การเปรียบเทียบความแม่นยำของการพยากรณ์ด้วยตัวแบบอนุกรมเวลาแบบผสม

นายชญานิน บุญมานะ

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาสถิติ ภาควิชาสถิติ
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A COMPARATIVE PREDICTION ACCURACY OF HYBRID TIME SERIES MODELS

Mr. Chayanin Boonmana

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Statistics
Department of Statistics
Faculty of Commerce and Accountancy
Chulalongkorn University
Academic Year 2015
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การเปรียบเทียบความแม่นยำของการพยากรณ์ด้วยตัวแบบอนุกรมเวลาแบบผสม
โดย	นายชญานิน บุญมานะ
สาขาวิชา	สถิติ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร. นัท กุลวานิช

คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

....คณบดีคณะพาณิชยศาสตร์และการบัญชี

(รองศาสตราจารย์ ดร. พสุ เดชะรินทร์)

คณะกรรมการสอบวิทยานิพนธ์

....ประธานกรรมการ

(รองศาสตราจารย์ ดร. สุพล ดุรงค์วัฒนา)

....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(อาจารย์ ดร. นัท กุลวานิช)

....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. วิฐรา พึ่งพาพงศ์)

....กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร. อรุณี กำลัง)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ชญานิน บุญมานะ : การเปรียบเทียบความแม่นยำของการพยากรณ์ด้วยตัวแบบอนุกรมเวลาแบบผสม (A COMPARATIVE PREDICTION ACCURACY OF HYBRID TIME SERIES MODELS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: อ. ดร. นัท กุลวานิช, 175 หน้า.

การวิจัยนี้ศึกษาเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) โดยใช้ชุดข้อมูลจริงและชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ โดยตัวแบบใดที่มีค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ดีที่สุด จากการศึกษาพบว่า ตัวแบบผสมระหว่าง ARIMA และซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาในกรณีที่จำลองด้วยตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,1) และตัวแบบ ARIMA(2,1,2) ส่วนตัวแบบผสมระหว่าง ARIMA และเครือข่ายประสาทเทียมนั้นให้ความแม่นยำในการพยากรณ์สูงที่สุดสำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาในกรณีที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0) และตัวแบบ ARIMA(2,1,1) และสำหรับการพยากรณ์ในชุดข้อมูลจริงของราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ที่มีลักษณะอนุกรมเวลาสอดคล้องกับตัวแบบ ARIMA(1,1,1) ผลการศึกษาพบว่าตัวแบบผสมระหว่าง ARIMA และซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุดซึ่งสอดคล้องกับผลจากชุดข้อมูลจำลอง

ภาควิชา สถิติ ลายมือชื่อนิสิต ....

สาขาวิชา สถิติ ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

ปีการศึกษา 2558

5781522626 : MAJOR STATISTICS
KEYWORDS: ARIMA MODEL / ARTIFICIAL NEURAL NETWORK MODEL / SUPPORT VECTOR MACHINE MODEL / HYBRID MODEL

CHAYANIN BOONMANA: A COMPARATIVE PREDICTION ACCURACY OF HYBRID TIME SERIES MODELS. ADVISOR: NAT KULVANICH, Ph.D., 175 pp.

This research is aimed to compare the prediction accuracy between three time series models, traditional ARIMA model, a hybrid model combing ARIMA model and Artificial neural network model, and a hybrid model combing ARIMA model and Support vector machine model by using real stock price datasets of the Siam commercial bank company and time series datasets simulated from ARIMA(0,1,1), ARIMA(0,1,2), ARIMA(1,1,0), ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(2,1,0), ARIMA(2,1,1) and ARIMA(2,1,2) model. Root mean square error (RMSE) is used to compare the prediction accuracy from each model. The model which has the lowest RMSE is the best model. The results suggest that a hybrid model combing ARIMA model and Support vector machine model has the highest prediction accuracy for the case of ARIMA(0,1,2), ARIMA(1,1,1) and ARIMA(2,1,2) model. However, the prediction accuracy of a hybrid model combing ARIMA model and Artificial neural network model is found to be highest in the case of ARIMA(0,1,1), ARIMA(1,1,0), ARIMA(1,1,2), ARIMA(2,1,0) and ARIMA(2,1,1) model.

A similar three forecasting models was developed and evaluated for the weekly closing price of stock price of the Siam commercial bank company. The results suggest that a hybrid model combing ARIMA(1,1,1) model and Support vector machine model has the highest prediction accuracy. We observed concordant results on real and simulated data.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department:	Statistics	Student's Signature	....
Field of Study:	Statistics	Advisor's Signature	....
Academic Year:	2015		

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความอนุเคราะห์ ความช่วยเหลือ และความเอาใจใส่ของอาจารย์ ดร. นัท กุลวานิช อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้วิจัยขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูงที่ได้ให้คำปรึกษา ชี้นำแนวทางในการทำวิจัย ให้คำแนะนำเพื่อปรับปรุงแก้ไขวิทยานิพนธ์ ให้ข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการทำวิจัย อีกทั้งยังคอยช่วยแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการดำเนินงาน และเป็นกำลังใจในการทำงานให้กับผู้วิจัยด้วยดีเสมอมา

ผู้วิจัยขอกราบขอบพระคุณ รองศาสตราจารย์ ดร. สุพล ดุรงค์วัฒนา ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. วิฐรา พึ่งพาพงศ์ และอาจารย์ ดร. อรุณี กำลัง กรรมการสอบวิทยานิพนธ์ เป็นอย่างสูงที่ท่านอาจารย์ทั้งสามท่านได้เสียสละเวลามาเพื่อสอบ ตรวจสอบ และให้คำแนะนำเพื่อแก้ไขวิทยานิพนธ์ฉบับนี้ให้สมบูรณ์มากยิ่งขึ้น อีกทั้งผู้วิจัยขอกราบขอบพระคุณคณาจารย์ประจำภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ได้ให้โอกาสทางการศึกษา ได้ถ่ายทอดความรู้ทั้งในด้านการเรียนและการดำรงชีวิต ประสิทธิประสาทวิชาความรู้ทางด้านสถิติให้แก่ผู้วิจัยเสมอมาจนกระทั่งสำเร็จการศึกษาในครั้งนี้ และผู้วิจัยขอกราบขอบพระคุณคณาจารย์ประจำภาควิชาคณิตศาสตร์และสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ทุกท่านที่ได้ปูพื้นฐานทางด้านคณิตศาสตร์และสถิติอย่างดีเยี่ยมให้แก่ผู้วิจัยอันเป็นประโยชน์ทั้งในด้านการศึกษาระดับปริญญาโทและในการทำวิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่สนับสนุนและเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจผู้วิจัย ซึ่งเป็นแรงผลักดันที่ดีที่สุดให้แก่ผู้วิจัยในการศึกษาเล่าเรียนจนสำเร็จการศึกษา และขอขอบคุณเพื่อน ๆ ทุกคน ที่คอยช่วยเหลือ ให้คำแนะนำ และเป็นกำลังใจให้กับผู้วิจัยตลอดมา

CHULALONGKORN UNIVERSITY

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง.....ฎ

สารบัญภาพ.....ฐ

บทที่ 1 บทนำ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย..... 4

1.3 ข้อตกลงเบื้องต้น..... 4

1.4 ขอบเขตของการวิจัย..... 4

1.4.1 ข้อมูลจริง..... 4

1.4.2 ข้อมูลจำลอง..... 6

1.5 คำจำกัดความที่ใช้ในงานวิจัย..... 7

1.6 เกณฑ์ที่ใช้ในการตัดสินใจ..... 8

1.7 วิธีการศึกษา..... 9

1.8 ประโยชน์ที่คาดว่าจะได้รับ..... 13

บทที่ 2 ทฤษฎีและตัวสถิติที่เกี่ยวข้อง..... 14

2.1 ตัวแบบ Integrated autoregressive moving average หรือ ตัวแบบ ARIMA(p,d,q)..... 14

2.1.1 ตัวแบบ Autoregressive หรือ ตัวแบบ AR(p)..... 15

2.1.2 ตัวแบบ Moving average หรือ ตัวแบบ MA(q)..... 21

2.1.3 ตัวแบบ Autoregressive moving average หรือ ตัวแบบ ARMA(p,q)..... 23

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

2.1.4 ตัวแบบ Integrated autoregressive moving average หรือ ตัวแบบ ARIMA(p,d,q) 26

2.1.5 ขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบ ARIMA 27

2.1.6 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบ ARIMA 40

2.2 ตัวแบบเครือข่ายประสาทเทียม (Artificial neural network model: ANN) 42

2.2.1 ข้อดีและข้อเสียของการใช้เครือข่ายประสาทเทียม 43

2.2.2 การจัดเตรียมข้อมูลก่อนประมวลผลข้อมูลและหลังประมาณผลข้อมูล (Data preprocessing and postprocessing) 44

2.2.3 การแปลงข้อมูลให้อยู่ในรูปปรกติมาตรฐานน้อยที่สุด-มากที่สุด (Min-Max normalization) 44

2.2.4 เทคนิคการเรียนรู้ของเครือข่ายประสาทเทียม (Artificial neural network learning) 45

2.2.4.1 เทคนิคการเรียนรู้แบบมีผู้สอน (Supervised learning) 45

2.2.4.2 เทคนิคการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) 45

2.2.5 การเชื่อมโยงของเครือข่ายประสาทเทียม (Artificial neural network linking) 46

2.2.5.1 เครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network) 46

2.2.6 เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) 49

2.2.7 กฎการแพร่แบบย้อนกลับ (Back-propagation rules) 52

2.2.8 โมเมนตัม (Momentum term) 53

2.2.9 เกณฑ์การหยุด (Termination criteria) 55

2.2.10 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบเครือข่ายประสาทเทียม 56

2.3 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีน (Support vector machine model: SVM) 58

หน้า

2.3.1 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล (Support vector machine model for classification) 58

2.3.1.1 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล กรณีที่สามารถแบ่งกลุ่มข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน 59

2.3.1.2 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล กรณีที่ไม่สามารถแบ่งกลุ่มข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน 63

2.3.1.3 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล กรณีทำการแบ่งกลุ่มข้อมูลทั้งหมดโดยใช้ระนาบแบบไม่เป็นเชิงเส้นตรง 66

2.3.2 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression) 69

2.3.3 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีน 73

2.4 ตัวแบบผสม (Hybrid model) 75

2.4.1 ขั้นตอนการสร้างตัวแบบผสม 75

บทที่ 3 วิธีการดำเนินการศึกษา 76

3.1 ขอบเขตของการวิจัย 76

3.1.1 ข้อมูลจริง 76

3.1.2 ข้อมูลจำลอง 78

3.2 ขั้นตอนในการดำเนินการศึกษา 79

3.3 แผนภาพขั้นตอนการทำงานของโปรแกรม 83

บทที่ 4 ผลการวิจัย 87

Chulalongkorn University

หน้า

4.1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง 91

4.2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง 107

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 113

5.1 สรุปผลการวิจัย 113

5.1.1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง 114

5.1.2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง 118

5.2 สรุปผลโดยรวม 120

5.3 ข้อเสนอแนะ 121

รายการอ้างอิง 123

ภาคผนวก ก 125

ภาคผนวก ข 161

ประวัติผู้เขียนวิทยานิพนธ์ 175

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

ตารางที่ หน้า

ตารางที่ 1- 1 เกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ของตัวแบบ ARIMA(p,d=1,q) 5

ตารางที่ 2- 1 ลักษณะของฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) สำหรับตัวแบบ ARMA.......... 32

ตารางที่ 3- 1 เกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ของตัวแบบ ARIMA(p,d=1,q) 77

ตารางที่ 4- 1 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1) 91

ตารางที่ 4- 2 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,2) 93

ตารางที่ 4- 3 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,0).......... 95

ตารางที่ 4- 4 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,1).......... 97

ตารางที่ 4- 5 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,2).......... 99

ตารางที่ 4- 6 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,0).......... 101

ตารางที่ 4- 7 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,1).......... 103

ตารางที่ 4- 8 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,2).......... 105

ตารางที่ 4- 9 ค่า MSE, RMSE และ MAE ที่ได้จากการพยากรณ์ราคาปิดของหุ้น SCB โดยใช้ชุดข้อมูลจริง 107

ตารางที่ หน้า

ตารางที่ 5- 1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง... 114

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญภาพ

ภาพที่ หน้า

ภาพที่ 1- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ ARIMA(1,1,1) ... 5

ภาพที่ 2- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ AR(1) เมื่อ $\phi > 0$ ... 18

ภาพที่ 2- 2 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ AR(2) เมื่อ $\phi_1 = 1.5$ และ $\phi_2 = -0.75$ ... 20

ภาพที่ 2- 3 ฟังก์ชันสหสัมพันธ์ในตัวและฟังก์ชันสหสัมพันธ์บางส่วนของตัวแบบ MA(1) ... 22

ภาพที่ 2- 4 ข้อมูล U.S. GNP ที่มีการหาผลต่างอันดับที่ 1 เพื่อแปลงอนุกรมเวลาแบบไม่คงที่ ให้เป็นอนุกรมเวลาแบบคงที่ ... 26

ภาพที่ 2- 5 กระบวนการทำงานภายในเซลล์ประสาท ... 42

ภาพที่ 2- 6 กระบวนการทำงานในเซลล์เสมือนเป็นการจำลองการทำงานภายในโหนด ... 43

ภาพที่ 2- 7 ตัวแบบเครือข่ายประสาทเทียมซึ่งมีทั้งหมด 3 ชั้น (Layer) ... 47

ภาพที่ 2- 8 การทำงานภายในโหนดที่ j ของชั้นซ่อน ... 47

ภาพที่ 2- 9 การทำงานภายในโหนดที่ k ของชั้นผลลัพธ์ ... 48

ภาพที่ 2- 10 ความสัมพันธ์ระหว่างผลบวกกำลังสองของค่าคลาดเคลื่อนกับค่าถ่วงน้ำหนัก w_1 ... 50

ภาพที่ 2- 11 การแพร่แบบย้อนกลับของค่าคลาดเคลื่อนจากชั้นผลลัพธ์มายังชั้นข้อมูลนำเข้า ... 52

ภาพที่ 2- 12 การปรับค่าถ่วงน้ำหนักเมื่อพิจารณาค่าโมเมนตัมร่วมกับอัตราการเรียนรู้ ... 54

ภาพที่ 2- 13 เส้นตรงที่แบ่งข้อมูลทั้ง 2 กลุ่มในตัวแบบซัพพอร์ทเวกเตอร์แมชชีน ... 59

ภาพที่ 2- 14 สมการที่ใช้แบ่งกลุ่มข้อมูลทั้ง 2 กลุ่มในตัวแบบซัพพอร์ทเวกเตอร์แมชชีน ... 60

ภาพที่ 2- 15 ตัวแปรที่ทำให้เกิดค่าคลาดเคลื่อนจากการแบ่งกลุ่มข้อมูล ... 63

ภาพที่ 2- 16 การแปลงข้อมูลโดยใช้ฟังก์ชันเคอร์เนลจากพื้นที่ข้อมูลนำเข้าให้ข้อมูล มีเรียงตัวใหม่ในพื้นที่คุณลักษณะ ... 67

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาพที่ หน้า

ภาพที่ 2- 17 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอยกรณีที่ไม่สามารถแยกแยะข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน เมื่อใช้ฟังก์ชันการสูญเสียไม่ไวแบบเอ็บซิลอน 69

ภาพที่ 3- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ ARIMA(1,1,1) 77

ภาพที่ 4- 1 ลักษณะเบื้องต้นของข้อมูลจริงราคาปิดหุ้น SCB 88

ภาพที่ 4- 2 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ของข้อมูลจริงราคาปิดหุ้น SCB........ 88

ภาพที่ 4- 3 ลักษณะเบื้องต้นของข้อมูลจริงราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1...... 89

ภาพที่ 4- 4 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ของข้อมูลจริงราคาปิดหุ้น SCB หลังจากการหารผลต่างอันดับที่ 1 90

ภาพที่ 4- 5 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบ ARIMA ในชุดข้อมูลฝึกสอน........ 110

ภาพที่ 4- 6 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบ ARIMA ในชุดข้อมูลทดสอบ 110

ภาพที่ 4- 7 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม ในชุดข้อมูลฝึกสอน........ 111

ภาพที่ 4- 8 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม ในชุดข้อมูลทดสอบ........ 111

ภาพที่ 4- 9 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับ ซัพพอร์ทเวกเตอร์แมชชีน ในชุดข้อมูลฝึกสอน 112

ภาพที่ 4- 10 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับ ซัพพอร์ทเวกเตอร์แมชชีน ในชุดข้อมูลทดสอบ........ 112

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ภายใต้สภาพเศรษฐกิจและสังคมในปัจจุบันที่มีความซับซ้อนประกอบกับมีการเปลี่ยนแปลงอย่างรวดเร็ว นักลงทุนต่างต้องคำนึงถึงผลประโยชน์และความสูญเสียที่อาจเกิดขึ้นจากการตัดสินใจนำสินทรัพย์ที่ตนมีอยู่ไปลงทุน เช่น การซื้อขายหุ้นในตลาดหลักทรัพย์ แม้จะเป็นทางเลือกในการลงทุนที่มีโอกาสได้รับผลตอบแทนสูง แต่ก็มีความเสี่ยงสูงกว่าการลงทุนในรูปแบบอื่น ๆ นักลงทุนจึงต้องมีการวางแผนหรือการตัดสินใจที่ถูกต้องและเป็นระบบเพื่อให้เกิดประโยชน์สูงสุดแก่ตนเองทั้งในระยะสั้นและระยะยาว นอกจากนักลงทุนสามารถใช้ความรู้ ประสบการณ์ และวิจารณญาณของตนเองในการคาดคะเนหรือทำนายราคาปิดหุ้นล่วงหน้าได้ แต่การใช้ข้อมูลราคาปิดหุ้นในอดีตมาพยากรณ์ราคาปิดหุ้นในอนาคตอาจช่วยลดความเสี่ยงเนื่องจากความไม่แน่นอนของเหตุการณ์ที่จะเกิดขึ้นในอนาคตได้ ด้วยเหตุนี้จึงจำเป็นอย่างยิ่งที่จะต้องสรรหาวิธีการทางสถิติที่มีประสิทธิภาพในการพยากรณ์ เพื่อให้ค่าพยากรณ์ที่ได้มีความแม่นยำและมีค่าคลาดเคลื่อนจากการพยากรณ์น้อยที่สุด ซึ่งจะทำให้นักลงทุนเกิดความมั่นใจต่อค่าพยากรณ์ที่ได้และนำไปใช้ในการวางแผนและการตัดสินใจ

การวิเคราะห์อนุกรมเวลา (Time series analysis) เป็นวิธีการทางสถิติวิธีหนึ่งที่มีการนำข้อมูลในอดีตที่เก็บรวบรวมตามลำดับเวลาอย่างต่อเนื่องมาศึกษาหารูปแบบการเคลื่อนไหวของข้อมูลด้วยการสร้างสมการหรือตัวแบบเชิงคณิตศาสตร์แทนลักษณะการเคลื่อนไหวดังกล่าว สำหรับพยากรณ์ข้อมูลในอนาคต (ทรงศิริ แต้สมบัติ, 2553; ภูมิฐาน รังคกูลนุวัฒน์, 2556) ซึ่งมีงานวิจัยมากมายที่พยายามศึกษาและพัฒนาตัวแบบอนุกรมเวลา เช่น ปี ค.ศ. 1970 ได้มีการนำเสนอตัวแบบ ARIMA หรือรู้จักกันดีในชื่อตัวแบบของ Box และ Jenkins (Robert H. Shumway & David S. Stoffer, 2010; ภูมิฐาน รังคกูลนุวัฒน์, 2556) มาใช้อธิบายลักษณะการเคลื่อนไหวของอนุกรมเวลาเมื่อข้อมูลมีสหสัมพันธ์กันในตัวและเป็นแบบคงที่ (Stationary time series) พบว่า ตัวแบบนี้จะให้ค่าพยากรณ์ที่มีความถูกต้องสูงกว่าค่าพยากรณ์ที่ได้จากวิธีการพยากรณ์ด้วยตัวแบบอื่น ๆ ในการพยากรณ์ระยะสั้น ซึ่งเป็นตัวแบบที่เหมาะสำหรับใช้พยากรณ์ข้อมูลอนุกรมเวลาที่เป็นแบบคงที่

(Stationary) ไม่มีแนวโน้ม (Trend) และเป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพามิเตอร์ แต่ข้อมูลอนุกรมเวลาส่วนใหญ่ลักษณะความสัมพันธ์ดังกล่าวมักไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพามิเตอร์ ในปี ค.ศ. 1991 จึงได้มีการนำเสนอตัวแบบเครือข่ายประสาทเทียม (Artificial neural network model: ANN) ขึ้น (สายชล สินสมบูรณ์ทอง, 2558; เอกสิทธิ์ พัชรวงศ์ศักดา, 2557) เพื่อใช้ในการพยากรณ์ข้อมูลอนุกรมเวลา เป็นตัวแบบที่มีความแม่นยำสูง มีความแกร่ง (Robust) เมื่อมีค่านอกเกณฑ์ (Outlier) สามารถเรียนรู้ที่จะทำงานในชุดข้อมูลที่มีค่าคลาดเคลื่อนได้ดี ส่วนใหญ่ใช้ในการพยากรณ์ข้อมูลที่มีความซับซ้อน โดยเฉพาะข้อมูลอนุกรมเวลาในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ รวมทั้งเป็นตัวแบบที่ง่ายต่อการใช้งานเนื่องจากไม่จำเป็นต้องมีเงื่อนไข (Assumptions) ในการสร้างตัวแบบ ต่อมาในปี ค.ศ. 1995 ได้มีการพัฒนาตัวแบบใหม่ขึ้นคือ ตัวแบบซัพพอร์ทเวกเตอร์แมชชีน (Support vector machine model: SVM) (Steve R. Gunn, 1998; ต้อง ศรีคชา, 2550) เป็นตัวแบบที่ให้ผลการพยากรณ์ที่ดี มีความแม่นยำสูง สำหรับใช้พยากรณ์ข้อมูลอนุกรมเวลาที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์โดยมีแนวคิดเดียวกับตัวแบบเครือข่ายประสาทเทียมแต่แตกต่างกันที่กระบวนการลดความเสี่ยงให้มีค่าต่ำสุด (Risk minimization) กล่าวคือ ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนจะใช้กระบวนการลดความเสี่ยงเชิงโครงสร้างให้มีค่าต่ำที่สุด (Structural risk minimization: SRM) ส่วนตัวแบบเครือข่ายประสาทเทียมจะใช้กระบวนการลดความเสี่ยงเชิงการทดลองให้มีค่าต่ำที่สุด (Empirical risk minimization: ERM) แต่ข้อมูลอนุกรมเวลาทางธุรกิจส่วนใหญ่ โดยเฉพาะราคาปิดหุ้นในตลาดหลักทรัพย์มักมีโครงสร้างทั้งส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์และไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ ดังนั้น ตัวแบบ ARIMA, ตัวแบบเครือข่ายประสาทเทียม หรือตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเพียงตัวแบบเดียวอาจไม่สามารถอธิบายลักษณะของข้อมูลอนุกรมเวลาเหล่านั้นได้อย่างถูกต้องและมีความแม่นยำมากนัก

ตัวแบบผสม (Hybrid model) (Zhang, 2003) ได้ถูกพัฒนาขึ้นมาสำหรับแก้ปัญหาดังกล่าว เป็นตัวแบบที่ได้รับความนิยมในการพยากรณ์ข้อมูลอนุกรมเวลาทางธุรกิจ โดยการนำตัวแบบที่ใช้สำหรับพยากรณ์ข้อมูลอนุกรมเวลาในส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ คือ ตัวแบบ ARIMA มารวมกับตัวแบบที่ใช้สำหรับพยากรณ์ข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ คือ ตัวแบบเครือข่ายประสาทเทียม และตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเพื่อทำให้การพยากรณ์มีความถูกต้องและมีความแม่นยำมากยิ่งขึ้น

พงษ์ศิริ ศิริพานิช (พงษ์ศิริ ศิริพานิช, 2550) ได้ทำการพยากรณ์อนุกรมเวลาด้วยตัวแบบผสม ARIMA และเครือข่ายประสาทเทียม โดยใช้ข้อมูลราคาปิดหุ้น PTT และใช้เกณฑ์ในการเปรียบเทียบความแม่นยำของตัวแบบ 3 เกณฑ์ ได้แก่ ค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย และเปอร์เซ็นต์ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย สรุปได้ว่า สำหรับการพยากรณ์ระยะยาว (30 วัน) ตัวแบบผสมระหว่างตัวแบบ ARIMA กับตัวแบบเครือข่ายประสาทเทียมมีความแม่นยำในการพยากรณ์สูงกว่าการใช้ตัวแบบ ARIMA เพียงตัวแบบเดียว นอกจากนี้ Da-yong Zhang และคณะ (Da-yong Zhang, Pu Chen, & Hong-wei Song, 2008) ได้ทำการพยากรณ์ราคาปิดหุ้นในตลาดหลักทรัพย์ด้วยตัวแบบผสม ARMA และซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ข้อมูลราคาปิดหุ้น S&P500 และ Nikkei225 และใช้เกณฑ์ในการเปรียบเทียบความแม่นยำของตัวแบบ 4 เกณฑ์ ได้แก่ ค่าคลาดเคลื่อนกำลังสองเฉลี่ย, รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย และเปอร์เซ็นต์ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย สรุปได้ว่า ตัวแบบผสมระหว่างตัวแบบ ARMA กับตัวแบบซัพพอร์ทเวกเตอร์แมชชีนมีความแม่นยำในการพยากรณ์สูงกว่าการใช้ตัวแบบซัพพอร์ทเวกเตอร์เพียงตัวแบบเดียว

ภัทร วรภู (ภัทร วรภู, 2556) ได้ทำการเปรียบเทียบความแม่นยำของการพยากรณ์อนุกรมเวลาระหว่างตัวแบบผสมและตัวแบบเดี่ยว โดยทำการเปรียบเทียบตัวแบบ ARIMA, ตัวแบบเครือข่ายประสาทเทียม, ตัวแบบซัพพอร์ทเวกเตอร์แมชชีน, ตัวแบบผสมระหว่างตัวแบบ ARIMA กับตัวแบบเครือข่ายประสาทเทียม, ตัวแบบผสมระหว่างตัวแบบ ARIMA กับตัวแบบซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ Hybrid combined โดยใช้ข้อมูล The british pound/US dollar exchange rate และใช้เกณฑ์ในการเปรียบเทียบความแม่นยำของตัวแบบ 4 เกณฑ์ ได้แก่ ค่าคลาดเคลื่อนกำลังสองเฉลี่ย, รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย และเปอร์เซ็นต์ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย สรุปได้ว่า ตัวแบบผสมไม่ได้มีความแม่นยำในการพยากรณ์สูงกว่าตัวแบบเดี่ยวเสมอไป

ในการศึกษาครั้งนี้ผู้วิจัยสนใจที่จะทำการศึกษาและเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริงของราคาปิดหุ้น SCB และชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,0), ตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ

ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ โดยตัวแบบใดที่มีค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ให้ผลดีสุด

1.2 วัตถุประสงค์ของการวิจัย

เพื่อเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB

1.3 ข้อตกลงเบื้องต้น

ในการศึกษาครั้งนี้มีข้อตกลงเบื้องต้นสำหรับการดำเนินงานวิจัยดังนี้

ศึกษาตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยที่ตัวแปรที่สนใจศึกษา คือ ข้อมูลราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ซึ่งเป็นข้อมูลที่มีลักษณะเป็นอนุกรมเวลา (Time series) หรืออนุกรมเวลาแบบสุ่ม (Stochastic process หรือ Random process) นั่นคือ มีการเก็บรวบรวมข้อมูลของตัวแปรหนึ่งตามลำดับเวลาอย่างต่อเนื่อง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.4 ขอบเขตของการวิจัย

ในการศึกษาครั้งนี้จะทำการศึกษาในส่วนของข้อมูลจริงและข้อมูลจำลอง ภายใต้ขอบเขตการวิจัยดังนี้

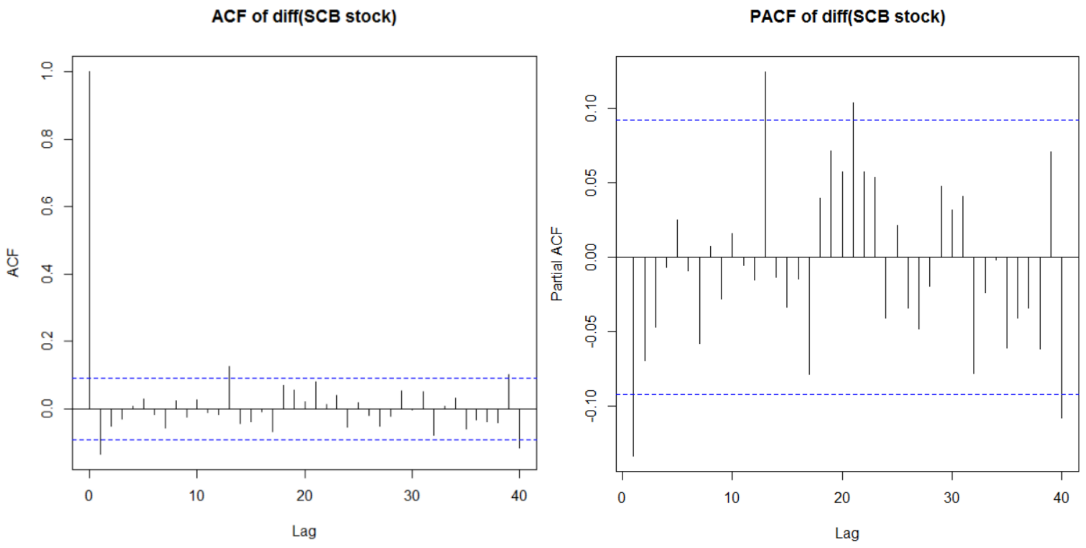
1.4.1 ข้อมูลจริง

ชุดข้อมูลราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ตั้งแต่วันที่ 2 เดือนกรกฎาคม พ.ศ. 2550 ถึงวันที่ 8 เดือนกุมภาพันธ์ พ.ศ. 2559 จำนวน 456 สัปดาห์ ชุดข้อมูลนี้ได้มาจาก http://finance.yahoo.com/q?s=SCB.BK โดยทำการศึกษาและเปรียบเทียบตัวแบบภายใต้ขอบเขตของการวิจัย ดังนี้

1.4.1.1 ตัวแบบ ARIMA(p,d,q) ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 กล่าวคือ จะทำการเปรียบเทียบตัวแบบทั้งหมดจำนวน 9 ตัวแบบ คือ ตัวแบบ ARIMA(0,1,0), ตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อข้อมูลราคาปิดหุ้น SCB เป็นแบบคงที่ (Stationary) และไม่มีแนวโน้ม (Trend) จากนั้นจะทำการพิจารณาคัดเลือกตัวแบบข้างต้นโดยตัวแบบที่เหมาะสมที่สุด คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด ซึ่งแสดงได้ดังนี้

p \ q	0	1	2
0	2,045.402	1,775.070	1,764.654
1	1,911.304	1,764.347	1,766.271
2	1,845.700	1,766.208	1,765.949

ตารางที่ 1- 1 เกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ของตัวแบบ ARIMA(p,d=1,q)



ภาพที่ 1- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ ARIMA(1,1,1)

1.4.1.2 ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม โดยนำค่าส่วนเหลือ (Residuals) ที่ได้จากตัวแบบ ARIMA ไปพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัวแบบเครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนดตั้งแต่ 1 ถึง 5 โหนดโดยใช้ Sigmoid logistics function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดจำนวนรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ กล่าวคือ จะทำการเปรียบเทียบตัวแบบเครือข่ายประสาทเทียมทั้งหมดจำนวน 5 ตัวแบบ คือ ตัวแบบเครือข่ายประสาทเทียม 1-1-1, ตัวแบบเครือข่ายประสาทเทียม 1-2-1, ตัวแบบเครือข่ายประสาทเทียม 1-3-1, ตัวแบบเครือข่ายประสาทเทียม 1-4-1 และตัวแบบเครือข่ายประสาทเทียม 1-5-1 จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมข้างต้นโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า

1.4.1.3 ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยนำค่าส่วนเหลือ (Residuals) ที่ได้จากตัวแบบ ARIMA ไปพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอยภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา เมื่อกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial basis function เป็น Kernel function และกำหนดจำนวนรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนข้างต้นโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า

1.4.2 ข้อมูลจำลอง

ทำการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อชุดข้อมูลจำลองอนุกรมเวลาเป็นแบบคงที่ (Stationary) และไม่มีแนวโน้ม (Trend) โดยกระบวนการจำลองจะนำค่าพารามิเตอร์ที่ Estimate ได้

จากการสร้างตัวแบบ ARIMA ในชุดข้อมูลจริงของราคาปิดหุ้น SCB มาใช้สำหรับการจำลอง โดยแต่ละตัวแบบจะทำการจำลองชุดข้อมูลอนุกรมเวลาภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ จากนั้นนำค่าส่วนเหลือ (Residuals) ที่ได้จากชุดข้อมูลจำลองไปพยากรณ์ด้วยตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ภายใต้ขอบเขตของการวิจัยในข้อมูลจริงข้างต้น

1.5 คำจำกัดความที่ใช้ในงานวิจัย

อนุกรมเวลา (Time series)

คือ ชุดข้อมูลที่สังเกตค่าตามลำดับเวลาการเกิด นอกจากนี้ อนุกรมเวลา คือ อนุกรมของค่าวัดหรือค่าสังเกตตามลำดับเวลาของตัวแปรสุ่ม โดยทั่วไปมักมีระยะห่างของเวลาเท่า ๆ กัน เช่น รายวัน รายเดือน รายไตรมาส รายปี มักนำเสนอโดยใช้แผนภาพเส้น

การจำลอง (Simulation)

คือ วิธีการที่ใช้วิเคราะห์ปัญหาที่เกี่ยวข้องกับตัวแปรสุ่มที่ซับซ้อนหรือไม่มีวิธีการหาผลลัพธ์เชิงวิเคราะห์ โดยใช้เลขคล้ายสุ่มแทนตัวแปรสุ่มที่เกี่ยวข้อง

ตัวแบบผสม (Hybrid model)

คือ ตัวแบบสำหรับพยากรณ์ข้อมูลอนุกรมเวลาที่มีแนวคิดว่าข้อมูลอนุกรมเวลาชุดหนึ่งจะมีทั้งส่วนประกอบที่เป็นเชิงฟังก์ชันเชิงเส้นตรงและไม่เป็นฟังก์ชันเชิงเส้นตรงรวมกัน

ความแม่นยำ (Accuracy)

คือ สมบัติของการวัดค่าหลาย ๆ ครั้ง ที่สามารถให้ค่าที่ถูกต้องหรือใกล้เคียงกับค่าจริง หรือสมบัติของตัวประมาณที่ให้ค่าประมาณที่มีค่าคลาดเคลื่อนจากค่าจริงน้อย ซึ่งวัดโดยค่าคลาดเคลื่อนกำลังสองเฉลี่ย, รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย และเปอร์เซ็นต์ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย เป็นต้น

ค่าพี (p-value)

คือ ค่านัยสำคัญน้อยที่สุดที่ทำให้ปฏิเสธสมมุติฐานว่าง นั่นคือ ค่าความน่าจะเป็นที่แสดงถึงความเสี่ยงในการปฏิเสธสมมุติฐานว่างเมื่อสมมุติฐานว่างเป็นจริง

ถ้าให้ T แทนตัวสถิติทดสอบ และ t แทนค่าของสถิติทดสอบ การคำนวณค่าพี แบ่งได้เป็น 3 กรณี ดังนี้

1. ในการทดสอบด้านเดียวโดยช่วงปฏิเสธอยู่ด้านซ้าย : $p - value = P(T \le t \mid H_0 \text{ is true})$

2. ในการทดสอบด้านเดียวโดยช่วงปฏิเสธอยู่ด้านขวา : $p - value = P(T \ge t \mid H_0 \text{ is true})$

3. ในการทดสอบสองด้าน : $p - value = 2P(T \ge |t| \mid H_0 \text{ is true})$

นอกจากนี้ ค่าพี คือ ความน่าจะเป็นของค่าคลาดเคลื่อนที่เกิดจากการใช้ตัวอย่างสุ่มชุดหนึ่งเพื่อใช้ในการตัดสินใจ หากความน่าจะเป็นของค่าคลาดเคลื่อนมีค่าน้อยกว่าค่าความน่าจะเป็นของค่าคลาดเคลื่อนที่กำหนด จะปฏิเสธสมมุติฐานว่าง หรืออาจกล่าวได้ว่าค่าความน่าจะเป็นของค่าคลาดเคลื่อนที่ยอมรับได้ แต่หลักฐานจากตัวอย่างให้ค่าความน่าจะเป็นของค่าคลาดเคลื่อนน้อยกว่าที่กำหนด จะสามารถปฏิเสธสมมุติฐานว่างได้

ราคาปิดหุ้น (Close price)

คือ ราคาของหุ้นที่เกิดจากการซื้อขายในตลาดหลักทรัพย์เป็นรายการสุดท้ายของแต่ละวัน

1.6 เกณฑ์ที่ใช้ในการตัดสินใจ

จากการศึกษาและวิเคราะห์ข้อมูลราคาปิดของหุ้น SCB ด้วยตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน จะทำการพิจารณาว่าตัวแบบที่ได้จากการวิเคราะห์ตัวแบบใดจะมีความเหมาะสมมากที่สุด โดยการพิจารณาคัดเลือกตัวแบบที่มีค่าคลาดเคลื่อนที่ต่ำกว่า ด้วยการพิจารณาค่าวัดดังต่อไปนี้

1.6.1 ค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE) เป็นค่าที่ใช้วัดความแม่นยำของตัวประมาณที่วัดจากกำลังสองของค่าคลาดเคลื่อนของการพยากรณ์ มีสูตรในการคำนวณดังนี้

$$MSE = \frac{\sum_{t=1}^{n} \left(x_t - \hat{x}_t \right)^2}{n - p} \qquad (1\text{-}1)$$

เมื่อ n แทน ขนาดของอนุกรมเวลา และ p แทน จำนวนพารามิเตอร์ในตัวแบบ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1.6.2 รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นค่าที่ใช้วัดความแม่นยำของตัวประมาณที่วัดจากรากที่สองของค่าคลาดเคลื่อนกำลังสองเฉลี่ย ซึ่งมีหน่วยวัดเดียวกับค่าสังเกต มีสูตรในการคำนวณดังนี้

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{t=1}^{n}\left(x_t - \hat{x}_t\right)^2}{n-p}} \tag{1-2}$$

เมื่อ n แทน ขนาดของอนุกรมเวลา และ p แทน จำนวนพารามิเตอร์ในตัวแบบ

1.6.3 ค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) เป็นค่าที่ใช้วัดความแม่นยำของตัวประมาณที่วัดจากขนาดของค่าคลาดเคลื่อนของการพยากรณ์โดยไม่คำนึงถึงทิศทางของค่าคลาดเคลื่อน ซึ่งมีหน่วยวัดเดียวกับค่าสังเกต มีสูตรในการคำนวณดังนี้

$$MAE = \frac{\sum_{t=1}^{n}\left|x_t - \hat{x}_t\right|}{n-p} \tag{1-3}$$

เมื่อ n แทน ขนาดของอนุกรมเวลา และ p แทน จำนวนพารามิเตอร์ในตัวแบบ

โดยทั้ง 3 เกณฑ์นั้น ตัวแบบใดที่มีค่าคลาดเคลื่อนต่ำสุด จะเป็นตัวแบบที่ดีที่สุด

<u>**หมายเหตุ**</u> : เนื่องจากเกณฑ์ที่ใช้ในการเปรียบเทียบตัวแบบมีด้วยกันหลายค่าวัดและยังไม่มีผลการวิจัยใดที่ระบุได้แน่ชัดว่าค่าวัดใดมีความถูกต้องและเหมาะสมที่สุด ดังนั้น งานวิจัยนี้จะเลือกใช้รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ เนื่องจากเป็นค่าวัดที่มีหน่วยวัดเดียวกับค่าสังเกต ซึ่งจะทำให้สามารถเปรียบเทียบตัวแบบได้สะดวกและมีความชัดเจนมากยิ่งขึ้น

1.7 วิธีการศึกษา

1. ศึกษาตัวแบบและทฤษฎีที่เกี่ยวข้อง

2. เก็บรวบรวมข้อมูลจริงราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ตั้งแต่วันที่ 2 เดือนกรกฎาคม พ.ศ. 2550 ถึงวันที่ 8 เดือนกุมภาพันธ์ พ.ศ. 2559 จำนวน 456 สัปดาห์ จากเว็บไซต์ http://finance.yahoo.com/q?s=SCB.BK โดยแบ่งชุดข้อมูลราคาปิดหุ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

SCB ออกเป็นชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับประเมินความถูกต้องของตัวแบบ

3. ทำการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) ซึ่งในแต่ละตัวแบบจะทำการจำลองชุดข้อมูลภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยในแต่ละรอบของการจำลองจะทำการจำลองชุดข้อมูลอนุกรมเวลาจำนวน 456 สัปดาห์ และแบ่งออกเป็นชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับประเมินความถูกต้องของตัวแบบ และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ และคำนวณหาค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากการจำลองข้อมูลในแต่ละรอบ

4. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนจากข้อมูลจำลองในแต่ละรอบ มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network model) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัวแบบเครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนด 1 ถึง 5 โหนดโดยใช้ Sigmoid logistic function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบเครือข่ายประสาทเทียมในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจำลองในแต่ละรอบ

หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) ในแต่ละรอบของการจำลอง หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ

5. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนจากข้อมูลจำลองในแต่ละรอบมาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression) ภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา และกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial function เป็น Kernel function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจำลองในแต่ละรอบ หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) ในแต่ละรอบของการจำลอง หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ

6. นำชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์จากข้อมูลจริงมาสร้างตัวแบบ ARIMA ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 จากนั้นทำการพิจารณาคัดเลือกตัวแบบ

ข้างต้นโดยตัวแบบที่เหมาะสมที่สุด คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด

7. ทำการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริง ซึ่งจัดเป็นข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) หลังจากนั้นคำนวณหาค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น

8. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network model) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัวแบบเครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนด 1 ถึง 5 โหนดโดยใช้ Sigmoid logistic function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบเครือข่ายประสาทเทียมในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

9. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression) ภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา และกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial function เป็น Kernel function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมชองค่าพารามิเตอร์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE)

10. ทำการเปรียบเทียบความแม่นยำของตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ โดยตัวแบบใดที่มีค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ให้ผลดีสุด

11. วิเคราะห์และสรุปผลการวิจัย

1.8 ประโยชน์ที่คาดว่าจะได้รับ

เพื่อเป็นแนวทางสำหรับพยากรณ์ราคาปิดหุ้นตัวอื่น ๆ ในตลาดหลักทรัพย์ต่อไป ด้วยตัวแบบอนุกรมเวลาแบบผสม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2

ทฤษฎีและตัวสถิติที่เกี่ยวข้อง

2.1 ตัวแบบ Integrated autoregressive moving average หรือ ตัวแบบ ARIMA(p,d,q)

ตัวแบบ ARIMA เป็นตัวแบบที่พัฒนาขึ้นโดย George E.P.Box และ Gwilym M.Jenkins โดยได้เขียนไว้ในหนังสือชื่อ Time series analysis: Forecasting and Control และพิมพ์จำหน่ายครั้งแรกในปี ค.ศ. 1970 เป็นวิธีการพยากรณ์ที่นำข้อมูลอนุกรมเวลาในอดีตที่เก็บรวบรวมตามลำดับเวลาอย่างต่อเนื่องมาหาตัวแบบที่เหมาะสม และใช้ตัวแบบที่ได้มาพยากรณ์ข้อมูลอนุกรมเวลาในอนาคต ซึ่งเป็นวิธีการหนึ่งที่มีความแม่นยำ (Accuracy) เหมาะสมจะนำไปประยุกต์ใช้กับข้อมูลทางธุรกิจและเศรษฐกิจ เนื่องจากรูปแบบของข้อมูลจะเปลี่ยนแปลงไปเรื่อย ๆ โดยข้อมูลอนุกรมเวลาที่นำมาวิเคราะห์นั้นต้องเป็นแบบคงที่ (Stationary) หรือบางครั้งเรียกว่า คงที่อย่างอ่อน (Weakly stationary) กล่าวคือ

1. ค่าเฉลี่ยของตัวแปร x ในแต่ละช่วงเวลา t มีค่าคงที่ หรือเขียนได้ว่า $E(x_t) = \mu$ สำหรับทุกช่วงเวลา $t = 1,2,...,n$

2. ความแปรปรวนของตัวแปร x ในแต่ละช่วงเวลา t มีค่าคงที่ หรือเขียนได้ว่า $\text{var}(x_t) = \gamma(0,0) = E[(x_t - \mu)^2]$ สำหรับทุกช่วงเวลา $t = 1,2,...,n$

3. ความแปรปรวนร่วมของตัวแปร x ณ เวลา $t+h$ และเวลา t จะมีค่าคงที่ โดยความแปรปรวนร่วมระหว่างข้อมูลอนุกรมเวลา x_t ที่ต่างช่วงเวลากัน จะขึ้นอยู่กับระยะห่างของช่วงเวลาทั้งสอง ซึ่งก็คือ h นั่นเอง และไม่ขึ้นกับเวลา t หรือกล่าวอีกอย่างหนึ่งว่า ความแปรปรวนร่วมระหว่างข้อมูลอนุกรมเวลา x_t ที่ต่างช่วงเวลากัน จะไม่ขึ้นอยู่กับว่าขณะนั้นตัวแปร x_t อยู่ที่ ณ เวลา $t+h$ หรือ t หรือเขียนได้ว่า $\gamma(t+h,t) = \text{cov}(x_{t+h}, x_t) = \text{cov}(x_h, x_0) = \gamma(h,0)$ สำหรับทุกช่วงเวลา $t = 1,2,...,n$

และข้อมูลอนุกรมเวลาต้องไม่มีแนวโน้ม (Trend) จึงจะสามารถนำตัวแบบของ Box และ Jenkins ไปใช้พยากรณ์ข้อมูลอนุกรมเวลาได้ ซึ่งการพยากรณ์ด้วยวิธีการดังกล่าวจะเลือกตัวแบบที่ใช้ในการพยากรณ์ โดยพิจารณาจากลักษณะของฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function:

ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) โดยตัวแบบที่เป็นไปได้ในเบื้องต้นอาจมีมากกว่า 1 ตัวแบบ ดังนั้น จึงต้องมีขั้นตอนการตรวจสอบเพื่อเลือกตัวแบบที่เหมาะสมที่สุดเพื่อใช้ในการพยากรณ์อนุกรมเวลาดังกล่าว ซึ่งตัวแบบ ARIMA ประกอบไปด้วย 3 ส่วนหลัก ๆ ได้แก่ ตัวแบบ Autoregressive, ตัวแบบ Moving average และกระบวนการหาผลต่าง (Differencing) ที่ถูกนำมาใช้แปลงข้อมูลอนุกรมเวลาแบบไม่คงที่ให้เป็นข้อมูลอนุกรมเวลาแบบคงที่ โดยมีรายละเอียดดังนี้

2.1.1 ตัวแบบ Autoregressive หรือ ตัวแบบ AR(p)

ตัวแบบ AR(p) เป็นตัวแบบที่แสดงถึงค่าปัจจุบันของอนุกรมเวลา x_t เป็นฟังก์ชันของค่าอนุกรมเวลาในอดีต p ค่า คือ $x_{t-1}, x_{t-2}, ..., x_{t-p}$ เมื่อ p คือ จำนวนช่วงเวลาที่ช้ากว่ากันของค่าอนุกรมเวลาในอดีต p หน่วยเวลาที่ต้องการสำหรับพยากรณ์ค่าอนุกรมเวลาในปัจจุบัน ซึ่งสามารถเขียนอยู่ในรูปสมการ ได้ดังนี้

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t \qquad (2\text{-}1)$$

เมื่อ $\{x_t ; t = 0, \pm 1, \pm 2, ...\}$ เป็นอนุกรมเวลาแบบคงที่ (Stationary time series)

$\phi_1, \phi_2, ..., \phi_p$ เป็นค่าคงที่ โดยที่ $\phi_p \neq 0$

และ w_t เป็นตัวรบกวนขาวแบบเกาส์เซียน (Gaussian white noise) ที่มีค่าเฉลี่ย 0 และความแปรปรวน σ_w^2

ถ้าอนุกรมเวลา x_t มีค่าเฉลี่ยเท่ากับ μ โดยที่ $\mu \neq 0$ เราสามารถเขียนตัวแบบ AR(p) ข้างต้นให้อยู่ในรูปสมการ ได้ดังนี้

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t \qquad (2\text{-}2)$$

เมื่อ $\alpha = \mu(1 - \phi_1 - ... - \phi_p)$

จากสมการที่ (2-1) เราสามารถใช้ Backshift operator เขียนตัวแบบ AR(p) ได้ดังนี้

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p) x_t = w_t \qquad (2\text{-}3)$$

เมื่อ $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p$ คือ Autoregressive operator

ตัวอย่าง ตัวแบบอนุกรมเวลา AR(1) สามารถเขียนได้ในรูปแบบสมการ $x_t = \phi x_{t-1} + w_t$ หรือ $(1-\phi B)x_t = w_t$ โดยเงื่อนไขที่ทำให้ตัวแบบอนุกรมเวลา AR(1) เป็น Causal process คือ รากของสมการ $\phi(B) = 1 - \phi B$ ต้องอยู่นอกวงกลมหนึ่งหน่วย (Outside of the unit circle) กล่าวคือ $\phi(B) = 0$ เมื่อ $|B| > 1$ เท่านั้น จะได้รากของสมการ $\phi(B) = 0$ คือ $B = \dfrac{1}{\phi}$ ดังนั้น $|B| > 1$ ก็ต่อเมื่อ $|\phi| < 1$ ซึ่งเป็นเงื่อนไขที่ทำให้อนุกรมเวลา x_t เป็น Causal process

โดยการทำซ้ำแบบย้อนหลัง (Iterating backwards) ไป k ช่วงเวลา จะได้ว่า

$$\begin{aligned} x_t &= \phi x_{t-1} + w_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t \\ &= \phi^2 x_{t-2} + \phi w_{t-1} + w_t \\ &\vdots \\ &= \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j} \end{aligned} \tag{2-4}$$

จากเงื่อนไข $|\phi| < 1$ ข้างต้นทำให้สามารถเขียนตัวแบบ AR(1) ได้ในรูปของกระบวนการเชิงเส้น (Linear process) ดังนี้

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j} \tag{2-5}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

และกล่าวได้ว่าสมการที่ (2-5) หาค่า limit ได้ใน mean square sense เนื่องจาก

$$\lim_{k\to\infty} \left(x_t - \sum_{j=0}^{k-1} \phi^j w_{t-j} \right)^2 = \lim_{k\to\infty} \phi^{2k} E\left(x_{t-k}^2\right) = 0 \tag{2-6}$$

จากการพิจารณากระบวนการ AR(1) ซึ่งนิยามในสมการที่ (2-5) พบว่าเป็นอนุกรมเวลาแบบคงที่ (Stationary time series) โดยมีค่าเฉลี่ย คือ

$$E(x_t) = \sum_{j=0}^{\infty} \phi^j E(w_{t-j}) = 0 \tag{2-7}$$

ฟังก์ชันความแปรปรวนร่วมในตัว (Autocovariance function) คือ

$$\gamma(t+h,t)=\mathrm{cov}(x_{t+h},x_t)=E\left[\left(\sum_{j=0}^{\infty}\phi^j w_{t+h-j}\right)\left(\sum_{k=0}^{\infty}\phi^k w_{t-k}\right)\right]$$

$$=\sigma_w^2\sum_{j=0}^{\infty}\phi^j\phi^{j+h}=\sigma_w^2\phi^h\sum_{j=0}^{\infty}\phi^{2j}=\frac{\sigma_w^2\phi^h}{1-\phi^2},\quad h\geq 0 \qquad (2\text{-}8)$$

ฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) คือ

$$\rho(h)=\frac{\gamma(h)}{\gamma(0)}=\phi^h,\quad h\geq 0 \qquad (2\text{-}9)$$

ฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) คือ

$$\phi_{11}=corr(x_1,x_0)=\rho(1)=\phi \qquad (2\text{-}10)$$

และ $$\phi_{22}=corr\left(x_2-x_2^1,x_0-x_0^1\right)$$

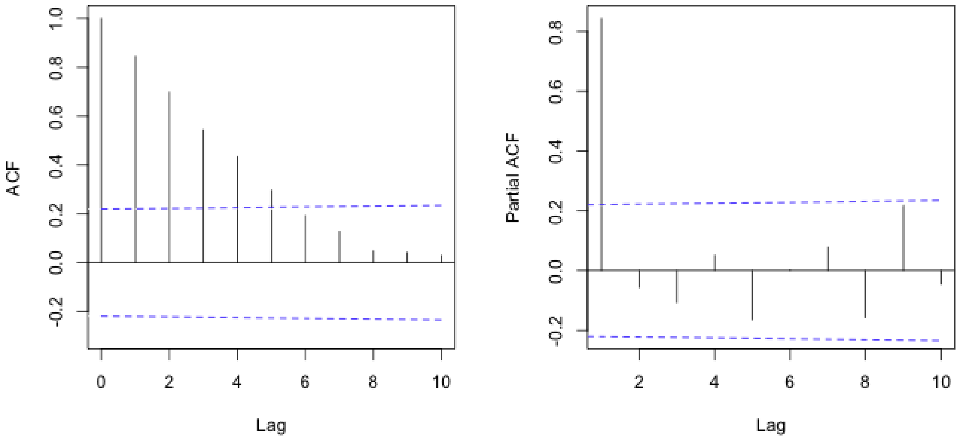
พิจารณาหาค่า x_2^1 จากสมการการถดถอย $x_2^1=\beta x_1$ จากนั้นจะทำการหาค่า β ที่ทำให้ $E(x_2-\beta x_1)^2$ มีค่าน้อยที่สุด โดยการหาอนุพันธ์ของ $E(x_2-\beta x_1)^2$ เทียบกับ β และให้เท่ากับ 0 จะได้ $\beta=\phi$ นั่นคือ $x_2^1=\phi x_1$

พิจารณาหาค่า x_0^1 จากสมการการถดถอย $x_0^1=\beta x_1$ จากนั้นจะทำการหาค่า β ที่ทำให้ $E(x_0-\beta x_1)^2$ มีค่าน้อยที่สุด โดยการหาอนุพันธ์ของ $E(x_0-\beta x_1)^2$ เทียบกับ β และให้เท่ากับ 0 จะได้ $\beta=\phi$ นั่นคือ $x_0^1=\phi x_1$

แต่เนื่องจาก $\gamma(h)=\gamma(0)\phi^h$ ดังนั้น $corr\left(x_2-x_2^1,x_0-x_0^1\right)=\gamma(2)-2\phi\gamma(1)+\phi^2\gamma(0)=0$

นั่นคือ $$\phi_{22}=corr\left(x_2-x_2^1,x_0-x_0^1\right)=0 \qquad (2\text{-}11)$$

ในทำนองเดียวกันจะได้ว่า $\phi_{hh}=0$ สำหรับทุก $h>1$



ภาพที่ 2- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ AR(1) เมื่อ $\phi > 0$

(ที่มา: https://www.unc.edu/courses/2010fall/ecol/563/001/)

<u>ตัวอย่าง</u> ตัวแบบอนุกรมเวลา AR(2) สามารถเขียนได้ในรูปแบบสมการ $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$ หรือ $(1-\phi_1 B - \phi_2 B^2)x_t = w_t$ โดยเงื่อนไขที่ทำให้ตัวแบบอนุกรมเวลา AR(2) เป็น Causal process คือ รากของสมการ $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$ ต้องอยู่นอกวงกลมหนึ่งหน่วย (Outside of the unit circle) กล่าวคือ $\phi(B) = 0$ เมื่อ $|B| > 1$ เท่านั้น จะได้ รากของสมการ $\phi(B) = 0$ คือ $B = \dfrac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2}$ ดังนั้น $\left|\dfrac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2}\right| > 1$ ก็ต่อเมื่อ $\phi_1 + \phi_2 < 1, \phi_2 - \phi_1 < 1$ และ $|\phi_2| < 1$ ซึ่งเป็นเงื่อนไขที่ทำให้อนุกรมเวลา เป็น Causal process

การคำนวณหาค่าฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) สำหรับตัวแบบ AR(2) สามารถหาได้จากสมการ $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$

โดยคูณทั้งสองข้างของสมการด้วย x_{t-h} จะได้ $x_t x_{t-h} = \phi_1 x_{t-1} x_{t-h} + \phi_2 x_{t-2} x_{t-h} + w_t x_{t-h}$

ใส่ค่าคาดหมายตลอดสมการ จะได้ $E(x_t x_{t-h}) = \phi_1 E(x_{t-1} x_{t-h}) + \phi_2 E(x_{t-2} x_{t-h}) + E(w_t x_{t-h})$

เนื่องจาก $E(w_t x_{t-h}) = E\left(w_t \sum_{j=0}^{\infty} \psi_j w_{t-h-j}\right) = 0$

ดังนั้น $E(x_t x_{t-h}) = \phi_1 E(x_{t-1} x_{t-h}) + \phi_2 E(x_{t-2} x_{t-h})$

หรือ สามารถเขียนสมการได้ในรูปของ $\gamma(h) = \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2), \quad h = 1,2,\ldots$

โดยหารทั้งสองข้างของสมการด้วย $\gamma(0)$ จะได้สมการเชิงอนุพันธ์เอกพันธ์ (Homogeneous differential equations) อันดับที่ 2 ของฟังก์ชันสหสัมพันธ์ในตัวสำหรับตัวแบบ AR(2) คือ

$$\rho(h)-\phi_1\rho(h-1)-\phi_2\rho(h-2)=0, \quad h=1,2,\ldots \tag{2-12}$$

ภายใต้เงื่อนไขเริ่มต้น (Initial conditions) คือ $\rho(0)=1$ และ $\rho(-1)=\dfrac{\phi_1}{1-\phi_2}$ ทั้งนี้ $\rho(1)=\rho(-1)$

สมมติให้ z_1 และ z_2 เป็นรากของสมการ $\alpha(z)=1-\phi_1 z-\phi_2 z^2$
โดยที่ $\alpha(z_1)=\alpha(z_2)=0$

<u>กรณีที่ 1</u> : z_1 และ z_2 เป็นจำนวนจริงที่แตกต่างกัน กล่าวคือ $z_1 \neq z_2$
ดังนั้น ผลเฉลยทั่วไป (General solution) ของสมการที่ (2-12) คือ $\rho(h)=c_1 z_1^{-h}+c_2 z_2^{-h}$
เมื่อ z_1^{-h} และ z_2^{-h} เป็นอิสระเชิงเส้นกัน

<u>กรณีที่ 2</u> : z_1 และ z_2 เป็นจำนวนจริงที่เท่ากัน กล่าวคือ $z_1 = z_2 (= z_0)$
ดังนั้น ผลเฉลยทั่วไป (General solution) ของสมการที่ (2-12) คือ $\rho(h)=z_0^{-h}(c_1+c_2 h)$
เมื่อ z_0^{-h} และ $z_0^{-h}h$ เป็นอิสระเชิงเส้นกัน

<u>กรณีที่ 3</u> : z_1 และ z_2 เป็นคู่สังยุคของจำนวนเชิงซ้อน (Complex conjugate pair) กล่าวคือ $z_1=\bar{z}_2$

ดังนั้น ผลเฉลยทั่วไป (General solution) ของสมการที่ (2-12) คือ $\rho(h)=c_1 z_1^{-h}+\bar{c}_1\bar{z}_1^{-h}$ โดยที่ $c_2=\bar{c}_1$ และจากสูตรของออยเลอร์ (Euler’s formula) จะได้ว่า $\rho(h)=a|z_1|^{-h}\cos(h\theta+b)$ เมื่อ $\theta\in[-\pi,\pi]$

การคำนวณหาค่าฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) สำหรับตัวแบบ AR(2) สามารถหาได้โดยใช้ The Durbin-Levinson Algorithm ซึ่งนิยามโดย

$$\phi_{nn} = \frac{\rho(n) - \sum_{k-1}^{n-1} \phi_{n-1,k}\,\rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k}\,\rho(k)} \quad \text{สำหรับ } n \ge 1 \tag{2-13}$$

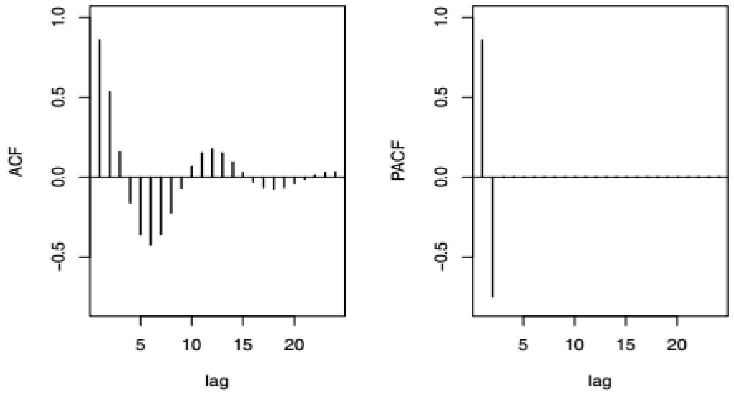
ดังนั้น $\phi_{11} = \rho(1) = \dfrac{\phi_1}{1-\phi_2}$ (2-14)

$$\phi_{22} = \frac{\rho(2) - \rho(1)^2}{1-\rho(1)^2} = \frac{\left[\phi_1\left(\frac{\phi_1}{1-\phi_2}\right) + \phi_2\right] - \left(\frac{\phi_1}{1-\phi_2}\right)^2}{1 - \left(\frac{\phi_1}{1-\phi_2}\right)^2} = \phi_2 \tag{2-15}$$

$$\phi_{21} = \phi_1 \tag{2-16}$$

และ $\phi_{33} = \dfrac{\rho(3) - \phi_1\rho(2) - \phi_2\rho(1)}{1 - \phi_1\rho(1) - \phi_2\rho(2)} = 0$ (2-17)

ในทำนองเดียวกันจะได้ว่า $\phi_{hh} = 0$ สำหรับทุก $h > 2$



ภาพที่ 2- 2 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ AR(2) เมื่อ $\phi_1 = 1.5$ และ $\phi_2 = -0.75$

(ที่มา: Time series analysis and its applications with R examples)

2.1.2 ตัวแบบ Moving average หรือ ตัวแบบ MA(q)

ตัวแบบ MA(q) เป็นตัวแบบที่แสดงถึงค่าปัจจุบันของอนุกรมเวลา x_t เป็นฟังก์ชันของตัวรบกวนขาวแบบเกาส์เซียนตั้งแต่อดีตจนถึงปัจจุบัน คือ $w_t, w_{t-1}, w_{t-2}, ..., w_{t-q}$ เมื่อ q คือ จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) ของตัวรบกวนขาวแบบเกาส์เซียนตั้งแต่อดีตจนถึงปัจจุบัน q หน่วยเวลา ซึ่งสามารถเขียนในรูปสมการ ได้ดังนี้

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q} \tag{2-18}$$

เมื่อ $\{x_t ; t = 0, \pm 1, \pm 2, ...\}$ เป็นอนุกรมเวลาแบบคงที่ (Stationary time series)

$\theta_1, \theta_2, ..., \theta_q$ เป็นค่าคงที่ โดยที่ $\theta_q \neq 0$

และ w_t เป็นตัวรบกวนขาวแบบเกาส์เซียน (Gaussian white noise) ที่มีค่าเฉลี่ย 0 และความแปรปรวน σ_w^2

จากสมการที่ (2-18) เราสามารถใช้ Backshift operator เขียนตัวแบบ MA(q) ได้ดังนี้

$$x_t = \theta(B) w_t \tag{2-19}$$

เมื่อ $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - ... - \theta_q B^q$ คือ Moving average operator

<u>ตัวอย่าง</u> ตัวแบบอนุกรมเวลา MA(1) สามารถเขียนได้ในรูปแบบสมการ $x_t = w_t + \theta w_{t-1}$ หรือ $x_t = (1 - \theta B) w_t$ โดยเงื่อนไขที่ทำให้ตัวแบบอนุกรมเวลา MA(1) เป็น Invertible process (Infinite AR representation) คือ รากของสมการ $\theta(B) = 1 - \theta B$ ต้องอยู่นอกวงกลมหนึ่งหน่วย (Outside of the unit circle) กล่าวคือ $\theta(B) = 0$ เมื่อ $|B| > 1$ เท่านั้น จะได้รากของสมการ $\theta(B) = 0$ คือ $B = \dfrac{1}{\theta}$ ดังนั้น $|B| > 1$ ก็ต่อเมื่อ $|\theta| < 1$ ซึ่งเป็นเงื่อนไขที่ทำให้อนุกรมเวลา x_t เป็น Invertible process

ถ้า $|\theta| < 1$ แล้วเราสามารถเขียนสมการ $x_t = \theta(B) w_t$ เมื่อ $\theta(B) = 1 - \theta B$ ได้ในรูปของสมการ $\pi(B) x_t = w_t$ เมื่อ $\pi(B) = \theta^{-1}(B) = \dfrac{1}{1 + \theta B} = \sum_{j=0}^{\infty} (-\theta)^j B^j$

โดยมีค่าเฉลี่ย คือ

$$E(x_t)=\sum_{j=0}^{1}\theta_j E(w_{t-j})=0 \qquad (2\text{-}20)$$

ฟังก์ชันความแปรปรวนร่วมในตัว (Autocovariance function) คือ

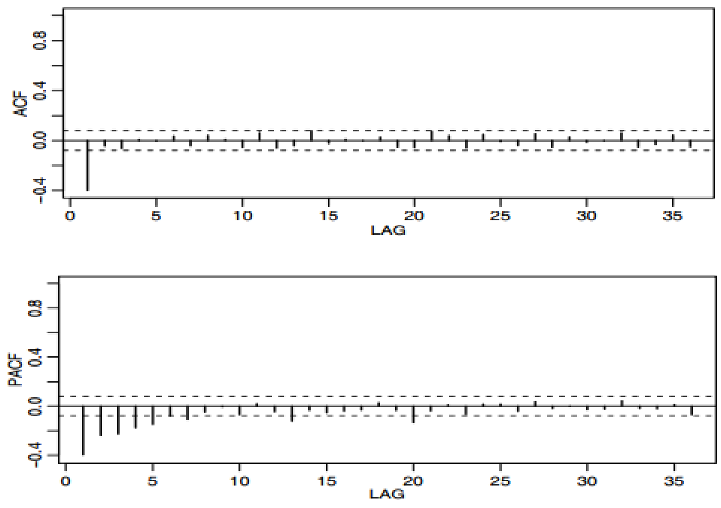
$$\gamma(h)=\begin{cases}(1+\theta^2)\sigma_w^2, & h=0\\ \theta\sigma_w^2, & h=1\\ 0, & h>1\end{cases} \qquad (2\text{-}21)$$

ฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) คือ

$$\rho(h)=\begin{cases}\dfrac{\theta}{(1+\theta^2)}, & h=1\\ 0, & h>1\end{cases}$$

และ ฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) คือ

$$\phi_{hh}=-\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2(h+1)}}, \quad h\geq 1 \qquad (2\text{-}23)$$



ภาพที่ 2- 3 ฟังก์ชันสหสัมพันธ์ในตัวและฟังก์ชันสหสัมพันธ์บางส่วนของตัวแบบ MA(1)

(ที่มา: Time series analysis and its applications with R examples)

2.1.3 ตัวแบบ Autoregressive moving average หรือ ตัวแบบ ARMA(p,q)

ตัวแบบ ARMA(p,q) เป็นตัวแบบที่แสดงถึงค่าปัจจุบันของอนุกรมเวลา x_t เป็นฟังก์ชันของค่าอนุกรมเวลาในอดีต p หน่วยเวลา และตัวรบกวนขาวแบบเกาส์เซียนตั้งแต่อดีตจนถึงปัจจุบัน q หน่วยเวลา ซึ่งสามารถเขียนในรูปสมการ ได้ดังนี้

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \ldots + \theta_q w_{t-q} \qquad (2\text{-}24)$$

เมื่อ $\{x_t ; t = 0, \pm 1, \pm 2, \ldots\}$ เป็นอนุกรมเวลาแบบคงที่ (Stationary time series)

$\phi_1, \phi_2, \ldots, \phi_p$ เป็นค่าคงที่ โดยที่ $\phi_p \neq 0$

$\theta_1, \theta_2, \ldots, \theta_q$ เป็นค่าคงที่ โดยที่ $\theta_q \neq 0$

และ w_t เป็นตัวรบกวนขาวแบบเกาส์เซียน (Gaussian white noise) ที่มีค่าเฉลี่ย 0 และความแปรปรวน σ_w^2

ถ้าอนุกรมเวลา x_t มีค่าเฉลี่ยเท่ากับ μ โดยที่ $\mu \neq 0$ เราสามารถเขียนตัวแบบ ARMA(p,q) ข้างต้นให้อยู่ในรูปสมการ ได้ดังนี้

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \ldots + \theta_q w_{t-q} \qquad (2\text{-}25)$$

เมื่อ $\alpha = \mu(1 - \phi_1 - \ldots - \phi_p)$

จากสมการที่ (2-25) เราสามารถใช้ Backshift operator เขียนตัวแบบ ARMA(p,q) ได้ดังนี้

$$\phi(B) x_t = \theta(B) w_t \qquad (2\text{-}26)$$

เมื่อ $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$ คือ Autoregressive operator

และ $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$ คือ Moving average operator

<u>ข้อสังเกต</u> : เมื่อ $q = 0$ สมการที่ (2-24) จะเป็นตัวแบบ Moving average หรือ ตัวแบบ AR(p)

เมื่อ $p = 0$ สมการที่ (2-24) จะเป็นตัวแบบ Autoregressive หรือ ตัวแบบ MA(q)

ตัวอย่าง : ตัวแบบอนุกรมเวลา ARMA(1,1) สามารถเขียนได้อยู่ในรูปแบบของสมการคือ $x_t = \phi x_{t-1} + w_t + \theta w_{t-1}$ หรือ $(1-\phi B)x_t = (1-\theta B)w_t$ โดยเงื่อนไขที่ทำให้ตัวแบบอนุกรมเวลา ARMA(1,1) เป็น Causal process คือ รากของสมการ $\phi(B) = 1-\phi B$ ต้องอยู่นอกวงกลมหนึ่งหน่วย (Outside of the unit circle) กล่าวคือ $\phi(B) = 0$ เมื่อ $|B| > 1$ เท่านั้น จะได้รากของสมการ $\phi(B) = 0$ คือ $B = \frac{1}{\phi}$ ดังนั้น $|B| > 1$ ก็ต่อเมื่อ $|\phi| < 1$ ซึ่งเป็นเงื่อนไขที่ทำให้อนุกรมเวลา x_t เป็น Causal process และเงื่อนไขที่ทำให้ตัวแบบอนุกรมเวลา ARMA(1,1) เป็น Invertible process (Infinite AR representation) คือ รากของสมการ $\theta(B) = 1-\theta B$ ต้องอยู่นอกวงกลมหนึ่งหน่วย (Outside of the unit circle) กล่าวคือ $\theta(B) = 0$ เมื่อ $|B| > 1$ เท่านั้น จะได้รากของสมการ $\theta(B) = 0$ คือ $B = \frac{1}{\theta}$ ดังนั้น $|B| > 1$ ก็ต่อเมื่อ $|\theta| < 1$ ซึ่งเป็นเงื่อนไขที่ทำให้อนุกรมเวลา x_t เป็นแบบ Invertible process

เราสามารถเขียนตัวแบบอนุกรมเวลา ARMA(1,1) ให้อยู่ในรูปกระบวนการเชิงเส้น (Linear process) โดยใช้ ψ -weight ได้ดังนี้

$$(1-\phi B)(\psi_0 + \psi_1 B + \psi_2 B^2 + \ldots) = (1-\theta B) \tag{2-27}$$

โดยวิธีการเทียบสัมประสิทธิ์ จะได้ $\psi_0 = 1, \psi_1 = -\theta + \phi$ และ $\psi_j = \phi \psi_{j-1}$ สำหรับ $j > 1$
นั่นคือ $\psi_j = (-\theta + \phi)(\phi)^{j-1}$ สำหรับ $j \geq 1$
ดังนั้น ตัวแบบอนุกรมเวลา ARMA(1,1) ที่มีรูปแบบสมการ $x_t = \phi x_{t-1} + w_t + \theta w_{t-1}$ สามารถเขียนให้อยู่ในรูปของ Causal process ได้ดังนี้

$$x_t = w_t + (-\theta + \phi)\sum_{j=1}^{\infty} \phi^{j-1} w_{t-j} \tag{2-28}$$

และเขียนให้อยู่ในรูปของ Invertible process ได้ดังนี้

$$x_t = (-\theta + \phi)\sum_{j=1}^{\infty} \theta^{j-1} x_{t-j} + w_t \tag{2-29}$$

โดยมีค่าเฉลี่ย คือ

$$E(x_t) = E\left[w_t + (-\theta + \phi)\sum_{j=1}^{\infty} \phi^{j-1} w_{t-j} \right] = 0 \tag{2-30}$$

การคำนวณหาค่าฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) สำหรับตัวแบบ ARMA(1,1) สามารถหาได้จากสมการเชิงอนุพันธ์เอกพันธ์ (Homogeneous differential equations) อันดับที่ 1 ดังนี้

$$\gamma(h) - \phi\gamma(h-1) = 0, \quad h = 2,3,\ldots \tag{2-31}$$

ภายใต้เงื่อนไขเริ่มต้น คือ $\gamma(0) = \phi\gamma(1) + \sigma_w^2\left[1 + \theta\phi + \theta^2\right]$ และ $\gamma(1) = \phi\gamma(0) + \sigma_w^2\theta$

ทำการแก้สมการหาค่า $\gamma(0)$ และ $\gamma(1)$ จากสมการข้างต้นจะได้ $\gamma(0) = \sigma_w^2 \dfrac{1 + 2\theta\phi + \theta^2}{1 - \phi^2}$ และ $\gamma(1) = \sigma_w^2 \dfrac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}$

ดังนั้น ผลเฉลยทั่วไป (General solution) ของสมการที่ (2-31) คือ

$$\gamma(h) = c\phi^h, \quad h = 1,2,\ldots \tag{2-32}$$

ทำการแก้สมการหาค่า c จากสมการที่ (2-32) จะได้ $\gamma(1) = c\phi$ นั่นคือ $c = \dfrac{\gamma(1)}{\phi}$

ดังนั้น ผลเฉลยเฉพาะ (Specific solution) ของสมการที่ (2.31) คือ

$$\gamma(h) = \frac{\gamma(1)}{\phi}\phi^h = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}\phi^{h-1}, \quad h \ge 1 \tag{2-33}$$

นำ $\gamma(0)$ หารตลอดสมการที่ (2-33) จะได้ฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) ของตัวแบบอนุกรมเวลา ARMA(1,1) คือ

$$\rho(h) = \frac{(1 + \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \theta^2}\phi^{h-1}, \quad h \ge 1 \tag{2-34}$$

<u>**หมายเหตุ**</u> : ค่าฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) และค่าฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) ของตัวแบบ ARMA มีลักษณะเหมือนกัน คือ มีค่าลดลงอย่างรวดเร็ว (Tails off)

2.1.4 ตัวแบบ Integrated autoregressive moving average หรือ ตัวแบบ ARIMA(p,d,q)

เป็นตัวแบบ Autoregressive moving average หรือ ตัวแบบ ARMA(p,q) ที่มีการหาผลต่าง (Differencing) อันดับที่ d (หรือกล่าวได้ว่าอนุกรมเวลา x_t อยู่ในรูปผลรวมอันดับที่ d (Integrated of order d) นั่นคือ $x_t \sim I(d)$ เพื่อแปลงข้อมูลอนุกรมเวลาแบบไม่คงที่ (Nonstationary time series) ให้เป็นข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) ซึ่งสามารถเขียนในรูปสมการได้ดังนี้

$$\nabla^d x_t = (1-B)^d x_t \tag{2-35}$$

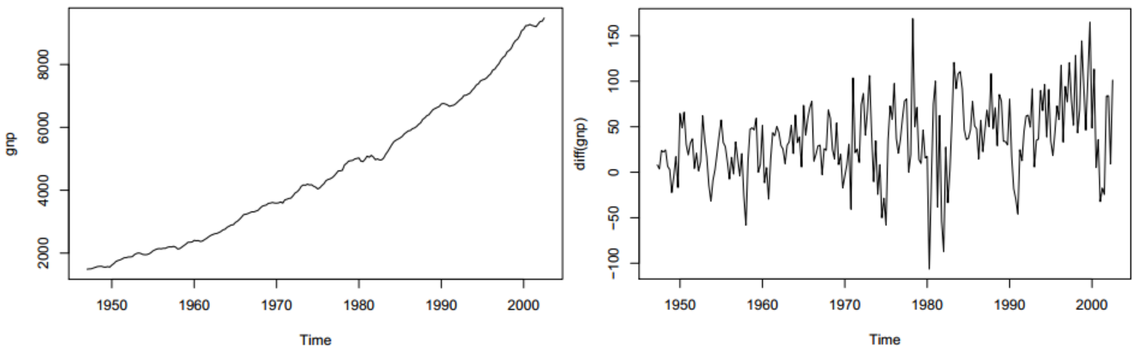
หรือสามารถเขียนสมการข้างต้นได้เป็น

$$\varphi(B)(1-B)^d x_t = \theta(B) w_t \tag{2-36}$$

ถ้าอนุกรมเวลา $\nabla^d x_t$ มีค่าเฉลี่ยเท่ากับ μ โดยที่ $\mu \neq 0$ เราสามารถเขียนตัวแบบ ARIMA(p,d,q) ข้างต้นให้อยู่ในรูปสมการ ได้ดังนี้

$$\varphi(B)(1-B)^d x_t = \delta + \theta(B) w_t \tag{2-37}$$

เมื่อ $\delta = \mu(1-\phi_1 - ... - \phi_p)$



ภาพที่ 2- 4 ข้อมูล U.S. GNP ที่มีการหาผลต่างอันดับที่ 1 เพื่อแปลงอนุกรมเวลาแบบไม่คงที่ให้เป็นอนุกรมเวลาแบบคงที่

(ที่มา: Time series analysis and its applications with R examples)

2.1.5 ขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบ ARIMA

<u>ขั้นตอนที่ 1</u> : ทำการพล็อตแผนภาพการกระจาย (Scatter diagram) ของราคาปิดหุ้น SCB เพื่อดูลักษณะของข้อมูลอนุกรมเวลาเบื้องต้น เช่น ข้อมูลอนุกรมเวลามีแนวโน้ม (Trend) มีความแปรผันตามฤดูกาล (Seasonal variation) หรือไม่ เป็นต้น

การทดสอบข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) โดยใช้การทดสอบ Unit root

ในปี ค.ศ. 1979 มีนักสถิติ 2 ท่าน คือ Dickey และ Fuller ได้เสนอวิธีการทางสถิติที่ใช้สำหรับทดสอบข้อมูลอนุกรมเวลาที่อยู่ในรูปแบบ ARMA(p,q) ว่าเป็นแบบคงที่หรือไม่ ซึ่งเราสามารถนำมาใช้ทดสอบว่า อันดับที่ควรทำผลต่าง (d) ควรเป็นที่เท่าใดจึงจะได้ข้อมูลอนุกรมเวลาแบบคงที่ ซึ่งสามารถแบ่งออกได้ 2 วิธี ดังนี้

<u>วิธีที่ 1</u> : Dickey-fuller

ใช้สำหรับทดสอบข้อมูลอนุกรมเวลาที่อยู่ในรูปแบบ ARMA(p,q) ว่ามีแนวโน้มแบบสุ่มหรือไม่ โดยแบ่งสมการที่ใช้ทดสอบ Unit root ดังนี้

1. $x_t = \phi x_{t-1} + w_t$ (Random walk)
2. $x_t = \alpha + \phi x_{t-1} + w_t$ (Random walk with drift)
3. $x_t = \alpha + \beta t + \phi x_{t-1} + w_t$ (Random walk with drift and linear deterministic trend)

จะเห็นว่าสมการที่ 3 มีตัวแปร Deterministic trend (t) และค่าคงที่ (α) มาร่วมในการทดสอบ Unit root ด้วย ส่วนสมการที่ 2 มีเฉพาะค่าคงที่เท่านั้น และสมการที่ 1 ไม่มีทั้งค่าคงที่และ Deterministic trend การเลือกใช้สมการที่ 1-3 มีหลักเกณฑ์ดังนี้

เมื่อเราพล็อตกราฟของข้อมูลอนุกรมเวลาที่ต้องการทดสอบว่าเป็นแบบคงที่หรือไม่ แล้วพบว่าข้อมูลอนุกรมเวลานั้นมีลักษณะขึ้น ๆ ลง ๆ อยู่รอบ ๆ จุดกำเนิด เราควรเลือกใช้สมการที่ 1 และหากพบว่าข้อมูลอนุกรมเวลาไม่มีแนวโน้มที่เพิ่มขึ้นหรือลดลงเมื่อเวลาผ่านไป แต่มีลักษณะขึ้น ๆ

ลง ๆ อยู่รอบ ๆ ค่าคงที่ค่าหนึ่ง เราควรเลือกใช้สมการที่ 2 และหากพบว่าข้อมูลอนุกรมเวลานั้นมีแนวโน้มที่เพิ่มขึ้นหรือลดลงเมื่อเวลาผ่านไป เราควรเลือกใช้สมการที่ 3

เมื่อเรานำ x_{t-1} ไปลบออกทั้งสองข้างของสมการที่ 1-3 เราจะได้สมการดังต่อไปนี้

4. $\Delta x_t = \gamma x_{t-1} + w_t$ (Random walk)

5. $\Delta x_t = \alpha + \gamma x_{t-1} + w_t$ (Random walk with drift)

6. $\Delta x_t = \alpha + \beta t + \gamma x_{t-1} + w_t$ (Random walk with drift & linear deterministic trend)

โดยที่ $\gamma = \phi - 1$

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \gamma = 0$

$H_a : \gamma < 0$

ซึ่งเทียบเท่ากับ $H_0 : \phi = 1$

$H_a : \phi < 1$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $t_c = \dfrac{\hat{\gamma}}{se(\hat{\gamma})}$ หรือ $t_c = \dfrac{\hat{\phi} - 1}{se(\hat{\phi})}$

<u>หมายเหตุ</u> : ถ้าผลการทดสอบสมมุติฐาน คือ ปฏิเสธ H_0 แสดงว่า ข้อมูลอนุกรมเวลาเป็นแบบคงที่ (Stationary time series) ที่ระดับนัยสำคัญ 0.05

<u>วิธีที่ 2</u> : Augmented dickey-fuller (ADF)

เป็นวิธีการทดสอบ Unit root อีกวิธีหนึ่งซึ่งพัฒนามาจากวิธี Dickey-fuller ใช้สำหรับทดสอบข้อมูลอนุกรมเวลาที่อยู่ในรูปแบบ ARMA(p,q) เพื่อให้ตัวแปรสุ่มคลาดเคลื่อน (w_t) ของตัวแบบที่ใช้ทดสอบ Unit root ในกรณีที่มีปัญหาสหสัมพันธ์เชิงอนุกรม (Serial correlation) ในระดับที่สูงมากมีคุณสมบัติเป็นตัวรบกวนขาว โดยแบ่งสมการที่ใช้ทดสอบ Unit root ดังนี้

1. $\Delta x_t = \gamma x_{t-1} + \sum_{i=1}^{p-1} c_i \Delta x_{t-i} + w_t$ (Random walk)

2. $\Delta x_t = \alpha + \gamma x_{t-1} + \sum_{i=1}^{p-1} c_i \Delta x_{t-i} + w_t$ (Random walk with drift)

3. $\Delta x_t = \alpha + \beta t + \gamma x_{t-1} + \sum_{i=1}^{p-1} c_i \Delta x_{t-i} + w_t$ (Random walk with drift & linear time trend)

โดยที่ $\gamma = (\phi_1 + \phi_2 + \ldots + \phi_p - 1)$ และ p คือช่วงเวลาที่ช้ากว่ากัน (Lag) ที่ใช้ในสมการข้างต้น

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \gamma = 0$

$H_a : \gamma < 0$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $t_c = \dfrac{\hat{\gamma}}{se(\hat{\gamma})}$

หมายเหตุ : ถ้าผลการทดสอบสมมุติฐาน คือ ปฏิเสธ H_0 แสดงว่า ข้อมูลอนุกรมเวลาเป็นแบบคงที่ (Stationary time series) ที่ระดับนัยสำคัญ 0.05

วิธีการหาผลต่าง (Differencing)

จากที่เราทราบมาแล้วว่าตัวแบบของ Box และ Jenkins จะต้องนำไปใช้กับข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) เท่านั้น ดังนั้นหากเราพบว่าอนุกรมเวลาเป็นแบบไม่คงที่ (Nonstationary time series) เราจะต้องแปลงข้อมูลอนุกรมเวลานั้นให้เป็นแบบคงที่เสียก่อน จึงจะนำมาใช้กับตัวแบบของ Box และ Jenkins ได้ และวิธีการหนึ่งที่มักถูกนำมาใช้แปลงข้อมูลอนุกรมเวลาแบบไม่คงที่ให้เป็นข้อมูลอนุกรมเวลาแบบคงที่ก็คือ วิธีการหาผลต่าง (Differencing)

ตัวอย่าง : พิจารณาอนุกรมเวลาที่อยู่ในรูปแบบ Random walk with drift กล่าวคือ $x_t = \alpha + x_{t-1} + w_t$ พบว่าอนุกรมเวลา x_t เป็นแบบไม่คงที่ เนื่องจากค่าเฉลี่ยในแต่ละช่วงเวลา t ไม่คงที่ จึงทำการแปลงอนุกรมเวลา x_t โดยการหาผลต่างอันดับที่ 1 จะทำให้ได้อนุกรมเวลาแบบคงที่ ซึ่งแสดงได้ดังนี้

$$\nabla x_t = x_t - x_{t-1} = \alpha + w_t \text{ หรือ } (1-B)x_t = \alpha + w_t \quad (2\text{-}38)$$

ในกรณีนี้เราจะกล่าวว่า อนุกรมเวลา x_t อยู่ในรูปผลรวมอันดับที่ 1 (Integrated of order 1) หรือแทนด้วย $x_t \sim I(1)$

<u>หมายเหตุ</u> : หลังจากทำการหาผลต่างอันดับที่ 1 จะต้องนำข้อมูลอนุกรมเวลาชุดใหม่นั้นมาทดสอบ Unit root อีกครั้ง ว่าข้อมูลอนุกรมเวลาชุดใหม่นั้นเป็นแบบคงที่หรือไม่ ถ้าเป็นแบบไม่คงที่อยู่ให้ทำการหาผลต่างอีกครั้ง แล้วจึงนำข้อมูลอนุกรมเวลาชุดใหม่ที่ได้มาทดสอบ Unit root ต่อไป

<u>ขั้นตอนที่ 2</u> : การระบุตัวแบบ (Model identification) เป็นขั้นตอนของการตรวจสอบค่าสหสัมพันธ์ของข้อมูลอนุกรมเวลาที่กำลังพิจารณาว่ามีความสัมพันธ์ในตัวเชิงเส้นตรงกับช่วงเวลาอื่น ๆ หรือไม่ การตรวจสอบนี้สามารถทำได้โดยการคำนวณค่าฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) จากข้อมูลอนุกรมเวลาที่เก็บรวบรวมมา แล้วนำมาใช้ตัดสินใจเบื้องต้นว่าควรเลือกใช้ตัวแบบของ Box และ Jenkins ชนิดใด เช่น ควรเลือกตัวแบบ AR หรือตัวแบบ MA หรือตัวแบบ ARMA

ฟังก์ชันสหสัมพันธ์ในตัว (Autocorrelation function: ACF) แทนด้วยสัญลักษณ์ $\rho(h)$ เป็นฟังก์ชันที่ให้ค่าสหสัมพันธ์ในตัวของค่าสังเกตที่ช้ากว่ากัน h ช่วงเวลาในอนุกรมเวลาแบบคงที่ (Stationary time series) โดยอนุกรมเวลาอาจเป็นแบบต่อเนื่องหรือไม่ต่อเนื่องในเวลา ซึ่งคำนวณได้จากสูตร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$\rho(h) = \frac{\gamma(t+h,t)}{\sqrt{\gamma(t+h,t+h)\gamma(t,t)}} = \frac{\gamma(h)}{\gamma(0)} \qquad (2\text{-}39)$$

โดยอสมการของ Cauchy-Schwarz จะได้ว่า $-1 \le \rho(h) \le 1$

การทดสอบ Ljung-box-pierce Q-statistics

เป็นการทดสอบว่าข้อมูลอนุกรมเวลาเป็นอิสระกับช่วงเวลาอื่น ๆ หรือไม่ สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \rho(1) = \rho(2) = \ldots = \rho(H) = 0$

H_a : มี $\rho(k) \neq 0$ อย่างน้อย 1 ค่า สำหรับ $k = 1,2,\ldots,H$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $Q = n(n+2)\sum_{k=1}^{H} \frac{\hat{\rho}^2(k)}{n-k}$

เมื่อ n แทน ขนาดของอนุกรมเวลา และ H แทน ช่วงเวลาห่างสูงสุดระหว่างค่าสังเกต x_t

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $Q \geq \chi^2_{\alpha,H}$

หมายเหตุ : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า ข้อมูลอนุกรมเวลา ณ ช่วงเวลาปัจจุบันมีความสัมพันธ์กับข้อมูลอนุกรมเวลาในอดีต H ช่วงเวลาที่ผ่านมา ที่ระดับนัยสำคัญ 0.05 จากนั้นทำการพิจารณาที่ค่า H อื่น ๆ ถ้าให้ข้อสรุปเดียวกัน เราสามารถสรุปได้ว่า ข้อมูลอนุกรมเวลาไม่เป็นอิสระกับช่วงเวลาอื่น ๆ อย่างมีนัยสำคัญ

การทดสอบความมีนัยสำคัญของค่าสหสัมพันธ์ในตัวจากตัวอย่าง

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \rho(h) = 0$

$H_a : \rho(h) \neq 0$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $Z_c = \frac{\hat{\rho}(h)}{1/\sqrt{n}}$

เมื่อ n แทน ขนาดของอนุกรมเวลา

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $|Z_c| \geq Z_{\frac{\alpha}{2}}$

หมายเหตุ : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า ข้อมูลอนุกรมเวลา ณ เวลา $t+h$ มีความสัมพันธ์เชิงเส้นตรงกับข้อมูลอนุกรมเวลา ณ เวลา t ที่ระดับนัยสำคัญ 0.05

ฟังก์ชันสหสัมพันธ์บางส่วน (Partial autocorrelation function: PACF) ของอนุกรมเวลาแบบคงที่ (Stationary time series) แทนด้วยสัญลักษณ์ ϕ_{hh} สำหรับ $h = 1,2,...$ ซึ่งคำนวณได้จากสูตร

$$\phi_{11} = corr(x_1, x_0) = \rho(1) \text{ และ } \phi_{hh} = corr(x_h - x_h^{h-1}, x_0 - x_0^{h-1}),\ h \geq 2 \qquad (2\text{-}40)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การทดสอบความมีนัยสำคัญของค่าสหสัมพันธ์บางส่วนจากตัวอย่าง

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \phi_{hh} = 0$

$H_a : \phi_{hh} \neq 0$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $Z_c = \dfrac{\hat{\phi}_{hh}}{1/\sqrt{n}}$

เมื่อ n แทน ขนาดของอนุกรมเวลา

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $|Z_c| \geq Z_{\frac{\alpha}{2}}$

<u>หมายเหตุ</u> : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า ข้อมูลอนุกรมเวลา ณ เวลา $t+h$ มีความสัมพันธ์เชิงเส้นตรงกับข้อมูลอนุกรมเวลา ณ เวลา t ที่ระดับนัยสำคัญ 0.05 เมื่อไม่มีอิทธิพลของข้อมูลอนุกรมเวลา ณ เวลา $t+1, t+2, ..., t+h-1$ เข้ามาเกี่ยวข้อง

จากการพิจารณาค่าฟังก์ชันสหสัมพันธ์ในตัวและค่าฟังก์ชันสหสัมพันธ์บางส่วนสำหรับคัดเลือกตัวแบบที่น่าจะเหมาะสมที่สุดสำหรับข้อมูลอนุกรมเวลาที่กำลังพิจารณา สามารถสรุปได้ดังนี้

ลักษณะของฟังก์ชันสหสัมพันธ์ในตัวและฟังก์ชันสหสัมพันธ์บางส่วนสำหรับตัวแบบ ARMA

	ตัวแบบ AR(p)	ตัวแบบ MA(q)	ตัวแบบ ARMA(p,q)
ฟังก์ชันสหสัมพันธ์ในตัว (ACF)	มีค่าลดลงอย่างรวดเร็ว	สิ้นสุดหลังจาก q ช่วงเวลาที่แล้ว	มีค่าลดลงอย่างรวดเร็ว
ฟังก์ชันสหสัมพันธ์บางส่วน (PACF)	สิ้นสุดหลังจาก p ช่วงเวลาที่แล้ว	มีค่าลดลงอย่างรวดเร็ว	มีค่าลดลงอย่างรวดเร็ว

ตารางที่ 2- 1 ลักษณะของฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) สำหรับตัวแบบ ARMA

<u>ขั้นตอนที่ 3</u> : การประมาณค่าพารามิเตอร์ (Parameter estimation)

จากขั้นตอนที่ 2 เราทราบแล้วว่าควรจะเลือกตัวแบบ Box และ Jenkins ชนิดใด และทราบถึงวิธีการคาดเดาว่าควรใช้ช่วงเวลาที่ช้ากว่ากัน (Lag) เท่าใดกับตัวแบบชนิดนั้น ในขั้นตอนนี้จะทำการประมาณค่าพารามิเตอร์ของตัวแบบ ARIMA(p,q) โดยมีรูปแบบของสมการคือ

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \ldots + \theta_q w_{t-q} \qquad (2\text{-}41)$$

โดยมีวิธีการประมาณค่าพารามิเตอร์ 4 วิธี ดังนี้

1. วิธีกำลังสองน้อยสุดแบบมีเงื่อนไข (Conditional least square estimation)

การประมาณค่าพารามิเตอร์ด้วยวิธีนี้จะเริ่มจากการกำหนดค่าเริ่มต้น (Initial value) ของค่าส่วนเหลือขึ้นมาก่อน โดยกำหนดให้ $w_0, w_{-1}, w_{-2}, \ldots, w_{1-q} = 0$ ซึ่งในการคำนวณจะมีการทำซ้ำ (Iterations) หลายรอบจนกว่าจะได้ค่าประมาณที่คงที่และทำให้ค่าผลบวกกำลังสองของค่าคลาดเคลื่อน (SSE) มีค่าต่ำที่สุด

2. วิธีกำลังสองน้อยสุดแบบไม่มีเงื่อนไข (Unconditional least square estimation)

การประมาณค่าพารามิเตอร์ด้วยวิธีนี้จะเริ่มจากการนำค่าพยากรณ์ถอยหลัง (Back-casting) ในค่าเริ่มแรกของค่าส่วนเหลือ หมายถึง $\hat{w}_0, \hat{w}_{-1}, \hat{w}_{-2}, \ldots, \hat{w}_{1-q}$ มาใช้ในการประมาณค่าพารามิเตอร์ที่ทำให้ค่าผลบวกกำลังสองของค่าคลาดเคลื่อน (SSE) มีค่าต่ำที่สุด

3. วิธีภาวะน่าจะเป็นสูงสุดแบบมีเงื่อนไข (Conditional maximum likelihood estimation)

การประมาณค่าพารามิเตอร์ด้วยวิธีนี้จะเริ่มจากการกำหนดค่าเริ่มต้น (Initial value) ของค่าส่วนเหลือขึ้นมาก่อน โดยกำหนดให้ $w_0, w_{-1}, w_{-2}, \ldots, w_{1-q} = 0$ แต่วิธีภาวะน่าจะเป็นสูงสุดต้องการข้อสมมติเกี่ยวกับลักษณะการแจกแจงแบบปรกติของส่วนเหลือ โดยที่ค่าประมาณที่ได้จากวิธีภาวะน่าจะเป็นสูงสุดจะเป็นค่าประมาณที่ทำให้ฟังก์ชันภาวะน่าจะเป็นมีค่าสูงที่สุด

4. วิธีภาวะน่าจะเป็นสูงสุดแบบไม่มีเงื่อนไข (Unconditional maximum likelihood estimation)

การประมาณค่าพารามิเตอร์ด้วยวิธีนี้จะเริ่มจากการนำค่าพยากรณ์ถอยหลัง (Back-casting) ในค่าเริ่มแรกของค่าส่วนเหลือ หมายถึง $\hat{w}_0, \hat{w}_{-1}, \hat{w}_{-2}, \ldots, \hat{w}_{1-q}$ มาใช้ในการประมาณค่าพารามิเตอร์ที่ทำให้ฟังก์ชันภาวะน่าจะเป็นมีค่าสูงที่สุด

<u>ขั้นตอนที่ 4</u> : การตรวจสอบความเหมาะสมของตัวแบบ (Diagnostic checking)

หลังจากระบุตัวแบบและประมาณค่าพารามิเตอร์ในตัวแบบแล้ว ต้องตรวจสอบความเหมาะสมของตัวแบบที่กำหนดจากค่าส่วนเหลือ (Residuals) แทนด้วย e_t โดยที่

$$e_t = \left(x_t - \hat{x}_t^{t-1}\right) \Big/ \sqrt{\hat{P}_t^{t-1}} \tag{2-42}$$

เมื่อ $\hat{x}_t^{t-1}$ แทน ค่าพยากรณ์ 1 ช่วงเวลาล่วงหน้า (One-step-ahead prediction)
และ $\hat{P}_t^{t-1}$ แทน ความแปรปรวนของค่าคลาดเคลื่อนของการพยากรณ์ 1 ช่วงเวลาล่วงหน้า (Estimated one-step-ahead error variance)

ซึ่งเกณฑ์ที่ใช้พิจารณาก็คือ w_t ในตัวแบบที่เลือกต้องมีคุณสมบัติเป็นตัวรบกวนขาวแบบเกาส์เซียนที่มีค่าเฉลี่ยเป็นศูนย์ ความแปรปรวนคงที่ และเป็นอิสระกับช่วงเวลาอื่น ๆ (ไม่มีความสัมพันธ์กันเอง)

การทดสอบว่าส่วนเหลือมีค่าเฉลี่ยเป็นศูนย์และความแปรปรวนคงที่

พิจารณาได้จากการนำส่วนเหลือ (e_t) มาพล็อตแผนภาพการกระจาย (Scatter plot) ที่มีแกนตั้งเป็นส่วนเหลือ (e_t) และแกนนอนเป็นช่วงเวลา (t) ซึ่งกราฟต้องมีลักษณะการกระจายแบบสุ่มรอบ ๆ ศูนย์และมีภาวะความแปรปรวนเท่ากัน (Homoscedasticity)

การทดสอบความเป็นอิสระของส่วนเหลือ (Tests of independence) โดยใช้ Ljung-box-pierce Q-statistics

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : \rho_e(1) = \rho_e(2) = ... = \rho_e(H) = 0$

H_a : มี $\rho_e(h) \neq 0$ อย่างน้อย 1 ค่า สำหรับ $h = 1,2,...,H$

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $$Q = n(n+2)\sum_{h=1}^{H}\frac{\hat{\rho}_e^2(h)}{n-h}$$

เมื่อ n แทน ขนาดของอนุกรมเวลา และ h แทน ช่วงเวลาห่างสูงสุดระหว่างค่าสังเกต x_t

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $Q \geq \chi^2_{\alpha,H-p-q}$

<u>หมายเหตุ</u> : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า w_t เป็นอิสระกับช่วงเวลาอื่น ๆ ที่ระดับนัยสำคัญ 0.05

การทดสอบภาวะปรกติของส่วนเหลือ (Tests of normality)

พิจารณาได้จากฮิสโทแกรมของส่วนเหลือ ต้องมีลักษณะใกล้เคียงกับเส้นโค้งปรกติ และกราฟ Q-Q ต้องมีลักษณะการกระจายอยู่ในแนวเส้นตรง ถ้ามีจุดของค่าสังเกตกระจายห่างจากเส้นตรงมากเพียงใด แสดงถึงการแจกแจงไม่เป็นปรกติมากเท่านั้น หรือสามารถทดสอบสมมุติฐานโดยใช้การทดสอบคอลโมโกรอฟ-สมีร์นอฟ ดังนี้

สมมุติฐานการทดสอบ $H_0 : w_t$ มีการแจกแจงแบบปรกติ

$H_a : w_t$ ไม่ได้มีการแจกแจงแบบปรกติ

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $D = max\left|F(x_t) - S(x_t)\right|$

เมื่อ $F(x_t)$ และ $S(x_t)$ แทน ฟังก์ชันความน่าจะเป็นสะสมของตัวอย่าง x_t

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $D >$ ค่าวิกฤตที่ได้จากตารางคอลโมโกรอฟ-สมีร์นอฟ

<u>หมายเหตุ</u> : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า w_t มีการแจกแจงแบบปรกติที่ระดับนัยสำคัญ 0.05

การทดสอบความสุ่มของส่วนเหลือ (Tests of randomness) โดยใช้การทดสอบแบบรันส์ (Runs tests)

สามารถทำได้โดยการทดสอบสมมุติฐาน ดังต่อไปนี้

สมมุติฐานการทดสอบ $H_0 : w_t$ เป็นไปอย่างสุ่ม

$H_a : w_t$ ไม่เป็นไปอย่างสุ่ม

กำหนดระดับนัยสำคัญ $\alpha = 0.05$

ตัวสถิติทดสอบ $$Z_c = \frac{R - \left(\frac{2n_1n_2}{n} + 1\right)}{\sqrt{\frac{2n_1n_2(2n_1n_2 - n)}{n^2(n-1)}}}$$

เมื่อ $R =$ จำนวนรันส์ (Runs) = อัตราการเปลี่ยนแปลงค่าของตัวแปร โดยตัวแปรในที่นี้มีค่าได้ 2 ค่า คือเป็น A ถ้าข้อมูลตัวอย่างมากกว่าหรือเท่ากับค่ามัธยฐาน และจะมีค่าเป็น B ถ้าข้อมูลตัวอย่างมีค่าน้อยกว่าค่ามัธยฐาน ดังนั้น R จึงหมายถึง จำนวนครั้งที่มีการเปลี่ยนจาก A เป็น B หรือ B เป็น A โดยที่ n_1 คือ จำนวนค่าของ A, n_2 คือ จำนวนค่าของ B และ $n = n_1 + n_2 =$ ขนาดของอนุกรมเวลา

เกณฑ์การตัดสินใจ จะปฏิเสธ H_0 ถ้า $Z_c \geq Z_{1-\frac{\alpha}{2}}$

หมายเหตุ : ถ้าผลการทดสอบสมมุติฐาน คือ ไม่ปฏิเสธ H_0 แสดงว่า w_t เป็นไปอย่างสุ่ม ที่ระดับนัยสำคัญ 0.05

หลังจากที่ทำการตรวจสอบคุณสมบัติของ w_t หากพบว่า w_t ไม่มีคุณสมบัติเป็นตัวรบกวนขาวแบบเกาส์เซียน เราจะต้องกลับไปเริ่มทำขั้นตอนที่ 2 และ 3 ใหม่ แล้วทำการตรวจสอบความเหมาะสมของตัวแบบใหม่อีกครั้ง หากพบว่า w_t ไม่มีคุณสมบัติเป็นตัวรบกวนขาวแบบเกาส์เซียนอีกก็ต้องกลับไปทำขั้นตอนที่ 2 และ 3 ใหม่ ทำเช่นนี้ไปเรื่อย ๆ จนกว่า w_t ในตัวแบบสุดท้ายที่ประมาณขึ้นมีคุณสมบัติเป็นตัวรบกวนขาวแบบเกาส์เซียน

ในทางปฏิบัติอาจพบว่า มีมากกว่า 1 ตัวแบบที่เหมาะสมกับข้อมูลอนุกรมเวลา กรณีนี้จะทำการเลือกเพียงหนึ่งตัวแบบที่จะนำไปสร้างสมการสำหรับพยากรณ์และหาค่าพยากรณ์ การพิจารณา

ความเหมาะสมของตัวแบบ ARMA(p,q) เพียงหนึ่งตัวแบบมักจะใช้ค่าสถิติเป็นเกณฑ์ในการพิจารณา ได้แก่

1. เกณฑ์สารสนเทศของอะกะอิเกะ (Akaike's information criterion: AIC)

เป็นเกณ์การคัดเลือกตัวแบบที่เหมาะสมภายใต้ตัวแบบที่มีการใช้ข้อมูลชุดเดียวกันแต่มีจำนวนพารามิเตอร์แตกต่างกันที่ใช้อธิบายความสัมพันธ์ระหว่างตัวแปรตอบสนองกับตัวแปรอธิบายชุดหนึ่ง ตัวแบบที่เหมาะสมกว่า คือตัวแบบที่มีค่า AIC ต่ำกว่า มักใช้ในการวิเคราะห์การถดถอยเชิงเส้นและการวิเคราะห์อนุกรมเวลา

กรณีที่มีการประมาณค่าพารามิเตอร์ของตัวแบบด้วยวิธีกำลังสองน้อยสุด มีสูตรในการคำนวณดังนี้

$$AIC = -2\log\left(\frac{SSE}{n}\right) + 2p \qquad (2\text{-}43)$$

กรณีที่มีการประมาณค่าพารามิเตอร์ของตัวแบบด้วยวิธีภาวะน่าจะเป็นสูงสุด มีสูตรในการคำนวณดังนี้

$$AIC = -2\log L + 2p \qquad (2\text{-}44)$$

เมื่อ p แทน จำนวนพารามิเตอร์ในตัวแบบ

n แทน จำนวนข้อมูลที่นำมาพิจารณา

SSE แทน ผลบวกกำลังสองของค่าคลาดเคลื่อน

และ L แทน ฟังก์ชันภาวะน่าจะเป็นสูงสุด

2. เกณฑ์สารสนเทศของเบส์ (Bayesian information criterion: BIC)

เป็นเกณฑ์การคัดเลือกตัวแบบที่เหมาะสมอีกเกณฑ์หนึ่งที่ปรับจากตัวสถิติ AIC โดยนำขนาดตัวอย่างมาพิจารณาด้วย ตัวแบบที่เหมาะสมกว่า คือตัวแบบที่มีค่า BIC ต่ำกว่า มีสูตรในการคำนวณดังนี้

$$BIC = n\log\left(\frac{MSE}{n}\right) + \frac{2(p+2)n\sigma^2}{MSE} - \frac{2n^2\sigma^4}{MSE} \quad (2\text{-}45)$$

เมื่อ p แทน จำนวนพารามิเตอร์ในตัวแบบ

n แทน จำนวนข้อมูลที่นำมาพิจารณา

MSE แทน ค่าคลาดเคลื่อนกำลังสองเฉลี่ยของตัวแบบเต็ม (Full model)

และ σ แทน ค่าเบี่ยงเบนมาตรฐานของค่าคลาดเคลื่อนในตัวแบบ

ในกรณีที่ตัวแปรตอบสนองเป็นตัวแปรจำแนกประเภท ค่า BIC แสดงได้ด้วย $G^2 - v\log n$ เมื่อ G^2 คือ ตัวสถิติอัตราส่วนภาวะน่าจะเป็นสำหรับตัวแบบที่มีองศาเสรี v สูตรนี้อาจเรียกว่า Schwarz criterion (SC)

3. เอสบีซี (Schwarz-Bayesian criterion: SBC)

เป็นเกณฑ์การคัดเลือกตัวแบบที่เหมาะสมอีกเกณฑ์หนึ่งที่ปรับจากตัวสถิติ AIC โดยนำขนาดตัวอย่างมาพิจารณาด้วย ตัวแบบที่เหมาะสมกว่า คือตัวแบบที่มีค่า SCB ต่ำกว่า มีสูตรในการคำนวณดังนี้

$$SBC = -2\log L + p\log n \quad (2\text{-}46)$$

เมื่อ p แทน จำนวนพารามิเตอร์ในตัวแบบ

n แทน จำนวนข้อมูลที่นำมาพิจารณา

L แทน ฟังก์ชันภาวะน่าจะเป็นสูงสุด

<u>ขั้นตอนที่ 5</u> : การพยากรณ์ (Forecasting)

เมื่อตรวจสอบความเหมาะสมของตัวแบบแล้วพบว่าตัวแบบที่กำหนดเหมาะสมกับข้อมูลอนุกรมเวลา ขั้นตอนต่อไปซึ่งเป็นขั้นตอนสุดท้ายของวิธีของ Box และ Jenkins เป็นการนำสมการพยากรณ์ที่สร้างจากตัวแบบดังกล่าวไปประมาณค่า ณ เวลาใดเวลาหนึ่งในอนาคตโดยใช้ชุดข้อมูลทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กำหนดเซตของข้อมูลอนุกรมเวลา $x=\{x_n , x_{n-1} ,..., x_1\}$ โดยที่ x_t เป็นข้อมูลอนุกรมเวลาแบบคงที่และทราบค่าพารามิเตอร์ของตัวแบบ ค่าพยากรณ์ของข้อมูลอนุกรมเวลาล่วงหน้าไป m ช่วงเวลา (m-step ahead forecast) คำนวณได้จาก

$$x_{n+m}^{n} = E(x_{n+m} \mid x) = E(x_{n+m} \mid x_n , x_{n-1} ,..., x_1) \qquad (2\text{-}47)$$

ซึ่งการพยากรณ์ตามสมการข้างต้น จะทำให้ค่าคลาดเคลื่อนกำลังสองเฉลี่ยจากการพยากรณ์ (Minimum mean square error predictor) มีค่าน้อยที่สุด นั่นคือ ทำให้ $E(x_{n+m}-g(x))^2$ มีค่าน้อยที่สุด เมื่อ $g(x)$ แทน ฟังก์ชันค่าสังเกตของอนุกรมเวลา

พิจารณาการพยากรณ์ข้อมูลอนุกรมด้วยตัวแบบ ARIMA(p,1,q)

สมมติให้ตัวแบบ ARIMA(p,1,q) เขียนสมการได้ดังนี้

$$\Delta x_t = \alpha + \phi_1 \Delta x_{t-1} + \phi_2 \Delta x_{t-2} + ... + \phi_p \Delta x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

$$x_t - x_{t-1} = \alpha + \phi_1 (x_{t-1} - x_{t-2}) + ... + \phi_p (x_{t-p} - x_{t-p-1}) + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

$$x_t = \alpha + (\phi_1 + 1) x_{t-1} + ... + (\phi_p - \phi_{p-1}) x_{t-p} - \phi_p x_{t-p-1} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q}$$

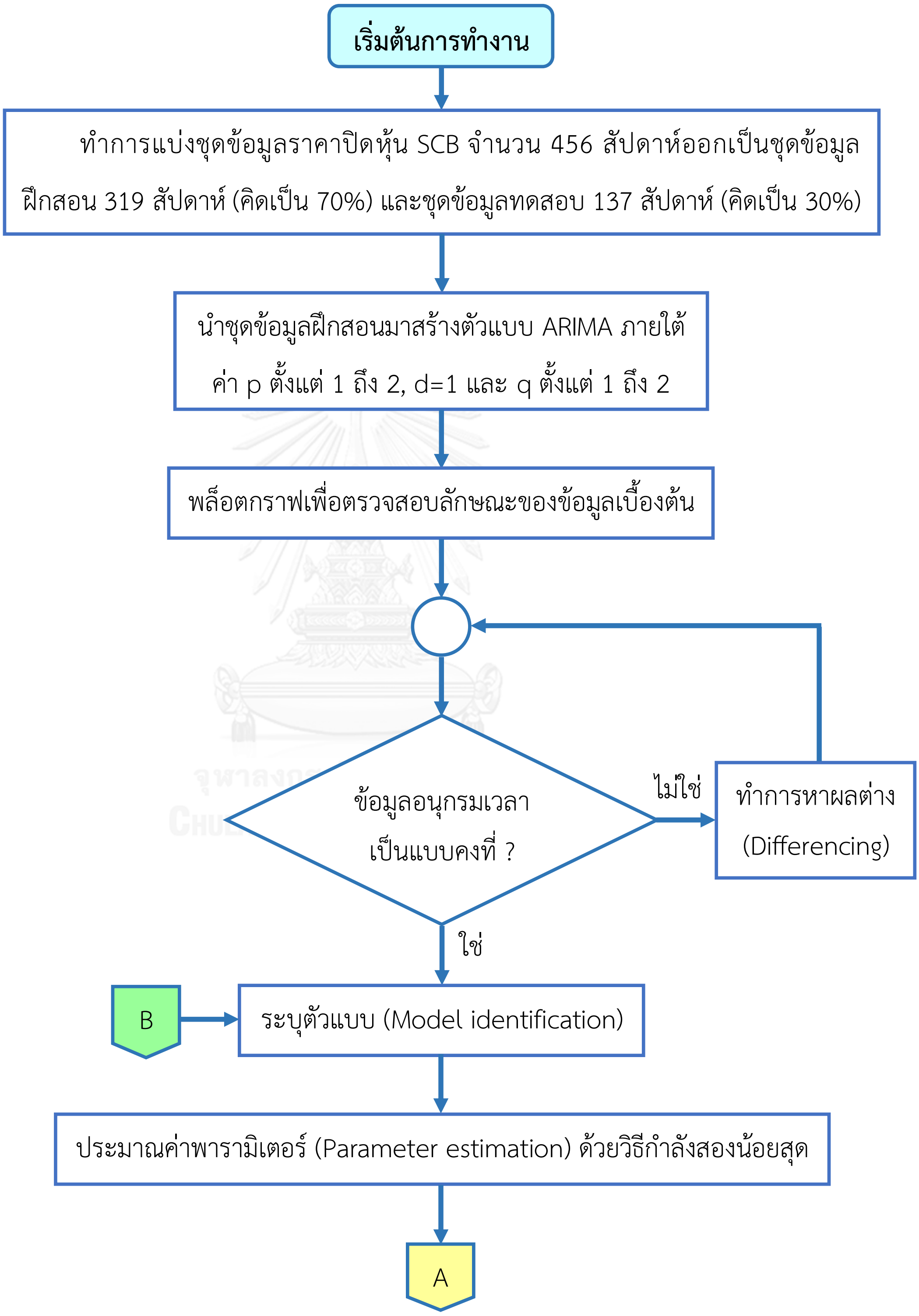
หรือเขียนใหม่ได้ว่า

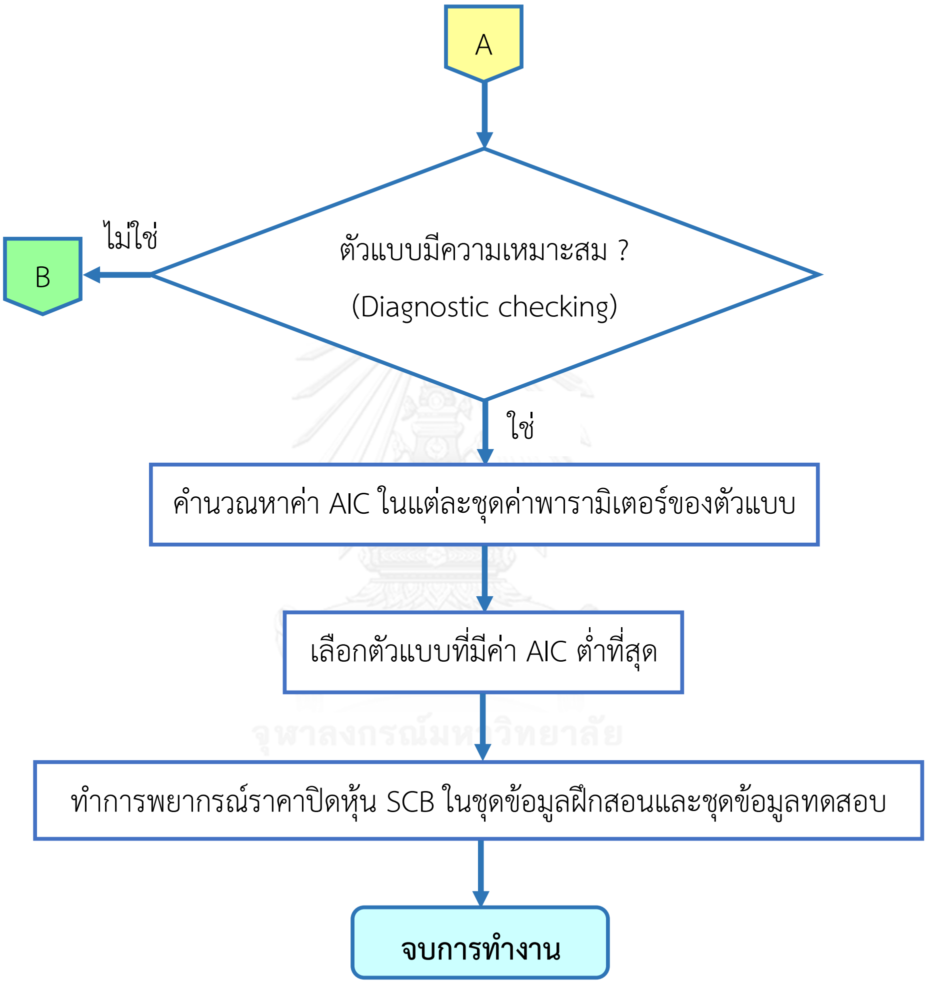
$$x_t = \alpha_0 + \alpha_1 x_{t-1} + ... + \alpha_p x_{t-p} + \alpha_{p+1} x_{t-p-1} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + \theta_q w_{t-q} \qquad (2\text{-}48)$$

เมื่อ $\alpha_0 = \alpha, \alpha_1 = \phi_1 + 1, \alpha_j = \phi_j - \phi_{j-1}$ สำหรับ $j = 2,..., p$ และ $\alpha_{p+1} = -\phi_p$

จากนั้นใช้สมการข้างต้นในการพยากรณ์ข้อมูลอนุกรมเวลาล่วงหน้าไป m ช่วงเวลา

2.1.6 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบ ARIMA



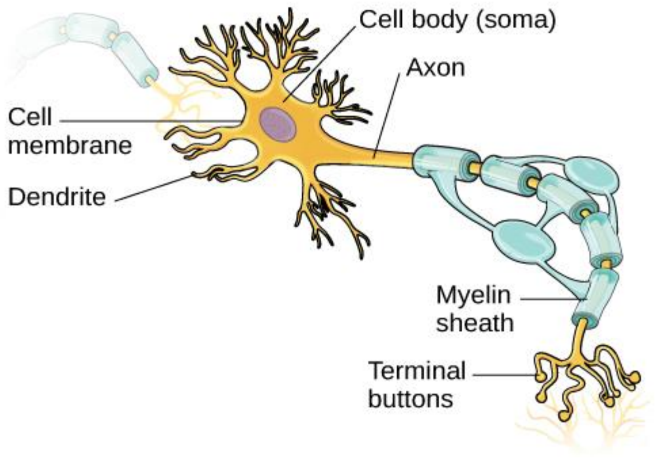


จุฬาลงกรณ์มหาวิทยาลัย

2.2 ตัวแบบเครือข่ายประสาทเทียม (Artificial neural network model: ANN)

ตัวแบบเครือข่ายประสาทเทียม เป็นตัวแบบเชิงคณิตศาสตร์ที่มีที่มาจากงานวิจัยด้านปัญญาประดิษฐ์ (Artificial intelligence: AI) ซึ่งถูกคิดค้นขึ้นในปี ค.ศ. 1991 โดย Mc Culloch และ Pitts เพื่อลอกเลียนแบบความสามารถของสมองมนุษย์ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือที่มีความสามารถในการเรียนรู้ที่ซับซ้อน ได้แก่ การจดจำรูปแบบ (Pattern recognition) และการอนุมานความรู้ (Knowledge deduction) ซึ่งนิยมใช้ในการพยากรณ์ด้านธุรกิจโดยเฉพาะการเงินเพราะสามารถพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ได้ดี เครือข่ายประสาทเทียมประกอบด้วยชุดของเซลล์ประสาท (Neural) ที่เชื่อมต่อกันและจุดประสานประสาท (Synapses) โดยโครงสร้างของการส่งสัญญาณประสาทประกอบจากการเชื่อมต่อระหว่างเซลล์ประสาทหลายพันล้านเซลล์ เซลล์ประสาทแต่ละเซลล์ประกอบด้วยแขนงรับสัญญาณประสาทซึ่งเป็นเสมือนหน่วยรับข้อมูลเข้า เรียกว่า เดนไดรท์ (Dendrites) และส่วนปลายของเซลล์ประสาทในการส่งสัญญาณประสาทซึ่งเสมือนหน่วยส่งข้อมูลออกของเซลล์ เรียกว่า แอคซอน (Axons) โดยการส่งสัญญาณประสาทดังกล่าวอาจทำให้เกิดได้ทั้งการกระตุ้นและยับยั้ง ทั้งนี้นอกจากลักษณะดังกล่าวแล้ว วิธีการประมวลผลภายในเซลล์ประสาทแต่ละเซลล์ยังมีการขยายหรือลดขนาดของสัญญาณอีกด้วย โดยสัญญาณจากเดนไดรท์ต่าง ๆ จะรวมกันเข้าสู่เซลล์ประสาท และหากสัญญาณรวมมีความแรงเกินค่าระดับ (Threshold) ของเซลล์ประสาทนั้น เซลล์ประสาทก็จะส่งสัญญาณออกทางแอคซอนต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

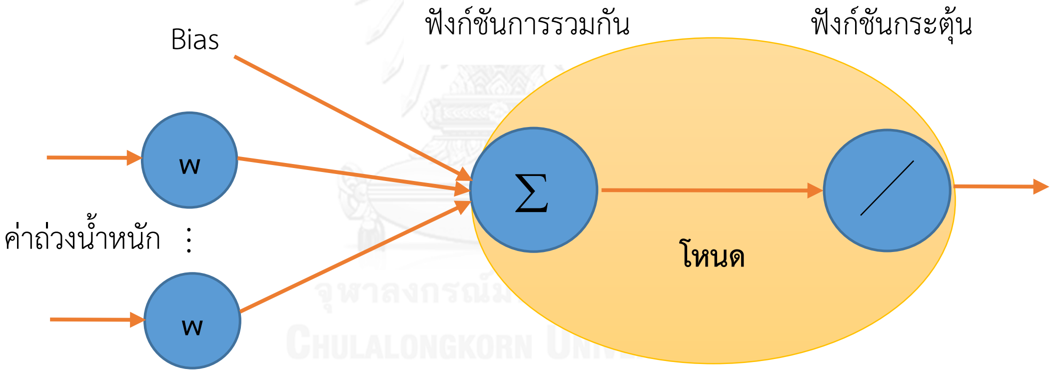


ภาพที่ 2- 5 กระบวนการทำงานภายในเซลล์ประสาท

(ที่มา: http://cnx.org/contents/Sr8Ev5Og@1.13:SO2ufnKm@2/)

เครือข่ายประสาทเทียมมีคุณลักษณะคล้ายกับการส่งผ่านสัญญาณประสาทในสมองของมนุษย์กล่าวคือ มีความสามารถในการรวบรวมความรู้ (Knowledge) โดยผ่านกระบวนการเรียนรู้ (Learning process) และความรู้เหล่านี้จะจัดเก็บอยู่ในเครือข่ายในรูปแบบค่าถ่วงน้ำหนัก (Weight) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่ ๆ เข้าไป ค่าถ่วงน้ำหนักทำหน้าที่เปรียบเสมือนความรู้ที่รวบรวมไว้เพื่อใช้ในการแก้ปัญหาเฉพาะอย่างของมนุษย์

การประมวลผลต่าง ๆ เกิดขึ้นในหน่วยประมวลผลย่อย เรียกว่า โหนด (Node) ซึ่งโหนดเป็นการจำลองลักษณะการทำงานมาจากเซลล์การส่งสัญญาณ (Signal) ระหว่างโหนดที่เชื่อมต่อกัน (Connection) จำลองมาจากการเชื่อมต่อของเดนไดรท์และแอคซอนในระบบประสาทของมนุษย์ ภายในโหนดจะมีฟังก์ชันกำหนดสัญญาณส่งออกที่เรียกว่า ฟังก์ชันกระตุ้น (Activation function) หรือฟังก์ชันการแปลง (Transfer function) ซึ่งทำหน้าที่เปรียบเสมือนกระบวนการทำงานในเซลล์ดังรูป



ภาพที่ 2- 6 กระบวนการทำงานในเซลล์เสมือนเป็นการจำลองการทำงานภายในโหนด

2.2.1 ข้อดีและข้อเสียของการใช้เครือข่ายประสาทเทียม

ข้อดีของการใช้เครือข่ายประสาทเทียม คือ ความแกร่ง (Robust) เมื่อมีข้อมูลรบกวน (Noisy data) ที่มีการกระจายตัวมาก ส่ายไปส่ายมา และมีค่านอกเกณฑ์ (Outlier) เนื่องจากเครือข่ายประสาทเทียมมีโหนดจำนวนมาก (เซลล์ประสาทเทียม) ด้วยการถ่วงน้ำหนักเพื่อการเชื่อมต่อกับโหนดต่าง ๆ เครือข่ายประสาทเทียมสามารถเรียนรู้ที่จะทำงานในชุดข้อมูลที่มีค่าคลาดเคลื่อนได้ดี ส่วน

ข้อเสียของการใช้เครือข่ายประสาทเทียม คือ การแปลความหมายที่เข้าใจได้ยากและบ่อยครั้งเครือข่ายประสาทเทียมจำเป็นต้องใช้ระยะเวลาในการฝึกหัดนานหลายชั่วโมง

2.2.2 การจัดเตรียมข้อมูลก่อนประมวลผลข้อมูลและหลังประมาณผลข้อมูล (Data preprocessing and postprocessing)

ข้อมูลดิบ (Raw data) ที่อยู่ในฐานข้อมูลส่วนใหญ่อาจประกอบไปด้วยข้อมูลที่ล้าสมัยหรือฟุ่มเฟือยเกินไป (Obsolete หรือ Redundant), ข้อมูลสูญหาย (Missing value) และค่านอกเกณฑ์ (Outlier) ในการสร้างตัวแบบเครือข่ายประสาทเทียม ฐานข้อมูลจำเป็นที่จะต้องได้รับการจัดเตรียมข้อมูลก่อนประมวลผลข้อมูลในรูปของการแปลงข้อมูลหรือการปรับเปลี่ยนรูปแบบข้อมูล (Data transformation) เพื่อทำให้การสร้างตัวแบบใช้เวลาน้อยลงรวมถึงทำให้ค่าพยากรณ์ที่ได้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งมีวิธีการแปลงข้อมูลหลายวิธี เช่น การแปลงข้อมูลให้อยู่ในรูปปรกติมาตรฐานน้อยที่สุด-มากที่สุด (Min-Max normalization), การแปลงข้อมูลให้อยู่ในรูปคะแนน Z (Z-score standardization), การเปลงข้อมูลให้อยู่ในช่วง -1 ถึง 1 เป็นต้น แต่เนื่องจากราคาปิดของหุ้นมีค่าเป็นบวกเสมอ เพื่อให้สอดคล้องกับข้อมูลข้างต้น จึงใช้การแปลงข้อมูลให้อยู่ในรูปปรกติมาตรฐานน้อยที่สุด-มากที่สุด และภายหลังจากการพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมแล้ว ค่าพยากรณ์ที่ได้ต้องทำการแปลงกลับให้อยู่ในรูปแบบเดิมก่อนจึงจะสามารถนำไปใช้ในการวิเคราะห์ต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2.3 การแปลงข้อมูลให้อยู่ในรูปปรกติมาตรฐานน้อยที่สุด-มากที่สุด (Min-Max normalization)

ข้อมูลที่อยู่ในรูปปรกติมาตรฐานน้อยที่สุด-มากที่สุด ใช้พิจารณาว่าค่าของข้อมูลมากกว่าค่าน้อยที่สุดอยู่เท่าไร และหารความแตกต่างนี้ด้วยพิสัย ซึ่งค่าที่ได้จะอยู่ในช่วง 0 ถึง 1 นั่นคือ

$$x_t^* = \frac{x_t - \min(x_t)}{range(x_t)} = \frac{x_t - \min(x_t)}{\max(x_t) - \min(x_t)} \tag{2-49}$$

โดยที่ x_t^* แทน ค่าของข้อมูลที่ทำให้เป็นปรกติมาตรฐานน้อยที่สุด-มากที่สุด

x_t แทน ค่าของข้อมูลเริ่มต้น

$\min(x_t)$ แทน ค่าน้อยที่สุด

และ $\max(x_t)$ แทน ค่ามากที่สุด

2.2.4 เทคนิคการเรียนรู้ของเครือข่ายประสาทเทียม (Artificial neural network learning)

เทคนิคการเรียนรู้ของเครือข่ายประสาทเทียม จะมีประสิทธิภาพเพียงใดนั้นขึ้นอยู่กับค่าถ่วงน้ำหนัก (Weight) โดยทั่วไปสามารถจำแนกเทคนิคการเรียนรู้ของเครือข่ายประสาทเทียมได้เป็น 2 ประเภทหลัก ๆ คือ เทคนิคการเรียนรู้แบบมีผู้สอนและเทคนิคการเรียนรู้แบบไม่มีผู้สอน

2.2.4.1 เทคนิคการเรียนรู้แบบมีผู้สอน (Supervised learning)

เทคนิคการเรียนรู้แบบมีผู้สอน จะมีการกำหนดข้อมูลชุดฝึกสอน (Training data set) ให้กับเครือข่ายประสาทเทียม ซึ่งชุดข้อมูลนี้ประกอบด้วยข้อมูลนำเข้า (Input data) และข้อมูลเป้าหมาย (Target data) ที่ต้องการ จากนั้นเครือข่ายประสาทเทียมจะทำการคำนวณค่าถ่วงน้ำหนักที่เหมาะสมให้กับชุดข้อมูลฝึกสอนนี้ โดยข้อมูลผลลัพธ์ (Output data) ที่ได้จากเครือข่ายประสาทเทียมจะถูกคำนวณหาค่าคลาดเคลื่อน (Error) ถ้ายังมีค่าคลาดเคลื่อนสูงอยู่ การฝึกสอนจะดำเนินต่อไปจนกว่าค่าคลาดเคลื่อนจะลดลงต่ำกว่าค่าที่ยอมรับได้ (Accept level) จึงจะหยุดการฝึกสอน ซึ่งเทคนิคในประเภทนี้จะเน้นการเรียนรู้จากข้อมูลที่มีอยู่ในอดีตเพื่อนำมาสร้างตัวแบบสำหรับพยากรณ์หรือคาดการณ์สิ่งที่เกิดขึ้นในอนาคต ตัวแบบในที่นี้อาจจะเป็นสมการเชิงคณิตศาสตร์หรือกฎต่าง ๆ ก็ได้ นอกจากนี้เทคนิคการเรียนรู้แบบมีผู้สอนยังสามารถแบ่งย่อยได้อีก คือ การจำแนกประเภทข้อมูล (Classification) และการถดถอย (Regression) ซึ่งทั้งสองเทคนิคนี้มีลักษณะคล้ายกันมากแต่แตกต่างกันที่ผลลัพธ์ที่ต้องการพยากรณ์นั่นเอง

2.2.4.2 เทคนิคการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning)

เทคนิคการเรียนรู้แบบไม่มีผู้สอน จะอาศัยชุดข้อมูลนำเข้า (Input data set) เพียงอย่างเดียวในการฝึกสอนเครือข่ายประสาทเทียมโดยไม่มีข้อมูลเป้าหมาย (Target data) แต่จะใช้ข้อมูลผลลัพธ์ (Output data) จากเครือข่ายประสาทเทียมแทน เมื่อป้อนข้อมูลนำเข้าสู่เครือข่ายประสาทเทียม เครือข่ายประสาทเทียมจะคำนวณค่าความสัมพันธ์ที่มีอยู่ภายในกลุ่มข้อมูลนำเข้า โดยอาศัยค่าถ่วงน้ำหนักเป็นตัวแยกความแตกต่างของข้อมูลนำเข้าและนำไปเก็บไว้ในโหนดของข้อมูลผลลัพธ์ของ

เครือข่ายประสาทเทียม ซึ่งเทคนิคในประเภทนี้จะเน้นการพิจารณาข้อมูลเป็นหลัก เช่น พิจารณาว่าข้อมูลมีความสัมพันธ์กันในลักษณะใดบ้าง เป็นต้น นอกจากนี้เทคนิคในประเภทนี้ยังแบ่งย่อยได้อีก คือ เทคนิคการค้นหากฎความสัมพันธ์ (Association rules) และการแบ่งกลุ่มข้อมูล (Clustering)

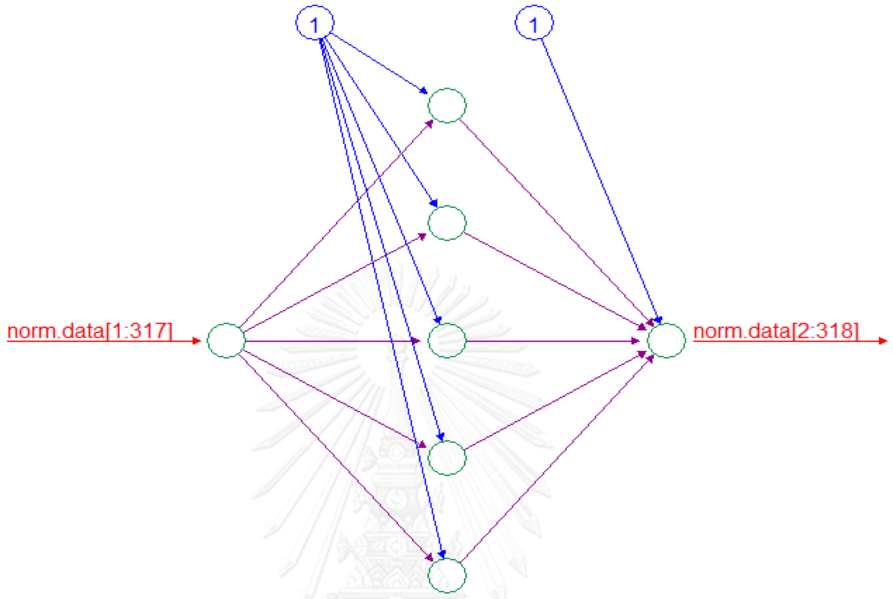
2.2.5 การเชื่อมโยงของเครือข่ายประสาทเทียม (Artificial neural network linking)

เพื่อให้เครือข่ายประสาทเทียมสามารถเรียนรู้ได้อย่างมีประสิทธิภาพ จำเป็นต้องมีการเชื่อมโยงกันระหว่างเซลล์ประสาท โดยทั่วไปสามารถแบ่งการเชื่อมโยงของเครือข่ายประสาทเทียมได้ 2 ลักษณะ คือ เครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network) และเครือข่ายประสาทเทียมแบบมีการย้อนกลับ (Feedback artificial neural network)

2.2.5.1 เครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network)

เครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า เป็นเครือข่ายประสาทเทียมที่การประมวลผลจะอาศัยชุดข้อมูลปัจจุบันและส่งค่าที่ประมวลผลได้ไปยังชั้นถัด ๆไป กล่าวคือ เครือข่ายประสาทเทียมประเภทนี้จะประกอบด้วยชั้นต่าง ๆ โดยชั้นแรกจะเป็นชั้นข้อมูลนำเข้า (Input layer) และชั้นสุดท้ายเป็นชั้นผลลัพธ์ (Output layer) ส่วนระหว่างชั้นข้อมูลนำเข้ากับชั้นผลลัพธ์อาจจะมีหรือไม่มีชั้นซ่อน (Hidden layer) อยู่ภายในก็ได้ซึ่งขึ้นกับกฎการเรียนรู้ (Learning rule) ที่ใช้ในการสอนเครือข่ายประสาทเทียม เช่น ถ้าเป็นเครือข่ายเพอร์เซปตรอนแบบหลายชั้น (Multi-layer perceptron) จะมีชั้นซ่อนอยู่ระหว่างชั้นข้อมูลนำเข้ากับชั้นผลลัพธ์ซึ่งอาจมีมากกว่า 1 ชั้นก็ได้ การเชื่อมต่อระหว่างชั้นของเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้าจะมีค่าถ่วงน้ำหนัก (Weight) เป็นตัวเชื่อม และสัญญาณนำเข้าที่เข้ามาจะถูกส่งไปตามทิศทางของลูกศรที่มีการจำกัดทิศทางการเคลื่อนที่เป็นแบบทางเดียวหรือเคลื่อนที่ไปข้างหน้าโดยเครือข่ายมีการเชื่อมต่อกันอย่างสมบูรณ์ (Complete connected network) นั่นคือ ที่โหนดทุกโหนดในชั้นที่กำหนดเชื่อมต่อกันทุกโหนดกับชั้นถัดไปจนถึงชั้นผลลัพธ์โดยไม่มีการย้อนกลับ โดยทั่วไปจำนวนโหนดในชั้นข้อมูลนำเข้าขึ้นอยู่กับจำนวนและประเภทของข้อมูล จำนวนของชั้นซ่อนและจำนวนของโหนดในแต่ละชั้นซ่อนขึ้นอยู่กับผู้ใช้เป็นผู้กำหนด และในชั้นผลลัพธ์อาจจะมี 1 โหนดหรือมากกว่า 1 โหนดก็ได้ เช่น ต้องการสร้างตัวแบบเครือข่ายประสาทเทียมสำหรับพยากรณ์ราคาปิดของหุ้น SCB โดยใช้ชุดข้อมูลฝึกสอน

จำนวน 318 สัปดาห์ สำหรับสร้างตัวแบบ ซึ่งประกอบด้วยชั้นข้อมูลนำเข้า 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน 1 ชั้นภายใต้จำนวนโหนด 5 โหนดโดยใช้ Sigmoid logistics function เป็น Activation function และชั้นผลลัพธ์ 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function ซึ่งสามารถแสดงตัวแบบของเครือข่ายประสาทเทียม ได้ดังนี้

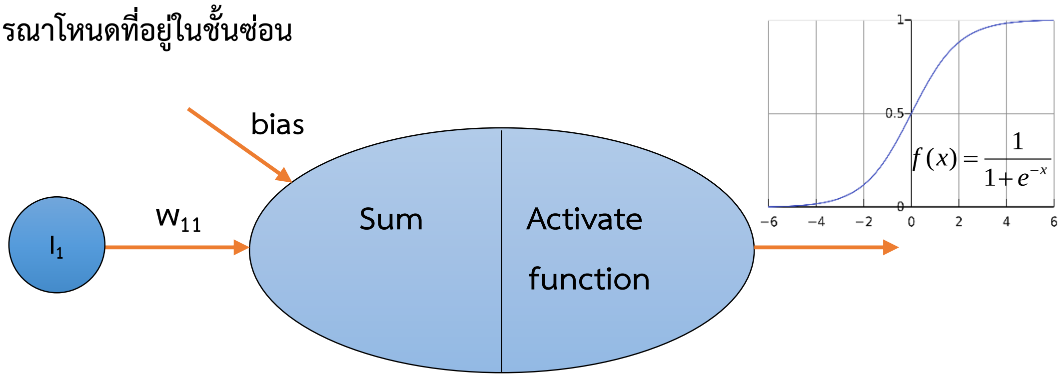


ภาพที่ 2- 7 ตัวแบบเครือข่ายประสาทเทียมซึ่งมีทั้งหมด 3 ชั้น (Layer)

กำหนดเวกเตอร์ $\tilde{x} = (x_1, x_2, ..., x_{317})$ เป็นชุดข้อมูลสำหรับชั้นข้อมูลนำเข้า (Input layer) ในการสร้างตัวแบบเครือข่ายประสาทเทียม แต่ละโหนดที่อยู่ในชั้นซ่อนและชั้นผลลัพธ์ จะประกอบไปด้วย 2 ขั้นตอน คือ

CHULALONGKORN UNIVERSITY

พิจารณาโหนดที่อยู่ในชั้นซ่อน



ภาพที่ 2- 8 การทำงานภายในโหนดที่ j ของชั้นซ่อน

<u>ขั้นตอนที่ 1</u> : คำนวณค่าผลรวมของโหนดในชั้นซ่อน (Hidden layer) โดยใช้ฟังก์ชันการรวมกัน (Combination function: $\sum$) ซึ่งเป็นการนำค่าของโหนดข้อมูลนำเข้าที่เชื่อมโยงมาคูณกับค่าถ่วงน้ำหนักในแต่ละเส้นเชื่อมโยง แล้วจึงนำมารวมกันเป็นค่าสุดท้าย ดังนั้น ค่าผลรวมของโหนดที่ j ในชั้นซ่อน คือ

$$net_j = w_{1j}x_1 + bias \quad \text{สำหรับ} \quad j = 1,2,\ldots,5 \qquad (2\text{-}50)$$

โดยที่ x_i แทน ข้อมูลนำเข้าที่ i

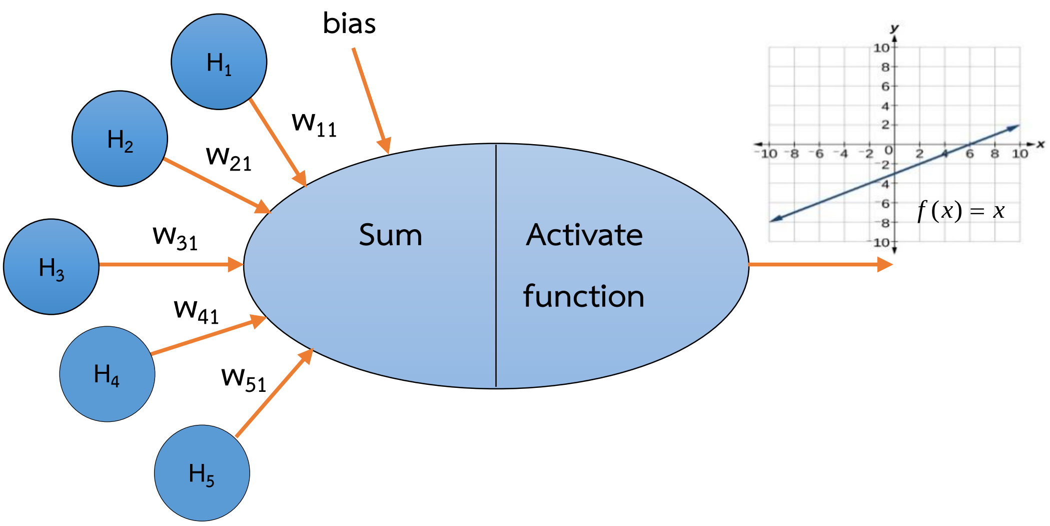
และ w_{ij} แทน ค่าถ่วงน้ำหนักที่สัมพันธ์กับข้อมูลนำเข้าที่ i ไปยังโหนดที่ j ในชั้นซ่อน

<u>ขั้นตอนที่ 2</u> : ทำการปรับค่าผลรวมของข้อมูลในชั้นซ่อนด้วยฟังก์ชันกระตุ้น (Activation function) เพื่อให้ได้ผลลัพธ์ (y_j) ของชั้นซ่อนโดยใช้ Sigmoid logistic function เพื่อให้สอดคล้องกับการแปลงข้อมูลก่อนการประมวลผลข้อมูลในรูปค่าปรกติมาตรฐานน้อยที่สุด-มากที่สุด (Min-Max normalization) และขอบเขตของผลลัพธ์ที่ได้หลังจากการปรับด้วย Sigmoid logistic function นี้อยู่ระหว่าง 0 และ 1 โดยที่

$$y_j = f(net_j) = \frac{1}{1+e^{-net_j}} \qquad (2\text{-}51)$$

พิจารณาโหนดที่อยู่ในชั้นผลลัพธ์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



ภาพที่ 2- 9 การทำงานภายในโหนดที่ k ของชั้นผลลัพธ์

<u>ขั้นตอนที่ 1</u> : คำนวณค่าผลรวมของโหนดในชั้นผลลัพธ์ (Output layer) โดยใช้ฟังก์ชันการรวมกัน (Combination function: $\sum$) ดังนั้น ค่าผลรวมของโหนดที่ k ในชั้นผลลัพธ์ คือ

$$net_k = \sum_{j=1}^{5} w_{jk} y_j = w_{11} y_1 + w_{21} y_2 + \ldots + w_{51} y_5 + bias \quad \text{สำหรับ } k = 1 \tag{2-52}$$

<u>ขั้นตอนที่ 2</u> : จากนั้นทำการปรับค่าผลรวมของข้อมูลในชั้นผลลัพธ์ด้วยฟังก์ชันกระตุ้น (Activation function) โดยใช้ Linear function เพื่อให้ได้ผลลัพธ์ (y_k) ของชั้นผลลัพธ์ โดยที่

$$y_k = f(net_k) = net_k \tag{2-53}$$

2.2.6 เทคนิคการแพร่แบบย้อนกลับ (Back-propagation)

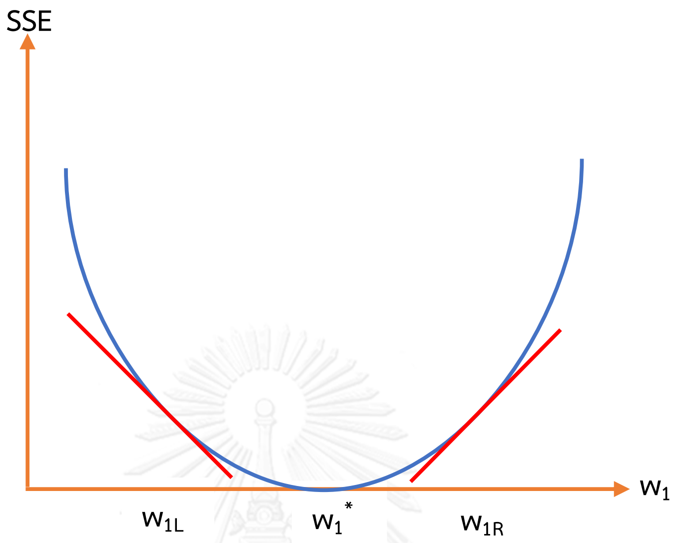
การแพร่แบบย้อนกลับเป็นขั้นตอนที่ใช้สอนเครือข่ายประสาทเทียมแบบเพอร์เซปตรอนหลายชั้น (Multi-layer perceptron) ซึ่งตัวแบบเครือข่ายประสาทเทียมมีการเชื่อมโยงกันเป็นเครือข่ายแบบเป็นชั้น ๆ เครือข่ายชนิดนี้มีการเชื่อมโยงกัน 3 ชั้น ประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) ถัดมาเป็นชั้นซ่อน (Hidden layer) และชั้นสุดท้ายเป็นชั้นผลลัพธ์ (Output layer) โดยก่อนที่จะทำการสอนเครือข่ายประสาทเทียมแบบหลายชั้นนั้นจำเป็นต้องกำหนดค่าเริ่มต้นให้กับค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นทุกชั้น โดยค่านี้ได้มาจากการสุ่ม (Randomness)

การแพร่แบบย้อนกลับนั้นมีที่มาจาก วิธีการปรับค่าถ่วงน้ำหนักเพื่อให้ได้ค่าที่เหมาะสมนั้นจะใช้วิธีการสอนว่าข้อมูลเป้าหมาย (Target data) ของแต่ละข้อมูลนำเข้า (Input data) นั้นคืออะไร และใช้ผลบวกกำลังสองของค่าคลาดเคลื่อน (Sum of squares error: SSE) ของข้อมูลผลลัพธ์ (Output data) มาใช้เป็นตัวชี้นำในการปรับค่าถ่วงน้ำหนัก โดยการสร้างชุดของค่าถ่วงน้ำหนักที่ทำให้ SSE มีค่าน้อยที่สุดด้วย**วิธีลดองศา (Gradient descent method)** ซึ่งจะให้ทิศทางที่ควรจะปรับค่าถ่วงน้ำหนักเพื่อลดค่า SSE ดังนั้นเทคนิคการแพร่แบบย้อนกลับจึงเป็นเทคนิคเรียนรู้แบบมีผู้สอน

กำหนดเวกเตอร์ของค่าถ่วงน้ำหนักจำนวน m ค่า คือ $w = (w_1, w_2, \ldots, w_m)$ ในตัวแบบเครือข่ายประสาทเทียม **องศาของ SSE เมื่อเทียบกับเวกเตอร์ของค่าถ่วงน้ำหนัก** w คือ อนุพันธ์บางส่วนของ SSE เมื่อเทียบกับค่าถ่วงน้ำหนักแต่ละค่า กล่าวคือ

$$\nabla SSE(w) = \left[\frac{\partial SSE}{\partial w_1}, \frac{\partial SSE}{\partial w_2}, \ldots, \frac{\partial SSE}{\partial w_m}\right] \quad (2\text{-}54)$$

พิจารณาการทำงานของวิธีลดองศาในกรณีที่มีค่าถ่วงน้ำหนัก w_1 เพียงค่าเดียว



ภาพที่ 2- 10 ความสัมพันธ์ระหว่างผลบวกกำลังสองของค่าคลาดเคลื่อนกับค่าถ่วงน้ำหนัก w_1

จากกราฟข้างต้น เราต้องการหาค่าถ่วงน้ำหนัก w_1 ที่ทำให้ SSE มีค่าน้อยที่สุด สมมติว่าค่าถ่วงน้ำหนักที่เหมาะสมที่สุดของ w_1 คือ w_1^* ดังนั้นเราต้องทำการเคลื่อนที่ค่าถ่วงน้ำหนักปัจจุบัน $(w_{current})$ ให้ใกล้กับค่าถ่วงน้ำหนักที่เหมาะสมที่สุด (w_1^*) กล่าวคือ

$$w_{new} = w_{current} + \Delta w_{current} \quad (2\text{-}55)$$

โดยที่ $\Delta w_{current}$ เป็นการเปลี่ยนแปลงค่าถ่วงน้ำหนักในตำแหน่งปัจจุบันของ w_1

สมมติว่าค่าถ่วงน้ำหนัก $w_{current}$ อยู่ใกล้เคียงกับค่าถ่วงน้ำหนัก w_{1L} เราต้องทำการปรับเพิ่มค่าถ่วงน้ำหนัก $w_{current}$ เพื่อให้ใกล้เคียงกับค่าถ่วงน้ำหนัก w_1^* โดยทำการหาค่า $\frac{\partial SSE}{\partial w_1}$ ซึ่งก็คือความชันของเส้นโค้ง SSE ที่ w_1 ที่ใกล้กับ w_{1L} พบว่าความชันของเส้นโค้งนี้มีค่าเป็นลบ ในทางตรงกันข้ามถ้าค่าถ่วงน้ำหนัก $w_{current}$ อยู่ใกล้เคียงกับค่าถ่วงน้ำหนัก w_{1R} เราต้องทำการปรับลดค่าถ่วงน้ำหนัก $w_{current}$ เพื่อให้ใกล้เคียงกับค่าถ่วงน้ำหนัก w_1^* พบว่าค่าถ่วงน้ำหนัก w_1 ที่ใกล้กับ w_{1R}

ความชันของเส้นโค้งมีค่าเป็นบวก ดังนั้น ทิศทางสำหรับการปรับค่าถ่วงน้ำหนัก $w_{current}$ มีค่าเป็น $-sign\left(\frac{\partial SSE}{\partial w_{current}}\right)$

พิจารณาค่าอนุพันธ์บางส่วนของ SSE ที่ค่าถ่วงน้ำหนัก $w_{current}$ พบว่า เมื่อเส้นโค้งมีความชันเพิ่มขึ้น การปรับค่าถ่วงน้ำหนัก $w_{current}$ ในทิศทางของ $-sign\left(\frac{\partial SSE}{\partial w_{current}}\right)$ จะมาก เนื่องจากค่าของอนุพันธ์ที่จุดนั้นมีค่ามากขึ้น ในทางตรงกันข้ามเมื่อเส้นโค้งค่อนข้างแบนราบ การปรับค่าถ่วงน้ำหนัก $w_{current}$ ในทิศทางของ $-sign\left(\frac{\partial SSE}{\partial w_{current}}\right)$ จะน้อย เนื่องจากความชันที่จุดนั้นมีค่าน้อยลง จากนั้นทำการคูณค่าอนุพันธ์บางส่วนข้างต้นกับค่าคงที่ซึ่งช่วยให้การเคลื่อนที่ของค่าถ่วงน้ำหนักในเครือข่ายประสาทเทียมไปยังจุดที่มีค่า SSE น้อยที่สุด เรียกค่าคงที่นั้นว่า **อัตราการเรียนรู้ (Learning rate)** แทนด้วย η มีค่าอยู่ระหว่าง 0 และ 1 โดยทั่วไปค่าที่เหมาะสมจะอยู่ในช่วง 0.01 ถึง 0.5 ดังนั้น สามารถเขียนรูปแบบของ $\Delta w_{current}$ ได้ในรูปสมการ ดังนี้

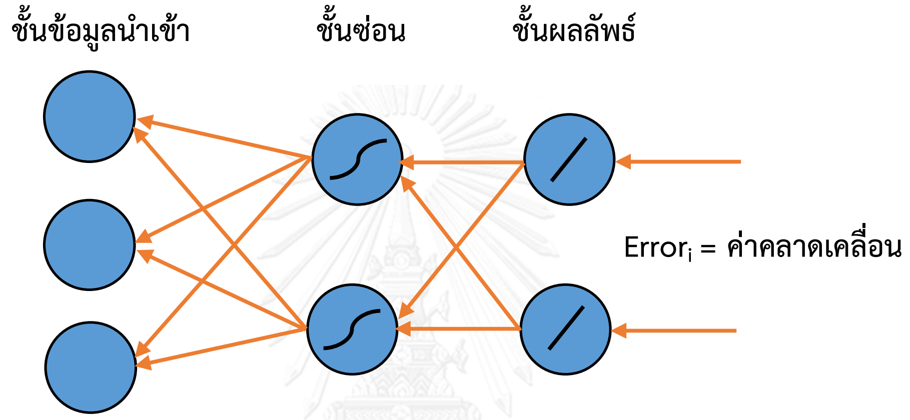
$$\Delta w_{currnt} = -\eta\left(\frac{\partial SSE}{\partial w_{current}}\right) \tag{2-56}$$

นั่นคือ การเปลี่ยนแปลงค่าถ่วงน้ำหนักปัจจุบันมีค่าเท่ากับค่าคงที่ที่มีค่าเป็นลบ $(-\eta)$ คูณกับความชันของค่าคลาดเคลื่อนที่ค่าถ่วงน้ำหนักปัจจุบัน

จากกราฟระหว่างผลบวกกำลังสองของค่าคลาดเคลื่อนกับค่าถ่วงน้ำหนัก w_1 โดยที่ w_1^* เป็นค่าถ่วงน้ำหนักที่เหมาะสมที่สุดสำหรับถ่วงน้ำหนัก w_1 ซึ่งมีค่าถ่วงน้ำหนักปัจจุบันเป็น $w_{current}$ ตามกฎการลดองศา $\Delta w_{currnt} = -\eta\left(\frac{\partial SSE}{\partial w_{current}}\right)$ พบว่าค่าถ่วงน้ำหนัก $w_{current}$ จะปรับตัวตามทิศทางของค่าถ่วงน้ำหนัก w_1^* แต่ถ้าอัตราการเรียนรู้ (η) ซึ่งเป็นตัวคูณในสูตรของ Δw_{currnt} มีค่ามากเกินไปจะทำให้ค่าถ่วงน้ำหนักใหม่ (w_{new}) ที่ได้มีค่ามากกว่าค่าถ่วงน้ำหนักที่เหมาะสมที่สุด (w_1^*) และค่าถ่วงน้ำหนัก w_{new} อาจจะอยู่ห่างไกลจากค่าถ่วงน้ำหนัก w_1^* มากกว่าค่าถ่วงน้ำหนัก $w_{current}$ ในทางตรงกันข้ามถ้าอัตราการเรียนรู้ (η) มีค่าน้อยเกินไป แสดงว่ากำหนดให้เครือข่ายประสาทเทียมมีการเปลี่ยนแปลงค่าถ่วงน้ำหนักที่น้อยซึ่งจำเป็นต้องใช้ระยะเวลาในการเรียนรู้ที่มากขึ้น

2.2.7 กฎการแพร่แบบย้อนกลับ (Back-propagation rules)

ปัญหาที่เกิดขึ้นจากการที่ไม่มีค่าเป้าหมาย (Target data) ที่ออกมาจากแต่ละโหนดในชั้นซ่อนนั้นทำให้มีความจำเป็นที่ต้องอาศัยการแพร่ของค่าคลาดเคลื่อนจากชั้นผลลัพธ์ย้อนกลับมายังชั้นซ่อน นั่นคือ **ค่าคลาดเคลื่อน (Error)** จะถูกคำนวณที่ชั้นผลลัพธ์แล้วแพร่ย้อนกลับไปยังชั้นข้อมูลนำเข้า และมีการปรับเปลี่ยนค่าถ่วงน้ำหนักของเส้นเชื่อมระหว่างโหนดต่าง ๆ ใหม่อีกครั้งจนได้ค่าที่มีค่าคลาดเคลื่อนน้อยที่สุด แสดงดังรูป



ภาพที่ 2- 11 การแพร่แบบย้อนกลับของค่าคลาดเคลื่อนจากชั้นผลลัพธ์มายังชั้นข้อมูลนำเข้า

การปรับปรุงค่าคลาดเคลื่อนเกิดหลังจากที่เครือข่ายประสาทเทียมดำเนินการส่งสัญญาณไปข้างหน้าแล้ว โหนดแต่ละโหนดในชั้นผลลัพธ์จะให้ค่าผลลัพธ์ (Output data) ที่ต้องนำมาเปรียบเทียบกับค่าเป้าหมาย (Target data) ในชุดข้อมูลฝึกสอน ความแตกต่างนั้นคือ ค่าคลาดเคลื่อนสำหรับแต่ละโหนดในชั้นผลลัพธ์ ซึ่งค่าถ่วงน้ำหนักของทุกการเชื่อมโยงไปยังชั้นผลลัพธ์จะถูกปรับเพื่อลดค่าคลาดเคลื่อนโดยใช้วิธีลดองศาและใช้ Linear function เป็น Activation function จากนั้นค่าคลาดเคลื่อนของโหนดในชั้นซ่อนจะถูกปรับเช่นกัน โดยใช้ Sigmoid logistic function เป็น Activation function กระบวนการนี้จะดำเนินไปจนกระทั่งถึงชั้นข้อมูลนำเข้า

ค่าคลาดเคลื่อนของโหนดที่ j แทนด้วย Error$_j$ สามารถคำนวณได้โดยใช้อนุพันธ์บางส่วนของ Activation function เทียบกับ net$_j$ โดยขึ้นอยู่กับโหนดว่าอยู่ในชั้นผลลัพธ์หรือชั้นซ่อน นั่นคือ

$$Error_j = (t\arg et_j - output_j)\frac{\partial y_j}{\partial net_j} = (t\arg et_j - output_j)\frac{\partial f(net_j)}{\partial net_j} \quad (2\text{-}57)$$

$$\text{หรือ } Error_j = \begin{cases} output_j(1-output_j)(t\arg et_j - output_j) & \text{สำหรับโหนดในชั้นผลลัพธ์} \\ output_j(1-output_j)\sum_{j=1}^{N} Error_j \times w_{jk} & \text{สำหรับโหนดในชั้นซ่อน} \end{cases} \quad (2\text{-}58)$$

หลังจากนั้นนำค่าคลาดเคลื่อนที่ได้มาทำการปรับปรุงค่าถ่วงน้ำหนักของเส้นเชื่อมระหว่างโหนดด้วยสมการ

$$w_{ij,new} = w_{ij,current} + (\eta \times Error_j \times output_i) \quad (2\text{-}59)$$

โดยที่ w_{ij} แทน ค่าถ่วงน้ำหนักของเส้นเชื่อมต่อระหว่างโหนดที่ i และโหนดที่ j

η แทน อัตราการเรียนรู้

$Error_j$ แทน ค่าคลาดเคลื่อนของโหนดที่ j

และ $output_i$ แทน ค่าผลลัพธ์ที่คำนวณได้จากโหนดที่ i

2.2.8 โมเมนตัม (Momentum term)

โมเมนตัม (α) ถูกคิดค้นโดย Rumelhart Hinton และ William เป็นค่าคงที่ที่ช่วยไม่ให้การเปลี่ยนแปลงของค่าถ่วงน้ำหนักมากเกินไปและทำให้ค่าถ่วงน้ำหนักเข้าใกล้ค่าที่เหมาะสมที่สุด พบว่าการเพิ่มค่าคงที่โมเมนตัมนั้นจะทำให้ขั้นตอนการแพร่แบบย้อนกลับในเครือข่ายประสาทเทียมมีประสิทธิภาพเพิ่มมากขึ้นและทำให้มีการเรียนรู้ที่เร็วขึ้น ซึ่งค่าโมเมนตัมมีค่าอยู่ระหว่าง 0 และ 1 โดยที่ค่าโมเมนตัมที่เหมาะสมจะมีค่าเข้าใกล้ 1 นอกจากนั้นจะต้องกำหนดค่าอัตราการเรียนรู้ (η) ให้สอดคล้องกับค่าโมเมนตัมด้วย เช่น ถ้าอัตราการเรียนรู้สูงก็ควรที่จะมีค่าโมเมนตัมที่ต่ำ ทำให้การเปลี่ยนแปลงของค่าถ่วงน้ำหนักนั้นไม่มากจนเกินไป โดยที่ค่าถ่วงน้ำหนักจะถูกปรับโดยสมการ

$$\Delta w_{currnt} = -\eta\left(\frac{\partial SSE}{\partial w_{current}}\right) + \alpha \Delta w_{previous} \quad (2\text{-}60)$$

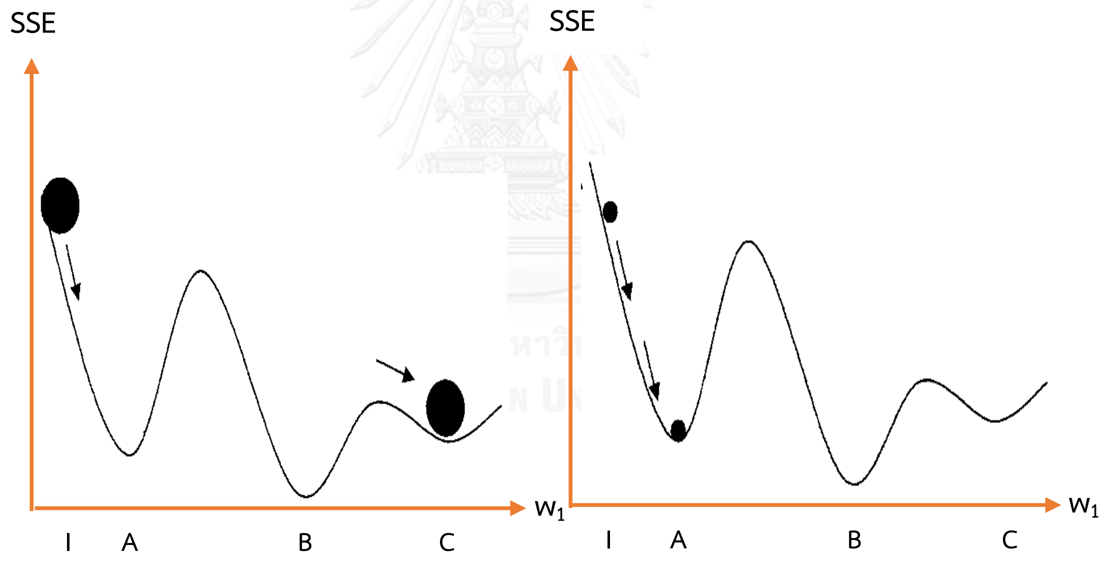
เมื่อ $\Delta w_{previous}$ แทน การปรับค่าถ่วงน้ำหนักที่ผ่านมา และ $0 \le \alpha \le 1$

จากสมการข้างต้น พบว่า ค่าโมเมนตัมที่มากจะมีอิทธิพลต่อการปรับค่าถ่วงน้ำหนักปัจจุบัน (Δw_{currnt}) เพื่อเคลื่อนที่ไปในทิศทางเดียวกันกับการปรับค่าถ่วงน้ำหนักที่ผ่านมา $\Delta w_{previous}$ การรวมโมเมนตัมในการแพร่แบบย้อนกลับจะทำให้การปรับค่าถ่วงน้ำหนักเป็นการเฉลี่ยเอ็กโปเนนเชียลของการปรับค่าถ่วงน้ำหนักที่ผ่านมาทั้งหมด นั่นคือ

$$\Delta w_{current} = -\eta \sum_{k=0}^{\infty} \alpha^k \left(\frac{\partial SSE}{\partial w_{current-k}} \right) \tag{2-61}$$

พบว่า ถ้าค่าโมเมนตัมมีค่ามากขึ้นจะทำให้มีอิทธิพลต่อการปรับค่าถ่วงน้ำหนักที่ผ่านมามีค่ามากยิ่งขึ้น ในทางตรงกันข้าม ถ้าค่าโมเมนตัมมีค่าน้อยจะช่วยลดอิทธิพลต่อการปรับค่าถ่วงน้ำหนักที่ผ่านมา และถ้าค่าโมเมนตัมเป็นศูนย์จะทำให้เทอมทั้งหมดนั้นหายไป

พิจารณากราฟระหว่างผลบวกกำลังสองของค่าคลาดเคลื่อนกับค่าถ่วงน้ำหนัก w_1



ภาพที่ 2- 12 การปรับค่าถ่วงน้ำหนักเมื่อพิจารณาค่าโมเมนตัมร่วมกับอัตราการเรียนรู้

พบว่า ค่าถ่วงน้ำหนักเริ่มต้นที่ตำแหน่ง i ซึ่งมีค่าต่ำสุด Local (Local minimum) ของ SSE อยู่ที่ตำแหน่ง A และตำแหน่ง C โดยที่ค่าต่ำสุด Global (Global minimum) ของ SSE ที่เหมาะสมที่สุดอยู่ที่ตำแหน่ง B สมมติว่าโมเมนตัม (α) มีค่าน้อย แทนด้วยลูกบอลขนาดเล็ก ถ้าทำการกลิ้งลูกบอลขนาดเล็กไปตามเส้นโค้ง ลูกบอลอาจจะหยุดอยู่ที่ตำแหน่ง A ทำให้สามารถหาค่าที่ต่ำสุด Local ของ SSE ที่ตำแหน่ง A ได้ แต่ไม่สามารถหาค่าต่ำสุด Global ของ SSE ที่ตำแหน่ง B ได้ ในทางตรงกัน

ข้ามถ้าสมมติว่าโมเมนตัม (α) มีค่ามาก แทนด้วยลูกบอลขนาดใหญ่ ถ้าทำการกลิ้งลูกบอลขนาดใหญ่ไปตามเส้นโค้ง ลูกบอลอาจจะมีโมเมนตัมมากจนผ่านเลยค่าต่ำสุด Global ของ SSE ที่ตำแหน่ง B ไปและหยุดอยู่ที่ค่าต่ำสุด Local ของ SSE ที่ตำแหน่ง C แทน ดังนั้นในการสร้างตัวแบบเครือข่ายประสาทเทียมจะต้องทำการพิจารณากำหนดทั้งค่าอัตราการเรียนรู้ (η) และค่าโมเมนตัม (α) โดยทำการทดลองเลือกค่าอัตราการเรียนรู้และค่าโมเมนตัม ณ ระดับต่าง ๆ เพื่อหาค่าที่ทำให้ได้ผลลัพธ์ที่ดีที่สุด

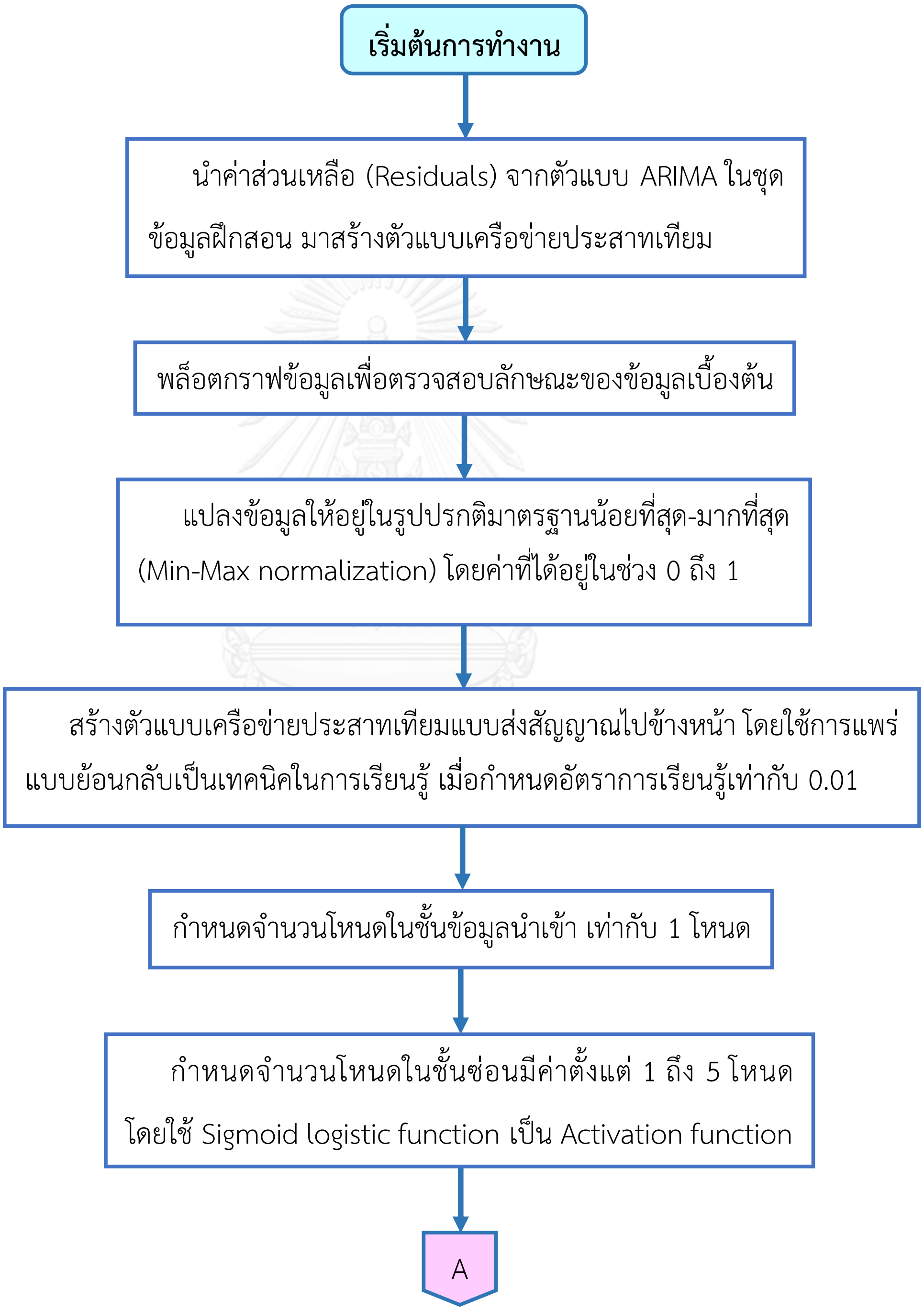
2.2.9 เกณฑ์การหยุด (Termination criteria)

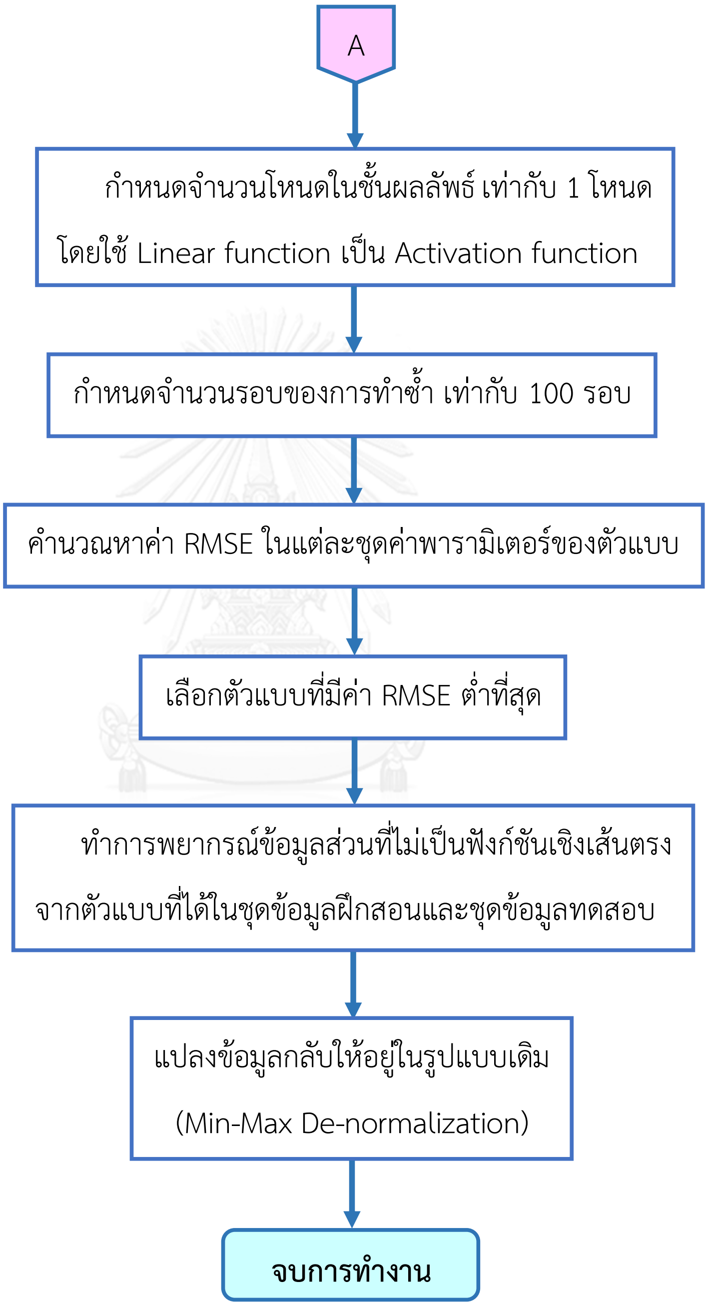
เกณฑ์ในการหยุดฝึกสอนนั้นขึ้นอยู่กับผู้ที่ทำการออกแบบเครือข่ายประสาทเทียมว่าต้องการที่จะให้เครือข่ายประสาทเทียมมีความแม่นมากน้อยเพียงใด โดยทั่วไปนิยมใช้ผลบวกกำลังสองของค่าคลาดเคลื่อน (Sum of squares error: SSE) เป็นเกณฑ์ในการประเมินซึ่งมีวิธีในการตรวจสอบการหยุดดังนี้

1. ใช้ชุดข้อมูลเริ่มต้นเป็นข้อมูลตรวจสอบความถูกต้อง
2. ประมวลผลข้อมูลเพื่อฝึกสอนเครือข่ายประสาทเทียมข้างต้นกับข้อมูลฝึกสอนที่เหลือ
3. ใช้ค่าถ่วงน้ำหนักที่เรียนรู้จากข้อมูลฝึกสอนกับข้อมูลตรวจสอบความถูกต้อง
4. ทำการตรวจสอบความถูกต้องของค่าถ่วงน้ำหนัก 2 ชุด ชุดแรกคือชุดของค่าถ่วงน้ำหนักปัจจุบันสำหรับข้อมูลฝึกสอน และชุดที่สองคือชุดของค่าถ่วงน้ำหนักที่ดีที่สุด (มีค่า SSE ต่ำที่สุด) สำหรับข้อมูลตรวจสอบความถูกต้อง
5. เมื่อชุดของค่าถ่วงน้ำหนักปัจจุบันมีค่า SSE มากกว่าชุดของค่าถ่วงน้ำหนักที่ดีที่สุด แล้วอัลกอริทึมจะหยุดการฝึกสอนลง

Chulalongkorn University

2.2.10 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบเครือข่ายประสาทเทียม




A
กำหนดจำนวนโหนดในชั้นผลลัพธ์ เท่ากับ 1 โหนด โดยใช้ Linear function เป็น Activation function
กำหนดจำนวนรอบของการทำซ้ำ เท่ากับ 100 รอบ
คำนวณหาค่า RMSE ในแต่ละชุดค่าพารามิเตอร์ของตัวแบบ
เลือกตัวแบบที่มีค่า RMSE ต่ำที่สุด
ทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง จากตัวแบบที่ได้ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ
แปลงข้อมูลกลับให้อยู่ในรูปแบบเดิม
(Min-Max De-normalization)
จบการทำงาน

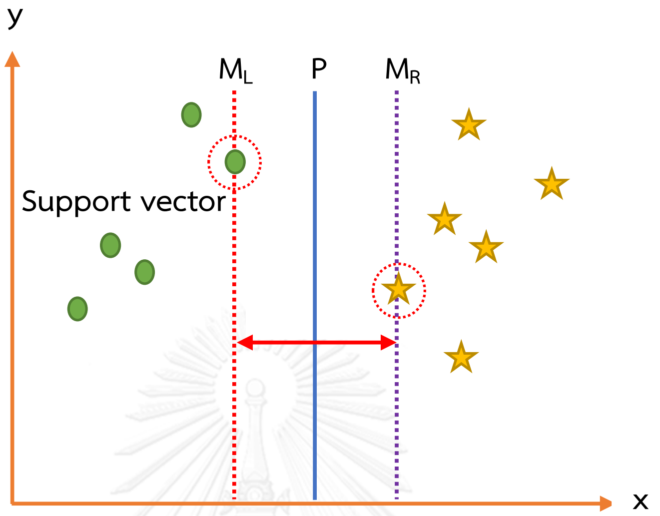
2.3 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีน (Support vector machine model: SVM)

ตัวแบบซัพพอร์ทเวกเตอร์แมชชีน ถูกพัฒนาขึ้นในปี ค.ศ. 1995 โดย Vapnik มีแนวคิดมาจากตัวแบบเครือข่ายประสาทเทียม ซึ่งเป็นการใช้ความรู้ทางด้านสถิติและกระบวนการลดความเสี่ยงเชิงโครงสร้างให้มีค่าต่ำที่สุด (Structural risk minimization: SRM) จัดเป็นเทคนิคการเรียนรู้แบบมีผู้สอน (Supervised learning) เพื่อให้สามารถสร้างตัวจำแนกประเภทข้อมูล (Classifier) ด้วยระนาบหลายมิติที่มีความเป็นนัยทั่วไป (Generalize) สูง กล่าวคือ สามารถทำงานได้ดีกับฐานข้อมูลตัวอย่างที่ไม่รู้จัก (Unknown database) ด้วยกระบวนการปรับรูปแบบข้อมูลจากข้อมูลที่มีมิติต่ำ (Low dimension dataset) บนพื้นที่ข้อมูลนำเข้า (Input space) ให้อยู่ในรูปแบบของข้อมูลที่มีมิติสูง (High dimension dataset) บนพื้นที่ข้อมูลคุณลักษณะ (Feature space หรือ High dimension space) โดยใช้ฟังก์ชันในการปรับรูปแบบข้อมูลเรียกว่า **ฟังก์ชันเคอร์เนล (Kernel function)** ซึ่งความสามารถดังกล่าวช่วยให้การสร้างตัวจำแนกข้อมูลด้วยสมการกำลังสอง (Quadratic equation) บนพื้นที่ข้อมูลคุณลักษณะเป็นไปได้ง่ายขึ้นและมีความชัดเจนในการจำแนกกลุ่มมากยิ่งขึ้นด้วย นอกจากนี้ตัวจำแนกข้อมูลที่ดีควรมีโครงสร้างแบบเส้นตรง (Linear classifier) และสามารถสร้างพื้นที่ระยะห่างระหว่างตัวจำแนกข้อมูลกับค่าที่ใกล้ที่สุดของแต่ละกลุ่มข้อมูลได้มากที่สุดเพื่อประสิทธิภาพในการแยกแยะประเภทของชุดข้อมูลแต่ละประเภทออกจากกันอย่างชัดเจน ซึ่งเส้นที่เหมาะสมดังกล่าวเรียกว่า **ระนาบแบ่งกลุ่มข้อมูลที่เหมาะสม (Optimal separating hyperplane)** โดยทั่วไปตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสามารถแบ่งได้เป็น 2 ประเภท คือ ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล (Support vector machine model for classification) และตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression)

2.3.1 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล (Support vector machine model for classification)

ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูล เป็นสมการเชิงคณิตศาสตร์ที่ใช้ในการจำแนกค่าคุณลักษณะของข้อมูล 2 กลุ่มที่วางตัวอยู่ในพื้นที่คุณลักษณะ (Feature space) ออกจากกันโดยจะสร้างเส้นแบ่ง (Hyperplane) ที่เป็นเส้นตรงขึ้นมาและเพื่อให้

ทราบว่าเส้นตรงที่แบ่งข้อมูล 2 กลุ่มออกจากกันนั้นเส้นตรงใดเป็นเส้นตรงที่ดีที่สุด โดยเส้นตรงนั้นจะเพิ่มเส้นขอบ (Margin) ออกไปทั้งสองข้าง โดยเส้นขอบที่เพิ่มนั้นจะขนานกับเส้นเดิมเสมอ เส้นขอบที่เพิ่มขึ้นมานี้จะขยายออกไปจนกว่าจะสัมผัสกับค่าของกลุ่มตัวอย่างที่ใกล้ที่สุด ดังรูป



ภาพที่ 2- 13 เส้นตรงที่แบ่งข้อมูลทั้ง 2 กลุ่มในตัวแบบซัพพอร์ทเวกเตอร์แมชชีน

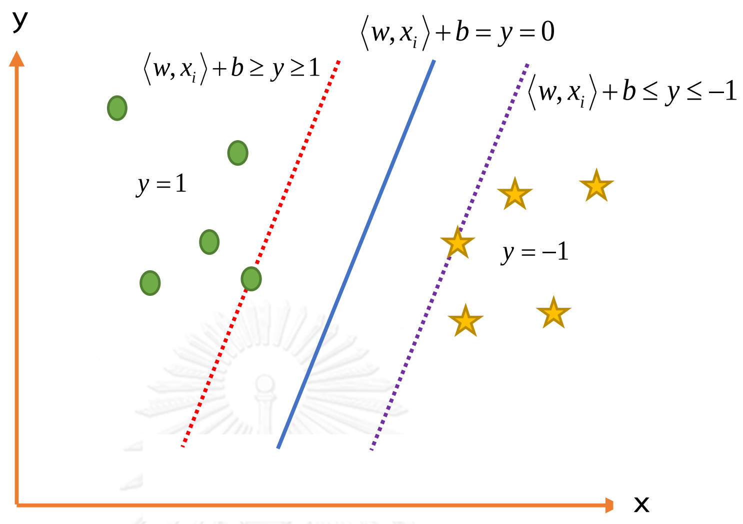
จากกราฟข้างต้น พบว่าเส้นตรง M_L และเส้นตรง M_R คือเส้นขอบที่ขยายออกไปด้านซ้ายและด้านขวาตามลำดับ และ P คือเส้นตรงที่แบ่งข้อมูลทั้ง 2 กลุ่ม เมื่อเส้นตรง M_L และเส้นตรง M_R ขยายออกไปจนสัมผัสค่าของข้อมูลที่ใกล้ที่สุด ซึ่งค่าของข้อมูลที่อยู่บนเส้นขอบของทั้งสองฝั่งนั้นเรียกว่า **ซัพพอร์ทเวกเตอร์ (Support vector)** จะวัดค่าระยะความห่างขอบเส้นขอบ โดยเส้น P จะเปลี่ยนความชันไปเรื่อย ๆ เพื่อที่จะหาความกว้างสูงสุดของเส้นขอบ (Maximum margin)

ดังนั้น กระบวนการโดยรวมของซัพพอร์ทเวกเตอร์แมชชีนจะเป็นการหาค่าความชันของเส้น P ที่มีขนาดของเส้นขอบสูงสุดนั่นเอง

2.3.1.1 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูลกรณีที่สามารถแบ่งกลุ่มข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน

ถ้านำแนวคิดของซัพพอร์ทเวกเตอร์แมชชีนมาเขียนเป็นสมการเชิงคณิตศาสตร์เพื่อใช้ในการแก้ปัญหาข้างต้น โดยข้อมูลที่นำมาวางลงในพื้นที่คุณลักษณะนั้นเป็นกลุ่มของข้อมูลที่อยู่ในรูปเวกเตอร์ $D = \{(x_1, y_1),\ldots,(x_l, y_l)\}$ เมื่อ $x \in R^n$ สำหรับ $i = 1,2,\ldots,l$ และ $y \in \{-1,1\}$ จะทำ

การหาระนาบที่แบ่งข้อมูลตัวอย่างที่เป็นบวกและข้อมูลตัวอย่างที่เป็นลบออกจากกันโดยข้อมูลตัวอย่างดังกล่าวจะถูกแบ่งด้วยเส้นตรง P บนไฮเปอร์เพลน (Hyperplane) ซึ่งแบ่งข้อมูลที่มีลักษณะเป็นเชิงเส้นตรงสองกลุ่มออกจากกันโดยที่ไม่มีค่าคลาดเคลื่อนจากการแบ่งกลุ่ม ดังรูป



ภาพที่ 2- 14 สมการที่ใช้แบ่งกลุ่มข้อมูลทั้ง 2 กลุ่มในตัวแบบซัพพอร์ทเวกเตอร์แมชชีน

จากกราฟข้างต้น เส้นตรง M_L แทนด้วยสมการ $\langle w, x_i \rangle + b \geq y \geq 1$ ซึ่งค่าของข้อมูล $y > 1$ ก็จะถูกกำหนดใหม่เป็น $y = 1$ เช่นเดียวกันกับเส้นตรง M_R ที่แทนด้วยสมการ $\langle w, x_i \rangle + b \leq y \leq -1$ ซึ่งค่าของข้อมูล $y < -1$ ก็จะถูกกำหนดใหม่เป็น $y = -1$ ดังนั้น สมการที่ใช้แบ่งกลุ่มข้อมูล คือ

$$\langle w, x_i \rangle + b \geq 1 \quad \text{สำหรับ} \quad y_i = 1 \tag{2-62}$$

และ

$$\langle w, x_i \rangle + b \leq 1 \quad \text{สำหรับ} \quad y_i = -1 \tag{2-63}$$

เมื่อ w แทน เวกเตอร์ค่าถ่วงน้ำหนัก (Weight)

b แทน ค่าความเอนเอียง (Bias)

x_i แทน เวกเตอร์ข้อมูลนำเข้า (Input data)

และ y_i แทน ค่ากลุ่มของข้อมูล โดยที่ $y_i \in \{-1,1\}$

เราสามารถนำสมการที่ (2-62) และ (2-63) ข้างต้นมาเขียนรวมกันได้เป็นสมการใหม่ ดังนี้

$$y_i[\langle w, x_i \rangle + b] \geq 1 \text{ สำหรับทุก } i = 1,2,\ldots,l \tag{2-64}$$

สำหรับการหาระนาบสำหรับการแบ่งกลุ่มที่เหมาะสม (Optimal separating hyperplane) ทำได้โดยการหาค่าระยะห่าง (Distance) จากตำแหน่งของซัพพอร์ทเวกเตอร์ x ถึงไฮเปอร์เพลน (w,b) เมื่อ

$$d(w,b;x)=\frac{|\langle w,x_i\rangle+b|}{\|w\|} \tag{2-65}$$

โดยการคำนวณหาค่าความกว้างของเส้นขอบที่มากที่สุด (Maximum margin) สามารถทำได้ดังนี้

$$\begin{aligned}\rho(w,b)&=\min_{x_i,y_i=-1} d(w,b;x_i)+\min_{x_i,y_i=+1} d(w,b;x_i)\\ &=\min_{x_i,y_i=-1}\frac{|\langle w,x_i\rangle+b|}{\|w\|}+\min_{x_i,y_i=+1}\frac{|\langle w,x_i\rangle+b|}{\|w\|}\\ &=\frac{1}{\|w\|}\left(\min_{x_i,y_i=-1}|\langle w,x_i\rangle+b|+\min_{x_i,y_i=+1}|\langle w,x_i\rangle+b|\right)\\ &=\frac{2}{\|w\|}\end{aligned} \tag{2-66}$$

พบว่า “ถ้าขนาดของเวกเตอร์ค่าถ่วงน้ำหนัก w มีค่าน้อยเท่าไหร่ ความกว้างของเส้นขอบก็จะมีค่ามากขึ้นเท่านั้น” ดังนั้น การหาระนาบสำหรับการแบ่งกลุ่มที่เหมาะสมจึงเป็นการหาค่าที่น้อยที่สุดของฟังก์ชันต่อไปนี้

$$Minimize\ \ \Phi(w)=\frac{1}{2}\|w\|^2 \tag{2-67}$$

$$Subject\ \ to\ \ y_i\left[\langle w,x_i\rangle+b\right]\geq 1 \text{ สำหรับทุก } i=1,2,\dots,l$$

ต้องการหาค่าเวกเตอร์ค่าถ่วงน้ำหนัก w ที่ทำให้ฟังก์ชัน $\Phi(w)$ มีค่าน้อยที่สุด ภายใต้เงื่อนไข $y_i\left[\langle w,x_i\rangle+b\right]\geq 1$ สำหรับทุก $i=1,2,\dots,l$ ดังนั้นจะใช้วิธีของลากรองจ์ (Lagrangian method) โดยใช้ตัวคูณลากรองจ์ (Lagrange multiplier) มาช่วยหาค่าถ่วงน้ำหนัก w ซึ่งทำให้ L มีค่าน้อยที่สุด นั่นคือ

$$L=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}\alpha_i\left(y_i\left[\langle w,x_i\rangle+b\right]-1\right) \tag{2-68}$$

$$Subject\ \ to\ \ \alpha_i \geq 0 \text{ สำหรับทุก } i = 1,2,...,l$$

เมื่อ $\alpha = (\alpha_1,...,\alpha_l)'$ แทน ตัวคูณลากรองจ์

ทำการหาอนุพันธ์เทียบกับ w และ b แล้วกำหนดให้เท่ากับ 0 จะได้

$$\frac{\partial L}{\partial w} = 0 \quad \text{จะได้} \quad w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{2-69}$$

$$\frac{\partial L}{\partial b} = 0 \quad \text{จะได้} \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2-70}$$

จากสมการที่ (2-68), (2-69) และ (2-70) โดยเปลี่ยนรูปแบบของปัญหาเดิม (Primal problem) ให้เป็นรูปแบบของปัญหาควบคู่ (Dual problem) ได้ดังนี้

$$Maximize\ -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l}\alpha_k \tag{2-71}$$

$$Subject\ \ to\ \ \alpha_i \geq 0 \text{ และ } \sum_{i=1}^{l}\alpha_i y_i = 0 \text{ สำหรับทุก } i = 1,2,...,l$$

สำหรับค่าประมาณของตัวคูณลากรองก์สามารถหาได้จาก

$$\alpha^* = \arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{k=1}^{l}\alpha_k\right) \tag{2-72}$$

พิจารณาขอบเขตที่ได้ของค่าประมาณของตัวคูณลากรองจ์จากการแก้สมการที่ (2-72) ภายใต้เงื่อนไข $\alpha_i \geq 0$ และ $\sum_{i=1}^{l}\alpha_i y_i = 0$ สำหรับทุก $i = 1,2,...,l$ เพื่อหาตำแหน่งของซัพพอร์ทเวกเตอร์จากข้อมูลในแต่ละกลุ่มที่มีค่าประมาณของตัวลากรองจ์ไม่เท่ากับศูนย์ พบว่า

$$w^* = \sum_{i=1}^{l}\alpha_i y_i x_i \tag{2-73}$$

และ

$$b^* = -\frac{1}{2}\langle w^*, x_r + x_s \rangle \tag{2-74}$$

เมื่อ x_r และ x_s แทน ซัพพอร์ทเวกเตอร์จากข้อมูลในแต่ละกลุ่ม

ดังนั้น สมการที่เหมาะสมสำหรับการแบ่งกลุ่มข้อมูลบนไฮเปอร์เพลน คือ

$$f(x) = \operatorname{sgn}\left(\langle w^*, x\rangle + b\right) \qquad (2\text{-}75)$$

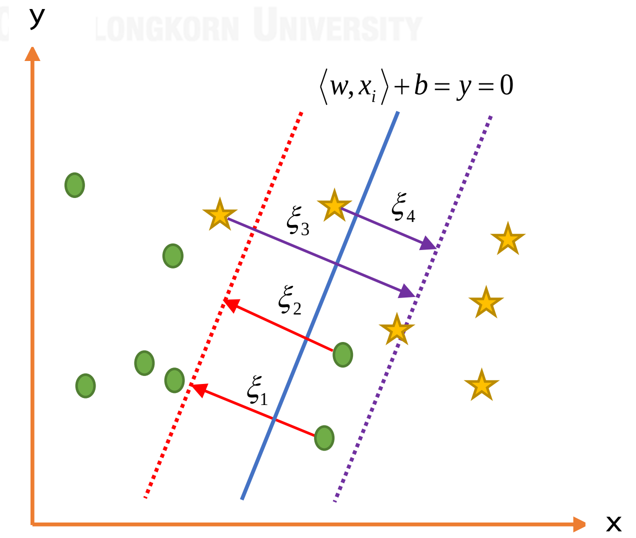
หรือสามารถเขียนได้ให้อยู่ในรูปของ

$$f(x) = h\left(\langle w^*, x\rangle + b\right) \qquad (2\text{-}76)$$

เมื่อ $h(z) = \begin{cases} -1 & \text{เมื่อ } z < -1 \\ z & \text{เมื่อ } -1 \le z \le 1 \\ +1 & \text{เมื่อ } z > 1 \end{cases}$

2.3.1.2 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูลกรณีที่ไม่สามารถแบ่งกลุ่มข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน

โดยทั่วไปเราสามารถทำการหาได้เพียงแค่ไฮเปอร์เพลนที่สามารถแบ่งกลุ่มข้อมูลด้วยเส้นตรงออกจากกันให้ได้มากที่สุด และมีข้อมูลบางส่วนที่เกิดค่าคลาดเคลื่อนจากการแบ่งกลุ่ม นั่นคือ โครงสร้างของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการแก้ปัญหาในกรณีนี้ประกอบด้วย 2 ส่วนหลัก คือ การเพิ่มความกว้างของเส้นขอบให้มากที่สุด และการลดค่าคลาดเคลื่อนให้มีค่าน้อยที่สุด



ภาพที่ 2- 15 ตัวแปรที่ทำให้เกิดค่าคลาดเคลื่อนจากการแบ่งกลุ่มข้อมูล

กำหนดให้ $\xi = (\xi_1, \xi_2, ..., \xi_n)^T$ แทน เวกเตอร์ของตัวแปรที่ทำให้เกิดค่าคลาดเคลื่อนจากการแบ่งกลุ่มข้อมูล (Misclassification errors หรือ Slack variables) ภายใต้ Penalty function

$$F_\sigma(\xi) = \sum_{i=1}^{l} \xi_i^\sigma \text{ , } \sigma > 0 \qquad (2\text{-}77)$$

ดังนั้น สมการที่ใช้แบ่งกลุ่มข้อมูลในกรณีที่ไม่สามารถแบ่งกลุ่มข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน คือ

$$y_i[\langle w, x_i \rangle + b] \geq 1 - \xi_i \text{ และ } \xi_i \geq 0 \text{ สำหรับทุก } i = 1,2,...,l \qquad (2\text{-}78)$$

เมื่อ w แทน เวกเตอร์ค่าถ่วงน้ำหนัก (Weight)

b แทน ค่าความเอนเอียง (Bias)

x_i แทน เวกเตอร์ข้อมูลนำเข้า (Input data)

y_i แทน ค่ากลุ่มของข้อมูล โดยที่ $y_i \in \{-1,1\}$

และ ξ_i แทน ตัวแปรที่ทำให้เกิดค่าคลาดเคลื่อนจากการแบ่งกลุ่มข้อมูล

สำหรับการหาระนาบสำหรับการแบ่งกลุ่มข้อมูลที่เหมาะสม (Optimal separating hyperplane) ในกรณีที่ $\sigma = 1$ จะเป็นการหาค่าที่น้อยที่สุดของฟังก์ชันต่อไปนี้

$$Minimize\ \Phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \qquad (2\text{-}79)$$

$$Subject\ to\ y_i[\langle w, x_i \rangle + b] \geq 1 - \xi_i \text{ และ } \xi_i \geq 0 \text{ สำหรับทุก } i = 1,2,...,l$$

โดยที่ C แทน ค่าคงที่ซึ่งทำหน้าที่ปรับความสมดุลระหว่างการให้ความสำคัญกับการเพิ่มความกว้างของเส้นขอบให้มากที่สุดและการลดค่าคลาดเคลื่อนที่เกิดขึ้นจากการแบ่งกลุ่มข้อมูลให้มีค่าน้อยที่สุด ซึ่งเป็นค่าที่ผู้วิจัยสามารถกำหนดค่าได้เองโดยปกติจะกำหนดให้มีค่ามากกว่า 1 และค่า C ยังทำให้ตัวแบบมีความคงทนต่อค่านอกเกณฑ์ (Outlier) อีกด้วย

ต้องการหาค่าเวกเตอร์ w และ ξ ที่ทำให้ฟังก์ชัน $\Phi(w,\xi)$ มีค่าน้อยที่สุด ภายใต้เงื่อนไข $y_i[\langle w, x_i \rangle + b] \geq 1 - \xi_i$ และ $\xi_i \geq 0$ สำหรับทุก $i = 1,2,...,l$ ดังนั้นจะใช้วิธีของลากรองจ์

(Lagrangian method) โดยใช้ตัวคูณลากรองจ์ (Lagrange multiplier) มาช่วยหาค่าเวกเตอร์ w และ ξ ซึ่งทำให้ L มีค่าน้อยที่สุด นั่นคือ

$$L=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\alpha_i\left(y_i\left[\langle w,x_i\rangle+b\right]-1+\xi_i\right)-\sum_{i=1}^{l}\beta_i\xi_i \tag{2-80}$$

$$\textit{Subject to } \alpha_i \ge 0 \text{ และ } \beta_i \ge 0 \text{ สำหรับทุก } i=1,2,\ldots,l$$

เมื่อ $\alpha=(\alpha_1,\ldots,\alpha_l)'$ และ $\beta=(\beta_1,\ldots,\beta_l)'$ แทน ตัวคูณลากรองจ์

ทำการหาอนุพันธ์เทียบกับ w,ξ และ b แล้วกำหนดให้เท่ากับ 0 จะได้

$$\frac{\partial L}{\partial w}=0 \quad \text{จะได้} \quad w=\sum_{i=1}^{l}\alpha_i y_i x_i \tag{2-81}$$

$$\frac{\partial L}{\partial b}=0 \quad \text{จะได้} \quad \sum_{i=1}^{l}\alpha_i y_i=0 \tag{2-82}$$

$$\frac{\partial L}{\partial \xi}=0 \quad \text{จะได้} \quad \alpha_i+\beta_i=C \tag{2-83}$$

จากสมการที่ (2-80), (2-81), (2-82) และ (2-83) โดยเปลี่ยนรูปแบบของปัญหาเดิม (Primal problem) ให้เป็นรูปแบบของปัญหาควบคู่ (Dual problem) ได้ดังนี้

$$\textit{Maximize } -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle+\sum_{k=1}^{l}\alpha_k \tag{2-84}$$

$$\textit{Subject to } 0\le\alpha_i\le C \text{ และ } \sum_{i=1}^{l}\alpha_i y_i=0 \text{ สำหรับทุก } i=1,2,\ldots,l$$

สำหรับค่าประมาณของตัวคูณลากรองก์สามารถหาได้จาก

$$\alpha^*=\arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j\langle x_i,x_j\rangle-\sum_{k=1}^{l}\alpha_k\right) \tag{2-85}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.3.1.3 สมการพื้นฐานของตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการจำแนกประเภทข้อมูลกรณีทำการแบ่งกลุ่มข้อมูลทั้งหมดโดยใช้ระนาบแบบไม่เป็นเชิงเส้นตรง

ในความเป็นจริงนั้นข้อมูล 2 กลุ่มไม่ได้วางตัวในพื้นที่คุณลักษณะและไม่สามารถแบ่งได้ด้วยเส้นตรง แต่ข้อมูลอาจจะจับกลุ่มกันในตำแหน่งต่าง ๆ ทำให้ไม่สามารถที่จะใช้สมการของซัพพอร์ทเวกเตอร์แมชชีนแบบเชิงเส้นได้ ดังนั้นจึงต้องหาเครื่องมือเพื่อเข้ามาช่วยในการแปลงข้อมูลเหล่านั้นจากพื้นที่ข้อมูลนำเข้า (Input space) ให้ข้อมูลมีเรียงตัวใหม่ในพื้นที่หลายมิติ (Higher dimensional space) หรือพื้นที่คุณลักษณะ (Feature space) เครื่องมือนั้นเรียกว่า **"ฟังก์ชันเคอร์เนล" (Kernel function)** ซึ่งนิยามโดย

$$K\left(x, x^{'}\right) = \left\langle \phi(x), \phi\left(x^{'}\right) \right\rangle \tag{2-86}$$

กล่าวคือ ค่าผลคูณภายใน (Inner product) บนพื้นที่ข้อมูลคุณลักษณะ (Feature space) จะมีค่าเท่ากับฟังก์ชันเคอร์เนลบนพื้นที่ข้อมูลนำเข้า (Input space)

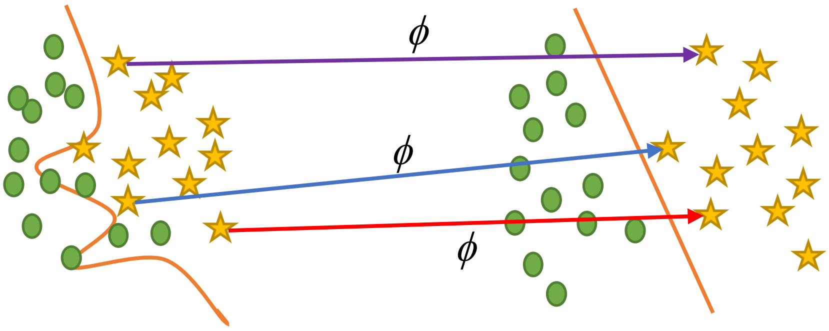
เงื่อนไขของเมอร์เชอร์ (Mercer's conditions)

ถ้า K เป็นฟังก์ชัน Symmetric positive definite บน Reproducing Kernel Hilbert Spaces (RKHS) แล้วเราสามารถนิยามฟังก์ชันเคอร์เนล ได้ดังนี้

$$K\left(x, x^{'}\right) = \sum_{m}^{\infty} a_m \phi_m(x) \phi_m\left(x^{'}\right),\ a_m \geq 0 \tag{2-87}$$

$$\iint K\left(x, x^{'}\right) g(x) g\left(x^{'}\right) dx dx^{'} > 0\ ,\ g \in L^2 \tag{2-88}$$

เมื่อ L^2 แทน ฟังก์ชันที่สามารถแยกองค์ประกอบการคูณกันเป็นสององค์ประกอบได้



ภาพที่ 2- 16 การแปลงข้อมูลโดยใช้ฟังก์ชันเคอร์เนลจากพื้นที่ข้อมูลนำเข้าให้ข้อมูลมีเรียงตัวใหม่ในพื้นที่คุณลักษณะ

ประเภทของฟังก์ชันเคอร์เนล

1. Polynomial

$$K(x,x^{'}) = \langle x, x^{'} + 1 \rangle^{d}$$

2. Gaussian radial basis function

$$K(x,x^{'}) = \exp\left(-\frac{\|x - x^{'}\|^{2}}{2\sigma^{2}}\right)$$

3. Exponential radial basis function

$$K(x,x^{'}) = \exp\left(-\frac{\|x - x^{'}\|}{2\sigma^{2}}\right)$$

4. Multi-layer perceptron

$$K(x,x^{'}) = \tanh\left(\rho\langle x, x^{'} \rangle + \varsigma\right)$$

5. Fourier series

$$K(x,x')=\frac{\sin\left(N+\frac{1}{2}\right)(x-x')}{\sin\left(\frac{1}{2}(x-x')\right)}$$

6. Additive kernel

$$K(x,x')=\sum_{i}K_i(x,x')$$

7. Tensor product

$$K(x,x')=\prod_{i}K_i(x_i,x_i')$$

8. Splines

$$K(x,x')=\sum_{r=0}^{k}x^r x'^r+\sum_{s=1}^{N}(x-\tau_s)_+^k(x'-\tau_s)_+^k$$

9. B splines

$$K(x,x')=B_{2N+1}(x-x')$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ดังนั้น จากฟังก์ชันเคอร์เนลข้างต้นเราสามารถแทนค่าฟังก์ชันดังกล่าวลงไปในตำแหน่งของ $\langle x_i, x_j\rangle$ ในสมการที่ (2-84) ได้เป็นรูปแบบของปัญหาควบคู่ (Dual problem) ดังนี้

$$Maximize\ -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_{k=1}^{l}\alpha_k \qquad (2\text{-}89)$$

$Subject\ \ to\ \ 0\le\alpha_i\le C$ และ $\sum_{i=1}^{l}\alpha_i y_i=0$ สำหรับทุก $i=1,2,\ldots,l$

สำหรับค่าประมาณของตัวคูณลากรองก์สามารถหาได้จาก

$$\alpha^* = \arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{j}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{k=1}^{l}\alpha_k\right) \tag{2-90}$$

ดังนั้น สมการที่เหมาะสมสำหรับการแบ่งกลุ่มข้อมูลบนไฮเปอร์เพลน คือ

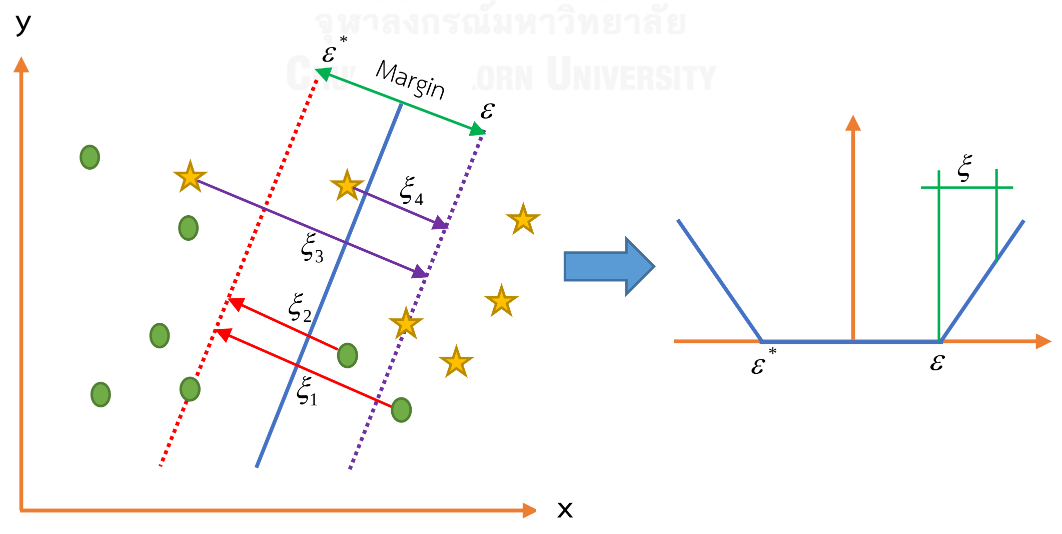
$$f(x) = \operatorname{sgn}\left(\sum_{i\in SVs}\langle w^*, x\rangle + b\right) \tag{2-91}$$

โดยที่ $\langle w^*, x\rangle = \sum_{i=1}^{l}\alpha_i y_i K(x_i, x)$ และ $b^* = -\frac{1}{2}\sum_{i=1}^{l}\alpha_i y_i\left[K(x_i, x_r) + K(x_i, x_r)\right]$

2.3.2 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression)

ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนนอกจากใช้สำหรับการจำแนกประเภทข้อมูล (Classification) ดังที่กล่าวมาแล้วข้างต้นยังสามารถนำมาประยุกต์ใช้สำหรับการวิเคราะห์การถดถอย (Regression analysis) และการวิเคราะห์อนุกรมเวลา (Time series analysis) ได้โดยมีวัตถุประสงค์เพื่อพยากรณ์สิ่งที่เกิดขึ้นในอนาคตโดยใช้ข้อมูลที่มีอยู่ในอดีต

จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 2- 17 ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอยกรณีที่ไม่สามารถแยกแยะข้อมูลทั้งหมดได้ด้วยเส้นตรงบนไฮเปอร์เพลน เมื่อใช้ฟังก์ชันการสูญเสียไม่ไวแบบเอ็บซิลอน

กำหนดข้อมูลนำเข้าอยู่ในรูปเวกเตอร์ $D = \{(x_1, y_1),...,(x_l, y_l)\}$ เมื่อ $x \in R^n, y \in R$ สำหรับ $i = 1,2,...,l$ การหาค่าถ่วงน้ำหนักที่เหมาะสมที่สุดสามารถหาได้จากการปรับค่าที่เหมาะสมของฟังก์ชัน $\Phi(w)$ ตามกระบวนการลดความเสี่ยงเชิงโครงสร้างให้มีค่าต่ำที่สุด (Structural risk minimization: SRM) ซึ่งประกอบไปด้วย 2 ส่วนหลัก คือ การเพิ่มความกว้างของเส้นขอบให้มากที่สุด (Maximum margin) และการลดค่าคลาดเคลื่อนโดยประสบการณ์ (Empirical error) ให้มีค่าน้อยที่สุด โดยที่

$$\Phi(w) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l} L_\varepsilon(y_i - f(x_i, w))\right) \qquad (2\text{-}92)$$

โดยที่ $L_\varepsilon(y_i - f(x_i, w))$ แทน ฟังก์ชันการสูญเสียไม่ไวแบบเอ็บซิลอน (ε -insensitive loss function) ซึ่งนิยามโดย

$$L_\varepsilon(y_i - f(x_i, w)) = \begin{cases} 0 & \text{สำหรับ } |y_i - f(x_i, w)| \le \varepsilon \\ |y_i - f(x_i, w)| - \varepsilon & \text{กรณีอื่น ๆ} \end{cases} \qquad (2\text{-}93)$$

กำหนดให้ ξ_i และ ξ_i^* แทน ตัวแปรที่ทำให้เกิดค่าคลาดเคลื่อน (Slack variables) และอยู่ห่างจากระยะขอบเอ็บซิลอน (ε -tube) ซึ่งสมการที่ (2-92) ข้างต้นสามารถเขียนใหม่ได้เป็น

$$\Phi(w, \xi_i, \xi_i^*) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i^*\right) \qquad (2\text{-}94)$$

สำหรับการหาฟังก์ชันการถดถอยที่เหมาะสมที่สุด (Optimal regression function) จะเป็นการหาค่าที่น้อยที่สุดของฟังก์ชันต่อไปนี้

$$Minimize\ \Phi(w, \xi_i, \xi_i^*) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i^*\right) \qquad (2\text{-}95)$$

$$Subject\ \ to\ \ y_i[\langle w, x_i\rangle + b] \ge 1 - \xi_i,\ \ y_i[\langle w, x_i\rangle + b] \ge 1 - \xi_i^* \ \text{และ}\ \ \xi_i, \xi_i^* \ge 0$$

ต้องการหาค่าเวกเตอร์ w และ ξ^* ที่ทำให้ฟังก์ชัน $\Phi(w, \xi_i, \xi_i^*)$ มีค่าน้อยที่สุด ภายใต้เงื่อนไข $y_i[\langle w, x_i\rangle + b] \geq 1 - \xi_i$ และ $\xi_i, \xi_i^* \geq 0$ สำหรับทุก $i = 1,2,...,l$ ดังนั้นจะใช้วิธีของลากรองจ์ (Lagrangian method) โดยใช้ตัวคูณลากรองจ์ (Lagrange multiplier) มาช่วยหาค่าเวกเตอร์ w และ ξ^* ซึ่งทำให้ L มีค่าน้อยที่สุด เมื่อ

$$L = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i^*\right) - \sum_{i=1}^{l}\alpha_i^*\left(y_i - \langle w, x_i\rangle - b + \varepsilon + \xi_i^*\right)$$
$$- \sum_{i=1}^{l}\alpha_i\left(\langle w, x_i\rangle + b - y_i + \varepsilon + \xi_i\right) - \sum_{i=1}^{l}\left(\beta_i^*\xi_i^* + \beta_i\,\xi_i\right) \qquad (2\text{-}96)$$

Subject to $\alpha_i, \alpha_i^* \geq 0$ และ $\beta_i, \beta_i^* \geq 0$ สำหรับทุก $i = 1,2,...,l$

ทำการหาอนุพันธ์เทียบกับ w, ξ^* และ b แล้วกำหนดให้เท่ากับ 0 จะได้

$$\frac{\partial L}{\partial w} = 0 \quad \text{จะได้} \quad w = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)x_i \qquad (2\text{-}97)$$

$$\frac{\partial L}{\partial \xi^*} = 0 \quad \text{จะได้} \quad \alpha_i^* + \beta_i^* = C \qquad (2\text{-}98)$$

$$\frac{\partial L}{\partial b} = 0 \quad \text{จะได้} \quad \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) = 0 \qquad (2\text{-}99)$$

พิจารณาจากสมการที่ (2-96), (2-97), (2-98) และ (2-99) โดยเปลี่ยนรูปแบบของปัญหาเดิม (Primal problem) ให้เป็นรูปแบบของปัญหาควบคู่ (Dual problem) สำหรับการถดถอยน้อยที่สุด (Minimize regression) ดังนี้

$$\textit{Maximize} \quad -\varepsilon\sum_{i=1}^{l}\left(\alpha_i + \alpha_i^*\right) + \sum_{i=1}^{l}\left(\alpha_i + \alpha_i^*\right)y_i - \frac{1}{2}\sum_{j=1}^{l}\sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)\langle x_i, x_j\rangle \qquad (2\text{-}100)$$

Subject to $0 \leq \alpha_i, \alpha_i^* \leq C$ และ $\sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^*$ สำหรับทุก $i = 1,2,...,l$

ดังนั้น สมการการถดถอยที่เหมาะสมสำหรับพยากรณ์ข้อมูลบนไฮเปอร์เพลน คือ

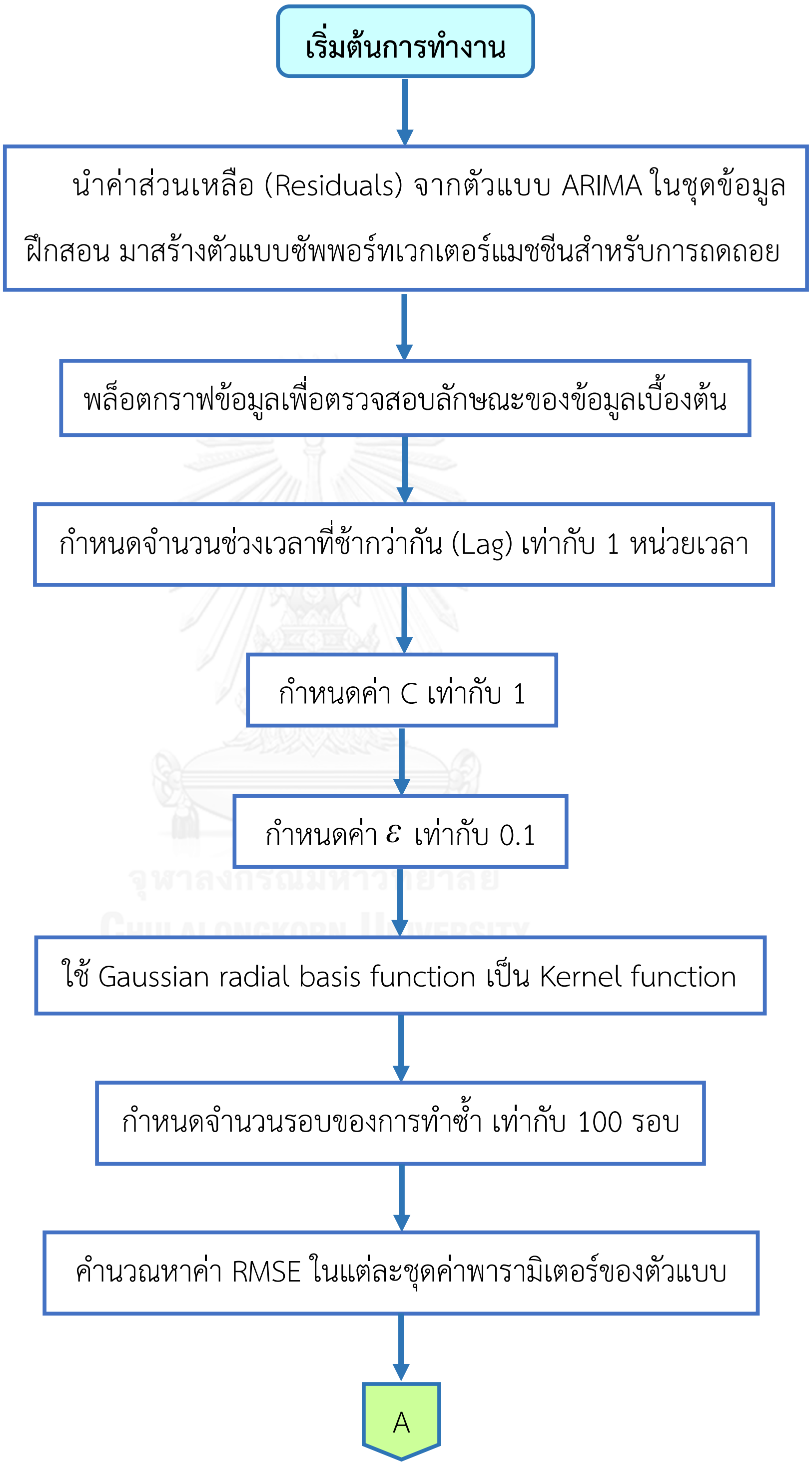
$$f(x) = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)\langle x_i, x\rangle + b \qquad (2\text{-}101)$$

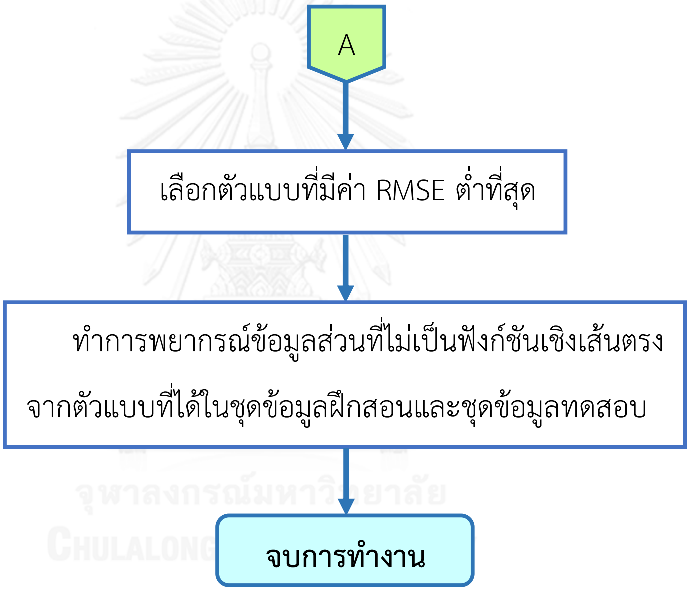
พิจารณาฟังก์ชันเคอร์เนลข้างต้นเราสามารถแทนค่าฟังก์ชันดังกล่าวลงไปในตำแหน่งของ $\langle x_i , x \rangle$ ในสมการที่ (2-101) ได้เพื่อช่วยในการแปลงข้อมูลนำเข้าที่มีลักษณะไม่เป็นเชิงเส้นตรงบนพื้นที่ข้อมูลนำเข้าให้ข้อมูลมีเรียงตัวใหม่ที่มีลักษณะเป็นเชิงเส้นตรงบนพื้นที่หลายมิติหรือพื้นที่คุณลักษณะ จะได้สมการการถดถอยที่เหมาะสมสำหรับพยากรณ์ข้อมูลบนไฮเปอร์เพลน คือ

$$f(x) = \sum_{i=1}^{l} \left(\alpha_i - \alpha_i^*\right) K\left(x, x_i\right) + b \qquad (2\text{-}102)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.3.3 แผนภาพขั้นตอนการวิเคราะห์ข้อมูลด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีน




A
เลือกตัวแบบที่มีค่า RMSE ต่ำที่สุด
ทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง
จากตัวแบบที่ได้ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ
จบการทำงาน

2.4 ตัวแบบผสม (Hybrid model)

ตัวแบบผสม ถูกคิดค้นขึ้นในปี ค.ศ. 2003 โดย Zhang เป็นตัวแบบที่ได้รับความนิยมสำหรับใช้ในการพยากรณ์ข้อมูลอนุกรมเวลาทางธุรกิจ เช่น ด้านการเงิน โดยการนำตัวแบบที่ใช้สำหรับพยากรณ์ข้อมูลอนุกรมเวลาในส่วนที่เป็นฟังก์ชันเชิงเส้นตรง คือ ตัวแบบ ARIMA มารวมกับตัวแบบที่ใช้สำหรับพยากรณ์ข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง คือ ตัวแบบเครือข่ายประสาทเทียมหรือตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเพื่อทำให้ค่าพยากรณ์ที่ได้มีความถูกต้องและมีความแม่นยำมากยิ่งขึ้น ซึ่งตัวแบบผสมสามารถแสดงได้ด้วยสมการเชิงคณิตศาสตร์ ดังนี้

$$y_t = L_t + N_t + \varepsilon_t \qquad (2\text{-}103)$$

เมื่อ y_t แทน ค่าสังเกตของข้อมูลอนุกรมเวลา ณ เวลา t

L_t แทน ข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรง ณ เวลา t

N_t แทน ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง ณ เวลา t

และ ε_t แทน ค่าคลาดเคลื่อน ณ เวลา t

2.4.1 ขั้นตอนการสร้างตัวแบบผสม

1. ทำการวิเคราะห์ข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA เพื่อพยากรณ์ข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรง คือ $\hat{L}_t$

2. คำนวณค่าส่วนเหลือจากตัวแบบ ARIMA โดยที่ $\hat{\varepsilon}_t = y_t - \hat{L}_t$

3. นำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA ไปพยากรณ์ข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง คือ $\hat{N}_t$ ด้วยตัวแบบเครือข่ายประสาทเทียมหรือตัวแบบซัพพอร์ทเวกเตอร์แมชชีน

4. คำนวณหาค่าพยากรณ์รวม (Total Forecasting) จากสมการ $\hat{y}_t = \hat{L}_t + \hat{N}_t$

บทที่ 3

วิธีการดำเนินการศึกษา

ในงานวิจัยนี้มีวัตถุประสงค์เพื่อเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) โดยใช้ชุดข้อมูลจริงจำนวน 456 สัปดาห์ ที่เก็บรวบรวมมาจากเว็บไซต์ http://finance.yahoo.com/q?s=SCB.BK และชุดข้อมูลจำลองภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ โดยตัวแบบใดที่มีค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ให้ผลดีสุด ซึ่งทำการวิเคราะห์ข้อมูลทั้งหมดด้วยโปรแกรม R เวอร์ชัน 3.3.1 และโปรแกรม SAS เวอร์ชัน 9.3 ภายใต้ขอบเขตและวิธีการดำเนินการดังนี้

3.1 ขอบเขตของการวิจัย

ในการศึกษาครั้งนี้จะทำการศึกษาในส่วนของข้อมูลจริงและข้อมูลจำลอง ภายใต้ขอบเขตการวิจัยดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.1.1 ข้อมูลจริง

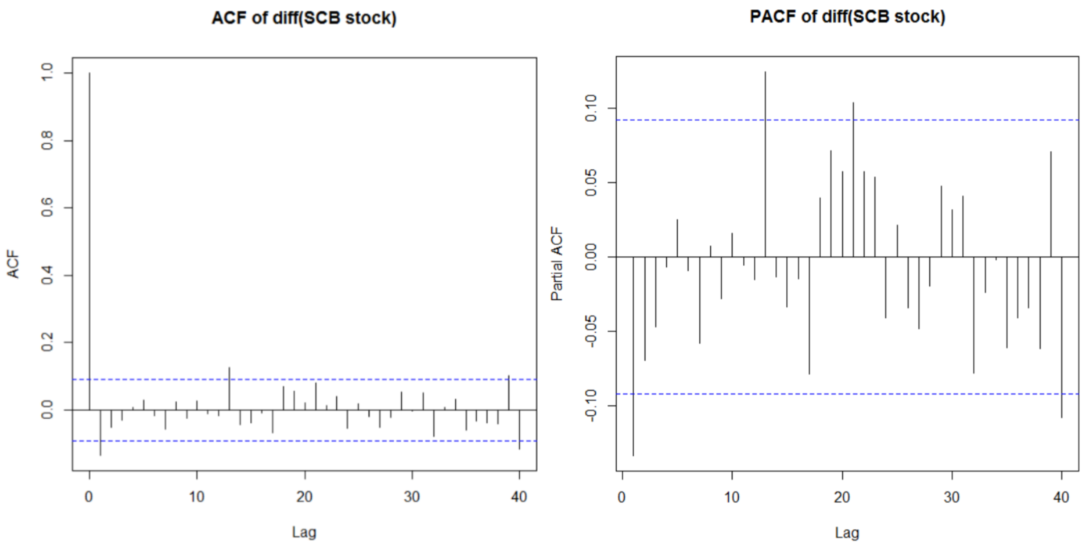
ชุดข้อมูลราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ตั้งแต่วันที่ 2 เดือนกรกฎาคม พ.ศ. 2550 ถึงวันที่ 8 เดือนกุมภาพันธ์ พ.ศ. 2559 จำนวน 456 สัปดาห์ ชุดข้อมูลนี้ได้มาจาก http://finance.yahoo.com/q?s=SCB.BK โดยทำการศึกษาและเปรียบเทียบตัวแบบภายใต้ขอบเขตของการวิจัย ดังนี้

3.1.1.1 ตัวแบบ ARIMA(p,d,q) ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 กล่าวคือ จะทำการเปรียบเทียบตัวแบบทั้งหมดจำนวน 9 ตัวแบบ คือ ตัวแบบ ARIMA(0,1,0), ตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อ

ข้อมูลราคาปิดหุ้น SCB เป็นแบบคงที่ (Stationary) และไม่มีแนวโน้ม (Trend) จากนั้นจะทำการพิจารณาคัดเลือกตัวแบบข้างต้นโดยตัวแบบที่เหมาะสมที่สุด คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด ซึ่งแสดงได้ดังนี้

p \ q	0	1	2
0	2,045.402	1,775.070	1,764.654
1	1,911.304	1,764.347	1,766.271
2	1,845.700	1,766.208	1,765.949

ตารางที่ 3- 1 เกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ของตัวแบบ ARIMA(p,d=1,q)



ภาพที่ 3- 1 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสหสัมพันธ์บางส่วน (PACF) ของตัวแบบ ARIMA(1,1,1)

3.1.1.2 ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม โดยนำค่าส่วนเหลือ (Residuals) ที่ได้จากตัวแบบ ARIMA ไปพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัว

แบบเครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนดตั้งแต่ 1 ถึง 5 โหนดโดยใช้ Sigmoid logistics function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดจำนวนรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ กล่าวคือ จะทำการเปรียบเทียบตัวแบบเครือข่ายประสาทเทียมทั้งหมดจำนวน 5 ตัวแบบ คือ ตัวแบบเครือข่ายประสาทเทียม 1-1-1, ตัวแบบเครือข่ายประสาทเทียม 1-2-1, ตัวแบบเครือข่ายประสาทเทียม 1-3-1, ตัวแบบเครือข่ายประสาทเทียม 1-4-1 และตัวแบบเครือข่ายประสาทเทียม 1-5-1 จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมข้างต้นโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า

3.1.1.3 ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยนำค่าส่วนเหลือ (Residuals) ที่ได้จากตัวแบบ ARIMA ไปพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอยภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา เมื่อกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial basis function เป็น Kernel function และกำหนดจำนวนรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนข้างต้นโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.1.2 ข้อมูลจำลอง

ทำการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อชุดข้อมูลจำลองอนุกรมเวลาเป็นแบบคงที่ (Stationary) และไม่มีแนวโน้ม (Trend) โดยกระบวนการจำลองจะนำค่าพารามิเตอร์ที่ Estimate ได้จากการสร้างตัวแบบ ARIMA ในชุดข้อมูลจริงของราคาปิดหุ้น SCB มาใช้สำหรับการจำลอง โดยแต่ละตัวแบบจะทำการจำลองชุดข้อมูลอนุกรมเวลาภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ จากนั้นนำค่าส่วนเหลือ (Residuals) ที่ได้จากชุดข้อมูลจำลองไปพยากรณ์ด้วยตัวแบบผสมระหว่าง ARIMA

กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ภายใต้ขอบเขตของการวิจัยในข้อมูลจริงข้างต้น

3.2 ขั้นตอนในการดำเนินการศึกษา

1. ศึกษาตัวแบบและทฤษฎีที่เกี่ยวข้อง

2. เก็บรวบรวมข้อมูลจริงราคาปิดหุ้น SCB รายสัปดาห์ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ตั้งแต่วันที่ 2 เดือนกรกฎาคม พ.ศ. 2550 ถึงวันที่ 8 เดือนกุมภาพันธ์ พ.ศ. 2559 จำนวน 456 สัปดาห์ จากเว็บไซต์ http://finance.yahoo.com/q?s=SCB.BK โดยแบ่งชุดข้อมูลราคาปิดหุ้น SCB ออกเป็นชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับประเมินความถูกต้องของตัวแบบ

3. ทำการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) ซึ่งในแต่ละตัวแบบจะทำการจำลองชุดข้อมูลอนุกรมเวลาภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยในแต่ละรอบของการจำลองจะทำการจำลองชุดข้อมูลอนุกรมเวลาจำนวน 456 สัปดาห์ และแบ่งออกเป็นชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับประเมินความถูกต้องของตัวแบบ และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ และคำนวณหาค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากการจำลองข้อมูลในแต่ละรอบ

4. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนจากข้อมูลจำลองในแต่ละรอบ มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network model) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัวแบบ

เครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนด 1 ถึง 5 โหนดโดยใช้ Sigmoid logistic function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบเครือข่ายประสาทเทียมในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจำลองในแต่ละรอบ หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) ในแต่ละรอบของการจำลอง หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนจากข้อมูลจำลองในแต่ละมาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression) ภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา และกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial function เป็น Kernel function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจำลองในแต่ละรอบ หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรง

ในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) ในแต่ละรอบของการจำลอง หลังจากนั้นคำนวณหาค่าเฉลี่ยของค่าคลาดเคลื่อนกำลังสองเฉลี่ย, ค่าเฉลี่ยของรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย และค่าเฉลี่ยของค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย จากการจำลองทั้งหมด 1,000 รอบ

6. นำชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์จากข้อมูลจริงมาสร้างตัวแบบ ARIMA ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 จากนั้นทำการพิจารณาคัดเลือกตัวแบบข้างต้นโดยตัวแบบที่เหมาะสมที่สุด คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด

7. ทำการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริง ซึ่งจัดเป็นข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE) หลังจากนั้นคำนวณหาค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น

8. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feedforward artificial neural network model) โดยใช้เทคนิคการแพร่แบบย้อนกลับ (Back-propagation) เป็นเทคนิคการเรียนรู้เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01 ซึ่งตัวแบบเครือข่ายประสาทเทียมประกอบด้วยชั้นข้อมูลนำเข้า (Input layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด, ชั้นซ่อน (Hidden layer) 1 ชั้นภายใต้จำนวนโหนด 1 ถึง 5 โหนดโดยใช้ Sigmoid logistic function เป็น Activation function และชั้นผลลัพธ์ (Output layer) 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activation function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบเครือข่ายประสาทเทียมโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ด้วยตัวแบบเครือข่ายประสาทเทียม

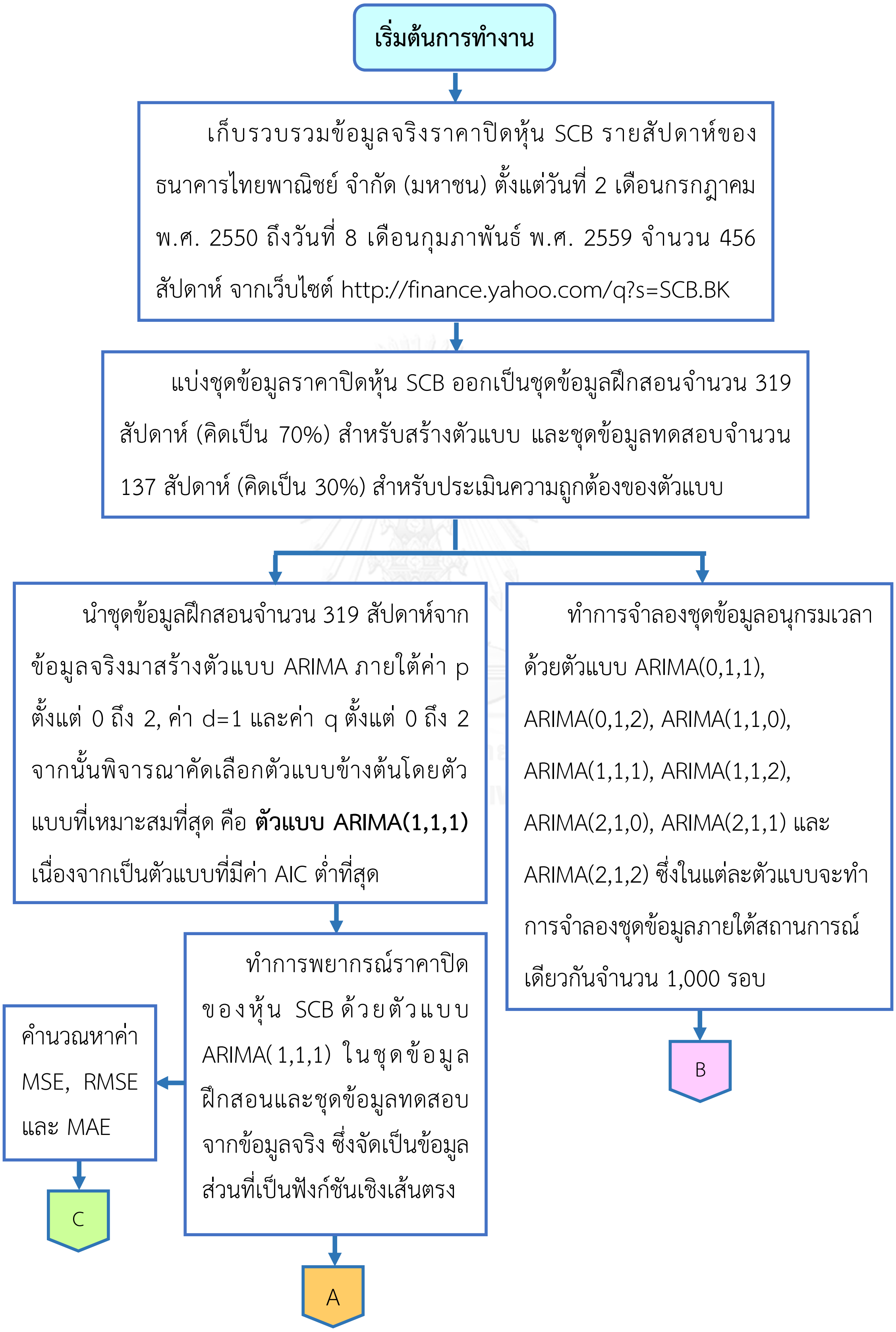
ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบเครือข่ายประสาทเทียมเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE)

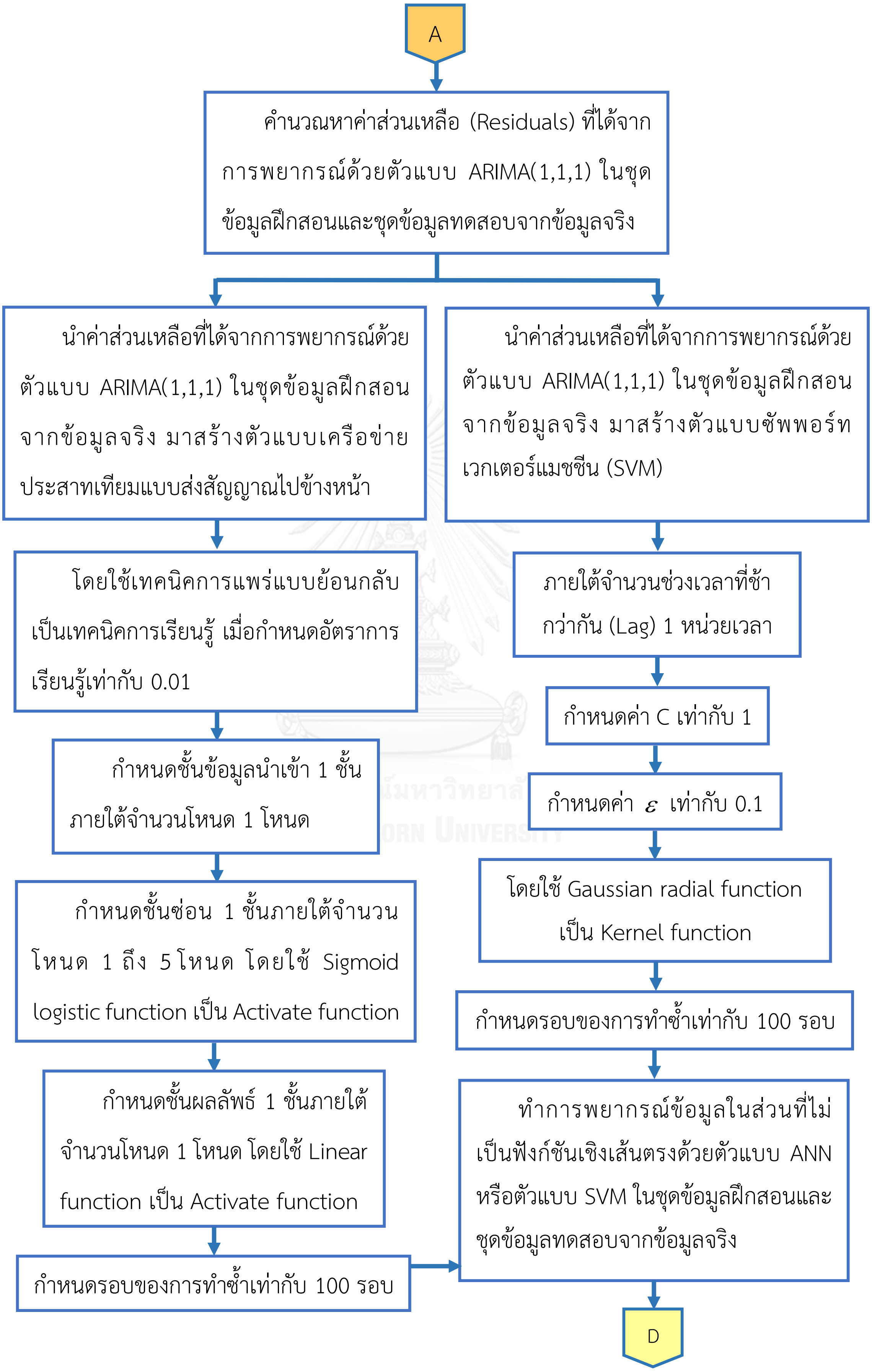
9. นำค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย (Support vector machine model for regression) ภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา และกำหนดค่า C เท่ากับ 1, กำหนดค่า ε เท่ากับ 0.1 โดยใช้ Gaussian radial function เป็น Kernel function เมื่อกำหนดรอบของการทำซ้ำ (Iterations) เท่ากับ 100 รอบ จากนั้นทำการพิจารณาคัดเลือกตัวแบบซัพพอร์ทเวกเตอร์แมชชีนโดยตัวแบบที่เหมาะสมกว่า คือ ตัวแบบที่มีรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ยหรือค่า RMSE ต่ำกว่า และทำการพยากรณ์ข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมชองค่าพารามิเตอร์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริงข้างต้น หลังจากนั้นคำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์ที่ได้จากการพยากรณ์ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน และคำนวณหาค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Mean square error: MSE), รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) และค่าคลาดเคลื่อนสมบูรณ์เฉลี่ย (Mean absolute error: MAE)

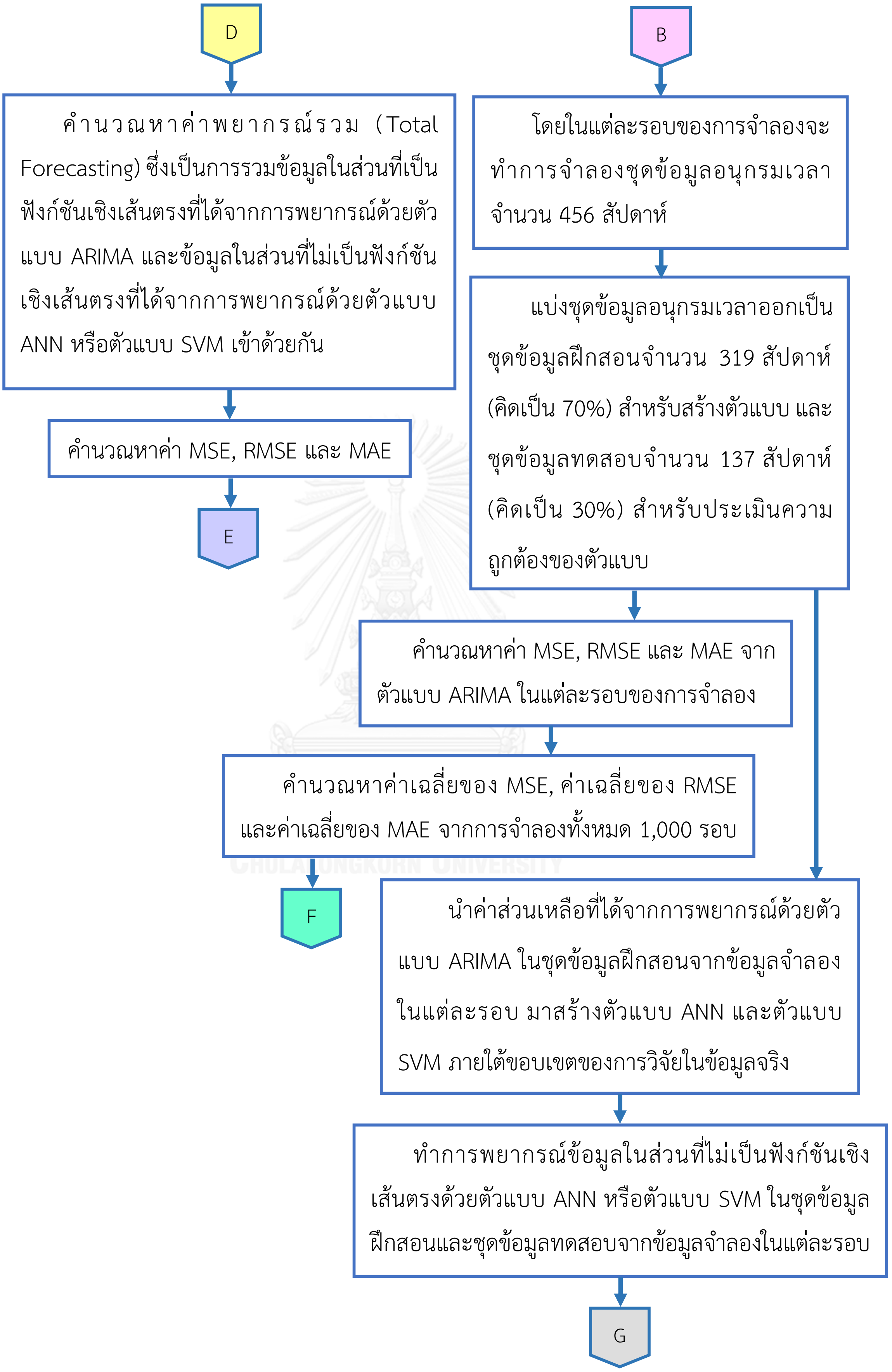
10. ทำการเปรียบเทียบความแม่นยำของตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ โดยตัวแบบใดที่มีค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ให้ผลดีสุด

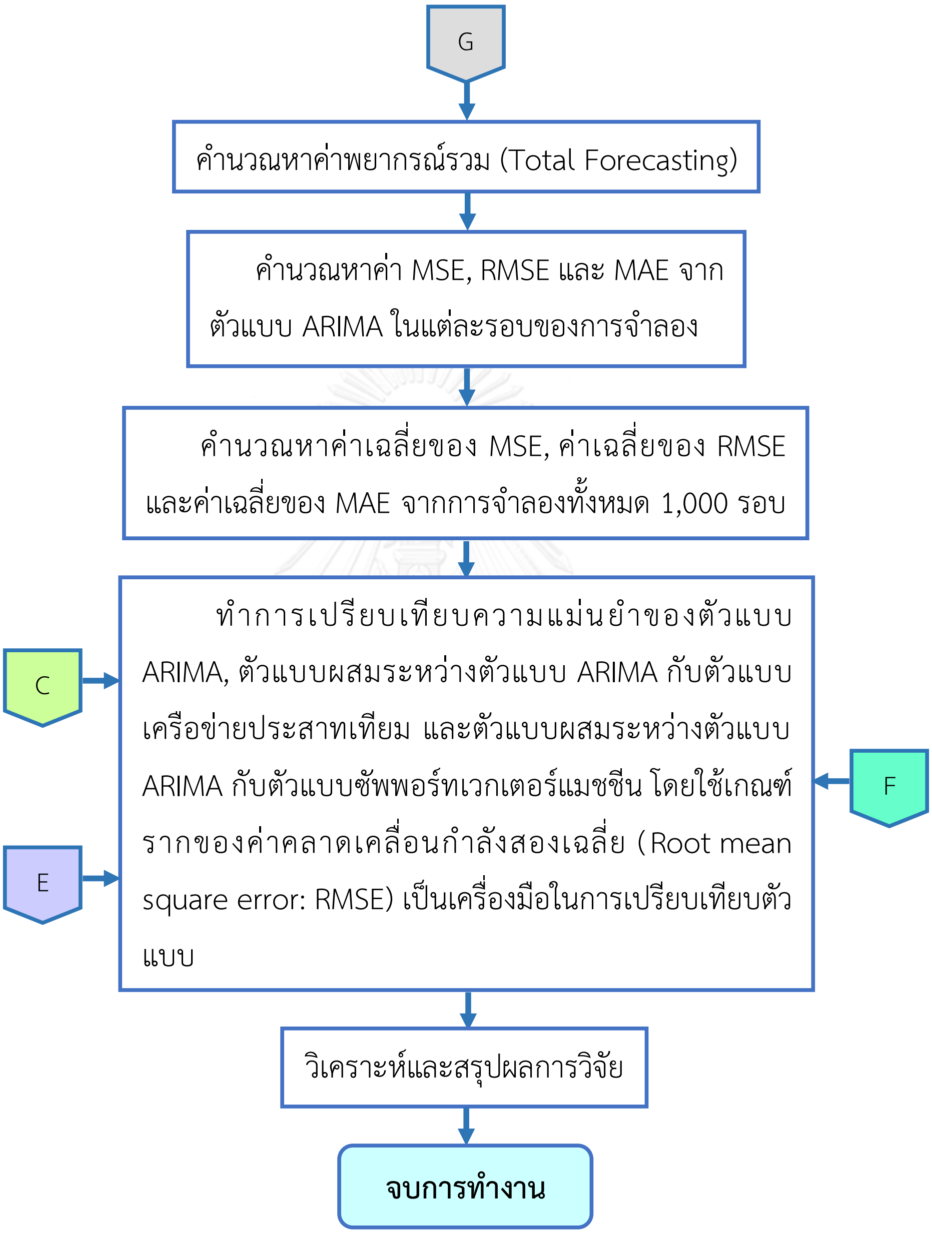
11. วิเคราะห์และสรุปผลการวิจัย

3.3 แผนภาพขั้นตอนการทำงานของโปรแกรม




A
คำนวณหาค่าส่วนเหลือ (Residuals) ที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริง
นำค่าส่วนเหลือที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า
โดยใช้เทคนิคการแพร่แบบย้อนกลับเป็นเทคนิคการเรียนรู้ เมื่อกำหนดอัตราการเรียนรู้เท่ากับ 0.01
กำหนดชั้นข้อมูลนำเข้า 1 ชั้น ภายใต้จำนวนโหนด 1 โหนด
กำหนดชั้นซ่อน 1 ชั้นภายใต้จำนวนโหนด 1 ถึง 5 โหนด โดยใช้ Sigmoid logistic function เป็น Activate function
กำหนดชั้นผลลัพธ์ 1 ชั้นภายใต้จำนวนโหนด 1 โหนด โดยใช้ Linear function เป็น Activate function
กำหนดรอบของการทำซ้ำเท่ากับ 100 รอบ
นำค่าส่วนเหลือที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนจากข้อมูลจริง มาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีน (SVM)
ภายใต้จำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) 1 หน่วยเวลา
กำหนดค่า C เท่ากับ 1
กำหนดค่า ε เท่ากับ 0.1
โดยใช้ Gaussian radial function เป็น Kernel function
กำหนดรอบของการทำซ้ำเท่ากับ 100 รอบ
ทำการพยากรณ์ข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงด้วยตัวแบบ ANN หรือตัวแบบ SVM ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจริง
D


D
คำนวณหาค่าพยากรณ์รวม (Total Forecasting) ซึ่งเป็นการรวมข้อมูลในส่วนที่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA และข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ด้วยตัวแบบ ANN หรือตัวแบบ SVM เข้าด้วยกัน
คำนวณหาค่า MSE, RMSE และ MAE
E
B
โดยในแต่ละรอบของการจำลองจะทำการจำลองชุดข้อมูลอนุกรมเวลาจำนวน 456 สัปดาห์
แบ่งชุดข้อมูลอนุกรมเวลาออกเป็นชุดข้อมูลฝึกสอนจำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับประเมินความถูกต้องของตัวแบบ
คำนวณหาค่า MSE, RMSE และ MAE จากตัวแบบ ARIMA ในแต่ละรอบของการจำลอง
คำนวณหาค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE จากการจำลองทั้งหมด 1,000 รอบ
F
นำค่าส่วนเหลือที่ได้จากการพยากรณ์ด้วยตัวแบบ ARIMA ในชุดข้อมูลฝึกสอนจากข้อมูลจำลองในแต่ละรอบ มาสร้างตัวแบบ ANN และตัวแบบ SVM ภายใต้ขอบเขตของการวิจัยในข้อมูลจริง
ทำการพยากรณ์ข้อมูลในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงด้วยตัวแบบ ANN หรือตัวแบบ SVM ในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบจากข้อมูลจำลองในแต่ละรอบ
G


G
คำนวณหาค่าพยากรณ์รวม (Total Forecasting)
คำนวณหาค่า MSE, RMSE และ MAE จากตัวแบบ ARIMA ในแต่ละรอบของการจำลอง
คำนวณหาค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE จากการจำลองทั้งหมด 1,000 รอบ
C
E
F
ทำการเปรียบเทียบความแม่นยำของตัวแบบ ARIMA, ตัวแบบผสมระหว่างตัวแบบ ARIMA กับตัวแบบเครือข่ายประสาทเทียม และตัวแบบผสมระหว่างตัวแบบ ARIMA กับตัวแบบซัพพอร์ทเวกเตอร์แมชชีน โดยใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ
วิเคราะห์และสรุปผลการวิจัย
จบการทำงาน

บทที่ 4

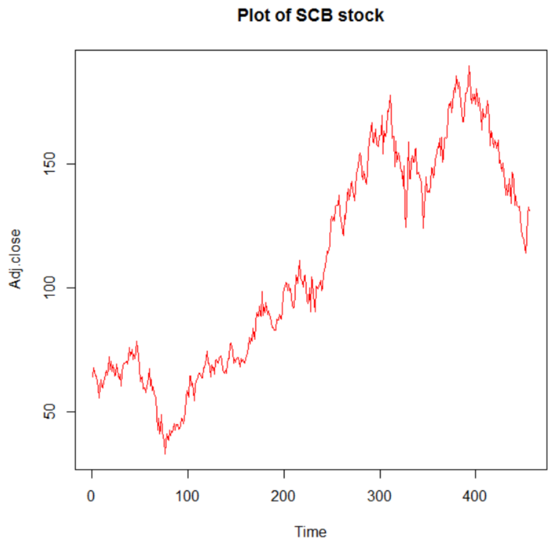
ผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB ของธนาคารไทยพาณิชย์จำกัด (มหาชน) โดยใช้ชุดข้อมูลจริงของราคาปิดหุ้น SCB รายสัปดาห์ ตั้งแต่วันที่ 2 เดือนกรกฎาคม พ.ศ. 2550 ถึงวันที่ 9 เดือนกุมภาพันธ์ พ.ศ. 2559 จำนวน 456 สัปดาห์ ซึ่งแบ่งเป็นชุดข้อมูลฝึกสอน จำนวน 319 สัปดาห์ (คิดเป็น 70%) สำหรับสร้างตัวแบบ และชุดข้อมูลทดสอบจำนวน 137 สัปดาห์ (คิดเป็น 30%) สำหรับตรวจสอบความถูกต้องของตัวแบบ จากนั้นทำการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) ซึ่งในแต่ละตัวแบบจะทำการจำลองชุดข้อมูลอนุกรมเวลาภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ และใช้เกณฑ์ในการพิจารณาความแม่นยำของแต่ละตัวแบบ คือ รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) โดยตัวแบบใดที่ให้ค่า RMSE ต่ำสุด จะเป็นตัวแบบที่ให้ผลดีสุด

อักษรย่อและสัญลักษณ์ต่าง ๆ ที่ปรากฏในการนำเสนอผลการวิจัยในตารางต่าง ๆ แทนความหมาย ดังนี้

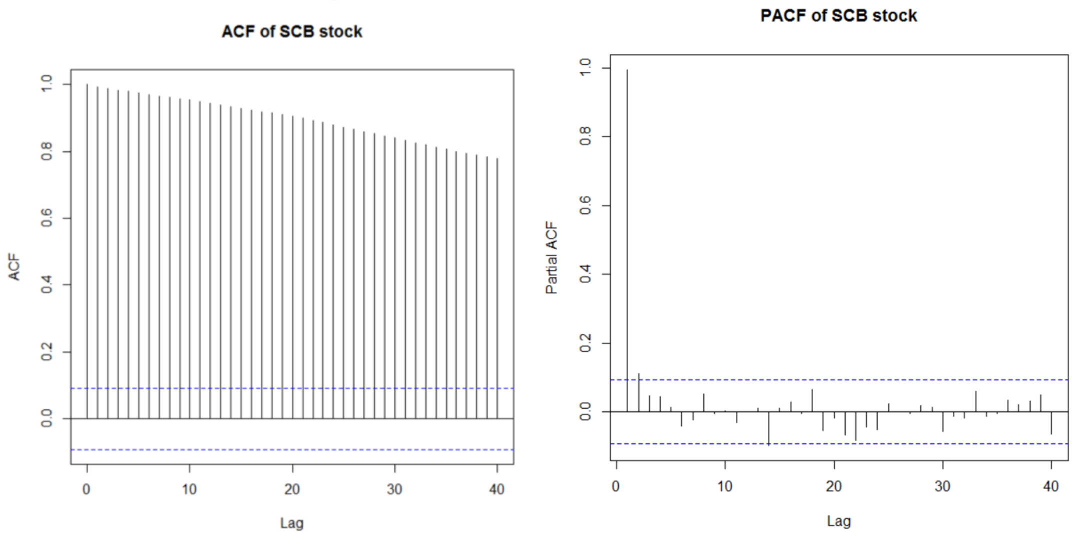
ARIMA	แทน ตัวแบบ Integrated autoregressive moving average หรือ ARIMA(p,d,q)
ARIMA-ANN	แทน ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม
ARIMA-SVM	แทน ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน
Train	แทน ชุดข้อมูลฝึกสอน (Training data set)
Test	แทน ชุดข้อมูลทดสอบ (Testing data set)
ตัวเลขใน (...)	แทน ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation)

พิจารณาลักษณะเบื้องต้นของข้อมูลจริงราคาปิดหุ้น SCB รายสัปดาห์ ของธนาคารไทยพาณิชย์จำกัด (มหาชน) โดยการพล็อตกราฟที่มีแกนตั้ง คือ ราคาปิดหุ้น SCB และแกนนอน คือ เวลา



ภาพที่ 4- 1 ลักษณะเบื้องต้นของข้อมูลจริงราคาปิดหุ้น SCB

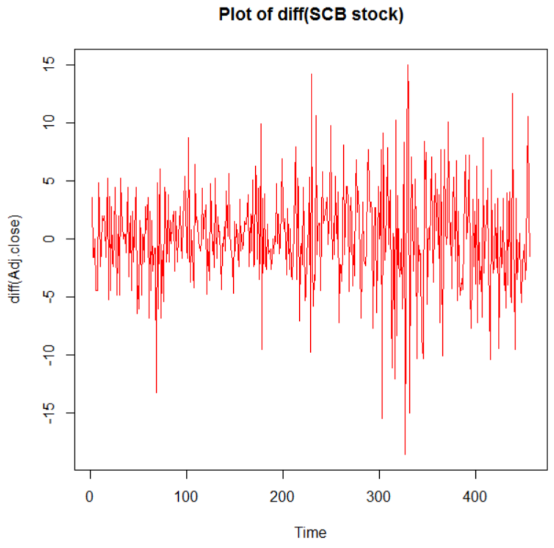
จากภาพที่ 4-1 พบว่า ราคาปิดหุ้น SCB มีแนวโน้ม (Trend) ที่เพิ่มขึ้นเมื่อเวลาผ่านไป นั่นคือ ราคาปิดหุ้น SCB เป็นข้อมูลอนุกรมเวลาแบบไม่คงที่ (Nonstationary time series) โดยมีลักษณะของฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ดังนี้



ภาพที่ 4- 2 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ของข้อมูลจริงราคาปิดหุ้น SCB

โดยการทดสอบ Augmented dickey-fuller (ADF) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ราคาปิดหุ้น SCB เป็นข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) เพราะว่า $p - value = 0.9132 > \alpha = 0.05$

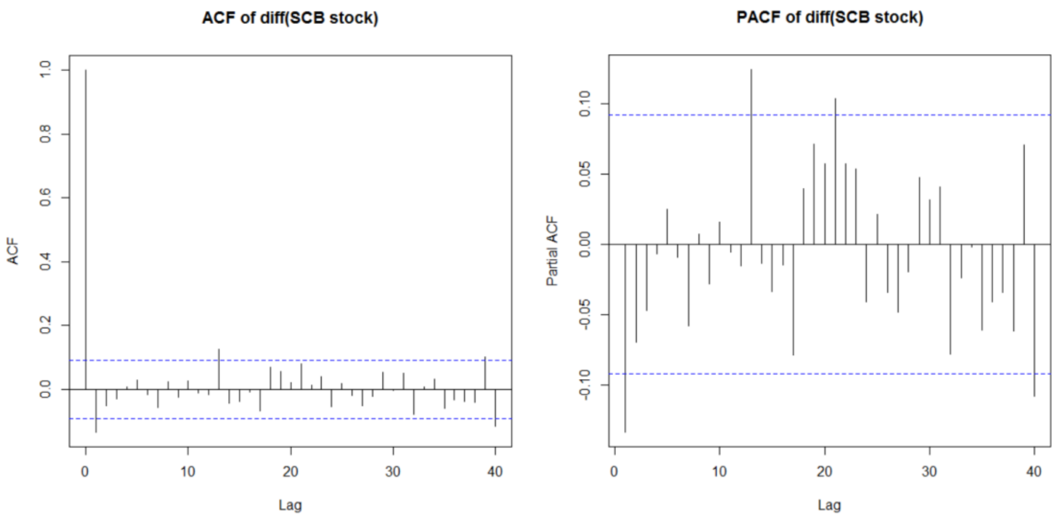
ทำการหาผลต่างอันดับที่ 1 ในชุดข้อมูลจริงราคาปิดหุ้น SCB ข้างต้น และพิจารณาลักษณะเบื้องต้นของราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1 อีกครั้ง โดยการพล็อตกราฟที่มีแกนตั้ง คือ ราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1 และแกนนอน คือ เวลา



ภาพที่ 4- 3 ลักษณะเบื้องต้นของข้อมูลจริงราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1

CHULALONGKORN UNIVERSITY

จากภาพที่ 4-3 พบว่า ราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1 มีลักษณะผันผวนในอัตราที่คงที่รอบจุดกำเนิด นั่นคือ ราคาปิดหุ้น SCB เป็นข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) โดยมีลักษณะของฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ดังนี้



ภาพที่ 4- 4 ฟังก์ชันสหสัมพันธ์ในตัว (ACF) และฟังก์ชันสัมพันธ์สัมพันธ์บางส่วน (PACF) ของข้อมูลจริงราคาปิดหุ้น SCB หลังจากการหารผลต่างอันดับที่ 1

โดยการทดสอบ Augmented dickey-fuller (ADF) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ราคาปิดหุ้น SCB หลังจากการหาผลต่างอันดับที่ 1 เป็นข้อมูลอนุกรมเวลาแบบคงที่ (Stationary time series) เพราะว่า $p-value=0.01<\alpha=0.05$

สำหรับงานวิจัยนี้จะนำเสนอผลการเปรียบเทียบตัวแบบโดยแบ่งออกเป็น 2 ส่วน ดังนี้

ส่วนที่ 1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง

ส่วนที่ 2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง

4.1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง

ตารางที่ 4- 1 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	17.9047 (2.5074)	31.6692 (1.1031)	4.2232 (0.2940)	5.6269 (0.0976)	3.2376 (0.2827)	4.4000 (0.0843)
ARIMA-ANN	15.8461 (0.8452)	30.7938 (0.9868)	**3.9793 (0.1065)**	**5.5485 (0.0885)**	3.0487 (0.0777)	4.3544 (0.0922)
ARIMA-SVM	17.5112 (2.1901)	31.3799 (1.1896)	4.1769 (0.2595)	5.6008 (0.1056)	3.2068 (0.2529)	4.3725 (0.0883)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-1 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(0,1,1) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(0,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(0,1,1) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(0,1,1) มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(0,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(0,1,1) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำใน

การพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(0,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(0,1,1) มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 3.51$ และค่า $p - value = 0.2218$ ซึ่งมากกว่าระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 2 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,2)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	16.8715 (1.1268)	31.2301 (1.4333)	4.1057 (0.1347)	5.5872 (0.1286)	3.1375 (0.0969)	4.3126 (0.0722)
ARIMA-ANN	17.2194 (1.9932)	32.6375 (1.2105)	4.1428 (0.2382)	5.7119 (0.1050)	3.2116 (0.2173)	4.4659 (0.1311)
ARIMA-SVM	16.5274 (1.0851)	30.9956 (1.3259)	**4.0634 (0.1311)**	**5.5661 (0.1194)**	3.1042 (0.0895)	4.2756 (0.0676)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-2 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(0,1,2) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(0,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(0,1,2) และตัวแบบผสมระหว่าง ARIMA(0,1,2) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(0,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(0,1,2) และตัวแบบผสมระหว่าง ARIMA(0,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(0,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(0,1,2) และตัวแบบผสมระหว่าง ARIMA(0,1,2) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(0,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(0,1,2) และตัวแบบผสมระหว่าง ARIMA(0,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 6.46$ และค่า $p - value = 0.1341$ ซึ่งมากกว่าระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 3 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,0)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	42.7047 (10.4087)	92.8749 (24.8893)	6.4909 (0.8459)	9.5688 (1.2808)	5.1936 (0.8683)	8.0164 (1.3621)
ARIMA-ANN	27.9934 (7.0722)	61.4334 (16.9365)	**5.2518** **(0.6441)**	**7.7504** **(1.1731)**	4.1071 (0.5537)	6.2733 (1.0459)
ARIMA-SVM	41.5305 (9.4157)	91.3336 (24.3367)	6.3993 (0.7767)	9.4765 (1.2626)	5.1422 (0.8005)	7.9579 (1.3466)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-3 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(1,1,0) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(1,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,0) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,0) มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(1,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,0) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(1,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,0) มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีบางตัวแบบที่ให้ความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 26.31$ และค่า $p - value = 0.0366$ ซึ่งน้อยกว่าระดับนัยสำคัญ 0.05 จากนั้นจะทำการทดสอบต่อไปว่าตัวแบบคู่ใดบ้างที่มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน ซึ่งทำได้โดยใช้การทดสอบของ Fisher’s Least Significant Difference (LSD) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 4 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,1)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	16.6752 (0.9876)	31.3430 (1.5502)	4.0818 (0.1192)	5.5969 (0.1358)	3.1126 (0.1176)	4.3736 (0.1327)
ARIMA-ANN	17.3899 (1.9187)	33.1324 (2.4670)	4.1643 (0.2210)	5.7522 (0.2116)	3.2165 (0.2049)	4.5396 (0.2027)
ARIMA-SVM	16.2766 (0.9819)	31.0718 (1.5256)	**4.0327** **(0.1199)**	**5.5726** **(0.1339)**	3.0784 (0.1162)	4.3379 (0.1298)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-4 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(1,1,1) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีบางตัวแบบที่ให้ความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 19.32$ และค่า $p - value = 0.0492$ ซึ่งน้อยกว่าระดับนัยสำคัญ 0.05 จากนั้นจะทำการทดสอบต่อไปว่าตัวแบบคู่ใดบ้างที่มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน ซึ่งทำได้โดยใช้การทดสอบของ Fisher’s Least Significant Difference (LSD) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 5 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(1,1,2)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	17.9097 (1.1782)	31.6099 (0.4604)	4.2301 (0.1409)	5.6221 (0.0409)	3.2677 (0.1405)	4.3552 (0.0233)
ARIMA-ANN	15.5032 (0.3274)	30.9144 (0.6801)	**3.9372** **(0.0415)**	**5.5597** **(0.0611)**	3.0301 (0.0436)	4.3281 (0.0439)
ARIMA-SVM	17.4436 (1.0664)	31.4636 (0.4865)	4.1746 (0.1293)	5.6091 (0.0433)	3.2253 (0.1268)	4.3276 (0.0318)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-5 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(1,1,2) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(1,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,2) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,2) มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(1,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,2) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(1,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(1,1,2) มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 2.36$ และค่า $p-value = 0.2975$ ซึ่งมากกว่าระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 6 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,0)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	32.6787 (9.1731)	43.5534 (13.8165)	5.6640 (0.8644)	6.5339 (1.0379)	4.5818 (0.7884)	5.2505 (0.9325)
ARIMA-ANN	23.9346 (3.5034)	37.2724 (4.7569)	**4.8786** **(0.3672)**	**6.0926** **(0.3926)**	3.7981 (0.3313)	4.8474 (0.3780)
ARIMA-SVM	31.5943 (8.3643)	43.1729 (12.5858)	5.5665 (0.7961)	6.5044 (0.9497)	4.4965 (0.7265)	5.2202 (0.8541)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-6 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(2,1,0) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(2,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,0) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,0) มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(2,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,0) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(2,1,0) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,0) มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 13.70$ และค่า $p-value = 0.0680$ ซึ่งมากกว่าระดับนัยสำคัญ 0.05

ตารางที่ 4- 7 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,1)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	17.2696 (1.0874)	32.2049 (1.1359)	4.1540 (0.1307)	5.6742 (0.1008)	3.1718 (0.1749)	4.4578 (0.1177)
ARIMA-ANN	15.7958 (0.6305)	31.4881 (1.0193)	**3.9736 (0.0788)**	**5.6107 (0.0903)**	3.0585 (0.0782)	4.4027 (0.0474)
ARIMA-SVM	16.8228 (0.9434)	31.8561 (1.0826)	4.1000 (0.1152)	5.6433 (0.0966)	3.1291 (0.1597)	4.4184 (0.1072)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-7 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(2,1,1) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(2,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,1) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,1) มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับเครือข่ายประสาทเทียม ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบผสมระหว่าง ARIMA(2,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,1) ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA(2,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA(2,1,1) มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 4.00$ และค่า $p-value = 0.2000$ ซึ่งมากกว่าระดับนัยสำคัญ 0.05

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4- 8 ค่าเฉลี่ยของ MSE, ค่าเฉลี่ยของ RMSE และค่าเฉลี่ยของ MAE ที่ได้จากการจำลองชุดข้อมูลอนุกรมด้วยตัวแบบ ARIMA(2,1,2)

ตัวแบบ	ค่าเฉลี่ยของ MSE		ค่าเฉลี่ยของ RMSE		ค่าเฉลี่ยของ MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	16.5388 (0.7724)	31.2033 (2.1023)	4.0659 (0.0956)	5.5835 (0.1876)	3.0986 (0.0757)	4.3902 (0.1701)
ARIMA-ANN	16.5797 (1.4186)	31.3785 (1.6262)	4.0682 (0.1719)	5.5998 (0.1456)	3.1584 (0.1722)	4.4269 (0.1292)
ARIMA-SVM	16.1344 (0.7743)	30.8265 (2.0531)	**4.0156 (0.0971)**	**5.5492 (0.1845)**	3.0619 (0.0768)	4.3362 (0.1595)

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-8 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(2,1,2) ภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบ โดยใช้ชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(2,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(2,1,2) และตัวแบบผสมระหว่าง ARIMA(2,1,2) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(2,1,2) และตัวแบบผสมระหว่าง ARIMA(2,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ส่วนการพยากรณ์โดยใช้ชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(2,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่าเฉลี่ยของ RMSE ต่ำกว่าตัวแบบ ARIMA(2,1,2) และตัวแบบผสมระหว่าง ARIMA(2,1,2) กับเครือข่ายประสาทเทียม ตามลำดับ กล่าวคือ ตัวแบบผสมระหว่าง ARIMA(2,1,2) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(2,1,2) และตัวแบบผสมระหว่าง ARIMA(2,1,2) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีบางตัวแบบที่ให้ความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 39.81$ และค่า $p-value = 0.0245$ ซึ่งน้อยกว่าระดับนัยสำคัญ 0.05 จากนั้นจะทำการทดสอบต่อไปว่าตัวแบบคู่ใดบ้างที่มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน ซึ่งทำได้โดยใช้การทดสอบของ Fisher’s Least Significant Difference (LSD) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง

ตารางที่ 4- 9 ค่า MSE, RMSE และ MAE ที่ได้จากการพยากรณ์ราคาปิดของหุ้น SCB โดยใช้ชุดข้อมูลจริง

ตัวแบบ	MSE		RMSE		MAE	
	Train	Test	Train	Test	Train	Test
ARIMA	14.4778	29.2408	3.8049	5.4075	2.9121	4.1488
ARIMA-ANN	14.4853	29.4017	3.8059	5.4223	2.9147	4.1728
ARIMA-SVM	13.8246	28.5037	**3.7181**	**5.3389**	2.8466	4.0888

หมายเหตุ ช่องที่ระบายสี หมายถึง ตัวแบบที่เหมาะสมที่สุด เมื่อใช้เกณฑ์ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ

จากตารางที่ 4-9 การศึกษาตัวแบบ ARIMA ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 พบว่า ตัวแบบที่เหมาะสมที่สุดสำหรับใช้พยากรณ์ราคาปิดของหุ้น SCB คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด ซึ่งมีรูปแบบของสมการคือ $\nabla x_t = -0.9181 \nabla x_{t-1} - 0.9949\, w_{t-1} + w_t$ โดยให้ค่า RMSE เท่ากับ 3.8049 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.4075 สำหรับชุดข้อมูลทดสอบ

การศึกษาตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม โดยนำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA(1,1,1) มาสร้างตัวแบบเครือข่ายประสาทเทียม พบว่า ตัวแบบเครือข่ายประสาทเทียม 1-4-1 เป็นตัวแบบที่เหมาะสมที่สุด กล่าวคือ เป็นตัวแบบเครือข่ายประสาทเทียมที่ชั้นข้อมูลนำเข้ามีจำนวนโหนด 1 โหนด, ชั้นซ่อนมีจำนวนโหนด 4 โหนด และชั้นผลลัพธ์มีจำนวนโหนด 1 โหนด เนื่องจากเป็นตัวแบบที่ให้ค่า RMSE ต่ำที่สุด จากนั้นทำการคำนวณหาค่าพยากรณ์รวม (Total forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น

SCB ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบเครือข่ายประสาทเทียม 1-4-1 เข้าด้วยกัน โดยให้ค่า RMSE เท่ากับ 3.8059 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.4223 สำหรับชุดข้อมูลทดสอบ

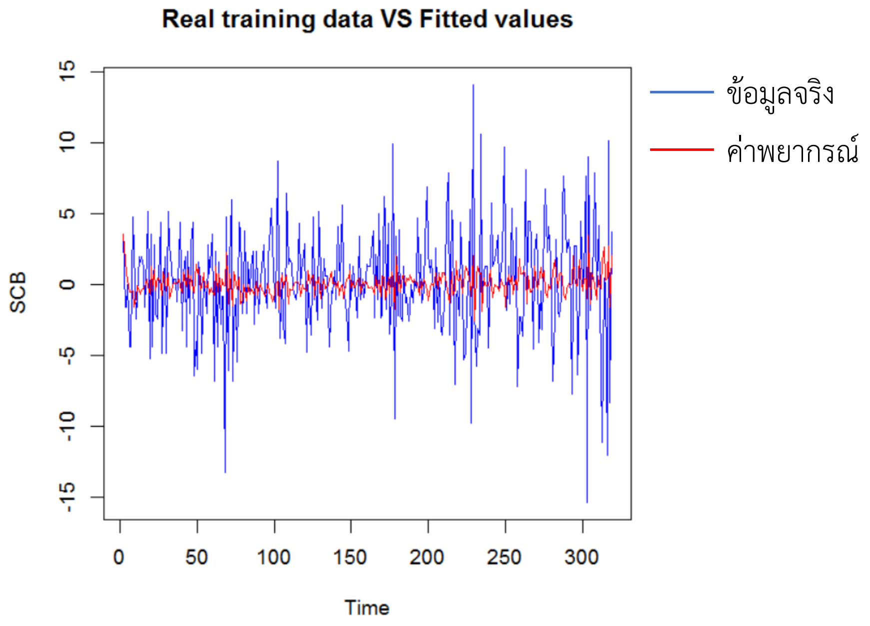
การศึกษาตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยนำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA(1,1,1) มาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย พบว่า ตัวแบบซัพพอร์ทเวกเตอร์แมชชีนที่มีจำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา, ค่า C เท่ากับ 1, ค่า ε เท่ากับ 0.1 และค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.544 เป็นตัวแบบที่เหมาะสมที่สุด เนื่องจากเป็นตัวแบบที่ให้ค่า RMSE ต่ำที่สุด จากนั้นทำการคำนวณหาค่าพยากรณ์รวม (Total forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน โดยให้ค่า RMSE เท่ากับ 3.7181 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.3389 สำหรับชุดข้อมูลทดสอบ

ดังนั้น สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่า RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า การพยากรณ์ราคาปิดหุ้น SCB โดยใช้ชุดข้อมูลจริงในชุดข้อมูลฝึกสอน ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่า RMSE ต่ำกว่าตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม ตามลำดับ ส่วนการพยากรณ์ราคาปิดหุ้น SCB ในชุดข้อมูลทดสอบ ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน ให้ค่า RMSE ต่ำกว่าตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม ตามลำดับ ดังนั้นกล่าวโดยสรุปได้คือ ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ทั้งในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

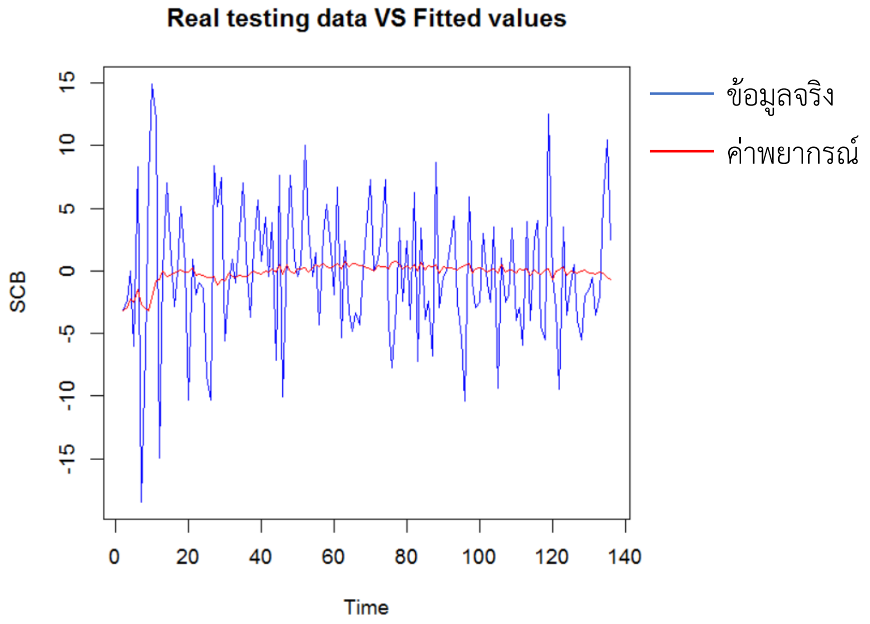
สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของ

การจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์ มีบางตัวแบบที่ให้ความแม่นยำในการพยากรณ์ที่แตกต่างกัน เพราะว่า $F_c = 99.27$ และค่า $p-value = 0.0100$ ซึ่งน้อยกว่าระดับนัยสำคัญ 0.05 จากนั้นจะทำการทดสอบต่อไปว่าตัวแบบคู่ใดบ้างที่มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน ซึ่งทำได้โดยใช้การทดสอบของ Fisher’s Least Significant Difference (LSD) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียมมีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

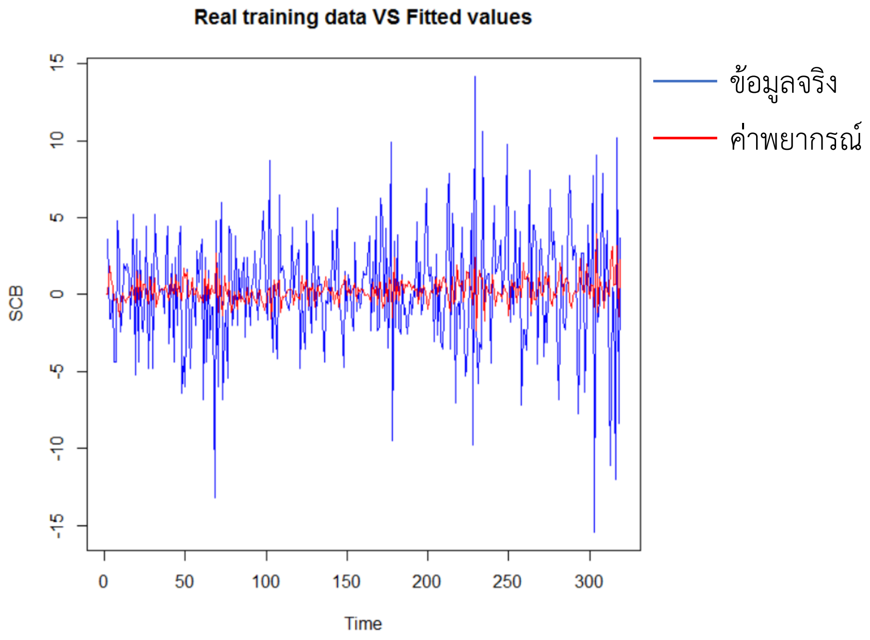
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



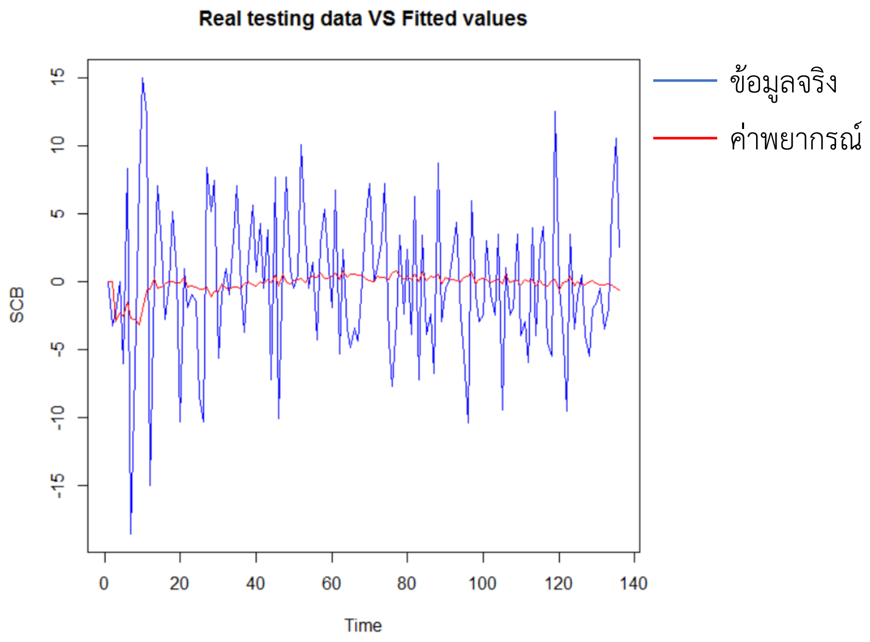
ภาพที่ 4- 5 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบ ARIMA ในชุดข้อมูลฝึกสอน



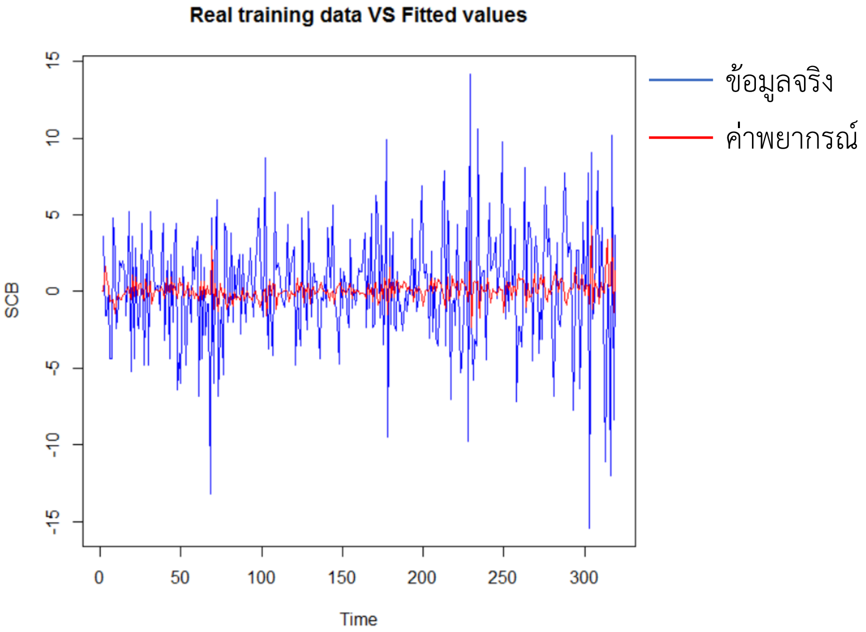
ภาพที่ 4- 6 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบ ARIMA ในชุดข้อมูลทดสอบ



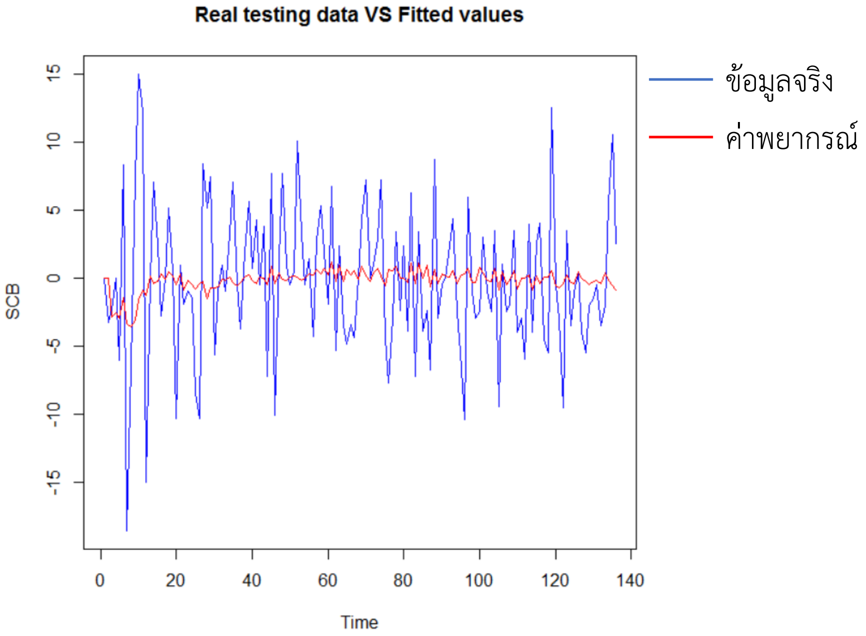
ภาพที่ 4- 7 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม ในชุดข้อมูลฝึกสอน



ภาพที่ 4- 8 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม ในชุดข้อมูลทดสอบ



ภาพที่ 4- 9 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในชุดข้อมูลฝึกสอน



ภาพที่ 4- 10 การเปรียบเทียบระหว่างค่าพยากรณ์กับข้อมูลจริงจากตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในชุดข้อมูลทดสอบ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

การศึกษาเปรียบเทียบความแม่นยำของตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB ของธนาคารไทยพาณิชย์จำกัด (มหาชน) โดยใช้ชุดข้อมูลจริงและชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อใช้เกณฑ์ในการพิจารณาความแม่นยำของแต่ละตัวแบบจากรากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) ซึ่งตัวแบบใดที่ให้ค่า RMSE ต่ำสุดจะเป็นตัวแบบที่ให้ผลดีสุด โดยสรุปผลการวิจัยได้ดังนี้

5.1 สรุปผลการวิจัย

โดยผลการวิจัยจะแบ่งออกเป็น 2 ส่วน ดังนี้

ส่วนที่ 1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง

ส่วนที่ 2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.1.1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง

ตารางที่ 5- 1 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจำลอง

จำลองชุดข้อมูลอนุกรมเวลา	อันดับความแม่นยำของตัวแบบ	ค่าเฉลี่ยของ RMSE		ตัวแบบที่เหมาะสมที่สุด
		Train	Test	
ARIMA(0,1,1)	1. ตัวแบบผสม ARIMA-ANN	3.9793 (0.1065)	5.5485 (0.0885)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	4.1769 (0.2595)	5.6008 (0.1056)	
	3. ตัวแบบ ARIMA	4.2232 (0.2940)	5.6269 (0.0976)	
ARIMA(0,1,2)	1. ตัวแบบผสม ARIMA-SVM	4.0634 (0.1311)	5.5661 (0.1194)	ตัวแบบผสม ARIMA-SVM
	2. ตัวแบบ ARIMA	4.1057 (0.1347)	5.5872 (0.1286)	
	3. ตัวแบบผสม ARIMA-ANN	4.1428 (0.2382)	5.7119 (0.1050)	
ARIMA(1,1,0)	1. ตัวแบบผสม ARIMA-ANN	5.2518 (0.6441)	7.7504 (1.1731)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	6.3993 (0.7767)	9.4765 (1.2626)	
	3. ตัวแบบ ARIMA	6.4909 (0.8459)	9.5688 (1.2808)	

จำลองชุดข้อมูลอนุกรมเวลา	อันดับความแม่นยำของตัวแบบ	ค่าเฉลี่ยของ RMSE		ตัวแบบที่เหมาะสมที่สุด
		Train	Test	
ARIMA(0,1,1)	1. ตัวแบบผสม ARIMA-ANN	3.9793 (0.1065)	5.5485 (0.0885)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	4.1769 (0.2595)	5.6008 (0.1056)	
	3. ตัวแบบ ARIMA	4.2232 (0.2940)	5.6269 (0.0976)	
ARIMA(0,1,2)	1. ตัวแบบผสม ARIMA-SVM	4.0634 (0.1311)	5.5661 (0.1194)	ตัวแบบผสม ARIMA-SVM
	2. ตัวแบบ ARIMA	4.1057 (0.1347)	5.5872 (0.1286)	
	3. ตัวแบบผสม ARIMA-ANN	4.1428 (0.2382)	5.7119 (0.1050)	
ARIMA(1,1,0)	1. ตัวแบบผสม ARIMA-ANN	5.2518 (0.6441)	7.7504 (1.1731)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	6.3993 (0.7767)	9.4765 (1.2626)	
	3. ตัวแบบ ARIMA	6.4909 (0.8459)	9.5688 (1.2808)	
ARIMA(1,1,1)	1. ตัวแบบผสม ARIMA-SVM	4.0327 (0.1199)	5.5726 (0.1339)	ตัวแบบผสม ARIMA-SVM
	2. ตัวแบบ ARIMA	4.0818 (0.1192)	5.5969 (0.1358)	
	3. ตัวแบบผสม ARIMA-ANN	4.1643 (0.2210)	5.7522 (0.2116)	

Chulalongkorn University

จำลองชุดข้อมูลอนุกรมเวลา	อันดับความแม่นยำของตัวแบบ	ค่าเฉลี่ยของ RMSE		ตัวแบบที่เหมาะสมที่สุด
		Train	Test	
ARIMA(1,1,2)	1. ตัวแบบผสม ARIMA-ANN	3.9372 (0.0415)	5.5597 (0.0611)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	4.1746 (0.1293)	5.6091 (0.0433)	
	3. ตัวแบบ ARIMA	4.2301 (0.1409)	5.6221 (0.0409)	
ARIMA(2,1,0)	1. ตัวแบบผสม ARIMA-ANN	4.8786 (0.3672)	6.0926 (0.3926)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	5.5665 (0.7961)	6.5044 (0.9497)	
	3. ตัวแบบ ARIMA	5.6640 (0.8644)	6.5339 (1.0379)	
ARIMA(2,1,1)	1. ตัวแบบผสม ARIMA-ANN	3.9736 (0.0788)	5.6107 (0.0903)	ตัวแบบผสม ARIMA-ANN
	2. ตัวแบบผสม ARIMA-SVM	4.1000 (0.1152)	5.6433 (0.0966)	
	3. ตัวแบบ ARIMA	4.1540 (0.1307)	5.6742 (0.1008)	
ARIMA(2,1,2)	1. ตัวแบบผสม ARIMA-SVM	4.0156 (0.0971)	5.5492 (0.1845)	ตัวแบบผสม ARIMA-SVM
	2. ตัวแบบ ARIMA	4.0659 (0.0956)	5.5835 (0.1876)	
	3. ตัวแบบผสม ARIMA-ANN	4.0682 (0.1719)	5.5998 (1.6262)	

จากตารางที่ 5-1 พบว่า สำหรับการเปรียบเทียบตัวแบบโดยใช้ค่าเฉลี่ยของ RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ ผลปรากฏว่า การพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,1) และตัวแบบ ARIMA(2,1,2) สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ทั้งในชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ ส่วนการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่ได้จากจำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0) และตัวแบบ ARIMA(2,1,1) สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE ผลปรากฏว่า สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0) และตัวแบบ ARIMA(2,1,1) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์ มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(1,1,0) และตัวแบบ ARIMA(1,1,1) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่

ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(2,1,2) สรุปได้ว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบผสม ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

5.1.2 ผลการเปรียบเทียบความแม่นยำของค่าพยากรณ์ที่ได้จากตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ชุดข้อมูลจริง

จากการศึกษาตัวแบบ ARIMA ภายใต้ค่า p ตั้งแต่ 0 ถึง 2, ค่า d เท่ากับ 1 และค่า q ตั้งแต่ 0 ถึง 2 พบว่า ตัวแบบที่เหมาะสมที่สุดสำหรับใช้พยากรณ์ราคาปิดหุ้น SCB คือ ตัวแบบ ARIMA(1,1,1) เนื่องจากเป็นตัวแบบที่มีเกณฑ์สารสนเทศของอะกะอิเกะหรือค่า AIC ต่ำที่สุด ซึ่งมีรูปแบบของสมการคือ $\nabla x_t = -0.9181\nabla x_{t-1} - 0.9949\, w_{t-1} + w_t$ โดยให้ค่า RMSE เท่ากับ 3.8049 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.4075 สำหรับชุดข้อมูลทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากการศึกษาตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม โดยนำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA(1,1,1) มาสร้างตัวแบบเครือข่ายประสาทเทียม พบว่า ตัวแบบเครือข่ายประสาทเทียม 1-4-1 เป็นตัวแบบที่เหมาะสมที่สุด กล่าวคือ เป็นตัวแบบเครือข่ายประสาทเทียมที่ชั้นข้อมูลนำเข้ามีจำนวนโหนด 1 โหนด, ชั้นซ่อนมีจำนวนโหนด 4 โหนด และชั้นผลลัพธ์มีจำนวนโหนด 1 โหนด เนื่องจากเป็นตัวแบบที่ให้ค่า RMSE ต่ำที่สุด จากนั้นทำการคำนวณหาค่าพยากรณ์รวม (Total forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบเครือข่ายประสาทเทียม 1-4-1 เข้าด้วยกัน โดยให้ค่า RMSE เท่ากับ 3.8059 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.4223 สำหรับชุดข้อมูลทดสอบ

จากการศึกษาตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน โดยนำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA(1,1,1) มาสร้างตัวแบบซัพพอร์ทเวกเตอร์แมชชีนสำหรับการถดถอย พบว่าตัวแบบซัพพอร์ทเวกเตอร์แมชชีนที่มีจำนวนช่วงเวลาที่ช้ากว่ากัน (Lag) เท่ากับ 1 หน่วยเวลา, ค่า C เท่ากับ 1, ค่า ε เท่ากับ 0.1 และค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.544 เป็นตัวแบบที่เหมาะสมที่สุด เนื่องจากเป็นตัวแบบที่ให้ค่า RMSE ต่ำที่สุด จากนั้นทำการคำนวณหาค่าพยากรณ์รวม (Total forecasting) ซึ่งเป็นการรวมข้อมูลส่วนที่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบ ARIMA(1,1,1) และข้อมูลส่วนที่ไม่เป็นฟังก์ชันเชิงเส้นตรงที่ได้จากการพยากรณ์ราคาปิดหุ้น SCB ด้วยตัวแบบซัพพอร์ทเวกเตอร์แมชชีนเข้าด้วยกัน โดยให้ค่า RMSE เท่ากับ 3.7181 สำหรับชุดข้อมูลฝึกสอน และ RMSE เท่ากับ 5.3389 สำหรับชุดข้อมูลทดสอบ

ดังนั้น สำหรับการพยากรณ์ราคาปิดหุ้น SCB โดยใช้ชุดข้อมูลจริงซึ่งมีลักษณะข้อมูลอนุกรมเวลาสอดคล้องกับตัวแบบ ARIMA(1,1,1) ในชุดข้อมูลฝึกสอนและข้อมูลชุดทดสอบ โดยใช้ค่า RMSE เป็นเครื่องมือในการเปรียบเทียบตัวแบบ สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA(1,1,1) กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA(1,1,1) และตัวแบบผสมระหว่าง ARIMA(1,1,1) กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด

สำหรับการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่า RMSE ผลปรากฏว่า ที่ระดับนัยสำคัญ 0.05 พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน นอกจากนี้ ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์ที่แตกต่างกันเช่นกัน แต่ไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ที่แตกต่างกัน

5.2 สรุปผลโดยรวม

จากผลการวิจัยในการเปรียบเทียบความแม่นยำของตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการพยากรณ์ราคาปิดหุ้น SCB ของธนาคารไทยพาณิชย์ จำกัด (มหาชน) โดยใช้ชุดข้อมูลจริงและชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,1), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0), ตัวแบบ ARIMA(2,1,1) และตัวแบบ ARIMA(2,1,2) เมื่อใช้เกณฑ์รากของค่าคลาดเคลื่อนกำลังสองเฉลี่ย (Root mean square error: RMSE) เป็นเครื่องมือในการเปรียบเทียบตัวแบบ ผลปรากฏว่า สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,1) และตัวแบบ ARIMA(2,1,2) สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(1,1,0), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0) และตัวแบบ ARIMA(2,1,1) สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และตัวแบบ ARIMA มีความแม่นยำในการพยากรณ์ต่ำที่สุด และการพยากรณ์ในชุดข้อมูลจริงของราคาปิดหุ้น SCB ซึ่งมีลักษณะข้อมูลอนุกรมเวลาสอดคล้องกับตัวแบบ ARIMA(1,1,1) สรุปได้ว่า ตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน มีความแม่นยำในการพยากรณ์สูงที่สุด รองลงมาคือ ตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม มีความแม่นยำในการพยากรณ์ต่ำที่สุด ซึ่งสอดคล้องกับผลจากชุดข้อมูลจำลอง

สำหรับการเปรียบเทียบความแม่นยำของตัวแบบ เมื่อใช้การทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test ผลปรากฏว่า สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1), ตัวแบบ ARIMA(0,1,2), ตัวแบบ ARIMA(1,1,2), ตัวแบบ ARIMA(2,1,0) และตัวแบบ ARIMA(2,1,1) พบว่า เราไม่สามารถสรุปได้ว่าตัวแบบใดมีความแม่นยำในการพยากรณ์สูงที่สุด เนื่องจากที่ระดับนัยสำคัญ 0.05 เราไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบทั้งสาม

ตัวแบบนั้นมีความแม่นยำในการพยากรณ์ที่แตกต่างกัน สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(1,1,0) และตัวแบบ ARIMA(1,1,1) พบว่า เราไม่สามารถสรุปได้ว่าระหว่างตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ตัวแบบใดมีความแม่นยำในการพยากรณ์สูงที่สุด เนื่องจากที่ระดับนัยสำคัญ 0.05 เราไม่พบนัยสำคัญทางสถิติที่ว่าตัวแบบทั้งสองตัวแบบมีความแม่นยำในการพยากรณ์ที่แตกต่างกัน สำหรับการพยากรณ์ในชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(2,1,2) และการพยากรณ์ในชุดข้อมูลจริงของราคาปิดหุ้น SCB พบว่า เราไม่สามารถสรุปได้เช่นกันว่าระหว่างตัวแบบ ARIMA และตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียมตัวแบบใดมีความแม่นยำในการพยากรณ์สูงที่สุด เนื่องจากที่ระดับนัยสำคัญ 0.05 เราไม่พบนัยสำคัญทางสถิติที่ว่า ตัวแบบทั้งสองตัวแบบมีความแม่นยำในการพยากรณ์ที่แตกต่างกัน กล่าวคือ ตัวแบบผสม (Hybrid model) ไม่ได้มีความแม่นยำในการพยากรณ์สูงกว่าตัวแบบ ARIMA เสมอไป ดังนั้น การเลือกใช้ตัวแบบใดสำหรับพยากรณ์ข้อมูลอนุกรมเวลานั้นขึ้นอยู่กับวัตถุประสงค์ของผู้ที่นำไปใช้งาน แต่การเลือกใช้ตัวแบบ ARIMA ก็เป็นทางเลือกที่น่าสนใจทางเลือกหนึ่ง เนื่องจากเป็นตัวแบบเพียงตัวแบบเดียวที่เป็น Stocastic process ในขณะที่ตัวแบบอื่นนั้นเป็น Pure math นอกจากนั้นยังช่วยประหยัดเวลาในการสร้างตัวแบบอีกด้วย เนื่องจากตัวแบบ ARIMA มีขั้นตอนการวิเคราะห์ข้อมูลที่ไม่ซับซ้อนอีกทั้งการเรียนรู้ทางด้านทฤษฎีเพื่อทำความเข้าใจในตัวแบบ ARIMA ก็ไม่จำเป็นต้องอาศัยความรู้ทางคณิตศาสตร์ขั้นสูงเท่ากับการเรียนรู้ทฤษฎีของตัวแบบผสม (Hybrid model) เป็นต้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.3 ข้อเสนอแนะ

จากงานวิจัยนี้ผู้ที่สนใจอาจจะนำไปศึกษาต่อได้อีกในเรื่องดังต่อไปนี้

1. ในงานวิจัยนี้เลือกใช้ตัวแบบในการเปรียบเทียบความแม่นยำของค่าพยากรณ์เพียง 3 ตัวแบบเท่านั้น คือ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ซึ่งในความเป็นจริงแล้วยังมีตัวแบบผสม (Hybrid model) อีกหลายตัวแบบที่น่าสนใจ เช่น ตัวแบบผสมระหว่าง ARIMA กับ Pegals, ตัวแบบผสมระหว่าง ARIMA กับ Adaptive neuron-fuzzy inference system, ตัวแบบผสมระหว่าง ARIMA กับ Exponential smoothing, ตัวแบบผสมระหว่าง ARIMA กับ Moving average of

percentage change เป็นต้น ซึ่งผู้ที่สนใจอาจนำตัวแบบดังกล่าวข้างต้นมาพิจารณาร่วมด้วยเพื่อทำให้การเปรียบเทียบความแม่นยำของตัวแบบมีประสิทธิภาพมากยิ่งขึ้น

2. ในงานวิจัยนี้เลือกใช้ตัวแบบ ARIMA ในการพยากรณ์ข้อมูลอนุกรมเวลา ซึ่งผู้ที่สนใจอาจศึกษาตัวแบบ SARIMA (Seasonal integrated autoregressive moving average) เพิ่มเติม และนำตัวแบบดังกล่าวมาสร้างเป็นตัวแบบผสม (Hybrid model) เช่น ตัวแบบผสมระหว่าง SARIMA กับเครือข่ายประสาทเทียม, ตัวแบบผสมระหว่าง SARIMA กับซัพพอร์ทเวกเตอร์แมชชีน เป็นต้น

3. การสร้างตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม ในงานวิจัยนี้ได้มีการค่าส่วนเหลือ (Residuals) ที่ได้จากตัวแบบ ARIMA มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า (Feed-forward artificial neural network) ซึ่งในการศึกษาครั้งต่อไปผู้ที่สนใจอาจนำค่าส่วนเหลือที่ได้จากตัวแบบ ARIMA มาสร้างตัวแบบเครือข่ายประสาทเทียมแบบมีการย้อนกลับ (Feedback artificial neural network) แทนการใช้ตัวแบบเครือข่ายประสาทเทียมแบบส่งสัญญาณไปข้างหน้า นอกจากนี้อาจเลือกใช้วิธีการแปลงข้อมูลก่อนการสร้างตัวแบบด้วยวิธีอื่น ๆ แทนการใช้วิธี Max-Min normalization เช่น Z-score standardization เป็นต้น เพื่อเปรียบเทียบว่าให้ผลเหมือนหรือแตกต่างกันอย่างไร และอาจปรับเปลี่ยนประเภทของ Activation function ที่อยู่ในซ่อนและชั้นผลลัพธ์ เช่น ในชั้นซ่อนอาจทำการเลือกใช้ Hyperbolic tangent function เป็น Activation function แทนการใช้ Sigmoid logistic function เป็นต้น

4. การสร้างตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน ในการศึกษาครั้งต่อไปอาจเลือกใช้ Kernel function ประเภทอื่น ๆ แทนการใช้ Gaussian radial basis function เช่น Exponential radial basis function เป็นต้น และอาจพิจารณาเลือกใช้ค่า C และค่า ε ณ ระดับต่าง ๆ แทน เพราะอาจทำให้ตัวแบบที่ได้มีความแม่นยำมากยิ่งขึ้น

รายการอ้างอิง

ภาษาไทย

ต้อง ศรีคชา. (2550). การคาดการณ์ราคาหุ้นระยะสั้นโดยวิธีการผสมผสานทำซ้ำตัวกรองความชันการปรับตัวเข้าหา และการปรับตัวการเรียนรู้นิวรอลฟัซซี. (วิทยานิพนธ์ปริญญาดุษฎีบัณฑิต), สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพมหานคร.

ทรงศิริ แต้สมบัติ. (2553). การพยากรณ์เชิงปริมาณ. กรุงเทพมหานคร: สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์.

พงษ์ศิริ ศิริพานิช. (2550). การพยากรณ์อนุกรมเวลาด้วยตัวแบบผสมระหว่าง ARIMA และเครือข่ายประสาทเทียม. (วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต), มหาวิทยาลัยศิลปากร, นครปฐม.

ภัทร วรภู. (2556). การเปรียบเทียบความแม่นยำของการพยากรณ์อนุกรมเวลาระหว่างตัวแบบผสมและตัวแบบเดี่ยว. (วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต), จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร.

ภูมิฐาน รังคกูลนุวัฒน์. (2556). การวิเคราะห์อนุกรมเวลาสำหรับเศรษฐศาสตร์และธุรกิจ. กรุงเทพมหานคร: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

สายชล สินสมบูรณ์ทอง. (2558). การทำเหมืองข้อมูล. กรุงเทพมหานคร: จามจุรีโปรดักท์.

เอกสิทธิ์ พัชรวงศ์ศักดา. (2557). การวิเคราะห์ข้อมูลด้วยเทคนิคดาต้าไมน์นิงเบื้องต้น. กรุงเทพมหานคร: บริษัท เอเชีย ดิจิตอลการพิมพ์ จำกัด.

ภาษาอังกฤษ

Da-yong Zhang, Pu Chen, & Hong-wei Song. (2008). *Stock market forecasting model based on a hybrid ARMA and support vector machines*. Paper presented at the The 15th international conference on management science and engineering, USA.

Robert H. Shumway, & David S. Stoffer. (2010). *Time series analysis and its applications with R examples*. USA: Springer.

Steve R. Gunn. (1998). Support vector machines for classification and regression. Retrieved from www.ecs.soton.ac.uk/~srg/publications/pdf/SVM.pdf

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

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

คำสั่งการวิเคราะห์ข้อมูลด้วยโปรแกรม R

ตัวอย่างการวิเคราะห์ข้อมูลด้วยตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน กรณีใช้ชุดข้อมูลจริงราคาปิดของหุ้น SCB และชุดข้อมูลอนุกรมเวลาที่ได้จากการจำลองด้วยตัวแบบ ARIMA(1,1,1) โดยใช้โปรแกรม R เวอร์ชัน 3.3.1 ซึ่งมีคำสั่งดังต่อไปนี้

```
##################### Library are used to analysis #####################
library(tseries)
library(forecast)
library(Metrics)
library(DMwR)
library(devtools)
library(neuralnet)
library(kernlab)

attach(scb)
nrow.scb<-dim(scb)[1]

#####################################################################
################### Considering real SCB stock data set ##################
#####################################################################

##################### SCB stock before differencing #####################
adj.close<-scb[,7]

###### Plot of SCB stock
adj.close.plot<-plot(adj.close,type='l',xlab="Time",ylab="Adj.close",col="red",main=
"Plot of SCB stock")

###### Compute ACF & PACF of SCB stock for first 40 lags
```

```
acf.0<-acf(adj.close,main="ACF of SCB stock",type="correlation",40)
pacf.0<-pacf(adj.close,main="PACF of SCB stock",40)

###### Test for stationary using augmented dickey and fuller test (ADF)
test.adf.0<-adf.test(adj.close,alternative="stationary")
test.adf.0

#################### SCB stock after 1st. differencing ####################
diff.adj.close<-diff(ts(adj.close))

###### Plot of diff(SCB stock)
diff.adj.close.plot<-plot(diff.adj.close,type='l',xlab="Time",
ylab="diff(Adj.close)",col="red",main=
"Plot of diff(SCB stock)")

###### Compute ACF & PACF of diff(SCB stock) for first 40 lags
acf.1<-acf(diff.adj.close,main="ACF of diff(SCB stock)",type="correlation",40)
pacf.1<-pacf(diff.adj.close,main="PACF of diff(SCB stock)",40)

###### Test for stationary using augmented dickey and fuller test (ADF)
test.adf.1<-adf.test(diff.adj.close,alternative="stationary")
test.adf.1

##################################################################
############## Data set splitting (training 70% and testing 30%) ##############
##################################################################
diff.adj.close.train<-ts(diff.adj.close[1:ceiling(length(diff.adj.close)*0.7)])
diff.adj.close.test<-ts(diff.adj.close[(ceiling(length(diff.adj.close)*0.7)+1)
:length(diff.adj.close)])
plot.diff.adj.close.splitting<-plot(diff.adj.close,type='l',xlab="Time",ylab="diff(Adj.close)
of SCB stock",col="red",main="Training data set VS Testing data set")
```

```
abline(v=319,col="blue",lty=5)

#################################################################
########################## ARIMA models ##########################
#################################################################

###### Using diff(SCB stock) in training data set to buliding the ARIMA model
###### Fitting ARIMA(p,d=1,q) models
###### Considering initial order p from 0 to max(lag p) & order q from 0 to max(lag
q)
p<-2 ###/// input no. of max(lag p)
q<-2 ###/// input no. of max(lag q)
d<-1 ###/// 1.st differenting
all.model.arima<-c()
for (i in 0:p){
	for(j in 0:q){
		fit<-Arima(diff.adj.close.train,order=c(i,d,j))
		order<-arimaorder(fit)
		AIC<-fit$aic
		BIC<-fit$bic
		SBC<-fit$aicc
		all.model.arima<-c(all.model.arima,order,AIC,BIC,SBC)
	}
}
head<-c("p","d","q","AIC","BIC","SBC")
names(all.model.arima)<-rep(head,(p+1)*(q+1))

###### Show all posible ARIMA(p,d,q) models with (AIC, BIC, SBC) criterions
num.mod<-(p+1)*(q+1)
display.model.arima<-c()
for(i in 1:num.mod){
```

```
        ARIMA<-c()
        count.a<-(6*i)-5
        count.b<-6*i
        ARIMA<-all.model.arima[count.a:count.b]
        display.model.arima<-rbind(display.model.arima,ARIMA)
}
display.model.arima

###### Choosing the best ARIMA(p,d,q) models using "AIC"
func.best.arima.model<-c()
for(i in 1:num.mod){
        min.aic<-min(display.model.arima[,4])
        if(min.aic==display.model.arima[i,4]){
                p.best<-display.model.arima[i,1]
                d.best<-display.model.arima[i,2]
                q.best<-display.model.arima[i,3]
                fit.model<-Arima(diff.adj.close.train,order=c(p.best,d.best,q.best))
                func.best.arima.model<-c(func.best.arima.model,summary(fit.model))
        }
}

###### Computing residuals from the best ARIMA models in training data set
final.arima.train<-arima(diff.adj.close.train,order=c(p.best,d.best,q.best))
final.arima.train
fitted.arima.train<-fitted(final.arima.train)
res.arima.train<-diff.adj.close.train-fitted.arima.train

###### Comparing real training data VS fitted values from the best ARIMA models in training data set
plot. diff. adj. close. train<- plot( diff. adj. close. train,col= " blue" ,ylab= " diff( SCB stock)",main="Real training data VS Fitted values")
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
plot.fitted.arima.train<-lines(fitted.arima.train,col="red")

###### Diagnostic checking in training data set
time.train<-1:length(diff.adj.close.train)
plot.res.train<-plot(time.train,res.arima.train,ylab="Residuals",xlab="Time",main="Plot of residuals in training data set from ARIMA model")
abline(h=0)
box.test.train<-Box.test(res.arima.train,type=c("Box-Pierce","Ljung-Box"))
box.test.train
acf.res.arima.train<-acf(res.arima.train,main="ACF of residuals in training data set",type="correlation",40)
pacf.res.arima.train<-pacf(res.arima.train,main="PACF of residuals in training data set",40)
histogram.res.arima.train<-hist(res.arima.train)

###### Computing MSE, RMSE & MAE from the best ARIMA models in training data set
crite.arima.train<-regr.eval(diff.adj.close.train,fitted.arima.train,
stat=c("mae","mse","rmse"))
crite.arima.train

####################################################################

###### Using diff(SCB stock) in testing data set to checking validity from ARIMA model
final.arima.test<-arima(diff.adj.close.test,order=c(p.best,d.best,q.best))
fitted.arima.test<-fitted(final.arima.test)

###### Computing residuals from the best ARIMA models in testing data set
res.arima.test<-diff.adj.close.test-fitted.arima.test

###### Comparing real testing data VS fitted values from the best ARIMA models in testing data set
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
plot. diff. adj. close. test<- plot( diff. adj. close. test,col= " blue" ,ylab= " diff( SCB stock)",main="Real testing data VS Fitted values")
plot.fitted.arima.test<-lines(fitted.arima.test,col="red")

###### Diagnostic checking in testing data set
time.test<-1:length(diff.adj.close.test)
plot.res.test<-plot(time.test,res.arima.test,ylab="Residuals",xlab="Time",main="Plot of residuals in testing data set from ARIMA model")
abline(h=0)
box.test.test<-Box.test(res.arima.test,type=c("Box-Pierce","Ljung-Box"))
box.test.test
acf. res. arima. test<- acf( res. arima. test,main= " ACF of residuals in testing data set",type="correlation",40)
pacf.res.arima.test<-pacf(res.arima.test,main="PACF of residuals in testing data set",40)
histogram.res.arima.test<-hist(res.arima.test)

###### Computing MSE, RMSE & MAE from the best ARIMA models in testing data set
crite.arima.test<-regr.eval(diff.adj.close.test,fitted.arima.test,stat=c("mae","mse","rmse"))
crite.arima.test

######################################################################
###################### Hybrid ARIMA+ANN models #####################
######################################################################

###### Using residuals in training data set from ARIMA model to building the ANN model
###### Max-Min Normalization the residuals from the best ARIMA models in training data set
norm. res. train<- ( res. arima. train- min( res. arima. train) ) / ( max( res. arima. train) - min(res.arima.train))
norm.res.train.set<-data.frame(norm.res.train)
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
###### Considering initial value of input neurons, hidden neurons, output neurons
and no.iterations
no.iterations<-100          ###/// input no.iterations
no.input<-1                 ###/// input no.lags or no.input neurons in "input layers"
no.hidden<-5                ###/// input max(hidden neurons) in "hidden layers"
no.output<-1                ###/// input no.output neurons in "output layers"
all.model.ann.train<-c()
for(i in 1:no.iterations){
    for(j in 1:no.hidden){
        ann.train<-neuralnet(norm.res.train[2:319]~norm.res.train[1:318],
        data=norm.res.train.set,hidden=j,learningrate=0.01,act.fct="logistic",
        linear.output=TRUE)
        preds.train<-ann.train$net.result[[1]]
        mse.train<-mse(norm.res.train[2:319],preds.train)
        rmse.train<-rmse(norm.res.train[2:319],preds.train)
        mae.train<-mae(norm.res.train[2:319],preds.train)
        order.ann.train<-c(no.input,j,no.output)
        all.model.ann.train<-c(all.model.ann.train,order.ann.train,i,mse.train,
        rmse.train,mae.train)
    }
}
head<-c("input neurons","hidden neurons","output neurons","iterations","MSE","RMSE",
"MAE")
names(all.model.ann.train)<-rep(head,no.input*no.hidden*no.output*no.iterations)

###### Show all posible artificial neuron network models with MSE, RMSE & MAE
num.mod.ann<-no.input*no.hidden*no.output*no.iterations
display.model.ann.train<-c()
for(i in 1:num.mod.ann){
    ANN<-c()
    count.a<-(7*i)-6
```

```
		count.b<-7*i
		ANN<-all.model.ann.train[count.a:count.b]
		display.model.ann.train<-rbind(display.model.ann.train,ANN)
}
display.model.ann.train

###### Choosing the best artificial neuron network models using "MSE"
func.best.ann.model.train<-c()
for(i in 1:num.mod.ann){
		min.mse<-min(display.model.ann.train[,5])
		if(min.mse==display.model.ann.train[i,5]&display.model.ann.train[i,1]==1){
				input.best.train<-display.model.ann.train[i,1]
				hidden.best.train<-display.model.ann.train[i,2]
				output.best.train<-display.model.ann.train[i,3]
				interation.best.train<-display.model.ann.train[i,4]
				best.para.ann.train<-c(input.best.train,hidden.best.train,
				output.best.train,interation.best.train)
				names(best.para.ann.train)<-c("best input neurons",
				"best hidden neurons","best output neurons","best iterations")
				print(best.para.ann.train)
				}
func.best.ann.model.train<-c(func.best.ann.model.train)
}

###### Forecasting from the best ANN models in training data set
final.ann.train<-neuralnet(norm.res.train[2:319]~norm.res.train[1:318],
data=norm.res.train.set,hidden=hidden.best.train,learningrate=0.01,act.fct="logistic",
linear.output=TRUE)
norm.res.train.forec<-final.ann.train$net.result[[1]]

###### Plot diagrams in ANN models from training data set
```

```
plot.ann.train<-plot.nn(final.ann.train,col.entry="springgreen4",col.entry.synapse=
"firebrick3",col.hidden="springgreen4",col.hidden.synapse="darkmagenta",
col.out="springgreen4",col.out.synapse="firebrick3",show.weights=T)

###### Transform forecasting value to the original scale in training data set
(Denormalization)
res.train.forec<-(norm.res.train.forec*(max(res.arima.train)-min(res.arima.train)))+
min(res.arima.train)

###### Combined forecasting data form ARIMA and ANN models ---> " Total
forecasting" in training data set
real.train.data<-c(rep(0,1),diff.adj.close.train)
forec.arima.train<-c(rep(0,1),forecast(final.arima.train)$fitted)
total.arima.ann.train<-c()
        forec.ann.train<-c(rep(0,2),res.train.forec)
        total.forec.arima.ann.train<-c(rep(0,2),forec.arima.train[3:320]+
        forec.ann.train[3:320])
total.arima.ann.train<-ts(cbind(total.arima.ann.train,real.train.data,forec.arima.train,
forec.ann.train,total.forec.arima.ann.train))
total.arima.ann.train

###### Comparing real training data VS fitted values from the best Hybrid ARIMA+ANN
models in training data set
plot.arima.ann.train<-plot(diff.adj.close.train,col="blue",ylab=
"diff(SCB stock)",main="Real training data VS Fitted values")
plot.fitted.arima.ann.train<-lines(total.forec.arima.ann.train,col="red")

###### Computing MSE, RMSE and MAE from Hybrid ARIMA+ ANN models in training
data set
crite.arima.ann.train<-regr.eval(real.train.data[3:320],total.forec.arima.ann.train[3:320],
stat=c("mae","mse","rmse"))
```

```
crite.arima.ann.train

##################################################################

###### Using residuals in testing data set to checking validation from Hybrid ARIMA+ANN model
###### Max-Min Normalization the residuals from the best ARIMA models in testing data set
norm.res.test<-(res.arima.test-min(res.arima.test))/(max(res.arima.test)-min(res.arima.test))
norm.res.test.set<-data.frame(norm.res.test)

###### Forecasting from the best ANN models in testing data set
final.ann.test<-neuralnet(norm.res.test[2:136]~norm.res.test[1:135],
data=norm.res.test.set,hidden=hidden.best.train,learningrate=0.01,act.fct="logistic",
linear.output=TRUE)
norm.res.test.forec<-final.ann.test$net.result[[1]]

###### Plot diagrams in ANN models from testing data set
plot.ann.test<-plot.nn(final.ann.test,col.entry="springgreen4",col.entry.synapse=
"firebrick3",col.hidden="springgreen4",col.hidden.synapse="darkmagenta",
col.out="springgreen4",col.out.synapse="firebrick3",show.weights=T)

###### Transform forecasting value to the original scale in testing data set (Denormalization)
res.test.forec<-(norm.res.test.forec*(max(res.arima.test)-min(res.arima.test)))+
min(res.arima.test)

###### Combined forecasting data form ARIMA and ANN models ---> "Total forecasting" in testing data set
real.test.data<-c(rep(0,1),diff.adj.close.test)
```

```
forec.arima.test<-c(rep(0,1),forecast(final.arima.test)$fitted)
total.arima.ann.test<-c()
        forec.ann.test<-c(rep(0,2),res.test.forec)
        total.forec.arima.ann.test<-c(rep(0,2),forec.arima.test[3:137]+
        forec.ann.test[3:137])
total.arima.ann.test<-ts(cbind(total.arima.ann.test,real.test.data,forec.arima.test,
forec.ann.test,total.forec.arima.ann.test))
total.arima.ann.test

###### Comparing real testing data VS fitted values from the best Hybrid ARIMA+ANN
models in testing data set
plot.arima.ann.test<-plot(diff.adj.close.test,col="blue",ylab=
"diff(SCB stock)",main="Real testing data VS Fitted values")
plot.fitted.arima.ann.test<-lines(total.forec.arima.ann.test,col="red")

###### Computing MSE, RMSE and MAE from Hybrid ARIMA+ANN models in testing
data set
crite.arima.ann.test<-regr.eval(real.test.data[3:137],total.forec.arima.ann.test[3:137],
stat=c("mae","mse","rmse"))
crite.arima.ann.test

#####################################################################
###################### Hybrid ARIMA+SVM models #####################
#####################################################################
###### Using residuals in training data set from ARIMA model to building the SVM
model
data.set.svm.train<-data.frame(res.arima.train)

###### Considering initial value of lags, c.value and no.iterations
no.iterations<-100      ###/// input no.iterations
lag.value<-1            ###/// input no.lags
```

```
c.value<-1                  ###/// input c.value
all.model.svm.train<-c()
for(i in 1:no.iterations){
        srange.train<-sigest(res.arima.train[2:319]~res.arima.train[1:318],
        data=data.set.svm.train)
        svm.train<-ksvm(res.arima.train[2:319]~res.arima.train[1:318],data=
        data.set.svm.train,kernel="rbfdot",kpar=list(sigma=srange.train[2]),C=c.value)
        preds.train<-predict(svm.train)
        mse.svm.train<-mse(res.arima.train[2:319],preds.train)
        rmse.svm.train<-rmse(res.arima.train[2:319],preds.train)
        mae.svm.train<-mae(res.arima.train[2:319],preds.train)
        order.svm.train<-c.value
        all.model.svm.train<-c(all.model.svm.train,lag.value,srange.train[2],
        order.svm.train,i,mse.svm.train,rmse.svm.train,mae.svm.train)
}
head<-c("lags","sigma.value","c.value","iteration","MSE","RMSE","MAE")
names(all.model.svm.train)<-rep(head,length(c.value)*no.iterations*length(lag.value))

###### Show all posible support vector machine models with MSE, RMSE and MAE
num.mod.svm<-length(c.value)*no.iterations*length(lag.value)
display.model.svm.train<-c()
for(i in 1:num.mod.svm){
        SVM<-c()
        count.a<-(7*i)-6
        count.b<-7*i
        SVM<-all.model.svm.train[count.a:count.b]
        display.model.svm.train<-rbind(display.model.svm.train,SVM)
}
display.model.svm.train
```

Choosing the best support vector machine models in training data set using "MSE"

```
func.best.svm.model.train<-c()
for(i in 1:num.mod.svm){
        min.mse.svm<-min(display.model.svm.train[,5])
        if(min.mse.svm==display.model.svm.train[i,5]&display.model.svm.train[i,1]==1){
                lag.best.train<-display.model.svm.train[i,1]
                sigma.best.train<-display.model.svm.train[i,2]
                c.best.train<-display.model.svm.train[i,3]
                iteration.best.train<-display.model.svm.train[i,4]
                best.para.svm.train<-c(lag.best.train,sigma.best.train,c.best.train,
                iteration.best.train)
                names(best.para.svm.train)<-c("best lags","best sigma","best c","best
iterations")
                print(best.para.svm.train)
                }
func.best.svm.model.train<-c(func.best.svm.model.train)
}
```

Forecasting from the best SVM models in training data set

```
fit.model.svm.train<-ksvm(res.arima.train[2:319]~res.arima.train[1:318],
data=data.set.svm.train,kernel="rbfdot",kapar=list(sigma=sigma.best.train),
C=c.best.train)
preds.res.train.svm<-predict(fit.model.svm.train)
```

Combined forecasting data form ARIMA and SVM models ---> "Total forecasting" in training data set

```
real.train.data<-c(rep(0,1),diff.adj.close.train)
forec.arima.train<-c(rep(0,1),forecast(final.arima.train)$fitted)
total.arima.svm.train<-c()
        forec.svm.train<-c(rep(0,2),preds.res.train.svm)
```

```
total.forec.arima.svm.train<-c(rep(0,2),forec.arima.train[3:320]+
forec.svm.train[3:320])
total.arima.svm.train<-ts(cbind(total.arima.svm.train,real.train.data,forec.arima.train,
forec.svm.train,total.forec.arima.svm.train))
total.arima.svm.train

###### Comparing real training data VS fitted values from the best Hybrid ARIMA+SVM
models in training data set
plot.arima.svm.train<-plot(diff.adj.close.train,col="blue",
ylab="diff(SCB stock)",main="Real training data VS Fitted values")
plot.fitted.arima.svm.train<-lines(total.forec.arima.svm.train,col="red")

###### Computing MSE, RMSE and MAE from Hybrid ARIMA+ANN models in training
data set
crite.arima.svm.train<-regr.eval(real.train.data[3:320],total.forec.arima.svm.train[3:320],
stat=c("mae","mse","rmse"))
crite.arima.svm.train

##################################################################

###### Using residuals in testing data set to checking validation from Hybrid
ARIMA+SVM model
data.set.svm.test<-data.frame(res.arima.test)

###### Forecasting from the best SVM models in testing data set
final.svm.test<-ksvm(res.arima.test[2:136]~res.arima.train[1:135],
data=data.set.svm.test,kernel="rbfdot",kapar=list(sigma=sigma.best.train),
C=c.best.train)
preds.res.test.svm<-predict(final.svm.test)
```

```
###### Combined forecasting data form ARIMA and SVM models ---> " Total forecasting" in testing data set
real.test.data<-c(rep(0,1),diff.adj.close.test)
forec.arima.test<-c(rep(0,1),forecast(final.arima.test)$fitted)
total.arima.svm.test<-c()
	forec.svm.test<-c(rep(0,2),preds.res.test.svm)
	total.forec.arima.svm.test<-c(rep(0,2),forec.arima.test[3:137]+
	forec.svm.test[3:137])
total.arima.svm.test<-ts(cbind(total.arima.svm.test,real.test.data,forec.arima.test,
forec.svm.test,total.forec.arima.svm.test))
total.arima.svm.test

###### Comparing real testing data VS fitted values from Hybrid ARIMA+SVM models in testing data set
plot.arima.svm.test<-plot(total.arima.svm.test[,1],col="blue",
ylab="diff(SCB stock)",main="Real testing data VS Fitted values")
plot.fitted.arima.svm.test<-lines(total.arima.svm.test[,4],col="red")

###### Computing MSE, RMSE & MAE from Hybrid ARIMA+SVM models in testing data set
crite.arima.svm.test<-regr.eval(real.test.data[3:137],total.forec.arima.svm.test[3:137],
stat=c("mae","mse","rmse"))
crite.arima.svm.test

####################################################################
################# Simulations ARIMA(p=1,d=1,q=1) models ################
####################################################################

no.simmu<-1000                ### input no.iterations of simulations
simmu.func.train<-c()
simmu.func.test<-c()
```

```
simmu.data.train<-c()
simmu.data.test<-c()
for(i in 1:no.simmu){
	sim.data<-arima.sim(list(order=c(1,1,1),ar=final.arima.train$coef[1],
	ma=final.arima.train$coef[2]),n=456)
	sim.training.data<-sim.data[2:ceiling(length(sim.data)*0.7)]
	sim.testing.data<-sim.data[(ceiling(length(sim.data)*0.7)+1):456]
	sim.crite.train<-regr.eval(diff.adj.close.train,sim.training.data,
	stat=c("mae","mse","rmse"))
	sim.crite.test<-regr.eval(diff.adj.close.test,sim.testing.data,
	stat=c("mae","mse","rmse"))
	combined.train.data<-c(i,sim.crite.train)
	combined.test.data<-c(i,sim.crite.test)
	simmu.func.train<-cbind(simmu.func.train,combined.train.data)
	simmu.func.test<-cbind(simmu.func.test,combined.test.data)
	simmu.data.train<-cbind(simmu.data.train,sim.training.data)
	simmu.data.test<-cbind(simmu.data.test,sim.testing.data)
}

###### Show all simulations ARIMA(1,1,1) models with MSE, MAE and RMSE in training
data set
num.mod<-no.simmu
display.model.simmu.train<-c()
for(i in 1:num.mod){
	no.simmulate<-c()
	count.sim.a<-(4*i)-3
	count.sim.b<-4*i
	no.simulation<-simmu.func.train[count.sim.a:count.sim.b]
	display.model.simmu.train<-rbind(display.model.simmu.train,no.simulation)
}
display.model.simmu.train
```

Calculate E[MSE], E[RMSE] and E[MAE] from all simulations ARIMA(1,1,1) models in training data set

```
crite.arima.sim.train.func<-c()
	mean.mae.arima.sim.train<-mean(display.model.simmu.train[,2])
	mean.mse.arima.sim.train<-mean(display.model.simmu.train[,3])
	mean.rmse.arima.sim.train<-mean(display.model.simmu.train[,4])
	crite.arima.sim.train.func<-c(mean.mae.arima.sim.train,
	mean.mse.arima.sim.train,mean.rmse.arima.sim.train)
	names(crite.arima.sim.train.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.sim.train.func
```

Calculate Standard deviation from all simulations ARIMA(1,1,1) models in training data set

```
sd.mae.arima.sim.train<-sd(display.model.simmu.train[,2])
sd.mae.arima.sim.train
sd.mse.arima.sim.train<-sd(display.model.simmu.train[,3])
sd.mse.arima.sim.train
sd.rmse.arima.sim.train<-sd(display.model.simmu.train[,4])
sd.rmse.arima.sim.train
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Show all simulations ARIMA(1,1,1) models with MSE, MAE and RMSE in testing data set

```
display.model.simmu.test<-c()
for(i in 1:num.mod){
	no.simmulate<-c()
	count.sim.a<-(4*i)-3
	count.sim.b<-4*i
	no.simulation<-simmu.func.test[count.sim.a:count.sim.b]
	display.model.simmu.test<-rbind(display.model.simmu.test,no.simulation)
}
display.model.simmu.test
```

Calculate E[MSE], E[RMSE] and E[MAE] from all simulations ARIMA(1,1,1) models in testing data set

```
crite.arima.sim.test.func<-c()
	mean.mae.arima.sim.test<-mean(display.model.simmu.test[,2])
	mean.mse.arima.sim.test<-mean(display.model.simmu.test[,3])
	mean.rmse.arima.sim.test<-mean(display.model.simmu.test[,4])
	crite.arima.sim.test.func<-c(mean.mae.arima.sim.test,
	mean.mse.arima.sim.test,mean.rmse.arima.sim.test)
	names(crite.arima.sim.test.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.sim.test.func
```

Calculate Standard deviation from all simulations ARIMA(1,1,1) models in testing data set

```
sd.mae.arima.sim.test<-sd(display.model.simmu.test[,2])
sd.mae.arima.sim.test
sd.mse.arima.sim.test<-sd(display.model.simmu.test[,3])
sd.mse.arima.sim.test
sd.rmse.arima.sim.test<-sd(display.model.simmu.test[,4])
sd.rmse.arima.sim.test
```

Compute residuals in training and testing data set from each simulations ARIMA(1,1,1) models

```
res.train.arima.sim<-c()
res.test.arima.sim<-c()
for(i in 1:num.mod){
	train.res.sim<-diff.adj.close.train-simmu.data.train[,i]
	train.all.data.res.sim<-c(train.res.sim)
	test.res.sim<-diff.adj.close.test-simmu.data.test[,i]
	test.all.data.res.sim<-c(test.res.sim)
	res.train.arima.sim<-cbind(res.train.arima.sim,train.all.data.res.sim)
```

```
        res.test.arima.sim<-cbind(res.test.arima.sim,test.all.data.res.sim)
}

###################################################################
################# Simulations Hybrid ARIMA+ANN models #################
###################################################################

###### Considering artificial neuron network models from each simulations
ARIMA(1,1,1) models
###### By using residuals from all simulations ARIMA(1,1,1) models in training data
set
all.model.ann.train.sim<-c()
force.norm.train.sim.data<-c()
for(l in 1:no.simmu){
        norm.train.data.sim<-(res.train.arima.sim[,l]-min(res.train.arima.sim[,l]))/
        (max(res.train.arima.sim[,l])-min(res.train.arima.sim[,l]))
        sim.norm.train.data.set<-data.frame(norm.train.data.sim)
        for(i in 1:no.iterations){
                for(j in 1:no.hidden){
                        sim.ann1<-neuralnet(norm.train.data.sim[2:319]~
                        norm.train.data.sim[1:318],data=sim.norm.train.data.set,
                        hidden=j,learningrate=0.01,act.fct="logistic",linear.output=TRUE)
                        sim.preds1<-sim.ann1$net.result[[1]]
                        sim.mse<-mse(norm.train.data.sim[2:319],sim.preds1)
                        sim.rmse<-rmse(norm.train.data.sim[2:319],sim.preds1)
                        sim.mae<-mae(norm.train.data.sim[2:319],sim.preds1)
                        sim.order.ann<-c(no.input,j,no.output)
                        combined.ann.train<-c(sim.order.ann,l,i,sim.mse,sim.rmse,
                        sim.mae)
                        all.model.ann.train.sim<-cbind(all.model.ann.train.sim,
                        combined.ann.train)
```

```
                force.norm.train.sim.data<-cbind(force.norm.train.sim.data,
                sim.preds1)
            }
        }
}
head<-c("input neurons","hidden neurons","output neurons","no.simulation",
"no.iterations","MSE","RMSE","MAE")
names(all.model.ann.train.sim)<-rep(head,no.input*no.iterations*no.hidden*
no.simmu)

###### Show all simulations artificial neuron network models in training data set with
MSE, RMSE and MAE
sim.num.mod.ann<-no.input*no.iterations*no.hidden*no.simmu
sim.display.model.ann.train<-c()
for(i in 1:sim.num.mod.ann){
        ANN<-c()
        count.ann.sim.a<-(8*i)-7
        count.ann.sim.b<-8*i
        ANN<-all.model.ann.train.sim[count.ann.sim.a:count.ann.sim.b]
        sim.display.model.ann.train<-rbind(sim.display.model.ann.train,ANN)
}
sim.display.model.ann.train

###### Transform forecasting value to the original scale in training data set
(Denormalization)
denorm.sim.num.mod.ann<-sim.num.mod.ann/no.simmu
force.denorm.train.sim.data<-c()
for(i in 1:no.simmu){
        for(j in 1:denorm.sim.num.mod.ann){
                train.force.ann.sim<-(force.norm.train.sim.data[,j]*
                (max(res.train.arima.sim[,i])-min(res.train.arima.sim[,i])))+
```

```
                    min(res.train.arima.sim[,i])
                    force.denorm.train.sim.data<-cbind(force.denorm.train.sim.data,
                    train.force.ann.sim)
          }
}

###### Combined forecasting data form each simulations ARIMA and ANN models ---
> "Total forecasting" in training data set
real.train.data<-c(rep(0,1),diff.adj.close.train)
forec.arima.train.sim<-c()
for(i in 1:no.simmu){
          forec.arima.train.sim.data<-c(rep(0,1),simmu.data.train[,i])
          rep<-matrix(rep(forec.arima.train.sim.data,no.iterations*no.hidden),
          ncol=no.iterations*no.hidden)
          forec.arima.train.sim<-cbind(forec.arima.train.sim,rep)
}
forec.arima.train.sim
forec.ann.train.sim<-c()
for(i in 1:sim.num.mod.ann){
          forec.ann.train.sim.data<-c(rep(0,2),force.denorm.train.sim.data[,i])
          forec.ann.train.sim<-cbind(forec.ann.train.sim,forec.ann.train.sim.data)
}
forec.ann.train.sim
total.forec.arima.ann.train.sim<-c()
for(i in 1:sim.num.mod.ann){
          total.forec.arima.ann.train.sim.data<-c(rep(0,2),forec.arima.train.sim[3:320,i]+
          forec.ann.train.sim[3:320,i])
          total.forec.arima.ann.train.sim<-cbind(total.forec.arima.ann.train.sim,total.forec.
          arima.ann.train.sim.data)
}
total.forec.arima.ann.train.sim
```

Computing MSE, RMSE and MAE from each simulations Hybrid ARIMA+ANN models in training data set

```
compute.crite.arima.ann.train.sim.func<-c()
for(i in 1:sim.num.mod.ann){
	crite.arima.ann.train.sim<-regr.eval(real.train.data[3:320],
	total.forec.arima.ann.train.sim[3:320,i],stat=c("mae","mse","rmse"))
	compute.crite.arima.ann.train.sim.func<-
	c(compute.crite.arima.ann.train.sim.func,crite.arima.ann.train.sim)
}
```

Show all simulations Hybrid ARIMA+ANN models in training data set with MSE, RMSE and MAE

```
sim.display.model.arima.ann<-c()
for(i in 1:sim.num.mod.ann){
	ARIMA.ANN<-c()
	count.arima.ann.sim.a<-(3*i)-2
	count.arima.ann.sim.b<-3*i
	ARIMA.ANN<-compute.crite.arima.ann.train.sim.func[count.arima.ann.sim.a:
	count.arima.ann.sim.b]
	sim.display.model.arima.ann<-rbind(sim.display.model.arima.ann,ARIMA.ANN)
}
sim.display.model.arima.ann
```

Calculate E[MSE], E[RMSE] and E[MAE] from all simulations Hybrid ARIMA+ANN models in training data set

```
crite.arima.ann.sim.train.func<-c()
	mean.mae.arima.ann.sim.train<-mean(sim.display.model.arima.ann[,1])
	mean.mse.arima.ann.sim.train<-mean(sim.display.model.arima.ann[,2])
	mean.rmse.arima.ann.sim.train<-mean(sim.display.model.arima.ann[,3])
	crite.arima.ann.sim.train.func<-c(mean.mae.arima.ann.sim.train,
```

```
        mean.mse.arima.ann.sim.train,mean.rmse.arima.ann.sim.train)
        names(crite.arima.ann.sim.train.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.ann.sim.train.func

###### Calculate Standard deviation from all simulations Hybrid ARIMA+ANN models
in training data set
sd.mae.arima.ann.sim.train<-sd(sim.display.model.arima.ann[,1])
sd.mae.arima.ann.sim.train
sd.mse.arima.ann.sim.train<-sd(sim.display.model.arima.ann[,2])
sd.mse.arima.ann.sim.train
sd.rmse.arima.ann.sim.train<-sd(sim.display.model.arima.ann[,3])
sd.rmse.arima.ann.sim.train

###### Using residuals from all simulation ARIMA(1,1,1) models in testing data set for
forecasting in ANN models
all.model.ann.test.sim<-c()
force.norm.test.sim.data<-c()
for(l in 1:no.simmu){
        norm.test.data.sim<-(res.test.arima.sim[,l]-min(res.test.arima.sim[,l]))/
        (max(res.test.arima.sim[,l])-min(res.test.arima.sim[,l]))
        sim.norm.test.data.set<-data.frame(norm.test.data.sim)
        for(i in 1:no.iterations){
                for(j in 1:no.hidden){
                        sim.ann2<-neuralnet(norm.test.data.sim[2:136]~
                        norm.test.data.sim[1:135],data=sim.norm.test.data.set,
                        hidden=j,learningrate=0.01,act.fct="logistic",linear.output=TRUE)
                        sim.preds2<-sim.ann2$net.result[[1]]
                        sim.mse<-mse(norm.test.data.sim[2:136],sim.preds2)
                        sim.rmse<-rmse(norm.test.data.sim[2:136],sim.preds2)
                        sim.mae<-mae(norm.test.data.sim[2:136],sim.preds2)
                        sim.order.ann<-c(no.input,j,no.output)
```

```
                    combined.ann.test<-c(sim.order.ann,l,i,sim.mse,sim.
                    rmse,sim.mae)
                    all.model.ann.test.sim<-cbind(all.model.ann.test.sim,
                    combined.ann.test)
                    force.norm.test.sim.data<-cbind(force.norm.test.sim.data,
                    sim.preds2)
            }
      }
}
head<-c("input neurons","hidden neurons","output neurons","no.simulation",
"no.iterations","MSE","RMSE","MAE")
names(all.model.ann.test.sim)<-rep(head,no.input*no.iterations*no.hidden*no.simmu)

###### Show all simulation artificial neuron network models in testing data set with
MSE, RMSE and MAE
sim.display.model.ann.test<-c()
for(i in 1:sim.num.mod.ann){
      ANN<-c()
      count.ann.sim.a<-(8*i)-7
      count.ann.sim.b<-8*i
      ANN<-all.model.ann.test.sim[count.ann.sim.a:count.ann.sim.b]
      sim.display.model.ann.test<-rbind(sim.display.model.ann.test,ANN)
}
sim.display.model.ann.test

###### Transform forecasting value to the original scale in testing data set
(Denormalization)
force.denorm.test.sim.data<-c()
for(i in 1:no.simmu){
      for(j in 1:denorm.sim.num.mod.ann){
            test.force.ann.sim<-(force.norm.test.sim.data[,j]*
```

```
                    (max(res.test.arima.sim[,i])-min(res.test.arima.sim[,i])))+
                    min(res.test.arima.sim[,i])
                    force.denorm.test.sim.data<-cbind(force.denorm.test.sim.data,
                    test.force.ann.sim)
        }
}

###### Combined forecasting data form each simulations ARIMA and ANN models ---
> "Total forecasting" in testing data
real.test.data<-c(rep(0,1),diff.adj.close.test)
forec.arima.test.sim<-c()
for(i in 1:no.simmu){
        forec.arima.test.sim.data<-c(rep(0,1),simmu.data.test[,i])
        rep.test<-matrix(rep(forec.arima.test.sim.data,no.iterations*no.hidden),
        ncol=no.iterations*no.hidden)
        forec.arima.test.sim<-cbind(forec.arima.test.sim,rep.test)
}
forec.arima.test.sim
forec.ann.test.sim<-c()
for(i in 1:sim.num.mod.ann){
        forec.ann.test.sim.data<-c(rep(0,2),force.denorm.test.sim.data[,i])
        forec.ann.test.sim<-cbind(forec.ann.test.sim,forec.ann.test.sim.data)
}
forec.ann.test.sim
total.forec.arima.ann.test.sim<-c()
for(i in 1:sim.num.mod.ann){
        total.forec.arima.ann.test.sim.data<-c(rep(0,2),forec.arima.test.sim[3:137,i]+
        forec.ann.test.sim[3:137,i])
        total.forec.arima.ann.test.sim<-cbind(total.forec.arima.ann.test.sim,
        total.forec.arima.ann.test.sim.data)
}
```

```
total.forec.arima.ann.test.sim

###### Computing MSE, RMSE and MAE from each simulations Hybrid ARIMA+ANN
models in testing data set
compute.crite.arima.ann.test.sim.func<-c()
for(i in 1:sim.num.mod.ann){
	crite.arima.ann.test.sim<-regr.eval(real.test.data[3:137],
	total.forec.arima.ann.test.sim[3:137,i],stat=c("mae","mse","rmse"))
	compute.crite.arima.ann.test.sim.func<-
	c(compute.crite.arima.ann.test.sim.func,crite.arima.ann.test.sim)
}

###### Show all simulations Hybrid ARIMA+ANN models in testing data set with MSE,
RMSE and MAE
sim.display.model.arima.ann.test<-c()
for(i in 1:sim.num.mod.ann){
	ARIMA.ANN<-c()
	count.arima.ann.sim.a<-(3*i)-2
	count.arima.ann.sim.b<-3*i
	ARIMA.ANN<-compute.crite.arima.ann.test.sim.func[count.arima.ann.sim.a:
	count.arima.ann.sim.b]
	sim.display.model.arima.ann.test<-rbind(sim.display.model.arima.ann.test,
	ARIMA.ANN)
}
sim.display.model.arima.ann.test

###### Calculate E[MSE], E[RMSE] and E[MAE] from all simulations Hybrid ARIMA+ANN
models in testing data set
crite.arima.ann.sim.test.func<-c()
	mean.mae.arima.ann.sim.test<-mean(sim.display.model.arima.ann.test[,1])
	mean.mse.arima.ann.sim.test<-mean(sim.display.model.arima.ann.test[,2])
```

```
        mean.rmse.arima.ann.sim.test<-mean(sim.display.model.arima.ann.test[,3])
        crite.arima.ann.sim.test.func<-c(mean.mae.arima.ann.sim.test,
        mean.mse.arima.ann.sim.test,mean.rmse.arima.ann.sim.test)
        names(crite.arima.ann.sim.test.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.ann.sim.test.func

###### Calculate Standard deviation from all simulations Hybrid ARIMA+ANN models in testing data set
sd.mae.arima.ann.sim.test<-sd(sim.display.model.arima.ann.test[,1])
sd.mae.arima.ann.sim.test
sd.mse.arima.ann.sim.test<-sd(sim.display.model.arima.ann.test[,2])
sd.mse.arima.ann.sim.test
sd.rmse.arima.ann.sim.test<-sd(sim.display.model.arima.ann.test[,3])
sd.rmse.arima.ann.sim.test

################################################################
################# Simulations Hybrid ARIMA+SVM models #################
################################################################

###### Considering support vector machine models from each simulations ARIMA(1,1,1) models
###### By using residuals from all simulation ARIMA(1,1,1) models in training data set
all.model.svm.train.sim<-c()
force.svm.train.sim.data<-c()
for(l in 1:no.simmu){
        res.train.data.sim<-res.train.arima.sim[,l]
        sim.train.data.set<-data.frame(res.train.data.sim)
        sim.lag.value<-lag.best.train
        for(i in 1:no.iterations){
                sim.srange<-sigest(res.train.data.sim[2:319]~res.train.data.sim[1:318],
                data=sim.train.data.set)
```

```
		sim.svm1<-ksvm(res.train.data.sim[2:319]~res.train.data.sim[1:318],
		data=sim.train.data.set,kernel="rbfdot",kpar=list(sigma=sim.srange[2]),
		C=c.best.train)
		sim.preds1<-predict(sim.svm1)
		sim.mse.svm<-mse(res.train.data.sim[2:319],sim.preds1)
		sim.rmse.svm<-rmse(res.train.data.sim[2:319],sim.preds1)
		sim.mae.svm<-mae(res.train.data.sim[2:319],sim.preds1)
		sim.order.svm<-c(sim.lag.value,c.best.train,sim.srange[2],l)
		all.model.svm.train.sim<-c(all.model.svm.train.sim,sim.order.svm,i,
		sim.mse.svm,sim.rmse.svm,sim.mae.svm)
		force.svm.train.sim.data<-cbind(force.svm.train.sim.data,sim.preds1)
	}
}
head<-c("lag.value","c.value","sigma.value","no.simulation",
"no.iteration","MSE","RMSE","MAE")
names(all.model.svm.train.sim)<-rep(head,no.iterations*no.simmu)

###### Show all simulations support vector machine models in training data set with
MSE, RMSE and MAE
sim.num.mod.svm<-no.iterations*no.simmu
sim.display.model.svm.train<-c()
for(i in 1:sim.num.mod.svm){
	SVM<-c()
	count.svm.sim.a<-(8*i)-7
	count.svm.sim.b<-8*i
	SVM<-all.model.svm.train.sim[count.svm.sim.a:count.svm.sim.b]
	sim.display.model.svm.train<-rbind(sim.display.model.svm.train,SVM)
}
sim.display.model.svm.train
###### Combined forecasting data from each simulations ARIMA and SVM models ---
> "Total forecasting" in training data set
```

```
real.train.data<-c(rep(0,1),diff.adj.close)
forec.arima.train.sim<-c()
for(i in 1:no.simmu){
	forec.arima.train.sim.data<-c(rep(0,1),simmu.data.train[,i])
	rep<-matrix(rep(forec.arima.train.sim.data,no.iterations),ncol=no.iterations)
	forec.arima.train.sim<-cbind(forec.arima.train.sim,rep)
}
forec.arima.train.sim
forec.svm.train.sim<-c()
for(i in 1:sim.num.mod.svm){
	forec.svm.train.sim.data<-c(rep(0,2),force.svm.train.sim.data[,i])
	forec.svm.train.sim<-cbind(forec.svm.train.sim,forec.svm.train.sim.data)
}
forec.svm.train.sim
total.forec.arima.svm.train.sim<-c()
for(i in 1:sim.num.mod.svm){
	total.forec.arima.svm.train.sim.data<-c(rep(0,2),forec.arima.train.sim[3:320,i]+
	forec.svm.train.sim[3:320,i])
	total.forec.arima.svm.train.sim<-cbind(total.forec.arima.svm.train.sim,
	total.forec.arima.svm.train.sim.data)
}
total.forec.arima.svm.train.sim

###### Computing MSE, RMSE and MAE from each simulations Hybrid ARIMA+SVM
models in training data set
compute.crite.arima.svm.train.sim.func<-c()
for(i in 1:sim.num.mod.svm){
	crite.arima.svm.train.sim<-regr.eval(real.train.data[3:320],
	total.forec.arima.svm.train.sim[3:320,i],stat=c("mae","mse","rmse"))
	compute.crite.arima.svm.train.sim.func<-
	c(compute.crite.arima.svm.train.sim.func,crite.arima.svm.train.sim)
```

```
}

###### Show all simulations Hybrid ARIMA+SVM models in training data set with MSE, RMSE and MAE
sim.display.model.arima.svm<-c()
for(i in 1:sim.num.mod.svm){
	ARIMA.SVM<-c()
	count.arima.svm.sim.a<-(3*i)-2
	count.arima.svm.sim.b<-3*i
	ARIMA.SVM<-compute.crite.arima.svm.train.sim.func[count.arima.svm.sim.a:
	count.arima.svm.sim.b]
	sim.display.model.arima.svm<-rbind(sim.display.model.arima.svm,ARIMA.SVM)
}
sim.display.model.arima.svm

###### Calculate E[MSE], E[RMSE] and E[MAE] from all simulations Hybrid ARIMA+SVM models in training data set
crite.arima.svm.sim.train.func<-c()
	mean.mae.arima.svm.sim.train<-mean(sim.display.model.arima.svm[,1])
	mean.mse.arima.svm.sim.train<-mean(sim.display.model.arima.svm[,2])
	mean.rmse.arima.svm.sim.train<-mean(sim.display.model.arima.svm[,3])
	crite.arima.svm.sim.train.func<-c(mean.mae.arima.svm.sim.train,
	mean.mse.arima.svm.sim.train,mean.rmse.arima.svm.sim.train)
	names(crite.arima.svm.sim.train.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.svm.sim.train.func

###### Calculate Standard deviation from all simulations Hybrid ARIMA+SVM models in training data set
sd.mae.arima.svm.sim.train<-sd(sim.display.model.arima.svm[,1])
sd.mae.arima.svm.sim.train
sd.mse.arima.svm.sim.train<-sd(sim.display.model.arima.svm[,2])
```

```
sd.mse.arima.svm.sim.train
sd.rmse.arima.svm.sim.train<-sd(sim.display.model.arima.svm[,3])
sd.rmse.arima.svm.sim.train

###### Using residuals from all simulation ARIMA(1,1,1) models in testing data set for
forecasting in SVM models
all.model.svm.test.sim<-c()
force.svm.test.sim.data<-c()
for(l in 1:no.simmu){
	res.test.data.sim<-res.test.arima.sim[,l]
	sim.test.data.set<-data.frame(res.test.data.sim)
	sim.lag.value<-lag.best.train
	for(i in 1:no.iterations){
		sim.srange<-sigest(res.test.data.sim[2:136]~res.test.data.sim[1:135],
		data=sim.test.data.set)
		sim.svm2<-ksvm(res.test.data.sim[2:136]~res.test.data.sim[1:135],
		data=sim.test.data.set,kernel="rbfdot",kpar=list(sigma=sim.srange[2]),
		C=c.best.train)
		sim.preds2<-predict(sim.svm2)
		sim.mse.svm<-mse(res.test.data.sim[2:136],sim.preds2)
		sim.rmse.svm<-rmse(res.test.data.sim[2:136],sim.preds2)
		sim.mae.svm<-mae(res.test.data.sim[2:136],sim.preds2)
		sim.order.svm<-c(sim.lag.value,c.best.train,sim.srange[2],l)
		all.model.svm.test.sim<-c(all.model.svm.test.sim,sim.order.svm,i,
		sim.mse.svm,sim.rmse.svm,sim.mae.svm)
		force.svm.test.sim.data<-cbind(force.svm.test.sim.data,sim.preds2)
	}
}
head<-c("lag.value","c.value","sigma.value","no.simulation",
"no.iteration","MSE","RMSE","MAE")
names(all.model.svm.test.sim)<-rep(head,no.iterations*no.simmu)
```

```
###### Show all simulations support vector machine models in testing data set with MSE, RMSE and MAE
sim.display.model.svm.test<-c()
for(i in 1:sim.num.mod.svm){
	SVM<-c()
	count.svm.sim.a<-(8*i)-7
	count.svm.sim.b<-8*i
	SVM<-all.model.svm.test.sim[count.svm.sim.a:count.svm.sim.b]
	sim.display.model.svm.test<-rbind(sim.display.model.svm.test,SVM)
}
sim.display.model.svm.test

###### Combined forecasting data form each simulations ARIMA and SVM models ---> "Total forecasting" in testing data set
real.test.data<-c(rep(0,1),diff.adj.close.test)
forec.arima.test.sim<-c()
for(i in 1:no.simmu){
	forec.arima.test.sim.data<-c(rep(0,1),simmu.data.test[,i])
	rep.test<-matrix(rep(forec.arima.test.sim.data,no.iterations),ncol=no.iterations)
	forec.arima.test.sim<-cbind(forec.arima.test.sim,rep.test)
}
forec.arima.test.sim
forec.svm.test.sim<-c()
for(i in 1:sim.num.mod.svm){
	forec.svm.test.sim.data<-c(rep(0,2),force.svm.test.sim.data[,i])
	forec.svm.test.sim<-cbind(forec.svm.test.sim,forec.svm.test.sim.data)
}
forec.svm.test.sim
total.forec.arima.svm.test.sim<-c()
for(i in 1:sim.num.mod.svm){
```

```
        total.forec.arima.svm.test.sim.data<-c(rep(0,2),forec.arima.test.sim[3:137,i]+
        forec.svm.test.sim[3:137,i])
        total.forec.arima.svm.test.sim<-cbind(total.forec.arima.svm.test.sim,total.
        forec.arima.svm.test.sim.data)
}
total.forec.arima.svm.test.sim

###### Computing MSE, RMSE and MAE from each simulations Hybrid ARIMA+SVM
models in testing data set
compute.crite.arima.svm.test.sim.func<-c()
for(i in 1:sim.num.mod.svm){
        crite.arima.svm.test.sim<-regr.eval(real.test.data[3:137],
        total.forec.arima.svm.test.sim[3:137,i],stat=c("mae","mse","rmse"))
        compute.crite.arima.svm.test.sim.func<-
        c(compute.crite.arima.svm.test.sim.func,crite.arima.svm.test.sim)
}

###### Show all simulations Hybrid ARIMA+SVM models in testing data set with MSE,
RMSE and MAE
sim.display.model.arima.svm.test<-c()
for(i in 1:sim.num.mod.svm){
        ARIMA.SVM<-c()
        count.arima.svm.sim.a<-(3*i)-2
        count.arima.svm.sim.b<-3*i
        ARIMA.SVM<-compute.crite.arima.svm.test.sim.func[count.arima.svm.sim.a:
        count.arima.svm.sim.b]
        sim.display.model.arima.svm.test<-rbind(sim.display.model.arima.svm.test,
        ARIMA.SVM)
}
sim.display.model.arima.svm.test
```

```
###### Calculate E[MSE], E[RMSE] and E[MAE] from all simulations Hybrid ARIMA+SVM
models in testing data set
crite.arima.svm.sim.test.func<-c()
        mean.mae.arima.svm.sim.test<-mean(sim.display.model.arima.svm.test[,1])
        mean.mse.arima.svm.sim.test<-mean(sim.display.model.arima.svm.test[,2])
        mean.rmse.arima.svm.sim.test<-mean(sim.display.model.arima.svm.test[,3])
        crite.arima.svm.sim.test.func<-c(mean.mae.arima.svm.sim.test,
        mean.mse.arima.svm.sim.test,mean.rmse.arima.svm.sim.test)
        names(crite.arima.svm.sim.test.func)<-c("E[MAE]","E[MSE]","E[RMSE]")
crite.arima.svm.sim.test.func

###### Calculate Standard deviation from all simulations Hybrid ARIMA+SVM models
in testing data set
sd.mae.arima.svm.sim.test<-sd(sim.display.model.arima.svm.test[,1])
sd.mae.arima.svm.sim.test
sd.mse.arima.svm.sim.test<-sd(sim.display.model.arima.svm.test[,2])
sd.mse.arima.svm.sim.test
sd.rmse.arima.svm.sim.test<-sd(sim.display.model.arima.svm.test[,3])
sd.rmse.arima.svm.sim.test

##################################################################
######################### Comparing models ########################
##################################################################

###### Comparing 3 models from real SCB stock and simulation in training data set
real.train.data.set<-c(crite.arima.train,crite.arima.ann.train,crite.arima.svm.train)
simulation.train.data.set<-c(crite.arima.sim.train.func,crite.arima.ann.sim.train.func,
crite.arima.svm.sim.train.func)
criterion<-c("MAE","MSE","RMSE","MAE","MSE","RMSE","MAE","MSE","RMSE")
name.model<-c("ARIMA","ARIMA","ARIMA","ARIMA+ANN","ARIMA+ANN","ARIMA+ANN",
"ARIMA+SVM","ARIMA+SVM","ARIMA+SVM")
```

```
final.train.data.compare<-data.frame(cbind(name.model,criterion,real.train.data.set,
simulation.train.data.set))
final.train.data.compare

###### Comparing 3 models from real SCB stock and simulation in testing data set
real.test.data.set<-c(crite.arima.test,crite.arima.ann.test,crite.arima.svm.test)
simulation.test.data.set<-c(crite.arima.sim.test.func,crite.arima.ann.sim.test.func,
crite.arima.svm.sim.test.func)
criterion<-c("MAE","MSE","RMSE","MAE","MSE","RMSE","MAE","MSE","RMSE")
name.model<-c("ARIMA","ARIMA","ARIMA","ARIMA+ANN","ARIMA+ANN","ARIMA+ANN",
"ARIMA+SVM","ARIMA+SVM","ARIMA+SVM")
final.test.data.compare<-data.frame(cbind(name.model,criterion,real.test.data.set,
simulation.test.data.set))
final.test.data.compare

####################################################################
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

คำสั่งและผลลัพธ์จากการวิเคราะห์ข้อมูลด้วยโปรแกรม SAS

ตัวอย่างการทดสอบหานัยสำคัญทางสถิติโดยใช้แผนแบบทดลองสุ่มในบล็อกสมบูรณ์ไม่มีซ้ำ (Randomized complete block design without replication) ซึ่งเป็น Dependent Test เพื่อทำการเปรียบเทียบตัวแบบ 3 ตัวแบบที่ไม่เป็นอิสระต่อกัน (Dependent) ซึ่งจัดเป็นปัจจัยทดลอง (Treatment) ได้แก่ ตัวแบบ ARIMA, ตัวแบบผสมระหว่าง ARIMA กับเครือข่ายประสาทเทียม และตัวแบบผสมระหว่าง ARIMA กับซัพพอร์ทเวกเตอร์แมชชีน และชุดข้อมูล (Data set) เป็นปัจจัยของการจัดแบ่งบล็อก (Blocking factor) ได้แก่ ชุดข้อมูลฝึกสอน (Training data set) และชุดข้อมูลทดสอบ (Testing data set) โดยมีตัวแปรตอบสนอง คือ ค่าเฉลี่ยของ RMSE โดยใช้โปรแกรม SAS เวอร์ชัน 9.3 ซึ่งมีคำสั่งดังต่อไปนี้

<u>กรณีที่ 1</u> : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,1)

```
option ls=82 ps=60 nodate center;
data sim_arima011;
        input three_model $ dataset $ rmse;
        datalines;
                arima           train           4.2232
                arima           test            5.6269
                arima_ann       train           3.9793
                arima_ann       test            5.5485
                arima_svm       train           4.1769
                arima_svm       test            5.6008
;run;
proc print data=sim_arima011; run;
proc anova data=sim_arima011;
        class three_model dataset;
        model rmse=three_model dataset;
        means three_model/cldiff clm lsd;
        run;
```

ตารางที่ 1 ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.25056988	1.08352329	265.84	0.0037
Error	2	0.00815173	0.00407586		
Corrected Total	5	3.25872161			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.02859484	0.01429742	3.51	0.2218
dataset	1	3.22197504	3.22197504	790.50	0.0013

กรณีที่ 2 : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(0,1,2)

```
option ls=82 ps=60 nodate center;
data sim_arima012;
    input three_model $ dataset $ rmse;
    datalines;
        arima       train   4.1057
        arima       test    5.5872
        arima_ann   train   4.1428
        arima_ann   test    5.7119
        arima_svm   train   4.0634
        arima_svm   test    5.5661
;run;
proc print data=sim_arima012; run;
proc anova data=sim_arima012;
    class three_model dataset;
```

```
        model rmse=three_model dataset;
        means three_model/cldiff clm lsd;
        run;
```

<u>ตารางที่ 2</u> ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบ จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,2)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.46890912	1.15630304	1107.20	0.0009
Error	2	0.00208869	0.00104435		
Corrected Total	5	3.47099782			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.01348564	0.00674282	6.46	0.1341
dataset	1	3.45542348	3.45542348	3308.69	0.0003

<u>กรณีที่ 3</u> : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(1,1,0)

```
option ls=82 ps=60 nodate center;
data sim_arima110;
        input three_model $ dataset $ rmse;
        datalines;
                arima          train          6.4909
                arima          test           9.5688
                arima_ann      train          5.2518
                arima_ann      test           7.7504
                arima_svm      train          6.3993
                arima_svm      test           9.4765
```

```
;run;
proc print data=sim_arima110; run;
proc anova data=sim_arima110;
        class three_model dataset;
        model rmse=three_model dataset;
        means three_model/cldiff clm lsd;
        run;
```

ตารางที่ 3 ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(1,1,0)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	15.42103768	5.14034589	92.02	0.0108
Error	2	0.11172782	0.05586391		
Corrected Total	5	15.53276551			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	2.93995040	1.46997520	26.31	0.0366
dataset	1	12.48108728	12.48108728	223.42	0.0044

t Tests (LSD) for rmse					
Comparsions significant at the 0.05 level are indicated by ***.					
three_model Comparison		Difference Between Means	95% Confidence Limits		
arima	arima_svm	0.0919	-0.9250	1.1089	
arima	arima_ann	1.5287	0.5118	2.5457	***
arima_svm	arima	-0.0919	-1.1089	0.9250	
arima_svm	arima_ann	1.4368	0.4198	2.4538	***
arima_ann	arima	-1.5287	-2.5457	-0.5118	***
arima_ann	arima_svm	-1.4368	-2.4538	-0.4198	***

<u>กรณีที่ 4</u> : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(1,1,1)

```
option ls=82 ps=60 nodate center;
data sim_arima111;
	input three_model $ dataset $ rmse;
	datalines;
		arima		train		4.0818
		arima		test		5.5969
		arima_ann	train		4.1643
		arima_ann	test		5.7522
		arima_svm	train		4.0327
		arima_svm	test		5.5726
;run;
proc print data=sim_arima111; run;
proc anova data=sim_arima111;
	class three_model dataset;
	model rmse=three_model dataset;
	means three_model/cldiff clm lsd;
	run;
```

ตารางที่ 4 ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(1,1,1)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.61921704	1.20640568	1761.42	0.0006
Error	2	0.00136981	0.00068491		
Corrected Total	5	3.62058685			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.02646364	0.01323182	19.32	0.0492
dataset	1	3.59275340	3.59275340	5245.61	0.0002

t Tests (LSD) for rmse					
Comparsions significant at the 0.05 level are indicated by ***.					
three_model Comparison		Difference Between Means	95% Confidence Limits		
arima_ann	arima	0.11890	0.00630	0.23150	***
arima_ann	arima_svm	0.15560	0.04300	0.26820	***
arima	arima_ann	-0.11890	-0.23150	-0.00630	***
arima	arima_svm	0.03670	-0.07590	0.14930	
arima_svm	arima_ann	-0.15560	-0.26820	-0.04300	***
arima_svm	arima	-0.03670	-0.14930	0.07590	

กรณีที่ 5 : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(1,1,2)

option ls=82 ps=60 nodate center;

```
data sim_arima112;
        input three_model $ dataset $ rmse;
        datalines;
                arima           train           4.2301
                arima           test            5.6221
                arima_ann       train           3.9372
                arima_ann       test            5.5597
                arima_svm       train           4.1746
                arima_svm       test            5.6091
;run;
proc print data=sim_arima112; run;
proc anova data=sim_arima112;
        class three_model dataset;
        model rmse=three_model dataset;
        means three_model/cldiff clm lsd;   run;
```

ตารางที่ 5 ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(1,1,2)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.33446426	1.11148809	147.74	0.0067
Error	2	0.01504675	0.00752338		
Corrected Total	5	3.34951101			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.03553076	0.01776538	2.36	0.2975
dataset	1	3.29893350	3.29893350	438.49	0.0023

<u>กรณีที่ 6</u> : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(2,1,0)

```
option ls=82 ps=60 nodate center;
data sim_arima210;
        input three_model $ dataset $ rmse;
        datalines;
                arima        train        5.6640
                arima        test         6.5339
                arima_ann    train        4.8786
                arima_ann    test         6.0926
                arima_svm    train        5.5665
                arima_svm    test         6.5044
;run;
proc print data=sim_arima210; run;
proc anova data=sim_arima210;
        class three_model dataset;
        model rmse=three_model dataset;
        means three_model/cldiff clm lsd;
        run;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

<u>ตารางที่ 6</u> ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบ จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(2,1,0)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	1.97692287	0.65897429	39.69	0.0247
Error	2	0.03321000	0.01660500		
Corrected Total	5	2.01013287			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.45504366	0.22752183	13.70	0.0680
dataset	1	1.52187921	1.52187921	91.65	0.0107

กรณีที่ 7 : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(2,1,1)

```
option ls=82 ps=60 nodate center;
data sim_arima211;
	input three_model $ dataset $ rmse;
	datalines;
		arima		train	4.1540
		arima		test	5.6742
		arima_ann	train	3.9736
		arima_ann	test	5.6107
		arima_svm	train	4.1000
		arima_svm	test	5.6433
;run;
proc print data=sim_arima211; run;
proc anova data=sim_arima211;
	class three_model dataset;
	model rmse=three_model dataset;
	means three_model/cldiff clm lsd;
	run;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 7 ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(2,1,1)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.69793610	1.23264537	643.18	0.0016
Error	2	0.00383294	0.00191647		
Corrected Total	5	3.70176904			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.01532937	0.00766468	4.00	0.2000
dataset	1	3.68260673	3.68260673	1921.56	0.0005

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

กรณีที่ 8 : ใช้ชุดข้อมูลอนุกรมเวลาที่จำลองด้วยตัวแบบ ARIMA(2,1,2)

```
option ls=82 ps=60 nodate center;
data sim_arima212;
	input three_model $ dataset $ rmse;
	datalines;
		arima		train		4.0659
		arima		test		5.5835
		arima_ann	train		4.0682
		arima_ann	test		5.5998
		arima_svm	train		4.0156
		arima_svm	test		5.5492
;run;
proc print data=sim_arima212; run;
proc anova data=sim_arima212;
	class three_model dataset;
```

```
model rmse=three_model dataset;
means three_model/cldiff clm lsd;
run;
```

<u>ตารางที่ 8</u> ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบ จากการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(2,1,2)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.50336820	1.16778940	30731.3	<.0001
Error	2	0.00007600	0.00003800		
Corrected Total	5	3.50344420			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.00302556	0.00151278	39.81	0.0245
dataset	1	3.50034264	3.50034264	92114.3	<.0001

t Tests (LSD) for rmse					
Comparsions significant at the 0.05 level are indicated by ***.					
three_model Comparison		Difference Between Means	95% Confidence Limits		
arima_ann	arima	0.009300	-0.017223	0.035823	
arima_ann	arima_svm	0.051600	0.025077	0.078123	***
arima	arima_ann	-0.009300	-0.035823	0.017223	
arima	arima_svm	0.042300	0.015777	0.068823	***
arima_svm	arima_ann	-0.051600	-0.078123	-0.025077	***
arima_svm	arima	-0.042300	-0.068823	-0.015777	***

จุฬาลงกรณ์มหาวิทยาลัย

<u>กรณีที่ 9</u> : ใช้ชุดข้อมูลจริงของราคาปิดหุ้น SCB ที่มีลักษณะอนุกรมเวลาสอดคล้องกับตัวแบบ ARIMA(1,1,1)

```
option ls=82 ps=60 nodate center;
data arima111;
	input three_model $ dataset $ rmse;
	datalines;
		arima		train		3.8049
		arima		test		5.4075
		arima_ann	train		3.8059
		arima_ann	test		5.4223
		arima_svm	train		3.7181
		arima_svm	test		5.3389
;run;
proc print data= arima111; run;
proc anova data= arima111;
	class three_model dataset;
	model rmse=three_model dataset;
	means three_model/cldiff clm lsd;
	run;
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

<u>ตารางที่ 9</u> ผลลัพธ์ของการหานัยสำคัญทางสถิติในการเปรียบเทียบตัวแบบจากชุดข้อมูลจริงของราคาปิดหุ้น SCB ที่มีลักษณะอนุกรมเวลาสอดคล้องกับตัวแบบ ARIMA(1,1,1)

The ANOVA Procedure					
Dependent Variable: rmse					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	3.91289538	1.30429846	28928.7	<.0001
Error	2	0.00009017	0.00004509		
Corrected Total	5	3.91298555			

Source	DF	ANOVA SS	Mean Square	F Value	Pr > F
three_model	2	0.00895137	0.00447569	99.27	0.0100
dataset	1	3.90394401	3.90394401	86587.6	<.0001

t Tests (LSD) for rmse					
Comparsions significant at the 0.05 level are indicated by ***.					
three_model Comparison		Difference Between Means	95% Confidence Limits		
arima_ann	arima	0.007900	-0.020991	0.036791	
arima_ann	arima_svm	0.085600	0.056709	0.114491	***
arima	arima_ann	-0.007900	-0.036791	0.020991	
arima	arima_svm	0.077700	0.048809	0.106591	***
arima_svm	arima_ann	-0.085600	-0.114491	-0.056709	***
arima_svm	arima	-0.077700	-0.106591	-0.048809	***

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นายชญานิน บุญมานะ เกิดวันอาทิตย์ที่ 10 มิถุนายน พ.ศ. 2533 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (วท.บ.) เกียรตินิยมอันดับหนึ่ง สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์และสถิติ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2555 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาสถิติ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2557

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University