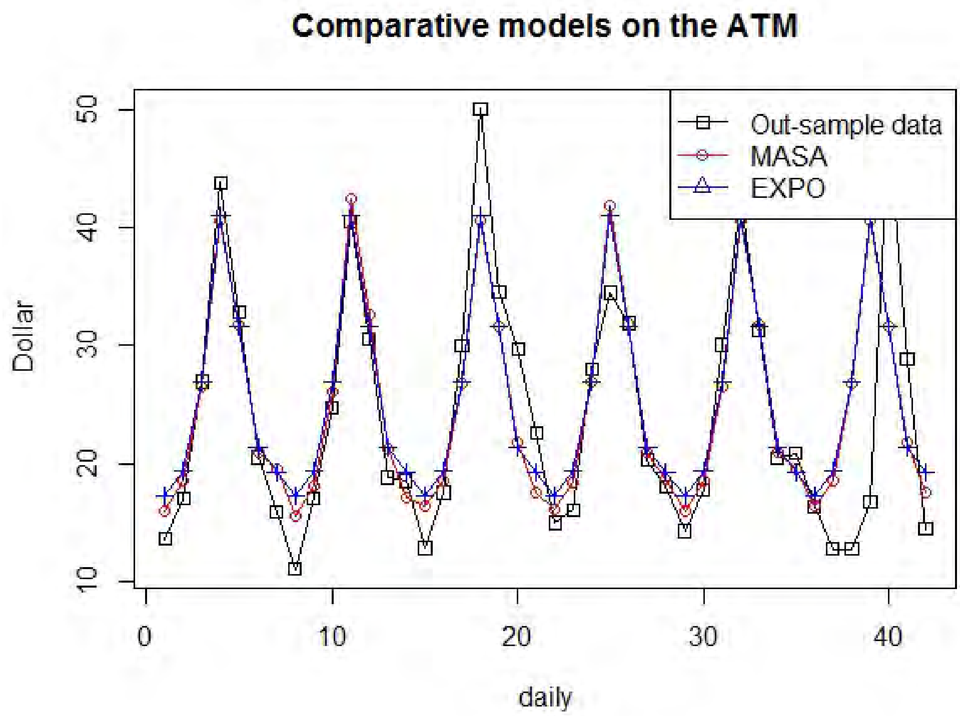


Figure 5.27: Comparative results of the MASA model and the exponential smoothing model for in-sample 95% (ATM ID NN5-030)

	MASA model	SARIMA model	Exponential smoothing model
Parameters of the model	Table 5.2	$p = 1, d = 1, q = 2,$ $P = 2, D = 0, Q = 0$ $s = 14$	$\alpha = 0.0286, \beta = 0,$ $\gamma = 0.0001$
SMAPE	15.26072	19.81673	16.879

Table 5.3: Comparative result of ATM ID NN5-030 for in-sample 95%

The SMAPEs of the MASA, SARIMA and exponential smoothing models and all parameters are shown in Table 5.3. Figure 5.28 - 5.30 show the residuals of the ACF of each model. The ACF of the MASA model, the SARIMA model and the exponential smoothing model are similar. Correspondingly, the residuals of the MASA model, the SARIMA model and the exponential smoothing model are independent. Table 5.4 shows the comparative analysis of 111 ATM time series data using SMAPE. The MASA model wins 68 time series from 111 time series and has the best average SMAPE. Note: the period of $s = 14$ is applied to SARIMA.

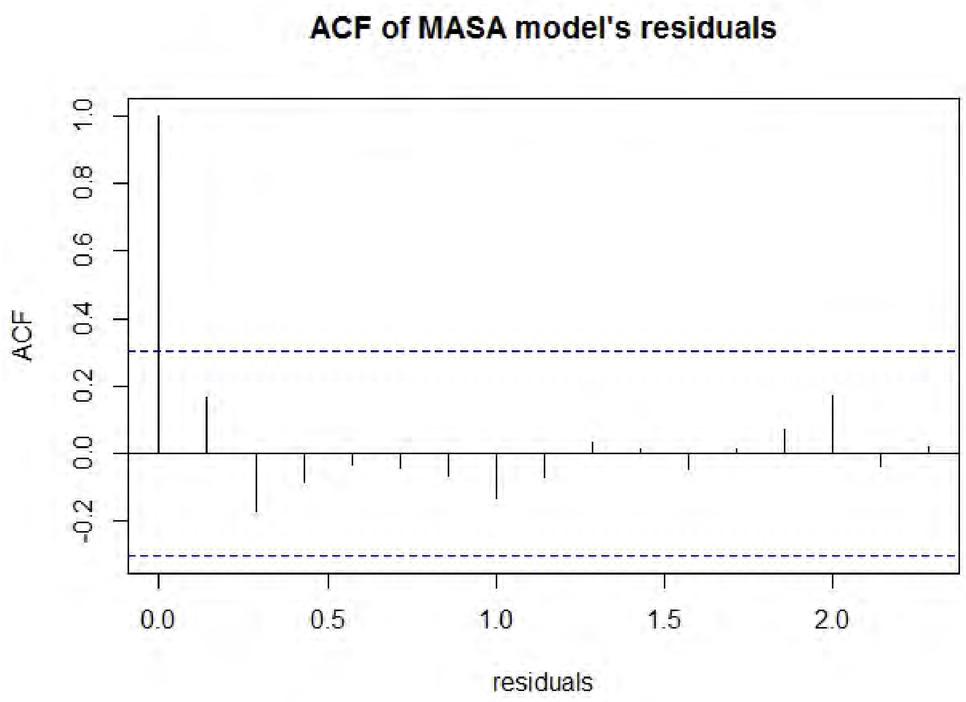


Figure 5.28: ACF of MASA model residuals (in-sample 95%)

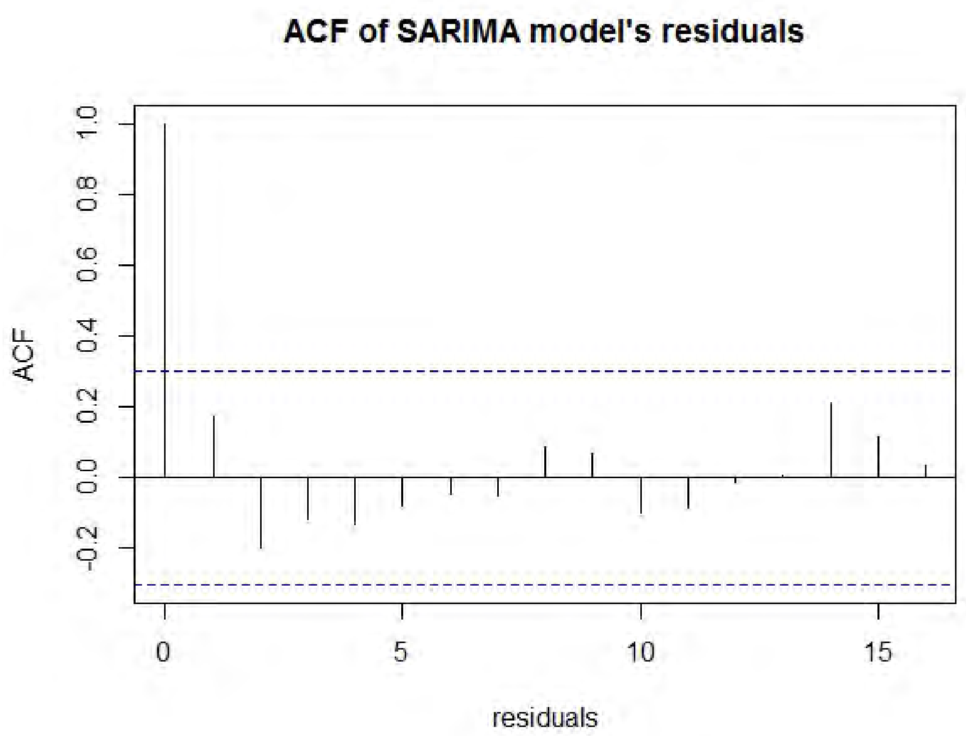


Figure 5.29: ACF of SARIMA model residuals (in-sample 95%)

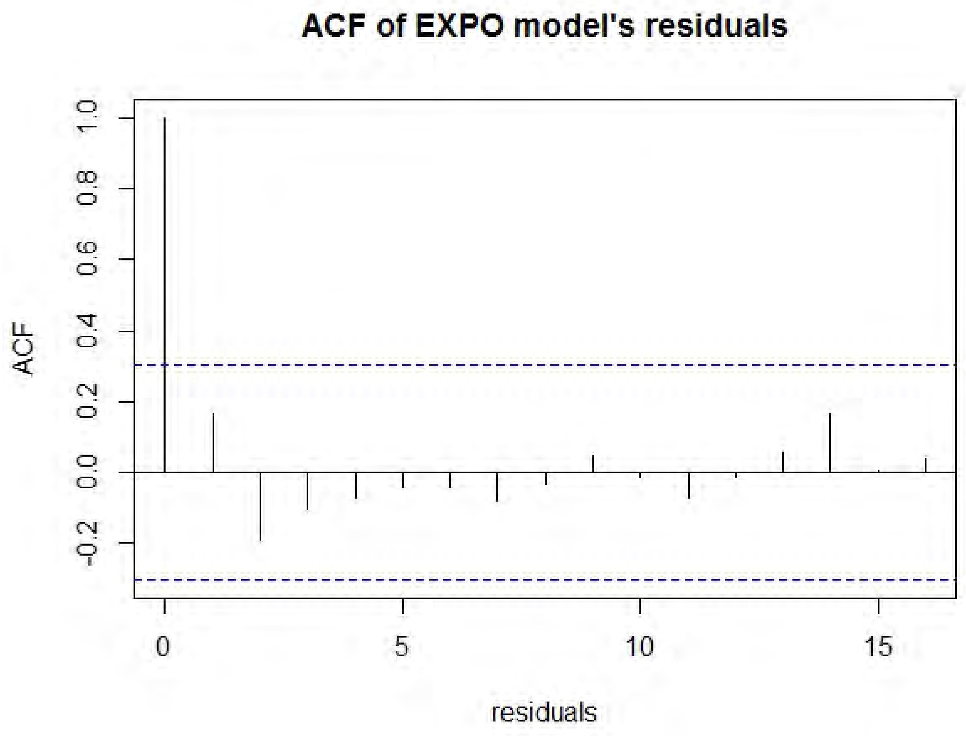


Figure 5.30: ACF of exponential smoothing model residuals (in-sample 95%)

	MASA model	SARIMA model	Exponential smoothing model
Number of best SMAPE	68	2	41
Averagre SMAPE	19.74685	27.04075	21.15076

Table 5.4: Comparative result of ATM ID NN5-030 with 111 time series (in-sample 95%)

Table 5.5 shows the hypothesis testing of the models. The p-values in Table 5.5 are very small which are significant at $\alpha \leq 0.05$. As a result, the hypothesis H_0 is rejected. It can be concluded that the set of the SMAPE of the MASA model is different from the SARIMA model and the exponential smoothing model by the significant level 0.05.

Significant level $\alpha = 0.05$	MASA and SARIMA	MASA and Exponential smoothing
W-value	38	2206
p-value	0.0000	0.00799
Length of data	111	111

Table 5.5: Wilcoxon signed-rank test of the models for in-sample 95%

5.2.2 Case II : in-sample data 90% and out-sample data 10%.

Figure 5.31 and 5.32 show the comparative models of the ATM ID NN5-030 time series data. The forecast data of the SARIMA model is very different with the forecast data of the MASA model and the out-sample data which the forecast data of the SARIMA model converges to one value. Moreover, the behavior of the forecast data from the MASA model and the exponential smoothing model is quite similar.

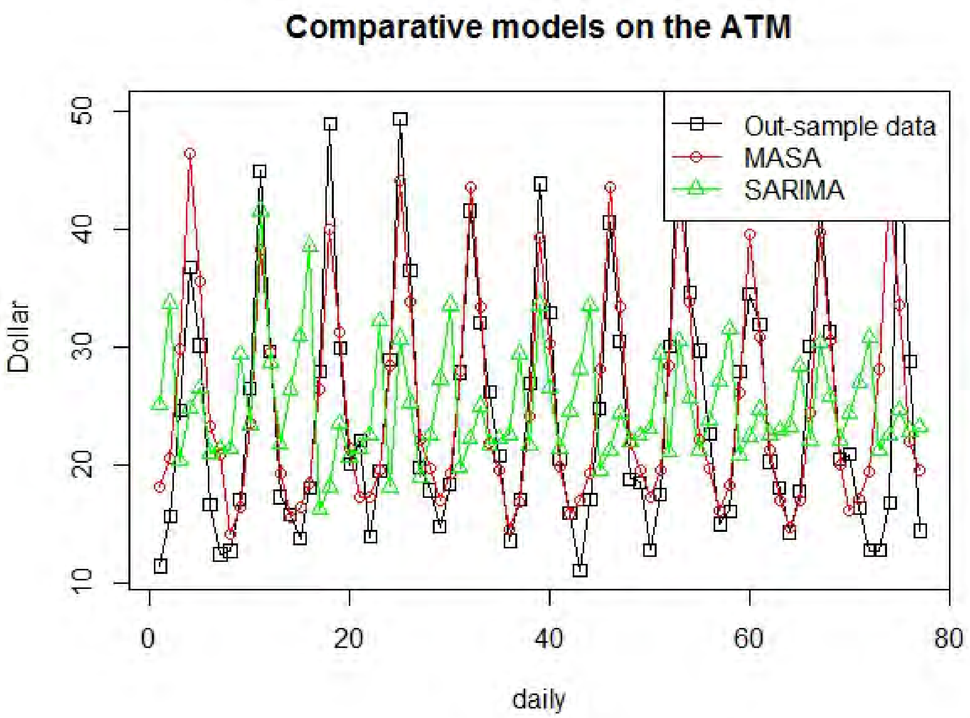


Figure 5.31: Comparative results of the MASA model and SARIMA model for in-sample 90% (ATM ID NN5-030)

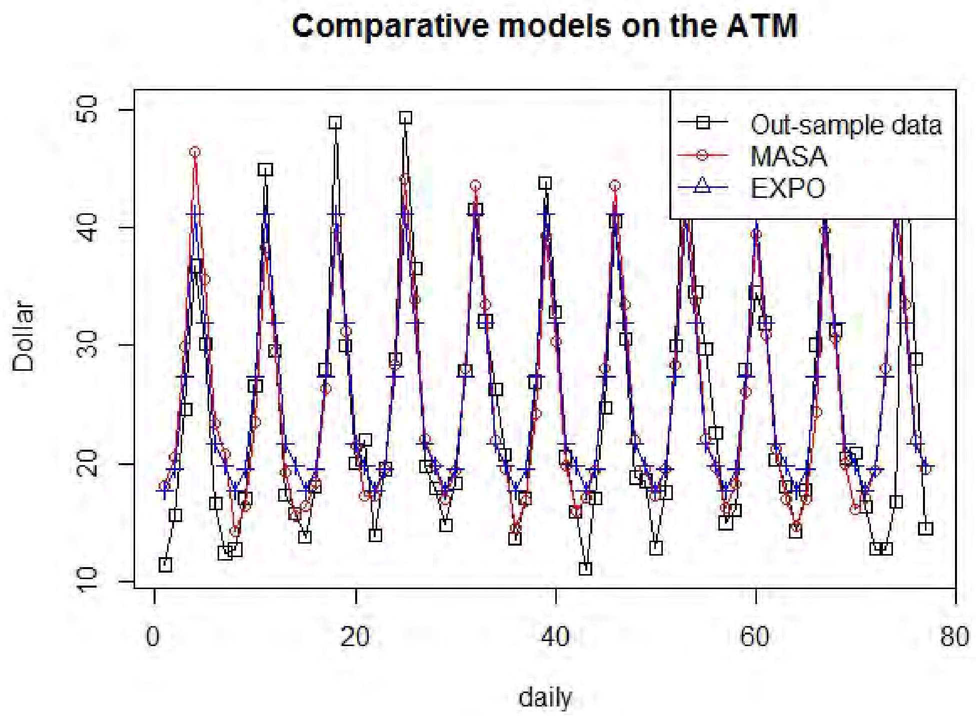


Figure 5.32: Comparative results of the MASA model and exponential smoothing model for in-sample 90% (ATM ID NN5-030)

	MASA model	SARIMA model	Exponential smoothing model
Parameters of the model	Table 5.1	$p = 1, d = 1, q = 2,$ $P = 2, D = 0, Q = 0$ $s = 14$	$\alpha = 0.0274, \beta = 0$ $\gamma = 0.0001$
SMAPE	14.93942	37.76245	15.82657

Table 5.6: Comparative result of ATM ID NN5-030 for in-sample 90%

The SMAPEs of the MASA, SARIMA and exponential smoothing models and all parameters are shown in Table 5.6. The Figure 5.33 - 5.35 show the residuals of ACF of each model. The ACF of the MASA model shows the best performance, and the ACF of the SARIMA model shows the worst. Thus, the residuals of the MASA model and the exponential smoothing model are independent but the residuals of the SARIMA model are not independent. Table 5.7 shows the comparative analysis of 111 ATM time series data using SMAPE. The MASA model wins 76 time series from 111 time series and has the best average SMAPE.

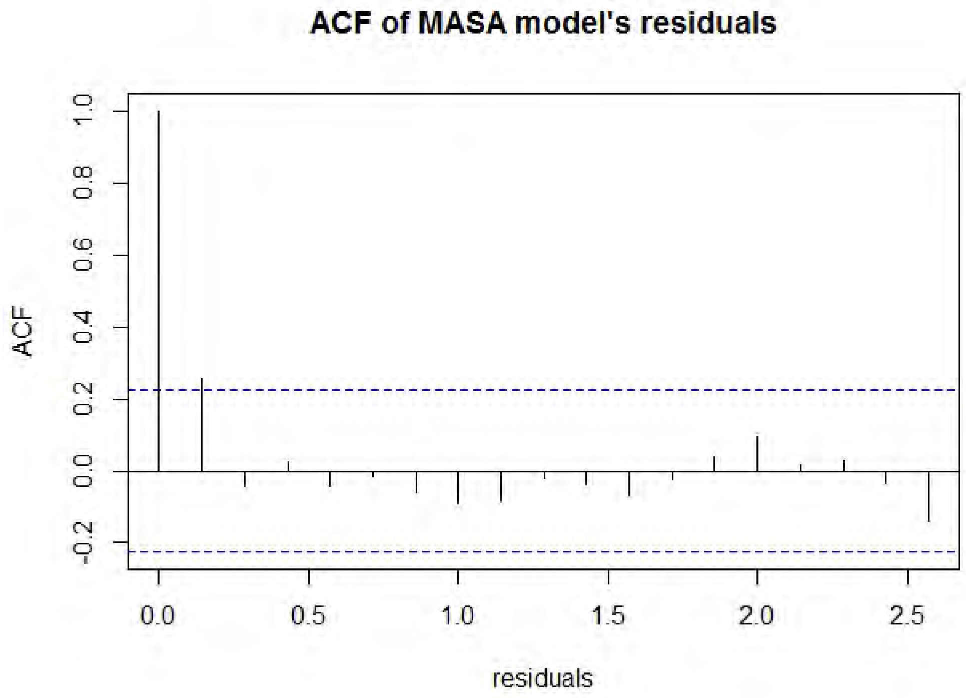


Figure 5.33: ACF of MASA model residuals (in-sample 90%)

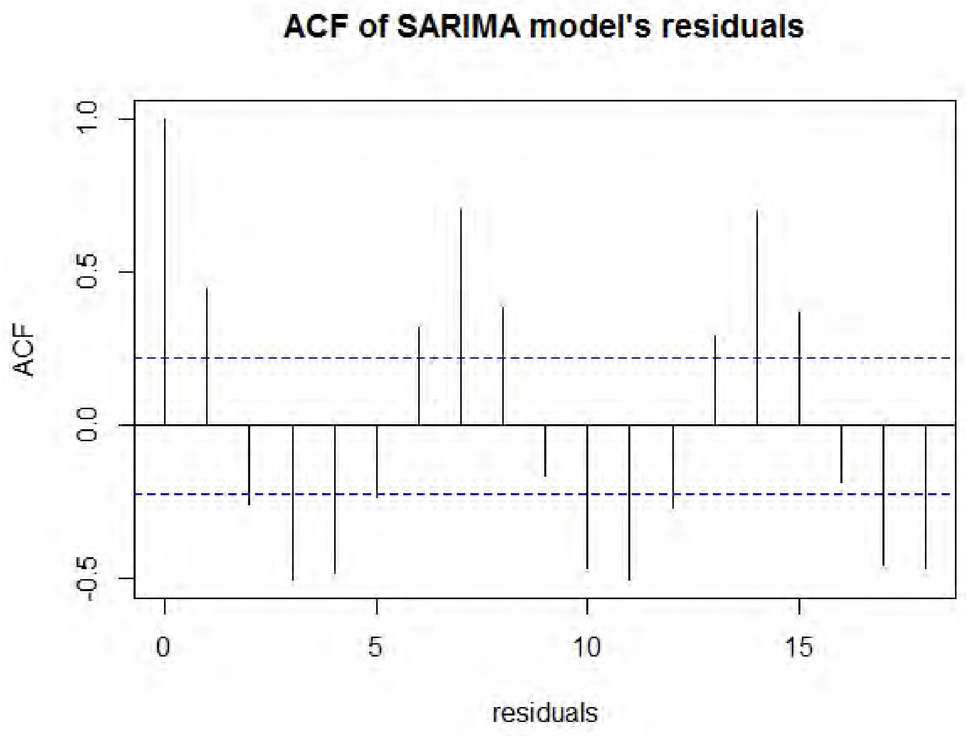


Figure 5.34: ACF of SARIMA model residuals (in-sample 90%)

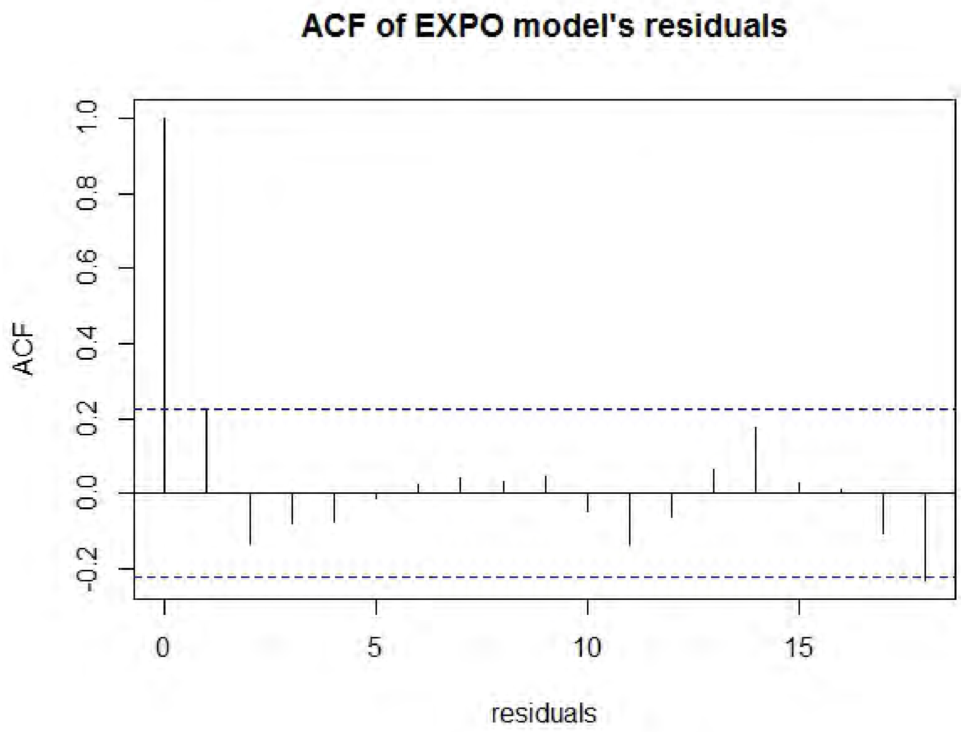


Figure 5.35: ACF of exponential smoothing model residuals (in-sample 90%)

	MASA model	SARIMA model	Exponential smoothing model
Number of best SMAPE	76	1	34
Averagre SMAPE	21.20529	40.20777	26.30625

Table 5.7: Comparative result of ATM ID NN5-030 with 111 time series (in-sample 90%)

Table 5.8 shows the hypothesis model testing. In this research, the Wilcoxon signed-rank test is used for comparing two related sets of the SMAPEs of the models. The hypothesis testing of the model is defined by H_0: difference between the pairs of models follows a symmetric distribution around zero and H_1: difference between the pairs of models does not follow a symmetric distribution around zero.

The p-values in Table 5.8 are small which are significant at $\alpha \leq 0.05$. As a result, the hypothesis H_0 is rejected. It can be concluded that the set of the SMAPEs of the MASA model is different from the SARIMA model and the exponential smoothing model by the significant level 0.05.

Significant level $\alpha = 0.05$	MASA and SARIMA	MASA and Exponential smoothing
W-value	3	1694
p-value	0.0000	0.00003198
Length of data	111	111

Table 5.8: Wilcoxon signed-rank test of the models for in-sample 90%

CHAPTER VI

CONCLUSION

This research proposes the MASA model to forecast the cash withdrawals of ATM time series. The process consists of two algorithms. The first algorithm is the MASA model-building algorithm. It is used for generating the MASA model. The second algorithm is the MASA model-forecasting algorithm. It is used for predicting the future value using the MASA model. Both algorithms are based on the aggregate and the disaggregate techniques on the subsequences of the time series.

From the empirical results, the aggregate and disaggregate technique can reduce the sway that occurs in the time series. The generated subsequences can filter the patterns and construct appropriate models that hid in the time series.

The results in the section 5.1 show the MASA model can forecast the future values close to the out-sample data. In the section 5.2, the MASA model can forecast more accurate than the SARIMA model and the exponential smoothing model based on SMAPE.

In this research, the algorithm uses the fixed number of the subsequences which the fixed number may be not appropriate with the time series. Therefore, the technique of the selecting number of the subsequences should be employed.

Moreover, the disaggregation technique uses only the average of the values in the empirical study which the new disaggregate technique should be investigated.

References

Takeshi Amemiya and Roland Y Wu. 1972. The effect of aggregation on prediction in the autoregressive model. Journal of the American Statistical Association 67,339:628–632.

Robert R Andrawis, Amir F Atiya, and Hisham El-Shishiny. 2011. Forecast combinations of computational intelligence and linear models for the nn5 time series forecasting competition. International Journal of Forecasting 27,3:672–688.

Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. Journal of econometrics 31,3:307–327.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. Time series analysis: forecasting and control. John Wiley & Sons.

S. Crone. 2008. Time series forecasting competition for computational intelligence. http://www.neural-forecasting competition.com, 2008 .

Saad M Darwish. 2013. A methodology to improve cash demand forecasting for atm network. International journal of Computer and electrical engineering 5,4:405.

Robert F Engle. 1983. Estimates of the variance of us inflation based upon the arch model. Journal of Money, Credit and Banking 15,3:286–301.

Rainer Handel, Manfred N Huber, and Stefan Schroder. 1998. ATM networks: concepts, protocols, applications. Addison-Wesley Longman Ltd.

Rob J Hyndman, Yeasmin Khandakar, et al. 2007. Automatic time series for forecasting: the forecast package for r. Technical report, Monash University, Department of Econometrics and Business Statistics.

Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? Journal of econometrics 54,1-3:159–178.

Richard Lowry. 2014. Concepts and applications of inferential statistics.

Bedman Narteh. 2013. Service quality in automated teller machines: an empirical investigation. Managing Service Quality: An International Journal 23,1:62–89.

Francisco A Pino, Pedro A Morettin, and Raúl P Mentz. 1987. Modelling and forecasting linear combinations of time series. International Statistical Review/Revue Internationale de Statistique pp. 295–313.

Nayera Sadek, Alireza Khotanzad, and Thomas Chen. 2003. Atm dynamic bandwidth allocation using f-arima prediction model. In Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on, pp. 359–363. IEEE.

Rimvydas Simutis, Darius Dilijonas, and Lidija Bastina. 2008. Cash demand forecasting for atm using neural networks and support vector regression algorithms. In 20th International Conference, EURO Mini Conference, Continuous Optimization and Knowledge-Based Technologies(EurOPT-2008), Selected Papers, Vilnius, pp. 416–421.

S.Prasanna. Probability and mathematical statistics. University of Louisville, KV 40292 USA ,http://www.math.louisville.edu/ pksaho01/teaching/Math662TB-09S.pdf.

Jörg D Wichard. 2011. Forecasting the nn5 time series with hybrid models. International Journal of Forecasting 27,3:700–707.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. Biometrics bulletin 1,6: 80–83.

WS Wei William. 1990. Time series analysis. Univariate and Multivariate Methods, New York .

APPENDICES

APPENDIX A

COMPARATIVE MODELS

This section introduces the comparative models that uses to evaluate the accuracy of the MASA model. The comparative models comprise of two models. The first model is the exponential smoothing model and the second model is the seasonal autoregressive integrated moving average model.

A.1 Exponential smoothing model

In this section, the exponential smoothing model is introduced. It is divided into three parts. The first part is the basic exponential smoothing model. The second part is the double exponential smoothing. And the third part is triple exponential smoothing.

A.1.1 Basic exponential smoothing model

Let $\{X_t\}$ be the time series data (starting at time $t = 0$), and the exponential smoothing be the $\{l_t\}$, which it is a estimate of the next value of X. The simple of exponential smoothing is given by the formula:

$$l_0 = X_0$$

$$l_t = \alpha X_t + (1 - \alpha) l_{t-1},$$

$t > 0$,where l is the smoothing factor, and $0 < \alpha < 1$.

$\hat{X}_t$ can be forecasted as follow

$$\hat{X}_t = l_t.$$

A.1.2 Double exponential smoothing model

Let $\{X_t\}$ be the time series data (starting at time $t = 0$) and $\{b_t\}$ be the estimate of the trend at time t. The double exponential smoothing is given by the formulas:

$$l_0 = X_0$$

$$l_1 = X_1$$

$$b_1 = l_1 - l_0$$

and for $t = 1, 2, \dots$ by

$$l_t = \alpha X_t + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

where α is the data smoothing factor, $0 < \alpha < 1$, and β is the trend smoothing factor, $0 < \beta < 1$.

Let $\hat{X}_{t+h}$ be the forecasted data where $h > 0$. $\hat{X}_{t+h}$ can be forecasted as follow

$$\hat{X}_{t+h} = l_t + hb_t$$

The initial value b_0 can be set as $\dfrac{X_n - X_0}{n}$ for some $n > 1$.

A.1.3 Triple exponential smoothing model

Let $\{X_t\}$ be the time series data (starting at time $t = 0$) with a seasonal change of length L and $\{c_t\}$ be the estimate of seasonal. The triple exponential smoothing is given by the formulas:

$$l_0 = X_0$$

$$l_1 = X_1$$

$$b_1 = l_1 - l_0$$

and for $t = 1, 2, \dots$ by

$$l_t = \alpha \frac{X_t}{c_{t-L}} + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

$$c_t = \gamma \frac{X_t}{l_t} + (1-\gamma)c_{t-L}$$

Let $\hat{X}_{t+h}$ be the forecasted data where $h > 0$. $\hat{X}_{t+h}$ can be forecasted as follow

$$\hat{X}_{t+h} = (l_t + hb_t)c_{t-L+1+(m-1)modL}$$

where α is the data smoothing factor, $0 < \alpha < 1$, β is the trend smoothing factor, $0 < \beta < 1$, and

γ is the seasonal change smoothing factor, $0 < \gamma < 1$.

The initial trend b_0 can be calculated by

$$b_0 = \frac{1}{L}(\frac{X_{L+1} - X_1}{L} + \frac{X_{L+2} - X_2}{L}, \ldots, \frac{X_{L+L} - X_L}{L})$$

The initial seasonal c_i for $i = 1, 2, \ldots, L$ can be calculated by

$$c_i = \frac{1}{N}\sum_{j=1}^{N} \frac{X_{L(j-1)+i}}{A_j},$$

for all $i = 1, 2, \ldots, L$, where

$$A_j = \frac{\sum_{i=1}^{L} X_{L(j-1)+i}}{L},$$

for all $j = 1, 2, \ldots, N$

Note that, A_j is the average value of X in the j th cycle of the data.

A.2 Seasonal autoregressive integrated moving average model

A general seasonal autoregressive integrated moving average or SARIMA model of order $(p, d, q) \times (P, D, Q)_s$ representing the time series Z_t can be written as

$$\phi(B)\varphi(B^s)\nabla^d\nabla^D Z_t = \theta(B)\varsigma(B^s)a_t$$

where, Z_t represent the time series data at time t and a_t represent the white noise terms at time t. B^s is a lag operator defined by $B^s Z_t = Z_{t-s}$.

∇ is defined by $\nabla Z_t = Z_t - Z_{t-1} = (1 - B)Z_t$ and $\nabla = (1 - B)$.

$\phi(B)$ and $\theta(B)$ are the autoregressive (AR) and the moving averages (MA) operators of order and respectively. They are defined as

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$$

$$\theta(B) = 1 - \sum_{j=1}^{q} \theta_j B^j$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients and $\theta_1, \theta_2, \ldots, \theta_q$ are the moving averages coefficients.

$\varphi(B^s)$ and $\varsigma(B^s)$ are the seasonal autoregressive (AR) and the seasonal moving averages (MA) operators of order and respectively. They are defined as

$$\varphi(B^s) = 1 - \sum_{i=1}^{P} \varphi_i B^{si}$$

$$\varsigma(B^s) = 1 - \sum_{j=1}^{Q} \varsigma_j B^{sj}$$

where $\varphi_1, \varphi_2, \ldots, \varphi_P$ are the autoregressive coefficients and $\varsigma_1, \varsigma_2, \ldots, \varsigma_Q$ are the moving averages coefficients.

APPENDIX B

PROGRAMMING OF THE MASA MODEL

In this section, the programming of the MASA model is introduced. This research uses R programming to solve this problem. The MASA model in R programming consists of the eight steps as follow:

Step 1: The time series data is called by library xlsx.

Step 2: The MASA model is generated by the building MASA algorithm.

Step 3: The future values are forecasted by the forecasting MASA algorithm.

Step 4: The SMAPE of the MASA model is calculated.

Step 5: The comparative models are generated. They are called the SARIMA model and the exponential smoothing model.

Step 6: The graphs of the empirical result are plotted

Step 7: The graphs of comparative analysis are plotted.

Step 8: The Wilcoxon signed-rank test is applied and the SMAPEs of the models are shown.

The details are shown below.

Step 1 : The time series data is called by library "xlsx ".

```
> library(xlsx)
> library(forecast)
> getwd()
> setwd("C:\\Users\\Administrator\\Desktop\\Thesis")
> filename <- "NN5_COMPLETE.xls"
> nn_gc1_B <- read.xlsx(filename, sheetIndex=1, startRow=18,
colIndex=107, endRow =752 , header=FALSE)
> for (i in 1:length(nn_gc1_B[,1])){
>   if (is.na(nn_gc1_B[i,1])){
>     nn_gc1_B[i,1]<- nn_gc1_B[i-1,1]
>    }
```

```
> }
> nn_gc1_B <- nn_gc1_B[!is.na(nn_gc1_B)]
> ts1<-ts((nn_gc1_B),frequency=4)
> plot(ts1,col="blue",main = "Cash of ATM time series",
ylab = expression("Dollar"), xlab =expression("daily"))
```

Step 2 : The MASA model is generated by the building MASA algorithm.

```
### The data are divided into in sample data and out sample data
and then the in sample data are divided into training data and
validating data.
> l = 7          # Period of aggregate group
> N_of_Data <- floor(length(nn_gc1_B)*0.95)
> N_of_test <- length(nn_gc1_B)-N_of_Data
> N_of_Data <- floor(length(nn_gc1_B)*0.95)
> N_of_test <- length(nn_gc1_B)-N_of_Data
> N_of_Data1 <- N_of_Data*0.95
> N_of_Data1 <- floor(N_of_Data1)
> N_of_Data1 <- l*floor(N_of_Data1/l)
> N_of_Data2 <- N_of_Data-N_of_Data1
> N_of_Data2 <- l*floor(N_of_Data2/l)
> N_of_Data <- N_of_Data1+N_of_Data2
> N_of_test <- (floor(length(nn_gc1_B)/l)*l)-N_of_Data
> write.csv(nn_gc1_B[((N_of_Data1+N_of_Data2)+1)
:(floor(length (nn_gc1_B)/l)*l)],"test.csv")
> write.csv(nn_gc1_B[1:(N_of_Data1+N_of_Data2)],"Data.csv")
> write.csv(nn_gc1_B[1:N_of_Data1],"Data1.csv")
> write.csv(nn_gc1_B[(N_of_Data1+1):(N_of_Data1+N_of_Data2)]
, "Data2.csv")
> Data1 <- read.csv("Data1.csv")        # Training data
> Data2 <- read.csv("Data2.csv")        # Validating data
> Data <- read.csv("Data.csv")          # in sample data
> test <- read.csv("test.csv")          # Out sample data
> plot(ts(Data1[,2]),col="blue")     #Graph of Training data
> plot(ts(Data2[,2]),col="blue")     #Graph of Validating data
> plot(ts(Data[,2]),col="blue")      #Graph of in sample data
> plot(ts(test[,2]),col="blue")      #Graph of out sample data
```

```
###The training data are aggregated.
> AData1 <- 1:(N_of_Data1/l)     #Aggregate training data
> for(i in 1:length(AData1)){
>   sum <- 0
>   for(j in (1+(l*(i-1))):(l+(l*(i-1)))) {
>     sum <- sum + Data1[j,2]
>   }
>   AData1[i] <- sum
>   sum <- 0
> }
> plot(ts(AData1,frequency=4))

###The validating data are aggregated.
> AData2 <- 1:(N_of_Data2/l)  #Aggregate validating data
> for(i in 1:length(AData2)){
>   sum <- 0
>   for(j in (1+(l*(i-1))):(l+(l*(i-1)))) {
>     sum <- sum + Data2[j,2]
>   }
>   AData2[i] <- sum
>   sum <- 0
> }
> plot(ts(AData2,frequency=4))

### The subsequence of aggregate training are generated.
> m = 3           # The maximum common difference
> list_AData1 <- list()
> list_AData1<-list()
> list_Seq_1<-list()
> k <- 1
> for (i in 1:m) {
>   for (j in 1:i) {
>     s <- seq(from=j,to=length(AData1),by=i)
>     u <- seq(from=j,to=length(AData1),by=i)
>     for (t in 1:length(s)){
```

```
>         s[t]<-AData1[s[t]]
>       }
>       list_AData1[[k]]<-s
>       list_Seq_1[[k]]<-u
>       k<-k+1
>    }}

### The subsequence of aggregate validating are generated.
> list_AData2<-list()
> list_Seq_2<-list()
> k <- 1
> for (i in 1:m) {
>    for (j in 1:i) {
>       s <- seq(from=j,to=length(AData2),by=i)
>       u <- seq(from=j,to=length(AData2),by=i)
>       for (t in 1:length(s)){
>         s[t]<-AData2[s[t]]
>       }
>       list_AData2[[k]]<-s
>       list_Seq_2[[k]]<-u
>       k<-k+1
>    }}
> k<-k-1

### The best ARIMA model is fitted and SMAPE of each subsequence
is calculated.
> list_forecast_arima <- list()
> SMAPE_arima <- c()
> Detail_Model1<-list()
##ARIMA
> for (i in 1:k){
>    AData1ts <- ts(list_AData1[[i]],frequency =1)
>    AData2ts <- ts(list_AData2[[i]],frequency =1)
>    fit <- auto.arima(AData1ts, max.p=7, max.q=7, seasonal=FALSE)
>    Detail_Model1[[i]]<-fit
>    f<-forecast(fit, h=length(AData2ts))
```

```
>    list_forecast_arima[[i]] <- f$mean
>    Forecast <- list_forecast_arima[[i]]
>    A<-abs(Forecast[1:length(AData2ts)]-AData2ts[1:length
( AData2ts)])
>    B<-(abs(Forecast[1:length(AData2ts)])+abs(AData2ts[1:
length(AData2ts)]))/2
>    SMAPE_arima[i] <-sum(A/B)*100/length(AData2ts)
> }
```

Step 3 : The future values are forecasted by the forecasting MASA algorithm.

```
### The aggregate forecasted value and its subsequence are
identified from each future value.
> list_forecast_NewModel <- list()      # List of forecasted values
> MAPE_NewModel <- c()
> Using_Model <- c()
> for (i in 1:k){
>    list_forecast_NewModel[[i]]<-list_forecast_arima[[i]]
>    MAPE_NewModel[i]<-SMAPE_arima[i]
>   Using_Model[i] <- "arima"
> }
> N_of_Data
> AData <- 1:(N_of_Data/1)
> for(i in 1:length(AData)){
>    sum <- 0
>    for(j in (1+(1*(i-1))):(1+(1*(i-1)))) {
>       sum <- sum + Data[j,2]
>    }
>    AData[i] <- sum
>    sum <- 0
> }
> plot(ts(AData,frequency=4))
> Atest <- 1:(N_of_test/1)
> for(i in 1:length(Atest)){
>    sum <- 0
>    for(j in (1+(1*(i-1))):(1+(1*(i-1)))) {
>       sum <- sum + test[j,2]
```

```
>     }
>     Atest[i] <- sum
>     sum <- 0
> }
> list_AData <- list()
> list_AData<-list()
> list_Seq_In<-list()
> k <- 1
> for (i in 1:m) {
>     for (j in 1:i) {
>         s <- seq(from=j,to=length(AData),by=i)
>         u <- seq(from=j,to=length(AData),by=i)
>         for (t in 1:length(s)){
>             s[t]<-AData[s[t]]
>         }
>         list_AData[[k]]<-s
>         list_Seq_In[[k]]<-u
>         k<-k+1
>     }}
> k<-k-1
> list_Atest <- list()
> list_Atest<-list()
> list_Seq_Out<-list()
> k <- 1
> for (i in 1:m) {
>     for (j in 1:i) {
>         s <- seq(from=j,to=length(Atest),by=i)
>         u <- seq(from=j,to=length(Atest),by=i)
>         for (t in 1:length(s)){
>             s[t]<-Atest[s[t]]
>         }
>         list_Atest[[k]]<-s
>         list_Seq_Out[[k]]<-u
>         k<-k+1
>     }}
> k<-k-1
```

```
> list_forecast_NewModel <- list()
> Detail_Model2<-list()
> for (i in 1:k){
>   print(i)
>   ADatats <- ts(list_AData[[i]],frequency = 1)
>   fit <- arima(ADatats,  order=c(Detail_Model1[[i]] $arma[1]
,Detail_Model1[[i]]$arma[6],Detail_Model1[[i]]$arma[2]),
seasonal=c(Detail_Model1[[i]]$arma[3],Detail_Model1[[i]]
$arma[7],Detail_Model1[[i]]$arma[4]))
>   f<-forecast(fit, h=(length(test[,2])/1))
>   list_forecast_NewModel[[i]] <- f$mean
>   Detail_Model2[[i]]<-fit
> }

### The aggregate values are forecasted by considering the
minimum SMAPE of subsequences.
> OrderOfsequence <- c()
> SortMAPE_NewModel <- sort(MAPE_NewModel)
> for (i in 1:k){
>   for (j in 1:k){
>     if(SortMAPE_NewModel[i]==MAPE_NewModel[j]){
>       OrderOfsequence[i]<-j}
>   }
> }
> Forecast_NewModel <- 1:(length(test[,2])/1) ;Forecast_NewModel
[1:(length(test[,2])/1)]<-NaN
> Forecast_NewModelfull <- Forecast_NewModel
[!is.na(Forecast_NewModel)]
> NumberofForecast<-0
> NumberofSequence <- 1:(length(test[,2])/1);Forecast_NewModel
[1:(length(test[,2])/1)]<-NaN
> t<-1
> while(length(Forecast_NewModelfull)!=(length(test[,2])/1)){
>   OrderOfsequence[t]
>   ts<-list_forecast_NewModel[[OrderOfsequence[t]]]
>   N <-list_Seq_Out[[OrderOfsequence[t]]]
```

```
>    for(i in 1:length(N)){
>       if(is.na(Forecast_NewModel[N[i]])){
>          Forecast_NewModel[N[i]]<-ts[i]
>          NumberofSequence[N[i]]<-OrderOfsequence[t]
>       }
>    }
>    Forecast_NewModelfull <- Forecast_NewModel
[!is.na(Forecast_NewModel)]
>    t<-t+1
> }
> t<-t-1
> Using_seq <- list()
> Detail_Final_Model<-list()
> Using_Final_Model <- c()
> for(i in 1:length(NumberofSequence)){
>    Using_seq[[i]]<-list_Seq_In[[NumberofSequence[i]]]
>    Using_Final_Model[i]<-Using_Model[NumberofSequence[i]]
>    Detail_Final_Model[[i]]<-> Detail_Model1 [[NumberofSequence[i]]]
> }

### The aggregate forecasted values are disaggregated.
> ratio1<-c()                              #Average ratios
> ratio2<-c()                              #Weighted average ratios
> Forecast_Data<-c()
> for(r in 1:length(Forecast_NewModel)){
>    SeqIn <- list_Seq_In[[NumberofSequence[r]]]
>    ADataIn <- list_AData[[NumberofSequence[r]]]
>    k<-(SeqIn[2]-SeqIn[1])
>    j<-(k-SeqIn[1])
>    n<-length(AData)
>    sum<-0
>    for(t in 1:l){
>       i<-1
>       while(((k*i)-j)<= n){
>          sum<-sum+(Data[(((k*i)-j-1)*l+t),2]/AData[(i*k)-j])
>          i<-i+1}
```

```
>       sum<-sum/length(ADataIn)
>       ratio1[t]<-sum
>    }
>    sumOfratio1<-sum(ratio1)
>    for(t in 1:l){
>       ratio2[t]<-ratio1[t]/sumOfratio1
>    }
>    for(t in 1:l){
>       Forecast_Data[l*(r-1)+t]<-Forecast_NewModel[r]*ratio2[t]
>    }
> }
```

Step 4 : SMAPE of the MASA model is calculated.

```
> tsnewmodel<-ts(Forecast_Data,frequency = 7)     #The forecasted values
of the MASA model
> tstest<-ts(test[,2],frequency = 7 )       #The out sample data
> A<-abs(Forecast_Data-test[,2])
> B<-(abs(Forecast_Data)+abs(test[,2]))/2
> SMAPNewModel <-sum(A/B)*100/length(Forecast_Data) #SMAPE of
MASA model
```

Step 5 : The comparative models are generated. There are the SARIMA model and the exponential smoothing model.

```
### SARIMA
> tsData<-ts(Data[,2],frequency = 1)
> fitSarima <- auto.arima(tsData, max.p=25, max.q=25)
> fSarima<-forecast(fitSarima, h=length(tstest))
> tsSarima<-ts(fSarima$mean,frequency=7)  # The forecasted values
of the SARIMA model
> A<-abs(tsSarima-test[,2])
> B<-(abs(tsSarima)+abs(test[,2]))/2
> SMAPESArima <-sum(A/B)*100/length(tsSarima) #SMAPE of SARIMA model

### Exponential Smoothing (ets)
> tsData1<-ts(Data[,2],frequency = 1)
```

```
> fitexpo3 <- ets(tsData1,model="ZZZ")
> fexpo3<-forecast(fitexpo3, h=length(test[,2]))
> tsexpo3<-ts(fexpo3$mean,frequency=7) # The forecasted values of
the exponential smoothing model
> A<-abs(tsexpo3-test[,2])
> B<-(abs(tsexpo3)+abs(test[,2]))/2
> SMAPEexpo3 <-sum(A/B)*100/length(tsexpo3) #SMAPE of the
exponential smoothing model
```

Step 6 : The graph of the empirical result are plotted.

```
> plot(tstest,col="black",main = "Empirical result of the MASA model"
, ylab = expression("Dollar"), xlab =expression("daily"))
> points(tstest, pch=0,col="black")
> lines(tsnewmodel,col="red")
> points(tsnewmodel, pch=1,col="red")
> categories1 <- c("Out-sample data", "MASA")
> colors1 <- c("black", "red")
> legend("topright", col=colors1, categories1,pch=c(0,1), bg="white"
, lwd=1)
> acf(tstest-tsnewmodel)
> Pacf(tstest-tsnewmodel)
```

Step 7 : The graph of comparative analysis are plotted.

```
> plot(tstest,col="black",main = "Comparative models on the
ATM ID NN5-106",ylab = expression("Dollar"), xlab =
expression("daily"), pch = 10)
> points(tstest, pch=0,col="black")
> lines(tsnewmodel,col="red")
> points(tsnewmodel, pch=1,col="red")
> lines(tsSarima,pch=".",col="green")
> points(tsSarima, pch=2,col="green")
> lines(tsexpo3,col="blue")
> points(tsexpo3, pch=3,col="blue")
> categories <- c("Out-sample data", "MASA", "SARIMA","ETS")
> colors <- c("black", "red", "green", "blue")
```

```
> legend("topright", col=colors, categories,pch=c(0,1,2,3)
, bg="white", lwd=1)
```

Step 8 : Willcoxon signed-rank test is applied and the SMAPEs of the models are shown.

```
> BJ<-Box.test(tstest-tsnewmodel,type="Ljung-Box",
lag=floor (log(length(tstest))))
> print("red");SMAPNewModel            #SMAPE of MASA model
> print("blue");SMAPESArima            #SMAPE of SARIMA model
> print("yellow");SMAPEexpo3           #SMAPE of the exponential
smoothing model
```

Biography

Paisit Khanarsa was born on December 2nd, 1992, in Bangkok, Thailand. He received a bachelors degree in Mathematics from the Department of Mathematics, Faculty of Science, Kasetsart University in 2014. He has been financially supported by the Science Achievement Scholarship of Thailand (SAST).

Publication: Khanarsa, Paisit, and Krung Sinapiromsaran. "Multiple ARIMA subsequences aggregate time series model to forecast cash in ATM." Knowledge and Smart Technology (KST), 2017 9th International Conference on. IEEE, 2017.