การลดความเสี่ยงของการซื้อขายหุ้นโดยการลงทุนแบบหุ้นคู่ขั้นสูง

นางสาวนวรัตน์ เอกก้านตรง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2558

RISK MITIGATION OF STOCK TRADE USING AN ADVANCED PAIRS TRADING STRATEGY

NAWARAT EKKARNTRONG

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Applied Mathematics
Suranaree University of Technology
Academic Year 2015

RISK MITIGATION OF STOCK TRADE USING AN ADVANCED PAIRS TRADING STRATEGY

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

Thesis Examining Committee

____

(Asst. Prof. Dr. Eckart Schulz)

Chairperson

____

(Prof. Dr. Pairote Sattayatham)

Member (Thesis Advisor)

____

(Prof. Dr. Bhusana Premanode)

Member (Thesis Co-advisor)

____

(Asst. Prof. Dr. Arjuna Chaiyasena)

Member

____

(Asst. Prof. Dr. Benjawan Rodjanadid)

Member

____

(Prof. Dr. Sukit Limpijumnong)

Vice Rector for Academic Affairs

and Innovation

____

(Prof. Dr. Santi Maensiri)

Dean of Institute of Science

To my mother

Songward Ekkarntrong

นวรัตน์ เอกก้านตรง : การลดความเสี่ยงของการซื้อขายหุ้นโดยวิธีการซื้อขายแบบหุ้นคู่ขั้นสูง, (RISK MITIGATION OF STOCK TRADE USING AN ADVANCED PAIRS TRADING STRATEGY) อาจารย์ที่ปรึกษา : ศาสตราจารย์ ดร. ไพโรจน์ สัตยธรรม, ดร. ภูษณ ปรีย์มาโนช, 130 หน้า.

การลงทุน / การทำนาย / ความเสี่ยง / ARIMA / MCMC / SVR

กระบวนการกลับเข้าสู่สมดุล (Mean Reversion process) ของการซื้อขายหุ้นคู่ (Pairs Trading) เป็นกลยุทธ์การตลาดที่เป็นกลางซึ่งเป็นอิสระจากการเคลื่อนไหวของตลาดและอยู่ภายใต้ข้อสันนิษฐานที่ว่าราคาของทั้งคู่ในที่สุดก็จะกลับไปเป็นค่าเฉลี่ยของตัวมันเอง ปัญหาคือการกลับเข้าสู่สมดุลของหุ้นใดๆ ไม่สอดคล้องกัน ทั้งนี้ขึ้นอยู่กับภาวะตลาด งานวิจัยนี้นำเสนอขั้นตอนวิธีการใหม่ที่เรียกว่า “multi-class Pairs Trading” ซึ่งเป็นความก้าวหน้าของวิธีการ co-integration ในการซื้อขายหุ้นคู่ (Pairs Trading) วิธีการที่เสนอใช้วิธีการกลับเข้าสู่สมดุล (Mean reversion) ร่วมกับค่าสัมประสิทธิ์ของความแปรปรวน (CV) และการจัดกลุ่มชุดข้อมูลที่จับคู่กัน นอกจากนี้วิธีการดังกล่าวยังให้พื้นที่ปลอดภัยสำหรับการซื้อขายเมื่อหุ้นคู่มีการเปลี่ยนแปลงทิศทาง ข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลที่เก็บรวบรวมจากหุ้น 134 ซึ่งอยู่ใน Global Dow ใช้ราคาทุกวันตั้งแต่ปี 2002 ถึงปี 2013 เป็นเวลา 10 ปี ผลการจำลองแสดงให้เห็นว่าวิธีการที่นำเสนอมีประสิทธิภาพดีกว่าวิธีการ cointegration อย่างเห็นได้ชัด ดังนั้นประโยชน์ของตัวแบบที่นำเสนอลดความเสี่ยงและเพิ่มผลตอบแทนของหุ้นคู่

จากการใช้วิธีการกลับเข้าสู่สมดุลร่วมกับค่าสัมประสิทธิ์ของความแปรปรวน (CV) ในอัลกอริทึมการซื้อขายหุ้นคู่เพื่อลดความเสี่ยงในการซื้อขาย ถ้าการเคลื่อนไหวหรือราคาในอนาคตสามารถคาดการณ์ได้ ความเสี่ยงจะลดลงอย่างหลีกเลี่ยงไม่ได้ วัตถุประสงค์ที่สองคือการทำนายราคาหุ้นของหุ้นคู่ โดยตัวแบบที่ใช้ในงานมิจัยนี้คือ Autoregressive Moving Average (ARIMA) ตัวแบบ Markov Chain Monte Carlo (MCMC) และตัวแบบ Support Vector Regression (SVR)

สาขาวิชาคณิตศาสตร์ ลายมือชื่อนักศึกษา ____________________

ปีการศึกษา 2558 ลายมือชื่ออาจารย์ที่ปรึกษา ________________

NAWARAT EKKARNTRONG : RISK MITIGATION OF STOCK TRADE USING AN ADVANCED PAIRS TRADING STRATEGY. THESIS ADVISOR : PROF. PAIROTE SATTAYATHAM, Ph.D, THESIS CO-ADVISOR : BHUSANA PREMANODE, Ph.D., Ph.D, DBA 130 PP.

PAIRS TRADING / MEAN REVERSION / COEFFICIENT OF VARIANCE / ARBITRAGE / RISK MITIGATION / PREDICTION / ARIMA / MCMC / SVR

The Mean Reversion process of pairs trading is a market neutral strategy, which is independent of market movements and carried an assumption that each price of the pair will eventually revert to its mean. The problem is that the Mean Reversion of any stock is not consistent, depending on market condition. This paper proposes a novel algorithm, called ‘multi-class pairs trading’, in which is an advance of cointegration method in pairs trading. The proposed model uses Mean Reversion and coefficient of variance to segregating and grouping a paired dataset, respectively. Additionally, it provides a buffer-trading zone when the paired stocks are changing their directions. In a portfolio trading, it extends an opportunity for a highly correlated and paired stock to cross-trade with any lowly correlated and paired stock. The data were collected from 134 stocks listed in the Global Dow, 10-year daily price from 2002 to 2013. The simulation results show that the cointegrated pairs trading using the proposed method outperforms those of the conventional cointegrated pairs trading outstandingly. Thus, benefits of the proposed model are to build a new series of risk mitigation and maximise returns of cointegrated stocks.

As using the mean reversion and CV in Pairs trading to mitigate the risk in trading, if the movement or the future price of the next time step to trade can be predicted, the risk shall be inevitably reduced. The second objective is to predict the stock prices of that paired stocks. The Autoregressive Integrated Moving Average (ARIMA) Model, Markov Chain Monte Carlo (MCMC) model, and Support Vector Regression (SVR) Model were used in this research.

School of Mathematics Student's Signature ________________

Academic Year 2014 Advisor's Signature ________________

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Professor Dr. Pairote Sattayatham, and my co-advisor, Dr. Bhusana Premanode, for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for my Ph.D study.

Beside my advisor and co-advisor, I would like to thank the rest of my thesis committee: Asst.Prof.Dr. Eckart Schulz, Asst. Prof. Dr. Arjuna Chaiyasena, and Asst. Prof. Dr. Benjawan Rodjanadid, for their perceptive comments and encouragement, but also for the hard question which incited me to widen my research from various aspects.

My sincere thanks also goes to Assoc. Prof. Dr. Jumlong Vongprasert and Piyatat Chatvorawit, who help me in programming part. Without they precious support it would not be possible to conduct this research.

Special thanks to a staff member at the School of Mathematics, Anusorn Rujirapa, for her help in dealing with the university bureaucracy.

The Development and Promotion of Science and Technology Talents Project (DPST) also deserves my gratitude. They have continuously supported me by offering grants from high school to through Ph.D study, along with providing chances for attending academic conferences and doing some part of this research in United Kingdom.

I thank my friends in the following institution, Dr. Amornrat, Dr. Suntorn, Piyatat, Dr. Sasithorn, Wiparat, and our Mathematics family at SUT, for their

help in my life at SUT.

Lastly, I am greatly grateful to members of my “family team”, my parents and my brother, for supporting me throughout writing this thesis and my life in general.

Nawarat Ekkarntrong

CONTENTS

Page

CHAPTER

CONTENTS (Continued)

Page

CONTENTS (Continued)

CONTENTS (Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Pairs trading is a trading strategies that attempts to be market neutral and capture the spread between two correlated stocks as they return to the mean price. It is also known as ”statistical arbitrage”.

The first practical Statistical pair trading is caused by Nunzio Tartaglia, a quantitative at Morgan Stanley in the mid 1980s. He and a group of scientists form a team with the goal to develop quantitative arbitrage strategies using state-of-art statistical techniques. One of the techniques was trading securities in pairs. This technique concerned with identifying pairs of securities whose price tended to move together. In 1987, Tartaglia and his group used pairs trading with the great success. The group disbanded in 1989 after that they worked in various other trading companies and the idea of pairs trading spread. The technique called Pairs Trading has since increased in popularity and has become a common trading strategy used by hedge funds and institutional investors.

If the movement or the future paired stocks prices of the next time step to trade can be predicted, the risk shall be reduced. Thus the prediction has a part in this study.

A main goal of this research is to mitigate the risk in trading. Therefore this study proposed the combined models of the Pairs Trading model and prediction model.

1.1 Motivation

From a valuation point of view the general idea for investing in the marketplace is to sell overvalued securities and buy the undervalued ones. However, it is possible to determine that a security is overvalued or undervalued only if we also know the true value of the security in absolute terms. But, this is very hard to do. Pairs trading attempts to resolve this using the idea of relative pricing; that is, if two securities have similar characteristics, then the prices of both securities must be more or less the same. Note that the specific price of the security is not of importance. The price may be wrong. It is only important that the prices of the two securities be the same. If the prices happen to be different, it could be that one of the securities is overpriced, the other security is underpriced, or the mispricing is a combination of both.

Pairs trading involves selling the higher-priced security and buying the lower-priced security with the idea that the mispricing will correct itself in the future. The mutual mispricing between the two securities is captured by the notion of spread. The greater the spread, the higher the magnitude of mispricing and greater the profit potential. A long–short position in the two securities is constructed such that it has a negligible beta and therefore minimal exposure to the market. Hence, the returns from the trade are uncorrelated to market returns, a feature typical of market neutral strategies.

Therefore the key to success in pairs trading lies in the identification of security pairs.

After using the Pairs Trading, the risk will be reduced. Moreover, if the paired stocks can be predicted, the risk shall be reduced more. Therefore this study combined the Pairs Trading with the prediction model to mitigate the risk in trading.

1.1.1 Literature Review

An early attempt at Pairs Trading is credited to Nunzio Tartaglia, a quantitative analyst at Morgan Stanley in the 1980s. Tartaglia gathered a group of professionals with the aim of forming a quantitative arbitrage strategy using statistical techniques. One technique that they implemented was trading pairs of securities. The procedure distinguishes between pairs of security prices that move together. The abnormality in the relationship indicates that the pair will be traded with anticipation that the abnormality will be neutralised in the future. Different schools of thought offer an alternative that is Mean Reversion. In normal circumstance, positive and negative returns on financial assets are temporary because return reverses to the mean in the long run; the speed of the reversing process can vary from one day to one year (Hillebrand, 2004). Lo and Mackinlay (1998), Fama and French (1988), and Poterba and Summers (1988) demonstrated using empirical evidence that positive market return persists over the short term. However, in the long term, profit opportunity is reverted. Campbell and Viceira (1999), Wachter (2002) and Campbell, Chan, and Viceira (2003) confirmed the findings by illustrating that Mean Reversion possesses the characteristics of equity index return over the long term. Additionally, Bessembinder, Coughenour, Seguin, and Smoller (1995) determined that Mean Reversion that exists in the financial markets uses empirical evidence from the term structure of future prices. The data sample of the authors' study was based on 11 different future markets including financial, metals, and agriculture markets. The daily settlement price from January 1982 to December 1991 was used. The disadvantage of the study methodology is that it can only spot Mean Reversion in the equilibrium condition of the market, and it cannot be applied when the market is in disequilibrium. Gatev, Goetzmann, and Rouwenhorst (2006) conducted an investigation into the risk and return characteristics of Pairs Trading using data from 1962 to 2002. The authors showed that simple Mean Reversion for a single stock index could not produce clear values. However, the values can be generated when trading suitably formulated pairs of stocks. Perlin (2007) proposed a multivariate version of pairs trading, which developed an artificial pair for a stock based on the information of m assets. This method assessed the performance of three versions of the multivariate approach for the Brazilian stock market using data for 57 assets from 2000 to 2006. The examination of performance was conducted using the calculation of raw returns, excessive returns, beta, and alpha. Do, Faff, and Hamza (2006) investigated a uniform and an analytical framework to implement Pairs Trading on arbitrary pairs and suggested an asset pricing-based model to parameterise pairs trading that included theoretical considerations rather than statistical history. Huck (2010) proposed a general and flexible framework for the selection of random pairs. Multiple return forecasts based on bivariate information sets and multi-criteria decision techniques were implemented.

As an overview on techniques in finance by Kovalerchuk et al. (2000), the prediction methods can be classified into three categories: numerical models (ARIMA models, Instance-based learning, neural networks, etc.), rule-based models (decision tree and DNF learning, naive Bayesian classifier, hidden Markov model, etc.), and relational data mining (inductive logic programming).One of the most popular and frequently used stochastic time series models is the Autoregressive Integrated Moving Average (ARIMA) model.The Markov Chain Monte Carlo (MCMC) methods are particularly attractive for practical finance applications. It was realized that most Bayesian inference could be done by MCMC, whereas very little be done without MCMC. Recently, Artificial Neural Networks (ANNs) have been attracting increasing attentions in the time series forecasting. Nowadays,the Support Vector Machine (SVM), a new statistic learning theory, has been receiving increasing attention for classification and forecasting. The Support Vector Regression (SVR) is used in forecasting problem.

1.2 Objectives

There are two objectives in this research. The first objective of this research is to introduce an advanced model of the current co-integration, called, multiclass Pairs Trading. The other objective is concerned with forecasting of stock data. As an Autoregressive Integrated Moving Average (ARIMA) model, a Markov Chain Monte Carlo (MCMC) method, and a Support Vector Regression (SVR) approach have been successfully used for modelling and predicting financial time series and they are used in many researches, so these three models are used in this research. The stock data is predicted by using these three prediction models as follows: Autoregressive Integrated Moving Average (ARIMA) model, Markov Chain Monte Carlo (MCMC) method, and support vector regression (SVR) approach.

1.2.1 Forecasting Methods

Normally, there are five fundamental steps in quantitative forecasting: i) problem definition; ii) grouping information; iii) preparatory analysis; iv) choosing and fitting models and v) performance measurements.

1.2.2 A New Novel multiclass Pairs Trading

This newly invented technique provides a new set of risk mitigation by providing a buffer-trading zone when the paired stocks are changing their directions. In portfolio trading, it extends an opportunity for a highly correlated and paired stock to cross-trade with any lowly correlated and paired stock. Thus, the proposed model maximises returns and minimises risk of cointegrated Pairs Trading stocks. The proposed model employs Mean Reversion and Coefficient of Variance (CV) algorithm (Premanode, Vonprasert, and Toumazou, 2013), and is now called 'Mean Reversion and CV', to segregate and group any paired stock indices under the cointegration method. The model consists of the following concepts: i) the application of Mean Reversion to segregate nonlinear and nonstationary time series datasets to different local datasets, ii) the grouping of the local datasets segregated with the coefficient of variance, iii) the calculation of the highest returns of a paired stock employing the multiclass Pairs Trading algorithm, and then comparing with the results of a conventional cointegration method, and iv) computing the expected return of the top ten pairs in the multiclass Pairs Trading that were cross-traded. The data of this study is the daily price for 127 stocks in the Global Dow, which included blue chips from leading companies of national reputation. The simulation results show that the cointegrated Pairs Trading using the proposed method outperforms those of the conventional co-integrated pairs trading outstandingly. Thus, benefits of the proposed model are to build a new series of risk mitigation and maximise returns of co-integrated stocks.

1.2.3 Prediction Models

There are three prediction models, ARIMA, MCMC, and SVR, in this research. The performance of these three models when predicting stock prices movements were shown.

1.3 Organization

This thesis is organized into seven chapters as follows. Chapter I, the motivation behind this research was described, as well as its objectives and organization. In Chapter II describes the theoretical related to the Pairs Trading. In Chapter III discusses the development of various forecasting methods. Chapter IV shows and discusses the time series data that be used in this research. In Chapter V describes the proposed model, multiclass Pairs Trading, and the cointegration pairs trading. The performance of this newly proposed model for Pairs Trading was compared with the performance of the cointegration Pairs Trading, as well as robustness test. In Chapter VI discusses all three prediction models using in this research , i.e., ARIMA, MCMC, and SVR models. The comparison of these three forecasting models are also discussed, as well as robustness test. Chapter VII provides a highlight and benefit of the proposed model, a combined models of Pair Trade and a prediction model. It also gives a conclusion on a comparison of the three prediction models, ARIMA, MCMC, and SVR.

Additionally, in Appendix, programme files and all Figures and Tables that did not show in the previous chapter present here.

CHAPTER II

PRELIMINARIES AND LITERATURE REVIEW

Definitions and facts of the concepts on Pairs Trading strategy, mainly covering topics related to Pairs Trading are documented in this chapter.

Let me explain the main idea behind the pairs trading strategy. The general algorithm for investing in the marketplace is to sell overvalued securities and buy the undervalued ones. However, it is possible to determine that a security is overvalued or undervalued only if we also know the true value of the security in absolute terms. But, this is very difficult to do. Pairs trading attempts to resolve this using the idea of relative pricing; that is, if two securities have similar characteristics, then the prices of both securities must be more or less the same.

2.1 Preliminary Concepts

Time Series Data

A time series is a sequence of observations in chronological order. In the chapter VI, there are three statistical models for time series. These models are extensively used in econometric, business forecasting, and many scientific applications.

A stochastic process is a sequence of random variables and can be viewed as the ”theoretical” or ”population” analog of a time series—on the other hand, a time series can be studied a sample from the stochastic process. ”Stochastic” is a synonym for random.

Stationary Processes

When a time series process is observed, the oscillations seem random, but often with the same type of stochastic behavior from one time period to the next. for instance, returns on stocks or changes in interest rates can be very different from the previous year, but the mean, standard deviation, and other statistical properties often are similar from one year to the next. Similar, the demand for many customer products, such as sunscreen, winter coats, and electricity, has random as well as seasonal variation, but each summer is similar to past summers, each winter to past winter, at least over shorter time periods. Stationary stochastic processes are probability models for time series with time-invariant behavior.

A process is said to be strictly stationary if all aspects of its behavior are unchanged by shifts in time (Ruey S. Tsay., 2002). Mathematically, stationary is defined as the requirement that for every m and n, the distributions of $Y_1, \ldots, Y_n$ and $Y_{1+m}, \ldots, Y_{n+m}$ are the same; that is, the probability distribution of a sequence of n observations does not depend on their time origin. Strictly stationary is a very strong assumption, because it requires that ”all aspects” of behavior be constant in time. A process is weakly stationary if its mean, variance, and covariance are unchanged by time shifts. More accurately, $Y_1, Y_2, \ldots$ is a weakly stationary process if

- $E(Y_i) = \mu$ (a constant) for all i;
- $Var(Y_i) = \sigma^2$ (a constant) for all i; and
- $Corr(Y_i, Y_j) = \rho(|i - j|)$ for all i and j for some function $\rho(h)$.

Thus, the mean and the variance do not change with time and the correlation between two observations depends only on the lag, the time distance between them.

The function ρ is called the *autocorrelation function* of the process. The covariance between Y_t and Y_{t+h} is denoted by $\gamma(h)$ and $\gamma(\cdot)$ is called *autocovariance function.*

As mentioned, many financial time series are not stationary, but often the changes in them, perhaps after they have been log transformed, are stationary.

Correlation and Autocorrelation Function

The correlation coefficient (Ruey S. Tsay., 2002) between two random variables X and Y is defined as

$$\rho_{x,y} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} = \frac{E[(X-\mu_x)(Y-\mu_y)]}{\sqrt{E(X-\mu_x)^2 E(Y-\mu_y)^2}}, \tag{2.1}$$

where μ_x and μ_y are the mean of $X$and Y, respectively, and it is assumed that the variances exist. The strength of linear dependence between X and Y is measured by this coefficient, and it can be shown that $-1 \leq \rho_{x,y} \leq 1$ and $\rho_{x,y} = \rho_{y,x}$. The two random variables are uncorrelated if $\rho_{x,y} = 0$.

2.1.1 Cointegration

Cointegration analysis is a technique that is regularly applied in econometrics (Carmona, R.,2014,Ruppert, D. ,2011). In finance it can be used to find trading strategies based on mean-reversion.

Suppose one could find a stock whose price series was stationary and therefore mean-reverting. This would be a wonderful investment opportunity. Whensoever the price was below the mean, one could buy the stock and realize a profit when the price returned to the mean. In addition, one could realize profits by selling short whenever the price was above the mean. Sometimes one can find two or more assets with prices so closely connected that a linear combination of their prices is stationary. Then, a portfolio using as portfolio weights the *cointegrating vector*, which is the vector of coefficients of this linear combination, will have a stationary price. Cointegration analysis is a means for finding cointegration vectors. In 1987, Engle and Granger first mentioned cointegration in their work that won the Nobel Prize 2003 for economics. Conintegration has many application in macroeconomic analysis since then. Recently, it has performed more and more noticeable role in funds management and portfolio construction. As the statistical properties of cointegration, it is an attractive in application for academics and practitioners.

Two time series, $Y_{1,t}$ and $Y_{2,t}$, are cointegrated if each is $I(1)$ but if there exists a λ such that $Y_{1,t} - \lambda Y_{2,t}$ is stationary.

Consider a set of economic variables $y_{i,t}, i = 1, \ldots, p$, in long-run equilibrium when

$$\beta_1 y_{1,t} + \beta_2 y_{2,t} + \ldots + \beta_p y_{p,t} = \mu + \epsilon_t \tag{2.2}$$

where p is number of variables in the cointegration equation, μ is the long-run equilibrium and ϵ_t is the cointegration error.

For simplicity, eq. 2.2 can be represented in matrix form as

$$\beta' y_t = \mu + \epsilon_t \tag{2.3}$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_p)'$ and $y_t = (y_{1,t}, y_{2,t}, \ldots, y_{p,t})'$.

The cointegration error is the deviation from the long-run equilibrium and can be represented by

$$\epsilon_t = \beta' y_t - \mu. \tag{2.4}$$

The equilibrium is only significant if the residual series or cointegration error ϵ_t is stationary.

As previously, Price series that are cointegrated can be used in statistical arbitrage. Unlike pure arbitrage, statistical arbitrage means an opportunity where a profit is only likely, not guaranteed. Pairs trading uses pairs of cointegrated asset prices and has been a popular statistical arbitrage technique. Pairs trading requires the trader to find cointegrated pairs of assets, to select from these the pairs that can be traded profitably after accounting for transaction costs, and finally to design the trading strategy which includes the buy and sell signals.

2.1.2 Mean Reversion

There are many definitions of mean reversion. Generally, mean reversion is an asset model, which presents that the asset price tens to fall (rise) after hit a maximum (minimum) (Premanode, B., 2013). The mean reversion process is a spread, but the variance does not growing in proportion to the time interval. The basic mean reversion model is the (arithmetic) Ornstein and Uhlenbeck(1930), a stochastic process that express the speed of a massive Brownian particle under the influence of friction. However, this process is stationary, Gaussian and Markovian.

The time series that tend to oscillate about the mean of the series; that is, they exhibit *mean reversion.*

Theoretical Considerations Related to Data Classification Using Mean Reversion and CV

In 2013, Premanode, B., Vonprasert, J., and Toumazou, C. proposed a novel multiclass algorithm for using the SVM family, known as a 'multiclass kernel'. The typical curve of stock prices tends to to oscillate about the mean of the series, so the point of reversal can be used to determine changes in its direction, i.e., from up to down, and vice versa. Then the datasets are partitioned at the reversal point. As the standard deviations of a nonstationary datasets are not the same, the datasets between each reversal points are measured. The procedure for using mean reversion and CV are in the following [VII]:

i) Compute the mean $\mu_n(t)$ of random variables $X_n(t)$.

ii) Compute the variance $V_n(t)$ of $X_n(t)$.

iii) After normalizing each $V_n(t)$ using $\mu_n(t)$, $\frac{V_n(t)}{\mu_n(t)}$.

iv) In an upward scenario where $V_1(t) < V_2(t), \ldots, n$, or a downward scenario where

   a) if $\frac{V_2(t)}{\mu_2(t)} < \frac{V_1(t)}{\mu_1(t)}$ or $\frac{V_2(t)}{\mu_2(t)} > \frac{V_1(t)}{\mu_1(t)}$, mark the intercept point on the x-axis and denote it as M_1, i.e., the value is $X_{rn}(t)$ where $r = 1, 2, \ldots, c$ and c is the last class generated by CV or

   b) if $\frac{V_2(t)}{\mu_2(t)} = \frac{V_1(t)}{\mu_1(t)}$, ignore to mark any intercept point on the x-axis.

v) Repeat iv) and stop when $\frac{V_n(t)}{\mu_n(t)}$ becomes the last data point (n). Next, plot $M_2, \ldots, M_n$.

vi) Compute CV for the data $X_{rn}(t)$ between the blocks of $M_1, M_2, \ldots, M_n$ where $n-1$ is the number of partitions/blocks.

The coefficient of variance (CV) that is used in the procedure above is represented by

$$CV_i = \frac{\rho_i}{\mu_i} \tag{2.5}$$

where ρ_i represents standard deviation and μ_i represents mean.

The original datasets $X_{rn}(t)$ was classified into different CV classes.

2.2 Pairs Trading

Pairs trading involves selling the higher-priced security and buying the other one with the idea that the mispricing will correct itself in the future. Our theoretical explanation for the co-movement of security prices stems from arbitrage pricing theory (APT). According to APT, if two securities have exactly the same risk factor exposures, then the expected return of the two securities for a given time frame is the same.

The traders wait for weakness in the correlation, and then go long on the lower-value while simultaneously going short on the over-value one, closing the position as the relationship returns to its mean. The strategy's profit is calculate from the difference in price change between the two instruments, rather than from the direction in which each moves. It is possible for the traders to profit during a variety of market conditions, including periods when the market goes up, down or sideways, and during periods of either low or high volatility.

2.2.1 The Benefits of Pairs Trading Strategy?

Pairs Trading is a market neutral strategy in its most fundamental form. The market neutral portfolios are constructed using just a pair of highly correlated instruments such as two stock, exchange-traded funds(ETFs), currencies, commodities or options, which is consist of a long position in one security and a short position in the other, in a predetermined ratio. At any given time, the portfolio is associated with a quantity called the *spread.* The quoted prices of the two securities and form a time series are used to calculate this quantity. Pairs Trading involves putting on position when the spread is substantially away from its mean value, with the expectation that the spread will revert back. The positions are then reversed upon convergence. There are two versions of pairs trading in the equity markets; namely, statistical arbitrage pairs and risk arbitrage pairs.

Statistical arbitrage pairs trading is based on the idea relative pricing. The underlying premise in relative pricing is that stocks with similar characteristics must be priced more or less the same. The spread in this case may be thought of as the degree of mutual mispricing. The greater the spread, the higher the magnitude of mispricing and greater the profit potential.

Risk arbitrage pairs occur in the context of a merger between two companies. The terms of the merger agreement establish a strict parity relationship between the values of the stocks of the two firms involved. The spread in this case is the magnitude of the deviation from the defined parity relationship. If the merger between the two companies is deemed a certainty, the the stock prices of the two firms must satisfy the parity relationship, and the spread between them will be zero. However, there is usually a certain level of uncertainty on the successful completion of merger after the announcement, because of various reasons like antitrust regulatory issues, proxy battles, and competing bidders, etc. This uncertainty is reflected in the nonzero value for the spread. Risk arbitrage involves taking on this uncertainty as risk and capturing the spread value as profits. Thus, unlike the case of statistical arbitrage pairs, which is based on valuation consideration, risk arbitrage trade is based strictly on a parity relationship between the prices of the two stocks.

2.2.2 History of Pairs Trading

An early attempt at Pairs Trading is attributed to Wall Street quant Nunzio Tartaglia, who was at Morgan Stanley in the mid 1980s (Vidyamurthy, G., 2004). At the time, he gathered a group of mathematicians, physicist, and computer scientists. Their mission was to developed by the group were automated to the point where they could generate trades in a mechanical fashion and, if needed, execute them seamlessly through automated trading systems. At that time, trading systems of this kind were considered the cutting edge of technology.

One of the techniques they used for trading involved trading securities in pairs. The process involved identifying pairs of securities whose prices tended to move together. Whenever an abnormality in the relationship was noticed, the pair would be traded with the idea that the abnormality would correct itself. This came to be known on the street as p̈airs trading.Tartaglia and his group employed pairs trading with great success in 1987. The group, however, disbanded in 1989. Member of the group found themselves in various other trading firms, and knowledge of the idea of pairs trading gradually spread. Pairs trading has since increased in popularity and has become a common trading strategy used by hedge funds and institutional investors.

One of the most popular market neutral strategies is Pairs Trading. The market neutral portfolios are constructed using two securities, composing of a long position in one security and a short position in the other, i.e., to sell the overvalued securities and buy the undervalued ones. At any given time, the portfolio is associated with a quantity called the spread. Pairs Trading relates putting on positions when the spread is significantly away from mean value, with the expectation that the spread will revert back. The positions are then reversed upon convergence. However, it is possible to determine that a security is overvalued or undervalued only if we also know the true value of the security in absolute terms, but, this is very difficult to do. Pairs trading attempts to resolve this using the idea of relative pricing; that is, if two securities have similar characteristics, then the prices of both securities must be more or less the same.

Pairs trading involves selling the higher-priced security and buying the lower-priced security with the idea that the mispricing will correct itself in the future.

The strategy involves assuming a long-short position when the spread is substantially away from the mean. This is done with the expectation that the mispricing is likely to collect itself. The position is then reversed and profits made when the spread reverts back.

Layout for Pairs Trading Strategy Design

The steps related are in the following:

1. Identify stock pairs that could potentially be cointegrated.

2. Once the potential pairs are identified, The proposed hypothesis that the stock pairs are indeed cointegrated based on statistical evidence from historical data is verified. Determining the cointegration coefficient and examining the spread time series to ensure that it is stationary and mean reverting are involved.

3. Then examining the cointegrated pairs to determine the delta.

2.2.3 Trading Strategy

The strategy starts with considering the stock that have historical in the same pattern of trading. If there is a deviation from the historical mean, this creates a trading opportunity, that can be exploited. Profit are made when the price relationship is restored.

For executing the strategy, a trader need a couple of trading rules to follow, i.e., to clarify when to open or close a portfolio. The general rule will be open a position when the Standard Deviation of each price become different significantly and close it when the ratio returns to the mean.

2.3 Pairs Trading Approaches

There are four main methods to implement pairs trading: the distance method(Gatev et al, 2006), the stochastic spread method (Elliot,Van Der Hoek, and Malcolm, 2004), the combined forecasts and multi-criteria decision methods (MCDM) (Huck, 2010) and the co-integration method(Vidyamurthy, 2004).

The Distance method

In the distance method, the co-movement in a pair is measured by the distance, or the sum of squared differences between the two normalized price series. The distance approach purely uses a statistical relationship between a pair of securities.

The Stochastic spread method

The stochastic spread approach explicitly models the mean reverting of the spread in a continuous time setting. Pairs trading based on this approach relies on an assumption that the spread can follow an Ornstein-Uhlenbeck process which actually is an AR(1) process in a continuous term.

The Combined forecasts and multi-criteria method

The Combined forecasts approach was proposed by Huck (2009, 2010). This method is based on three phases: forecasting, ranking, and trading. This approach differs from the others essentially in that it is developed without reference to any equilibrium model. Huck (2009, 2010) explained that the method provides much more trading possibilities and could detect the birth of the divergence which the other approaches cannot consider.

The Cointegration method

The co-integration method [VII] is an attempt to parameterize Pairs Trading, by exploring the possibility of co-integration. Co-integration is the phenomenon that two time series that are both integrated of order d, can be linearly combined to produce a single time series that is integrated of order $d - b, b > 0$, the most simple case of which is when $d = b = 1$.

Generally speaking, the framework is as follows: first, choose two co-integrated stock price series, then open a long/short position when stocks deviate from their long term equilibrium and finally, close the position after convergence or at the end of the trading period.

Consider two shares whose prices are integrated of order 1. P_i^t refers to the price of the ith asset called A_i at time t. If the share prices P_1^t and P_2^t are co-integrated , co-integration coefficients 1 and β exist so that a co-integration relationship can be constructed as follows:

$$P_1^t - \beta P_2^t = \epsilon_t, \tag{2.6}$$

where ϵ_t is a stationary process. When a divergence (based on the standard deviation of ϵ_t) from the equilibrium state is observed, the trading involves buying one share 1 and selling β shares 2.

With the concepts on Data Classification Using Mean Reversion and CV, the author envision to introduce the Mean Reversion and CV as apart of a new algorithm of Pairs Trading. Before presenting the new algorithm for Pairs Trading, the next chapter will explain forecasting methods and test statistic, which will be using for predicting the paired stocks datasets.

CHAPTER III

THE FORECASTING METHODS

One of the description of the word ’forecasting’ is an estimation of a future trend by inspection and analyzing the known information. Forecasting informs the decisions made by an organisation, i.e., market trends; economic and social analysis; capital and financial market; scheduling of product, transport, personnel and cash; acquiring resources; and determining resource requirements (Makridakis, Wheelwright, and Hyndman, 1998).

This chapter classifies the methods of forecasting in Section 3.1 and it describes the basic steps during forecasting tasks in Section 3.2.

The classical forecasting problem may be started as follows. The historical time series data with the values up to the present value are given. Then, the value of the next time step value has to be predicted as close as possible.

3.1 Classification of Forecasting Methods

After reviewing many web so far, the general classifications of forecasting methods are as follows; i) qualitative vs quantitative; ii) naïve; iii) reference class forecasting, which was developed by Flyvbjerg (2008) to eliminate or decrease bias when forecasting by concentrating on distribution of information about the past; iv) time series based on many models, i.e., Kalman filtering, moving average (MA), exponential smoothing, autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), extrapolation, linear and nonlinear prediction, trend estimation, etc.; v) casual/econometric; vi) artificial intelligence, e.g., artificial neural networks, group methods of data handling, support vector machines, data mining, machine learning, and pattern recognition.

Nevertheless, the most common categories of forecasting methods described by Makridakis, Wheelwright, and Hyndman (1998) are the following.

3.1.1 Qualitative

This procedure use expert view and combined experience to unlock the unknown future where a curious issue is considered. This category may not need a historical series of data.

3.1.2 Quantitative

The actual numbers, sufficient information and previous experience are used for the future trend estimation in this procedure. There are two major types, time series that predict discrete or continuous historical patterns based on periods of time and explanatory approaches that attempt to correlate two or more variables that need to be predicted.

3.2 Basics Steps during Forecasting Tasks

The forecasting methods in this research are based on quantitative methods and the basic steps as follows.

Step 1: Problem definition

The goal is to address how we can improve the accuracy of forecasting nonlinear nonstationary time series data using the prediction models which are shown in Chapter VI.

Step 2: Collection information

Nonlinear, nonstationary time series data was used in this study. These datasets were daily trading data recorded in the Global Dow . They contain daily stock prices over a 10-year period from 1 August 2002.

Step 3: Preliminary analysis

This step contains general methods for testing for parametric and nonparametric and testing multicolinearity tests.

Step 4: Choosing and fitting models

The comparison of selected models can be achieved using Akaike's information criterion (AIC), which was introduced by Hirotugu (1974). AIC is not a test of the model in the sense of hypothesis testing; it is a tool for model selection. The ranking from the poorest to the best model is given by the lowest AIC. AIC attempts to estimate the best model that explains data fitted with a minimum of free parameters, otherwise there may be over fitting.

step 5: Performance measurement

After the completion of step 4, the correct models are selected and finally they measure the performance using the standard statistical measures and comparative methods, i.e., μ, σ, MPE, MAPE, MSE, RMSE, AIC, BIC and Accuracy count.

Where the accuracy count is the upward and downward movements relative to the mean reversion points in the graphs of outcomes of the simulations compared with the graph of the original datasets.

Given a dataset, several competing models may be ranked according to their

Table 3.1 Performance measurements

Standard test statistic	Comparative method
Mean (μ)	Akaike information criterion (AIC)
Standard deviation (σ)	Bayesian Information criterion (BIC)
Variance (σ^2)	Accuracy count
Mean percentage error (MPE)	
Mean absolute percentage error (MAPE)	
Mean square error (MSE)	
Root Mean square error (RMSE)	
Coefficient of determination (R^2)	

information criterion. The AIC equation is expressed as follows:

$$AIC = 2K{-}2ln(L) \tag{3.1}$$

where K is the number of parameters in the statistical model and L is the maximized value of the likelihood function for the estimated model. Unless the sample size (n) is large with respect to the number of estimated parameters (K), use of AICc is recommended.

$$AIC_c = -2ln(L(\Theta|y)) + 2K\left(\frac{n}{n-K-1}\right) \tag{3.2}$$

Generally, the AICc is used when the ratio of n/K is small (less than 40), based on K from the global (most complicated) model.

3.3 Cross Validation Methods

As Schneider and Moore's study in 1997, cross-validation is a model evaluation method that splits training and test data, in which the test data is used to test the performance after the statistic models train or computed the training data. The three main methods to approach cross-validation are in the followings.

i) *Holdout*

The Holdout method is the simplest type of cross-validation. The datasets is separated into two sets: the training set and the test set. The estimation model fits the training set only and leaves the test data blind.

ii) *K-fold*

K-fold was proposed to improve the Holdout method. The K-fold method divides the whole datasets into k subsets and uses the Holdout method k times. In each subset, the training data are computed using the model and tested with the test data.

iii) *Leave-one-out*

This method applies bootstrap sampling by taking one particle (data unit) out of the overall training and test datasets whereas the remaining data are used for reference. The advantage is that the accuracy of the outcome but this is traded-off by the massive computational power requirements when handling large input datasets. Moreover, this method was designed only for model evaluation or in-sample forecasting so it is rather difficult to apply this method to test forecasting.

With the 5-step of the forecasting tasks, the data are usable to enter to any process. The next chapter details the data that will be used in the Chapter V and VI.

CHAPTER IV

THE DATA

Before fitting any model, data testing should be completed. This chapter introduces the datasets that were composed of 150 daily stocks recorded in the Global Dow.

The Global Dow is an equal-weighted stock index consisting of the stocks of 150 top companies from around the world as selected by Dow Jones editors based on the companies' long history of success and popularity among investors. The Global Dow is designed to reflect the global stock market and gives preferences to companies with a global reach.

4.1 Data Preparation

The datasets used in this study are daily stock prices that were composed of 150 daily stocks recorded in the Global Dow. The datasets contain daily stock prices over a 10-year period from 1 August 2002 (total of 3961 datasets). Saturday and Sunday price observations were removed prior to the analysis to avoid any bias in the results from weekend market closures.

In practice, financial data are time series which are discrete time continuous state process (Ullrich, 2009).

4.2 Normality Test for a Nonlinear Distribution

Since the stock prices and other financial information are normally nonlinear, the following tests are used to ensure that the variables specified in Section 4.1 are not linear, which affected in a good model selection that can be used for prediction in the Chapter VI.

4.2.1 Anderson Darling Test

The Anderson Darling test is a statistical test of whether a given sample of data is drawn from a specific distribution, e.g., the normal distribution. This test makes use of the specific distribution to calculate critical values. The Anderson-Darling statistic can be used to compare how well a data set fits different distributions.

The two hypotheses for the Anderson-Darling test for the normal distribution are given below:

- H0: The data follows the normal distribution
- H1: The data do not follow the normal distribution

The null hypothesis is that the data are normally distributed; the alternative hypothesis is that the data are non-normal.

The Anderson-Darling statistic is given by the following formula:

$$AD = -n - \frac{1}{n}\sum_{i=1}^{n}(2i-1)[lnF(X_i) + ln(1 - F(X_{n-i+1}))], \qquad (4.1)$$

where n is sample size, $F(X)$ is cumulative distribution function for the specified distribution and i is the i^{th} sample when the data is sorted in rising order.

4.2.2 Kolmogorov-Smirnov Test

In 1974, Stephens stated that the Kolmogorov–Smirnov test (K-S test) is a nonparametric test of the equality of continuous, one-dimensional probability distributions, which can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test). The K-S statistic quantifies the distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. This test can be modified to serve as a goodness of fit test.

The two hypotheses for the Kolmogorov–Smirnov test for the normal distribution are given below:

- H0: The data follows the normal distribution
- H1: The data do not follow the normal distribution

The null hypothesis is that the sample are normally distributed or that the sample are drawn from the same distribution (in the two-sample case).

In this case , samples are standardized and compared with a standard normal distribution by setting the mean and variance of the reference distribution equal to the sample estimates. The empirical distribution F_n for n is independently and identically distributed (i.i.d.) with observations X_i, is defined as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{X_i} \leq x, \tag{4.2}$$

where I_{X_i} is the indicator function, which is equal to 1 if $X_i \leq x$ and equal to 0 otherwise. The K-S static for a given c.d.f. F(x) is

$$D_x = \sup_x |F_n(x) - F(x)|, \tag{4.3}$$

where $\sup_x$ is the supremum of the set of distances. By Glivenko–Cantelli theorem, if the sample comes from the distribution $F(x)$, then D_n converges to 0 almost certainly (Wellner, 1981). However, as many researches, the K-S test is less powerful for testing normality than the Anderson-Darling test (Stephen, 1974) and that it requires a relatively large number of data points to reject the null hypothesis appropriately.

4.2.3 Pearson's chi-squared Test

Two random variables x and y are independent if the probability distribution of one variable is not affected by the presence of another. Assume f_ij is the observed frequency count of events belonging to both the i^{th} category of x and the j^{th} category of y. Moreover, assume e_{ij} to be the corresponding expected count if x and y are independent. The null hypothesis of the independence assumption is rejected if the p-value of the following Chi-squared test statistic is less than a given significance level (Moor, 1986).

$$\chi^2 = \sum_{i,j}^{n} \frac{(f_{ij} - e_{ij})^2}{e_{ij}}. \tag{4.4}$$

4.3 Unit Root Test for a Nonlinear Distribution

Financial time series such as stock prices sometimes can be described as a random walk process which is a non-stationary process with a unit root. There are several ways to test whether the series is stationary or non-stationary with a unit root. The well-know one is Dickey-Fuller (DF) test (Dickey and Fuller, 1979, Fuller, 1976). It test the null hypothesis that a series does contain a unit root, i.e., it is non-stationary, against the alternative of stationary. There are other tests, such as CRDW test (Sargan and Bhargava, 1983) based on the usual Durbin-

Watson statistic; and the non-parametric tests developed by Phillips and Perron based on the Z-test (Phillips and Perron, 1988), which involves transforming test statistic to eliminate autocorrelation in the model. As the DF test's simplicity and its more general nature, it is more popular than others.

4.3.1 Augmented Dickey-Fuller Test

The Augment Dickey-Fuller test (ADF), an augmented version of the Dickey–Fuller test for a larger and more complicated set of time series models, is a test for a unit root in a time series sample. The ADF is a negative number and when it is more negative, there is a good reason to reject the hypothesis that there is a unit root at some level of confidence. The testing procedure for the ADF test is the same as that for the Dickey–Fuller test when it is applied to the model (Dickey and Fuller, 1981), which is given by

$$\Delta y_t = \alpha + \beta_i + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \ldots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t, \tag{4.5}$$

where α is a constant, β is the coefficient on a time trend and p is the lag order of the autoregressive process. Specifying the constraints $\alpha = 0$ and $\beta = 0$ corresponds to modelling a random walk whereas using the constraint $\beta = 0$ corresponds to modelling a random walk with drift. The ADF conception with lags of order p allows for higher-order autoregressive processes. When the test is applied, the lag length p has to be defined and this can be fitted using AIC. In short, AIC is a tool for model selection and also for selecting the lagged length of eq. 4.5. Given a dataset, several competing models are ranked by their information criterion. The AIC equation is defined as follows:

$$AIC = 2k - 2ln(L), \tag{4.6}$$

where k is the number of parameters in the statistical model and L is the maximized value of the likelihood function for the estimated model. The unit root test is then fulfilled under the null hypothesis $\gamma = 0$ against the alternative hypothesis of $\gamma < 0$. A value for the test static can be calculated using the equation as follows:

$$DF_{\tau} = \frac{\hat{\gamma}}{SE(\hat{\gamma})}, \tag{4.7}$$

where SE is the standard error, equaling $\frac{S.D}{\sqrt{n}}$. Accepting the null hypothesis implies the presence of a unit root where the test statistic is less than (a larger negative) the critical value.

Table 4.1 presents the blue chip stocks of companies with a national reputation for reliability, quality, and the capability to operate profitably under extreme market conditions. The stocks are among the most widely and actively traded ones. The datasets contain daily stock prices over a 10-year period from 1 August 2002, i.e., 3,961 days. Saturday and Sunday price observations were removed prior to the analysis to avoid any bias in the results from weekend market closures.

In conclusion, 134 datasets of 150 datasets collected from Global Dow were use in this study.The new novel Pairs Trading will be presented, in the Next Chapter. The datasets will be used in the next two Chapter, Chapter V and VI.

Table 4.1 The 134 listed companies in Global Dow index in the year 2013

	Company	Countries	BB Ticker
1	3M Co.	U.S.	MMM US Equity
2	ABB Ltd.	Switzerland	ABB SS Equity
3	Abbott Laboratories	U.S.	ABT US Equity
4	Alcoa Inc.	U.S.	AA US Equity
5	Allianz SE	Germany	ALV GR Equity
6	Amazon.com Inc.	U.S.	AMZN US Equity
7	America Movil S.A.B. de C.V. Series L	Mexico	AMXL MM Equity
8	American Express Co.	U.S.	AXP US Equity
9	Amgen Inc.	U.S.	AMGN US Equity
10	Anglo American PLC	U.K.	AAL LN Equity
11	Anheuser-Busch InBev N.V.	Belgium	ABI BB Equity
12	Apple Inc.	U.S.	AAPL US Equity
13	ArcelorMittal	France	ARCELOR LX Equity
14	Assicurazioni Generali S.p.A.	Italy	G IM Equity
15	Astrazeneca PLC U.K.	U.K.	AZN LN Equity
16	AT&T Inc.	U.S.	T US Equity
17	BAE Systems PLC	U.K.	BA/ LN Equity
18	Banco Bilbao Vizcaya Argentaria S.A.	Spain	BBVA SM Equity
19	Banco Santander S.A.	Spain	SAN SM Equity
20	Bank of America Corp.	U.S.	BAC US Equity
21	Bank of New York Mellon Corp.	U.S.	BK US Equity
22	BASF SE	Germany	BAS GR Equity
23	Baxter International Inc.	U.S.	BAX US Equity
24	Bharti Airtel Ltd.	India	BHARTI IN Equity
25	BHP Billiton Ltd.	Australia	BHP AU Equity
26	BNP Paribas S.A.	France	BNP FP Equity
27	Boeing Co.	U.S.	BA US Equity
28	BP PLC	U.K.	BP/ LN Equity
29	Bridgestone Corp.	Japan	5108 JP Equity
30	Canon Inc.	Japan	7751 JT Equity
31	Carnival Corp.	U.S.	CCL US Equity
32	Carrefour S.A.	France	CA FP Equity
33	Caterpillar Inc.	U.S.	CAT US Equity
34	Chevron Corp.	U.S.	CVX US Equity
35	China Construction Bank Corp.	China	601939 CH Equity
36	China Mobile Ltd.	Hong Kong	941 HK Equity
37	China Petroleum & Chemical Corp.	China	600028 CH Equity

	Company	Countries	BB Ticker
38	China Unicom (Hong Kong) Ltd.	Hong Kong	762 HK Equity
39	Cisco Systems Inc.	U.S.	CSCO US Equity
40	CLP Holdings Ltd.	Hong Kong	2 HK Equity
41	Coca-Cola Co.	U.S.	KO US Equity
42	Colgate-Palmolive Co.	U.S.	CL US Equity
43	Compagnie de Saint-Gobain S.A.	France	SGO FP Equity
44	Companhia Energetica de Minas Gerais-CEMIG Pr	Brazil	CMIG4 BZ Equity
45	ConocoPhillips	U.S.	COP US Equity
46	Credit Suisse Group	Switzerland	CSGN VX Equity
47	Daimler AG	Germany	DAI GR Equity
48	Deere & Co.	U.S.	DE US Equity
49	Deutsche Bank AG	Germany	DBK GR Equity
50	E.I. DuPont de Nemours & Co.	U.S.	DD US Equity
51	E.ON AG	Germany	EOAN GR Equity
52	eBay Inc.	U.S.	EBAY US Equity
53	EDP-Energias de Portugal S.A.	Portugal	EDP PL Equity
54	Esprit Holdings Ltd.	Hong Kong	330 HK Equity
55	Express Scripts Inc.	U.S.	ESRX US Equity
56	Exxon Mobil Corp.	U.S.	XOM US Equity
57	FedEx Corp.	U.S.	FDX US Equity
58	First Solar Inc.	U.S.	FSLR US Equity
59	Freeport-McMoRan Copper & Gold Inc.	U.S.	FCX US Equity
60	Gazprom OAO ADS	Russia	GAZPROM RU Equity
61	GDF Suez S.A.	France	GSZ FP Equity
62	General Electric Co.	U.S.	GE US Equity
63	Gilead Sciences Inc.	U.S.	GILD US Equity
64	GlaxoSmithKline PLC	U.K.	GSK US Equity
65	Goldman Sachs Group Inc.	U.S.	GS US Equity
66	Google Inc. Cl A	U.S.	GOOG US Equity
67	Hewlett-Packard Co.	U.S.	HPQ US Equity
68	Home Depot Inc.	U.S.	HD US Equity
69	Honda Motor Co. Ltd.	Japan	7267 JP Equity
70	Honeywell International Inc.	U.S.	HON US Equity
71	HSBC Holdings PLC (UK Reg)	U.K.	HSBA LN Equity
72	Hutchison Whampoa Ltd.	Hong Kong	13 HK Equity
73	Industrial & Commercial Bank of China Ltd.	China	601398 CH Equity
74	Infosys Technologies Ltd.	India	INFO IN Equity
75	Intel Corp.	U.S.	INTC US Equity

	Company	Countries	BB Ticker
76	International Business Machines Corp.	U.S.	IBM US Equity
77	Johnson & Johnson	U.S.	JNJ US Equity
78	JPMorgan Chase & Co.	U.S.	JPM US Equity
79	Komatsu Ltd.	Japan	6301 JP Equity
80	Kraft Foods Inc. Cl A	U.S.	KRFT US Equity
81	L.M. Ericsson Telephone Co. Series B	Sweden	ERICB SS Equity
82	LG Electronics Inc.	South Korea	066570 KS Equity
83	LVMH Moet Hennessy Louis Vuitton	France	MC FP Equity
84	McDonald's Corp.	U.S.	MCD US Equity
85	Medtronic Inc.	U.S.	MDT US Equity
86	Merck & Co. Inc.	U.S.	MRK US Equity
87	Microsoft Corp.	U.S.	MSFT US Equity
88	Mitsubishi Corp.	Japan	8058 JP Equity
89	Mitsubishi UFJ Financial Group Inc.	Japan	8306 JP Equity
90	Mitsui & Co. Ltd.	Japan	8031 JP Equity
91	Mizuho Financial Group Inc.	Japan	8411 JP Equity
92	Monsanto Co.	U.S.	MON US Equity
93	NASDAQ OMX Group Inc.	U.S.	NDAQ US Equity
94	National Australia Bank Ltd.	Australia	NAB AU Equity
95	National Grid PLC	U.K.	NG/ LN Equity
96	Nestle S.A.	Switzerland	NESN VX Equity
97	News Corp. Cl A	U.S.	NWSA US Equity
98	Nike Inc. Cl B	U.S.	NKE US Equity
99	Nintendo Co. Ltd.	Japan	7974 JP Equity
100	Nippon Steel Corp.	Japan	5401 JP Equity
101	Nokia Corp.	Finland	NOK1V FH Equity
102	Novartis AG	Switzerland	4856075Z MC Equity
103	Panasonic Corp.	Japan	6752 JP Equity
104	PetroChina Co. Ltd.	China	601857 CH Equity
105	Petroleo Brasileiro S/A Pref	Brazil	PETR4 BZ Equity
106	Pfizer Inc.	U.S.	PFE US Equity
107	Philip Morris International Inc.	U.S.	PM US Equity
108	Potash Corp. of Saskatchewan Inc.	Canada	POT CN Equity
109	Procter & Gamble Co.	U.S.	PG US Equity
110	Reliance Industries Ltd.	India	RIL IN Equity
111	Renewable Energy Corp. ASA	Norway	REC NO Equity
112	Research in Motion Ltd.	Canada	BB CN Equity
113	Rio Tinto PLC	U.K.	RIO LN Equity

	Company	Countries	BB Ticker
114	Roche Holding AG Part. Cert.	Switzerland	RO SW Equity
115	Royal Bank of Canada	Canada	RY CN Equity
116	Royal Dutch Shell PLC A	U.K.	RDSA LN Equity
117	Samsung Electronics Co. Ltd.	South Korea	005930 KS Equity
118	SAP AG	Germany	SAP GR Equity
119	Schlumberger Ltd.	U.S.	SLB US Equity
120	Seven & I Holdings Co. Ltd.	Japan	3382 JP Equity
121	Siemens AG	Germany	SIE GR Equity
122	Societe Generale S.A.	France	GLE FP Equity
123	Sony Corp.	Japan	6758 JP Equity
124	Southwest Airlines Co.	U.S.	LUV US Equity
125	SunPower Corp. Cl A	U.S.	SPWR US Equity
126	Suntech Power Holdings Co. Ltd. ADS	China	SUPOHZ CH Equity
127	Taiwan Semiconductor Manufacturing Co. Ltd.	Taiwan	2330 TT Equity
128	Takeda Pharmaceutical Co. Ltd.	Japan	4502 JP Equity
129	Tata Steel Ltd.	India	TATA IN Equity
130	Telefonica S.A.	Spain	TEF SM Equity
131	Tesco PLC	U.K.	TSCO LN Equity
132	Time Warner Inc.	U.S.	TWX US Equity
133	Toshiba Corp.	Japan	6502 JP Equity
134	Total S.A.	France	FP FP Equity
135	Toyota Motor Corp.	Japan	7203 JP Equity
136	Travelers Cos. Inc.	U.S.	TRV US Equity
137	UBS AG	Switzerland	UBSN VX Equity
138	UniCredit S.p.A.	Italy	UCG IM Equity
139	United Parcel Service Inc. Cl B	U.S.	UPS US Equity
140	United Technologies Corp.	U.S.	UTX US Equity
141	Vale S.A. Pref A	Brazil	VALE5 BZ Equity
142	Veolia Environnement S.A.	France	VIE FP Equity
143	Verizon Communications Inc.	U.S.	VZ US Equity
144	Vestas Wind Systems A/S	Denmark	VWS DC Equity
145	Vinci S.A.	France	DG FP Equity
146	VISA Inc. Cl A	U.S.	V US Equity
147	Vodafone Group PLC	U.K.	VOD LN Equity
148	Wal-Mart Stores Inc.	U.S.	WMT US Equity
149	Walt Disney Co.	U.S.	DIS US Equity
150	Wells Fargo & Co.	U.S.	WFC US Equity

CHAPTER V

THE PAIRS TRADING MODEL

As in Chapter II, the Pairs Trading was described. Pairs trading involves selling the higher-priced security and buying the lower-priced security with the idea that the mispricing will correct itself in the future. This newly invented technique provides a new set of risk mitigation by providing a buffer-trading zone when the paired stocks are changing their directions. In portfolio trading, it extends an opportunity for a highly correlated and paired stock to cross-trade with any lowly correlated and paired stock. Thus, the proposed model maximises returns and minimises risk of cointegrated Pairs Trading stocks. The proposed model employs Mean Reversion and Coefficient of Variance (CV) algorithm (Premanode, Vonprasert, and Toumazou, 2013). In this Chapter are proposed a new novel algorithm for Pairs Trading.

5.1 The proposed multi-class Pairs Trading Model

The methodology of this research based on Pairs Trading using Mean Reversion and CV. The Mean Reversion technique analyses any dataset whose distributions move from upward to downward direction and vice versa. Following, we introduce classification technique using coefficient of variance to grouping the stock indexes (variable datasets, and now called datasets), followed by the Mean Reversion Technique, which is the fundamental framework for creating multi-class in the algorithm.

In theory, the conventional co-integrated Pairs Trading method identifies two stocks that move in time series together and calculate a correlation between them. The model begins by normalising the datasets using the mean (μ) and standard deviation (σ) followed to co-integration them with Pearson's correlation coefficient(ρ), and it represents by

$$\rho_{x_i,y_i} = \frac{cov(x_i, y_i)}{\sigma_{x_i}, \sigma_{y_i}} = \frac{E[(x_i - \mu_{x_i})(y_i - \mu_{y_i})]}{\sigma_{x_i}\sigma_{y_i}} \tag{5.1}$$

where $cov(x_i, y_i)$ represents the covariance of x_i, and y_i, when $i = 1, 2, ..., n$. Following, we select the paired stocks in order from high to low.

Next, this research introduces the Mean Reversion and CV (Premanode, Vonprasert and Toumazou, 2013) to analyse and group the datasets. The Mean Reversion algorithm is expressed as follows:

i) Compute the mean $\mu_i(t)$ of $x_i(t)$, where $i = 1, 2, ..., n$.

ii) Compute the variance $V_i(t)$ of $x_i(t)$.

iii) By normalising each $V_i(t)$ using $\mu_i(t)$, we obtain $\frac{V_i(t)}{\mu_i(t)}$.

iv) Using the datasets $x_i(t)$ from the upward scenario,we calculate and plot $V_1(t) > V_2(t) > \ldots > V_{i-1}(t) > V_i(t)$.

v) The same process is applied to the downward scenario where $V_1(t) < V_2(t) < \ldots < V_{i-1}(t) < V_i(t)$.

vi) If $\frac{V_i(t)}{\mu_i(t)} = \frac{V_{i-1}(t)}{\mu_{i-1}(t)}$, ignore the calculation, but move the plot one step forward.

vii) Repeat the steps in items iv) to vi) and stop when $i = n$.

viii) We obtain a curve of $x_i(t)$ that marks points of local maxima and minima.

In the next process, we introduce the coefficient of variance (CV) to compute the datasets, at which is represented by

$$CV_i = \frac{\rho_i}{\mu_i} \tag{5.2}$$

where ρ_i represents standard deviation and μ_i represents mean. Consequent to applying the Mean Reversion and CV, we derived a number of groups of datasets and termed them to CV. Each CV may then have different normal distribution, reflecting different values for the paired stock indices. Following to plotting standard deviation, we divide the datasets into six classes in time series; namely, $CV_1, CV_2, CV_3, CV_4, CV_5$ and CV_6. we then plot the mean of CV_1 to CV_6 between the mean of CV_3 and CV_4. Hence, in the normal distribution, standard deviation of the CV_1 should be significantly deviated greater than the CV_2. Applying the same rationale, standard deviation of the CV_6 is significantly deviated greater than CV_5. In each CV, we calculate the return Pairs Trading (Perline, 2007) using Eq.(5.3). The co-integrated Pairs Trading formula is expresses as follows:

$$R_{CO} = \sum_{t=1}^{T}\sum_{i=1}^{n} R_i(t) \cdot I_i^{L\&S}(t) \cdot W_i + \left(\sum_{t=1}^{T}\sum_{i=1}^{n} Tc_i(t) \cdot \left[ln\left(\frac{1-C}{1+C} \right) \right] \right) \tag{5.3}$$

where $R_i(t)$ represents the real return of asset i at time t, calculated by $ln\left(\frac{P_i(t)}{P_i(t-1)}\right)$; $I_i^{L\&S}(t)$ represents the dummy variable with a value of 1 if a Long position is created for the asset i, a value of -1 if a short position is created, and 0 otherwise; $Tc_i(t)$ represents the dummy variable that takes a value of 1 if a transaction is made for the asset i at time t and 0 otherwise; C represents the transaction cost per operation (by percentage); T represents the number of observations on the whole trading period, and

$$W_i(t) = \frac{1}{\sum_{i=1}^{n} |I_i^{L\&S}(t)|} \qquad \text{for} \begin{cases} 1 & \text{if trade exist;} \\ 0 & \text{if no trade.} \end{cases} \tag{5.4}$$

where $W_i(t)$ is the weighting variable that controls for portfolio construction at time t, assuming that the same weight is applied to each transaction.

5.2 Benefits of the multi-class Pairs Trading

Since the co-integrated Pairs Trading is used to buying a stock, commodity or currency under the expectation that the asset will rise or fall in value from time to time. As a result, the long position is exercised when the curve of a paired stock is at high peak (maxima). Whereas, the short position is exercised when the paired stock is moving at the low peak (minima). With the proposed multiclass Pairs Trading, there are two extra benefits, which are follows:

i) By applying the proposed model to the historical trading datasets, we then found that a number of paired stocks could distribute to any CV, depending on there values of Mean Reversion and CV. An example is given that the highest correlated paired stock may locate in CV_1. Once the trade begins within any CV, we can exercise either long or short positions in time series until the existing CV starts to change the new CV. In the situation where the stock starts to diverge, we then analyse the new CV and compile it with the historical CV datasets. Hence, the trading can resume. Since the stocks are traded within the same CV from time to time, the returns are maximised. Without using the proposed model, we will never know when the correlation of any paired indices is about to diverge.

ii) With respect to portfolio trading, there is a possibility that any stock indices in the different correlation can be cross-paired and cross-traded among them, provide that they have shared the same CV. Thus, it creates additional trading opportunities inasmuch as risk is minimised.

5.3 Results and Discussion for Pairs Trading Part

5.3.1 Generating the Mean Regression and CV

Referring to Bloomberg terminal, Table 4.1 summarises the 134 datasets of Global Dow indices in the year 2013. Following, Fig. 5.1 presents simulation procedure of the proposed multi-class Pairs Trading model using Mean Reversion and CV, and it is expressed in order as follows:

i) Assign a matrix $x_{ki}(t)$ where k represents the number of columns, $k = 134$ and i represents the number of rows, $i = 3213$

ii) By normalising the matrix of $x_{ki}(t)$, we obtain $A_{ki}(t)$

iii) Calculate $A_{ki}(t)$ for $k = 134$ and $i = 3213$

iv) By selecting the highest return of $A_{ki}(t)$ using the Person's correlation coefficient, we obtain $x_{p1}(t)$ and $x_{p2}(t)$ in time series, see results in Table 5.1

v) Use the Mean Reversion algorithm in 5.1 to compute each point of reverse of $x_{p1}(t)$ and $x_{p2}(t)$ in time series. Then mark the reversed local maxima and minima of $x_{p1}(t)$ and $x_{p2}(t)$ in time series

vi) Compute each local $x_{p1}(t)$ and $x_{p2}(t)$ in time series with the coefficient of variance (CV)

vii) Thus, the local $x_{p1}(t)$ and $x_{p2}(t)$ in time series are grouped into different $CV_1, CV_2, \ldots, CV_n$, and termed to $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$

viii) Calculate expected returns of the local $x_{p1}(t)$, $x_{p2}(t)$, $x_{p1}(t_{CV})$, and $x_{p2}(t_{CV})$

ix Next, we compare the expected returns of $x_{p1}(t)$ and $x_{p2}(t)$ (the original datasets) with the returns of $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$ (the datasets, which are applied the Mean Reversion and CV). The probabilities for calculating the expected returns of $x_{p1}(t)$, $x_{p2}(t)$, $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$ using Markov chain are listed in Table 5.4 and 5.5.

x) For robustness test, use the same procedures listed in item v) and item vi) calculating the expected returns of another ten cross-pairing that listed in table **??**. Then compare the expected returns of ten cross-pairing stocks of $x_{p1}(t)$ and $x_{p2}(t)$ (the original datasets) with the $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, the datasets which have applied the Mean Reversion and CV, are also shown in Table 5.8.

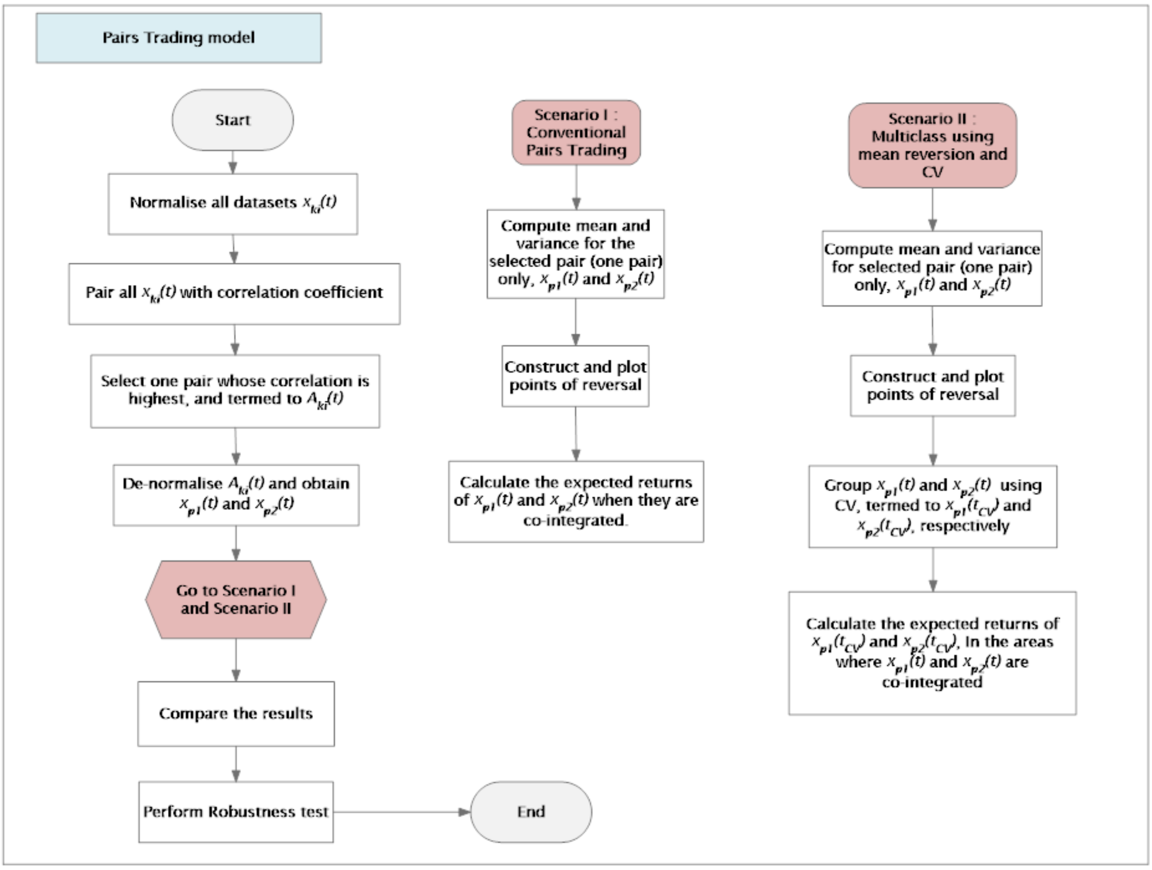
Pairs Trading model
Start
Normalise all datasets $x_{ki}(t)$
Pair all $x_{ki}(t)$ with correlation coefficient
Select one pair whose correlation is highest, and termed to $A_{ki}(t)$
De-normalise $A_{ki}(t)$ and obtain $x_{p1}(t)$ and $x_{p2}(t)$
Go to Scenario I and Scenario II
Compare the results
Perform Robustness test
End
Scenario I : Conventional Pairs Trading
Compute mean and variance for the selected pair (one pair) only, $x_{p1}(t)$ and $x_{p2}(t)$
Construct and plot points of reversal
Calculate the expected returns of $x_{p1}(t)$ and $x_{p2}(t)$ when they are co-integrated.
Scenario II : Multiclass using mean reversion and CV
Compute mean and variance for selected pair (one pair) only, $x_{p1}(t)$ and $x_{p2}(t)$
Construct and plot points of reversal
Group $x_{p1}(t)$ and $x_{p2}(t)$ using CV, termed to $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, respectively
Calculate the expected returns of $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, In the areas where $x_{p1}(t)$ and $x_{p2}(t)$ are co-integrated

Figure 5.1 Procedure of the multiclass Pairs Trading model

The workflow of the multi-class Pairs Trading demonstrated in Figure 5.1

is started by normalising all the datasets $x_{ki}(t)$, pairing $x_{ki}(t)$ with Pearson's coefficient. Then, we select the pair that has the highest value of CV and term to $A_{ki}(t)$, and de-normalising the paired of $A_{ki}(t)$. Finally, we obtain $x_{p1}(t)$ and $x_{p2}(t)$. The next step is to calculate the multi-class Pairs Trading using Scenario II. The results of Scenario II are then subject to compare with Scenario I which is the conventional cointegration of the paired trading. In Scenario I, we calculate the expected returns of co-integrated $x_{p1}(t)$ and $x_{p2}(t)$, see Table 5.8, using probability in Table 5.4 and 5.5 whereas we process Scenario II with the following:

i) compute mean and variance of $x_{p1}(t)$ and $x_{p2}(t)$

ii) construct point of reversal using items i) to viii) under Section 5.1

iii) group $x_{p1}(t)$ and $x_{p2}(t)$ and use Equation 5.2 to compute Mean Reversion and CV, then termed to $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$. Next, we calculate probabilities and the expected returns of $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, resulted in Table 5.5 and Table 5.8, respectively.

5.3.2 Results in pairing the normalised datasets

Consequent to the procedural workflow presented in Figure 5.1, all of the datasets are normalised. We introduce the Pearson's correlation coefficient to measure the degree of correlation among the paired stock indices. Because there are 127 datasets, we cross-map each stock price and neglect redundant pairings.

Because of pairing, there are 8911 pairs. We have found that Mitsubishi UFJ Financial Group Inc. (the X8306JP) and Mizuho Financial Group Inc. (the X8411JP) stock share the highest correlation coefficient of 0.990423021. Figure 5.2 presents two graphs, the X8306JP and the X8411JP. To ease a presentation, the x-axis represents a datasets, whereas the y-axis represents the normalised values

Table 5.1 Top ten pairs from the Global Dow Index that share a high correlation coefficient value

Rank	Stock #1	Stock #2	Correlation Coefficient
1	X8306JP	X8411JP	0.990423021
2	GLEFP	UCGIM	0.979811683
3	BBVASM	UCGIM	0.979511643
4	DBKGR	GLEFP	0.977928533
5	GLEFP	UBSNVX	0.971305147
6	BBVASM	GLEFP	0.971011881
7	IBMUS	NKEUS	0.970135778
8	DBKGR	UCGIM	0.969867105
9	AMZNUS	IBMUS	0.968048722
10	BBVASM	DBKGR	0.965423526

ranging from 3.00 to -1.00. This implies that the pairs of the X8306JP and the X8411JP performed close to the mean comparing to the standard deviation at the scale of ±3. We present the ranking of top ten pairs out of 8001 pairs and their correlation coefficients in Table 5.1.

5.3.3 Results in using Mean Reversion and CV

Referring to Table 5.1, we select the highest correlation coefficient pair, the X8306JP and the X8411JP and simulate those datasets separately with Mean Reversion and CV. They are outlined in the items i) to viii) in section **??**. At this stage, the datasets have been partitioned into different CV values in time series.

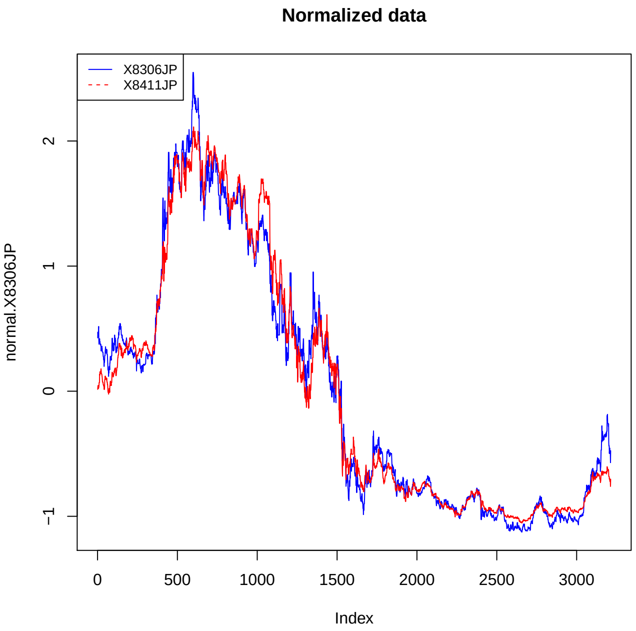

Figure 5.2 Performance of the highest correlation coefficient, X8306JP and X8411JP

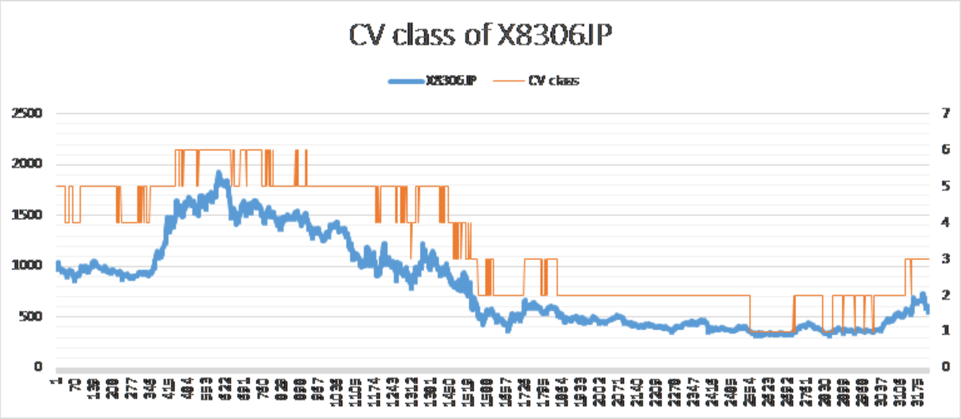

Figure 5.3 The X8306JP showing the different CVs comparing the original datasets

Figure 5.3 and 5.4 show the performance of Mean Reversion and CV by plotting six different CV classes, and two original datasets, X8306JP and X8411JP. Of those six CV classes, the x-axis represents the entire datasets in time series; whereas, the y_1-axis represents the stock values of X8306JP and X8411JP, and the CV values use the scale of the y_2-axis.

5.3.4 Risk mitigation using Mean Reversion and CV

There are six CV classes showing the minimum to maximum values of datasets in each class. Apparently, it is illustrated in Table 5.2 and 5.3. With the remark, the current the X8306JP and the X8411JP datasets have no longer formatted in time series.

For risk mitigation of any stock trading, we utilise contents in Table 5.2 and 5.3 starting from the following:

i) Collect historical minimum and maximum records/units of Pairs Trading for a particular period, e.g., 500 daily records/units of the X8306JP and the X8411JP

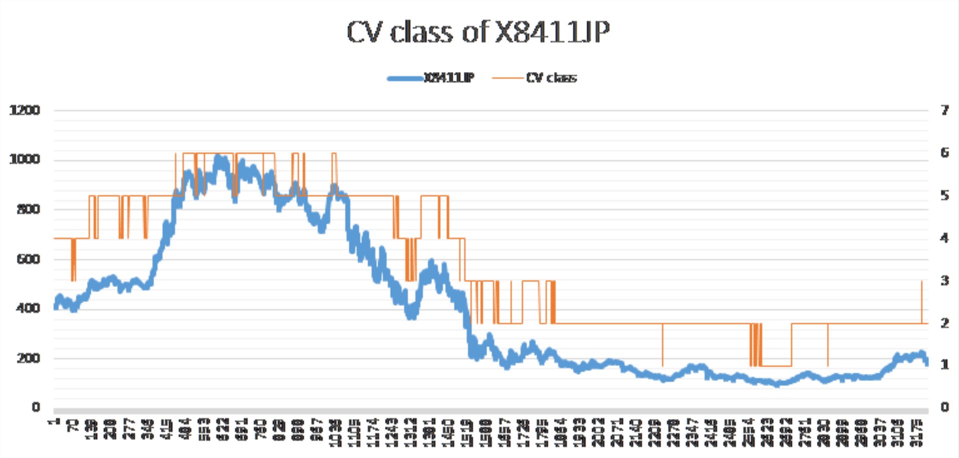

Figure 5.4 The X8411JP showing the different CVs comparing the original datasets

ii) Match the present observed prices of the X8306JP and the X8411JP with one of the CV classes

   a) In case of non-volatility, the future price will behave and situate in the same CV class, use Long and Short positions for trading. It is because we assume that the future stock prices of the X8306JP and the X8411JP will probability fit into the existing CV class

   b) If the new observed prices are highly volatile and run out of the situated CV class, stop trading

   c) If the new observed prices are equal to the previous prices, continue to trade by using the last position

iii) Update Table 5.2 and 5.3 and going item i)

iv) Check the new volatility with variance changes

v) To continue trading, loop the procedures in item ii) to item iv)

Table 5.2 Detailed classification of the stock X8306JP, prices in US dollars

X8306JP					
Class	CV	Range	Units	Mean	Variance
1	0.028041352	320-355.9368	208	337.774	89.712
2	0.122561653	355.9369-550.3055	1244	434.7178	2838.7
3	0.104160587	550.3056-813.2123	264	630.7197	4316
4	0.036502849	813.2124-939.0795	246	900.6911	1080.9
5	0.17382795	939.0796-1512.9	942	1176	41789
6	0.059390989	1512.9001-1930	309	1637.3	9455.6

In the Table 5.2, the CV class 2 of the X8306JP the highest number of points. The highest variance of the X8306JP is in the CV class 5.

Table 5.3 Detailed classification of the stock X8411JP, prices in US dollars

X8411JP

Class	CV	Range	Units	Mean	Variance
1	0.038475671	98-112.2439	129	106.6977	16.8532
2	0.203966775	112.2440-224.8609	1376	156.4949	1018.9
3	0.216937345	224.8610-404.7557	233	274.0043	3533.3
4	0.053668296	404.7558-488.5782	285	448.2702	578.7824
5	0.212921968	488.5783-877.5724	831	654.9639	19448
6	0.034902598	877.5724-1020	359	934.5515	1064

In the Table 5.3, the CV class 2 of the X8411JP the highest number of points. The highest variance of the X8411JP is in the CV class 5. It is similar to the results of the X8306JP.

5.3.5 Proof concept of the Mean reversion and CV

This section is to proof that in the cointegrated Pairs Trading using the proposed Mean Reversion and CV model can outperform the conventional cointegrated Pairs Trading (without using the Mean Reversion and CV).

Initially, we calculate probabilities of the X8306JP and the X8411JP assuming that the chance of the future stock prices moving either upward or downward is equal, at which both probabilities are 0.5. On contrary, the probabilities of the X8306JP and X8411JP using the Mean Reversion and CV are better than those of the conventional cointegrated Pairs Trading as displayed in Table 5.5.

In terms of comparison, the expected returns of the model using Mean Reversion and CV shown in Table 5.6 are better than the conventional Pairs Trading, at which listed in Table 5.4 and 5.5.

Additionally, we conduct robustness test by using other pairs of prices from the Global Dow indices which have shared a high correlation coefficient values listed in Table 5.8. The author found that the expected returns using the conventional Pairs Trading, are less than those of Mean Reversion and CV. Thus, we conclude that the proposed model is robust.

Calculation of Probabilities of the paired stocks, the X8306JP and the X8411JP

Using Equation 5.2 and Equation 5.4 to calculate of the expected returns of the cointegrated conventional Pairs Trading, and the cointegrated Pairs Trading using Mean Reversion and CV, then subtituting the value of some elements as follows

- $I_i^{L\&S}(t)$ is 1 if a long position is created for individual return, a value of -1 if a short position is created, and 0 otherwise;
- t represents the dummy variable that takes the value of 1 if a transaction is made for individuals at time t and 0 otherwise;
- C represents the transaction cost per operation and set to 0.25%;
- T represents the number of observations with 3961 data points;
- $W_i(t)$ is weight at position 1.

Each expected returns of the co-integrated $x_{p1}(t)$ and $x_{p2}(t)$ are calculated by using the value of the present observed variables multiplies with the probability of the lag and repeats infinitely in time series. The expected returns of any co-integrated Pairs Trading can be expressed by

$$ER_{CO} = \sum_{i=1}^{n} R^{i}_{CO}(t) p^{i}_{CO}(t) \tag{5.5}$$

where $R^{i}_{CO}(t)$ is the return of cointegrated $x_{p1}(t)$ and $x_{p2}(t)$ in scenario i, $p^{i}_{CO}(t)$ is the probability for the return $R^{i}_{CO}(t)$ in scenario i, and i counts the number of scenarios. However, we omit to calculate the first two observations after the stocks reverted. It is because we have taken into consideration that some stock can be highly volatile and immediately reverted. Additionally, the returns of co-integrated $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$ can be termed to $R^{i}_{CO}(t_{CV})$; and the results are listed in Table **??**. The expected returns of $R^{i}_{CO}(t_{CV})$ are inevitably similar to those of the expected returns of $R^{i}_{CO}(t)$. We calculate probability for expected returns of the conventional cointegrated by assuming that each stock in the same pair can revert to the co-integrated line and vice versa with a probability of 0.5. The total probability reversion of co-integrated pair is calculated to 0.5 multiplies with 0.5, equalling to 0.25. Hence, the total probability of non-reverted pairs moving along time series is 1.00 minus 0.25, equalling 0.75 as illustrated in Table 5.4.

The difference is that the calculation of the expected returns of $R^{i}_{CO}(t)$ used the probability listed in Table 5.5 rather than the fixed of probability employed in the calculation of $R^{i}_{CO}(t)$, in which is given to 0.75. It is because we assume that any stock prices during the trade can equally move up and down. We introduce Markov chain to calculate probabilities of the conventional co-integrated Pairs Trading the used Mean Reversion and CV. In the Markov chain's process, the value of the present observation is multiplied with the probability of the lag, and it repeats an infinite number of times. Table 5.5 indicates, the X8306JP and the X8411JP are ranging from 0.865853659 to 0.986334405. Whereas the probability of the conventional co-integrated Pairs Trading (without Mean Reversion and CV) remains to 0.75 as illustrated in Table 5.4.

Table 5.4 Calculations of the probabilities of conventional co-integrated Pairs Trading (without Mean Reversion and CV)

Index	Probabilities of conventional cointegrated Pairs Trading (without Mean Reversion and CV)
X8306JP	0.75
GLEFP	0.75

Table 5.5 Calculations of the probabilities of cointegrated Pairs Trading using Mean Reversion and CV

Index	Probabilities of cointegrated Pairs Trading Mean Reversion and CV					
Class	CV_1	CV_2	CV_3	CV_4	CV_5	CV_6
X8306JP	0.9663	0.9863	0.9316	0.8659	0.9565	0.9482
X8411JP	0.9457	0.9855	0.9013	0.9193	0.9700	0.9666

The Table 5.4 shows that the total probability of non-reverted pairs moving along time series is 0.75 for both the X8306JP and the X8411JP.

The Table 5.5 shows that the probability of non-reverted pairs moving along time series of the conventional co-integrated Pairs Trading the used Mean Reversion and CV are ranging from 0.8659 to 0.9863 and from 0.9013 to 0.9855, for the X8306JP and the X8411JP, respectively.

Calculation of expected returns

This section consists of two parts, at which the first part represents a calculation for expected returns of co-integrated $x_{p1}(t)$ and $x_{p2}(t)$, $R^i_{CO}(t)$, and the second part represents calculation of expected returns of co-integrated $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, $R^i_{CO}(t_{CV})$.

Table 5.6 represents the expected returns in US dollars of the co-integrated Pairs Trading using Mean Reversion and CV

Block no	Ranking	Data points	class-X83	Prob.-X83	class-X84	Prob.-X84	Returns of X83 and X84	Expected returns of X83 and X84
1	$2^{nd} - 34^{th}$	33	5	0.9565	5	0.9193	33.7055	33.7043
2	$37^{th} - 46^{th}$	10	4	0.8659	4	0.9193	10.2160	10.2160
3	$49^{th} - 60^{th}$	12	5	0.9565	5	0.9193	12.2602	12.2602
4	$78^{th} - 88^{th}$	11	4	0.8659	4	0.9193	11.2383	11.2383
5	$91^{st} - 127^{th}$	37	5	0.9565	5	0.9193	37.8006	37.8005
6	$130^{th} - 147^{th}$	18	5	0.9565	5	0.9700	18.3897	18.3897
7	$155^{th} - 158^{th}$	4	5	0.9565	5	0.9193	4.0865	4.0865
8	$161^{st} - 222^{nd}$	62	5	0.9565	5	0.9700	63.3417	63.3417
9	$233^{rd} - 238^{th}$	6	5	0.9565	5	0.9700	6.1299	6.1299
10	$242^{nd} - 245^{th}$	4	4	0.8659	4	0.9193	4.0866	4.0866
11	$249^{th} - 251^{st}$	3	4	0.8659	4	0.9193	3.0650	3.0650
12	$254^{th} - 271^{st}$	18	4	0.8659	4	0.9700	18.3896	18.3896
13	$276^{th} - 302^{nd}$	27	4	0.8659	4	0.9700	27.5845	27.5845
14	$312^{th} - 315^{th}$	4	4	0.8659	4	0.9700	4.0866	4.0866
15	$318^{th} - 321^{st}$	4	5	0.9565	5	0.9700	4.0866	4.0866
16	$325^{th} - 328^{th}$	4	5	0.9565	5	0.9700	4.0866	4.0866
17	$338^{th} - 342^{nd}$	5	4	0.8659	4	0.9193	5.1082	5.1082
18	$347^{th} - 439^{th}$	93	5	0.9565		5 0.9700	95.0136	95.0136
19	$442^{nd} - 445^{th}$	4	6	0.9482	6	0.9700	4.0867	4.0867
20	$450^{th} - 453^{rd}$	4	6	0.9482	6	0.9700	4.0866	4.0866
21	$457^{th} - 460^{th}$	4	6	0.9482	6	0.9700	4.0867	4.0867
22	$466^{th} - 468^{th}$	3	6	0.9482	6	0.9700	3.0650	3.0649
23	$476^{th} - 517^{th}$	42	6	0.9482	6	0.9666	42.9093	42.9093
24	$528^{th} - 550^{th}$	23	6	0.9482	6	0.9666	23.4980	23.4980

Block no	Ranking	Data points	class-X83	Prob.-X83	class-X84	Prob.-X84	Returns of	Expected returns of
							X83 and X84	X83 and X84
25	$554^{th} - 644^{th}$	91	6	0.9482	6	0.9666	92.9702	92.9702
26	$651^{st} - 656^{th}$	6	6	0.9482	6	0.9666	6.1299	6.1299
27	$661^{st} - 666^{th}$	6	5	0.9565	5	0.9700	6.1298	6.1298
28	$679^{th} - 699^{th}$	21	6	0.9482	6	0.9666	21.4547	21.4547
29	$703^{rd} - 755^{th}$	53	6	0.9482	6	0.9666	54.1476	54.1476
30	$758^{th} - 768^{th}$	11	5	0.9565	5	0.9666	11.2382	11.2382
31	$780^{th} - 785^{th}$	6	6	0.9482	6	0.9666	6.1299	6.1299
32	$792^{nd} - 798^{th}$	7	5	0.9565	5	0.9666	7.1515	7.1515
33	$802^{nd} - 811^{th}$	10	5	0.9565	5	0.9666	10.2165	10.2165
34	$814^{th} - 873^{rd}$	60	5	0.9565	5	0.9700	61.2991	61.2991
35	$893^{rd} - 895^{th}$	3	5	0.9565	5	0.9666	3.0649	3.0649
36	$898^{th} - 916^{th}$	19	5	0.9565	5	0.9700	19.4114	19.4114
37	$924^{th} - 1022^{nd}$	99	5	0.9565	5	0.9700	101.1434	101.1434
38	$1025^{th} - 1037^{th}$	13	5	0.9565	5	0.9666	13.2815	13.2815
39	$1040^{th} - 1177^{th}$	138	5	0.9565	5	0.9700	140.9879	140.9879
40	$1180^{th} - 1182^{nd}$	3	4	0.8659	4	0.9700	3.0650	3.0650
41	$1191^{st} - 1194^{th}$	4	4	0.8659	4	0.9700	4.0866	4.0866
42	$1197^{th} - 1245^{th}$	49	5	0.9565	5	0.9700	50.0610	50.0610
43	$1248^{th} - 1250^{th}$	3	5	0.9565	5	0.9193	3.0649	3.0649
44	$1257^{th} - 1261^{st}$	5	5	0.9565	5	0.9700	5.1083	5.1083
45	$1288^{th} - 1292^{nd}$	5	5	0.9565	5	0.9193	5.1082	5.1082
46	$1299^{th} - 1301^{st}$	3	4	0.8659	4	0.9013	3.0649	3.0649
47	$1311^{th} - 1313^{th}$	3	4	0.8659	4	0.9013	3.0650	3.0650
48	$1318^{th} - 1322^{nd}$	5	4	0.8659	4	0.9013	5.1083	5.1083
49	$1326^{th} - 1328^{th}$	3	5	0.9565	5	0.9193	3.0649	3.0649

Block no	Ranking	Data points	class-X83	Prob.-X83	class-X84	Prob.-X84	Returns of	Expected returns of
							X83 and X84	X83 and X84
50	$1338^{th} - 1348^{th}$	11	5	0.9565	5	0.9193	11.2382	11.2381
51	$1351^{st} - 1412^{th}$	62	5	0.9565	5	0.9699	63.3422	63.3422
52	$1423^{rd} - 1426^{th}$	4	5	0.9565	5	0.9699	4.0866	4.0866
53	$1431^{st} - 1443^{rd}$	13	5	0.9565	5	0.9699	13.2815	13.2815
54	$1452^{nd} - 1463^{rd}$	12	4	0.8659	4	0.9192	12.2598	12.2598
55	$1484^{th} - 1489^{th}$	6	4	0.8659	4	0.9193	6.1299	6.1299
56	$1495^{th} - 1509^{th}$	15	4	0.8659	4	0.9193	15.3248	15.3248
57	$1512^{th} - 1518^{th}$	7	3	0.9316	3	0.9013	7.1516	7.1516
58	$1522^{nd} - 1524^{th}$	3	3	0.9316	3	0.9013	3.0650	3.0650
59	$1528^{th} - 1531^{st}$	4	3	0.9316	3	0.9013	4.0867	4.0867
60	$1536^{th} - 1552^{nd}$	17	3	0.9316	3	0.9013	17.3680	17.3680
61	$1558^{th} - 1569^{th}$	12	2	0.9863	2	0.9013	12.2598	12.2598
62	$1572^{nd} - 1574^{th}$	3	2	0.9863	2	0.9855	3.0649	3.0649
63	$1577^{th} - 1583^{rd}$	7	2	0.9863	2	0.9013	7.1516	7.1516
64	$1586^{th} - 1588^{th}$	3	3	0.9316	3	0.9013	3.0650	3.0649
65	$1597^{th} - 1600^{th}$	4	2	0.9863	2	0.9013	4.0866	4.0866
66	$1603^{rd} - 1608^{th}$	6	3	0.9316	3	0.9013	6.1299	6.1299
67	$1611^{th} - 1616^{th}$	6	2	0.9863	2	0.9013	6.1299	6.1299
68	$1619^{th} - 1622^{nd}$	4	2	0.9863	2	0.9855	4.0866	4.0866
69	$1625^{th} - 1628^{th}$	4	2	0.9863	2	0.9013	4.0866	4.0866
70	$1638^{th} - 1678^{th}$	41	2	0.9863	2	0.9855	41.8877	41.8877
71	$1684^{th} - 1722^{nd}$	39	2	0.9863	2	0.9855	39.8444	39.8444
72	$1725^{th} - 1783^{rd}$	59	3	0.9316	3	0.9013	60.2774	60.2774
73	$1788^{th} - 1790^{th}$	3	2	0.9863	2	0.9855	3.0650	3.0650
74	$1794^{th} - 1797^{th}$	4	2	0.9863	2	0.9855	4.0866	4.0866

Block no	Ranking	Data points	class-X83	Prob.-X83	class-X84	Prob.-X84	Returns of	Expected returns of
							X83 and X84	X83 and X84
75	$1813^{th} - 1826^{th}$	14	3	0.93156	3	0.9013	14.3031	14.3031
76	$1838^{th} - 1840^{th}$	3	3	0.93156	3	0.9013	3.0650	3.0650
77	$1843^{rd} - 1846^{th}$	4	3	0.93156	3	0.9855	4.0866	4.0866
78	$1849^{th} - 2237^{th}$	389	2	0.9863	2	0.9855	397.4224	397.4224
79	$2241^{st} - 2556^{th}$	316	2	0.9863	2	0.9855	322.8418	322.8418
80	$2562^{nd} - 2566^{th}$	5	1	0.9663	1	0.9457	5.1082	5.1082
81	$2569^{th} - 2573^{rd}$	5	1	0.9663	1	0.9855	5.1083	5.1083
82	$2586^{th} - 2590^{th}$	5	1	0.9663	1	0.9855	5.1083	5.1083
83	$2593^{rd} - 2595^{th}$	3	1	0.9663	1	0.9457	3.0650	3.0650
84	$2598^{th} - 2602^{nd}$	5	1	0.9663	1	0.9855	5.1083	5.1083
85	$2605^{th} - 2710^{th}$	106	1	0.9663	1	0.9457	108.2950	108.2950
86	$2720^{th} - 2722^{nd}$	3	1	0.9663	1	0.9855	3.0650	3.0650
87	$2725^{th} - 2822^{nd}$	98	2	0.9863	2	0.9855	100.1218	100.1218
88	$2825^{th} - 2846^{th}$	22	1	0.9663	1	0.9855	22.4763	22.4763
89	$2850^{th} - 2857^{th}$	8	1	0.9663	1	0.9855	8.1732	8.1732
90	$2860^{th} - 2897^{th}$	38	2	0.9863	2	0.9855	38.8228	38.8228
91	$2901^{st} - 2939^{th}$	39	2	0.9863	2	0.9855	39.8444	39.8444
92	$2944^{th} - 2975^{th}$	32	2	0.9863	2	0.9855	32.6929	32.6929
93	$2979^{th} - 3004^{th}$	26	2	0.9863	2	0.9855	26.5629	26.5629
94	$3007^{th} - 3010^{th}$	4	1	0.9663	1	0.9855	4.0866	4.0866
95	$3013^{th} - 3126^{th}$	114	2	0.9863	2	0.9855	116.4682	116.4682
96	$3129^{th} - 3147^{th}$	19	3	0.9316	3	0.9855	19.4114	19.4114
97	$3153^{rd} - 3189^{th}$	37	3	0.9316	3	0.9855	37.8011	37.8011
98	$3195^{th} - 3212^{th}$	18	3	0.9316	3	0.9855	18.3897	18.3897

The Table 5.6 shows that the block number 78 is the highest data points, 389 points, i.e., the trader can trade for 389 days.

The different expected returns in each block of the X8306JP and the X8411JP are calculated by using the returns of the X8306JP and the X8411JP multiply by the same probability value of 0.75. As a result, the total expected return of both co-integrated the X8306JP and the X8411JP to US$ 2461.915799.

The expected returns of co-integrated $x_{p1}(t_{CV})$ and $x_{p2}(t_{CV})$, $R^i_{CO}(t_{CV})$ using Mean Reversion and CV consist of 98 blocks. In each block the number of data points is ranging from 3 to 389, depending on the distribution of CV classes, e.g., in block 1 there are 11 data points at the ranking of 78^{th} to 88^{th}. We omit to calculate the blocks that have the number of data less than 3. It is because the stocks may be highly volatile from the first two observations when the stocks have been reverted. The probabilities of both the X8306JP and the X8411JP are based on Markov chain, in which represent the smallest value of 0.865853659 and the highest value of 0.986334405. Apparently, the returns of co-integrated the X8306JP and the X8411JP, and the expected returns of co-integrated the X8306JP and the X8411JP using Mean Reversion and CV are demonstrated, given the total expected returns of both equals US$ 2781.944909. However, the allocation of each CV class undertakes values of observations. Thus, during the calculation process; each $R^i_{CO}(t_{CV})$ has never been mixed up.

Comparison of the performance of the conventional cointegration (without Mean Reversion and CV) with the cointegration using Mean Reversion and CV can be demonstrated by looking at values of the expected returns of both cases. The expected returns of the conventional cointegration and the proposed model using Mean Reversion and CV are US$ 2351.84 and US$ 2003.77, respectively. AS a result, the returns of co-integration using Mean Reversion and CV are higher than the conventional cointegration (without Mean Reversion and CV). Therefore, we conclude that the proposed cointegrated pairs trading using Mean Reversion and CV outperforms the conventional cointegrated pairs trading model. Therefore, the net premium in 10-year trading with the cointegrated pairs trading using Mean Reversion and CV, which calculated the difference of both cases, yields to US$ 320.0291104, equalling to 12.9991899%.

5.3.6 Results of nonlinear and non-stationary test

The testing results shows that distributions of the X8306JP and the X8411JP were not neither in normal nor stationary since the p-value is less than 0.05%, see Table 5.7.

The Table shows that the X8306JP and the X8411JP were non-stationary.

Robustness test

To compute the expected returns of the cross-paired trading, we assign the contents in Table 5.1, which are the top ten pairs that have been characterised for the highest correlation as input. Then, we use the same techniques that have been used to calculate the expected returns of X8306JP and X8411JP for computing the expected returns of the top ten pairs. The results are listed in Table 5.8 and Table 5.9. Whereas, Table 5.8 represents the expected returns of the conventional co-integrated pairs trading (without Mean Reversion and CV), and Table 5.9 represents the expected returns of the co-integrated pairs trading using Mean Reversion and CV.

Table 5.7 Normality and Unit root test for X8306JP and X8411JP

X8306JP (actual)	Statistics	p-value
Normality test		
hline Anderson-Darling	146.1787	< 2.2e-16
Lilliefors (Kolmogorov-Smirnov)	0.1783	< 2.2e-16
Pearson chi-square	4190.571	< 2.2e-16
Unit root test		
Augmented Dickey-Fuller	-0.9075	< 2.2e-16
X8411JP (actual)	Statistics	p-value
Normality test		
Anderson-Darling	186.466	< 2.2e-16
Lilliefors (Kolmogorov-Smirnov)	0.2138	< 2.2e-16
Pearson chi-square	7188.54	< 2.2e-16
Unit root test		
Augmented Dickey-Fuller	-0.7736	0.00015

Table 5.8 The expected returns in US dollars of the conventional co-integrated Pairs Trading

	X8411JP	UCGIM	UCGIM	GLEFP	UBSNVX	GLEFP	NKEUS	UCGIM	IBMUS	DBKGR
X8306JP	2461.92	2461.92	2461.92	2461.92	2461.92	2461.92	2461.92	2461.92	2461.93	2461.92
GLEFP	2461.93	2461.93	2461.93	0	2461.92	0	2231.28	2461.92	2229.75	1806.79
BBVASM 2461.93	2367.68	2367.68	2461.93	2461.93	2461.93	2461.93	2367.68	2461.92	2461.93	
DBKGR 2461.93	2461.93	2461.93	1806.79	2461.93	1806.79	2215.96	2461.93	2040.49	0	
GLEFP	2461.93	2461.92	2461.92	0	2461.92	0	2231.28	2461.92	2229.75	1806.79
BBVASM	2461.93	2367.68	2367.68	2461.93	2461.93	2461.93	2461.93	2367.68	2461.92	2461.93
IBMUS	2232.83	2461.92	2461.92	2229.75	2461.92	2229.75	2461.92	2461.92	0	2040.49
DBKGR	2461.93	2461.93	2461.93	1806.79	2461.93	1806.79	2215.96	2461.93	2040.49	0
AMZNUS 2396.04	2300.24	2300.24	2363.09	2229.83	2363.09	2461.91	2300.24	2061.95	2327.08	
BBVASM	2461.93	2367.68	2367.68	2461.93	2461.93	2461.93	2461.93	2367.68	2461.92	2461.93

Table 5.9 The expected returns in US dollars of the co-integrated Pairs Trading using Mean Reversion and CV

	X8411JP	UCGIM	UCGIM	GLEFP	UBSNVX	GLEFP	NKEUS	UCGIM	IBMUS	DBKGR
X8306JP	2781.94	2863.69	2863.69	2798.31	2844.28	2798.31	2780.93	2863.69	2749.27	2773.78
GLEFP	2832.03	2928.05	2928.05	0	2934.18	0	2557.18	2928.05	2572.51	2091.32
BBVASM	2777.89	2748.25	2748.25	2802.40	2862.68	2802.40	2776.87	2748.25	2768.67	2802.40
DBKGR	2814.66	2897.40	2897.40	2091.32	2903.53	2091.32	2504.06	2897.40	2292.57	0
GLEFP	2832.03	2928.05	2928.05	0	2934.18	0	2557.18	2928.05	2572.51	2091.32
BBVASM	2777.89	2748.25	2748.25	2802.40	2862.68	2802.40	2776.87	2748.25	2768.67	2802.40
IBMUS	2528.61	2908.63	2908.63	2572.51	2884.11	2572.51	2835.06	2908.63	0	2292.57
DBKGR	2814.66	2897.40	2897.40	2091.32	2903.53	2091.32	2504.06	2897.40	2292.57	0
AMZNUS	2814.68	2752.31	2752.31	2779.93	2654.25	2779.93	2881.03	2752.31	2382.50	2716.59
BBVASM	2777.89	2748.25	2748.25	2802.40	2862.68	2802.40	2776.87	2748.25	2768.67	2802.40

The results of computing the expected returns of the cointegrated Pairs Trading using Mean Reversion and CV are shown in Table 5.9. Apparently, the average expected returns of the cointegrated Pairs Trading using Mean Reversion and CV are US$ 253631.306 and US$ 2536.31306, respectively. The expected returns of the co-integrated Pairs Trading using Mean Reversion and CV outperforms those of the conventional cointegrated Pairs Trading (without Mean Reversion and CV), see Table 5.8. It is proven that the benefit of co-integrated Pairs Trading using Mean Reversion and CV, for those top ten cross-paired stocks with the 10-year investment, is US$ 27838.05873, equaling to 13.54%.

CHAPTER VI

THE PREDICTION MODELS

Pairs trading and its theoretical considerations were introduced in Chapter II. The risk in trading stock can be reduced by using Pairs trading method. In the previous chapter, Chapter V, a new novel pairs trading model is proposed. Moreover, the simulation results show that the cointegrated Pairs Trading using the proposed method outperforms those of the conventional co-integrated pairs trading outstandingly. Thus, benefits of the proposed model are to build a new series of risk mitigation and maximise returns of co-integrated stocks. If the movement or the future price of the next time step to trade can be predicted, the risk shall be inevitably reduced. Therefore, this study is to combine the Prediction model with the Pairs Trading.

This chapter describes about prediction model which use in this research, i.e., Autoregressive Integrated Moving Average (ARIMA) model, Markov Chain Monte Carlo (MCMC) methods, and Support Vector Machine (SVM). There are also a framework of each forecasting model, in which included theoretical considerations of the prediction models and including the simulation results and discussions further in the Chapter.

6.1 Introduction to Prediction Models

Kovalerchuk et al. described an overview on techniques in finance, the prediction methods can be classified into three categories: numerical models (ARIMA models, Instance-based learning, neural networks, etc.), rule-based models (decision tree and DNF learning, naive Bayesian classifier, hidden Markov model, etc.), and relational data mining (inductive logic programming).

One of the most popular and frequently used stochastic time series models is the Autoregressive Integrated Moving Average (ARIMA) model. The Markov Chain Monte Carlo (MCMC) methods are particularly attractive for practical finance applications. It was realized that most Bayesian inference could be done by MCMC, whereas very little be done without MCMC.

Recently, Artificial Neural Networks (ANNs) have been attracting increasing attentions in the time series forecasting. Nowadays,the Support Vector Machine (SVM), a new statistic learning theory, has been receiving increasing attention for classification and forecasting. The Support Vector Regression (SVR) is used in forecasting problem.

Hence, there are three models used in this study as follows: Autoregressive Integrated Moving Average (ARIMA) model, Markov Chain Monte Carlo (MCMC) method, and Support Vector Regression (SVR) approach. This section describes the prediction methods as mentioned above.

6.2 Autoregressive Integrated Moving Average (ARIMA) Model

Autoregressive Integrated Moving Average (ARIMA) models intend to describe the current behaviour of variables in terms of linear relationships with their past values. An ARIMA model can be decomposed in two parts. First, it has an Integrated (I) component (d), which represents the amount of differencing to be performed on the series to make it stationary. The second component of an ARIMA consists of an ARMA model for the series rendered stationary through differentiation. The ARMA component is further decomposed into AR and MA components.

6.2.1 Autoregressive (AR) Model

In economics and signal processing, an autoregressive (AR) model (Borchers, B., 2002, Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, and Charles Korede Ayo., 2014) is a random process that is usually used for modelling and prediction in various types of natural phenomena. AR models are a group of linear prediction formulas that attempt to predict the outputs of a system based on previous outputs. The autoregressive (AR) component captures the correlation between the current value of the time series and some of its past values. For example, AR(1) means that the current observation is correlated with its immediate past value at time $t-1$. The main assumption of the AR model is y_t linear combinations of the previous observed values up to a defined maximum lag (p), which is expressed as

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t \tag{6.1}$$

where y_t is the dependent variable value at the moment t, ϕ_t is a constant and ε_t is the error term which is i.i.d. $N(0, \sigma^2)$.

6.2.2 Moving Average (MA) Model

The Moving Average (MA) component represents the duration of the influence of a random (unexplained) shock. For example, MA(1) means that a shock on the value of the series at time t is correlated with the shock at $t-1$. The main assumption of the MA component is that y_t is a random error term plus some linear combination of the previous random error terms up to a defined maximum lag (q), which is expressed as

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} \tag{6.2}$$

where θ_t are constants.

6.2.3 Autoregressive Moving Average (ARMA) Model

When combining AR and MA, the lags of the different series appearing in the forecast equation are AR(p) and MA(q), where p and q are independent. To analyse a time series and fit the ARMA(p, q) model, we require all of observations to be i.i.d. $N(0, \sigma^2)$ that is with a zero mean normal distribution. The expression is given by (Brockwell and Davis, 2002)

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}. \tag{6.3}$$

Rearrange (6.3) to yield

$$y_t - \phi_1 y_{t-1} - \ldots - \phi_p y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} \tag{6.4}$$

and assign the back-shift operator B (where $By_t = y_{t-1}, B^2 y_t = y_{t-2}, andsoon$) to (6.4), before rearranging it to obtain

$$(1 - \phi_1 B - \ldots - \phi_p B^p) y_t = (1 + \theta_1 B + \ldots + \theta_q B^q) \varepsilon_t, \tag{6.5}$$

which can be re-written as

$$\phi_p(B) \Delta^d y_t = \theta_q(B) \varepsilon_t \qquad \text{or} \qquad \phi_p(B) y_t = \theta_q(B) . \varepsilon_t \tag{6.6}$$

where $\phi_p(B)$ and $\theta_q(B)$ are AR and MA operators, respectively.

6.2.4 Autoregressive Integrated Moving Average (ARIMA) Model

In the event that the process being observed is non-stationary, the differences of the series are computed using linear combinations until a stationary time series is found so the ARMA is superseded and referred to as ARIMA(p, d, q) where the I of the differences of the series to be transformed is stationary, and d is the order of difference required to produce a stationary process, a stochastic process whose joint probability distribution does not change when shifted in time, which is normally $0, 1$, or 2 depending on its lagged correlation. Finally, ARIMA(p, d, q) is written as

$$\phi_p(B)\Delta^d y_t = \theta_q(B)\varepsilon_t \tag{6.7}$$

where Δ^d is a difference operator.

Automatic Selection of an ARIMA Model

An automatic method for selecting an ARIMA model is very useful. An automatically selected model should not be accepted blindly as usual, but it has a reason to first select model with something chosen quickly and by objective criterion.

The `R` function `auto.arima` can select all three parameters, p, d, and q, for an ARIMA model. The differencing parameter d is selected using the KPSS test. If the null hypothesis of stationarity is accepted when the KPSS is applied to the original time series, then $d = 0$. Otherwise, the series is differenced until the KPSS accepts the null hypothesis. After that, p and q are selected using either AIC or BIC.

6.2.5 Simulation and Results of the ARIMA model

The datasets from Chapter IV were used to simulate the ARIMA model. The highest correlation paired stocks, the X8306JP and the X8411JP, were used to simulate the results in this section and also in the next two sections as well. These two datasets were then simulated by R programming scripts for ARIMA model. For out-of-sample forecasting, we selected the last 30% of the 3213 sets to be used as a reference. Next, we tested outcomes of the simulations with ARIMA model using the original datasets as input data. We then plotted them against the original test datasets (used as a reference), as shown in the graphs in Fig. 6.1 and 6.4.

The graphs are shown in Fig. 6.1 and 6.4 where the x-axis represents 963 test data points in the time series and the y-axis represents stock prices in US dollars. At which shows the deviations between the simulated graph of the ARIMA model compared with the original datasets. The two graphs are coincidentally in a line where the x-axis represents the data points in the time series and the y-axis represents the US dollars stock prices. The next step was to measure the performance of the ARIMA model using a variety of loss estimators, i.e., MAE, MAPE, MSE, RMSE, R2, AIC, BIC, and Accuracy count (up-down (%)). Table 6.1 and 6.2 show that the MAPE of the X8306JP and the X8411JP are 53.05702 and 66.02224, respectively. It is noticeable that the measurement results of MAPE was too high. That is the simulation results of the AR model which is a part of ARIMA and found that it persisted to the lags, diverting from the original datasets. Having counted the up and down movements along the x-axis, the percentage success of the model reached 72.63267%. This is because of the MA model adjusted the trends of the local datasets from time to time. Once the trends of the average either increased or decreased, the movements of the curves agreed with the changes.

Table 6.1 Simulation results using the ARIMA model to forecast the original X8306JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.53057	0.214543	0.132489
MAPE	53.05702	21.45433	13.24885
MSE	53551.47	9418.297	8083.087
RMSE	62.94142	69.70751	39.13242
R2	NA	NA	NA
AIC	19793.32	22309.71	24784.1
BIC	NA	NA	NA
Up-Down(%)	68.88658	69.0625	70.21944

After comprehensively analysing the results shown in Fig. 6.1 and 6.4 and Table 6.1 and 6.2, we conclude that the ARIMA model was not suitable to use with highly volatile and strictly non-stationary datasets. This was because the ARIMA model required the AR term to be stationary; and it cannot equip with any independent variables; thus, there are no extra independent variables other than the lag of its own to adjusting the model while predicting the 2ndAR, the 3rdAR, and so on. Thus, the error from the previous prediction carried over and become an input for the next prediction round, giving the accumulation of the error in the long term prediction.

The measurement of the performance of the ARIMA model for these two datasets with 80-20 and 90-10 ratio shown in Table 6.1 and 6.2. The plots of 80-20 and 90-10 ratio shown in the graphs in Fig. 6.2, 6.5, 6.3 and 6.6, respectively.

Table 6.2 Simulation results using the ARIMA model to forecast the original X8411JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.6602224	0.2924394	0.122757
MAPE	66.02224	29.24394	12.2757
MSE	9559.542	1879.355	680.8471
RMSE	29.94024	37.89151	22.14481
R2	NA	NA	NA
AIC	16980.82	19078.74	21129.47
BIC	NA	NA	NA
Up-Down(%)	72.63267	73.75	73.66771

6.3 Markov Chain Monte Carlo (MCMC) model

Markov Chain Monte Carlo (MCMC) methods are particularly attractive for practical finance applications for many reasons. Firstly, MCMC is a unified estimation procedure which simultaneously estimates both, parameters and state variables. Secondly, MCMC methods account for estimation and model risk. Finally, MCMC is just a conditional simulation methodology, and therefore avoids any maximization and long unconditional state simulation.

6.3.1 Background Related to the MCMC model

In the 1950s, Monte Carlo simulations were first used in the physics literature. In 1970, Hasting studied the optimality of these algorithms and the Metropolis-Hastings algorithm is introduced (Landauskas, M., 2011).

MCMC (Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park., 2011) is essentially Monte Carlo integration using Markov chains. In brief, Monte Carlo integration draws samples from required distribution and then provides sample averages for approximate expectations. MCMC draws these samples by running a smartly constructed Markov chain. There are many ways to construct these chains, including the Gibbs sampler, which are special cases of the general framework of Metropolis et al. and Hastings.

Let's begin with the concept of a *Markov process*. Consider a stochastic process $\{X_t\}$, where each X_t assumes a value in the space Θ. The process $\{X_t\}$ is a Markov process if it has the property that, given the value of X_t, the values of $X_h, h > t$, do not depend on the values $X_s, s < t$. In other words, $\{X_t\}$ is a Markov process if its conditional distribution function satisfies

$$P(X_h|X_s, s \leq t) = P(X_h|X_t), h > t. \tag{6.8}$$

If $\{X_t\}$ is a discrete-time stochastic process, then the prior property becomes

$$P(X_h|X_t, X_{t-1}, \ldots) = P(X_h|X_t), h > t. \tag{6.9}$$

Let A be a subset of Θ. The function

$$P_t(\theta, h, A) = P(X_h \in A|X_t = \theta), h > t \tag{6.10}$$

is called the *transition probability function* of Markov process.

Consider an inference problem with parameter vector θ and data X, where $\theta \in \Theta$. To make inference, we need to know the distribution $P(\theta|X)$. The idea of Markov chain simulation is to simulate a Markov process on Θ, which converges to a stationary distribution that is $P(\theta|X)$.

The solution to Markov chain simulation is to create a Markov process whose stationary transition distribution is a specified $P(\theta|X)$ and run the simulation sufficiently long so that the distribution of the current values of the process is close enough to the stationary transition distribution. So, for a given $P(\theta|X)$, many Markov chains with desired property can be constructed. The methods that use Markov chain simulation to obtain the distribution $P(\theta|X)$ is referred as *Markov Chain Monte Carlo (MCMC)* methods.

Note that the notation $\pi(\theta)$ is used for the target distribution of interest. In most cases the target will be the posterior distribution for the mode unknowns, $\pi(\theta) = p(\theta|y)$ by given the observations y.

6.3.2 Monte Carlo Modelling of Stock Prices

The process of a stock price is considered as a Brownian motion. Thus its value satisfies the equation:

$$dS = \mu S dt + \sigma S dz. \tag{6.11}$$

Consider a financial mean with log normally distributed returns. The random walk of price of such a financial mean is modeled according this formula (P. Wilmott, 2007):

$$S(t + \Delta t) = S(t) exp((\delta - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t} Z) \tag{6.12}$$

Here random value $Z \sim N(0,1)$ follows standard normal distribution, Δ is annual risk free return and σ is annual standard deviation of the logarithm of a stock price.

6.3.3 Markov chain Monte Carlo (MCMC)

Suppose it is needed to generate $x_i \sim \pi(x)$. When $x_i \sim \pi(x)$ is difficult to sample from, MCMC sampling technique could be performed. In fact MCMC is a set of techniques used for this purpose. Th e main idea of it is to construct a Markov chain $\{X_i\}_{i=0}^{\infty}$, such that

$$\lim_{i\to\infty} P(X_i = x) = \pi(x). \tag{6.13}$$

A Markov chain is predefined by an initial state $P(X_0 = x_0) = g(x_0)$ and the transition kernel $P(y|x) = P(X_{i+1} = y|X_i = x)$. Stationary distribution $\pi(x) = \lim_{i\to\infty} f(x_i)$ is unique if the chain is ergodic. Then:

$$\pi(y) = \sum_{x\in\Omega} \pi(x)P(y|x), \forall y \in \Omega. \tag{6.14}$$

Latter equality could be written as a set of $(n-1)$ linear equations:

$$\begin{cases} \pi(x_2) = \pi(x_1)P(x_2|x_1) + \pi(x_2)P(x_2|x_2) + \ldots + \pi(x_n)P(x_2|x_n) \\ \ldots \\ \pi(x_n) = \pi(x_1)P(x_n|x_1) + \pi(x_2)P(x_n|x_2) + \ldots + \pi(x_n)P(x_n|x_n) \end{cases} \tag{6.15}$$

here $n := |\Omega|$. There are a total number of $(n-1)$ equations and $n(n-1)$ transition probabilities $P(x_j|x_k)$, $k = \overline{1,n}$, $j = \overline{1,n-1}$. Thus there exist an infinite number of transition kernels $P(y|x)$, such that the stationary distribution of the Markov chain is $\pi(x)$.

Metropolis-Hastings algorithm (J.S.Daqpunar, 2007) is One of the techniques used for constructing such a transition kernel. Its idea is to choose any other transition kernel $Q(y|x)$. Then there exists a probability that $Q(y|x)$ is equal to $P(y|x)$.

$$P(y|x) = Q(y|x)\alpha(y|x), y \neq x, \alpha(y|x) \in [0,1]. \tag{6.16}$$

Considering the detailed balance condition of a time-homogeneous Markov chain yields:

$$\pi(x)Q(y|x)\alpha(y|x) = \pi(y)Q(x|y)\alpha(x|y), \forall x \neq y. \tag{6.17}$$

The general solution for eq. 6.17 is $\alpha(y|x) = r(x,y)\pi(y)Q(x|y)$. It is necessary to have a higher acceptance ration when sampling random numbers, therefor by adjusting $r(x,y)$ and considering higher acceptance ration while sampling random numbers (V.Prokaj, 2009) it is shown that:

$$\alpha(y|x) = min\left(1, \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}\right). \tag{6.18}$$

6.3.4 Nonparametric Probability density estimation

Consider a sample consisting of random independent and identically distributed values X_i. Kernel density estimate is chosen for evaluate the probability density of X_i.

$$\hat{f}(x) - \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i), K_h(x) = \frac{1}{h}K\left(\frac{x}{h}\right), \tag{6.19}$$

here $K(\cdot)$ is the kernel function, h is its width.

$$\begin{cases} \int_{-\infty}^{+\infty} K(x)dx = 1, \\ K(x) \geq 0. \end{cases} \Rightarrow \begin{cases} \int_{-\infty}^{+\infty} \hat{f}(x)dx = 1, \\ \hat{f}(x) \geq 0. \end{cases} \tag{6.20}$$

Below are some kernel functions that are frequently used. The triangular kernel function is useful if the data has sharp edged distribution. Gaussian kernel makes the estimate's PDF plot very smooth.

$$K(x) = \begin{cases} 1 - |x|, |x| \leq 1, \\ 0, |x| > 1. \end{cases} (tringular), \tag{6.21}$$

$$K(x) = \begin{cases} \frac{3}{4}(1 - x^2), |x| \leq 1, \\ 0, |x| > 1. \end{cases} (Yapanichnikov), \tag{6.22}$$

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{x^2}{2}} (Gauss). \tag{6.23}$$

Basically, such probability density estimation is about assigning kernel density to each X_i and including weighted sum of all other assignations. The contribution of any other X_j to the probability value at X_i is smaller if $X_i - X_j$ is bigger.

In MCMC simulation a sequence of values which are not independent but instead follow a stochastic process called a Markov chain is produced. The simulation use the algorithm to ensure that the chain will take values in the domain of the unknown θ and that its limiting distribution will be the target distribution $\pi(\theta)$. This means that there is a method of sampling values from the posterior distribution and therefore of making Monte Carlo inferences about θ in the form of sample averages and by means of histograms and kernel density estimates.

The MCMC algorithm produces a chain of values in which each value can depend on the previous value in the sequence.

6.3.5 Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm (Hastings, 1970; Metropolis et al., 1953) is currently the most general algorithm for MCMC simulation. Its basic form is easy to explain and implement and it has several useful generalizations and special cases for different purposes.

The basis of MCMC with the MH algorithm is to reject the original samples if they are outside the unit circle of the target and replace them by another computed sample.

With the MCMC algorithm, a chain of values $\theta^0, \theta^1, \ldots, \theta^N$ is generated in such a way that it can be used as a sample of the target density $\pi(\theta)$.

A general Metropolis-Hastings algorithm is in the following:

1. Start from an initial value θ^0, and select a proposal distribution q.

2. At each step where the current value is θ^{i-1}, propose a candidate for the new parameter θ^* from the distribution $q(\theta^{i-1}, \cdot)$.

3. If the proposed value θ^* is better than the previous value θ^{i-1} in the sense that
$$\pi(\theta^*)q(\theta^*, \theta) > \pi(\theta^{i-1})q(\theta, \theta^*),$$
it is accepted unconditionally.

4. If it is not better in the above sense, θ^* is accepted as the new value with a probability α given by
$$\alpha(\theta, \theta^*) = min\{1, \frac{\pi(\theta^*)q(\theta^*, \theta)}{\pi(\theta)q(\theta, \theta^*)}\}.$$

5. If θ^* is not accepted, then the chain stays at the current value, that is, we set $\theta^i = \theta^{i-1}$.

6. Repeat the simulation from step (2) until enough values have been generated.

As the MH algorithm is currently the most general algorithm for MCMC method, the research will simply use this algorithm.

There are many advantages for MCMC. Firstly, it is flexible. So you can adjust your models as much as you want and it still productively fit them as well. Secondly, it is Reliable, that is it will never hang on a local optimum. It is great in pulling out uncertainties of all kinds. Although the MCMC algorithm is complicated, the inference based on the posterior distributions is very easy and intuitive.

6.3.6 Simulation and Results of the MCMC Model

Similar to Section 6.2, the same datasets, the X8306JP and the X8411JP, were simulated by R programming scripts. Next, we tested the outcomes of the simulations, which were nonlinear and nonstationary, and plotted them against the original test datasets (used as a reference), as shown in Fig. 6.1 and 6.4.

The graphs are shown in Fig. 6.1 and 6.4 where the x-axis represents 963 test data points in the time series and the y-axis represents stock prices in US dollars. At which shows the deviations between the simulated graph of the ARIMA model compared with the original datasets. The two graphs are coincidentally in a line where the x-axis represents the data points in the time series and the y-axis represents the US dollars stock prices. The next step was to measure the performance of the ARIMA model using a variety of loss estimators, i.e., MAE, MAPE, MSE, RMSE, R2, AIC, BIC, and Accuracy count (up-down (%)). Table 6.1 and 6.2 show that the MAPE of X8306JP and X8411JP are 9.048187 and 12.72942, respectively. Furthermore, accuracy count of the MCMC model for the X8306JP and the X8411JP were better than the ARIMA model, i.e., 88.44953% and 88.59375%, respectively.

The measurement of the performance of the MCMC model for these two datasets with 80-20 and 90-10 ratio shown in Table 6.3 and 6.4. The plots of 80-20 and 90-10 ratio shown in the graphs in Fig. 6.2, 6.5, 6.3 and 6.6, respectively.

Table 6.1 and 6.2 show that the MAPE of X8306JP and X8411JP are 9.048187 and 12.72942, respectively. Furthermore, accuracy count of the MCMC model for X8306JP and X8411JP were better than the ARIMA model, i.e., 88.44953% and 88.59375%, respectively.

As the Table 6.1 and 6.2 and the simulation results of the MCMC model are better than that of the ARIMA model.

Table 6.3 Simulation results using the MCMC model to forecast the original X8306JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.09048187	0.09137966	0.09172327
MAPE	9.048187	9.137966	9.172327
MSE	2056.663	2469.144	3919.649
RMSE	45.35045	49.69048	62.6071
R2	0.9741764	0.9789096	0.9815841
AIC	25292.8	28606.79 31923.19	
BIC	25309.96	28624.35	31941.1
Up-Down(%)	88.44953	88.59375	88.71473

Table 6.4 Simulation results using the MCMC model to forecast the original X8411JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.1272942	0.1514191	0.1677348
MAPE	12.72942	15.14191	16.77348
MSE	716.5013	1004.316	1787.101
RMSE	26.76754	31.69095	42.27412
R2	0.9741764	0.9789096	0.9815841
AIC	23504.61	26531.87	29551.3
BIC	23521.77	26549.43	29569.21
Up-Down(%)	88.44953	88.59375	88.71473

6.4 Support Vector Regression (SVR) Model

Support Vector Machine (SVM) (Premanode, B., 2013, Premanode, B., Vonprasert, J., and Toumazou, C., 2013) is a well-known approach in the machine learning community. It is usually implemented for a classification problem in a supervised learning framework. In case of regression problem, SVM can also be used to predict or explain the values taken by a continuous dependent variable.

6.4.1 Machine Learning

Machine learning is a field in computer science that related from the study of pattern recognition and computational learning theory. It handles the issue of programming systems to learn automatically and improve with experience. For constructing a learning algorithm, complex pattern is recognized and intelligent decisions based on the data are made. The possible decisions are too complex to compute by hand. To solve this problem the machine learning such as artificial neural networks (ANN) and support vector machines (SVM) were developed. Machine learning algorithms commonly use probability theory, logic, optimization, search, statistics, linear algebra and control theory.

In 1946, the first computer system (ENIAC) was developed. This machine was operated manually, i.e., a human would make connections between the parts of the machine to perform computations.

Machine learning algorithm can be organized as follows.

i) Supervised learning creates a function that maps input to desired outputs. A training set of examples with the actual targets is provided and based on this training set; the algorithm generates correctly responses for all possible inputs. Supervised learning is the most famous method.

ii) Unsupervised learning does not give correct responses, then this algorithm attempts to recognize similarities between the inputs.

iii) Reinforcement learning lies between supervised and unsupervised learning. The algorithm is inform when the answer id wrong and there is no expanding pattern to improve performance, then the algorithm carry on repeating the loop until it can find the correct answer.

iv) Evolutionary learning is to learn from biological evolution and adapt to improve the survival rate when the circumstances change.

In 1963, Fisher devise the first algorithm for pattern recognition. Later in 1963, the generalized portrait algorithm, the template for support vector machines (SVMs), was introduced by Vapnik and Lerner. Currently, the performance of SVMs is better than other machine learning methods.

Regularly, SVMs consists of a set of related supervised learning methods. The algorithm indicates a hyperplane that characterizes a functional margin, which holds all possible data points in a finite dimensional nonlinear space. A kernel function $k(x, x')$, defines the cross-products separated by the hyperplane. Each data point shows its vector potential depending on its distance from the hyperplane.

6.4.2 Theoretical Consideration Related to the Support Vector Regression (SVR) Model

SVM can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities.

In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. The support vector algorithm is a nonlinear generalization developed by Vapnik and Lerner in the sixties.

Suppose we have a training data set $(x_1, y_1), \ldots, (x_\ell, y_\ell) \subset X \times \mathbb{R}$, for each $x_i \in X$ (where X denotes the space of the input patterns, e.g. $X = \mathbb{R}^d$) and corresponding value $y_i \in \mathbb{R}$ for $i = 1, \ldots, \ell$. In ϵ-SV regression [Vapnik, 1995], our goal is to find a function $f(x)$ that has at most ϵ deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as possible.

The estimating function f is taken in the form:

$$f(x) = (w \cdot \Phi(x)) + b \tag{6.24}$$

where $w \in \mathbb{R}^m, b \in \mathbb{R}$ is the bias, and Φ is a non-linear function from $\mathbb{R}^n$ to a high dimensional space $\mathbb{R}^m$ $(m > n)$. The objective is to find the values w and b such that the values of $f(x)$ can be determined by minimizing the risk:

$$R_{reg}(f) = C \sum_{i=1}^{n} L_\epsilon(y_i, f(x_i)) + \frac{1}{2}\|w\|^2. \tag{6.25}$$

where L_ϵ is the extension of ϵ-insensitive loss function originally proposed by Vapnik and defined as:

$$L_\epsilon(y, z) = \begin{cases} |y - z| - \epsilon, & |y - z| \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{6.26}$$

Introducing the slack variables ζ_i and ζ_i^* the above problem may be reformulated as

$$\underset{x}{\text{Minimize}} \qquad C\left[\sum_{i=1}^{\ell}(\zeta_i+\zeta_i^*)\right]+\frac{1}{2}\|w\|^2$$

subject to

$$\begin{aligned} y_i - w\cdot\Phi(x_i) - b &\leq \epsilon + \zeta_i \\ w\cdot\Phi(x_i) + b - y_i &\leq \epsilon + \zeta_i^* \\ \zeta_i &\geq 0 \\ \zeta_i^* &\geq 0. \end{aligned} \tag{6.27}$$

for $i = 1, 2, \ldots, \ell$ and where C above is a user specified constant.

Solution of the above problem (6.27) using primal dual method leads to the following dual problem:

Determine the Lagrange multipliers $\{\alpha_i\}_{i=1}^{i}$ and $\{\alpha_i^*\}_{i=1}^{i}$ that maximize the objective function.

$$Q(\alpha_i,\alpha_i^*) = \sum_{i=1}^{\ell} y_i(\alpha_i-\alpha_i^*) - \epsilon\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*) - \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j) \tag{6.28}$$

subjected to the following conditions:

$$(1)\sum_{i=1}^{\ell}(\alpha_i-\alpha_i^*)=0$$

$$(2)\begin{cases} 0\leq\alpha_i\leq C \\ 0\leq\alpha_i^*\leq C \end{cases}$$

for $i = 1, 2, \ldots, \ell$, where C is a user specified constant and $K : X \times X \to \mathbb{R}$ is the Mercer Kernel defined by:

$$K(x,z) = \Phi(x)\cdot\Phi(z) \tag{6.29}$$

This solution of the Primal yields

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \Phi(x_i) \tag{6.30}$$

Then b is calculated using Karush-Kuhn-Tucker (KKT) conditions

$$\begin{aligned} \alpha_i(\varepsilon + \zeta_i - y_i w \cdot \Phi(x_i) + b) &= 0, \\ \alpha_i^*(\varepsilon + \zeta_i + y_i w \cdot \Phi(x_i) - b) &= 0, \\ (C - \alpha_i)\zeta_i &= 0, \\ (C - \alpha_i^*)\zeta_i^* &= 0 \end{aligned} \tag{6.31}$$

for $i = 1, 2, \ldots, \ell$.

Since $\alpha_i, \alpha_i^* = 0$ and $\zeta_i^* = 0$ for $\alpha_i^* \in (0, C)$, then b can be computed as follows:

$$b = y_i - w \cdot \Phi(x_i) - \varepsilon \qquad \text{for} \qquad 0 < \alpha_i < C \tag{6.32}$$

$$b = y_i - w \cdot \Phi(x_i) + \varepsilon \qquad \text{for} \qquad 0 < \alpha_i^* < C \tag{6.33}$$

For those α_i and α^* for which the x_i's corresponding to $0 < \alpha_i < C$ and $0 < \alpha_i^* < C$ are called support vectors. Using expression for w and b in condition (6.31), $f(x)$ is computed as:

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)(\Phi(x_i) \cdot \Phi(x)) + b \tag{6.34}$$

$$= \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{6.35}$$

6.4.3 Simulation and Results of the SVR Model

Similar to Section 6.2, the same datasets, X8306JP and X8411JP, were simulated by R programming scripts. Next, we tested the outcomes of the simulations,

Table 6.5 Simulation results using the SVR model to forecast the original X8306JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.119858	0.0901522	0.06917873
MAPE	11.9858	9.01522	6.917873
MSE	3961.622	2197.663	1531.346
RMSE	62.94142	46.87924	39.13242
R2	0.9974257	0.9977784	0.9979817
AIC	20368.92	23084.57	25793.04
BIC	21140.94	23874.59	26598.95
Up-Down(%)	71.0718	74.21875	74.60815

which were nonlinear and nonstationary, and plotted them against the original test datasets (used as a reference), as shown in Fig. 6.1 and 6.4.

The graphs are shown in Fig. 6.1 and 6.4 where the x-axis represents 963 test data points in the time series and the y-axis represents stock prices in US dollars. At which shows the deviations between the simulated graph of the ARIMA model compared with the original datasets. The two graphs are coincidentally in a line where the x-axis represents the data points in the time series and the y-axis represents the US dollars stock prices. The next step was to measure the performance of the ARIMA model using a variety of loss estimators, i.e., MAE, MAPE, MSE, RMSE, R2, AIC, BIC, and Accuracy count (up-down (%)). Table 6.1 and 6.2 show that the MAPE of the X8306JP and the X8411JP are 11.9858 and 12.72942, respectively. Furthermore, accuracy count of the MCMC model for

Table 6.6 Simulation results using the SVR model to forecast the original X8411JP datasets

Error estimation	ratio		
	70-30	80-20	90-10
MAE	0.1277924	0.09545473	0.08043003
MAPE	12.77924	9.545473	8.043003
MSE	896.418	385.1804	490.3925
RMSE	29.94024	19.62601	22.14481
R2	0.9982478	0.9984209	0.9985441
AIC	17715.23	20131.97	22476.5
BIC	18487.26	20921.99	23282.41
Up-Down(%)	71.38398	71.5625 73.04075	

the X8306JP and the X8411JP were better than the ARIMA model, i.e., 71.0718% and 71.38398%, respectively.

The measurement of the performance of the SVR model for these two datasets with 80-20 and 90-10 ratio shown in Table 6.5 and 6.6. The plots of 80-20 and 90-10 ratio shown in the graphs in Fig. 6.2, 6.5, 6.3 and 6.6, respectively. As the results in Table 6.5 and 6.6, the MAPE of the results of the X8306JP and the X8411JP datasets were decrease to 6.917873 and 8.043003, respectively.

6.5 Simulation Results for ARIMA, MCMC, and SVR

This section shows the graphs of the simulation results for X8306JP and X8411JP with the three models as mentioned before.

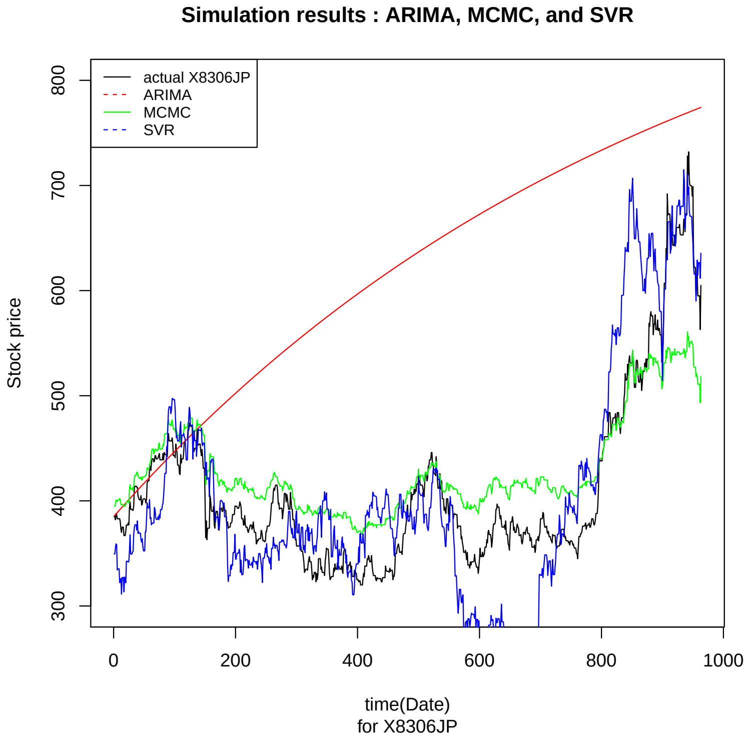

Figure 6.1 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 70-30 for X8306JP

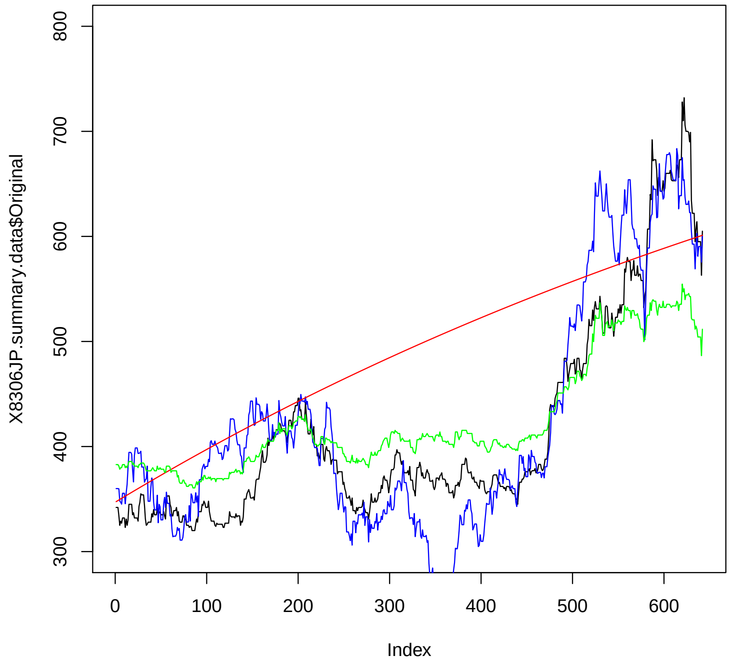

Figure 6.2 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 80-20 for X8306JP

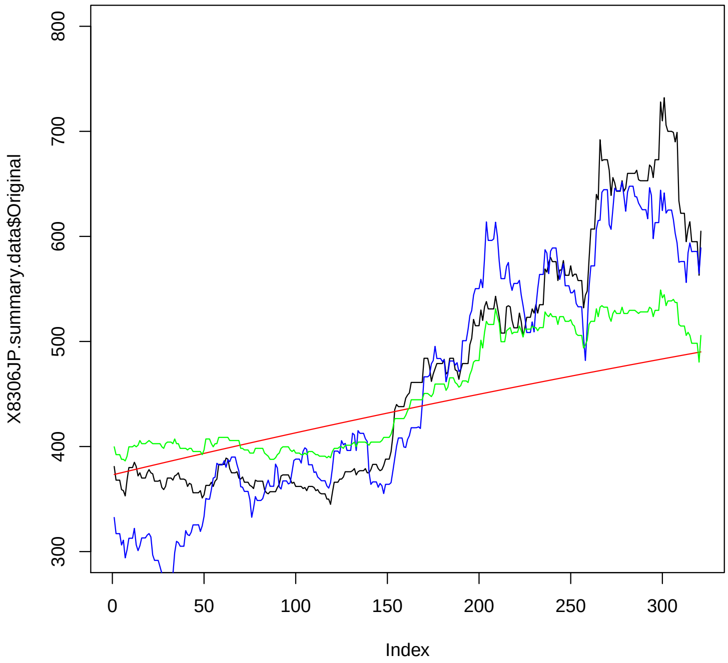

Figure 6.3 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 90-10 for X8306JP

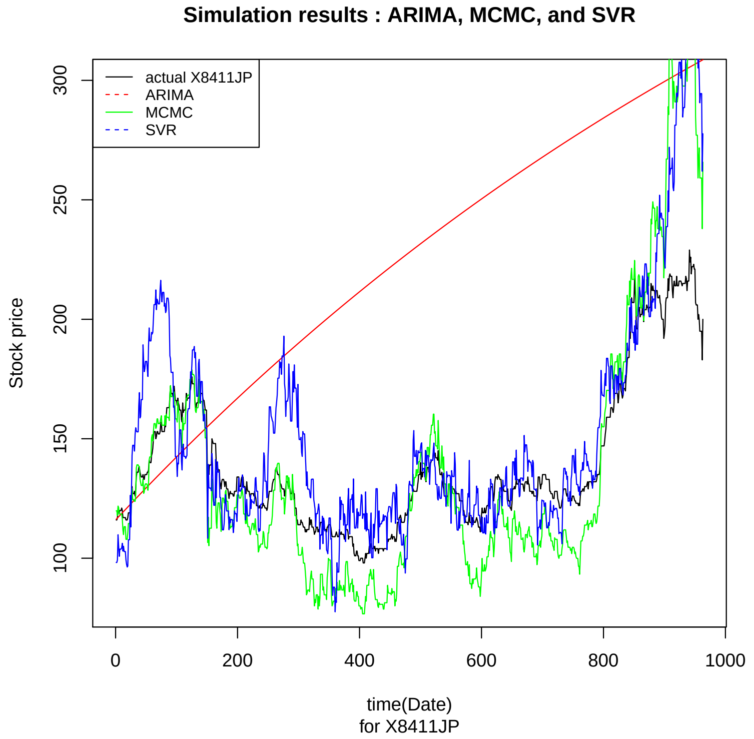

Figure 6.4 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 70-30 for X8411JP

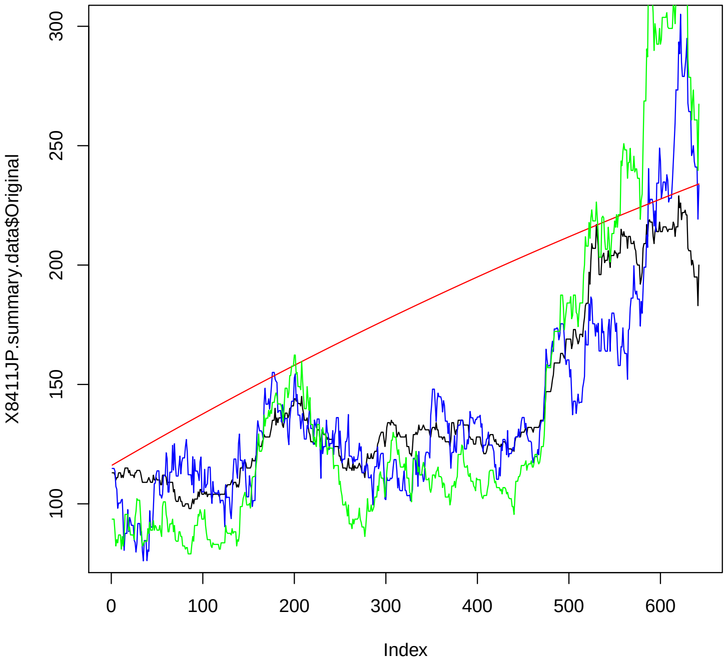

Figure 6.5 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 80-20 for X8411JP

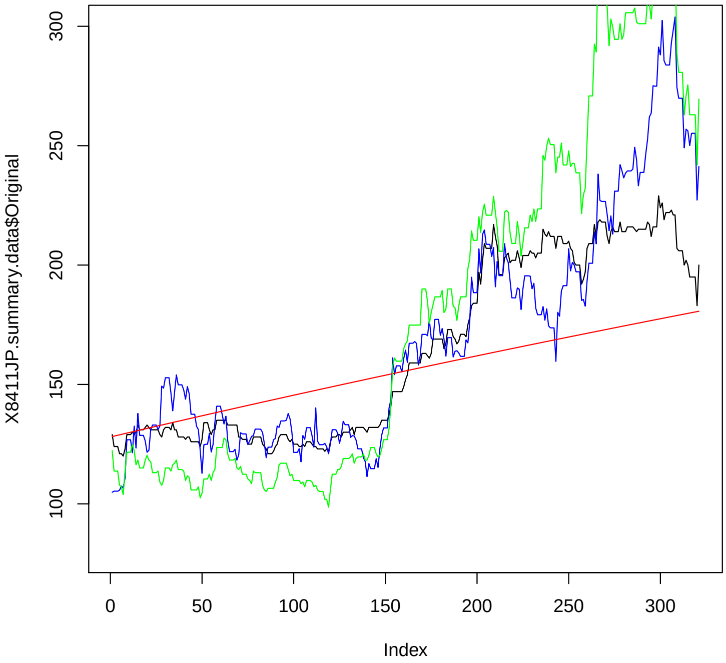

Figure 6.6 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 90-10 for X8411JP

The Fig. 6.1, 6.4, 6.2, 6.5, 6.3, and 6.6 show that the MCMC and the SVR model fit the test datatsets better than the ARIMA model.

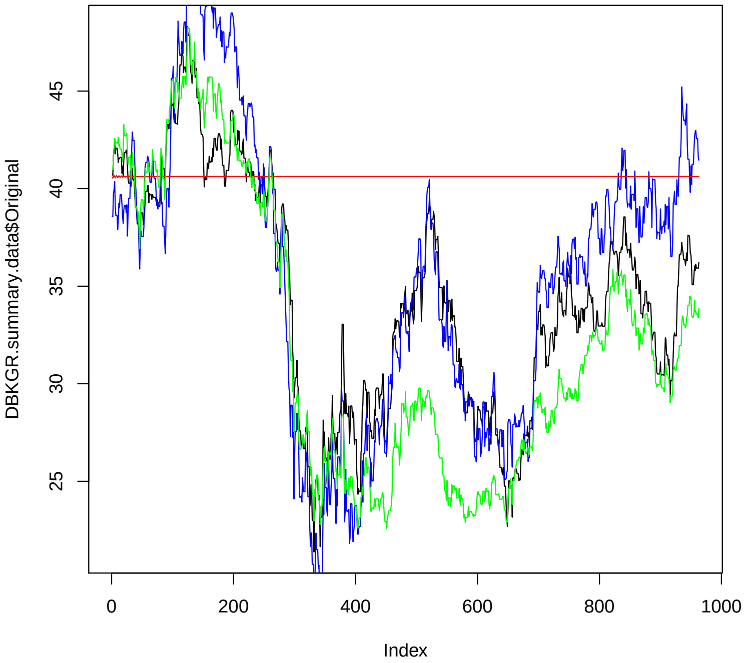

Figure 6.7 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 70-30 for DBKGR

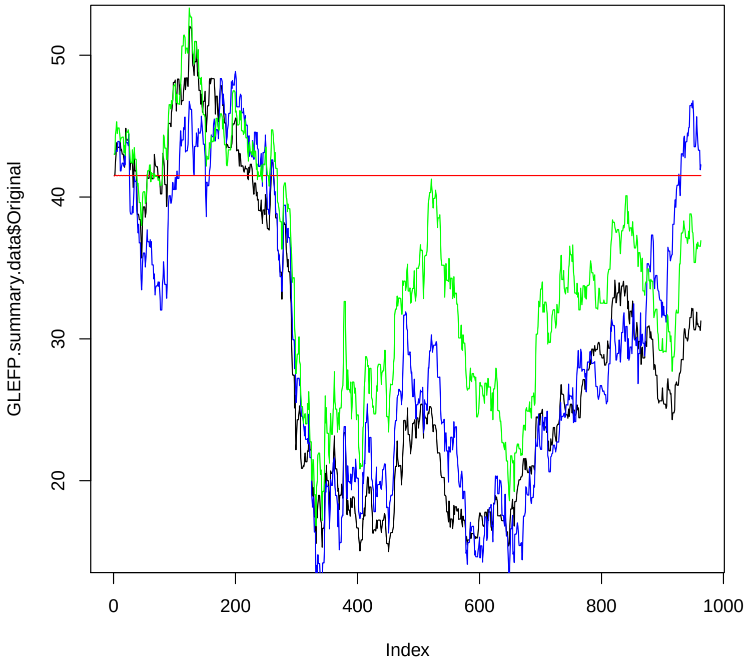

Figure 6.8 The graphs are the simulation using the ARIMA, MCMC, and SVR models with ratio 70-30 for GLEFP

Table 6.7 Simulation results using the ARIMA, MCMC, and SVR models to forecast the DBKGR datasets

Error estimation	ARIMA	MCMC	SVR
MAE	0.233773	0.08146731	0.08291266
MAPE	23.3773	8.146731	8.291266
MSE	72.2152	13.05435	12.47678
RMSE	3.532249	3.61308	3.532249
R2	NA	0.9409209	0.9976014
AIC	6359.17	13794.09	6849.144
BIC	NA	13811.24	7621.166
Up-Down(%)	56.71176	83.35068	77.93965

Table 6.8 Simulation results using the ARIMA, MCMC, and SVR models to forecast the GLEFP datasets

Error estimation	ARIMA	MCMC	SVR
MAE	0.6504123	0.2567641	0.123659
MAPE	65.04123	25.67641	12.3659
MSE	263.7531	49.14653	21.16357
RMSE	4.600389	7.010458	4.600389
R2	NA	0.9409209	0.9978771
AIC	7599.399	15299.97	8080.249
BIC	NA	15317.12	8852.272
Up-Down(%)	61.6025	83.35068	81.06139

The simulation results for the other high correlated coefficient paired stocks, the DBKGR and the GLEFP, with 70-30 ratio shown in Fig. 6.7 and 6.8. The measurement of the performance of the ARIMA, MCMC and SVR models for these two datasets, the DBKGR and the GLEFP , with 70-30 ratio shown in Table 6.7 and 6.8 as well. For the DBKGR, table 6.7 show that the MAPE of the MCMC is like that of the SVR model,8.146731 and 8.291266, respectively. For the GLEFP, table 6.8 show that the MAPE of the MCMC is greater that that of the SVR model,25.67641 and 12.3659, respectively. The Fig. 6.7 and 6.8 show that the SVR model show the best results for the paired stocks, the DBGKR and the GLEFP.

CHAPTER VII

CONCLUSION, DISCUSSION AND FUTURE WORK

The concept of Pairs Trading is a market neutral strategy that uses a portfolio of only two securities. A long position is adopted with respect to one safety and a short position with respect to the other. The strategy of pairs trading requires adopting a position when the spread is distant from the mean in anticipation of spread reversion. This thesis introduces a multi-class Pairs Trading model using Mean Reversion and CV that enhances the original approach of Mean Reversion Pairs Trading. The simulation results show that the co-integrated Pairs trading using the proposed method outperforms those of the conventional co-integrated Pairs Trading. Thus, benefits of the proposed model are to build a new set of risk mitigation and maximise returns of co-integrated stocks. After choosing the paired stocks, if the movement or the future price of the next time step to trade can be predicted, the risk shall be reduced. The simulation results show that the SVR model and the MCMC model outperform those of the ARIMA model. Future research could examine the formation of frequency domain datasets rather than times series as an alternative to correlation coefficient pairing.

REFERENCES

REFERENCES

Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park. (2011). MCMCpack: Markov Chain Monte Carlo in R. **Journal of Statistical Software**.

Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. (2014). Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. **Journal of Applied Mathematics.**

Bao, D. (2008). A generalized model for financial time series representation and prediction. **Springer Science+Business Media.**

Bessembinder, H., Coughenour, J.F., Seguin, P.J., and Smoller, M.M. (1995). Mean Reversion in Equilibrium Asset Prices: Evidence from the Futures Term Structure. **Journal of Finance.** 50(1): 361-375.

Borchers, B. (2002). **Notes on ARIMA Modelling.**

Campbell, J.Y., and Viceira, L. (1999). Consumption and Portfolio Decisions When Expected Returns are Time Varying. **Quarterly Journal of Economics.** 114(2): 433-396.

Campbell, J.Y., Chan, Y.L., and Viceira, L.M. (2003). A Multivariate Model of Strategic Asset Allocation. **Journal of Financial Economics.** 67(1): 41-80.

Carmona, R.(2014). **Statistical Analysis of Financial Data in R: Second Edition**. Springer.

Charles J. Geyer. **Handbook of Markov Chain Monte Carlo**.

Dalgaard, P. (2008). **Introductory Statistics with R: Second Edition**. Springer.

Do, B., Faff, R., and Hamza K. (2006). A New Approach to Modeling and Estimation for Pairs Trading.

Elliott, R.J., Van Der Hoek, J., and Malcolm, W.P. (2005). **Pairs Trading: Quantitative Finance.** 5(3): 271-276.

Fama, E.F., and French, K.R. (1988). Permanent and Temporary Components of Stock Prices. **Journal of Political Economy.** 96: 246-273.

Gatev, E., Goetzmann, W.N., and Rouwenhorst, K.G. (2006). Pairs Trading: Performance of a Relative Value Arbitrage Rule. **The Review of Financial Studies.** 19(3): 797-827.

Herlemont, D.(2004). Pairs Trading, Convergence Trading, Cointegration.

Hillebrand, E. (2004). **A Mean-Reversion Theory of Stock-Market Crashes.** Stanford University. Logged at $http : //www.cofar.uni - mainz.de/dgf2003/paper/paper75.pdf$ (Accessed on August 2,2012).

Huck, N. (2010). Pairs Trading and Outranking: The Multi-step-ahead Forecasting Case. **European Journal of Operational Research.** 207(3): 1702-1706.

Kovalerchuk, B., and Vityaev, E. (2000). **Data Mining in Finance Advances in Relational and Hybrid Methods**. Springer.

Lo, A.W., and Mackinlay, A.C. (1988). Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. **Review of Financial Studies.** 1(1): 41-66.

Mudchanatongsuk, S., Primbs, J.A., and Wong, W. (2008). **Optimal Pairs Trading: A Stochastic Control Approach.** American Control Conference.

Landauskas, M. (2011). Modelling of Stock Prices by the Markov Chain Monte Carlo Method. **Intellectual Economics.** Vol.5 2(10): 244-256.

Perlin, M.S. (2007). **M of a Kind: A Multivariate Approach at Pairs Trading.** Working Paper, Reading University. Logged at $http : //mpra.ub.uni - muenchen.de/8309/1/MPRA_paper_8309.pdf$ (Accessed on April 15, 2013).

Poterba, J., and Summers, L.H. (1988). Mean Reversion in Stock Returns: Evidence and Implications. **Journal of Financial Economics.** 22(1): 27-60.

Premanode, B., Vonprasert, J., and Toumazou, C. (2013). Prediction of Exchange Rates Using Averaging Intrinsic Mode Function and Multiclass Support Vector Regression. **Artificial Intelligence Research.** 2(2): 47-61.

Premanode, B. (2013). Prediction of Nonlinear Nonstationary Time Series Data Using a New Digital Filter and Support Vector Regression.

Ruey S. Tsay. (2002). **Analysis of Financial Time Series.** John Wiley & Sons.

Ruppert, D. (2011). **Statistics and Data Analysis for Financial Engineering** Springer.

Robert H. Shumway, and David S. Stoffer. (2010). **Time Series Analysis and Its Applications With R Examples**. Springer.

Sandro C. A., Vadim di Pietro, and Mark S. Seasholes. (2005). **Understanding the Profitability of Pairs Trading.** Working paper.

Schmidt A. D. (2008). **Pairs Trading: A Cointegration Approach.** Working paper.

Vidyamurthy, G. (2004). **Pairs Trading: Quantitative Methods and Analysis.** John Wiley & Sons.

Wachter, J.A. (2002). Portfolio and Consumption Decisions Under Mean-Reverting Returns: An Exact Solution for Complete Markets. **Journal of Financial and Quantitative Analysis.** 37(1): 63-92.

Winkel, M. (2011). **Part A Simulation.** Lecture.

APPENDIX

APPENDIX A

PROGRAMME FILES

In this appendix, there are Matlab and R scripts programe in this research.

Chapter V and VI Programme

Matlab script

script1 : Pairs Trading

```
clear all
%run for first paired stocks
%read data of 10 pair of stocks
pair10_g1 = xlsread('all.X8306JP7030.xlsx',1);
pair10_g2 = xlsread('all.X8411JP7030.xlsx',1);
[~,num_g1] = size(pair10_g1);
[time,num_g2] = size(pair10_g2);
return_diff = zeros(num_g1,num_g2);
return_cv = zeros(num_g1,num_g2);
return_cv1 = zeros(num_g1,num_g2);
area_diff = zeros(num_g1,num_g2);
for i1 = 1: 1
 for j1 = 1: 1
  xp1 = pair10_g1(:,i1);
  xp2 = pair10_g2(:,j1);
  %calculate mean
  mean_xp1 = mean(xp1);
    mean_xp2 = mean(xp2);
    %calculate sd
    sd_xp1 = std(xp1);
    sd_xp2 = std(xp2);
    %nomalize data
    n_xp1 = zeros(time,1);
    n_xp2 = zeros(time,1);
    for i = 1:time
     n_xp1(i) = (xp1(i)-mean_xp1)/sd_xp1;
     n_xp2(i) = (xp2(i)-mean_xp2)/sd_xp2;
    end
    %calculate mean of normalized data
        mean_nxp1 = mean(n_xp1);
        mean_nxp2 = mean(n_xp2);
        mean_nx1x2 = 0.5*(mean_nxp1+mean_nxp2);
        %calculate sd of normalized data
```

```
sd_nxp1 = std(n_xp1);
sd_nxp2 = std(n_xp2);
sd_nx1x2 = 0.5*(sd_nxp1+sd_nxp2);
%calculate return for xp1 and xp2
return_xp1 = zeros(time,1);
return_xp2 = zeros(time,1);
preturn_xp1 = zeros(time,1);
preturn_xp2 = zeros(time,1);
        %calculate log return for xp1 and xp2
lreturn_xp1 = zeros(time,1);
lreturn_xp2 = zeros(time,1);
vreturn_xp1 = zeros(time,1);
vreturn_xp2 = zeros(time,1);
         for t = 2:time
 return_xp1(t) = (xp1(t)-xp1(t-1))/xp1(t-1);
 preturn_xp1(t) = xp1(t)*return_xp1(t);
 return_xp2(t) = (xp2(t)-xp2(t-1))/xp2(t-1);
 preturn_xp2(t) = xp2(t)*return_xp2(t);
 lreturn_xp1(t) = log(xp1(t)/xp1(t-1));
 vreturn_xp1(t) = xp1(t)*lreturn_xp1(t);
 lreturn_xp2(t) = log(xp2(t)/xp2(t-1));
 vreturn_xp2(t) = xp2(t)*lreturn_xp2(t);
end
%calculate average return
avr_return_xp1 = mean(return_xp1);
avr_return_xp2 = mean(return_xp2);
%set
avr_return_cv = zeros(6,2);
%------------ 1st stock ----------
%for xp1
%set class for xp1
group_xp1_temp = zeros(time,1);
group_xp1 = zeros(time,1);
group_xp2 = zeros(time,1);
%find the 1st mean reverse
for k1 = 1:time
 if xp1(k1) <= mean_xp1
  group_xp1_temp(k1) = 1;
 elseif xp1(k1) > mean_xp1
  group_xp1_temp(k1) = 2;
 end
end
%consider group 1
%set c_xp1
xp1_1 = find(group_xp1_temp ==1);
c1_xp1 = zeros(size(xp1_1,1),1);
for i = 1:size(xp1_1,1)
    c1_xp1(i) = xp1(xp1_1(i));
end
% find 2nd mean reverse
%calculate mean for class 1
m_c1_xp1 = mean(c1_xp1);
sd_c1_xp1 = std(c1_xp1);
```

```
%------------lower mean----------
%set class for xp1
class_xp1 = zeros(size(xp1_1,1),1);
for i = 1:size(xp1_1,1)
 if c1_xp1(i)<= (m_c1_xp1 - sd_c1_xp1)
        class_xp1(i) = 1;
        group_xp1(xp1_1(i)) = class_xp1(i);
 elseif c1_xp1(i) > (m_c1_xp1 - sd_c1_xp1) && c1_xp1(i) < (m_c1_xp1 + sd_c1_xp1)
        class_xp1(i) = 2;
        group_xp1(xp1_1(i)) = class_xp1(i);
 elseif c1_xp1(i)>= m_c1_xp1 + sd_c1_xp1
        class_xp1(i) = 3;
        group_xp1(xp1_1(i)) = class_xp1(i);
 end
end
new_c1_xp1 = [c1_xp1 class_xp1];
%set CV
cv1 = zeros(6,1);
num_CV = zeros(6,2);
%------------class 1---------------
%calculate mean, var, cv for class 1
c11_xp1_temp = find(class_xp1 == 1);
c11_xp1 = zeros(size(c11_xp1_temp,1),1);
num_CV(1,1) = size(c11_xp1_temp,1);
for i = 1:size(c11_xp1_temp,1)
    c11_xp1(i) = c1_xp1(c11_xp1_temp(i));
end
m_c11_xp1 = mean(c11_xp1);
var_c11_xp1 = std(c11_xp1)^2;
cv1(1) = std(c11_xp1)/m_c11_xp1;
%calculate return
return_cv11 = zeros(size(c11_xp1_temp,1),1);
for i = 2:size(c11_xp1_temp,1)
    return_cv11(i) = log(c11_xp1(i)/c11_xp1(i-1));
end
avr_return_cv(1,1) = mean(return_cv11);
%------------------class 2-----------------
%calculate mean, var, cv for class 1
c12_xp1_temp = find(class_xp1 == 2);
c12_xp1 = zeros(size(c12_xp1_temp,1),1);
num_CV(2,1) = size(c12_xp1_temp,1);
for i = 1:size(c12_xp1_temp,1)
    c12_xp1(i) = c1_xp1(c12_xp1_temp(i));
end
m_c12_xp1 = mean(c12_xp1);
var_c12_xp1 = std(c12_xp1)^2;
cv1(2) = std(c12_xp1)/m_c12_xp1;
%calculate return
return_cv12 = zeros(size(c12_xp1_temp,1),1);
for i = 2:size(c12_xp1_temp,1)
    return_cv12(i) = log(c12_xp1(i)/c12_xp1(i-1));
end
avr_return_cv(2,1) = mean(return_cv12);
```

```
%------------------class 3------------
    %calculate mean, var, cv for class 1
    c13_xp1_temp = find(class_xp1 == 3);
    c13_xp1 = zeros(size(c13_xp1_temp,1),1);
    num_CV(3,1) = size(c13_xp1_temp,1);
    for i = 1:size(c13_xp1_temp,1)
        c13_xp1(i) = c1_xp1(c13_xp1_temp(i));
    end
    m_c13_xp1 = mean(c13_xp1);
    var_c13_xp1 = std(c13_xp1)^2;
    cv1(3) = std(c13_xp1)/m_c13_xp1;
    %calculate return
    return_cv13 = zeros(size(c13_xp1_temp,1),1);
    for i = 2:size(c13_xp1_temp,1)
        return_cv13(i) =
        log(c13_xp1(i)/c13_xp1(i-1));
    end
    avr_return_cv(3,1) = mean(return_cv13);

    %consider group 2
    %set c_xp2
    xp1_2 = find(group_xp1_temp == 2);
    c2_xp1 = zeros(size(xp1_2,1),1);
    for i = 1:size(xp1_2,1)
        c2_xp1(i) = xp1(xp1_2(i));
    end
    % find 2nd mean reverse
    %calculate mean for class 1
    m_c2_xp1 = mean(c2_xp1);
    sd_c2_xp1 = std(c2_xp1);
%---------------upper mean----------------
    %set class for xp1
    class_xp12 = zeros(size(xp1_2,1),1);
    for i = 1:size(xp1_2,1)
        if c2_xp1(i)<= (m_c2_xp1 - sd_c2_xp1)
            class_xp12(i) = 4;
            group_xp1(xp1_2(i)) = class_xp12(i);
        elseif c2_xp1(i) > (m_c2_xp1 - sd_c2_xp1) && c2_xp1(i) < (m_c2_xp1 + sd_c2_xp1)
            class_xp12(i) = 5;
            group_xp1(xp1_2(i)) = class_xp12(i);
        elseif c2_xp1(i)>= m_c2_xp1 + sd_c2_xp1
            class_xp12(i) = 6;
            group_xp1(xp1_2(i)) = class_xp12(i);
        end
    end
    new_c2_xp1 = [c2_xp1 class_xp12];
%----------------------class 4---------------
    %calculate mean, var, cv for class 4
    c14_xp1_temp = find(class_xp12 == 4);
    c14_xp1 = zeros(size(c14_xp1_temp,1),1);
    num_CV(4,1) = size(c14_xp1_temp,1);
    for i = 1:size(c14_xp1_temp,1)
        c14_xp1(i) = c2_xp1(c14_xp1_temp(i));
    end
    m_c14_xp1 = mean(c14_xp1);
    var_c14_xp1 = std(c14_xp1)^2;
    cv1(4) = std(c14_xp1)/m_c14_xp1;
    %calculate return
    return_cv14 = zeros(size(c14_xp1_temp,1),1);
    for i = 2:size(c14_xp1_temp,1)
        return_cv14(i) = log(c14_xp1(i)/c14_xp1(i-1));
    end
    avr_return_cv(4,1) = mean(return_cv14);
%-------------------class 5-------------
    %calculate mean, var, cv for class 5
    c15_xp1_temp = find(class_xp12 == 5);
    c15_xp1 = zeros(size(c15_xp1_temp,1),1);
    num_CV(5,1) = size(c15_xp1_temp,1);
    for i = 1:size(c15_xp1_temp,1)
        c15_xp1(i) = c2_xp1(c15_xp1_temp(i));
    end
    m_c15_xp1 = mean(c15_xp1);
    var_c15_xp1 = std(c15_xp1)^2;
    cv1(5) = std(c15_xp1)/m_c15_xp1;
    %calculate return
    return_cv15 = zeros(size(c15_xp1_temp,1),1);
    for i = 2:size(c15_xp1_temp,1)
        return_cv15(i) = log(c15_xp1(i)/c15_xp1(i-1));
    end
    avr_return_cv(5,1) = mean(return_cv15);
%---------------------class 6-------------------
    %calculate mean, var, cv for class 6
    c16_xp1_temp = find(class_xp12 == 6);
    c16_xp1 = zeros(size(c16_xp1_temp,1),1);
    num_CV(6,1) = size(c16_xp1_temp,1);
    for i = 1:size(c16_xp1_temp,1)
        c16_xp1(i) = c2_xp1(c16_xp1_temp(i));
    end
    m_c16_xp1 = mean(c16_xp1);
    var_c16_xp1 = std(c16_xp1)^2;
    cv1(6) = std(c16_xp1)/m_c16_xp1;
    %calculate return
    return_cv16 = zeros(size(c16_xp1_temp,1),1);
    for i = 2:size(c16_xp1_temp,1)
        return_cv16(i) = log(c16_xp1(i)/c16_xp1(i-1));
    end
    avr_return_cv(6,1) = mean(return_cv16);
%---------- 2nd stock ---------------
%-----------------------------------------
    %for xp2
    %set class for xp2
    group_xp2_temp = zeros(time,1);

    %find the 1st mean reverse
    for i = 1:time
        if xp2(i) <= mean_xp2
```

```
            group_xp2_temp(i) = 1;
        elseif xp2(i) > mean_xp2
            group_xp2_temp(i) = 2;
        end
    end
    %consider group 1
    %set c_xp1
    xp2_1 = find(group_xp2_temp ==1);
    c1_xp2 = zeros(size(xp2_1,1),1);
    for i = 1:size(xp2_1,1)
        c1_xp2(i) = xp2(xp2_1(i));
    end
    % find 2nd mean reverse
    %calculate mean for class 1
    m_c1_xp2 = mean(c1_xp2);
    sd_c1_xp2 = std(c1_xp2);
%---------------lower mean---------
    %set class for xp2
    class_xp2 = zeros(size(xp2_1,1),1);
    for i = 1:size(xp2_1,1)
     if c1_xp2(i)<= (m_c1_xp2 - sd_c1_xp2)
            class_xp2(i) = 1;
            group_xp2(xp2_1(i)) = class_xp2(i);
     elseif c1_xp2(i) > (m_c1_xp2 - sd_c1_xp2)&& c1_xp2(i) < (m_c1_xp2 + sd_c1_xp2)
            class_xp2(i) = 2;
            group_xp2(xp2_1(i)) = class_xp2(i);
     elseif c1_xp2(i)>= m_c1_xp2 + sd_c1_xp2
            class_xp2(i) = 3;
            group_xp2(xp2_1(i)) = class_xp2(i);
     end
    end
    new_c1_xp2 = [c1_xp2 class_xp2];
    %set CV
    cv2 = zeros(6,1);
%---------------------class 1--------------
    %calculate mean, var, cv for class 1
    c11_xp2_temp = find(class_xp2 == 1);
    c11_xp2 = zeros(size(c11_xp2_temp,1),1);
    num_CV(1,2) = size(c11_xp2_temp,1);
    for i = 1:size(c11_xp2_temp,1)
        c11_xp2(i) = c1_xp2(c11_xp2_temp(i));
    end
    m_c11_xp2 = mean(c11_xp2);
    var_c11_xp2 = std(c11_xp2)^2;
    cv2(1) = std(c11_xp2)/m_c11_xp2;
    %calculate return
    return_cv21 = zeros(size(c11_xp2_temp,1),1);
    for i = 2:size(c11_xp2_temp,1)
      return_cv21(i) = log(c11_xp2(i)/c11_xp2(i-1));
    end
    avr_return_cv(1,2) = mean(return_cv21);
%---------------------class 2-------------
    %calculate mean, var, cv for class 2
    c12_xp2_temp = find(class_xp2 == 2);
    c12_xp2 = zeros(size(c12_xp2_temp,1),1);
    num_CV(2,2) = size(c12_xp2_temp,1);
    for i = 1:size(c12_xp2_temp,1)
        c12_xp2(i) = c1_xp2(c12_xp2_temp(i));
    end
    m_c12_xp2 = mean(c12_xp2);
    var_c12_xp2 = std(c12_xp2)^2;
    cv2(2) = std(c12_xp2)/m_c12_xp2;
    %calculate return
    return_cv22 = zeros(size(c12_xp2_temp,1),1);
    for i = 2:size(c12_xp2_temp,1)
      return_cv22(i) = log(c12_xp2(i)/c12_xp2(i-1));
    end
    avr_return_cv(2,2) = mean(return_cv22);
%---------------------class 3----------
    %calculate mean, var, cv for class 3
    c13_xp2_temp = find(class_xp2 == 3);
    c13_xp2 = zeros(size(c13_xp2_temp,1),1);
    num_CV(3,2) = size(c13_xp2_temp,1);
    for i = 1:size(c13_xp2_temp,1)
        c13_xp2(i) = c1_xp2(c13_xp2_temp(i));
    end
    m_c13_xp2 = mean(c13_xp2);
    var_c13_xp2 = std(c13_xp2)^2;
    cv2(3) = std(c13_xp2)/m_c13_xp2;
    %calculate return
    return_cv23 = zeros(size(c13_xp2_temp,1),1);
    for i = 2:size(c13_xp2_temp,1)
        return_cv23(i) = log(c13_xp2(i)/c13_xp2(i-1));
    end
    avr_return_cv(3,2) = mean(return_cv23);
    %consider group 2
    %set c_xp1
    xp2_2 = find(group_xp2_temp == 2);
    c2_xp2 = zeros(size(xp2_2,1),1);
    for i = 1:size(xp2_2,1)
        c2_xp2(i) = xp2(xp2_2(i));
    end
    % find 2nd mean reverse
    %calculate mean for class 1
    m_c2_xp2 = mean(c2_xp2);
    sd_c2_xp2 = std(c2_xp2);
    %-----------------upper mean-----
    %set class for xp1
    class_xp22 = zeros(size(xp2_2,1),1);
    for i = 1:size(xp2_2,1)
     if c2_xp2(i)<= (m_c2_xp2 - sd_c2_xp2)
            class_xp22(i) = 4;
            group_xp2(xp2_2(i)) = class_xp22(i);
     elseif c2_xp2(i) > (m_c2_xp2 - sd_c2_xp2) && c2_xp2(i) < (m_c2_xp2 + sd_c2_xp2)
            class_xp22(i) = 5;
            group_xp2(xp2_2(i)) = class_xp22(i);
```

```
 elseif c2_xp2(i)>= m_c2_xp2 + sd_c2_xp2
        class_xp22(i) = 6;
        group_xp2(xp2_2(i)) = class_xp22(i);
 end
end
new_c2_xp2 = [c2_xp2 class_xp22];
%-------------------class 4-----------
%calculate mean, var, cv for class 4
c14_xp2_temp = find(class_xp22 == 4);
c14_xp2 = zeros(size(c14_xp2_temp,1),1);
num_CV(4,2) = size(c14_xp2_temp,1);
for i = 1:size(c14_xp2_temp,1)
    c14_xp2(i) = c2_xp2(c14_xp2_temp(i));
end
m_c14_xp2 = mean(c14_xp2);
var_c14_xp2 = std(c14_xp2)^2;
cv2(4) = std(c14_xp2)/m_c14_xp2;
%calculate return
return_cv24 = zeros(size(c14_xp2_temp,1),1);
for i = 2:size(c14_xp2_temp,1)
    return_cv24(i) = log(c14_xp2(i)/c14_xp2(i-1));
end
avr_return_cv(4,2) = mean(return_cv24);
%------------------class 5----------
%calculate mean, var, cv for class 5
c15_xp2_temp = find(class_xp22 == 5);
c15_xp2 = zeros(size(c15_xp2_temp,1),1);
num_CV(5,2) = size(c15_xp2_temp,1);
for i = 1:size(c15_xp2_temp,1)
    c15_xp2(i) = c2_xp2(c15_xp2_temp(i));
end
m_c15_xp2 = mean(c15_xp2);
var_c15_xp2 = std(c15_xp2)^2;
cv2(5) = std(c15_xp2)/m_c15_xp2;
%calculate return
return_cv25 = zeros(size(c15_xp2_temp,1),1);
for i = 2:size(c15_xp2_temp,1)
    return_cv25(i) =
    log(c15_xp2(i)/c15_xp2(i-1));
end
avr_return_cv(5,2) = mean(return_cv25);
%-------------------class 6--------
%calculate mean, var, cv for class 6
c16_xp2_temp = find(class_xp22 == 6);
c16_xp2 = zeros(size(c16_xp2_temp,1),1);
num_CV(6,2) = size(c16_xp2_temp,1);
for i = 1:size(c16_xp2_temp,1)
    c16_xp2(i) = c2_xp2(c16_xp2_temp(i));
end
m_c16_xp2 = mean(c16_xp2);
var_c16_xp2 = std(c16_xp2)^2;
cv2(6) = std(c16_xp2)/m_c16_xp2;
%calculate return
```

```
        return_cv26 = zeros(size(c16_xp2_temp,1),1);
        for i = 2:size(c16_xp2_temp,1)
         return_cv26(i) = log(c16_xp2(i)/c16_xp2(i-1));
        end
avr_return_cv(6,2) = mean(return_cv26);
%CV
CV = [cv1 cv2];
avr_return_cv;
%--------------------calculate prob. using MC-------------
%for x1 and x2
no_p1 = zeros(6,6);
no_p2 = zeros(6,6);
for pp = 1 : time-1
    for mm = 1 : 6
        if group_xp1(pp) == mm
            for cc = 1: 6
                if group_xp1(pp+1) == cc
                 no_p1(mm,cc) = no_p1(mm,cc)+1;
                end
            end
        end
    end
end
for pp = 1 : time-1
    for mm = 1 : 6
        if group_xp2(pp) == mm
            for cc = 1: 6
                if group_xp2(pp+1) == cc
                 no_p2(mm,cc) = no_p2(mm,cc)+1;
                end
            end
        end
    end
end
%calculate transition matrix
p_no1 = [no_p1(1,:)/sum(no_p1(1,:))  ;
        no_p1(2,:)/sum(no_p1(2,:));
    no_p1(3,:)/sum(no_p1(3,:))  ;
    no_p1(4,:)/sum(no_p1(4,:))  ;
    no_p1(5,:)/sum(no_p1(5,:))  ;
    no_p1(6,:)/sum(no_p1(6,:))  ];
p_no2 = [no_p2(1,:)/sum(no_p2(1,:))  ;
        no_p2(2,:)/sum(no_p2(2,:));
    no_p2(3,:)/sum(no_p2(3,:))  ;
    no_p2(4,:)/sum(no_p2(4,:))  ;
    no_p2(5,:)/sum(no_p2(5,:))  ;
    no_p2(6,:)/sum(no_p2(6,:))  ];
        %case : trad every day
        %for x1
        sum1_all = 0;
        trade1_all = zeros(time,1);
        w1_all = zeros(time,1);
        for tt = 1: time
```

```
            trade1_all(tt) = 1;
            sum1_all = sum1_all+trade1_all(tt);
            w1_all(tt) = 1/sum1_all;
        end
        %for x2
        sum2_all = 0;
        trade2_all = zeros(time,1);
        w2_all = zeros(time,1);
        for tt = 1: time
             trade2_all(tt) = 1;
             sum2_all = sum2_all+trade2_all(tt);
             w2_all(tt) = 1/sum2_all;
        end
        %calculate return
        case1_RE = zeros(time,1);
        c = 0.25;
        tc = 2*log((1-c)/(1+c));
        %calculate return : case trade every day
        case0_RE = zeros(time,1);
        for i = 1:time
             if xp1(i) < xp2(i)
             %long x1, short x2
                  case0_RE(i) =
                  lreturn_xp1(i)*w1_all(i) -lreturn_xp2(i)*w2_all(i) + tc;
             elseif xp1(i) > xp2(i)
             %long x2, short x1
                  case0_RE(i) =
                  -lreturn_xp1(i)*w1_all(i) +lreturn_xp2(i)*w2_all(i) + tc;
             else case0_RE(i) = 0;
             end
        end
        return_case0 = sum(case0_RE);
% case : CV
        %for x1
        sum1 = 0;
        trade1 = zeros(time,1);
        w1 = zeros(time,1);
        pp_no_1 = zeros(time,1);
    for tt = 1: time-2
      if group_xp1(tt+1) == group_xp1(tt)
         if group_xp1(tt+2) == group_xp1(tt+1)
            trade1(tt+1) = 1;
%prob. of cv of time tt+1 given cv of time tt+1
            pp_no_1(tt+1) =
            p_no1(group_xp1(tt),
            group_xp1(tt+1));
            sum1 = sum1+trade1(tt+1);
             w1(tt+1) = 1/sum1;
         else trade1(tt+1) = 0;
             sum1 = sum1+trade1(tt+1);
             w1(tt+1) = 1/sum1;
         end
       else trade1(tt+1) = 0;
```

```
         sum1 = sum1+trade1(tt+1);
          w1(tt+1) = 1/sum1;
       end
     end
        % weight for x1
        %num_trade1 = sum(trade1);
        nw1 = 1/sum(trade1);

        %for x2
        sum2 = 0;
        trade2 = zeros(time,1);
        w2 = zeros(time,1);
        pp_no_2 = zeros(time,1);
   for tt = 1: time-2
      if group_xp2(tt+1) == group_xp2(tt)
         if group_xp2(tt+2) == group_xp2(tt+1)
            trade2(tt+1) = 1;
%prob. of cv of time tt+1 given cv of time tt+1
           pp_no_2(tt+1) = p_no2(group_xp2(tt),group_xp2(tt+1));
           sum2 = sum2+trade2(tt+1);
           w2(tt+1) = 1/sum2;
         else trade2(tt+1) = 0;
            sum2 = sum2+trade2(tt+1);
            w2(tt+1) = 1/sum2;
         end
      else trade2(tt+1) = 0;
         sum2 = sum2+trade2(tt+1);
         w2(tt+1) = 1/sum2;
       end
   end
        % weight for x2
        nw2 = 1/sum(trade2);
        profit_xp1 = sum(preturn_xp1);
        %calculate diff
        diff = zeros(time,1);
        for i = 1 :time
            diff(i) = abs(xp1(i)-xp2(i));
        end
        area_diff(i1,j1) = sum(diff);
        %calculate return using diff
        t1 = zeros(time,1);
        t2 = zeros(time,1);
        sw1=0;
        sw2=0;
        wt1 = zeros(time,1);
        wt2 = zeros(time,1);
        for i = 1:time
          if diff(i) >= 0.1*min(xp1(i),xp2(i))
                t1(i) = 1;
                t2(i) = 1;
                sw1 = sw1 +t1(i);
                sw2 = sw2 +t2(i);
                wt1(i) = 1/sw1;
```

```
                    wt2(i) = 1/sw2;
                    if xp1(i) < xp2(i)
                    %long x1, short x2
                      case1_RE(i) =
                      lreturn_xp1(i)*wt1(i) - lreturn_xp2(i)*wt2(i) + tc;
                    elseif xp1(i) > xp2(i)
                    %long x2, short x1
                      case1_RE(i) =
                      -lreturn_xp1(i)*wt1(i) +lreturn_xp2(i)*wt2(i) + tc;
                    else case1_RE(i) = 0;
                    end
             else case1_RE(i) =0;
             end
          end
          sum_case1_RE = sum(case1_RE);
          prob_return = sum_case1_RE;
          %calculate ratio between x1 and x2
          xp1_xp2 = zeros(time,1);
          for i = 1 : time
               xp1_xp2(i) = xp1(i)/xp2(i);
          end
          %calculate return with CV
          case2_RE = zeros(time,1);
          case21_RE = zeros(time,1);
          num_trade1 = sum(trade1);
          num_trade2 = sum(trade2);
    for i = 1:time
        if trade1(i) == 1
            if trade2(i) == 1
            %long x1, short x2
             if diff(i) >= 0.1*min(xp1(i),xp2(i))
               if xp1(i) < xp2(i)
                %long x1, short x2
                  case2_RE(i) =
                  lreturn_xp1(i)*w1(i)*pp_no_1(i)-lreturn_xp2(i)*w2(i)*pp_no_2(i) + tc;
                  case21_RE(i) =
                  lreturn_xp1(i)*w1(i)-lreturn_xp2(i)*w2(i) + tc;
               elseif xp1(i) > xp2(i)
                %long x2, short x1
                  case2_RE(i) =
                 -lreturn_xp1(i)*w1(i)*pp_no_1(i)+ lreturn_xp2(i)*w2(i)*pp_no_2(i) + tc;
                  case21_RE(i) =
                 -lreturn_xp1(i)*w1(i)+lreturn_xp2(i)*w2(i) + tc;
               end
             end
           end
       end
     end
          dif_case_2_21 = abs(case21_RE-case2_RE);
          sum_case2_RE_1 =sum(case2_RE);
          return_diff(i1,j1) = 0.75*sum_case1_RE;
          return_cv(i1,j1) = sum_case2_RE_1;
%consider for each range of time to trade without CV
```

```
for kkk = 1: 3
        d_pair = [t1 t2];
        kd_pair = zeros(time+1,1);
        d_pair1 = zeros(time,1);
        for i = 1:time
          if t1(i) == 1
              if t2(i) == 1
               if diff(i) >= 0.1*min(xp1(i),xp2(i))
                    kd_pair(i) = i;
                    d_pair1(i) = 1;
               end
              end
          end
        end
        pd_t0 = zeros(time,1);
        pd_t1 = zeros(time,1);
        temp_td = zeros(time,1);
        ptd = 1;
        for i= 1 : time
            if kd_pair(i)>0
                temp_td(i) = kd_pair(i);
                if kd_pair(i+1) == 0
                    pd_t1(ptd) = max(temp_td);
                    temp_td(~temp_td) = nan;
                    pd_t0(ptd) = min(temp_td);
                    temp_td = zeros(time,1);
                    ptd = ptd+1;
                end
            end
        end
        pd_t0 = pd_t0(isfinite(pd_t0));
        pd_t0 = pd_t0(pd_t0~= 0);
        pd_t1 = pd_t1(isfinite(pd_t1));
        pd_t1 = pd_t1(pd_t1~= 0);
        no_pd_t = zeros(time,1);
        [std1 rr] = size(pd_t0);
        for i = 1 : std1
            no_pd_t(i) = pd_t1(i)-pd_t0(i)+1;
        end
        no_pd_t = no_pd_t(no_pd_t~= 0);
        % cutting trading time  < 3
        for i = 1 : std1
            if no_pd_t(i) < 3
                no_pd_t(i) = 0;
                cut0 = pd_t0(i);
                cut1 = pd_t1(i);
                for ic = cut0 : cut1
                    t1(ic) = 0;
                    t2(ic) = 0;
                end
            end
        end
end
```

```
 %price block
    xp1_0d_block = zeros(std1,1);
    xp2_0d_block = zeros(std1,1);
    xp1_1d_block = zeros(std1,1);
    xp2_1d_block = zeros(std1,1);
    for i = 1 : std1
        xp1_0d_block(i) = xp1(pd_t0(i));
        xp1_1d_block(i) = xp1(pd_t1(i));
        xp2_0d_block(i) = xp2(pd_t0(i));
        xp2_1d_block(i) = xp2(pd_t1(i));
    end
xd_block = [xp1_0d_block xp1_1d_block
xp2_0d_block xp2_1d_block ];
    %calculate return using diff
    t1 = zeros(time,1);
    t2 = zeros(time,1);
    sw1=0;
    sw2=0;
    wt1 = zeros(time,1);
    wt2 = zeros(time,1);
    for i = 1:time
        if diff(i) >= 0.1*min(xp1(i),xp2(i))
            t1(i) = 1;
            t2(i) = 1;
            sw1 = sw1 +t1(i);
            sw2 = sw2 +t2(i);
            wt1(i) = 1/sw1;
            wt2(i) = 1/sw2;
            if xp1(i) < xp2(i)
            %long x1, short x2
                case1_RE(i) =
                lreturn_xp1(i)*wt1(i) - lreturn_xp2(i)*wt2(i) + tc;
            elseif xp1(i) > xp2(i)
            %long x2, short x1
                case1_RE(i) =
                -lreturn_xp1(i)*wt1(i) +lreturn_xp2(i)*wt2(i) + tc;
            else case1_RE(i) = 0;
            end
        else case1_RE(i) =0;
        end
    end
    sum_case1_RE = sum(case1_RE);
    prob_return = 0.75*sum_case1_RE;
    return_diff(i1,j1) = sum_case1_RE;
%return block
    red_block = zeros(std1,1);
    for i = 1 : std1
        tt0 = pd_t0(i);
        tt1 = pd_t1(i);
        for ib = tt0 : tt1
            red_block(i) =
            red_block(i)+ case1_RE(ib);
        end
    end
    returnd_block = abs(red_block);
%consider for each range of time to trade
for kk = 1 : 3
    trade_pair = [trade1 trade2];
    k_pair = zeros(time+1,1);
    trade_pair1 = zeros(time,1);
    for i = 1:time
        if trade1(i) == 1
            if trade2(i) == 1
                if diff(i) >=
                0.1*min(xp1(i),xp2(i))
                    k_pair(i) = i;
                    trade_pair1(i) = 1;
                end
            end
        end
    end
    case4_RE = zeros(time,1);
    sum_case4 = zeros(time,1);
    p_t0 = zeros(time,1);
    p_t1 = zeros(time,1);
    temp_t = zeros(time,1);
    pt = 1;
    for i= 1 : time
        if k_pair(i)>0
            temp_t(i) = k_pair(i);
            if k_pair(i+1) == 0
                p_t1(pt) = max(temp_t);
                temp_t(~temp_t) = nan;
                p_t0(pt) = min(temp_t);
                temp_t = zeros(time,1);
                pt = pt+1;
            end
        end
    end
    p_t0 = p_t0(isfinite(p_t0));
    p_t0 = p_t0(p_t0~= 0);
    p_t1 = p_t1(isfinite(p_t1));
    p_t1 = p_t1(p_t1~= 0);
    no_p_t = zeros(time,1);
    for i = 1 :size(p_t0)
        no_p_t(i) = p_t1(i)-p_t0(i)+1;
    end
    no_p_t = no_p_t(no_p_t~= 0);
    [st rr1] = size(p_t0);
    % cutting trading time < 3
    for i = 1 : st
        if no_p_t(i) < 3
            no_p_t(i) = 0;
            cut0 = p_t0(i);
            cut1 = p_t1(i);
            for ic = cut0 : cut1
```

```
                    trade1(ic) = 0;
                    trade2(ic) = 0;
                end
            end
        end
end
 %price block
        xp1_0_block = zeros(st,1);
        xp2_0_block = zeros(st,1);
        xp1_1_block = zeros(st,1);
        xp2_1_block = zeros(st,1);
        for i = 1 : st
            xp1_0_block(i) = xp1(p_t0(i));
            xp1_1_block(i) = xp1(p_t1(i));
            xp2_0_block(i) = xp2(p_t0(i));
            xp2_1_block(i) = xp2(p_t1(i));
            xp1_cv_block = group_xp1(p_t0(i));
            xp2_cv_block = group_xp1(p_t0(i));
        end
        x_block = [xp1_0_block xp1_1_block
        xp2_0_block xp2_1_block ];
%CV block
        xp1_0_cv_block = zeros(st,1);
        xp2_0_cv_block = zeros(st,1);
        xp1_1_cv_block = zeros(st,1);
        xp2_1_cv_block = zeros(st,1);
        for i = 1 : st
         xp1_0_cv_block(i) = group_xp1(p_t0(i));
         xp2_0_cv_block(i) = group_xp1(p_t0(i));
        end
    cv_block = [xp1_0_cv_block xp2_0_cv_block ];
 %prob block
        pp1_block = zeros(st,1);
        pp2_block = zeros(st,1);
        for i =  1 : st
            pp1_block(i) = pp_no_1(p_t0(i));
            pp2_block(i) = pp_no_2(p_t0(i));
        end
        pp_block = [pp1_block pp2_block];
        %calculate ratio between x1 and x2
        xp1_xp2 = zeros(time,1);
        for i = 1 : time
            xp1_xp2(i) = xp1(i)/xp2(i);
        end
        %calculate return with CV
        case2_RE = zeros(time,1);
        case21_RE = zeros(time,1);
        num_trade1 = sum(trade1);
        num_trade2 = sum(trade2);
   for i = 1:time
      if trade1(i) == 1
         if trade2(i) == 1
         %long x1, short x2
```

```
          if diff(i) >= 0.1*min(xp1(i),xp2(i))
               if xp1(i) < xp2(i)
               %long x1, short x2
                 case2_RE(i) =
                 lreturn_xp1(i)*w1(i)*pp_no_1(i)-lreturn_xp2(i)*w2(i)*pp_no_2(i) +
                 tc;
                 case21_RE(i) =
                 lreturn_xp1(i)*w1(i)-lreturn_xp2(i)*w2(i) + tc;
               elseif xp1(i) > xp2(i)
               %long x2, short x1
                  case2_RE(i) =
                 -lreturn_xp1(i)*w1(i)*pp_no_1(i)+lreturn_xp2(i)*w2(i)*pp_no_2(i) +
                  tc;
                  case21_RE(i) =
                 -lreturn_xp1(i)*w1(i)+lreturn_xp2(i)*w2(i) + tc;
                end
            end
          end
      end
   end
     dif_case_2_21 = abs(case21_RE-case2_RE);
     sum_case2_RE_1 =sum(case2_RE);
     return_cv(i1,j1) = sum_case2_RE_1;
%return block
        re_block = zeros(st,1);
        for i = 1 : st
          tt0 = p_t0(i);
          tt1 = p_t1(i);
          for ib = tt0 : tt1
           re_block(i) = re_block(i)+
           case2_RE(ib);
          end
        end
        return_block = abs(re_block);
%calculate return using CV and xp1,xp2 %no prob.
        case3_RE = zeros(time,1);
        for i = 1:time
            if trade1(i) == 1
             if trade2(i) == 1
             %long x1, short x2
               if diff(i) >= 0.1*min(xp1(i),xp2(i))
                 if xp1(i) < xp2(i)
                   %long x1, short x2
                     case3_RE(i) =
                     lreturn_xp1(i)*w1(i) - lreturn_xp2(i)*w2(i) + tc;
                        elseif xp1(i) > xp2(i)
                         %long x2, short x1
                           case3_RE(i) =
                           -lreturn_xp1(i)*w1(i) +lreturn_xp2(i)*w2(i) +
                           tc;
                        else case3_RE(i) = 0;
                        end
                    else case3_RE(i) = 0;
                    end
                else case3_RE(i) = 0;
                end
            else case3_RE(i) =0;
            end
        end
        sum_case3_RE_1 =sum(case3_RE);
        return_cv1(i1,j1) = sum_case3_RE_1;
%return block
        re1_block = zeros(st,1);
        for i = 1 : st
            tt0 = p_t0(i);
            tt1 = p_t1(i);
            for ib = tt0 : tt1
                re1_block(i) = re1_block(i)+case21_RE(ib);
            end
        end
        return1_block = abs(re1_block);
    end
end
%take absolute
abs_r_diff = abs(return_diff);
abs_r_cv = abs(return_cv);
abs_r_cv1 = abs(return_cv1);
```

R script

script1 : correlation coefficient of stocks, plot the actual, the normalized, and the ration of the highest correlation coefficient paired stocks prices.

```
#Correlation Code
rm(list=ls())
library(kernlab)

#Read data
sh1 <- as.data.frame(read.table("sh1.txt", header=TRUE))
sh2 <- as.data.frame(read.table("sh2.txt", header=TRUE))
sh3 <- as.data.frame(read.table("sh3.txt", header=TRUE))
sh4 <- as.data.frame(read.table("sh4.txt", header=TRUE))
data <- cbind.data.frame(sh1, sh2, sh3, sh4)

#Remove data that have NA more tha 2/3 of data
limit <- 2*nrow(data)/3
data <- data[, which(as.numeric(colSums(!is.na(data)))> limit)]

#Remove all row of NA data
data <- na.omit(data)

# set date of data
Date <- data$Date
#remove Date column
data$Date <- NULL
```

```
cor.out <- cor(normal.data)
write.table(cor.out,”cor.out.txt”)

#function for finding the highest correlation
mosthighlycorrelated <- function(mydataframe, numtoreport)
{
  # find the correlations
  cormatrix <- cor(mydataframe)
  # set the correlations on the diagonal or
  # lower triangle
  # to zero,
  # so they will not be reported as the
  #highest ones:
  diag(cormatrix) <- 0
  cormatrix[lower.tri(cormatrix)] <- 0
  # flatten the matrix into a dataframe for
  #easy sorting
  fm <- as.data.frame(as.table(cormatrix))
  # assign human-friendly names
  names(fm) <- c(”First.Variable”,
  ”Second.Variable”,”Correlation”)
  # sort and print the top n correlations
  head(fm[order(abs(fm$Correlation),
  decreasing=T),],
  n=numtoreport)
}
#code for finding
top100.out <- mosthighlycorrelated(normal.data, 100)
#write file
write.table(top100.out,”top100.out.txt”)
#plot actual data
#save plot
pdf('C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/draft_thesisBook/d_TB_1_PT_2014/
X83X84plotActual.pdf')
plot(data$X8306JP, type = 'l', col = 'blue', ylim = c(80,2000))
lines(data$X8411JP, type = 'l', col = 'red')
dev.off()
# calculate return
n <- length(data)
#lrest <- log(prices[-1]/prices[-n])
require(quantmod)
#Delt(a)
lrets.X8306JP <- Delt(data$X8306JP)
lrets.X8411JP <- Delt(data$X8411JP)
#plot return
#save plot
pdf('C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/draft_thesisBook/d_TB_1_PT_2014/
X83X84plotReturns.pdf')
plot(lrets.X8306JP, type = 'l', col = 'blue')
lines(lrets.X8411JP, type = 'l', col = 'red')
legend(”topleft”, legend=c(”X8306JP”, ”X8411JP”),
       col= c(”blue”, ”red”), lty=1:2, cex=0.8)
```

```
# add a title and subtitle
title(”Returns”)
dev.off()

#write actual Paired stock data
pair.actual <- cbind(data$X8306JP,data$X8411JP)
colnames(pair.actual) <- c(”X8306JP”, ”X8411JP”)
write.table(pair.actual,”X8384.actual.txt”)

# Norlmalized data
library(clusterSim)
normal.X8306JP <- data.Normalization(data$X8306JP,type=”n1”,normalization=”column”)
normal.X8411JP <- data.Normalization(data$X8411JP,type=”n1”,normalization=”column”)

#Plot normalized
#save plot
pdf('C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/draft_thesisBook/d_TB_1_PT_2014/
X83X84normal.pdf')
plot(normal.X8306JP, type = ”l”, col = ”blue”)
lines(normal.X8411JP, col=”red”)
legend(”topleft”, legend=c(”X8306JP”, ”X8411JP”),
       col= c(”blue”, ”red”), lty=1:2, cex=0.8)
title(”Normalized␣data”)
dev.off()

#Plot Ratio
#save plot
pdf('C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/draft_thesisBook/d_TB_1_PT_2014/
X83X84ratio.pdf')
plot(data$X8306JP/data$X8411JP, type = ”l”)
legend(”topleft”, legend=c(”X8306JP/X8411JP”),
       col= c(”black”), lty=1:2, cex=0.8)
title(”ratio␣ofX8306JP␣and␣X8411JP”)
dev.off()
```

script2: simulation of ARIMA, MCMC, and SVR models for X8306JP and X8411JP with 70-30 ratio

```
rm(list=ls())

#Data Section
#++++++++++++++++++++++++++++++++++++++++++++++

#Read data
sh1 <- read.table(”sh1.txt”, header=TRUE)
sh2 <- read.table(”sh2.txt”, header=TRUE)
sh3 <- read.table(”sh3.txt”, header=TRUE)
sh4 <- read.table(”sh4.txt”, header=TRUE)
data <- cbind(sh1, sh2, sh3, sh4)
data <- data[-1]

#Remove data that have NA more tha 2/3 of data
limit <- 2*nrow(data)/3
data <- data[, which(as.numeric(colSums(
!is.na(data))) > limit)]
```

```
#Remove all row of NA data
data <- na.omit(data)
data <- as.matrix(sapply(data, as.numeric))
data <- as.data.frame(data)
index <- 1 : ceiling(length(data[,1])*.7)
data.train <- data[index, ]
data.test <- data[-index, ]

#Variables Selection
#++++++++++++++++++++++++++++++++++++++++++
#X8306JP
X8306JP.model <- lm(X8306JP~., data.train)
summary(X8306JP.model)
X8306JP.variable <- c(#"MMMUS","ABBSS",
"ABTUS", "AAUS", "AXPUS", "AMGN", #"AALLN",
"ABIBB",
#"GIM",
"TUS", "BA.LN", "BBVASM", "BACUS", "BKUS",
"BASGR", "BAXUS",
"BHARTIIN", "BHPAU", "BP.LN", "X5108JP", "CVXUS",
"X941HK", "SGOFP",
"CMIG4", "COPUS", "CSGNVX", "DEUS", "DBKGR",
"DDUS", "EOANGR", "EBAYUS",
"EDPPL", "X330HK", "FDXUS", "FCXUS", "GEUS",
"GILDUS", "GOOGUS", "HPQUS",
"HSBALN", "X13HK", "INTCUS", "IBMUS", "JNJUS",
"JPMUS", #"X6301JP",
"X066570KS", "MCFP", "X8411JP", "NDAQUS", "NABAU",
"NG.LN", "NWSAUS",
"NKEUS", "X5401JP", "PFEUS", "POTCN", "PGUS",
"RILIN", "BBCN", "ROSW",
"RYCN", "X005930KS", "SLBUS", "SIEGR", "GLEFP",
"X6758JP", "LUVUS",
"X4502JP", "TEFSM", "X6502JP", "X7203JP", "UCGIM",
"UPSUS", "UTXUS",
"VALE5BZ", "VIEFP", "VWSDC", "VODLN", "X8306JP")
X8306JP.data <- data.train[ , (names(data.train)
%in% X8306JP.variable)]

#X8411JP
X8411JP.model <- lm(X8411JP~., data.train)
summary(X8411JP.model)
X8411JP.variable <- c('MMMUS", "ABBSS", "ABTUS",
"AAUS", #5"ALVGR",
"AMXLMM", "AMGN",
"AALLN",                #9"ABIBB",
"TUS", "BACUS", "BKUS", #"BASGR",
"BHARTIIN",
"BHPAU", "BNPFP", "BAUS", "BP.LN", "X5108JP",
"X7751JT","CVXUS", #"X941HK",
"CSCOUS", "CLUS", "SGOFP", "COPUS",
"CSGNVX", "DAIGR", "DEUS", "DDUS", "EBAYUS",
”X330HK” ,”FDXUS” , ”GSKUS” , ”GOOGUS” , ”HPQUS” ,
”INTCUS” , ”IBMUS” , ”JNJUS” , #”JPMUS” ,
”X6301JP” , ”X066570KS” , ”X8306JP” , #”MONUS” ,
”NABAU” , ”NG.LN” , ”NWSAUS” , ”X7974JP” , ”X5401JP” ,
”X6752JP” , ”PETR4BZ” , ”PFEUS” , #”BBCN” ,
”ROSW” , ”RYCN” , ”SLBUS” , ”SIEGR” , ”GLEFP” ,
”X6758JP” , ”LUVUS” , ”TEFSM” , ”TSCOLN” , ”TWXUS” ,
”X6502JP” , ”FPFP” , #”X7203JP” ,
”UBSNVX” , ”UTXUS” , ”VALE5BZ” , ”VIEFP” , ”VZUS” ,
”VWSDC” , ”VODLN” , ”WMTUS” , ”WFCUS” , ”X8411JP”)
X8411JP.data <- data.train[ , (names(data.train)
                                        %in% X8411JP.variable)]

#SVR Section
#++++++++++++++++++++++++++++++++++++++++++
library(kernlab)

#X8306JP
svr.X8306JP.rbfdot <- ksvm(X8306JP~.,X8306JP.data,
kernel = ”rbfdot”)
svr.X8306JP.rbfdot.error <- svr.X8306JP.rbfdot@error
svr.X8306JP.polydot <- ksvm(X8306JP~.,X8306JP.data,
kernel = ”polydot”)
svr.X8306JP.polydot.error <- svr.X8306JP.polydot@error
svr.X8306JP.vanilladot <- ksvm(X8306JP~., X8306JP.data,
kernel = ”vanilladot”)
svr.X8306JP.vanilladot.error <- svr.X8306JP.vanilladot@error
svr.X8306JP.tanhdot <- ksvm(X8306JP~., X8306JP.data,
kernel = ”tanhdot”)
svr.X8306JP.tanhdot.error <- svr.X8306JP.tanhdot@error
svr.X8306JP.laplacedot <- ksvm(X8306JP~., X8306JP.data,
kernel = ”laplacedot”)
svr.X8306JP.laplacedot.error <- svr.X8306JP.laplacedot@error
svr.X8306JP.besseldot <- ksvm(X8306JP~., X8306JP.data,
kernel = ”besseldot”)
svr.X8306JP.besseldot.error <- svr.X8306JP.besseldot@error
svr.train.error <- cbind(svr.X8306JP.rbfdot.error, svr.X8306JP.polydot.error,
svr.X8306JP.polydot.error, svr.X8306JP.vanilladot.error,
svr.X8306JP.tanhdot.error, svr.X8306JP.laplacedot.error,
svr.X8306JP.besseldot.error)
svr.train.error.min <- min(svr.train.error)
if(svr.train.error.min == svr.X8306JP.rbfdot.error)
{
  svr.X8306JP.predict <-    predict(svr.X8306JP.rbfdot, data.test)
}else if (svr.train.error.min == svr.X8306JP.polydot.error) {
  svr.X8306JP.predict <-    predict(svr.X8306JP.polydot, data.test)
}else if (svr.train.error.min == svr.X8306JP.vanilladot.error) {
  svr.X8306JP.predict <-    predict(svr.X8306JP.vanilladot, data.test)
}else if (svr.train.error.min == svr.X8306JP.tanhdot.error) {
  svr.X8306JP.predict <-    predict(svr.X8306JP.tanhdot, data.test)
}else if (svr.train.error.min == svr.X8306JP.laplacedot.error) {
  svr.X8306JP.predict <-    predict(svr.X8306JP.laplacedot, data.test)
}else if (svr.train.error.min == svr.X8306JP.besseldot.error) {
  svr.X8306JP.predict <-   predict(svr.X8306JP.besseldot, data.test)
}
ntest <- length(data.test$X8306JP)
mae.svr.X8306JP <- sum(abs((data.test$X8306JP - svr.X8306JP.predict)/data.test$X8306JP))/ntest
mape.svr.X8306JP <- mae.svr.X8306JP*100
mse.svr.X8306JP <- sum((svr.X8306JP.predict - data.test$X8306JP)^2)/ntest
rmse.svr.X8306JP <- sqrt(mse.svr.X8306JP)
error.svr.X8306JP <- cbind(mae.svr.X8306JP, mape.svr.X8306JP,
mse.svr.X8306JP, rmse.svr.X8306JP)
error.svr.X8306JP

#X8411JP
svr.X8411JP.rbfdot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”rbfdot”)
svr.X8411JP.rbfdot.error <- svr.X8411JP.rbfdot@error
svr.X8411JP.polydot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”polydot”)
svr.X8411JP.polydot.error <- svr.X8411JP.polydot@error
svr.X8411JP.vanilladot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”vanilladot”)
svr.X8411JP.vanilladot.error <- svr.X8411JP.vanilladot@error
svr.X8411JP.tanhdot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”tanhdot”)
svr.X8411JP.tanhdot.error <- svr.X8411JP.tanhdot@error
svr.X8411JP.laplacedot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”laplacedot”)
svr.X8411JP.laplacedot.error <- svr.X8411JP.laplacedot@error
svr.X8411JP.besseldot <- ksvm(X8411JP~., X8411JP.data,
kernel = ”besseldot”)
svr.X8411JP.besseldot.error <- svr.X8411JP.besseldot@error
svr.train.error <- cbind(svr.X8411JP.rbfdot.error,
svr.X8411JP.polydot.error, svr.X8411JP.polydot.error,
svr.X8411JP.vanilladot.error,svr.X8411JP.tanhdot.error,
svr.X8411JP.laplacedot.error, svr.X8411JP.besseldot.error)
svr.train.error.min <- min(svr.train.error)
if(svr.train.error.min == svr.X8411JP.rbfdot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.rbfdot,   data.test)
}else if (svr.train.error.min == svr.X8411JP.polydot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.polydot,   data.test)
}else if (svr.train.error.min == svr.X8411JP.vanilladot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.vanilladot,   data.test)
}else if (svr.train.error.min == svr.X8411JP.tanhdot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.tanhdot,   data.test)
}else if (svr.train.error.min == svr.X8411JP.laplacedot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.laplacedot,   data.test)
}else if (svr.train.error.min == svr.X8411JP.besseldot.error)
{
  svr.X8411JP.predict <-   predict(svr.X8411JP.besseldot, data.test)
```

```
}

ntest <- length(data.test$X8411JP)
mae.svr.X8411JP <- sum(abs((data.test$X8411JP - svr.X8411JP.predict)/data.test$X8411JP))/ntest
mape.svr.X8411JP <- mae.svr.X8411JP*100
mse.svr.X8411JP <- sum((svr.X8411JP.predict - data.test$X8411JP)^2)/ntest
rmse.svr.X8411JP <- sqrt(mse.svr.X8411JP)
error.svr.X8411JP <- cbind(mae.svr.X8411JP,mape.svr.X8411JP,
mse.svr.X8411JP, rmse.svr.X8411JP)
error.svr.X8411JP

#ARIMA section
#++++++++++++++++++++++++++++++++++++++++

#X8306JP
arima.X8306JP.data <- ts(data.train$X8306JP)
arima.X8306JP.model <- arima(arima.X8306JP.data, order = c(1,0,0))
arima.X8306JP.predict <- (predict(arima.X8306JP.model, n.ahead = ntest))$pred
mae.arima.X8306JP <- sum(abs((data.test$X8306JP -
arima.X8306JP.predict)/data.test$X8306JP))/ntest
mape.arima.X8306JP <- mae.arima.X8306JP*100
mse.arima.X8306JP <- sum((arima.X8306JP.predict -
data.test$X8306JP)^2)/ntest
rmse.arima.X8306JP <- sqrt(mse.svr.X8306JP)
error.arima.X8306JP <- cbind(mae.arima.X8306JP,
mape.arima.X8306JP, mse.arima.X8306JP,
rmse.arima.X8306JP)error.arima.X8306JP
#X8411JP
arima.X8411JP.data <- ts(data.train$X8411JP)
arima.X8411JP.model <- arima(arima.X8411JP.data, order = c(1,0,0))
arima.X8411JP.predict <- (predict(
arima.X8411JP.model, n.ahead = ntest))$pred
mae.arima.X8411JP <- sum(abs((data.test$X8411JP -
arima.X8411JP.predict)/data.test$X8411JP))/ntest
mape.arima.X8411JP <- mae.arima.X8411JP*100
mse.arima.X8411JP <- sum((arima.X8411JP.predict -
data.test$X8411JP)^2)/ntest
rmse.arima.X8411JP <- sqrt(mse.svr.X8411JP)
error.arima.X8411JP <- cbind(mae.arima.X8411JP,
mape.arima.X8411JP, mse.arima.X8411JP,
rmse.arima.X8411JP)error.arima.X8411JP
#MCMC section
#++++++++++++++++++++++++++++++++++++++++
library(MCMCpack)
#X8306JP
mcmc.X8306JP.model <- MCMCregress( X8306JP~X8411JP, data = data.train)
mcmc.X8306JP.summary <- summary(mcmc.X8306JP.model)
mcmc.X8306JP.intercept <- mcmc.X8306JP.summary$statistics[1]
mcmc.X8306JP.coef <- mcmc.X8306JP.summary$statistics[2]
mcmc.X8306JP.predict <- (data.test$X8411JP *
mcmc.X8306JP.coef)+ mcmc.X8306JP.intercept
mae.mcmc.X8306JP <- sum(abs((data.test$X8306JP -
mcmc.X8306JP.predict)/data.test$X8306JP))/ntest
```

```
mape.mcmc.X8306JP <− mae.mcmc.X8306JP*100
mse.mcmc.X8306JP <− sum((mcmc.X8306JP.predict −
data.test$X8306JP)^2)/ntest
rmse.mcmc.X8306JP <− sqrt(mse.mcmc.X8306JP)
error.mcmc.X8306JP <− cbind(mae.mcmc.X8306JP,
mape.mcmc.X8306JP, mse.mcmc.X8306JP,
rmse.mcmc.X8306JP)error.mcmc.X8306JP
#X8411JP
mcmc.X8411JP.model <−
MCMCregress( X8411JP~X8306JP, data = data.train)
mcmc.X8411JP.summary <− summary(mcmc.X8411JP.model)
mcmc.X8411JP.intercept <− mcmc.X8411JP.summary$statistics[1]
mcmc.X8411JP.coef <− mcmc.X8411JP.summary$statistics[2]
mcmc.X8411JP.predict <− (data.test$X8306JP * mcmc.X8411JP.coef)
+ mcmc.X8411JP.intercept
mae.mcmc.X8411JP <− sum(abs((data.test$X8411JP −
mcmc.X8411JP.predict)/data.test$X8411JP))/ntest
mape.mcmc.X8411JP <− mae.mcmc.X8411JP*100
mse.mcmc.X8411JP <− sum((mcmc.X8411JP.predict −
data.test$X8411JP)^2)/ntest
rmse.mcmc.X8411JP <− sqrt(mse.mcmc.X8411JP)
error.mcmc.X8411JP <− cbind(mae.mcmc.X8411JP,
mape.mcmc.X8411JP, mse.mcmc.X8411JP,
rmse.mcmc.X8411JP)
error.mcmc.X8411JP

#Summary
X8306JP.summary.data <− as.data.frame(cbind(data.test$X8306JP,
svr.X8306JP.predict, arima.X8306JP.predict,
mcmc.X8306JP.predict))
colnames(X8306JP.summary.data) <− c(”Original”, ”SVR”, ”ARIMA”, ”MCMC”)
X8411JP.summary.data <− as.data.frame(cbind(data.test$X8411JP,
svr.X8411JP.predict, arima.X8411JP.predict,
mcmc.X8411JP.predict))
colnames(X8411JP.summary.data) <− c(”Original”, ”SVR”, ”ARIMA”, ”MCMC”)
#Plot X8306JP#save plot
pdf(’C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/
draft_thesisBook/d_TB_1_PT_2014/X8306JPplot7030.pdf’)
plot(X8306JP.summary.data$Original, type = ”l”,
ylim = c(300,800),
xlab = ”time(Date)”, ylab = ”Stock␣price”)
lines(X8306JP.summary.data$ARIMA, col=”red”)
lines(X8306JP.summary.data$MCMC, col=”green”)
lines(X8306JP.summary.data$SVR, col=”blue”)
legend(”topleft”, legend=c(”actual␣X8306JP”,
”ARIMA”, ”MCMC”, ”SVR”),
col= c(”black”, ”red”, ”green”, ”blue”), lty=1:2, cex=0.8)
# add a title and subtitle
title(”Simulation␣results␣:␣ARIMA,␣MCMC,␣and␣SVR”, ”for␣X8306JP”)
dev.off()
#Plot X8411JP
#save plot
pdf(’C:/Users/N.␣WowoW␣Ekkarntrong/Dropbox/Apps/Texpad/
draft_thesisBook/d_TB_1_PT_2014/X8411JPplot7030.pdf')
plot(X8411JP.summary.data$Original, type = "l", ylim = c(80,300),
xlab = "time(Date)", ylab = "Stock␣price")
lines(X8411JP.summary.data$ARIMA, col="red")
lines(X8411JP.summary.data$MCMC, col="green")
lines(X8411JP.summary.data$SVR, col="blue")
legend("topleft", legend=c("actual␣X8411JP", "ARIMA", 'MCMC", "SVR"),
col= c("black", "red", "green", "blue"), lty=1:2, cex=0.8)
# add a title and subtitle
title("Simulation␣results␣:␣ARIMA,␣MCMC,␣and␣SVR","for␣X8411JP")
dev.off()
#considering trend section
#++++++++++++++++++++++++++++++++++++++++++++++++
# set number of data
n <- nrow(data.test)-1
X8306JP.actual <- data.test$X8306JP

# X8306JP
# lag for svr actual && predicted
lag.X8306JP.actual <- diff(X8306JP.actual)
lag.svr.X8306JP <- diff(svr.X8306JP.predict)
#set count vector for count a right direction ;
#initial value
svr.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8306JP.actual[i] >= 0 &&   lag.svr.X8306JP[i] >= 0){
    svr.count.direction[i] <- 1
  } else if (lag.X8306JP.actual[i] < 0 &&   lag.svr.X8306JP[i] < 0){
    svr.count.direction[i] <- 1
  } else
    svr.count.direction[i] <- 0
}

# lag for mcmc predicted
lag.mcmc.X8306JP <- diff(mcmc.X8306JP.predict)
#set count vector for count a right direction ;
#initial value
mcmc.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8306JP.actual[i] >= 0 &&   lag.mcmc.X8306JP[i] >= 0){
    mcmc.count.direction[i] <- 1
  } else if (lag.X8306JP.actual[i] < 0 &&   lag.mcmc.X8306JP[i] < 0){
    mcmc.count.direction[i] <- 1
  } else
    mcmc.count.direction[i] <- 0
}
# lag for arima predicted
lag.arima.X8306JP <- diff(arima.X8306JP.predict)
#set count vector for count a right direction ;
```

```
#initial value
arima.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8306JP.actual[i] >= 0 &&   lag.arima.X8306JP[i] >= 0){
    arima.count.direction[i] <- 1
  } else if (lag.X8306JP.actual[i] < 0 &&   lag.arima.X8306JP[i] < 0){
    arima.count.direction[i] <- 1
  } else
    arima.count.direction[i] <- 0
}
right.direction.X8306JP <- cbind(sum(arima.count.direction),
sum(mcmc.count.direction), sum(svr.count.direction))
percent.direction.X8306JP <- right.direction.X8306JP/(n-1)*100

# X8411JP
X8411JP.actual <- data.test$X8411JP
# lag for svr actual && predicted
lag.X8411JP.actual <- diff(X8411JP.actual)
lag.svr.X8411JP <- diff(svr.X8411JP.predict)
#set count vector for count a right direction ;
#initial value
svr.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8411JP.actual[i] >= 0 &&   lag.svr.X8411JP[i] >= 0){
    svr.count.direction[i] <- 1
  } else if (lag.X8411JP.actual[i] < 0 &&   lag.svr.X8411JP[i] < 0){
    svr.count.direction[i] <- 1
  } else
    svr.count.direction[i] <- 0
}
# lag for mcmc predicted
lag.mcmc.X8411JP <- diff(mcmc.X8411JP.predict)
#set count vector for count a right direction ;
#initial value
mcmc.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8411JP.actual[i] >= 0 &&   lag.mcmc.X8411JP[i] >= 0){
    mcmc.count.direction[i] <- 1
  } else if (lag.X8411JP.actual[i] < 0 &&   lag.mcmc.X8411JP[i] < 0){
    mcmc.count.direction[i] <- 1
  } else
    mcmc.count.direction[i] <- 0
}
# lag for arima predicted
lag.arima.X8411JP <- diff(arima.X8411JP.predict)
#set count vector for count a right direction ;
#initial value
```

```
arima.count.direction <- matrix(0,n-1,1)
#for loop
for (i in 1 : n)
{
  if (lag.X8411JP.actual[i] >= 0 &&  lag.arima.X8411JP[i] >= 0){
    arima.count.direction[i] <- 1
  } else if (lag.X8411JP.actual[i] < 0 &&  lag.arima.X8411JP[i] < 0){
    arima.count.direction[i] <- 1
  } else
    arima.count.direction[i] <- 0
}

right.direction.X8411JP <- cbind(sum(arima.count.direction),
sum(mcmc.count.direction), sum(svr.count.direction))
percent.direction.X8411JP <- right.direction.X8411JP/(n-1)*100
#++++++++++++++++++++++++++++++++++++++++++++++
library(AICcmodavg)
library(MuMIn)
#+++++++++++++++++++++++++++++++++++++++++++++++
#SVR
#X8306JP
a.svr.X8306JP <- AIC(lm(X8306JP~., data.train))
b.svr.X8306JP <- BIC(lm(X8306JP~., data.train))
r.svr.X8306JP <-
summary(lm(X8306JP~., data.train))$r.squared
info.svr.X8306JP <- cbind(a.svr.X8306JP,
b.svr.X8306JP, r.svr.X8306JP)

#X8411JP
a.svr.X8411JP <-  AIC(lm(X8411JP~., data.train))
b.svr.X8411JP <- BIC(lm(X8411JP~., data.train))
r.svr.X8411JP <-
summary(lm(X8411JP~., data.train))$r.squared
info.svr.X8411JP <- cbind(a.svr.X8411JP,
b.svr.X8411JP, r.svr.X8411JP)
#ARIMA
#X8306JP
a.arima.X8306JP <- AIC(arima.X8306JP.model)
#ac.arima.X8306JP <- AICc(arima.X8306JP.model)
b.arima.X8306JP <- BIC(arima.X8306JP.model)
r.arima.X8306JP <- 0
info.arima.X8306JP <- cbind(a.arima.X8306JP,
b.arima.X8306JP, r.arima.X8306JP)
#X8411JP
a.arima.X8411JP <- AIC(arima.X8411JP.model)
#ac.arima.X8411JP <- AICc(arima.X8411JP.model)
b.arima.X8411JP <- BIC(arima.X8411JP.model)
r.arima.X8411JP <- 0
info.arima.X8411JP <- cbind(a.arima.X8411JP,
b.arima.X8411JP, r.arima.X8411JP)
#MCMC
#X8306JP
a.mcmc.X8306JP <- AIC(lm(X8306JP~X8411JP,
data.train))
ac.mcmc.X8306JP <- AICc(lm(X8306JP~X8411JP,
data.train))
b.mcmc.X8306JP <- BIC(lm(X8306JP~X8411JP,
data.train))
r.mcmc.X8306JP <-
summary(lm(X8306JP~X8411JP, data.train))$r.squared
info.mcmc.X8306JP <- cbind(a.mcmc.X8306JP,
ac.mcmc.X8306JP,
b.mcmc.X8306JP, r.mcmc.X8306JP)
#X8411JP
a.mcmc.X8411JP <- AIC(lm(X8411JP~X8306JP,
data.train))
ac.mcmc.X8411JP <- AICc(lm(X8411JP~X8306JP,
data.train))
b.mcmc.X8411JP <- BIC(lm(X8411JP~X8306JP,
data.train))
r.mcmc.X8411JP <-
summary(lm(X8411JP~X8306JP, data.train))$r.squared
info.mcmc.X8411JP <- cbind(a.mcmc.X8411JP,
ac.mcmc.X8411JP,
b.mcmc.X8411JP, r.mcmc.X8411JP)
#+++++++++++++++++++++++++++++++++++++++++++
## Normality tests
# The statement performing Shapiro-Wilk test
# is shapiro.test()and
# it supplies W statistic and the pvalue:
shapiro.test(data$X8306JP)
shapiro.test(data$X8411JP)
library(tseries) ## package tseries loading
jarque.bera.test(data$X8306JP)
library(nortest) ## package loading
# performs Shapiro-Francia test
sf.test(data$X8306JP)
# performs Anderson-Darling test
ad.test(data$X8306JP)
adf.test(data$X8306JP)
# performs Lilliefors test
lillie.test(data$X8306JP)
# performs Pearson's chi-square test
pearson.test(data$X8306JP)
library(fUnitRoots)
jarque.bera.test(data$X8411JP)
# performs Shapiro-Francia test
sf.test(data$X8411JP)
# performs Anderson-Darling test
ad.test(data$X8411JP)
adf.test(data$X8411JP)
# performs Lilliefors test
lillie.test(data$X8411JP)
# performs Pearson's chi-square test
pearson.test(data$X8411JP)
```

CURRICULUM VITAE

NAME : Nawarat Ekkarntrong **GENDER :** Female

DATE OF BIRTH : December 3, 1984 **NATIONALITY :** Thai

EDUCATION BACKGROUND :

- Bachelor of Science in Mathematics (First Class Honors), Khonkaen University, Thailand, 2006

- Master of Science in Computational Science , Chulalongkorn University, Thailand, 2009

SCHOLARSHIP :

- Development and Promotion of Science and Technology Talents Project (DPST), 2000-present

CONFERENCE :

- A COMPARISON TEST OF BINOMIAL TREE MODELS FOR SET50 INDEX OPTIONS, **The** 35^{th} **Congress on Science and Technology of Thailand (STT35)**, 15-19 October, 2010, The Tide Resort(Bangsaen Beach), Chonburi Thailand, Thailand

- PAIRS TRADING MODEL USING MEAN REVERSION, (The 41st Congress on Science and Technology of Thailand)

PUBLICATION :

- A COMPARISON TEST OF BINOMIAL TREE MODELS FOR SET50 INDEX OPTIONS, **Proceedings of The** 35^{th} **Congress on Science and Technology of Thailand (STT35)**