การเปรียบเทียบความแม่นยำของการพยากรณ์อนุกรมเวลาระหว่างตัวแบบผสมและตัวแบบเดี่ยว

นายภัทร วรภู

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

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.

COMPARISON OF TIME SERIES FORECASTING ACCURACY BETWEEN THE HYBRID AND INDIVIDUAL MODEL

Mr. Pattara Woraphu

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Industrial Engineering
Department of Industrial Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การเปรียบเทียบความแม่นยำของการพยากรณ์อนุกรมเวลาระหว่างตัวแบบผสมและตัวแบบเดี่ยว
โดย	นายภัทร วรภู
สาขาวิชา	วิศวกรรมอุตสาหการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ จิรพัฒน์ เงาประเสริฐวงศ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อังศุมาลิน เสนจันทร์ฒิไชย)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ จิรพัฒน์ เงาประเสริฐวงศ์)

.....กรรมการ

(รองศาสตราจารย์ ดร.ปารเมศ ชุติมา)

.....กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.วันชัย ริจิรวนิช)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภัทร วรภู : การเปรียบเทียบความแม่นยำของการพยากรณ์อนุกรมเวลาระหว่างตัวแบบผสมและตัวแบบเดี่ยว. (COMPARISON OF TIME SERIES FORECASTING ACCURACY BETWEEN THE HYBRID AND INDIVIDUAL MODEL) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. จิรพัฒน์ เงาประเสริฐวงศ์, 138 หน้า.

การพยากรณ์ข้อมูลอนุกรมเวลาเป็นงานที่มีความสำคัญเป็นจุดเริ่มต้นในการบริหารจัดการงานด้านต่างๆ การพยากรณ์ที่มีความแม่นยำจะทำให้การบริหารจัดการมีประสิทธิภาพมากยิ่งขึ้น ในปัจจุบันตัวแบบพยากรณ์อนุกรมเวลาที่แบ่งข้อมูลออกเป็นสองส่วน คือ ส่วนที่เป็นเชิงเส้นตรงและส่วนที่ไม่เป็นเชิงเส้นตรงหรือเรียกตัวแบบนี้ว่าตัวแบบ Traditional hybrid กำลังได้รับความนิยมมีความแม่นยำในการพยากรณ์ดีกว่าตัวแบบเดี่ยวๆ ในหลายงานวิจัย แต่ขณะเดียวกันกลับมีงานวิจัยที่ค้านตัวแบบผสมนั้นว่าไม่ได้ดีไปกว่าตัวแบบเดี่ยวในทุกๆ ข้อมูล

งานวิจัยชิ้นนี้เลือกตัวแบบ ARIMA ซึ่งเป็นตัวแบบเชิงเส้นตรงที่ใช้พยากรณ์ข้อมูลอนุกรมเวลาที่เป็นที่นิยมมากที่สุดและตัวแบบ ANN และ SVM ซึ่งเป็นตัวแบบที่ไม่เป็นเชิงเส้นตรงที่มีการนำมาใช้กันมาก นำมาผสมกันเรียกว่าตัวแบบ ARIMAANN และ ARIMASVM โดยทำการศึกษาเชิงเปรียบเทียบความแม่นยำของตัวแบบผสม ARIMAANN และ ARIMASVM เปรียบเทียบความแม่นยำกับตัวแบบเดี่ยวๆ ทั้ง ARIMA, ANN และ SVM พร้อมทั้งนำเสนอตัวแบบผสมอีกลักษณะหนึ่ง เรียกว่าตัวแบบ Combined (ARIMA+ANN+SVM) นำมาเปรียบเทียบความแม่นยำกับตัวแบบ ARIMAANN, ARIMASVM และเปรียบเทียบความแม่นยำกับตัวแบบเดี่ยว ทำการทดลองภายใต้ข้อมูลจำนวน 2 กลุ่มคือกลุ่มแรกเป็นข้อมูลที่ใช้กันในงานวิจัยการพยากรณ์อนุกรมเวลาจำนวน 3 ชุดข้อมูล ขณะที่อีกกลุ่มคือข้อมูลอนุกรมเวลาจริงในประเทศไทยที่มีความหลากหลายทั้งในแง่ของการใช้ข้อมูลและลักษณะโครงสร้างของข้อมูลจำนวน 7 ชุดข้อมูล ตัวแบบทั้ง 6 ตัวแบบ ประกอบด้วย ARIMA, ANN, SVM, ARIMAANN, ARIMASVM และตัวแบบ Combined (ARIMA+ANN+SVM) ถูกทดสอบด้วยข้อมูลรวมทั้งสิ้น 10 ชุด ประเมินผลความแม่นยำด้วย RMSE, MSE, MAE และ MAPE

ผลการทดลองพบว่า ตัวแบบ ARIMAANN ไม่ได้แม่นยำกว่าตัวแบบ ARIMA หรือ ANN ในทุกชุดข้อมูลและบางครั้งก็ให้ความแม่นยำที่ต่ำกว่าเช่นเดียวกับตัวแบบ ARIMASVM ขณะที่ตัวแบบ Combined (ARIMA+ANN+SVM) มีความแม่นยำกว่าตัวแบบ ARIMAANN และ ARIMASVM 9 ชุดข้อมูลและแม่นยำกว่าตัวแบบเดี่ยวทั้ง ARIMA, ANN และ SVM จำนวน 8 ชุดข้อมูล มีความแม่นยำใกล้เคียงกับตัวแบบเดี่ยวที่ดีที่สุด 2 ชุดข้อมูล โดยไม่มีข้อมูลชุดใดที่แย่กว่าเลย ดังนั้นตัวแบบ Combined (ARIMA+ANN+SVM) เป็นตัวแบบที่เป็นทางเลือกที่ดีโดยเฉพาะงานที่ต้องการความแม่นยำของการพยากรณ์สูง

ภาควิชา วิศวกรรมอุตสาหการ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมอุตสาหการ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

5370652321 : MAJOR INDUSTRIAL ENGINEERING
KEYWORDS: ARIMA / ANN / SVM / ARIMAANN / ARIMASVM / COMBINED MODEL

PATTARA WORAPHU: COMPARISON OF TIME SERIES FORECASTING ACCURACY BETWEEN THE HYBRID AND INDIVIDUAL MODEL. ADVISOR: ASSOC. PROF. JEERAPAT NGAOPRASERTWONG, 138 pp.

Time series forecasting is the important task, which is a starting point in many operational processes. The high forecasting accuracy will increase the efficiency of their operation. At present, the forecasting models which divide data into two components, linear and nonlinear components, called the traditional hybrid model is being popular. Many researchers claim that this hybrid approach is outperformance the individual models but some researchers indicated that this hybrid is not better or even worse.

In this study, ARIMA is selected as linear model and ANN and SVM as nonlinear models to compare the accuracy between the ARIMAANN and their individual models, ARIMA and ANN and compare ARIMASVM with both ARIMA and SVM. By proposing another hybrid approach called combined (ARIMA+ANN+SVM) model and compare the accuracy to the ARIMAANN, ARIMASVM and their individual models. The experiment is done on 10 datasets which can be divided into 2 groups, the first 3 datasets is well-known datasets shown in some literature, the second 7 datasets is the real datasets in Thailand in many fields of study and different in characteristics.

The result of this study shows that both ARIMAANN and ARIMASVM do not always outperform ARIMA or ANN and ARIMA or SVM in all datasets and sometimes even worse. However, the proposed combined (ARIMA+ANN+SVM) model outperforms the ARIMAANN and ARIMASVM in 9 datasets and better than the individual models in 8 datasets. For the rest 2 datasets, the accuracy is close to the best of individual models and none of datasets underperform the individual models. In conclusion, the combined (ARIMA+ANN+SVM) is a good alternative model for the important tasks that need a high forecasting accuracy.

Department: Industrial Engineering Student's Signature .....
Field of Study: Industrial Engineering Advisor's Signature .....
Academic Year: 2013

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ รองศาสตราจารย์ จิรพัฒน์ เงาประเสริฐวงศ์ อาจารย์ที่ปรึกษาเป็นอย่างสูงที่กรุณาให้คำแนะนำ ช่วยเหลือและร่วมติดตามการทดลองอย่างใกล้ชิดในทุกขั้นตอน ทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีและขอขอบพระคุณกรรมการสอบวิทยานิพนธ์ซึ่งประกอบด้วย รศ.ดร.ปารเมศ ชุติมา ผศ.ดร.อังศุมาลิน เสนจันทร์ฒิไชย และรศ.ดร.วันชัย ริจิรวนิช ที่ได้สละเวลาอันมีค่าเพื่อเป็นเพื่อกรรมการสอบวิทยานิพนธ์จนทำให้วิทยานิพนธ์มีความสมบูรณ์ในที่สุด

ผู้วิจัยขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่เปรียบเสมือนทุกอย่างของชีวิต ที่ให้กำลังใจ ความรัก ให้คำปรึกษา และช่วยเหลือในทุกๆด้าน จนทำให้ผู้ทำวิจัยสามารถทำการวิจัยจนสำเร็จการศึกษา

ขอขอบคุณ คุณสถิตเทพ สังข์ทอง คุณกรกฎ วัฒนวีร์ คุณเมธาวัฒน์ สงพัฒน์แก้วที่ให้ความช่วยเหลือตลอดมา

ถ้าวิทยานิพนธ์เล่มนี้ผิดพลาดประการใดผู้จัดทำต้องขออภัยมา ณ ที่นี้ด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.1.1 การเปรียบเทียบความแม่นยำของตัวแบบเดี่ยว 1

1.1.2 การเกิดขึ้นของตัวแบบผสม 2

1.1.3 การเปรียบเทียบความแม่นยำของตัวแบบผสมเทียบกับตัวแบบเดี่ยว 3

1.1.4 มูลเหตุของงานวิจัย 3

1.2 วัตถุประสงค์งานวิจัย 4

1.3 ขอบเขตของงานวิจัย 4

1.4 ข้อมูลอนุกรมเวลาที่นำมาใช้พยากรณ์ 5

1.4.1 The Wolf's sunspot dataจำนวนข้อมูล 288 ตัวอย่าง 5

1.4.2 The Canadian lynx dataจำนวนข้อมูล 114 ตัวอย่าง 6

1.4.3 The British pound/US dollar exchange rate data จำนวนข้อมูล 731 ตัวอย่าง 6

1.4.4 ข้อมูลปริมาณการใช้พลังงานไฟฟ้ารายเดือนทั้งประทศไทย (หน่วย:ล้านหน่วย) จำนวนข้อมูล 151 ตัวอย่าง 7

1.4.5 ข้อมูลยอดขายรถจักรยานยนต์รายเดือนทั้งประทศไทย (หน่วย:คัน) จำนวนข้อมูล 151 ตัวอย่าง 7

1.4.6 ข้อมูลอัตราแลกเปลี่ยนรายวันของเงินสกุลดอลลาร์สหรัฐและเงินบาท (หน่วย:บาท/1 US dollar) จำนวนข้อมูล 173 ตัวอย่าง 8

1.4.7 ข้อมูลรายวันราคาทองคำหนักหนึ่งบาท (หน่วย:บาท/ทองคำหนัก 1 บาท) จำนวนข้อมูล 220 ตัวอย่าง 8

1.4.8 ข้อมูลรายวันความเร็วลมโดยเฉลี่ยในรอบวัน (นอต) จำนวนข้อมูล 365 ตัวอย่าง 9

1.4.9 ข้อมูลปริมาณน้ำฝนรายเดือน (มิลลิเมตร) จำนวนข้อมูล 130 ตัวอย่าง 9

1.4.10 ข้อมูลระดับอุณหภูมิโดยเฉลี่ยรายเดือน (เซลเซียส) จำนวนข้อมูล 130 ตัวอย่าง 10

หน้า

1.5 ขั้นตอนการวิจัยและดำเนินงาน 10
1.6 ผลที่คาดว่าจะได้รับ 11
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 12
2.1 ตัวแบบ Autoregressive Integrated Moving Average (ARIMA) 12
2.1.1 ค่าอัตตสัมพันธ์ (Autocorrelation) และค่าอัตตสัมพันธ์บางส่วน (Partial autocorrelation) 12
2.1.2 ขั้นตอนการสร้างตัวแบบ ARIMA 13
2.2 ตัวแบบ Artificial neural network (ANN) 19
2.2.1 แรงบันดาลของ ANN 19
2.2.2 การประยุกต์ใช้ Neural network ในงานต่างๆ ได้แก่ 20
2.2.3 เครือข่าย Feedforward (Feedforward neural network) 21
2.2.4 ตัวแปรที่ใช้ในการสร้างตัวแบบ 23
2.2.5 การสำรวจข้อมูล (Data exploration) 24
2.2.6 การแปลงข้อมูลก่อนและหลังกระบวนการเรียนรู้ (Data preprocessing and postprocessing) 25
2.2.7 การแบ่งข้อมูลเพื่อใช้ในการเรียนรู้และทดสอบ (Data set splitting) 26
2.2.8 ฟังก์ชัน Activation (Activation function) 26
2.2.9 เทคนิคการเรียนรู้ 28
2.2.10 เทคนิคการปรับปรุงค่า Weights 28
2.2.11 อัลกอริทึม กระบวนการเรียนรู้ (Training Algorithm) 29
2.2.12 จำนวนรอบการเทรนซ้ำ (Experiment repeat) 30
2.2.13 เงื่อนไขหยุด (Stopping Criteria) 30
2.3 ตัวแบบ ANN สำหรับการพยากรณ์อนุกรมเวลา (ANN for time series forecasting) 31
2.3.1 โครงสร้างตัวแบบ ANN สำหรับการพยากรณ์อนุกรมเวลา (ANN for time series architecture) 33
2.4 ตัวแบบ Support vector machine (SVM) 34
2.4.1 Support vector machine for regression 38
2.4.2 Kernel function 39

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

2.4.3 ตัวแบบ Support vector machine สำหรับพยากรณ์ข้อมูลอนุกรมเวลา..... 40
2.5 ขั้นตอนการสร้างตัวแบบ Traditional hybrid 41
2.6 ขั้นตอนการสร้างตัวแบบ Hybrid combined..... 42
2.7 การประเมินความแม่นยำของตัวแบบ (Evaluation performance)..... 44
2.8 ทบทวนวรรณกรรม 45
บทที่ 3 วิธีการทดลอง 48
3.1 การสร้างตัวแบบ ARIMA..... 49
3.1.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ ARIMA 49
3.2 การสร้างตัวแบบ ANN..... 56
3.2.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ ANN..... 57
3.3 การสร้างตัวแบบ SVM..... 60
3.3.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ SVM..... 61
3.4 การสร้างตัวแบบ Traditional hybrid ARIMAANN และ ARIMASVM 64
3.5 การสร้างตัวแบบ Hybrid combined (ARIMA+ANN+SVM) 65
บทที่ 4 ผลการทดลองและการวิเคราะห์..... 68
4.1 ผลการทดลอง..... 68
4.2 การวิเคราะห์ความไวสำหรับตัวแบบ ANN..... 78
4.3 การวิเคราะห์ความไวของพารามิเตอร์ สำหรับตัวแบบ SVM..... 80
4.4 เวลาที่ใช้ในการคำนวณของแต่ละตัวแบบ 84
บทที่ 5 โปรแกรม GUI สำหรับพยากรณ์ตัวแบบผสม 86
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ 105
6.1 เปรียบเทียบความแม่นยำของตัวแบบ Traditional hybrid กับตัวแบบเดี่ยว 105
6.2 เปรียบเทียบความแม่นยำของตัวแบบ Hybrid combined (ARIMA+ANN+SVM) กับตัวแบบ Traditional hybrid 105
6.3 เปรียบเทียบความแม่นยำของตัวแบบ Hybrid combined (ARIMA+ANN+SVM) กับตัวแบบเดี่ยว ARIMA, ANN และ SVM..... 106
6.4 เปรียบเทียบประสิทธิภาพตัวแบบตามลักษณะข้อมูล..... 106
6.5 บทวิจารณ์และข้อเสนอแนะตัวแบบเดี่ยว..... 107

หน้า

6.6 บทวิจารณ์และข้อเสนอแนะตัวแบบ Hybrid 108
6.7 บทวิจารณ์และข้อเสนอแนะตัวแบบ Hybrid combined 109
รายการอ้างอิง 110
ประวัติผู้เขียนวิทยานิพนธ์ 138

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 1.1 เปรียบเทียบความแม่นยำระหว่างตัวแบบ Hybrid combined และ ตัวแบบเดี่ยว 3
ตารางที่ 3.1 แสดงจำนวนข้อมูลในชุด Train และ Test รวมถึงชื่อตัวแปรที่ใช้..... 48
ตารางที่ 3.2 ค่า AICc ในแต่ละอันดับของตัวแบบ ARIMA 53
ตารางที่ 3.3 ค่า MSE ในโครงข่ายต่างๆ เมื่อ Lag=1 ทำซ้ำ 50 รอบ..... 59
ตารางที่ 3.4 ค่า MSE ในโครงข่ายต่างๆ เมื่อ Lag=2 ทำซ้ำ 50 รอบ..... 59
ตารางที่ 3.5 ค่า MSE ที่พารามิเตอร์ต่างๆ ของตัวแบบ SVM เมื่อ Lag=1 และ σ=0.00001 63
ตารางที่ 3.6 ค่าเศษเหลือที่ได้จากตัวแบบ ARIMA..... 64
ตารางที่ 3.7 การได้มาซึ่งค่าพยากรณ์รวมและเศษเหลือรวมตัวแบบ ARIMAANN 65
ตารางที่ 3.8 ค่าพยากรณ์ของตัวแบบ ARIMA, ANN และ SVM 65
ตารางที่ 3.9 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0 66
ตารางที่ 3.10 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.001 67
ตารางที่ 3.11 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.998 67
ตารางที่ 3.12 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.999 67
ตารางที่ 3.13 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=1 67
ตารางที่ 4.1 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Sunspot 68
ตารางที่ 4.2 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Canadian lynx 69
ตารางที่ 4.3 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล GBP/USD 70
ตารางที่ 4.4 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Electricuse 71
ตารางที่ 4.5 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Motorcycle 72
ตารางที่ 4.6 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล USexchange..... 73
ตารางที่ 4.7 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Gold 74
ตารางที่ 4.8 ค่า RMSE, MSE, MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Windspeed..... 75
ตารางที่ 4.9 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Meanrain 76
ตารางที่ 4.10 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Maxtemp 77
ตารางที่ 4.11 ค่า MSE ที่พารามิเตอร์ C มีค่าต่างๆ..... 81
ตารางที่ 4.12 ค่า MSE ที่พารามิเตอร์ Epsilon มีค่าต่างๆ 82
ตารางที่ 4.13 ค่า MSE ที่พารามิเตอร์ Sigma มีค่าต่างๆ..... 84
ตารางที่ 4.14 เวลาที่ใช้ในการคำนวณของแต่ละตัวแบบ 85
ตารางที่ 6.1 เรียงลำดับประสิทธิภาพของตัวแบบตามลักษณะข้อมูล..... 107

สารบัญภาพ

หน้า

ภาพที่ 1.1 จำนวนจุดดับดวงอาทิตย์ที่เกิดขึ้นใหม่รายปี ตั้งแต่ค.ศ. 1700 ถึงค.ศ. 1987 (Zhang, 2003)..... 5
ภาพที่ 1.2 จำนวนแมวน้ำที่ถูกจับที่แม่น้ำ Mackensize ในประเทศแคนาดารายปี ตั้งแต่ค.ศ. 1821 ถึงค.ศ. 1934 (Zhang, 2003)..... 6
ภาพที่ 1.3 อัตราแลกเปลี่ยนรายวันเงินสกุล British pound ต่อ US dollar ตั้งแต่ค.ศ. 1821 ถึงค.ศ. 1934 (Zhang, 2003)..... 6
ภาพที่ 1.4 ปริมาณการใช้พลังงานไฟฟ้ารวม (ล้านหน่วย) ทั้งประเทศไทย ตั้งแต่เดือนมกราคม 2543 ถึง กรกฎาคม 2555 (ธนาคารแห่งประเทศไทย, 2555a)..... 7
ภาพที่ 1.5 ปริมาณการจำหน่ายรถมอเตอร์ไซต์ (คัน) ตั้งแต่เดือนมกราคม 2543 ถึง กรกฎาคม 2555 (ธนาคารแห่งประเทศไทย, 2555a)..... 7
ภาพที่ 1.6 ราคาขายถัวเฉลี่ย US Dollar/Baht ตั้งแต่วันที่ 1 มกราคม 2543 ถึง 14 กันยายน 2555 (ธนาคารแห่งประเทศไทย, 2555b)..... 8
ภาพที่ 1.7 ราคารับซื้อทองคำหนึ่งบาทรายวัน ตั้งแต่วันที่ 1 มกราคม 2543 ถึง 14 กันยายน 2555 (ห้างขายทองทองใบเยาวราช, 2555)..... 8
ภาพที่ 1.8 ความเร็วลมโดยเฉลี่ยในรอบวัน (นอต) ตั้งแต่วันที่ 1 มกราคม ถึง 31 ธันวาคม 2554 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)..... 9
ภาพที่ 1.9 ปริมาณน้ำฝนภาคใต้ฝั่งตะวันตกตั้งแต่ มกราคม 2545 ถึงตุลาคม 2555 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)..... 9
ภาพที่ 1.10 ระดับอุณหภูมิสูงสุดภาคเหนือ ตั้งแต่เดือนมกราคมคม 2545 ถึงเดือนตุลาคม 2555 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)..... 10
ภาพที่ 2.1 แสดงแผนผังขั้นตอนการสร้างตัวแบบ ARIMA..... 13
ภาพที่ 2.2 ตัวอย่างข้อมูลที่ไม่มีคุณสมบัติ Stationary..... 15
ภาพที่ 2.3 ตัวอย่างข้อมูลภายหลังทำการ Difference..... 15
ภาพที่ 2.4 กราฟ ACF และ PACF สำหรับตัวแบบ AR(1)..... 16
ภาพที่ 2.5 กราฟ ACF และ PACF สำหรับตัวแบบ MA(1)..... 16
ภาพที่ 2.6 กราฟ ACF และ PACF สำหรับตัวแบบ ARMA(1,1)..... 17
ภาพที่ 2.7 หลักการทำงานของตัวแบบ ANN (Beale et al., 2013)..... 19
ภาพที่ 2.8 โครงสร้างของ Feedforward network..... 22
ภาพที่ 2.9 Linear Transfer Function (Beale et al., 2013)..... 27
ภาพที่ 2.10 Log-Sigmoid Transfer Function (Beale et al., 2013)..... 27
ภาพที่ 2.11 Tan-Sigmoid Transfer Function (Beale et al., 2013)..... 28
ภาพที่ 2.12 แผนผังตัวแปรอิสระและตัวแปรตามสำหรับตัวแบบ ANN ในการพยากรณ์ข้อมูลอนุกรมเวลา..... 32

ภาพที่ 2.13 แผนผังของตัวแบบ Feedforward network ในการพยากรณ์ข้อมูลอนุกรมเวลา 33
ภาพที่ 2.14 เส้นตรงแบ่งกลุ่มข้อมูลที่มี 2 ตัวแปร สำหรับตัวแบบ SVM (Williams, 2011)..... 35
ภาพที่ 2.15 เส้นตรงที่ใช้แบ่งกลุ่มข้อมูลที่เป็นไปได้ สำหรับตัวแบบ SVM (Williams, 2011)..... 36
ภาพที่ 2.16 เส้นตรงที่ใช้แบ่งกลุ่มที่ดีที่สุด (Williams, 2011) 37
ภาพที่ 2.17 รูปแบบการแปลงจุด เพื่อให้สามารถหาเส้นตรงมาแบ่ง..... 37
ภาพที่ 2.18 ขั้นตอนการสร้างตัวแบบ Traditional hybrid 41
ภาพที่ 2.19 ขั้นตอนการสร้างตัวแบบ Combined 43
ภาพที่ 3.1 ข้อมูล Sunspot ที่ทำการแบ่งข้อมูลออกเป็นสองกลุ่ม..... 49
ภาพที่ 3.2 การทดสอบคุณสมบัติ Stationary ด้วยสถิติทดสอบ ADF..... 50
ภาพที่ 3.3 อันดับ ARIMA ที่ดีที่สุด..... 53
ภาพที่ 3.4 การทดสอบความเป็นอิสระของข้อมูลอนุกรมเวลาด้วย Box-Pierce and Ljung-Box Tests 54
ภาพที่ 3.5 ACF ของค่าเศษเหลือ..... 54
ภาพที่ 3.6 Scatter plot ของค่าเศษเหลือ..... 55
ภาพที่ 3.7 Histogram ของค่าเศษเหลือ 55
ภาพที่ 3.8 เปรียบเทียบค่าจริงและค่าพยากรณ์ ตัวแบบ ARIMA..... 56
ภาพที่ 3.9 ขั้นตอนการสร้างตัวแบบ ANN..... 58
ภาพที่ 3.10 เปรียบเทียบค่าจริงและค่าพยากรณ์ตัวแบบ ANN..... 60
ภาพที่ 3.11 ขั้นตอนการสร้างตัวแบบ SVM 61
ภาพที่ 3.12 เปรียบเทียบค่าจริงและค่าพยากรณ์ตัวแบบ SVM 63
ภาพที่ 4.1 ค่าพยากรณ์ทั้ง 6 ตัวแบบ ของข้อมูล Sunspot..... 68
ภาพที่ 4.2 ค่าพยากรณ์ทั้ง 6 ตัวแบบ ของข้อมูล Canadian lynx..... 69
ภาพที่ 4.3 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล GBP/USD..... 70
ภาพที่ 4.4 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Electricuse 71
ภาพที่ 4.5 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Motorcycle 72
ภาพที่ 4.6 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล USexchange..... 73
ภาพที่ 4.7 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Gold 74
ภาพที่ 4.8 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Windspeed..... 75
ภาพที่ 4.9 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Meanrain 77
ภาพที่ 4.10 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Meanrain..... 78
ภาพที่ 4.11 ค่า MSE ที่รอบการทำซ้ำต่างๆทั้ง 50 รอบ..... 79
ภาพที่ 4.12 ค่า MSE ที่รอบการทำซ้ำต่างๆตั้งแต่รอบที่ 51 ถึงรอบที่ 250 79
ภาพที่ 4.13 ค่า MSE ที่จำนวน Hidden neurons ตั้งแต่ 1-50 80
ภาพที่ 4.14 ค่า MSE ที่พารามิเตอร์ C ต่างๆ 81
ภาพที่ 4.15 ค่า MSE ที่พารามิเตอร์ Epsilon ต่างๆ 83
ภาพที่ 4.16 ค่า MSE ที่พารามิเตอร์ Sigma ต่างๆ..... 84

ภาพที่ 5.1 แผนผังแต่ของเมนูใหญ่และเมนูย่อยในโปรแกรม Forecasting 86
ภาพที่ 5.2 แถบเมนูหน้าต่างทั้ง 5 เมนู โปรแกรม Forecasting 87
ภาพที่ 5.3 แถบเมนู Import และแถบเมนูย่อย Import text file และ Import excel file ของโปรแกรม Forecasting 87
ภาพที่ 5.4 แถบเมนู Modeling และแถบเมนูย่อยของโปรแกรม Forecasting 88
ภาพที่ 5.5 หน้าต่างแถบเมนูย่อย ARIMA 88
ภาพที่ 5.6 หน้าต่างแถบเมนูย่อย ANN 89
ภาพที่ 5.7 หน้าต่างแถบเมนูย่อย SVM 89
ภาพที่ 5.8 หน้าต่างแถบเมนูย่อย COMBINED 90
ภาพที่ 5.9 หน้าต่างโปรแกรม R console เมื่อเริ่มเปิดขึ้นใหม่ 90
ภาพที่ 5.10 หน้าต่างเลือกไฟล์ในตำแหน่งที่บันทึกไว้ 91
ภาพที่ 5.11 หน้าต่างเลือกไฟล์ในตำแหน่งที่บันทึกไว้ 91
ภาพที่ 5.12 หน้าต่างรายงานผลข้อมูลที่ Import 92
ภาพที่ 5.13 กราฟของข้อมูลภายหลังการ Import 92
ภาพที่ 5.14 หน้าต่างการแบ่งข้อมูลแบบกำหนดจุดแบ่งกลุ่มข้อมูล 93
ภาพที่ 5.15 หน้าต่างการแบ่งข้อมูลแบบกำหนดเปอร์เซ็นต์การแบ่ง 93
ภาพที่ 5.16 กราฟข้อมูลพร้อมเส้นแบ่งช่วงแยกกลุ่มข้อมูล 93
ภาพที่ 5.17 อันดับที่ดีที่สุดจากการค้นหาของตัวแบบ ARIMA 94
ภาพที่ 5.18 การทดสอบความเป็นอิสระของ Residuals 94
ภาพที่ 5.19 Histogram ของ Residuals 95
ภาพที่ 5.20 Autocorrelation ของ Residuals 95
ภาพที่ 5.21 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ARIMA 96
ภาพที่ 5.22 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ANN 98
ภาพที่ 5.23 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ SVM 99
ภาพที่ 5.24 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ ARIMAANN 101
ภาพที่ 5.25 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ARIMASVM 102
ภาพที่ 5.26 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ Combined (ARIMA+ANN+SVM) 103

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ความแม่นยำในการพยากรณ์ข้อมูลอนุกรมเวลาถือเป็น สิ่งที่สำคัญ และเป็นจุดเริ่มต้นในการจัดการงานต่างๆ การพยากรณ์ที่แม่นยำนำมาสู่การบริหารจัดการที่มีประสิทธิภาพมากยิ่งขึ้น การพัฒนาตัวแบบพยากรณ์ให้มีการพยากรณ์ที่แม่นยำมากขึ้นจึงถือเป็นงานท้าทาย มีงานวิจัยเกิดขึ้นมากมายทั้งในแง่การพัฒนาทฤษฎีเพื่อเพิ่มความแม่นยำในการพยากรณ์ทดสอบโดยใช้การจำลองและการทดลองนำตัวแบบไปใช้กับข้อมูลจริง ในอดีตที่ผ่านมามีงานวิจัยตัวแบบพยากรณ์อนุกรมเวลาเกิดขึ้นมากมายและส่วนใหญ่เป็นวิธีการทางสถิติ จุดเปลี่ยนที่สำคัญของงานวิจัยตัวแบบพยากรณ์อนุกรมเวลา คือ การนำเสนอวิธีการทางสถิติของ Box-Jenkins (G.E.P. Box and G. Jenkins, 1970) ที่สร้างตัวแบบโดยการพิจารณาค่าอัตตสัมพันธ์ในตัวเอง (Autocorrelation) และอัตตสัมพันธ์ในตัวเองบางส่วน (Partial autocorrelation) ก่อให้เกิดตัวแบบที่เรียกกันโดยทั่วไปว่า Autoregressive Integrated Moving Average (ARIMA) และยังผลให้เกิดตัวแบบอื่นๆที่ดัดแปลงจาก ARIMA ตามมาหลายตัวแบบ

ต่อเมื่อประมาณ 20 ปีก่อน ได้เกิดตัวแบบพยากรณ์ใหม่ คือ Artificial intelligent และ Machine learning ขึ้นมา เป็นลักษณะของการสร้างตัวแบบที่ไม่จำเป็นต้องมีสมมติฐานรองรับ ต่างจากตัวแบบทางสถิติแบบดั้งเดิม มีความยืดหยุ่นในการสร้างตัวแบบ (Zhang et al., 2001) หนึ่งในตัวแบบที่ได้รับความนิยม คือ Artificial neural network (ANN) ถูกนำมาใช้ในการสร้างตัวแบบพยากรณ์อนุกรมเวลา (Zhang et al., 2001, de Groot and Würtz, 1991, Beale et al., 2013) และผลตอบรับก็เป็นที่น่าพอใจมาก ต่อมาในปี 1995 ก็เกิดตัวแบบสำหรับการประมาณค่าขึ้นมาใหม่ คือ Support vector machine (SVM) (Vapnik V, 1995) เป็นตัวแบบที่มีประสิทธิภาพความแม่นยำในการพยากรณ์ที่สูง มีลักษณะคล้าย ANN คือตัวแบบจะเน้นจับลักษณะข้อมูลไม่เป็นเชิงเส้นตรงได้ดี

1.1.1 การเปรียบเทียบความแม่นยำของตัวแบบเดี่ยว

หลังจากนั้นก็เกิดงานวิจัยในเชิงเปรียบเทียบ ความแม่นยำของตัวแบบ ARIMA และ ตัวแบบทาง ANN (Taskaya-Temizel and Casey, 2005, de Groot and Würtz, 1991) หลายๆ เทคนิคกันมากขึ้นในหลายๆชุดข้อมูลรวมทั้งการจำลองและทดสอบกับข้อมูลจริงในหลายๆชุดข้อมูล หลายๆประเภทของงาน แต่ผลการศึกษากลับพบว่าไม่มีตัวแบบใดตัวแบบหนึ่งดีกว่ากันในทุกๆ ข้อมูล (Zhang, 2003, Taskaya-Temizel and Casey, 2005, Khashei and Bijari, 2011) ทำให้เกิดประเด็นที่ต้องทำการศึกษาต่อ และก็พบเหตุผลที่เป็นไปได้ว่าตัวแบบ ARIMA เป็นตัวแบบการพยากรณ์เชิงเส้นตรง (Linear model) ฟังก์ชันพยากรณ์จะเป็นผลรวมเชิงเส้นของข้อมูลในช่วงเวลาอดีต ตัวแบบนี้จับลักษณะข้อมูลที่เป็นเส้นตรง (Linear pattern) ได้ดี แต่อาจจะไม่จับกับลักษณะข้อมูลที่ไม่เป็นเส้นตรง (Nonlinear pattern) ได้ดีพอ (Zhang, 2003, Shi et al., 2012, Khashei

and Bijari, 2011) ส่วน ANN และ SVM นั้นเป็นตัวแบบที่ไม่เป็นเชิงเส้นตรง (Nonlinear model) จึงจับลักษณะที่ไม่เป็นเชิงเส้นตรงของข้อมูลได้ดีแต่จับลักษณะที่เป็นเชิงเส้นตรงของข้อมูลได้ไม่ดีนัก เมื่อเทียบกับ ARIMA (Zhang, 2003, Shi et al., 2012, Khashei and Bijari, 2011)

1.1.2 การเกิดขึ้นของตัวแบบผสม

เมื่อมีการศึกษาตัวแบบพยากรณ์ในหลายๆชุดข้อมูล ก็พบว่า ข้อมูลในความเป็นจริงนั้นมีความสลับซับซ้อนและมักมีส่วนผสมทั้งลักษณะข้อมูลเชิงเส้นตรงและไม่เป็นเชิงเส้นตรงอยู่ร่วมในข้อมูลชุดเดียวกันเสมอ (Zhang, 2003, Khashei and Bijari, 2011) จึงเกิดงานวิจัยที่ทดลองรวมสองตัวแบบเข้าด้วยกันเพื่อที่จะดึงข้อดีของแต่ละตัวแบบ ข้อดีของตัวแบบ Linear ในการจับ Linear pattern และ ข้อดีของตัวแบบ Nonlinear ในการจับ Nonlinear pattern ด้วยหวังว่าจะทำการพยากรณ์มีความแม่นยำมากยิ่งขึ้น เรียกว่าตัวแบบผสม (Hybrid model)

ตัวแบบผสม (Hybrid model) ระหว่างตัวแบบ Linear กับตัวแบบ Nonlinear สามารถแบ่งออกได้เป็น 2 ประเภท (Shi et al., 2012) คือ

1. Traditional hybrid คือ ตัวแบบผสมที่มีกระบวนการทำต่อเนื่องกัน เริ่มจากตัวแบบ Linear เช่นตัวแบบ ARIMA ทำการสร้างตัวแบบแล้วพยากรณ์ จะได้ค่าเศษเหลือจากการพยากรณ์ (Residuals) จากนั้นนำค่าเศษเหลือตัวนี้ทำการพยากรณ์ต่อด้วยตัวแบบ Nonlinear เช่น ANN, SVM จึงจะได้ค่าพยากรณ์รวมที่นำไปใช้ เป็นต้น

2. Hybrid combined เป็นตัวแบบที่ทำการพยากรณ์แยกส่วนกัน คือ ทำการพยากรณ์ด้วยตัวแบบ Linear เช่นตัวแบบ ARIMA และ ตัวแบบ Nonlinear เช่น ตัวแบบ ANN และ SVM แยกจากกัน สามารถทำพร้อมๆกันบนเครื่องคอมพิวเตอร์เดียวกันได้ เมื่อสร้างตัวแบบเสร็จก็นำค่าพยากรณ์จากทั้ง 3 ตัวแบบมารวมกัน ในรูปแบบผลรวมเชิงเส้น แล้วจึงหาค่าน้ำหนัก (Weights) ที่ดีที่สุดแล้วที่ทำให้ค่าพยากรณ์รวมมีความแม่นยำมากที่สุด

ตัวแบบ Traditional hybrid ถือเป็นตัวแบบที่ได้รับความนิยม และนำมาใช้ในการพยากรณ์ข้อมูลในหลายๆ แหล่งข้อมูล ในหลายการประยุกต์ใช้งาน ซึ่งให้ผลการพยากรณ์ที่แม่นยำกว่าตัวแบบเดี่ยว ดังเช่น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.1.3 การเปรียบเทียบความแม่นยำของตัวแบบผสมเทียบกับตัวแบบเดี่ยว

1. เปรียบเทียบความแม่นยำระหว่างตัวแบบ Traditional hybrid และ ตัวแบบเดี่ยว

Traditional hybrid ดีกว่า ตัวแบบเดี่ยว	Traditional hybrid ไม่ดีกว่า ตัวแบบเดี่ยว
Zhang (Zhang, 2003) นำเสนอตัวแบบ hybrid ARIMA ANN ทำการทดลองกับข้อมูลอนุกรมเวลาที่เป็นที่นิยม ได้แก่ Sunspot data sets, Canadian lynx และ British pound/US dollar exchange rate ผลการทดลองพบว่าตัวแบบ hybrid ให้ผลความแม่นยำที่ดีกว่า ตัวแบบเดี่ยวๆ ทั้งสอง	Taskaya-Temizel และ Casey (Taskaya-Temizel and Casey, 2005) ได้ทำการทดลองใช้ Linear AR TDNN พยากรณ์ข้อมูลยอดขายสินค้าแต่ละประเภท 9 ชุดข้อมูล ที่มีปัจจัยเชิงฤดูกาล สรุปว่าตัวแบบ hybrid ไม่ได้ทำให้ความแม่นยำของการพยากรณ์สูงกว่าตัวแบบเดี่ยวๆ เสมอไป
Hansen และ Nelson (James V. Hansen and Ray D. Nelson, 2010) ทำการศึกษาเปรียบเทียบตัวแบบ hybrid ARIMAANN ได้ผลสรุปว่าตัวแบบ hybrid ARIMAANN นี้มีศักยภาพทำให้ได้ค่าพยากรณ์ที่มีความแม่นยำมากกว่าตัวแบบเดี่ยว	Shi, Jing และคณะ (Shi, Jing;Guo, Jinmei;Zheng, Songtao, 2012) ได้พยากรณ์ข้อมูลความเร็วลม โดยใช้ ARIMAANN และ ARIMASVM
Lin (Pai and Lin, 2005) ทำการศึกษาเปรียบเทียบตัวแบบ hybrid ARIMASVM ในการพยากรณ์ราคาหุ้นผลการทดลองพบว่า ตัวแบบ hybrid นี้ให้ค่าError ของการพยากรณ์ต่ำกว่าตัวแบบเดี่ยว	
Gutierrez-Estrada และคณะ (Gutiérrez-Estrada et al., 2007) นำเสนอตัวแบบ hybrid ARIMAANN เพื่อพยากรณ์ปริมาณปลาแอนโชวี่ที่จับได้ผลการทดลองพบว่าตัวแบบ hybrid ให้ผลการพยากรณ์ที่แม่นยำกว่าตัวแบบเดี่ยว	

ตารางที่ 1.1 เปรียบเทียบความแม่นยำระหว่างตัวแบบ Hybrid combined และ ตัวแบบเดี่ยว

1.1.4 มูลเหตุของงานวิจัย

ประเด็นในการศึกษาในงานวิจัยชิ้นนี้ คือ

1. เนื่องจากมีงานวิจัยที่ระบุว่าตัวแบบผสมที่ประกอบด้วยส่วนที่เป็นเชิงเส้นตรงและส่วนที่ไม่เป็นเชิงเส้นตรง มีความแม่นยำมากกว่าตัวแบบเดี่ยวๆ ขณะเดียวกันมีงานวิจัยที่ค้านประสิทธิภาพของตัวแบบผสม สำหรับบางข้อมูลก็ไม่ได้ดีไปกว่าตัวแบบเดี่ยวที่ดีที่สุดที่นำมาผสมกัน งานวิจัยชิ้นนี้จึงทำ

การทดลองเปรียบเทียบความแม่นยำของตัวแบบผสมเทียบกับตัวแบบเดี่ยว ในข้อมูลหลากหลายชุดในหลากหลายลักษณะข้อมูล โดยตัวแบบเดี่ยวที่นำมาทดลอง คือ ARIMA, ANN และ SVM ตัวแบบทั้งสามถือเป็นตัวแบบที่ได้รับความนิยม และมีประสิทธิภาพสูงในการพยากรณ์ ตัวแบบ ARIMA จะทำการพยากรณ์ข้อมูลส่วนที่เป็นเชิงเส้นตรง ส่วนตัวแบบ ANN และ SVM พยากรณ์ส่วนที่ไม่เป็นเชิงเส้นตรง จะได้ตัวแบบผสม คือ ARIMAANN และ ARIMASVM

ตัวแบบ ARIMAANN นำมาเปรียบเทียบความแม่นยำกับตัวแบบ ARIMA และ ANN

ตัวแบบ ARIMASVM นำมาเปรียบเทียบความแม่นยำกับตัวแบบ ARIMA และ SVM

2. ทำการทดลองนำตัวแบบ ARIMA, ANN และ SVM มารวมกัน ในรูปแบบ Hybrid combined หรือเขียนย่อๆว่า Combined (ARIMA+ANN+SVM) เนื่องจากรูปแบบตัวแบบ Hybrid combined สามารถนำตัวแบบหลายๆ ตัวแบบเข้ามารวมกันได้ ผิดกับตัวแบบ Traditional hybrid ที่เมื่อมีตัวแบบมากกว่า 2 ตัวจะเป็นการยุ่งยาก โดยที่ไม่สามารถการันตีได้ว่าจะได้ค่าพยากรณ์ที่มีความแม่นยำกว่าเดิมสำหรับทุกๆข้อมูล นอกจากนี้ยังเป็นการรวมกันระหว่างตัวแบบ ARIMA ซึ่งเป็นตัวแบบ Linear ที่โดดเด่น เป็นที่นิยม และ ANN กับ SVM ซึ่งเป็นตัวแบบ Nonlinear ที่มีประสิทธิภาพสูง เป็นที่นิยม มีกระบวนการสร้างตัวแบบที่แตกต่างกัน จึงทำการทดลองนี้เพื่อหวังว่าจะเป็นตัวแบบทางเลือกที่เพิ่มความแม่นยำในการพยากรณ์และสามารถนำไปใช้ได้จริง

การเปรียบเทียบความแม่นยำของตัวแบบ Combined (ARIMA+ANN+SVM) เปรียบเทียบในสองนัยยะ คือ เทียบกับกับตัวแบบผสม ARIMAANN และ ARIMASVM และเทียบกับตัวแบบเดี่ยว ARIMA, ANN และ SVM

1.2 วัตถุประสงค์งานวิจัย

1. พยากรณ์ข้อมูลอนุกรมเวลามาตรฐานและข้อมูลอนุกรมเวลาที่มีความซับซ้อนในประเทศไทยด้วยตัวแบบผสม (Hybrid model) เทียบผลความแม่นยำกับตัวแบบเดี่ยว (Individual model)
2. พัฒนาโปรแกรมประยุกต์พยากรณ์ข้อมูลอนุกรมเวลาโดยใช้ตัวแบบผสม ในลักษณะ Graphical User Interface (GUI)

1.3 ขอบเขตของงานวิจัย

ทำการพยากรณ์ข้อมูลอนุกรมเวลา 2 ชนิดแบ่งออกเป็น

ก. ข้อมูลอนุกรมเวลามาตรฐาน

- The Wolf’s sunspot data
- The Canadian lynx data
- The British pound/US dollar exchange rate data

ข้อมูลมาตรฐานทั้งสามเป็นข้อมูลที่เป็นที่รู้จักดีในหมู่นักพยากรณ์ มีการนำมาใช้ในงานวิจัยเปรียบเทียบประสิทธิภาพตัวแบบอยู่บ่อยครั้ง (Zhang, 2003, Taskaya-Temizel and Casey, 2005, Khashei and Bijari, 2011)

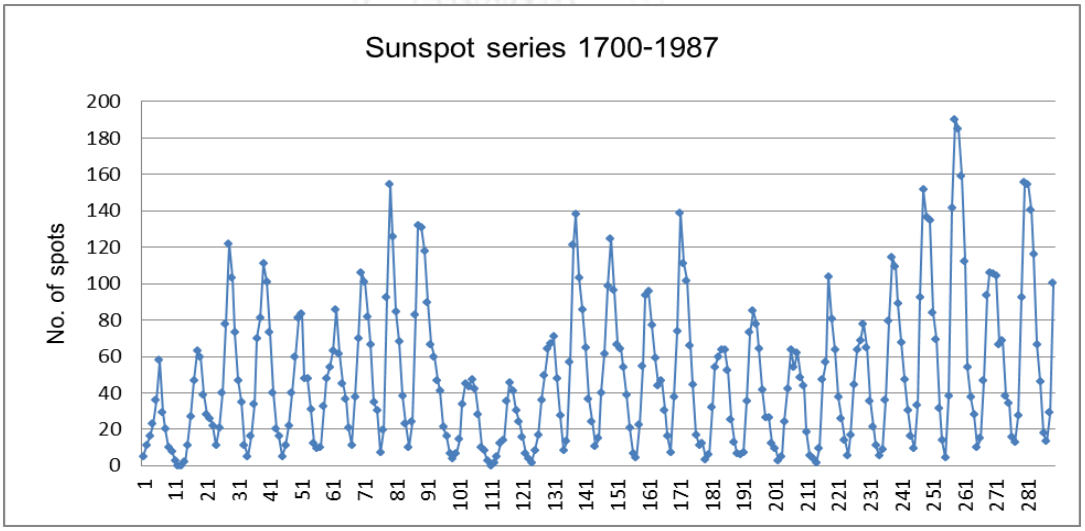
ข. ข้อมูลอนุกรมเวลาจริงในประเทศไทย โดยเลือกข้อมูลจากหลายๆ แหล่งข้อมูล ดังต่อไปนี้

- ข้อมูลปริมาณการใช้พลังงานไฟฟ้ารายเดือนทั้งประทศไทย (หน่วย:ล้านหน่วย)
- ข้อมูลยอดขายรถจักรยานยนต์รายเดือนทั้งประทศไทย (หน่วย:คัน)
- ข้อมูลอัตราแลกเปลี่ยนรายวันของเงินสกุลดอลลาร์สหรัฐและเงินบาท (หน่วย: บาท/1 US dollar)
- ข้อมูลรายวันราคาทองคำหนักหนึ่งบาท (หน่วย:บาท/ทองคำหนัก 1 บาท)
- ข้อมูลรายวันความเร็วลมโดยเฉลี่ยในรอบวัน (นอต)
- ข้อมูลปริมาณน้ำฝนตามโดยเฉลี่ยรายเดือน (มิลลิเมตร)
- ข้อมูลระดับอุณหภูมิโดยเฉลี่ยรายเดือน (เซลเซียส)

ค. ศึกษาและเปรียบเทียบความแม่นยำของการพยากรณ์ สำหรับ 1 ค่าล่วงหน้า (One step ahead forecasting)

1.4 ข้อมูลอนุกรมเวลาที่นำมาใช้พยากรณ์

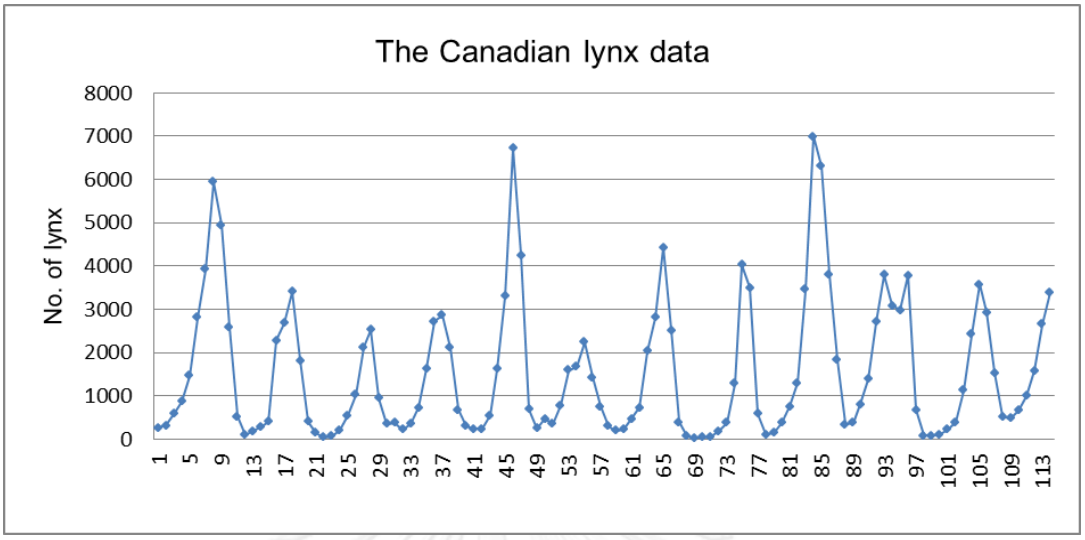
1.4.1 The Wolf's sunspot dataจำนวนข้อมูล 288 ตัวอย่าง



ภาพที่ 1.1 จำนวนจุดดับดวงอาทิตย์ที่เกิดขึ้นใหม่รายปี ตั้งแต่ค.ศ. 1700 ถึงค.ศ. 1987 (Zhang, 2003)

เป็นข้อมูลจุดดับบนดวงอาทิตย์ในแต่ละปี ข้อมูลมีความแปรปรวนไม่คงที่ วงรอบของข้อมูลอยู่ที่ 10-13 ปี

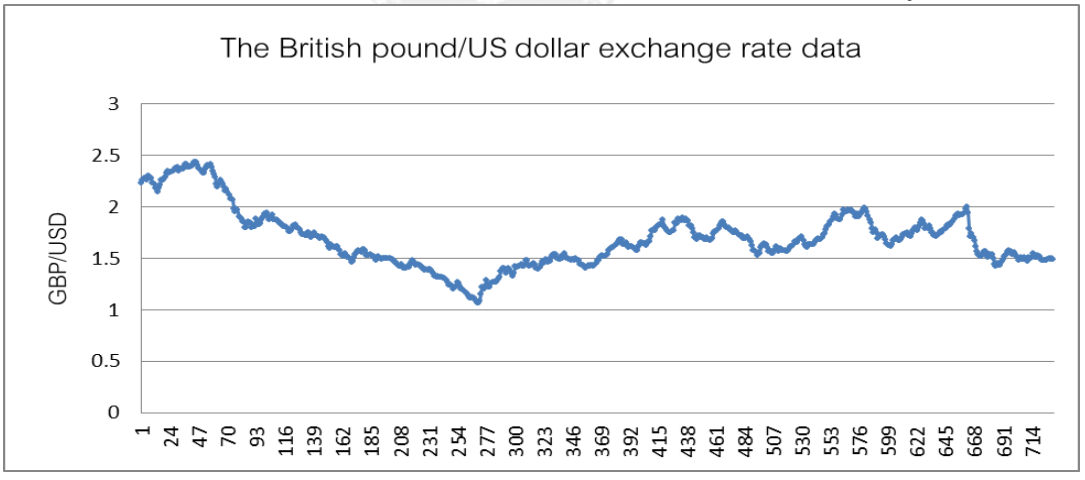
1.4.2 The Canadian lynx dataจำนวนข้อมูล 114 ตัวอย่าง



ภาพที่ 1.2 จำนวนแมวน้ำที่ถูกจับที่แม่น้ำ Mackensize ในประเทศแคนาดารายปี ตั้งแต่ค.ศ. 1821 ถึงค.ศ. 1934 (Zhang, 2003)

เป็นข้อมูลจำนวนแมวน้ำที่ถูกจับได้ที่แม่น้ำ Mackensize ประเทศแคนาดารายปี ข้อมูลจะเพิ่มขึ้นและลดลงอย่างรวดเร็วแบบ Exponential

1.4.3 The British pound/US dollar exchange rate data จำนวนข้อมูล 731 ตัวอย่าง



ภาพที่ 1.3 อัตราแลกเปลี่ยนรายวันเงินสกุล British pound ต่อ US dollar ตั้งแต่ค.ศ. 1821 ถึงค.ศ. 1934 (Zhang, 2003)

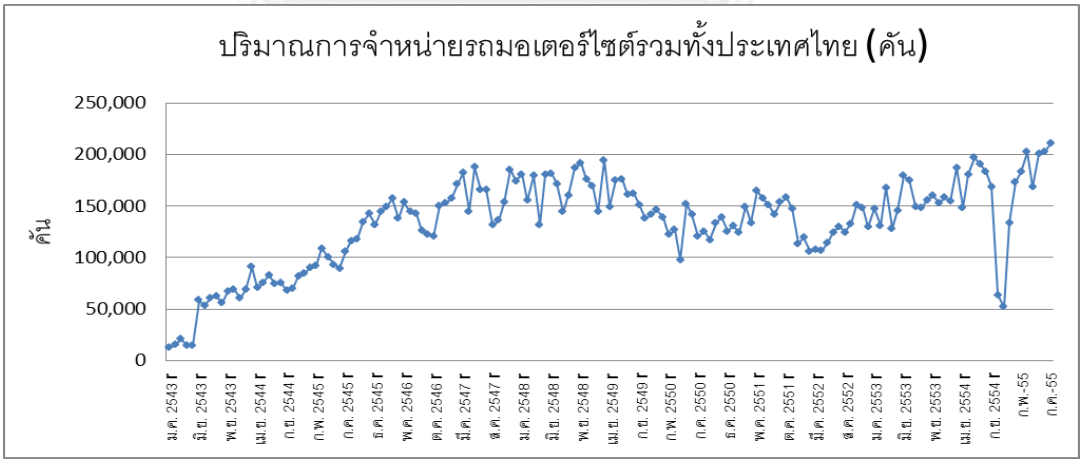
เป็นอัตราแลกเปลี่ยนเงินสกุลปอนด์และเงินดอลลาร์สหรัฐรายปี ข้อมูลมีการขึ้นลงแบบสุ่มในระยะสั้น ประกอบกับมีแนวโน้มทั้งเพิ่มขึ้นและลดลง

1.4.4 ข้อมูลปริมาณการใช้พลังงานไฟฟ้ารายเดือนทั้งประเทศไทย (หน่วย:ล้านหน่วย) จำนวนข้อมูล 151 ตัวอย่าง

ภาพที่ 1.4 ปริมาณการใช้พลังงานไฟฟ้ารวม (ล้านหน่วย) ทั้งประเทศไทย ตั้งแต่เดือนมกราคม 2543 ถึง กรกฎาคม 2555 (ธนาคารแห่งประเทศไทย, 2555a)

เป็นข้อมูลรายเดือนที่รวมปริมาณการใช้ไฟฟ้าทั้งประเทศ ข้อมูลมีปัจจัยเชิงฤดูกาลที่ชัดเจน และมีแนวโน้มที่เพิ่มขึ้น

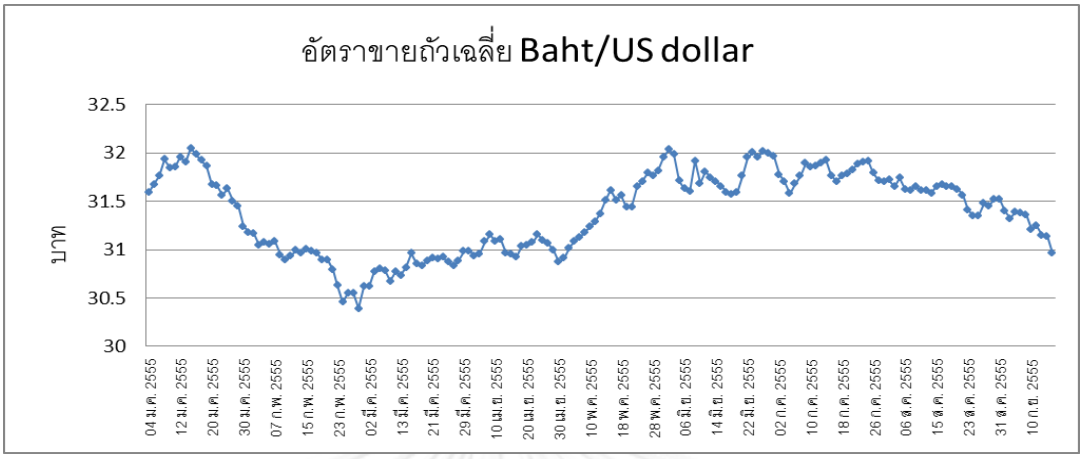
1.4.5 ข้อมูลยอดขายรถจักรยานยนต์รายเดือนทั้งประเทศไทย (หน่วย:คัน) จำนวนข้อมูล 151 ตัวอย่าง



ภาพที่ 1.5 ปริมาณการจำหน่ายรถมอเตอร์ไซต์ (คัน) ตั้งแต่เดือนมกราคม 2543 ถึง กรกฎาคม 2555 (ธนาคารแห่งประเทศไทย, 2555a)

เป็นข้อมูลยอดขายสินค้ารายเดือน ข้อมูลมีแนวโน้มเพิ่มขึ้นในช่วงแรก และเข้าสู่แนวโน้มที่มีค่าเฉลี่ยคงที่ และมีค่าต่ำมาก (Random shock) 2 ค่า

1.4.6 ข้อมูลอัตราแลกเปลี่ยนรายวันของเงินสกุลดอลลาร์สหรัฐและเงินบาท (หน่วย:บาท/1 US dollar) จำนวนข้อมูล 173 ตัวอย่าง



ภาพที่ 1.6 ราคาขายถัวเฉลี่ย US Dollar/Baht ตั้งแต่วันที่ 1 มกราคม 2543 ถึง 14 กันยายน 2555 (ธนาคารแห่งประเทศไทย, 2555b)

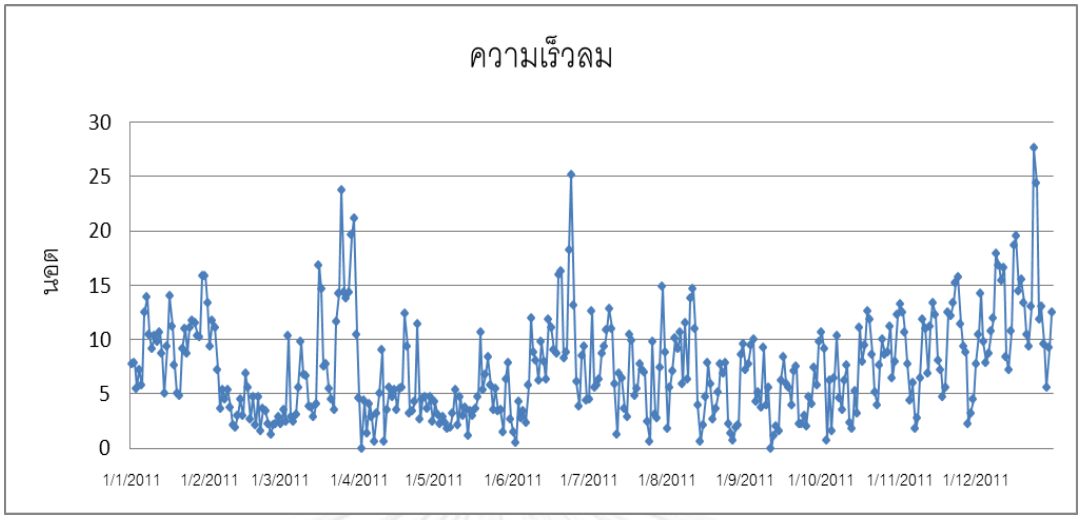
เป็นข้อมูลทางการเงินรายวัน ข้อมูลเป็นแบบสุ่ม มีแนวโน้มเพิ่มขึ้นและลดลงเป็นช่วงๆ

1.4.7 ข้อมูลรายวันราคาทองคำหนักหนึ่งบาท (หน่วย:บาท/ทองคำหนัก 1 บาท) จำนวนข้อมูล 220 ตัวอย่าง

ภาพที่ 1.7 ราคารับซื้อทองคำหนึ่งบาทรายวัน ตั้งแต่วันที่ 1 มกราคม 2543 ถึง 14 กันยายน 2555 (ห้างขายทองทองใบเยาวราช, 2555)

เป็นข้อมูลทางการเงินรายวัน ข้อมูลเป็นแบบสุ่ม มีแนวโน้มเพิ่มขึ้นและลดลงเป็นช่วงๆ

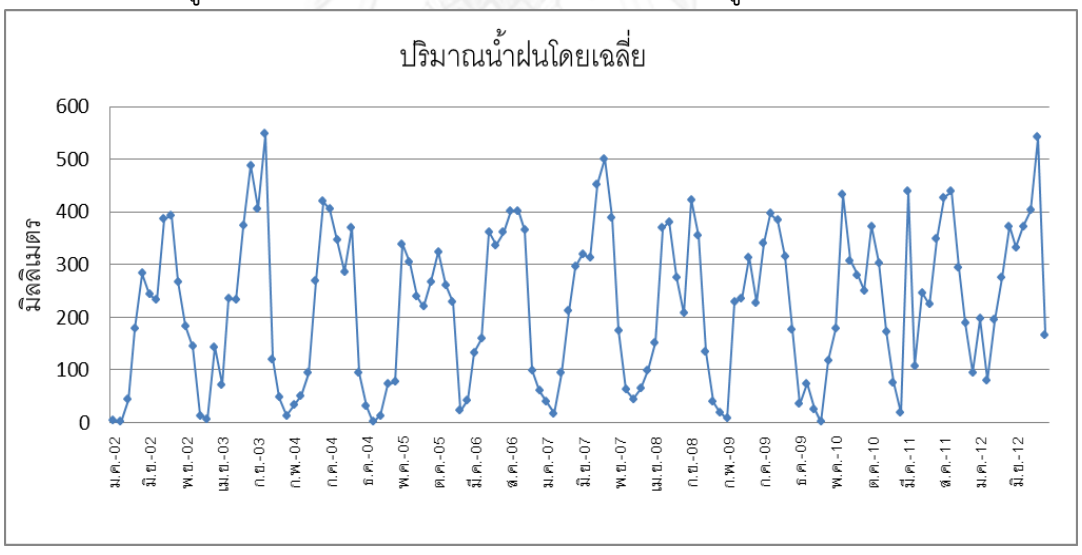
1.4.8 ข้อมูลรายวันความเร็วลมโดยเฉลี่ยในรอบวัน (นอต) จำนวนข้อมูล 365 ตัวอย่าง



ภาพที่ 1.8 ความเร็วลมโดยเฉลี่ยในรอบวัน (นอต) ตั้งแต่วันที่ 1 มกราคม ถึง 31 ธันวาคม 2554 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)

เป็นข้อมูลรายวัน ข้อมูลเป็นแบบสุ่ม มีความซับซ้อนวุ่นวาย (Chaos series)

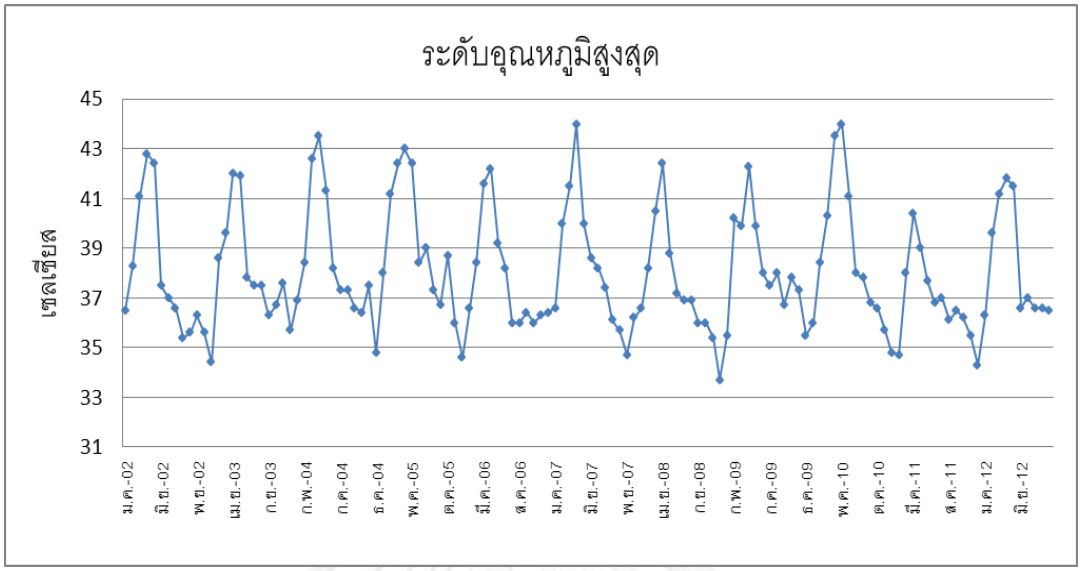
1.4.9 ข้อมูลปริมาณน้ำฝนรายเดือน (มิลลิเมตร) จำนวนข้อมูล 130 ตัวอย่าง



ภาพที่ 1.9 ปริมาณน้ำฝนภาคใต้ฝั่งตะวันตกตั้งแต่ มกราคม 2545 ถึงตุลาคม 2555 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)

เป็นข้อมูลทางอุตุนิยมวิทยารายเดือน ข้อมูลมีปัจจัยฤดูกาลรวมกับความแปรปรวน ทำให้ตำแหน่งสูงสุดและต่ำสุดมีความคลาดเคลื่อน

1.4.10 ข้อมูลระดับอุณหภูมิโดยเฉลี่ยรายเดือน (เซลเซียส) จำนวนข้อมูล 130 ตัวอย่าง



ภาพที่ 1.10 ระดับอุณหภูมิสูงสุดภาคเหนือ ตั้งแต่เดือนมกราคมคม 2545 ถึงเดือนตุลาคม 2555 (สำนักพัฒนาอุตุนิยมวิทยา, 2555)

เป็นข้อมูลทางอุตุนิยมวิทยารายเดือน ข้อมูลมีปัจจัยฤดูกาลรวมกับความแปรปรวน ทำให้ตำแหน่งสูงสุดและต่ำสุดมีความคลาดเคลื่อน

1.5 ขั้นตอนการวิจัยและดำเนินงาน

1. ศึกษาเกี่ยวกับการพยากรณ์อนุกรมเวลา สำหรับตัวแปรเดียว วิธีการใดบ้างที่ใช้พยากรณ์ ตัวแบบแต่ละตัวแบบมีจุดเด่นจุดด้อยอย่างไร

2. ศึกษาวิธีการพยากรณ์ข้อมูลอนุกรมเวลาด้วยตัวแบบผสม และตัดสินใจเลือกตัวแบบผสมที่นำมาใช้ทดลองในงานวิจัย ได้แก่ ARIMAANN, ARIMASVM และ Combined (ARIMA+ANN+SVM)

3. เลือกชุดข้อมูลเพื่อนำมาสร้างตัวแบบในการพยากรณ์

4. เปรียบเทียบความแม่นยำของการพยากรณ์โดยใช้ค่าวัดทางสถิติ ได้แก่ RMSE ,MAE, MSE, MAPE เทียบความแม่นยำของตัวแบบผสมและตัวแบบเดี่ยว ดังนี้

- Combined (ARIMA+ANN+SVM)
- ARIMAANN
- ARIMASVM
- ARIMA
- ANN
- SVM

5. สรุปผลงานวิจัยและจัดทำรูปเล่มวิทยานิพนธ์

1.6 ผลที่คาดว่าจะได้รับ

1. ทราบถึงความแม่นยำของตัวแบบผสม ARIMAANN และ ARIMASVM ที่ใช้พยากรณ์สำหรับข้อมูลแต่ละประเภท เทียบความแม่นยำกับตัวแบบเดี่ยว

2. ทราบถึงความแม่นยำของตัวแบบผสม Combined (ARIMA+ANN+SVM) ที่นำเสนอ เทียบกับตัวแบบผสม ARIMAANN และ ARIMASVM

3. สามารถนำตัวแบบที่ให้ความแม่นยำของการพยากรณ์ที่สูงที่สุดในแต่ละชุดข้อมูลนำไปพยากรณ์กับข้อมูลจริงในปัจจุบัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ตัวแบบ Autoregressive Integrated Moving Average (ARIMA)

ตัวแบบ ARIMA เป็นตัวแบบพยากรณ์อนุกรมเวลาทางสถิติยอดนิยม ใช้กันอย่างแพร่หลาย ฟังก์ชันพยากรณ์จะเป็นผลรวมเชิงเส้นตรงของข้อมูลในอดีต (Past observations) และค่าสัมประสิทธิ์ (Coefficient) บวกกับค่าผิดพลาดเชิงสุ่ม (Random errors) เขียนเป็นฟังก์ชันทางคณิตศาสตร์ได้ ดังนี้

$$y_t = \theta_0 + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - -\theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (2.1)$$

โดยที่

- y_t คือ ค่าจริง ณ เวลา t
- ε_t คือค่าผิดพลาดเชิงสุ่ม ณ เวลา t ซึ่งจะมีการแจกแจงเหมือนกันและเป็นอิสระต่อกัน (Independently and identically distributed) ด้วยค่าเฉลี่ย (Mean) เป็น 0 และค่าความแปรปรวน (Variance) คือ σ^2
- $\emptyset_i$ คือค่าสัมประสิทธิ์ของตัวแบบ โดยที่ $i = 0,1,2, \ldots, p$
- θ_j คือค่าสัมประสิทธิ์ของตัวแบบ โดยที่ $j = 0,1,2, \ldots, q$
- p และ q คือ อันดับของตัวแบบ

เป็นตัวแบบพยากรณ์ข้อมูลอนุกรมเวลาทางสถิติ ที่ประกอบด้วยส่วนประกอบหลักสองส่วน คือ AR (Autoregressive)

$$Y_t = \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \cdots + \emptyset_p Y_{t-p} + \varepsilon_t \quad (2.2)$$

MA (Moving Average)

$$Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (2.3)$$

และส่วน I ย่อมาจาก Integrated เป็นชื่อที่บอกให้รู้ว่ามีการแปลงข้อมูลแบบ Difference แต่ไม่มีเทอมปรากฏในสมการ

2.1.1 ค่าอัตตสัมพันธ์ (Autocorrelation) และค่าอัตตสัมพันธ์บางส่วน (Partial autocorrelation)

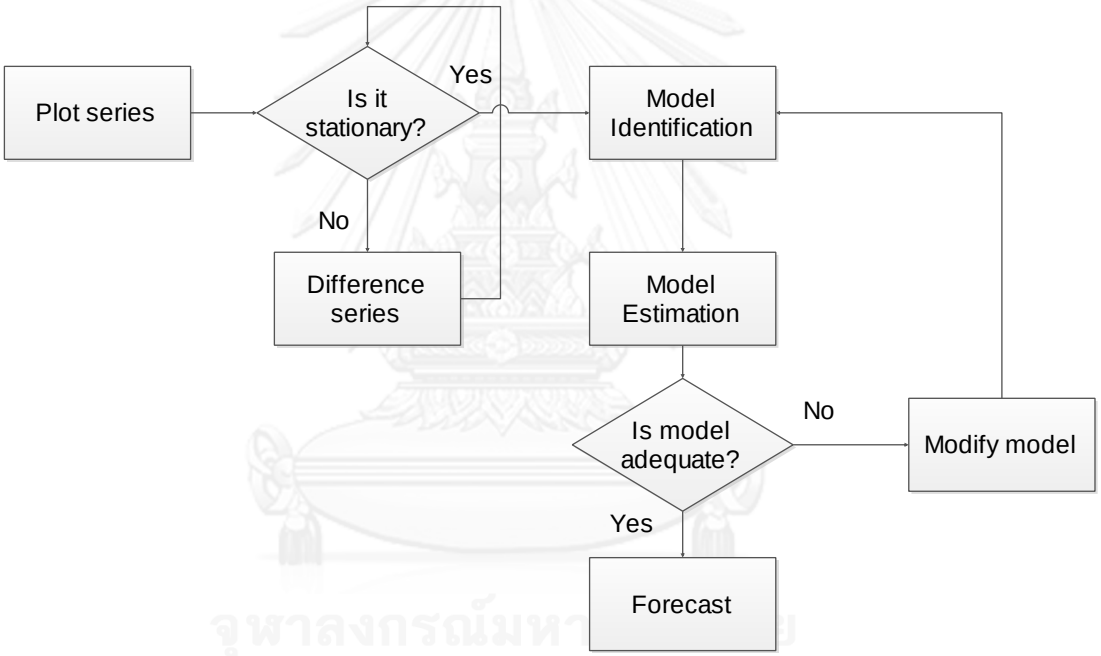
Autocorrelation (ACF) คือ ค่าความสัมพันธ์ของข้อมูลในชุดเดียวกัน หลักการคล้ายกับค่าสหสัมพันธ์ (Correlation) ในตัวแบบวิเคราะห์อื่นๆ เช่น Regression จุดต่างอยู่ที่ Autocorrelation

จะหาจากข้อมูลตัวแปรเดียว แต่ Correlation จะหาจากข้อมูล 2 ตัวแปรค่า Autocorrelation ณ จุดใกล้ๆ กันจะมีค่ามาก และมักจะค่อยๆลดลงเมื่อจุดนั้นห่างออกไป เช่น Y_{11} จะสัมพันธ์กับ Y_{10} มากแต่อาจสัมพันธ์กับ Y_4 น้อยกว่า แต่จะมีความสัมพันธ์ซ้อนกัน คือ Y_{10} จะสัมพันธ์กับ Y_9 มากด้วย แต่ทั้งนี้ทั้งนั้นก็ต้องดูจากกราฟแสดงข้อมูลอนุกรมเวลาด้วยเนื่องจากอาจมีปัจจัยเชิงฤดูกาล (Seasonal) หรือ วัฏจักร (Cycle) ที่ส่งผลกับค่า Autocorrelation ระหว่างจุดไกลๆ ที่มีค่ามากหรือน้อยในลักษณะเดียวกันในรอบปี เป็นต้น

Partial Autocorrelation (PACF) คือค่า Correlation ระหว่าง Lag ที่ไม่สามารถอธิบายได้ด้วย Autocorrelation สำหรับ Lag อันดับต่ำๆ

2.1.2 ขั้นตอนการสร้างตัวแบบ ARIMA

สามารถแสดงได้เป็นขั้นตอนดังต่อไปนี้



ภาพที่ 2.1 แสดงแผนผังขั้นตอนการสร้างตัวแบบ ARIMA

1. พลอตกราฟข้อมูล เพื่อตรวจสอบลักษณะข้อมูล เช่น มีแนวโน้มหรือไม่ มีปัจจัยที่เป็นองค์ประกอบฤดูกาลหรือไม่ ข้อมูลมีลักษณะแกว่งไปมามากน้อยแค่ไหน

2. พิจารณาคุณสมบัติความเป็น Stationary ของข้อมูล

Stationary คือ คุณสมบัติความนิ่งของข้อมูลค่าเฉลี่ย และค่าความแปรปรวนมีค่าคงที่ไม่เปลี่ยนแปลงตามเวลาไม่ว่าที่เวลาใดๆ สามารถแบ่งออกได้เป็น

- Stationary ในค่าเฉลี่ย คือ ค่าเฉลี่ยมีค่าคงที่ทุกช่วงเวลา
- Stationary ในความแปรปรวน คือ ความแปรปรวน หรือ การกระจายตัวของข้อมูลคงที่ มีการกระจายตัวที่สม่ำเสมอ ในทุกช่วงเวลา

เงื่อนไขที่สำคัญของตัวแบบ ARIMA คือ คุณสมบัติความเป็น Stationary ของข้อมูล ข้อมูลที่จะนำมาสร้างตัวแบบนั้นต้องมีคุณสมบัตินี้ จึงจำเป็นต้องตรวจสอบก่อน

2.1) การตรวจสอบ Stationary ของข้อมูล

- พิจารณาจากกราฟ ACF และ PACF
- พิจารณาจากการทดสอบทางสถิติ ตัวสถิติที่ใช้ทดสอบ คือ Augmented Dickey and Fuller (ADF) test สมมติฐานในการทดสอบ คือ

Ho: ข้อมูลไม่เป็น Stationary

H1: ข้อมูลเป็น Stationary

เกณฑ์การปฏิเสธ Ho คือ คือ ค่าระดับนัยสำคัญในการทดสอบ α> p-value ที่ได้จากการทดสอบ

ในกรณีที่ผลการทดสอบยอมรับ Ho นั่นคือ ข้อมูลไม่มีคุณสมบัติ Stationary จำเป็นต้องทำแปลงข้อมูลให้เข้าคุณสมบัติ Stationary เช่นการ Differencing

2.2) การ Difference คือ นำข้อมูลตัวที่ติดกันลบกัน เป็นคู่ๆ เหมาะสำหรับข้อมูลที่มีแนวโน้ม (Trend)

ให้ Y_t คือ ข้อมูลอนุกรมเวลาที่เวลา t ดังนั้น $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots, Y_1$ คือ ข้อมูลอนุกรมเวลาที่เวลา t-1, t-2, t-3...,1 ตามลำดับ

$$Y_t - Y_{t-1}$$

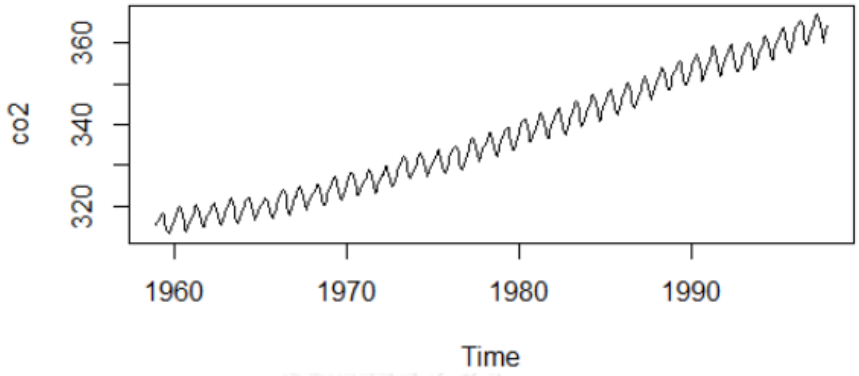
$$Y_{t-1} - Y_{t-2}$$

$$Y_{t-2} - Y_{t-3}$$

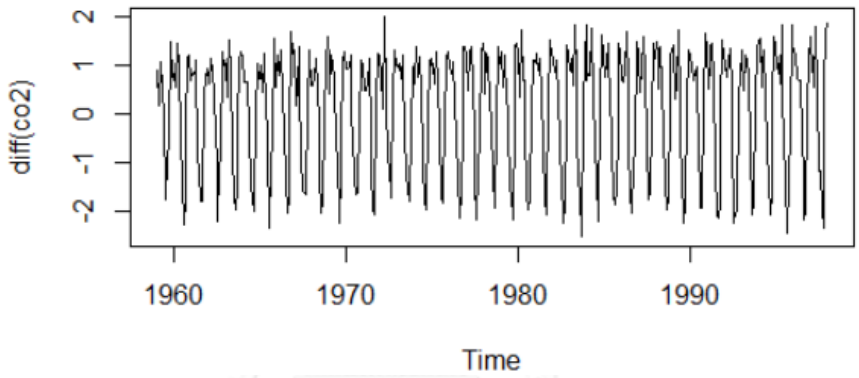
.....

$$Y_2 - Y_1$$

ภายหลังการ Difference ครั้งแรก ต้องนำข้อมูลชุดใหม่นั้นเข้าทดสอบสมมติฐานอีกครั้งหากยังทดสอบไม่ผ่าน ให้ทำการ Difference อีกครั้ง แล้วนำมาทดสอบสมมติฐาน โดยปกติแล้วมักจะทำการ Difference ไม่เกิน 2 ครั้ง ข้อมูลก็จะมีคุณสมบัติ Stationary ตัวอย่างการแปลงดังภาพ



ภาพที่ 2.2 ตัวอย่างข้อมูลที่ไม่มีคุณสมบัติ Stationary



ภาพที่ 2.3 ตัวอย่างข้อมูลภายหลังทำการ Difference

ภายหลังทำการ Difference ดังภาพ 2.3 กราฟของข้อมูลจากที่มีแนวโน้มขึ้น เปลี่ยนเป็นกราฟที่เป็นแนวเส้นตรง และความแปรปรวนก็ค่อนข้างคงที่ นอกจากนี้อาจทดสอบสมมติฐานคุณสมบัติ Stationary เพื่อตรวจสอบอีกครั้ง

3. Model Identification

คือ การระบุอันดับตัวแบบ ที่เป็นไปได้ อันดับค่าเหล่านั้น คือ อันดับ AR , MA และจำนวนครั้งการทำ Difference เขียนเป็นสัญลักษณ์ คือ ARIMA(p,d,q) เป็นตัวแบบมาตรฐานที่ครอบคลุม ตัวแบบ AR ตัวแบบ MA และตัวแบบ ARMA ที่ใช้กับข้อมูลที่มีความเป็น Stationary อยู่แล้ว

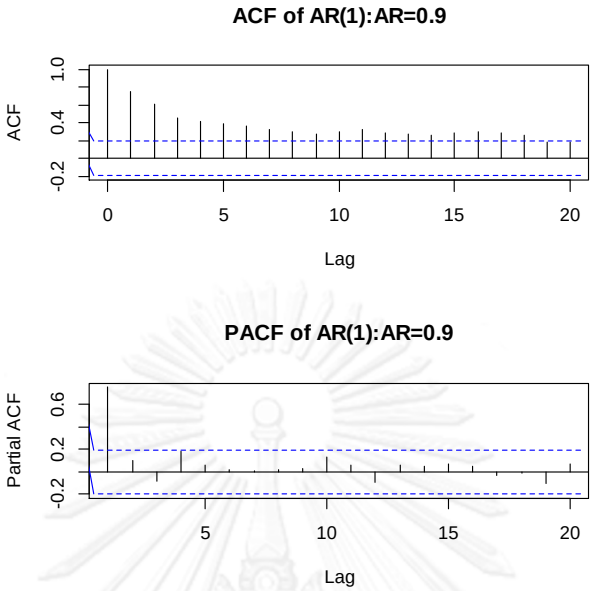
ARIMA(1,0,0) เทียบได้กับ AR(1)

ARIMA(0,0,1) เทียบได้กับ MA(1)

ARIMA(1,0,1) เทียบได้กับ ARMA(1,1)

ในขั้นตอนนี้จะทำการหาค่าอันดับ p,dและ q ที่เป็นไปได้ โดยพิจารณาเบื้องต้นจากกราฟ ACF และ PACF

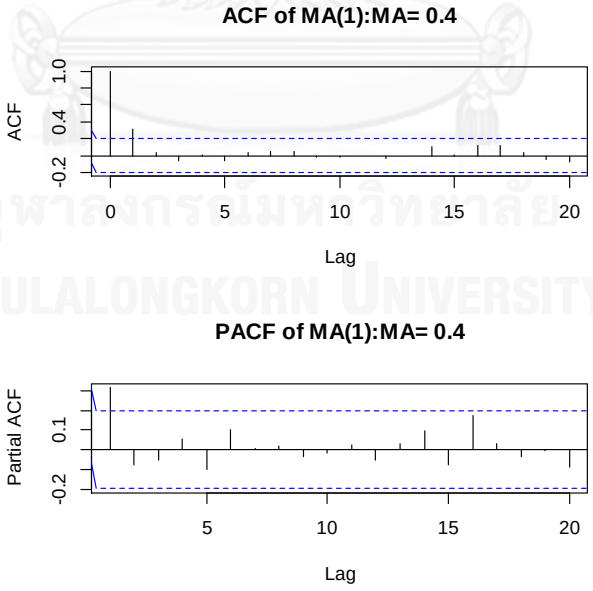
ACF เป็นกราฟที่แสดงค่า Correlation ระหว่างข้อมูลเป็นคู่ๆ ณ จุดเวลาที่ต่างกัน หรือเรียกว่า Lag ตัวอย่างตรวจสอบ ตัวแบบ ARIMA(1,0,0) หรือ AR(1)



ภาพที่ 2.4 กราฟ ACF และ PACF สำหรับตัวแบบ AR(1)

กราฟ ค่า ACF จะค่อยๆ ลดลงอย่างช้าๆ ขณะที่กราฟ PACF จะลดลงอย่างรวดเร็ว Lag ที่ค่า PACF ลดลงอย่างรวดเร็วจะแสดงค่าจำนวนอันดับ AR(p)

ตัวอย่างตรวจสอบ ตัวแบบ ARIMA(0,0,1) หรือ MA(1)

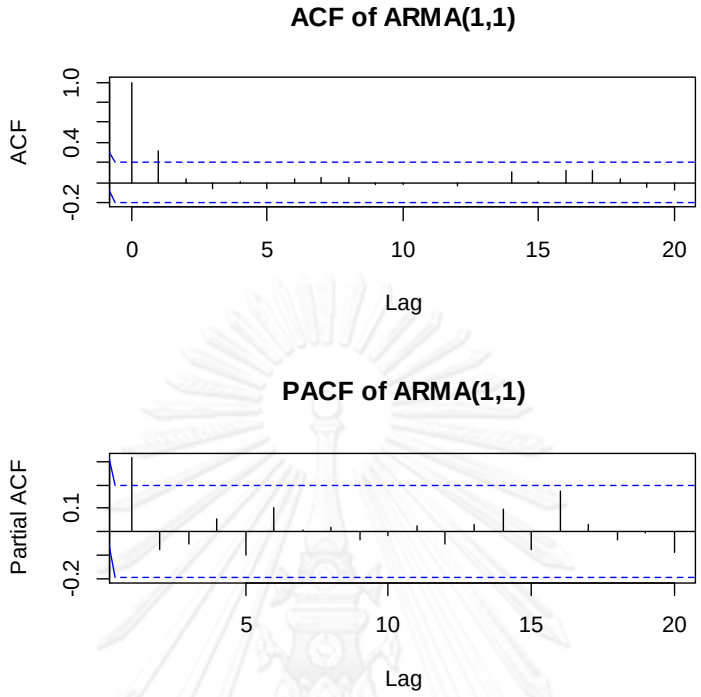


ภาพที่ 2.5 กราฟ ACF และ PACF สำหรับตัวแบบ MA(1)

กราฟ ค่า PACF จะค่อยๆ ลดลงอย่างช้าๆ ขณะที่กราฟ ACF จะลดลงอย่างรวดเร็ว Lag ที่ ค่า PACF ลดลงอย่างรวดเร็วจะแสดงค่าจำนวนอันดับ MA(q)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตัวอย่างตรวจสอบ ตัวแบบ ARIMA(1,0,1) หรือ ARMA(1,1)



ภาพที่ 2.6 กราฟ ACF และ PACF สำหรับตัวแบบ ARMA(1,1)

จากกราฟ ค่า PACF ลดลงอย่างรวดเร็วที่ Lag=1 บอกให้ทราบว่าค่าอันดับ AR(1) ส่วนกราฟ ACF มีค่าลดลงอย่างรวดเร็วหลังจากผ่าน Lag=1 บอกให้ทราบว่าค่าอันดับ MA(1)

สรุปวิธีการพิจารณาค่า ACF และ PACF ใช้เลือกอันดับ

จาก 3 ตัวอย่างข้างต้น เป็นตัวอย่างแสดงค่า ACF และ PACF ที่มาจากการจำลอง สำหรับในข้อมูลจริง เป็นการยากที่จะระบุอันดับ AR(p) และ MA(q) ได้อย่างถูกต้องจากการพิจารณา เพียงครั้งเดียว โดยเฉพาะอย่างยิ่งข้อมูลที่มีความการส่ายไปมา จึงจำเป็นต้องเลือก ตัวแบบอันดับข้างเคียงเข้ามาพิจารณา ให้เป็นกลุ่มตัวแบบที่เป็นไปได้ แล้วนำเข้าสู่กระบวนการประมาณค่า Parameter ต่อไป

<u>การระบุอันดับตัวแบบโดยใช้ค่า Information criteria</u> (Shi et al., 2012, Ömer Faruk, 2010, Nie et al., 2012)

จากความยุ่งยากในการระบุอันดับที่ถูกต้องของตัวแบบแบบเดิม จึงได้เกิดการระบุอันดับของตัวแบบโดยใช้ค่าสถิติที่ใช้ตรวจสอบและเปรียบเทียบ กับตัวแบบที่มีอันดับข้างเคียง ที่สำคัญและนิยมใช้กัน คือ Information Criterion แบ่งได้ออกเป็นสามชนิด คือ

Information Criterion คือ เครื่องมือในการเลือกตัวแบบ (Model Selection) เป็นค่าวัดทางสถิติยิ่งค่า Information Criterion ยิ่งน้อยยิ่งดี ยิ่งเหมาะสมที่จะเลือกอันดับของตัวแบบ ARIMA

ค่า Information Criterion แบ่งออกเป็น 3 ค่า ได้แก่

1) Akaike Information Criterion (AIC)

$$\mathrm{AIC} = -2\log(\mathrm{L}) + 2\mathrm{m} \quad (2.4)$$

2) Bayesian Information Criterion (BIC)

$$\mathrm{BIC} = -2\log(\mathrm{L}) + \mathrm{mlogn} \quad (2.5)$$

3) $$AICc = \mathrm{AIC} + \frac{2\mathrm{m}(\mathrm{m}+1)}{\mathrm{n}-\mathrm{m}-1} \quad (2.6)$$

โดยที่ L คือ ค่า likelihood ของค่าข้อมูล

n คือ จำนวนข้อมูลที่นำเข้าไปสร้างตัวแบบ

m คือ จำนวน Parameters ในตัวแบบ เช่นตัวแบบ ARIMA(p,d,q) จำนวน Parameter รวม คือ p+d+q

ตัวแบบใดมีค่า AIC, BIC หรือ AICc ต่ำที่สุดคือตัวแบบที่ต้องการ

ค่า AIC, BIC และ AICc ในตัวแบบ ARIMA เช่น ARIMA(1,1,1) และ ARIMA(2,1,1) จะเลือกอันดับที่ให้ค่า AIC, BIC หรือ AICc ต่ำกว่า ค่าทั้งสามจะเป็นไปในทิศทางเดียวกัน

สาเหตุที่ต้องมีการกำหนดขอบเขตค่า p และ q เนื่องจาก เพื่อความสะดวกรวดเร็วในการประมวลผล ค้นหาอันดับที่ดีที่สุด การเลือกอันดับโดยใช้ Information criteria นี้ ปัจจุบันเป็นที่นิยมใช้กันมาก เนื่องจากไม่ต้องไปพิจารณากราฟ ACF และ PACF ซึ่งเป็นการยากที่จะใช้เลือกอันดับที่ดีที่สุด โดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีความซับซ้อน

4. Model Estimation

การประมาณค่า Parameter คือ การประมาณค่าสัมประสิทธิ์ของตัวแบบ ซึ่งตัวแบบ ARIMA เป็นตัวแบบเชิงเส้นตรง มีลักษณะผลรวมเชิงเส้น (Linear combination) ของเทอม AR และเทอม MA ค่าสัมประสิทธิ์ที่ได้จะอยู่หน้าเทอม AR และ MA

วิธีการประมาณค่าที่นิยมใช้กัน มี 3 วิธี คือ

1) Maximum likelihood

2) Conditional least squares

3) Unconditional least squares

5. Model Diagnostic

หลังจากได้ค่าประมาณ Parameter ของตัวแบบ ARIMA(p,d,q) แล้ว จะสู่การตรวจสอบว่าตัวแบบที่เลือกมานั้น มีความถูกต้อง ตามหลักสถิติและสามารถนำไปใช้ได้จริง(Goodness of fit) หรือไม่

ตัวชี้วัดก็คือ ค่าเศษเหลือ หรือ Residuals มาจากความสัมพันธ์ คือ Residuals = ค่าจริง – ค่าพยากรณ์

เงื่อนไขที่จะยอมรับว่าตัวแบบ ที่เลือกมา มีความถูกต้องตามหลักสถิติ คือ

1) Residuals ต้องมีลักษณะการกระจายตัวแบบสุ่ม ไม่มีค่าอัตตสัมพันธ์ในตัวเอง (Independence) ทดสอบโดย

1.1) พิจารณาค่า ACF ของ residuals ควรที่จะอยู่ใกล้ 0

1.2) ทดสอบโดยใช้ตัวสถิติทดสอบ Ljung-Box Q และ runs tests

2) ความแปรปรวนของ Residuals มีค่าคงที่ (Homoscedasticity)

พิจารณากราฟ Residuals vs Time (Scatter plot)

3) Residuals มีการแจกแจงแบบปกติ (Normality)

3.1) ทดสอบโดยใช้ตัวสถิติทดสอบ Kolmogorov-Smirnov (K-S)

3.2) ทดสอบโดยใช้ตัวสถิติทดสอบ Anderson-Darling (AD)

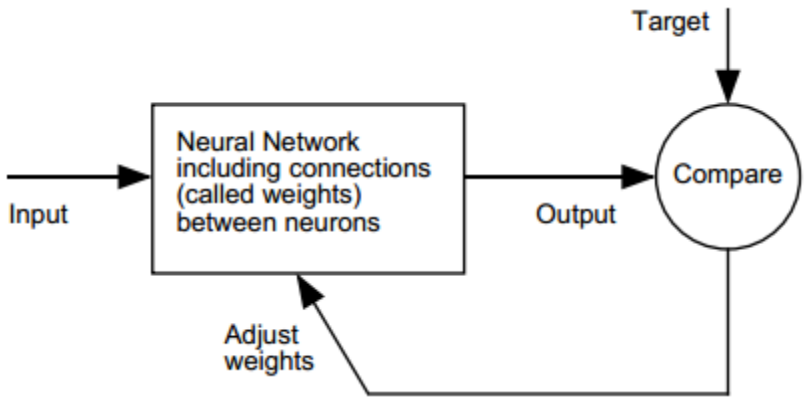
3.3) พิจารณากราฟ Histogram with normal curve, Normal probability plots, Quantile Quantile plots

2.2 ตัวแบบ Artificial neural network (ANN)

ตัวแบบ ANN ถือเป็นตัวแบบพยากรณ์ที่ได้รับความนิยมเป็นอย่างสูงในช่วงไม่กี่ปีที่ผ่านมา เนื่องมาจากความประสบความสำเร็จในการนำไปประยุกต์ใช้ในหลายๆ ปัญหา เป็นอย่างดี เช่นด้านการเงิน การแพทย์ วิศวกรรม ธรณีวิทยา ฟิสิกส์ และในด้านอื่นๆ ทุกด้าน ที่ครอบคลุมปัญหาทางด้านการพยากรณ์ทั้ง Classification, Control, Regression และการพยากรณ์อนุกรมเวลา (Time series forecasting) ความสำเร็จของ ANN มาจาก 2 ปัจจัยหลัก คือ ANN เป็นตัวแบบที่มีความสามารถที่หลากหลาย สามารถแก้ปัญหาพยากรณ์ข้อมูลที่มีความสลับซับซ้อนได้ดี มีความแม่นยำอยู่ในระดับสูง โดยปกติจะใช้แก้ปัญหาที่ไม่เป็นเชิงเส้นตรง (Nonlinear) ส่วนปัจจัยที่สอง คือง่ายต่อการใช้งาน เนื่องจากตัวแบบ ANN ไม่จำเป็นต้องมีเงื่อนไข (Assumption) ในการสร้างตัวแบบ ดังเช่นตัวแบบพยากรณ์ทางสถิติ เพียงแต่ต้องปรับจูนค่าพารามิเตอร์ของตัวแบบหลายครั้ง เพื่อให้ได้ตัวแบบพยากรณ์ที่มีความแม่นยำในระดับที่น่าพอใจ

2.2.1 แรงบันดาลของ ANN

ANN หรือเรียกว่า เครือข่ายประสาทเทียม เป็นตัวแบบที่ทำงานแบบขนาน เลียนแบบการระบบการทำงานในสมอง ส่วนประกอบต่างๆ จะเชื่อมกันด้วย Connection weights หรือเรียกสั้นๆ ว่า Weights ซึ่ง ANN จะเข้าสู่กระบวนการเรียนรู้ (Training) และปรับค่า Weights ไปเรื่อยๆ จนกระทั่งได้ค่าที่ใกล้เคียงค่าเป้าหมาย (Target output) ดังแสดงไว้ในภาพที่ 2.7



ภาพที่ 2.7 หลักการทำงานของตัวแบบ ANN (Beale et al., 2013)

2.2.2 การประยุกต์ใช้ Neural network ในงานต่างๆ ได้แก่

ปัจจุบันนี้มีการนำตัวแบบ ANN นำมาใช้ในงานต่างๆ หลายประเภท เช่น สถาบันการเงินหลายแห่งใช้ ANN ทำนายราคาหุ้นในอนาคต ซึ่งก็สามารถทำได้ เนื่องจากข้อมูลราคาหุ้นในอดีตนั้นมีแนวโน้ม (Trend) หรือมีปัจจัยที่พอจะบ่งบอกลักษณะ (Pattern) ของข้อมูล สิ่งเหล่านี้จะช่วยให้การสร้างตัวแบบมีความแม่นยำมากขึ้น

การใช้ตัวแบบ ANN พยากรณ์โอกาสที่ผู้ป่วยจะเป็นโรคเบาหวาน โรคมะเร็ง การพยากรณ์ค่าใช้จ่ายในโรงพยาบาล การควบคุมคุณภาพการผลิตสินค้าในโรงงานอุตสาหกรรม การใช้ตัวแบบ ANN ในการค้นหาวัตถุเป้าหมายในการทหาร การควบคุมทิศทางการเคลื่อนไหวของหุ่นยนต์ เป็นต้น กล่าวคือตัวแบบ ANN สามารถนำไปประยุกต์ใช้ได้ทุกงาน ที่เกี่ยวข้องกับการพยากรณ์ข้อมูล การจำแนกข้อมูลเป็นกลุ่ม การค้นหารูปแบบลักษณะของข้อมูล แต่ก่อนที่จะนำตัวแบบ ANN ไปประยุกต์ใช้ก็ต้องเข้าใจปัญหาอย่างชัดเจนก่อน ถ้าไม่เข้าใจขอบเขตการแก้ปัญหาของ ANN ที่ดีก็อาจทำให้ตัวแบบที่สร้างมาเชื่อถือไม่ได้ หรือไม่มีความแม่นยำเพียงพอในการใช้งานจริง ตัวอย่างปัญหาการพยากรณ์ที่ไม่สามารถใช้ตัวแบบ ANN ได้ เช่น ต้องการพยากรณ์ลอตเตอรีที่จะออกในงวดหน้า แต่ทว่ามีข้อมูลขนาดรองเท้าที่ใส่อยู่ นำไปเป็นตัวแปรอิสระใช้พยากรณ์ จะเห็นได้ชัดว่าข้อมูลทั้งสองไม่ได้มีความสัมพันธ์ (Relationship) กันเลย หรือปัญหาการพยากรณ์ลอตเตอรีที่จะออกในงวดหน้าโดยใช้ข้อมูลลอตเตอรีที่ออกในอดีตก็เป็นการยากที่จะนำข้อมูลในอดีตเหล่านั้นมาพยากรณ์ เนื่องจากลอตเตอรีที่ออกในอดีต ไม่ได้มีความสัมพันธ์กับลอตเตอรีที่จะออกในงวดหน้าเช่นกัน ดังนั้นสิ่งที่สำคัญและจำเป็นต้องเข้าใจ คือ ต้องทราบว่าตัวแปรอิสระที่จะนำไปพยากรณ์ตัวแปรตามนั้นต้องมีความสัมพันธ์กัน เกี่ยวเนื่องกัน หรือต้องสงสัยว่าเกี่ยวข้องกันทางใดทางหนึ่ง (สัมพันธ์ไปในทางเดียวกันหรือตรงกันข้าม) ซึ่งความสัมพันธ์ที่ว่ามานี้คงจะไม่ใช่ 100% เป็นลักษณะที่กำกวมบ้าง ไม่ชัดเจนไปเสียทีเดียว (Noisy) ก็ถือเป็นขอบเขตที่ ANN ที่จะทำงานต่อไปโดยการปล่อยให้ตัวแบบเรียนรู้ความสัมพันธ์ระหว่าง ข้อมูลนำเข้า (Input) และข้อมูลที่ออกมา (Output) หรือเรียกว่า Training สามารถแบ่งออกได้เป็น 2 ประเภท คือ

1. Supervised training คือตัวแบบที่มีการเทรน โดยตัวแบบจะ fit กับข้อมูลชุดเรียนรู้ (Training set) โดยจะนำเข้าข้อมูลเป็นแถวๆ (Record) ที่มีข้อมูลทั้ง Input และ Output ครบถ้วน มักเป็นข้อมูลเก่า เช่น ข้อมูลราคาหุ้นในอดีต ข้อมูลส่วนตัวของลูกค้าบัตรเครดิต ข้อมูลปฏิกิริยาที่หุ่นยนต์ตอบสนองเมื่อเจอกับสิ่งเร้าที่มากระทบ เป็นต้น จากนั้นทำการเทรน แล้วนำตัวแบบที่ได้ ไปทดสอบประสิทธิภาพกับข้อมูลชุดอื่นที่ตัวแบบไม่เคยเจอมาก่อน เรียกว่า ข้อมูลชุดทดสอบ (Test set) ตัวแบบ ANN ส่วนมากจัดอยู่ในหมวดนี้ ได้แก่ Feedforward network ,Multilayerperceptron (MLP), Nonlinear autoregressive (NAR) network , Nonlinear autoregressive with exogenous input (NARX) network, Time delay network, Recurrent network, Radial Basis network, Probabilistic network , Generalized Regression network

2. Unsupervised training คือตัวแบบที่ไม่มี Output มีเฉพาะ Input หลักการทำงานของตัวแบบประเภทนี้ คือ การจัดกลุ่ม (Clustering) ข้อมูลในแต่ละ record ที่มีความคล้ายคลึงกัน อยู่เป็นกลุ่มเดียวกัน และพยายามให้แต่ละกลุ่มมีความแตกต่างกันมากที่สุด ตัวแบบ neural network ที่จัดอยู่ในหมวดนี้ ได้แก่ Self Organizing Feature Map (SOFM, หรือ Kohonen) networks

งานวิจัยชิ้นนี้สนใจเฉพาะตัวแบบประเภท Supervised training โดยเฉพาะตัวแบบ NAR เนื่องจากเป็นตัวแบบที่ใช้ในงานนี้ และจะกล่าวลงในรายละเอียดต่อไป ส่วนตัวแบบอื่นๆ ทั้ง Supervised และ Unsupervised สามารถค้นคว้าศึกษาเพิ่มเติมได้ในตำรา วารสารวิชาการอื่นๆ

กล่าวโดยสรุปการใช้งาน ANN แบ่งออกเป็น 4 กลุ่ม คือ

Fitting a Function

เป็นการประมาณค่าฟังก์ชัน ลักษณะการทำงานคล้ายกับเทคนิค Regression ในการวิเคราะห์ทางสถิติ

Recognizing Patterns

เป็นการพยากรณ์ข้อมูลที่เป็นกลุ่ม จัดกลุ่มข้อมูลที่มีการเรียนรู้ (Supervised learning) หรือเรียกว่า Pattern recognition ลักษณะการทำงานคล้ายกับเทคนิค Logistic regression, Discriminant analysis ในการวิเคราะห์ทางสถิติ

Clustering Data

จัดกลุ่มข้อมูลที่มีความคล้ายกัน โดยใช้คำนวณระยะห่างระหว่างจุดเป็นตัวแบ่ง เป็นการจัดกลุ่มข้อมูลที่ไม่มีการเรียนรู้ (Unsupervised learning) เช่น การแบ่งกลุ่มลูกค้าที่เข้ามาซื้อสินค้าหรือเข้ามาใช้บริการ (Market segmentation) การทำเหมืองข้อมูล (Data mining)การจัดกลุ่มยีน Bioinformatic analysis ลักษณะการทำงานคล้ายกับเทคนิค Clustering ในการวิเคราะห์ทางสถิติ

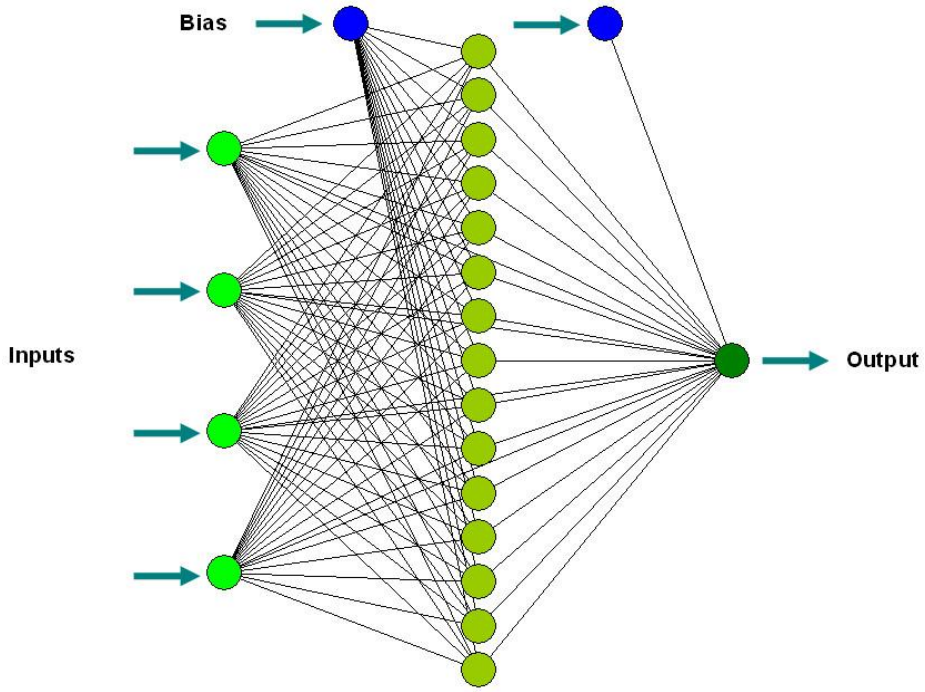
Time series forecasting

ตัวแบบ ANN สามารถประยุกต์นำมาใช้พยากรณ์ข้อมูลอนุกรมเวลาได้ คล้ายกับ Function fitting แต่ต้องปรับตัวแปรอิสระ โดยทำให้ข้อมูลในอดีตกลายเป็นตัวแปรอิสระ ตามช่วงเวลาต่างๆ หรือเรียกว่า Lag (กล่าวถึงรายละเอียดในหัวข้อถัดไป) ลักษณะการทำงานคล้ายกับเทคนิค ARIMA ในการวิเคราะห์ทางสถิติ

CHULALONGKORN UNIVERSITY

2.2.3 เครือข่าย Feedforward (Feedforward neural network)

นิยมเรียกอีกอย่างว่า Multilayer perceptron (MLP) เป็นโครงสร้างการทำงานของ ANN เป็นลักษณะเป็นไปตามรูปข้างล่าง



ภาพที่ 2.8 โครงสร้างของ Feedforward network

ค่า Inputs จากแต่ละตัวแปร จะถูกส่งผ่าน Input neurons ไปยัง Hidden neurons แบบทั่วถึงกันหมด (Fully connected) จากนั้นก็เข้าสู่ Output neurons ในขั้นสุดท้าย

ตัวแบบ Multilayer Perceptron (MLP) เป็นตัวแบบพยากรณ์ (Predictive model) ที่นิยมใช้พยากรณ์ข้อมูล แบ่งโครงสร้างออกเป็น 3 ชั้น คือ

1. Input layer คือ ชั้นที่รับข้อมูลเพื่อสร้างตัวแบบ จะมี 1 layer ประกอบด้วย Input neurons จำนวน Input neurons ขึ้นอยู่กับจำนวนตัวแปรอิสระที่เลือกใช้ในการสร้างตัวแบบ อาจมี 1 ตัว หรือหลายตัวก็ได้

2. Hidden layer คือ ชั้นถัดจาก Input layer ปกติจะมีชั้นเดียว เนื่องจากมีความสามารถเพียงพอต่อการพยากรณ์แทบทุกปัญหา (Zhang, 2003, Khashei and Bijari, 2011) ภายใน Hidden layer จะประกอบด้วย Hidden neurons เป็นรูปวงกลมดังภาพ จะมีจำนวนที่ไม่แน่นอน ต้องทำการทดลองหาค่า จำนวน Hidden neurons ที่เหมาะสม จัดเป็นหนึ่งในพารามิเตอร์ที่ผู้วิจัยต้องกำหนดขอบเขตแทนค่าเอง

3. Output layer คือ ชั้นสุดท้ายของ Network เป็นชั้นที่ได้ค่าพยากรณ์ที่นำไปใช้จริง จะมีเพียงชั้นเดียว ประกอบด้วย Output neurons กรณีที่พยากรณ์เชิงกลุ่ม จำนวน Output neurons จะเท่ากับจำนวนกลุ่มที่เราตั้งค่าไว้ เช่น ถ้าต้องการพยากรณ์กลุ่มของดาวเคราะห์ ว่าอยู่กลุ่ม A, B หรือ C โดยใช้ตัวแปรอิสระต่างๆ จำนวน Output neurons เท่ากับ 3 กรณี Regression ก็คล้ายคลึงกัน เช่น ต้องการพยากรณ์ค่าใช้จ่ายของโรงพยาบาลพร้อมๆ กับระยะเวลาที่ผู้ป่วยพักฟื้นใน

โรงพยาบาล กรณีนี้ก็จะมี 2 Neurons ขณะที่ Time series จะมี 1 Neuron กรณีพยากรณ์ค่าล่วงหน้าค่าเดียว แต่ถ้าต้องการพยากรณ์ 5 ค่า และ 10 ค่าล่วงหน้า ก็จะมี 2 Neurons

2.2.4 ตัวแปรที่ใช้ในการสร้างตัวแบบ

1. ตัวแปรต้น (Dependent variables) สามารถแบ่งได้ออกเป็น 4 ประเภท คือ

1.1 ข้อมูลแบบกลุ่ม (Nominal) ข้อมูลจะเป็นกลุ่มๆ ที่แบ่งแยกได้ชัดเจน ไม่มีอันดับ ไม่สามารถเปรียบเทียบว่ามากกว่าหรือน้อยกว่า ดีกว่าหรือแย่กว่าได้ เช่น ข้อมูลรหัสไปรษณีย์ เขตพื้นที่อำเภอต่างๆ จังหวัดต่างๆ ประเทศต่างๆ แผนกต่างๆ ศาสนานี่นับถือ เชื้อชาติ สีผิว เป็นต้น

1.2 ข้อมูลเชิงกลุ่มแบบอันดับ (Ordinal) ข้อมูลจะแบ่งเป็นกลุ่มคล้ายข้อมูลแบบ Nominal เพียงแต่สามารถจัดอันดับ (Ranking) เปรียบเทียบกันได้ เช่น ข้อมูลระดับคะแนนความพึงพอใจ

1.3 ข้อมูลที่เป็นตัวเลข (Scale) ข้อมูลที่มีหน่วยแบบต่อเนื่อง เปรียบมากกว่าหรือน้อยกว่าได้ หาขนาดความห่างหรือความมากหรือน้อยได้ เช่น อายุ รายได้ต่อเดือน จำนวนบุตร เป็นต้น

ข้อมูลทั้ง 3 กลุ่มนี้ สามารถทำการเปลี่ยนแปลงได้ตามความเหมาะสม เช่นข้อมูลเชิงตัวเลขอาจเปลี่ยนทำให้เป็นข้อมูลเชิงกลุ่ม ข้อมูลเชิงอันดับให้ทำให้เป็นข้อมูลเชิงตัวเลขได้ ข้อมูลเชิงอันดับก็ทำให้เป็นข้อมูลเชิงกลุ่มได้ ทั้งนี้ทั้งนั้นก็ขึ้นอยู่กับ วิธีการที่จะทำให้มีความเหมาะสม

2. ตัวแปรตาม (Independence variables) หรือเรียกว่า (Predictor variables) เป็นตัวแปรใช้ในการประเมินประสิทธิภาพตัวแบบ มีเฉพาะตัวแบบ ANN ที่มีการเรียนรู้ แบ่งออกได้เป็น 2 แบบ คือ

2.1 แบบเชิงกลุ่ม อาจเรียกว่า Factors หรือ Categorical แต่ผลที่ออกมามักเป็นค่าตัวเลขที่เป็นกลุ่ม เช่น (0,1,2,...) เป็นต้น เนื่องจากมีการใส่รหัส (Coding) เพื่อง่ายต่อการคำนวณ ผู้วิเคราะห์จึงต้องแปลงกลับไปอีกครั้งหนึ่ง การพยากรณ์ในลักษณะนี้จัดอยู่ในกลุ่ม Pattern recognition มีลักษณะคล้ายกับเทคนิค Logistic regression หรือ Discriminant ในทางสถิติสำหรับกรณีที่กลุ่มมีจำนวนมาก เช่น (0,1,2,3,4,5,6,7,8,9,10) มักจะทำให้ใช้เวลาในการเทรนเป็นเวลาที่นาน จึงเป็นจำเป็นต้องจัดการรวมกลุ่มที่มีลักษณะคล้ายไว้ในกลุ่มเดียวกัน

2.2 แบบเป็นตัวเลข เป็นเชิงปริมาณ อาจเรียกว่า Covariate หรือ Scale ค่าพยากรณ์ที่ออกจากตัวแบบจะเป็นค่าตัวเลขแบบมีค่าต่อเนื่อง เช่น 5,5.11,6.246 การพยากรณ์ในลักษณะนี้จัดอยู่ในกลุ่ม Fitting function และประยุกต์ใช้ใน Time series forecasting มีลักษณะคล้ายกับเทคนิค Multiple regression และ Nonlinear regression ในการวิเคราะห์ทางสถิติ

2.2.5 การสำรวจข้อมูล (Data exploration)

หลังจากที่ได้ปัญหาที่เหมาะสมที่จะใช้ ANN ในการแก้ปัญหาแล้ว ก็จะเข้าสู่ขั้นตอนการเตรียมข้อมูลให้พร้อมก่อนที่จะนำไปสร้างตัวแบบ สิ่งที่ต้องพิจารณา เช่น ข้อมูลที่เป็นแถวที่มีตัวแปรอิสระและตัวแปรตามในแถวเดียวกันอย่างครบถ้วน โดยทั่วไปมักจะมีตัวแปรอิสระหลายๆตัว แล้วทำการพยากรณ์ตัวแปรตามเพียงตัวแปรเดียว ซึ่งจำเป็นต้องคัดเลือกตัวแปรอิสระที่จำเป็นหรือคาดว่ามีความสัมพันธ์ต่อกัน ขั้นแรกก็ต้องใช้ประสบการณ์ของผู้วิจัยคัดเลือกตัวแปรอิสระเสียก่อน

จากนั้นทำการตรวจสอบข้อมูลว่ามีความเหมาะสมในการนำเข้าตัวแบบหรือยัง เช่น

1. กรณีมีข้อมูลที่หายไป (Missing data) ลักษณะข้อมูลที่หายไป ค่าที่หายไปอาจเป็นตัวแปรตาม หรือเป็นตัวแปรอิสระตัวแปรเดียวหรือหลายตัวแปร ทั้งนี้ถ้าผู้วิจัยต้องการจะตัดข้อมูลในแถวนั้นทิ้ง ต้องดูจำนวนข้อมูลประกอบว่าเหลือข้อมูลน้อยไปหรือไม่ (โดยทั่วไป ANN จะทำงานได้ดีก็ต่อเมื่อมีข้อมูลในหลักร้อยขึ้นไป) ถ้าเหลือน้อยไปอาจเก็บแถวนั้นไว้ แล้วแทนค่าที่ว่างนั้นด้วยค่าเฉลี่ย (Mean) ค่ามัธยฐาน (Median) ค่าฐานนิยม (Mode) ค่าเฉลี่ยตัวที่อยู่ติดกันหรือค่าอื่นที่มาจากการคำนวณทางสถิติและค่าที่นำมาแทนต้องสอดคล้องกับตัวแปรของค่าที่หายไปเหล่านั้น

2. ปกติตัวแบบ ANN จะทำการพยากรณ์ได้ดีก็ต่อเมื่อข้อมูลนำเข้าอยู่ในช่วงที่จำกัด มีขอบเขตที่แน่ชัด ข้อมูลที่เป็นตัวเลข (Continuous) จำเป็นต้องปรับช่วงให้อยู่ในช่วงที่เหมาะสมสำหรับข้อมูลประเภทกลุ่ม (Categorical) เช่น ข้อมูลตัวแปรเพศ={ชาย,หญิง} จะมีความยุ่งยากมากกว่าข้อมูลที่เป็นตัวเลข เนื่องจากตัวแบบ ANN จะทำงานได้ไม่ดีนักเมื่อข้อมูลในตัวแปรนั้นมีหลายค่า เช่น ตัวแปรอาชีพ={ทหาร,ตำรวจ,แพทย์,พยาบาล,ครู,อาจารย์ในมหาวิทยาลัย,นักศึกษา,พ่อค้าแม่ค้าก๋วยเตี๋ยว,พ่อค้าแม่ค้าเสื้อผ้า,ชาวนา,ชาวสวนยาง,..} เห็นได้ชัดว่าตัวแปรอาชีพในความเป็นจริงสามารถแบ่งอาชีพได้เป็นอาชีพย่อยๆ มากมาย ลักษณะนี้ตัวแบบ ANN จะให้ค่าพยากรณ์ที่ไม่ดีแน่นอน จำเป็นต้องจัดกลุ่มอาชีพใหม่ เช่น ตัวแปรอาชีพ={ข้าราชการ,เกษตรกร,พ่อค้าแม่ค้า,นักธุรกิจ}

3. ตัวแปรอิสระบางตัว เช่น รหัสไปรษณีย์ ที่อยู่ เป็นตัวแปรเชิงกลุ่ม เป็นตัวแปรที่มีค่าไม่ซ้ำกัน ดังนั้นตัวแปรแบบนี้สามารถตัดออกไป โดยไม่ส่งผลต่อความแม่นยำของการพยากรณ์ ยิ่งทำให้ความแม่นยำลดลง และเสียเวลาในการเทรน จึงไม่นำมาใช้เป็นตัวแปรอิสระ

4. การลบแถวข้อมูลที่มีค่าซ้ำกันหรือค่าเดียวกันทั้งหมด เนื่องจากข้อมูลในแถวนั้นๆ จัดเป็นข้อมูลที่ไม่เป็นประโยชน์ มีความซ้ำซ้อนทำให้เสียเวลาในการเทรน

5. จำนวนข้อมูลขั้นต่ำที่นำมาใช้ใน Neural networks เป็นการยากที่จะระบุจำนวนขั้นต่ำ เนื่องจาก มีปัจจัยที่ส่งผลกระทบมาก เช่น ความซับซ้อนของข้อมูล ความแปรปรวนของข้อมูล (Variation) เครือข่ายของตัวแบบที่เลือกใช้เป็นต้น สำหรับตัวแบบเครือข่าย Feedforward มีผู้วิจัยแนะนำว่าให้มีจำนวนข้อมูลเป็น 10 เท่าของจำนวน Weights รวมทั้งหมด เมื่อตัวแปรอิสระมีจำนวนเพิ่มขึ้น จำนวนข้อมูลจะต้องเพิ่มมากขึ้นแบบไม่เป็นเส้นตรง อาจอยู่ในเทอม เอกซ์โปเนนเชียล หรืออื่นๆ

กล่าวโดยสรุป ในงานวิจัยส่วนมากเสนอว่าจำนวนข้อมูลขั้นต่ำควรจะเป็นหลักร้อยขึ้นไปสำหรับปัญหาที่จำเป็นต้องใช้ตัวแปรอิสระมาก ก็ยิ่งต้องเพิ่มจำนวนข้อมูลมากยิ่งขึ้น อาจเป็นหลักหมื่น ใน

กรณีที่มีจำนวนข้อมูลน้อยกว่าหนึ่งร้อยชุด แต่จำเป็นต้องทำ ก็มีวิธีปรับปรุง คือ การทำ Boostrapping

อีกหนึ่งปัญหาในการนำตัวแบบ ANN ไปใช้ ที่พบมาก คือ ข้อมูลเชิงตัวเลขมีการกระจายตัวมาก ข้อมูลส่ายไปส่ายมา (Noise) และข้อมูลที่มีค่าสูงมากหรือต่ำมากที่โดดเด่น หรือเรียกว่า Outliers ตัวแบบ ANN จัดว่าเป็นตัวแบบที่คงทน (Tolerant) ต่อ Outlier พอสมควร หมายถึง แม้ว่าจะมี Outliers บ้างก็ไม่ทำค่าพยากรณ์แย่มากนัก แต่ถ้ามี Outliers มากก็จะทำให้ค่าพยากรณ์แย่ลง วิธีการจัดการ Outliers ที่นิยม คือ การตัดทิ้งไป ตัดข้อมูลทั้งแถว หรือการเปลี่ยนให้เป็น Missing value แล้วทำการแทนค่ากลับ

2.2.6 การแปลงข้อมูลก่อนและหลังกระบวนการเรียนรู้ (Data preprocessing and postprocessing)

การแปลงข้อมูลก่อนนำข้อมูลสร้างตัวแบบ สำหรับ ANNเป็นสิ่งที่ควรทำเนื่องจากจะทำให้การสร้างตัวแบบใช้เวลาน้อยลงหรือมักจะทำให้ได้คำตอบที่มีประสิทธิภาพมากยิ่งขึ้น

การแปลงข้อมูล (Data transformation) ให้อยู่ในช่วงที่เหมาะสม ได้แก่

1. การใส่ Logarithm ข้อมูลดิบ ส่วนใหญ่นิยม ใส่ Logarithm ฐาน 10 หรือ ฐานธรรมชาติ ลักษณะข้อมูลที่น่าจะเหมาะกับการใส่ Logarithm คือข้อมูลที่มีการเพิ่มขึ้นแบบก้าวกระโดด เป็นแบบ Exponential จัดได้ว่าข้อมูลมีความสัมพันธ์แบบผลคูณ (Multiplicative relations) การใส่ Logarithm เข้าไปเปรียบเสมือนการปรับความสัมพันธ์เป็นแบบผลบวก (Additive relations)

2. การแปลงข้อมูลเชิงเส้นตรง (Linear transformation) หรือเรียกว่า Normalized คือการนำข้อมูลดิบ มาจำกัดช่วง เช่นอยู่ในช่วง 0 ถึง 1 และ -1 ถึง 1

- ช่วง 0 ถึง 1

$$\hat{Y}_t = \frac{Y_t - Y_{min}}{Y_{max} - Y_{min}} \quad (2.7)$$

- ช่วง -1 ถึง 1

$$\hat{Y}_t = \left(\frac{2\times(Y_t - Y_{min})}{Y_{max} - Y_{min}}\right) - 1 \quad (2.8)$$

$\hat{Y}_t$ คือ ค่าหลังจากการแปลง

Y_t คือ ข้อมูลดิบ

Y_{max} คือ ข้อมูลดิบที่มีค่ามากที่สุด

Y_{min} คือ ข้อมูลดิบที่มีค่าน้อยที่สุด

3. การแปลงข้อมูลแบบ Standardization แปลงข้อมูลโดยใช้คุณสมบัติทางสถิติ ที่มีค่าเฉลี่ยและค่า Standard deviation ของข้อมูลดิบมาเกี่ยวข้องด้วย

$$\hat{Y}_t = \frac{Y_t - \bar{Y}}{std.Y} \quad (2.9)$$

$\hat{Y}_t$ คือ ค่าหลังจากการแปลง

Y_t คือ ข้อมูลดิบ

$\bar{Y}$ คือ ค่าเฉลี่ยข้อมูลดิบ

$std.Y$ คือ ค่า Standard deviation ของข้อมูลดิบ

4. การ Differencing คือ การนำข้อมูลดิบตัวติดกันมาลบกัน ดังเช่นการ Differencing ในการ fit ตัวแบบ ARIMA เป็นการทำข้อมูลให้เรียบขึ้น กำจัดแนวโน้ม (Trends) และ ลักษณะฤดูกาล (Seasonality) ออกไป

การแปลงข้อมูลให้อยู่ในช่วงที่เหมาะสมนั้น ถือเป็นเรื่องจำเป็นเนื่องจากตัวแบบ ANN ไม่สามารถทำการพยากรณ์ได้เมื่อข้อมูลมี Scale ที่หลายหลาก มีค่าขอบเขตของข้อมูลในตัวแปรที่นำเข้าต่างกันมาก งานวิจัยชิ้นนี้ เลือกทำการแปลงข้อมูลให้อยู่ในช่วง -1 ถึง 1 เพื่อให้ค่าความห่างของข้อมูล สอดคล้องกับ Activation ฟังก์ชันที่เลือกใช้ คือ Hyperbolic tangent sigmoid function

ภายหลังจากเทรน เรียบร้อยแล้ว Output ที่ได้จะต้องทำการแปลงกลับให้อยู่ในรูปแบบเดิมก่อนทำ Preprocessing เรียกกระบวนการนี้ว่า Postprocessing ค่าที่ได้นี้จะเป็นค่าพยากรณ์ที่สามารถนำไปใช้จริง

2.2.7 การแบ่งข้อมูลเพื่อใช้ในการเรียนรู้และทดสอบ (Data set splitting)

การสร้างตัวแบบและพยากรณ์ด้วย ANN นี้จำเป็นต้องแบ่งข้อมูลออกเป็นอย่างน้อย 2 ชุด คือ ข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ เกณฑ์ในการแบ่งจำนวนข้อมูล ทั้งชุด Train หรือชุด Test นั้นไม่ตายตัว มักแบ่งจำนวนข้อมูลชุด Train นั้นมากกว่า จำนวนข้อมุลชุด Test โดยทั่วไปมักแบ่งตามสัดส่วน เช่น จำนวนข้อมูลชุด Train/Test เป็นร้อยละ 70/30, 80/20 , 90/10 หรืออาจกำหนดจำนวนด้วยตนเอง เช่น ข้อมูลอนุกรมเวลาจำนวน 145 ข้อมูล เลือกจำนวนข้อมูลสำหรับเทรน 110 ข้อมูล และเลือกจำนวนข้อมูลชุดทดสอบ 35 ข้อมูล

นอกจากจำนวนข้อมูลแบ่งแล้ว ยังมีการสับเปลี่ยนตำแหน่งข้อมูลก่อนแบ่ง กล่าวคือ ข้อมูลจากเดิมที่อยู่ตำแหน่งแรก (ข้อมูลแถวแรก) หลังสุ่มจะอยู่ตำแหน่งที่ 60 เป็นต้น หรือแบ่งตรงๆ ไม่เปลี่ยนตำแหน่งข้อมูล ก็ทำได้ไม่มีกฎเกณฑ์ตายตัว

2.2.8 ฟังก์ชัน Activation (Activation function)

หรือเรียกว่า Transfer function, Threshold function จะอยู่ 2 ที่ คือ

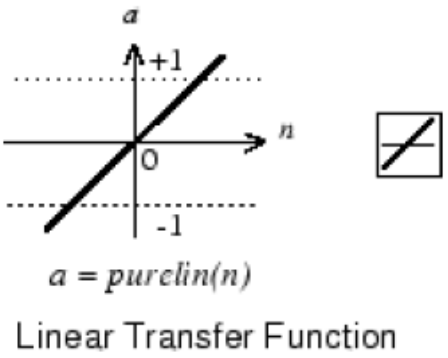
1. อยู่ใน Hidden neurons ทำการแปลงค่าข้อมูลที่นำเข้ามาจาก Input neurons แปลงให้เป็นค่าส่งออก ไปยัง Output neuron

2. อยู่ใน Output neuron ทำการแปลงค่าข้อมูลที่นำเข้ามาจาก Hidden neurons แปลงให้เป็นค่าส่งออกที่สามารถนำไปใช้ ซึ่งก็คือค่าพยากรณ์นั่นเอง

โดยทั่วไป Activation นั้นมีหลายฟังก์ชันที่ใช้กัน แต่ที่นิยมใช้ที่สุด ได้แก่

- Linear function หรือ Identity function

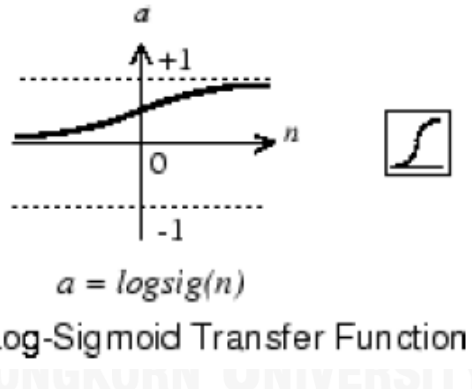
$f(X) = X$เป็นฟังก์ชันที่อยู่ใน Output neuron จะอยู่ในชั้นสุดท้ายของเครือข่าย ใช้ในการพยากรณ์ที่เป็นตัวเลข การประมาณค่า การพยากรณ์อนุกรมเวลา และนำมาใช้ในงานวิจัยชิ้นนี้ด้วย



ภาพที่ 2.9 Linear Transfer Function (Beale et al., 2013)

- Sigmoid logistics function

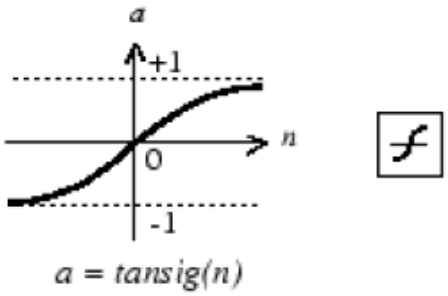
$\mathrm{f(x)} = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}}$ ทำการแปลงสถานะของข้อมูลนำเข้าให้อยู่ในรูปแบบการจัดแบ่งเชิงกลุ่ม เช่น 0 หรือ 1, ใช่ หรือ ไม่ใช่, ON หรือ OFF เป็นต้น ใช้ในทุกๆ Neuronsใน Hidden layer ยกเว้น Output neuronค่าที่ได้หลังจากผ่านฟังก์ชันนี้จะอยู่ระหว่าง 0 ถึง 1



ภาพที่ 2.10 Log-Sigmoid Transfer Function (Beale et al., 2013)

- Tan-sigmoid function หรือ Hyperbolic tangent function

$\mathrm{f(x)} = \frac{\mathrm{e^{x}-e^{-x}}}{\mathrm{e^{x}+e^{-x}}}$ คล้าย Sigmoid function ใช้ที่เดียวกัน ใช้แทนกันได้ แต่ค่าที่ได้หลังจากผ่านฟังก์ชันนี้ คือ -1 ถึง 1



ภาพที่ 2.11 Tan-Sigmoid Transfer Function (Beale et al., 2013)

ทั้ง Sigmoid และ Tan-sigmoid สามารถใช้ใน Output neurons ในปัญหาการพยากรณ์เชิงกลุ่ม เนื่องจากค่าที่ได้จะอยู่ช่วง ที่มีขอบเขตจำกัด และค่านี้เองจะแปลงกลับเป็นค่าพยากรณ์ที่ต้องการ เช่น ค่าพยากรณ์=0.345 ก็จะแปลงกลับเป็นกลุ่มตามที่ทำการใส่รหัสไว้ในข้างต้น แต่จะไม่เหมาะที่นำมาใช้พยากรณ์เชิงตัวเลข สำหรับปัญหา Regression และ Time series เนื่องจากใช้ Linear function จะดีกว่าเพราะว่าไม่ต้องแปลงกลับไปมา

งานวิจัยชิ้นนี้เลือกใช้ ฟังก์ชัน Hyperbolic tangent function เนื่องจากเป็นที่นิยมใช้ทั่วไป และให้ค่าพยากรณ์ที่มีความแม่นยำ (BekirKarlik and A. Vehbi Olgac, 2010)

2.2.9 เทคนิคการเรียนรู้

การนำข้อมูลเข้าสู่กระบวนการเทรน สามารถแบ่งได้เป็น 2 ลักษณะ คือ

1. Static training เป็นการนำข้อมูลเข้าเทรน ทีละแถวทุกตัวแปรในคราวเดียวพร้อมกัน ณ จุดเวลาเดียวกัน

2. Dynamic training เป็นการเทรนข้อมูลที่มีการนำค่า Output กลับเข้ามาเป็น Input อีกครั้ง เช่น ตัวแบบ NARX, Recurrent network

2.2.10 เทคนิคการปรับปรุงค่า Weights

เทคนิคการเทรนหรือเทคนิคการปรับค่า Weights สามารถแบ่งออกได้เป็น 3 รูปแบบ คือ

1. Incremental training (Online training) ค่า Weightsและ Biases ของ Network จะทำการปรับค่าทุกๆ ครั้ง ที่ข้อมูลชุดเทรน แต่ละแถวเข้าไปในตัวแบบ และจะทำการปรับค่า Weights ไปเรื่อยๆ จนกระทั่งครบทุกแถว ของข้อมูลเทรน จากนั้นก็ทำการปรับค่า Weights จนกระทั่งเข้าเงื่อนไขหยุด (Stopping rule) ซึ่งจะเร็วกว่าแบบ Batch training เมื่อข้อมูลมีจำนวนมาก และมีหลายตัวแปร

2. Batch training ค่า Weights และ Biases ของ Network จะทำการปรับค่าหลังจากที่ Inputเข้าไปใน Network ทั้งหมด Batch training มักนิยมใช้เนื่องจากจะทำการ Minimize total

error ในตอนหลังสุดทีเดียว แต่ก็ต้องทำการ ปรับเปลี่ยนค่า Weight (Update the weights) หลายครั้งจนกระทั่งเข้าสู่เงื่อนไขที่หยุดการใช้ batch training เหมาะกับข้อมูลขนาดเล็ก

3. Minibatch training ทำการแบ่งข้อมูลชุดเทรนเป็นกลุ่มๆ ขนาดเท่ากัน และทำการปรับค่า Weights หลังจากที่ข้อมูลในกลุ่มผ่านเข้าไปในตัวแบบ จะทำเช่นนี้จนกระทั่งครบทุกกลุ่ม จนเข้าเงื่อนไขหยุด ลักษณะการทำงานของ Minibatch นี้ เป็นการผสานจุดต่างระหว่าง Incremental และ Batch เอาไว้ด้วยกัน โดยเราสามารถกำหนดจำนวนข้อมูลเทรนภายในกลุ่มได้ ตามต้องการ ถ้าจำนวนข้อมูลภายในกลุ่มเป็น 1 ก็จะกลายเป็น Incremental ถ้าจำนวนข้อมูลเทรนภายในกลุ่มเท่ากับจำนวนข้อมูลทั้งหมดจะกลายเป็นแบบ Batch

2.2.11 อัลกอริทึม กระบวนการเรียนรู้ (Training Algorithm)

หรือเรียกว่า Optimization algorithm เป็นเทคนิคที่ใช้ในการเรียนรู้ คือทำการประมาณค่า Weights ในปัจจุบันมีเทคนิค Training algorithm เกิดขึ้นใหม่ๆ เป็นจำนวนมาก เทคนิคแรกเริ่มที่ยังคงใช้กันอยู่ในปัจจุบัน คือ Gradient descent algorithm ซึ่งมีส่วนประกอบที่สำคัญ คือ

1. Initial Learning Rate คือค่าอัตราการเรียนรู้แรกเริ่ม ยิ่งกำหนดให้ค่า learning rate สูงขึ้นจะทำให้โอกาสที่จะลู่เข้าสู่คำตอบ และเข้าสู่เงื่อนไขหยุด ก็มีมากขึ้นทำให้ตัวแบบใช้เวลาสร้างเร็วขึ้น แต่ก็ต้องแลกกับตัวแบบที่ได้อาจจะไม่ดี ให้ค่า Errors มาก

2.Lower Boundary of Learning Rate คือค่าขอบเขตล่างของอัตราการเรียนรู้ ค่านี้ใช้เฉพาะการเทรนแบบ Incremental หรือ Minibatch เท่านั้น ต้องกำหนดค่าให้มากกว่าหรือเท่ากับ 0 และน้อยกว่าค่า Initial learning rate

3. Momentum จะช่วยป้องกันตัวแบบลู่เข้าสู่เงื่อนไขหยุดการเรียนรู้ที่รวดเร็วเกินไป เนื่องจากการกำหนดค่า learning rate ที่มีค่ามาก เป็นเทอมที่คอยเบรก เพื่อยังให้ได้คำตอบที่น่าพอใจ มี Errors ต่ำนั่นเอง

เทคนิคการเรียนรู้อื่นๆ ได้แก่ Bayesian Regularization, BFGS Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell/Beale Restarts, Fletcher-Powell Conjugate Gradient, Polak-Ribiére Conjugate Gradient, One Step Secant, Variable Learning Rate Gradient Descent, Gradient Descent with Momentum, Gradient Descent และ Levenberg-Marquardt

เทคนิคการเรียนรู้ที่เร็วที่สุด คือ Levenberg-Marquardt โดยมีเทคนิค BFGS Quasi-Newton เร็วถัดไป ทั้งสองเทคนิคนิยมใช้กันในตัวแบบ Feedforward ซึ่งมีความรวดเร็วกว่าเทคนิค Gradient descent ทั้งสองเทคนิคที่กล่าวมานั้นมีประสิทธิภาพสูง โดยเฉพาะข้อมูลที่ใช้ไม่มากนัก แต่หากว่าข้อมูลมีจำนวนมาก มีหลายตัวแปร ก่อให้เกิดจำนวน Weights จำนวนมาก อาจถึงหลักพัน ก็จะใช้เวลาที่นาน

งานวิจัยชิ้นนี้เลือกใช้เทคนิคการเรียนรู้ Levenberg-Marquardt เนื่องจากใช้เวลาในการลู่เข้าสู่คำตอบที่เร็ว และสามารถหาค่า Local minima ได้ดี และไม่จำเป็นต้องระบุค่า Learning rate และค่า Momentum factor ดังเช่น เทคนิค BP เป็นที่นิยมใช้ในงานการหาค่าเหมาะสมที่ดีที่สุดสำหรับข้อมูลไม่เป็นเส้นตรง (Nonlinear optimization)

เทคนิค Levenberg-Marquardt นั้นเหมาะกับปัญหา Regression มากกว่าปัญหาที่เป็นการพยากรณ์เชิงกลุ่ม (Pattern recognition) ในกรณีที่ตัวแบบมีเครือข่ายขนาดใหญ่ และปัญหา Pattern recognition เทคนิค Resilient Backpropagation และ Scaled Conjugate Gradient จะหาคำตอบได้รวดเร็วกว่า

2.2.12 จำนวนรอบการเทรนซ้ำ (Experiment repeat)

เนื่องจาก ANN มีการหาค่าต่ำสุดของค่า Training Errors เช่นMean Square Error (MSE) ด้วยการหาอนุพันธ์ ซึ่งอาจเกิดปัญหาค่าอนุพันธ์ที่เป็น 0 นั้น อาจเป็นค่าต่ำสุดในช่วงๆ หนึ่งเท่านั้น เรียกว่า Local minima ไม่ใช่ค่าต่ำสุดของ MSE ทั้งข้อมูลชุดทดสอบเรียกว่า Global minima จุดนี้จึงเป็นปัญหาของ ANN แม้ว่าจะไม่สามารถยืนยันว่า MSE ที่ต่ำสุดนั้นมาจาก Global minima แต่ก็สามารถลดความเสี่ยงได้ โดยการรัน ANN ซ้ำๆ

ในการเทรน ANN แต่ละรอบ จะมีค่า Weights เริ่มต้น หรือเรียกว่า Initial weights ที่เปลี่ยนไปทุกๆรอบ จะทำให้ผลการพยากรณ์ในแต่ละรอบต่างกัน ใช้เวลาในการเทรนเพื่อเข้าสู่เงื่อนไขหยุดต่างกัน ภายใต้โครงสร้างตัวแบบเดียวกัน เช่น จำนวน Input neurons=4 , จำนวน Hidden neurons=4, จำนวน Output neurons=1 คือ ค่า weight เริ่มต้น เป็นปัจจัยที่ส่งผลกระทบต่อความแม่นยำของตัวแบบ ซึ่งเป็นการยากที่จะหาค่า Weights ที่ถูกต้องเหมาะสม นอกจากนี้จำนวนค่า Weight เริ่มต้น ก็ยังไม่แน่นอนขึ้นอยู่กับจำนวน Input neurons และ จำนวน Hidden neurons

จำนวนรอบการทำซ้ำของ ANN ภายใต้โครงสร้างตัวแบบเดียวกัน ปัจจุบันยังไม่มีวิธีการหาจำนวนรอบการทำซ้ำ ที่ชัดเจน การกำหนดจำนวนรอบของการทำซ้ำมากไป อาจใช้เวลาในการคำนวณมาก ผู้วิจัยหลายๆ ท่านมักกำหนด จำนวนรอบตามความสะดวก เช่น 8, 10, 20 หรือ 50 รอบ เป็นต้น งานวิจัยชิ้นนี้กำหนด จำนวนรอบการทำซ้ำเท่ากับ 50 รอบ เพื่อเพิ่มโอกาสที่จะได้ตัวแบบ ANN ที่มีความแม่นยำ

2.2.13 เงื่อนไขหยุด (Stopping Criteria)

คือ เงื่อนไขที่ตัวแบบหยุดเรียนรู้ มีดังนี้

1. Training iterations criteria หรือจำนวน Epoch คือจำนวนรอบการปรับค่า weight หรือค่าสัมประสิทธิ์ของข้อมูล ค่าจำนวนรอบนี้เป็นค่าที่กำหนดขึ้นมาเองหรืออาจกำหนดให้โปรแกรมคำนวณค่านี้โดยอัตโนมัติได้ กรณีเช่นนี้จำเป็นต้องกำหนดเงื่อนไขอื่นประกอบ เช่นเวลาในการรัน ค่า Error ที่ไม่ลดลงกี่ครั้ง จึงหยุดทำงาน จำนวน Epoch ที่เหมาะสมนั้นขึ้นอยู่กับเทคการเรียนรู้ที่เลือกใช้ บางเทคนิคมีการเรียนรู้ที่นาน จะต้องใช้จำนวน Epoch มาก บางเทคนิคมีการเรียนรู้ที่สั้น ก็มีจำนวน Epoch น้อย อย่างเช่นเทคนิค Levenberg-Marquardt, BFGS Quasi-Newton และ Scaled Conjugate Gradient เป็นเทคนิคที่ใช้เวลาในการเรียนรู้เร็ว จึงมีจำนวน Epoch น้อย หรือเทคนิค Gradient Descent เป็นเทคนิคแบบเดิม ใช้เวลาการเรียนรู้มาก จึงมีจำนวน Epoch มากในโปรแกรมสำเร็จรูปเช่น Matlab neural network tool box มักจะกำหนดจำนวน Epoch เอาไว้มากพอแล้วเสมือนเป็นจำนวนตั้งต้น (Default) ของตัวโปรแกรม แต่ผู้วิเคราะห์ก็ต้องพิจารณาเองว่าค่าพยากรณ์

ที่ได้นั้นอยู่ในระดับที่น่าพอใจแล้วหรือยัง ถ้ายังไม่พอใจก็อาจลองเพิ่มจำนวน Epoch ให้มากขึ้น สิ่งเหล่านี้ก็ควรจะกำหนดค่าพารามิเตอร์ให้ตายตัวเสียก่อน เพื่อจะได้ศึกษาผลของการเพิ่มจำนวน Epoch ที่มีต่อค่า Errors

2. Relative change in training errors ตัวแบบจะหยุดการเรียนรู้เมื่อผลต่างค่า Training errors ใน Epoch ก่อนหน้า 1 ค่ากับค่า Training errors ใน Epoch ปัจจุบัน มีค่าน้อยกว่าค่าที่กำหนด เช่น 0.000001 เป็นต้น เขียนเป็นสมการได้ดังนี้

$$|Training\ errors \text{ ณ } epoch \text{ ก่อนหน้า} - Training\ errors \text{ ณ} epoch \text{ ปัจจุบัน}| = \text{ค่าที่กำหนด} \quad (2.10)$$

3. Maximum steps without a decrease in errors หมายถึงจำนวนครั้งที่มีการปรับค่า Weights แล้วค่า Errors ในข้อมูลชุดเรียนรู้ไม่ลดลง ค่านี้เป็นค่าที่กำหนดเอง เช่นถ้ากำหนดเป็น 1 หมายความว่าค่า Errors นี้จะต้องลดลงไปเรื่อยๆ แบบติดต่อกัน ถ้าเพิ่มขึ้นเมื่อไหร่ ตัวแบบจะหยุดการเรียนรู้ทันที หรือกำหนดเป็น 6 หมายความว่าค่า Errors นี้จะต้องลดลง หรือแม้ว่าหากเพิ่มขึ้นมาบ้างก็จะต้องไม่เพิ่มขึ้นเกิน 6 ครั้งแบบติดต่อกัน ถ้าเกิน 6 ครั้งเมื่อไหร่ ตัวแบบจะหยุดการเรียนรู้ทันที

4. ระยะเวลาในการเรียนรู้ เป็นสิ่งที่ไม่เกี่ยวข้องกับตัวแบบในทางตรง แต่เป็นสิ่งที่โปรแกรมสำเร็จรูปหลายโปรแกรมนำเข้ามา เพื่อเป็นตัวป้องกันการเรียนรู้ที่นานเกินไป อาจเนื่องจากการปรับแต่งโครงสร้างของตัวแบบหรือการลดความเข้มงวดของเงื่อนไขหยุดการเรียนรู้ เปรียบเสมือนตัวช่วยเตือนผู้วิเคราะห์ ไม่ให้เสียเวลา แต่ทั้งนี้ทั้งนั้นถ้าผู้วิเคราะห์ต้องการศึกษาผลกระทบของแต่ละพารามิเตอร์ หรือเงื่อนไขหยุดบางประการ ซึ่งทำให้ต้องใช้เวลาในการเรียนรู้ที่นานก็สามารถขยายหรือยกเลิกเงื่อนไขเวลาที่ใช้

2.3 ตัวแบบ ANN สำหรับการพยากรณ์อนุกรมเวลา (ANN for time series forecasting)

ตัวแบบ ANN สามารถประยุกต์ใช้การพยากรณ์อนุกรมเวลาได้ ส่วนใหญ่นิยมใช้พยากรณ์ข้อมูลอนุกรมเวลาชุดเดียว ตัวแปรเดี่ยวๆ เช่นพยากรณ์ยอดขายสินค้า พยากรณ์ความต้องการใช้พลังงานทั้งประเทศ พยากรณ์ราคาหุ้น ราคาทองคำ เป็นต้น โดยที่ใช้ตัวแปรที่ต้องการพยากรณ์นั้นเพียงตัวแปรเดียว ไม่ใช้ตัวแปรอิสระอื่นๆ เพื่อมาพยากรณ์ ซึ่งเรียกว่า Univariate time series

การพยากรณ์ข้อมูลอนุกรมเวลา มีเป้าหมาย คือการพยากรณ์ข้อมูลล่วงหน้า โดยใช้ข้อมูลเก่าในอดีตของข้อมูลชุดนั้นเพียงอย่างเดียว มาพยากรณ์ค่าในอนาคต โดยไม่ใช้ตัวแปรอื่นๆ มาวิเคราะห์ โดยส่วนมากการพยากรณ์อนุกรมเวลา จะเกี่ยวข้องกับตัวเลข ปัญหานี้จึงคล้ายคลึงกับปัญหา Regression

การปรับเปลี่ยนปัญหาการพยากรณ์แบบ Regression ให้เป็นการพยากรณ์อนุกรมเวลา ที่สำคัญ คือ การเปลี่ยนตัวแปรอิสระ ดังเช่นตัวอย่าง ต้องการพยากรณ์ยอดขายสินค้าชนิดหนึ่ง จากเดิมที่มีตัวแปรอิสระหลายๆ ตัว แล้วนำมาเทรนตัวแบบ แต่สำหรับอนุกรมเวลา จะใช้ข้อมูลตัวแปร

ยอดขายสินค้า เพียงตัวแปรเดียว โดยนำตัวแปรยอดขายมาแบ่งเป็นตัวแปรย่อย ณ เวลาย้อนหลัง ใช้เป็นตัวแปรอิสระในการพยากรณ์ รูปแบบการแปลงตัวแปรที่จะพยากรณ์เป็นตัวแปรอิสระนั้น เป็นไปตามภาพ 2.12

Input1	Input2	Input3	(Output) Predicted value	Real observation
Y1	Y2	Y3	$\widehat{Y_4}$	Y4
Y2	Y3	Y4	$\widehat{Y_5}$	Y5
Y3	Y4	Y5	$\widehat{Y_6}$	Y6
Y4	Y5	Y6	$\widehat{Y_7}$	Y7
....	Training set	....	....	....
....		....	....	....
....	....	....	....	....
Y101	Y102	Y103	$\widehat{Y_{104}}$	Y104
Y105	Y106	Y107	$\widehat{Y_{107}}$	Y107
....	Testing set	....	....	....
Y130	Y131	Y132	Y133	Y134

ภาพที่ 2.12 แผนผังตัวแปรอิสระและตัวแปรตามสำหรับตัวแบบ ANN ในการพยากรณ์ข้อมูลอนุกรมเวลา

เช่นต้องการพยากรณ์ยอดขายที่เวลา Y_{t+1} โดยรู้ข้อมูลยอดขายตั้งแต่ $Y_1, Y_2, Y_3, \dots, Y_t$ จากนั้นกำหนดจำนวนจุดเวลาย้อนหลังมาทำเป็นตัวแปร (Lag variable) เช่นกำหนดจำนวน Lag เป็น 1 นั่นคือ ทำการพยากรณ์ Y_{t+1} ด้วย Y_t หรือกำหนดจำนวนLag เป็น 2 นั่นคือทำการพยากรณ์ Y_{t+1}ด้วย Y_t และ Y_{t-1} หรือกำหนดจำนวน Lag เป็น 3 นั่นคือทำการพยากรณ์ Y_{t+1} ด้วย Y_t, Y_{t-1}และ Y_{t-2} และเมื่อจำนวน Lag เพิ่มขึ้น ก็ทำเช่นเดียวกันนี้ไปเรื่อยๆ ตารางข้างบน แสดงการพยากรณ์ข้อมูล เมื่อกำหนดจำนวน Lag เป็น 3 จะมี Input 3 ตัว นำเข้าเพื่อทำการเทรน และจะได้ค่าพยากรณ์ 1 ตัว ทุกๆ ชุดที่ ใส่ Input เข้าไป

จำนวน Lag ที่กำหนดนั้น ปัจจุบันยังไม่มีกฎเกณฑ์ตายตัวว่าจะต้องเป็นเท่าไหร่ จึงจะได้คำตอบที่ดีที่สุด ขึ้นอยู่กับ ความซับซ้อนของข้อมูล ความแปรผันของข้อมูล ลักษณะเฉพาะของข้อมูล เช่น มีแนวโน้ม มีปัจจัยเชิงฤดูกาล มีการส่ายไปมาไร้รูปแบบ (Noise) จำเป็นต้องทำการทดลอง เริ่มจากกำหนดจำนวน Lag เป็น 1 เพิ่มไปเรื่อยๆ ทีละ 1 จนกระทั่ง Lag มีค่าเป็น 2,3,4,5,... แต่ส่วนใหญ่มักมีค่าไม่มาก อยู่ใน 1-10 เนื่องจากยิ่งกำหนดจำนวน Lag ยิ่งมาก ความสัมพันธ์ระหว่างข้อมูล ณ จุดเวลา Lag (Autocorrelation) นั้นจะน้อย และจะมีความสัมพันธ์กันเองระหว่าง Lag อื่นๆ มาก ยิ่ง

ทำให้ใช้เวลาในการเทรนนาน และทำให้ได้ผลลัพธ์ที่แย่ สำหรับข้อมูลอนุกรมอนุกรมเวลา ที่มีปัจจัยเชิงฤดูกาล เช่น ข้อมูลปริมาณการใช้ไฟฟ้ารายเดือน จะมีรอบของฤดูกาล ประมาณ 11,12,13 คือ ปริมาณการใช้ไฟฟ้ามากที่สุดในเดือน เมษายน ปี 2554 ในปี 2555 มากที่สุด ในเดือน เมษายน และในปี 2556 มากที่สุดในเดือน พฤษภาคม การกำหนดจำนวน Lag ให้คลุมรอบของปัจจัยฤดูกาลนี้ จะมีโอกาสเพิ่มความแม่นยำในการพยากรณ์ให้สูงขึ้น

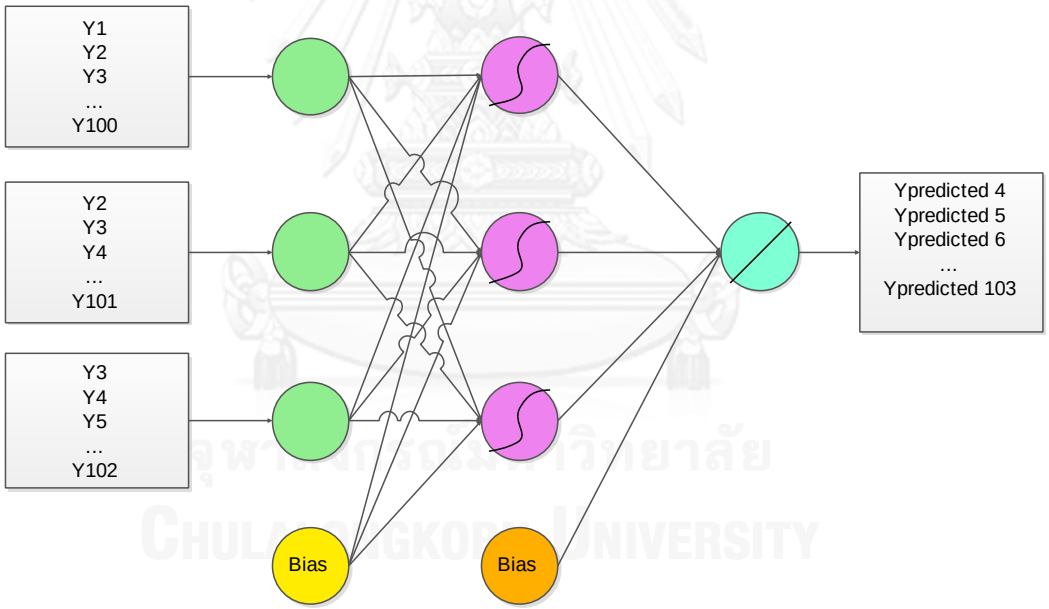
จากตารางข้างบน จำนวนค่าพยากรณ์ มีจำนวนน้อยกว่าข้อมูลจริง ดังความสัมพันธ์

จำนวนค่าพยากรณ์ = จำนวนข้อมูลจริง – จำนวน Lag ที่เลือกใช้ (2.11)

ดังนั้น ถ้าข้อมูลที่ใช้มีจำนวนน้อยการกำหนดจำนวน Lag ยิ่งมาก ยิ่งทำให้จำนวนข้อมูลลดลง ถ้าเรามีข้อมูลอนุกรมเวลา 130 ตัว กำหนดจำนวน Lag 20 ตัว นั่นจะทำให้จำนวนข้อมูลลดลงเหลือ 110 ตัว

2.3.1 โครงสร้างตัวแบบ ANN สำหรับการพยากรณ์อนุกรมเวลา (ANN for time series architecture)

รูปแบบเครือข่ายของตัวแบบ ANN ที่ใช้ในการพยากรณ์อนุกรมเวลาจะคล้ายกับตัวแบบ MLP แต่จะมีการปรับเปลี่ยนตัวแปรนำเข้า ดังภาพ 2.13



ภาพที่ 2.13 แผนผังของตัวแบบ Feedforward network ในการพยากรณ์ข้อมูลอนุกรมเวลา

สอดคล้องกับตารางด้านบน เส้นที่แสดงเป็นตัว S ใน Hidden neurons หมายถึง Hyperbolic tangent transfer function ส่วนเส้นตรงแนวทแยง ใน Output neuron หมายถึง Linear transfer function ซึ่งทั้งสองเป็นฟังก์ชัน Activation ที่เลือกใช้ในงานวิจัยชิ้นนี้

สามารถเขียนเป็นเป็นความสัมพันธ์ในรูปแบบสมการ ดังนี้

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, y_{t-3}, \cdots, y_{t-n}) + \varepsilon_t \quad (2.12)$$

- คือ ค่าพยากรณ์ที่เวลา t
- คือ ค่าจริงที่เวลา t-1
- คือ ค่าจริงที่เวลา t-2
- คือ ค่าจริงที่เวลา t-3
- คือ ค่าจริงที่เวลา t-n ,โดยที่ n คือจำนวน Lag ช่วงห่างเวลาที่นำเข้าสู่ตัวแบบ
- คือ ค่า Residual ที่เวลา t
- สามารถเรียกรูปแบบ Neural network ในลักษณะนี้ว่า Nonlinear Autoregressive หรือ NAR เนื่องจากนำค่าในอดีตมาพยากรณ์ค่าในอนาคต คล้ายกับ ARIMA เพียงแต่เป็นตัวแบบไม่เป็นเชิงเส้นตรง

ส่วนประกอบอื่นๆ ของโครงข่าย ทั้งจำนวน Hidden layer, Output neuron, Activation function, Training algorithm จะเหมือนกับตัวแบบ Neural networks สำหรับการพยากรณ์ Regression

จำนวนค่าพยากรณ์ล่วงหน้า สามารถแบ่งได้เป็น 2 แบบ ใหญ่ คือ

1.การพยากรณ์1 ค่าล่วงหน้า (One step ahead forecasting) ทำการพยากรณ์ล่วงหน้า 1 ค่าล่วงหน้า ถ้าต้องการพยากรณ์ข้อมูลในวันพรุ่งนี้ ต้องมีข้อมูลในวันนี้

2.การพยากรณ์ล่วงหน้า k ค่าล่วงหน้า (K steps ahead forecasting) ทำการพยากรณ์ล่วงหน้า k ค่าล่วงหน้าเช่นพยากรณ์ 5 หรือ 10 หรือ 50 ค่าล่วงหน้า

การพยากรณ์ล่วงหน้า 1 ค่าจะได้เปรียบ มีความแม่นยำในการพยากรณ์สูงกว่าการพยากรณ์หลายค่าล่วงหน้า งานวิจัยชิ้นนี้จะพุ่งเป้าเน้นไปที่พยากรณ์ 1 ค่าล่วงหน้า เท่านั้น เนื่องจากต้องการเปรียบเทียบประสิทธิภาพตัวแบบพยากรณ์อนุกรมเวลา 6 ตัวแบบ จึงต้องการเน้นไปที่ความแม่นยำที่สูงสุดของแต่ละตัวแบบเป็นหลัก และการพยากรณ์อนุกรมเวลาล่วงหน้ามากจะทำให้ความผิดพลาดในการพยากรณ์ (Cumulative errors) มีค่าสูงมากยิ่งขึ้น (Pai and Lin, 2005)

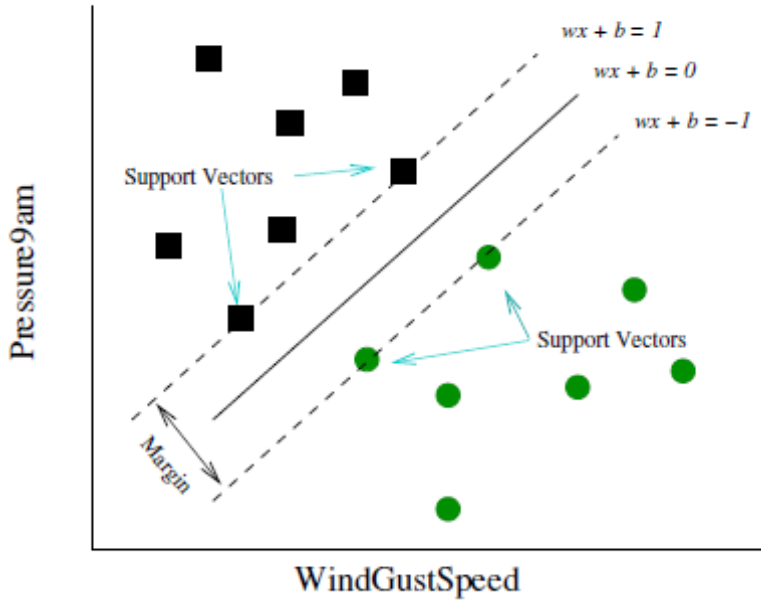
2.4 ตัวแบบ Support vector machine (SVM)

Support vector machine เป็นตัวแบบพยากรณ์ จัดอยู่ในกลุ่ม Machine learning ที่เริ่มเกิดขึ้นเมื่อปี 1995 โดย Vapnik (Vapnik V, 1995) เป็นตัวแบบที่ได้รับความสนใจจากผู้วิจัย เนื่องจากให้ผลการพยากรณ์ที่ดี มีความแม่นยำสูงตัวแบบ SVM มีลักษณะคล้ายกับตัวแบบ ANN ตรงที่สามารถนำไปประยุกต์ใช้พยากรณ์ข้อมูลแบบเดียวกัน ทั้ง การพยากรณ์เชิงกลุ่ม (Pattern recognition หรือ Classification) แต่แตกต่างกันตรงที่หลักการ Minimization กล่าวคือ Support vector machine จะทำให้ความเสี่ยงเชิงโครงสร้างต่ำสุด (Structural Risk Minimization:SRM) ขณะที่Neural networksใช้หลักการลดความเสี่ยงเชิงทดลองให้ต่ำสุด (Empirical Risk Minimization: ERM)

ERM จะพุ่งเน้นพยายามทำให้กระบวนการเทรน ในข้อมูลชุด Training เกิด Error ต่ำที่สุด นั่นคือ โครงสร้างของตัวแบบ ANN เช่น มีหลายจำนวน Hidden layer หรือ มีจำนวน Hidden neurons

หลายตัว อาจทำให้ตัวแบบเรียนรู้ดีเกินไปในข้อมูลชุด Training นั่นอาจทำให้ตัวแบบพยากรณ์ไม่แม่นยำในข้อมูลชุด Test ได้ เรียกปัญหานี้ว่า Overfitting แต่ SRM จะมีเทอมหรือฟังก์ชันที่เรียกว่า Regularization penalty กำหนดขอบเขตมาคอยควบคุม เสมือนเป็นตัวเบรก ไม่ให้ตัวแบบเรียนรู้ดีเกินไป ซึ่งก็เป็นการป้องกันการเกิดปัญหา Overfitting

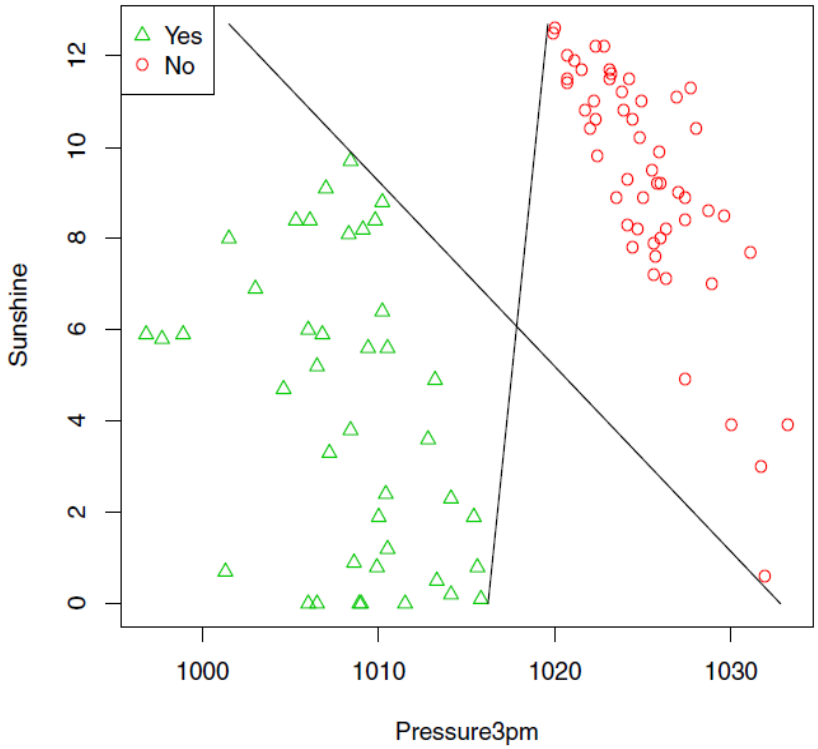
หลักการของ SVM มาจากการหาระนาบหรือเส้นแบ่งในกรณีหลายๆ มิติ เรียกว่า Hyperplane ที่แบ่งกลุ่มข้อมูลหรือเรียกว่าแบ่ง Class โดยมีระยะห่างระหว่าง 2 กลุ่มมากที่สุด ระยะห่างนั้น เรียกว่า Margin ดังภาพ 2.14



ภาพที่ 2.14 เส้นตรงแบ่งกลุ่มข้อมูลที่มี 2 ตัวแปร สำหรับตัวแบบ SVM (Williams, 2011)

ตัวอย่างการหา Hyperplane สำหรับตัวแบบ SVM ที่มี 2 ตัวแปร หรือ 2 มิติ คือมีตัวแปรอิสระและตัวแปรตามอย่างละตัว ตัวแปรตามมีสองกลุ่ม คือ ที่เป็นรูปสี่เหลี่ยมและรูปวงกลม ดังภาพ 2.14

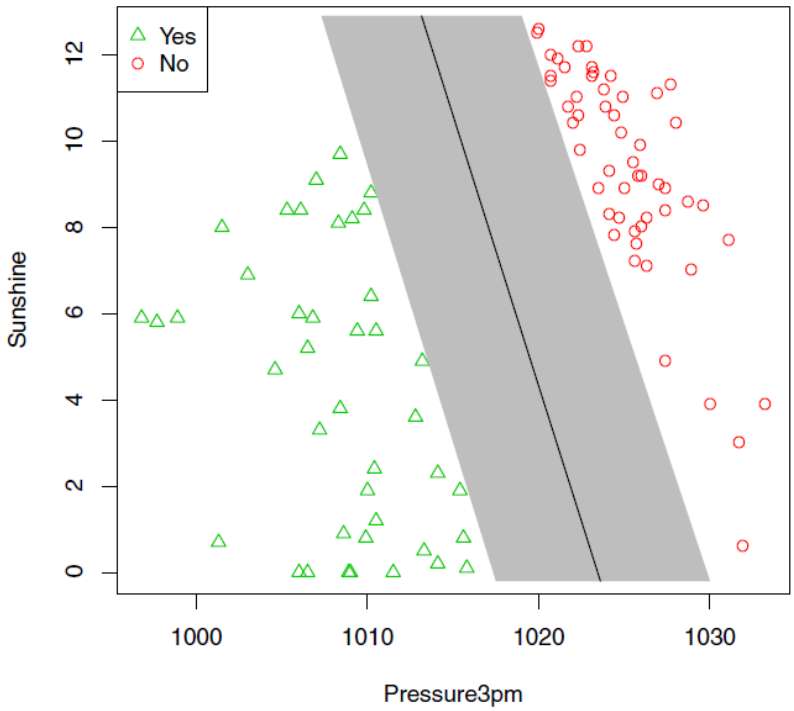
Chulalongkorn University



ภาพที่ 2.15 เส้นตรงที่ใช้แบ่งกลุ่มข้อมูลที่เป็นไปได้ สำหรับตัวแบบ SVM (Williams, 2011)

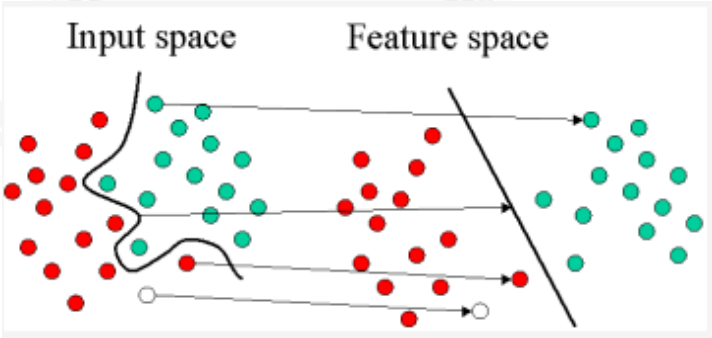
จากภาพ 2.15 จะเห็นได้ว่ามีเส้นตรงหลายเส้นที่สามารถแบ่งข้อมูลเป็นกลุ่มข้อมูลเป็น 2 Class แต่ SVM จะค้นหาเส้นตรงที่แบ่งกลุ่มข้อมูลโดยให้มีระยะ Margin มากที่สุด ซึ่งนั่นคือมีพื้นที่ (Space) ระหว่างกลุ่มมากที่สุด ดังภาพด้านล่างคือพื้นที่สีเทา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 2.16 เส้นตรงที่ใช้แบ่งกลุ่มที่ดีที่สุด (Williams, 2011)

เมื่อข้อมูลที่ซับซ้อน ยากที่จะหาเส้นตรงมาแบ่งกลุ่มได้ หรือเมื่อข้อมูลใหญ่ขึ้นมีตัวแปรอิสระหลายตัวหรือข้อมูลมีหลายมิติก็ใช้หลักการเดียวกันในการหา Hyperplane ที่ดีที่สุด แต่ยิ่งแทบเป็นไปไม่ได้เลยที่จะหา Hyperplane แนวตรง กรณีเช่นนี้ ต้องใช้การแปลงข้อมูล โดยใช้ Kernel function หรือเรียกว่า Kernel trick จะแปลงจุดพิกัดให้มีการจัดเรียงใหม่ (Mapping) เพื่อให้สามารถค้นหา Hyperplane แนวเส้นตรงได้ดังภาพด้านล่าง



ภาพที่ 2.17 รูปแบบการแปลงจุด เพื่อให้สามารถหาเส้นตรงมาแบ่ง

2.4.1 Support vector machine for regression

ฟังก์ชันเป็นดังนี้

$$y = w\emptyset(x) + b \tag{2.13}$$

โดยที่ $\emptyset(x)$ เรียกว่า dimensional feature space ซึ่งเป็น nonlinear mapped จากinput space x ค่า w และ b คือ สัมประสิทธิ์ หาได้จาก การ Minimize

$$R(C) = C\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon\,(d_i, y_i) + \frac{1}{2}\|w\|^2 \tag{2.14}$$

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon & |d-y| \geq \varepsilon, \\ 0 & others, \end{cases} \tag{2.15}$$

สมการที่ (2.14) $C\frac{1}{N}\sum_{i=1}^{N} L_\varepsilon\,(d_i, y_i)$ เรียกว่า Empirical error(risk) หา ε - insensitive loss function จากสมการที่ (2.15) เทอมThe $\frac{1}{2}\|w\|^2$ ในสมการที่ (2.14) คือ Regularization term ซึ่งจะหาค่า Flatness ของฟังก์ชัน ขณะที่ C > 0 ทำหน้าที่เป็นตัว Trade-off ระหว่าง Empirical risk และ Flatness ของตัวแบบ การ Trade-off ตรงนี้ทำให้ตัวแบบมีความคงทนต่อ Outliers นั่นทำให้ SVM แตกต่างจากเทคนิค Traditional error minimization

การเพิ่มขึ้นของ C จะทำให้ Empirical risk เพิ่มขึ้นเมื่อเทียบกับ Regularization term

การที่จะได้ค่า w และ b ในสมการ (2.14) ต้องทำการแปลงเป็น Primal function ในสมการที่โดยการเพิ่มตัวแปร Slack ที่เป็นบวกคือ ξ_i และ ξ_i^* ซึ่งแสดงระยะห่างระหว่างค่าจริงและค่าขอบเขตภายใต้ $\boldsymbol{\varepsilon}$–tube จะได้ชุดของสมการที่เป็นลักษณะ Quadratic programming ดังนี้

Minimize :

$$R(w, \xi, \xi^*) = \frac{1}{2}ww^T + C^*\left(\sum_{i=1}^{N}(\xi_i + \xi^*{}_i)\right) \tag{2.16}$$

$$w\phi(x_i) + b_i - d_i \leq \varepsilon + \xi_i^*, \tag{2.17}$$

$$d_i - w\phi(x_i) - b_i \leq \varepsilon + \xi_i^*, \tag{2.18}$$

$$\xi_i, \xi_i^* \geq 0 \tag{2.19}$$

$$i = 1,2, \ldots, N \tag{2.20}$$

ขั้นตอนสุดท้ายจะนำตัวคูณ Lagrange และทำการ Maximizing dual function ในสมการที่ (2.16) ซึ่งจะเปลี่ยนแปลงไปเป็นดั่งสมการที่ (2.21)

$$R(\alpha_i - \alpha_i^*) = \sum_{i=1}^{N} d_i\,(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right)K(x_i, x_j) \tag{2.21}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

subjected to:

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \quad (2.22)$$

$$0 \le \alpha_i \le C, \quad (2.23)$$

$$0 \le \alpha_i^* \le C, \quad (2.24)$$

$$i = 1,2,\dots,N. \quad (2.25)$$

ในสมการที่ (2.21) ตัว α_iและ α_i^* เรียกว่าตัวคูณ Lagrange โดยที่

$$\alpha_i * \alpha_i^* = 0,$$

$$f(x,\alpha,\alpha^*) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x,x_i) + b \quad (2.26)$$

สมการที่ (2.21) เป็นจุดที่สำคัญในการแก้ปัญหา Nonlinear SVM เนื่องจากมีเทอม $K(x,x_i)$ เรียกว่า Kernel function หรือเรียกว่า ‘‘kernel trick” (Vapnik V, 1995) สมการรูปแบบของ Kernel function นี้คือ dot product $K(x_i,x_j) = \emptyset(x_i) * \emptyset(x_j)$ หน้าที่ของ Kernel function คือ การแปลงข้อมูลจาก Input space ไปสู่ High dimensional feature space ใช้ทั้งการพยากรณ์ทั้ง Classification และ Regression เช่นเดียวกัน

2.4.2 Kernel function

Kernel function เป็นฟังก์ชันที่มีอยู่ในรูปแบบ Inner product ระหว่างจุดสองจุด ใน Feature space มีหน้าที่หลักคือทำให้ SVM สามารถแก้ปัญหาที่ไม่เป็นเชิงเส้นตรง (Nonlinear) ได้ ซึ่ง Kernel function สามารถมีได้หลายรูปแบบ ดังเช่น

1.Linear kernel $K(x_i,x_j) = \langle x_i, x_j \rangle$

2.Gaussian radial basis kernel $K(x_i,x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

3.Polynomial kernel $K(x_i,x_j) = (s\langle x_i, x_j \rangle + c)^d$

4. Sigmoid kernel $K(x_i,x_j) = \tanh(s\langle x_i, x_j \rangle + c)$

5. Convex combination of kernels

$$K(x_i,x_j) = \lambda_1 k_1(x_i,x_j) + \lambda_2 k_2(x_i,x_j)$$

6. Normalization kernel $K(x_i,x_j) = \frac{k'(x_i,x_j)}{k'(x_i,x_i)k'(x_j,x_j)}$

โดยที่ s, c, d และ λ_i เป็น Parameter สำหรับ Kernel ในแต่ละฟังก์ชัน

จนกระทั่งถึงปัจจุบันนี้ ยังไม่มีกฎเกณฑ์ตายตัวในการเลือกชนิดของ Kernel function ที่ดีที่สุดสำหรับทุกชุดข้อมูล (Hong et al., 2010)

งานวิจัยชิ้นนี้เลือก Gaussian kernel function เนื่องจากง่ายต่อการนำไปใช้เนื่องจากมีพารามิเตอร์ที่ต้องใส่ค่าตัวเดียว คือ σ และมีประสิทธิภาพที่ดีในการแปลง Nonlinear mapping จาก Input space ไปสู่ High dimensional feature space (Chen and Wang, 2007)

ดังนั้นการพยากรณ์ SVM โดยเลือกใช้ Gaussian kernel function จึงมี 3 พารามิเตอร์ที่ต้องใส่ค่า ได้แก่ ε, C และ σ

2.4.3 ตัวแบบ Support vector machine สำหรับพยากรณ์ข้อมูลอนุกรมเวลา

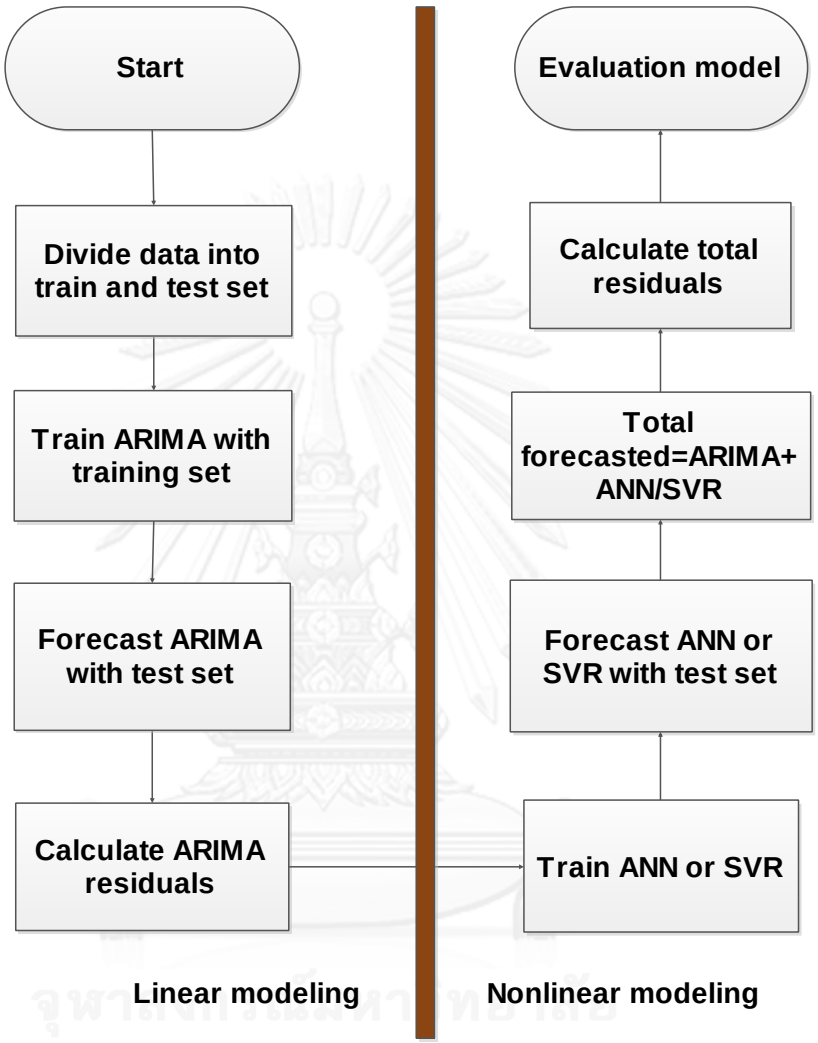
การพยากรณ์ข้อมูลอนุกรมเวลาด้วย SVM นั้นก็เหมือนกันกับการพยากรณ์ปัญหา Regression ที่มีตัวแปรอิสระและตัวแปรตาม แต่ต้องปรับข้อมูลอนุกรมเวลา ให้เป็นตัวแปร Lag ซึ่งกระบวนการทำก็เหมือนกับการพยากรณ์ด้วยตัวแบบ ANN ดังที่อธิบายในหัวข้อ 2.3

การเลือกจำนวน Lag ที่ใช้ในตัวแบบ ก็ไม่มีกฎเกณฑ์ตายตัว จึงทำการเลือก ตั้งแต่ 1 ถึง10 โดยสรุปมีพารามิเตอร์จำนวน 4 ตัวที่ผู้ใช้ต้องใส่ค่าลงไปในตัวแบบและทำการปรับจูนค่าจนกระทั่งได้ตัวแบบที่ให้ผลการพยากรณ์ที่น่าพอใจ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.5 ขั้นตอนการสร้างตัวแบบ Traditional hybrid

แผนผังขั้นตอนของตัวแบบผสมชนิดนี้เป็นดังภาพที่ 2.18



ภาพที่ 2.18 ขั้นตอนการสร้างตัวแบบ Traditional hybrid

ข้อกำหนดตัวแบบผสมชนิดนี้ คือมองว่าข้อมูลก่อนนำเข้าสู่ตัวแบบ สามารถแบ่งได้เป็น 2 องค์ประกอบ คือ องค์ประกอบที่เป็นเชิงเส้นตรง (Linear component) และองค์ประกอบที่ไม่เป็นเชิงเส้นตรง (Nonlinear component) และกำหนดให้ความสัมพันธ์ระหว่างองค์ประกอบทั้งสองเป็นบวก (Additive)

วิธีการ Traditional hybrid มีขั้นตอนดังนี้

1) ข้อมูลถูกแบ่งออกเป็นสองส่วน คือ Linear และ Nonlinear แสดงความสัมพันธ์ได้ดังนี้

$$Y_t = L_t + N_t \tag{2.27}$$

L_t = ส่วนประกอบที่เป็น Linear ทำการพยากรณ์ด้วยตัวแบบ ARIMA
N_t = ส่วนประกอบที่เป็น Nonlinear ทำการพยากรณ์ด้วยตัวแบบ ANN หรือ SVR

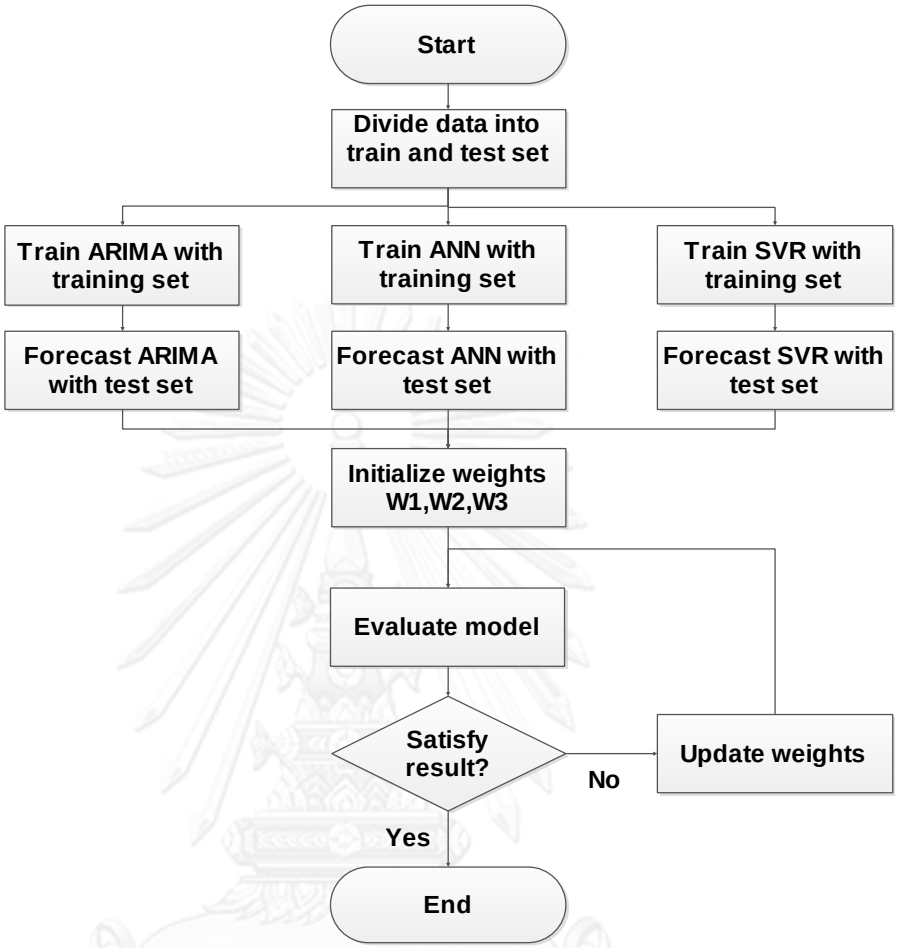
2) ทำการพยากรณ์ ARIMA จะได้ ค่าพยากรณ์ $\hat{L}_t$
3) หาค่าเศษเหลือจาก ARIMA ดังนี้ $Residuals = \varepsilon_t = Y_t - \hat{L}_t$
4) ค่า Residuals ที่ได้จาก ARIMA ถูกจัดให้เป็นส่วนประกอบ Nonlinear (N_t)ทำการพยากรณ์ต่อในตัวแบบ ANN หรือ SVR จะได้ค่าพยากรณ์ Residuals หรือเรียกค่าพยากรณ์ส่วน Nonlinear $(\hat{N}_t)$
5) ได้ค่าพยากรณ์รวม คือ $\hat{Y}_t = \hat{L}_t + \hat{N}_t$
6) นำค่าจริง ลบด้วยค่าพยากรณ์รวม จะได้ค่าเศษเหลือทั้งหมด $e_t = Y_t - \hat{Y}_t$
7) นำค่าที่ได้จาก 6) มาวิเคราะห์ ความแม่นยำของตัวแบบ

2.6 ขั้นตอนการสร้างตัวแบบ Hybrid combined

หลักการของตัวแบบผสมชนิดนี้ คือ คือมองว่าข้อมูลก่อนนำเข้าสู่ตัวแบบ สามารถแบ่งได้เป็น 2 องค์ประกอบ คือ องค์ประกอบที่เป็นเชิงเส้นตรง (Linear component) และองค์ประกอบที่ไม่เป็นเชิงเส้นตรง (Nonlinear component) เหมือนกับตัวแบบ Traditional hybrid แต่สร้างตัวแบบพยากรณ์แต่ละตัวแยกกันไป แล้วนำค่าพยากรณ์มารวมกัน โดยจัดให้อยู่ในรูปแบบผลรวมเชิงเส้นตรง (Linear combination) แล้วจึงหาค่าน้ำหนัก (Weights) ที่ทำให้ค่าพยากรณ์มีความผิดพลาด (Errors) น้อยที่สุด

แผนผังขั้นตอนของตัวแบบผสมชนิดนี้เป็นดัง **ผิดพลาด! ไม่พบแหล่งการอ้างอิง** ด้านล่าง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 2.19 ขั้นตอนการสร้างตัวแบบ Combined

วิธีการ Traditional hybrid มีขั้นตอนดังนี้

1) นำข้อมูลชุด Train เข้าสู่ตัวแบบ ARIMA, ANN และ SVM
2) ทำการพยากรณ์ข้อมูลชุด test สำหรับแต่ละตัวแบบ ARIMA, ANN และ SVM จะได้ค่าพยากรณ์ ตามลำดับดังนี้ $\hat{Y}_{ARIMA}, \hat{Y}_{ANN}, \hat{Y}_{SVM}$
3) นำค่าพยากรณ์ทั้ง 3 ตัวแบบ มาทำการใส่ค่าน้ำหนัก (weight) ซึ่งมีค่าตั้งแต่ 0 ถึง 1 โดยที่ผลรวมของค่า weight รวมกันมีค่าเท่ากับ 1

$$\hat{Y}_{TOTAL} = W_1\hat{Y}_{ARIMA} + W_2\hat{Y}_{ANN} + W_3\hat{Y}_{SVM} \quad (2.28)$$

โดยที่

$$W_1 + W_2 + W_3 = 1 \quad (2.29)$$

ในกรณีที่มีตัวแบบพยากรณ์มากกว่านี้ ก็สามารถเขียนอยู่ในเทอม รูปแบบทั่วไปได้ ดังนี้

$$\hat{Y}_{TOTAL} = \sum_{i=1}^{n} W_i\hat{Y}_i \quad (2.30)$$

มีเงื่อนไขว่า

$$\sum_{i=1}^{n} W_i = 1 \quad (2.31)$$

เมื่อ n = จำนวนชนิดของตัวแบบพยากรณ์

W_i = ค่า Weights ของตัวแบบพยากรณ์ที่ i

$\hat{Y}_i$ = ค่าพยากรณ์จากตัวแบบที่ i

4) ใส่ค่า Weights ตั้งต้น เพื่อค้นหาค่าที่ดีที่สุดที่ทำให้ค่า Errors ของการพยากรณ์มีค่าต่ำสุด
5) คำนวณค่า Errors จาก Weights นั้นๆ
6) ถ้าค่า Errors ที่ได้ยังสูงอยู่และไม่ใช่ Errors ที่ต่ำสุด ก็ต้องเปลี่ยนค่า Weights
7) ทำขั้นตอนที่ 6) และ 7) ซ้ำ ไปเรื่อยๆ จนกระทั่งได้ค่า Errors ต่ำสุด

สิ่งที่ต้องระวังในการค้นหาค่า Weights นั้น คือ จำนวนตำแหน่งทศนิยมที่เลือก เช่น กำหนดให้ค่า Weights ที่มีจำนวนทศนิยมมากเกินไป เช่น $W_1 = 0.00001,\ W_2 = 0.56999,\ W_3 = 0.43$ ปรับทีละ 0.00001 จะได้ค่า Weights รอบที่สอง คือ $W_1 = 0.00002,\ W_2 = 0.56998,\ W_3 = 0.43$ และต้องปรับไปเรื่อยๆ จนครบทุกโอกาส (All possible combination) จะใช้เวลาในการหาค่า Weights ที่ดีที่สุดนาน วิธีการที่ดีที่สุด คือ ทดลองใส่ค่า Weights จากทศนิยมน้อยๆ ก่อน เช่น เริ่มจากทศนิยมอันดับที่สอง $W_1 = 0.01,\ W_2 = 0.56,\ W_3 = 0.43$ ปรับทีละ 0.01 แล้วหาค่า Weights ที่ดีที่สุด และคำนวณค่า Errors จากนั้นทดลองเปลี่ยนเป็นทศนิยมอันดับที่สาม $W_1 = 0.001, W_2 = 0.569,\ W_3 = 0.430$ปรับทีละ 0.001 หาค่า Weights ที่ดีที่สุด และคำนวณค่า Errorsถึงจุดนี้ก็ต้องลองเปรียบเทียบค่า Errors ทั้งสองว่า ค่า Errors จากทศนิยมตำแหน่งที่สามนั้นน้อยกว่า ค่า Errors จากทศนิยมตำแหน่งที่สองมากน้อยแค่ไหน ถ้าน้อยกว่ามากก็อาจค้นหาจากทศนิยมตำแหน่งที่สี่ แล้วนำมาเทียบกับทศนิยมตำแหน่งที่สามและทำเช่นนี้ไปเรื่อยๆ ในทศนิยมตำแหน่งอื่นที่สูงขึ้น แต่ถ้าลดลงน้อย และพอใจแล้วก็หยุดทำ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.7 การประเมินความแม่นยำของตัวแบบ (Evaluation performance)

ตัวแบบพยากรณ์ทั้ง 6 ตัวแบบ ประกอบด้วย ARIMA, ANN, SVM, ARIMAANN, ARIMASVM และ Combined (ARIMA+ANN+SVM) จะถูกประเมินความแม่นยำและเปรียบเทียบประสิทธิภาพของตัวแบบ ด้วยตัววัดดังต่อไปนี้

$$\text{Root mean square error (RMSE)} = \sqrt{\frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n}} \quad (2.32)$$

$$\text{Mean Square Error (MSE)} = \frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n} \quad (2.33)$$

$$\text{Mean Absolute Error (MAE)} = \frac{\sum_{t=1}^{n} |Y_t - \hat{Y}_t|^2}{n} \quad (2.34)$$

$$\text{Mean Absolute Percentage Error (MAPE)} = \frac{\sum_{t=1}^{n} \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right|}{n} \times 100 \quad (2.35)$$

- โดยที่ n คือจำนวนข้อมูลที่ทำการทดสอบ
- Y_t คือข้อมูลจริงที่เวลา t
- $\hat{Y}_t$ คือค่าพยากรณ์ที่เวลา t

ข้อมูลที่นำมาทดสอบ จำกัดเฉพาะข้อมูลชุดทดสอบเท่านั้น

2.8 ทบทวนวรรณกรรม

งานวิจัย (Zhang, 2003) ได้ทำการพยากรณ์ข้อมูลอนุกรมเวลาโดยใช้ตัวแบบ ARIMAANN โดยสมมติเงื่อนไขความสัมพันธ์ของการ Hybrid เป็นผลบวก เปรียบเทียบการพยากรณ์กับตัวแบบเดี่ยว ทั้ง ARIMA และ ANN ทำการพยากรณ์ล่วงหน้า 1 ค่า (One step ahead forecasting) สำหรับตัวแบบ ANN ได้ใช้ตัวแบบในลักษณะ Feedforward network มี 1 Hidden layer 1 Output neuron ใช้ Logistic function เป็น Activation function ใช้เทคนิคการเรียนรู้ generalized reduced gradient (GRG2) สำหรับข้อมูลที่ใช้มีจำนวน 3 ชุด คือ ข้อมูลจำนวนจุดดับบนดวงอาทิตย์รายปี (Sunspot), ข้อมูลจำนวนแมวน้ำที่ถูกจับได้ในประเทศแคนาดา (Canadian lynx) และข้อมูลอัตราแลกเปลี่ยนเงินสกุลปอนด์อังกฤษต่อเงินสกุลดอลลาร์สหรัฐ ผลการทดลองพบว่าตัวแบบ ARIMAANN มีประสิทธิภาพดีกว่าตัวแบบ ARIMA และ ANN

งานวิจัย (Ömer Faruk, 2010) งานวิจัยชิ้นนี้ได้ศึกษาประสิทธิภาพของตัวแบบ ARIMAANN โดยสมมติเงื่อนไขความสัมพันธ์ของการ Hybrid เป็นผลบวก เทียบกับตัวแบบ ARIMA และ ANN โดยทำการพยากรณ์คุณภาพของน้ำที่แม่น้ำ Buyuk Menderes ภายใต้ 3 ตัววัด คือ ระดับอุณหภูมิของน้ำ, ปริมาณโบรอนในน้ำ และปริมาณออกซิเจนที่ละลายในน้ำ ใช้ข้อมูลตัวอย่างรายเดือนจำนวนทั้งสิ้น 108 ข้อมูล แบ่งข้อมูลออกเป็น 2 ชุด คือ ข้อมูลชุดสร้างตัวแบบจำนวน 72 ข้อมูล และข้อมูลชุดทดสอบจำนวน 36 ข้อมูล สำหรับตัวแบบ ANN ใช้เทคนิคการเรียนรู้ Scale conjugate gradient (SCG) ผลการทดลองพบว่า ตัวแบบ ARIMAANN มีความแม่นยำมากกว่าตัวแบบเดี่ยวทั้งสองค่า Correlation ระหว่าง ค่าพยากรณ์ของตัวแบบผสมกับค่าจริงของข้อมูลชุดทดสอบของ ปริมาณโบรอนในน้ำ, ปริมาณออกซิเจนที่ละลายในน้ำและระดับอุณหภูมิของน้ำ มีค่า 0.902, 0.893 และ 0.909 ตามลำดับ ซึ่งถือเป็นที่น่าพอใจในงานการพยากรณ์คุณภาพน้ำ

งานวิจัย (Tseng et al., 2002) นำเสนอตัวแบบผสม Hybrid ทำการผสมตัวแบบ Seasonal ARIMA (SARIMA) กับตัวแบบ Neural networks แบบ Backpropagation เรียกว่า SARIMABP ทำการเปรียบเทียบกับตัวแบบเดี่ยวๆ ที่มีการ Difference และ Deseasonalized และตัวแบบ SARIMA

รวมเป็น 3 ตัวแบบ นำมาพยากรณ์ข้อมูลอนุกรมเวลาที่มีปัจจัยเชิงฤดูกาล 2 ชุดข้อมูล ได้แก่ การผลิตเครื่องจักรรวมและเครื่องดื่มของประเทศไต้หวันรวม ทำการพยากรณ์แบบ 1 ค่าล่วงหน้า ใช้ข้อมูลทดลอง 2 ชุด แต่ละชุดแบ่งข้อมูลเป็น 2 กลุ่มคือชุดเรียนรู้และชุดทดสอบ เปรียบเทียบตัวแบบทั้ง 3 ด้วย MSE, MAE และ MAPE ผลการทดลองพบว่าตัวแบบ SARIMABP ให้ค่าทั้งสามต่ำที่สุด ดีกว่าตัวแบบทั้งสาม ทั้ง 2 ชุดข้อมูล

งานวิจัย (Pai and Lin, 2005) งานวิจัยชิ้นนี้ทำการพยากรณ์ราคาหุ้นจำนวน 10 ชุดข้อมูล โดยใช้ตัวแบบ Hybrid ARIMASVM เทียบความแม่นยำกับตัวแบบ ARIMA และ SVM ทำการพยากรณ์ 1 ค่าล่วงหน้า (One-step ahead forecasting) เปรียบเทียบความแม่นยำของแต่ละตัวแบบด้วย RMSE,MSE,MAE และ MAPE ผลการทดลองพบว่า ตัวแบบ Hybrid ARIMASVM ให้ผลการพยากรณ์มีความแม่นยำกว่าทั้งตัวแบบ ARIMA และ ตัวแบบ SVM ผู้วิจัยนำเสนอว่าตัวแบบผสมนี้มีประสิทธิภาพที่ดี เนื่องจากนำส่วนที่ดีของแต่ละตัวแบบ คือ ARIMA และ SVM มาใช้ นอกจากนี้ยังเสนอแนะ ว่าควรมีการนำตัวแบบ ARIMA และ SVM มาผสมกันโดยใช้เงื่อนไขอื่น นอกจากการบวกกัน เช่น การคูณ เป็นต้น และแนะนำการใช้เทคนิคในการค้นหาพารามิเตอร์ของตัวแบบ SVM เพื่อให้ได้ตัวแบบที่มีความแม่นยำและใช้เวลาค้นหาสั้นลง

งานวิจัย (Shi et al., 2012) งานวิจัยชิ้นนี้ทำการพยากรณ์ เพื่อจัดการในงานการผลิตกระแสไฟฟ้าด้วยความเร็วลม โดยทำการพยากรณ์ข้อมูลสองชุด คือ ความเร็วลม และกำลังไฟฟ้าจากกังหันลม ทดสอบโดยใช้ตัวแบบผสม 2 ชนิด คือ Hybrid ARIMAANN และ ARIMASVM เทียบความแม่นยำกับตัวแบบเดี่ยว คือ ARIMA, ANN และ SVM โดยใช้ตัวประเมินตัวแบบ คือ MAE และ RMSE ผลการทดลองสรุปได้ว่าตัวแบบผสม ไม่ได้ดีไปกว่าตัวแบบเดี่ยว อาจเนื่องมาจากเงื่อนไขระหว่างตัวแบบเชิงเส้นตรงคือ ARIMA และตัวแบบไม่เป็นเชิงเส้นตรง คือ ANN และ SVM ไม่ได้เป็นผลบวก อาจมีความสัมพันธ์เป็นอย่างอื่นเช่น ผลคูณ ซึ่งนั่นทำให้ค่าเศษเหลือที่นำไปสร้างตัวแบบทั้ง ANN และ SVM ไม่ได้เป็นข้อมูลที่ไม่เป็นเชิงเส้นตรงเพียงอย่างเดียวแต่มีความสัมพันธ์เชิงเส้นตรงรวมอยู่ด้วย investigated.

งานวิจัย (Wang et al., 2012) ทำการพยากรณ์ราคาหุ้น โดยใช้ตัวแบบ Exponential smoothing (ESM), ARIMA และ Back propagation neural network (BPNN) นำมาพยากรณ์และนำเสนอตัวแบบผสม Hybrid combined โดยนำตัวแบบทั้งสามมารวมกันด้วยค่าน้ำหนัก โดยใช้ Genetic algorithm ในการค้นหาค่า Weights เรียกตัวแบบที่นำเสนอว่า (Proposed hybrid model:PHM) วัดผลความขนาดความแม่นยำด้วย MAE, RMSE, MAPE, ME ละวัดผลทิศทางความแม่นยำด้วยตัววัด DA (Direction accuracy) เทียบกับตัวแบบทั้งสาม และเทียบกับตัวแบบ EWH คือตัวแบบ Hybrid ที่กำหนดให้น้ำหนักมีค่าเท่ากัน และ RWM (Random walk model) ซึ่งเป็นตัวแบบที่นิยมใช้ในการพยากรณ์ราคาหุ้น ผลการทดลองพบว่าตัวแบบ PHM มีความแม่นยำทั้งขนาดและทิศทางมากกว่าทุกๆตัวแบบ เหมาะที่จะนำไปใช้กับการพยากรณ์ราคาหุ้น

งานวิจัย (Wang et al., 2010) ทำการพยากรณ์กำลังไฟฟ้าเพื่อที่จะผลิตกระแสไฟฟ้าให้เพียงพอและประหยัดงานวิจัยชิ้นนี้นำเสนอตัวแบบผสม Hybrid combined โดยใช้ตัวแบบ SARIMA, Exponential smoothing แบบมีปัจจัยฤดูกาล และ Support vector machine แบบถ่วงน้ำหนัก โดยนำ Adaptive particle swarm optimization (APSO) มาช่วยในการค้นหาค่า Weights ที่

เหมาะสม ตัวแบบผสมในที่นี้ทำการรวมทั้งสามตัวแบบให้อยู่ในรูปแบบผลรวมเชิงเส้นตรง โดยที่ใช้ APSO ในการค้นหา Weights ทั้งสามตัวแบบ ใช้ Regression significance test ทดสอบสมการผลรวมเชิงเส้นตรงที่ได้ว่าดีและผ่านเงื่อนไขหรือไม่ ผลการทดลองพบว่าตัวแบบผสมที่นำเสนอนี้ดีกว่าตัวแบบเดี่ยว ทั้งสามตัวแบบ เช่น ความแม่นยำของตัวแบบที่นำเสนอเพิ่มขึ้นกว่าตัวแบบ S-ARIMA, SESM, W-SVM คิดเป็นเปอร์เซนต์ คือ 30.746%, 45.358%, 45.494% ตามลำดับ การพยากรณ์ด้วยตัวแบบเดี่ยวนั้นไม่สามารถเก็บรายละเอียดของข้อมูลได้ดีเพียงพอ การผสมตัวแบบจะเป็นข้อดีของแต่ละตัวแบบทำให้เก็บรายละเอียดของข้อมูลได้ดีมากยิ่งขึ้น ตัวแบบผสมที่นำเสนอนี้ สามารถใช้นำไปใช้พยากรณ์กำลังไฟฟ้า ในการจัดตารางการผลิต เพื่อจะลดปริมาณกำลังไฟฟ้าที่ปล่อยออกมาเกินความจำเป็นช่วยบริษัทผลิตกระแสไฟฟ้าในการตัดสินใจ

งานวิจัย (Chen and Wang, 2007) นำเสนอตัวแบบผสมโดยนำประสิทธิภาพของตัวแบบ SARIMA และตัวแบบSupport Vector Machine (SVM) ทำการพยากรณ์ข้อมูลอนุกรมเวลาที่มีปัจจัยเชิงฤดูกาล โดยใช้ข้อมูลตัวอย่างของมูลค่าผลผลิตเครื่องจักรกลในประเทศไต้หวัน เทียบประสิทธิภาพตัวแบบผสมกับตัวแบบเดี่ยว คือ SARIMA และ SVM สำหรับข้อมูลที่นำมาทดสอบแบ่งเป็นสองชุด คือ ชุดเรียนรู้เพื่อสร้างตัวแบบและชุดทดสอบตัวแบบ ข้อมูลชุดเรียนรู้ตัวแบบมีจำนวน 72 ตัวอย่าง ส่วนข้อมูลชุดทดสอบมีจำนวน 10 ตัวอย่าง สำหรับตัวแบบ SVM มีการนำเทคนิค Genetic algorithm มาใช้เพื่อช่วยค้นหาพารามิเตอร์ของ SVM ตัวแบบผสมใช้หลักการ Traditional hybrid สมมติความสัมพันธ์ระหว่าง SARIMA และ SVM เป็นผลบวก (Additive) โดยแบ่งเป็น 3 ตัวแบบย่อยที่ใช้ทดลอง คือ SARIMASVM1, SARIMASVM2, SARIMASVM3 ที่มีการนำ Lag นำเข้าตัวแบบ SVM ที่ต่างกัน สำหรับตัววัดประสิทธิภาพของตัวแบบใช้ Normalized mean square error (NMSE), mean absolute percentage error (MAPE) ตัวสถิติวัดความแม่นยำด้านทิศทาง และขนาดสัมประสิทธิ์ความสัมพันธ์ (R^2) ผลการทดลองพบว่าตัวแบบ SARIMASVM2 ซึ่งมี Lag นำเข้า คือ ($Y_{t-1}, Y_{t-12,}\varepsilon_t$) ให้ผลความแม่นยำดีที่สุดทั้งในแง่ขนาดความผิดพลาดที่น้อยที่สุด มีความแม่นยำดีสุดและสามารถพยากรณ์ทิศทางได้แม่นยำอีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3

วิธีการทดลอง

นำข้อมูลทั้ง 10 ชุดแบ่งข้อมูลออกเป็น 2 กลุ่ม คือข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ รวมทั้งตั้งชื่อตัวแปร เพื่อความสะดวกในการเปรียบเทียบ ดังตารางที่ 3.1

ข้อมูล	ชื่อตัวแปร	จำนวนข้อมูล	จำนวน Training	จำนวน Test	ช่วงข้อมูล Training	ช่วงข้อมูล Test
Sunspot	Sunspot	288	221	67	1-221	222-288
Canadian lynx	Canadian lynx	114	100	14	1-100	101-114
GBP/USD exchange rate	GBP/USD	731	679	52	1-679	680-731
ข้อมูลรายเดือนปริมาณการใช้พลังงานไฟฟ้าทั้งประทศไทย(หน่วย:ล้านหน่วย)	Electricuse	151	121	30	1-121	122-151
ข้อมูลรายเดือนยอดขายรถจักรยานยนต์ทั้งประทศไทย(หน่วย:คัน)	Motorcycle	151	121	30	1-121	122-151
ข้อมูลรายวันอัตราแลกเปลี่ยนรายวันของเงินสกุลดอลลาร์สหรัฐและเงินบาท(หน่วย:บาท/1 US dollar)	USexchange	173	143	30	1-143	144-173
ข้อมูลรายวันราคาทองคำหนักหนึ่งบาท(หน่วย:บาท/ทองคำหนัก 1 บาท)	Gold	220	190	30	1-190	191-220
ข้อมูลรายวันความเร็วลมโดยเฉลี่ยในรอบวัน(นอต)	Windspeed	365	335	30	1-335	336-365
ข้อมูลรายเดือนปริมาณน้ำฝนโดยเฉลี่ย(มิลลิเมตร)	Meanrain	130	100	30	1-100	101-130
ข้อมูลรายเดือนระดับอุณหภูมิสูงสุด(เซลเซียส)	Maxtemp	130	100	30	1-100	101-130

ตารางที่ 3.1 แสดงจำนวนข้อมูลในชุด Train และ Test รวมถึงชื่อตัวแปรที่ใช้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.1 การสร้างตัวแบบ ARIMA

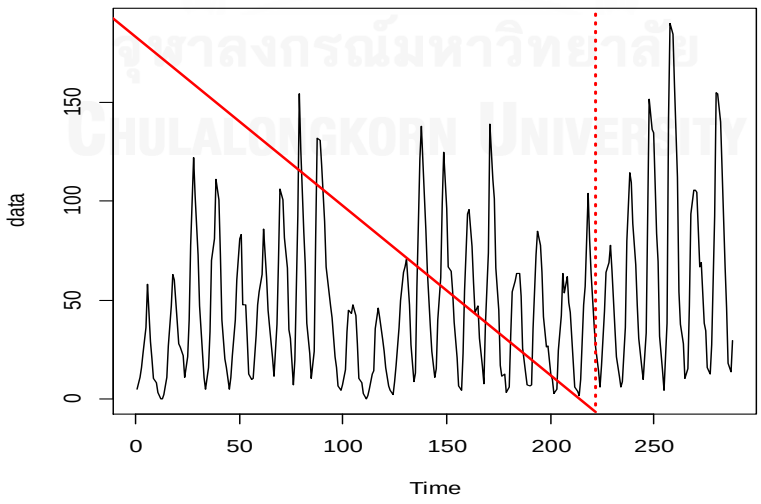
โดยใช้โปรแกรม R version 3.0.1 “Good Sport” package “forecast” ทำการเลือกอันดับตัวแบบด้วยค่า AICc information criteria ทำการหาอันดับตัวแบบที่ดีที่สุดในรูปแบบการค้นหาแบบอัตโนมัติ มีพารามิเตอร์

- กำหนดค่าอันดับ p ซึ่งเป็นอันดับของ AR เทอม ตั้งแต่ 0 ถึง 15 เนื่องจากอันดับ p คือข้อมูลในอดีต การกำหนดอันดับสูงสุดที่ 15 เพื่อให้รองรับกับข้อมูลที่มีปัจจัยฤดูกาลที่จำนวนรอบของฤดูกาลมักเป็น 12 เช่นข้อมูลปริมาณการใช้ไฟฟ้ารายเดือน เป็นต้น
- กำหนดค่าอันดับ q ซึ่งเป็นอันดับของ MA เทอม ตั้งแต่ 0 ถึง 5 โดยปกติแล้วอันดับ q คือส่วนของ Error หรือเรียกว่าความผิดพลาดเชิงสุ่มตาม Lag ต่างๆ โดยทั่วไปจะมีค่าไม่มากนัก
- อันดับ d เป็นจำนวนครั้งของการ Difference ไม่สามารถกำหนดได้ขึ้นอยู่กับข้อมูล เช่น ถ้าข้อมูลมีคุณสมบัติ Stationary อยู่แล้วก็ไม่มีอันดับ d (d=0) หรือหากข้อมูลไม่เป็น Stationary ทำการ Difference 1 ครั้ง d=1 ทำการแปลง 2 ครั้ง d=2
- ค้นหาอันดับของตัวแบบที่ทำให้ค่า Information criteria มีค่าน้อยที่สุด แล้วนำตัวแบบ ARIMA(p,d,q) นั้นไปพยากรณ์ข้อมูลในชุดทดสอบแล้วคำนวณค่า RMSE,MSE,MAE และ MAPE เพื่อนำไปเปรียบเทียบกับตัวแบบอื่นๆ

3.1.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ ARIMA

ขั้นตอนที่ 1 plot กราฟ ดูลักษณะข้อมูล ดังภาพที่ 3.1

ขั้นตอนที่ 2 แบ่งข้อมูลออกเป็นสองชุด คือข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ



ภาพที่ 3.1 ข้อมูล Sunspot ที่ทำการแบ่งข้อมูลออกเป็นสองกลุ่ม

จากภาพที่ 3.1 ข้อมูลชุดเรียนรู้อยู่ด้านซ้ายมือของเส้นประ มีจำนวน 221 ข้อมูล ส่วนข้อมูลชุดทดสอบอยู่ด้านขวามือของเส้นประมีจำนวน 67 ข้อมูล

ขั้นตอนที่ 3 ทดสอบคุณสมบัติ Stationary

เนื่องจากคุณสมบัติ Stationary เป็นสิ่งจำเป็นที่ข้อมูลจะต้องพึงมีก่อนที่จะทำการพยากรณ์ด้วยตัวแบบ ARIMA จึงทำการทดสอบด้วยตัวสถิติทดสอบ Augmented Dickey and Fuller (ADF) ดังภาพที่ 3.2

```
        Augmented Dickey-Fuller Test

data:  sunspot
Dickey-Fuller = -4.4595, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

ภาพที่ 3.2 การทดสอบคุณสมบัติ Stationary ด้วยสถิติทดสอบ ADF

เกณฑ์การทดสอบจะเปรียบเทียบค่า p-value กับระดับนัยสำคัญทางสถิติ α=0.05 กรณีนี้ p-value=0.01< α=0.05 นั่นคือ ข้อมูลชุดนี้มีคุณสมบัติ Stationary

ขั้นตอนที่ 4 ทำการค้นหาอันดับของตัวแบบ ARIMA โดยคัดเลือกตัวแบบจากค่า AICc ที่น้อยที่สุด

ARIMA(p,d,q)	AICc
ARIMA(0,0,0)	2193.415
ARIMA(0,0,1)	2003.531
ARIMA(0,0,2)	1893.922
ARIMA(0,0,3)	1869.124
ARIMA(0,0,4)	1853.124
ARIMA(0,0,5)	1854.794
ARIMA(1,0,0)	1954.472
ARIMA(1,0,1)	1886.214
ARIMA(1,0,2)	1855.872
ARIMA(1,0,3)	1852.608
ARIMA(1,0,4)	1852.249
ARIMA(1,0,5)	1E+20
ARIMA(2,0,0)	1833.131
ARIMA(2,0,1)	1832.621
ARIMA(2,0,2)	1834.583
ARIMA(2,0,3)	1836.651
ARIMA(2,0,4)	1829.372

ARIMA(2,0,5)	1829.298
ARIMA(3,0,0)	1833.068
ARIMA(3,0,1)	1833.679
ARIMA(3,0,2)	1827.72
ARIMA(3,0,3)	1823.82
ARIMA(3,0,4)	1817.817
ARIMA(3,0,5)	1833.878
ARIMA(4,0,0)	1835.916
ARIMA(4,0,1)	1836.355
ARIMA(4,0,2)	1834.055
ARIMA(4,0,3)	1841.451
ARIMA(4,0,4)	1840.225
ARIMA(4,0,5)	1841.92
ARIMA(5,0,0)	1838.481
ARIMA(5,0,1)	1838.138
ARIMA(5,0,2)	1830.825
ARIMA(5,0,3)	1832.791
ARIMA(5,0,4)	1823.835
ARIMA(5,0,5)	1E+20
ARIMA(6,0,0)	1836.279
ARIMA(6,0,1)	1828.183
ARIMA(6,0,2)	1832.04
ARIMA(6,0,3)	1833.198
ARIMA(6,0,4)	1E+20
ARIMA(6,0,5)	1E+20
ARIMA(7,0,0)	1827.032
ARIMA(7,0,1)	1820.805
ARIMA(7,0,2)	1821.683
ARIMA(7,0,3)	1822.213
ARIMA(7,0,4)	1820.171
ARIMA(7,0,5)	1822.451
ARIMA(8,0,0)	1820.31
ARIMA(8,0,1)	1818.984
ARIMA(8,0,2)	1820.947
ARIMA(8,0,3)	1823.141

ARIMA(8,0,4)	1822.347
ARIMA(8,0,5)	1824.63
ARIMA(9,0,0)	1820.652
ARIMA(9,0,1)	1821.488
ARIMA(9,0,2)	1828.852
ARIMA(9,0,3)	1828.431
ARIMA(9,0,4)	1830.456
ARIMA(9,0,5)	1E+20
ARIMA(10,0,0)	1823.16
ARIMA(10,0,1)	1826.141
ARIMA(10,0,2)	1E+20
ARIMA(10,0,3)	1E+20
ARIMA(10,0,4)	1E+20
ARIMA(10,0,5)	1E+20
ARIMA(11,0,0)	1824.422
ARIMA(11,0,1)	1826.321
ARIMA(11,0,2)	1827.383
ARIMA(11,0,3)	1E+20
ARIMA(11,0,4)	1E+20
ARIMA(11,0,5)	1E+20
ARIMA(12,0,0)	1826.132
ARIMA(12,0,1)	1834.531
ARIMA(12,0,2)	1825.152
ARIMA(12,0,3)	1E+20
ARIMA(12,0,4)	1E+20
ARIMA(12,0,5)	1E+20
ARIMA(13,0,0)	1829.112
ARIMA(13,0,1)	1831.755
ARIMA(13,0,2)	1831.432
ARIMA(13,0,3)	1830.536
ARIMA(13,0,4)	1832.81
ARIMA(13,0,5)	1E+20
ARIMA(14,0,0)	1830.242
ARIMA(14,0,1)	1832.386
ARIMA(14,0,2)	1834.706

ARIMA(14,0,3)	1E+20
ARIMA(14,0,4)	1E+20
ARIMA(14,0,5)	1E+20
ARIMA(15,0,0)	1831.695
ARIMA(15,0,1)	1E+20
ARIMA(15,0,2)	1834.736
ARIMA(15,0,3)	1E+20
ARIMA(15,0,4)	1E+20
ARIMA(15,0,5)	1E+20

ตารางที่ 3.2 ค่า AICc ในแต่ละอันดับของตัวแบบ ARIMA

ตารางที่ 3.2 แสดงค่า AICc ในแต่ละอันดับของตัวแบบ ARIMA ภายใต้กรอบอันดับในการค้นหาซึ่งทำการค้นหาตั้งแต่ค่า p=0 ถึง p=15 และค่า q=0 ถึง q=5 เลือกอันดับของตัวแบบที่ให้ค่า AICc น้อยที่สุด จะได้อันดับที่ดีที่สุด ได้แก่ p=3, d=0 และ q=4 ให้ผลการวิเคราะห์แสดงในภาพที่ 3.3

```
Series: train.observation
ARIMA(3,0,4) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ma1     ma2     ma3     ma4  intercept
      2.4766  -2.2967  0.7793  -1.2866  0.3148  0.0526  0.1319    42.6717
s.e.  0.0657   0.1103  0.0626   0.0965  0.1165  0.1104  0.0840     4.8163

sigma^2 estimated as 197.2:  log likelihood=-899.08
AIC=1816.15   AICc=1817.01   BIC=1846.74
```

ภาพที่ 3.3 อันดับ ARIMA ที่ดีที่สุด

ภาพที่ 3.3 แสดงอันดับที่ดีที่สุดที่ได้จากการค้นหาโดยใช้ AICc ดังตาราง 3.2 อันดับที่ดีที่สุด คือ ARIMA(3,0,4) ประกอบด้วย p=3, d=0 และ q=4 มีค่า AICc=1817.01 สำหรับค่า AIC และ BIC ซึ่งเป็น Information criteria จะมีค่าในทิศทางเดียวกัน ถ้าอันดับใดให้ค่า AICc น้อยก็จะให้ค่า AIC และ BIC น้อยด้วยเช่นกัน

ขั้นตอนที่ 4 ทดสอบคุณสมบัติทางสถิติของค่าเศษเหลือ

ก) ทดสอบความเป็นอิสระของข้อมูลอนุกรมเวลา ด้วย Box-Pierce and Ljung-Box Tests ดังภาพที่ 3.4

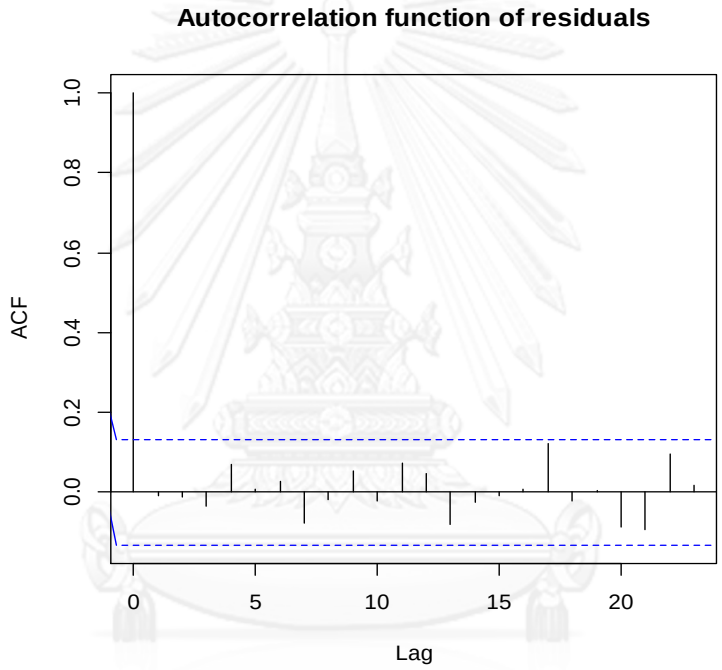
```
        Box-Ljung test

data:  fit$res
X-squared = 3.9132, df = 9, p-value = 0.917
```

ภาพที่ 3.4 การทดสอบความเป็นอิสระของข้อมูลอนุกรมเวลาด้วย Box-Pierce and Ljung-Box Tests

การทดสอบความเป็นอิสระของข้อมูลด้วย Box-Pierce and Ljung-Box มีสมมติฐานว่าง คือ ข้อมูลเป็นอิสระต่อกัน ซึ่ง p-value=0.917> α=0.05 ดังนั้นยอมรับสมมติฐานว่าง กล่าวคือ ข้อมูลเป็นอิสระกัน ซึ่งผ่านเงื่อนไขการวิเคราะห์ค่าเศษเหลือที่ได้จากตัวแบบ ARIMA

ข) ตรวจสอบกราฟค่าเศษเหลือ

ข.1) กราฟ Autocorrelation function (ACF) ของเศษเหลือ

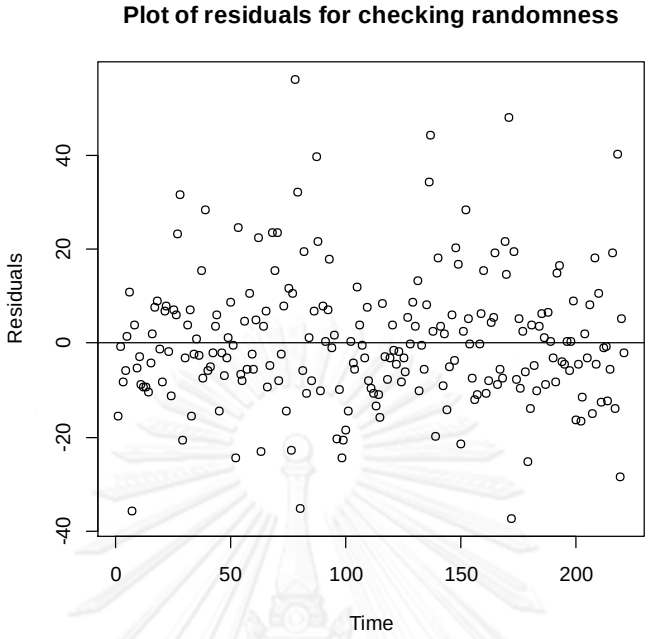


ภาพที่ 3.5 ACF ของค่าเศษเหลือ

กราฟ ACF ที่เหมาะสม เมื่อค่าเศษเหลือเข้าเงื่อนไขเป็นอิสระจากกัน จะเป็นไปตามภาพที่ 3.5 เส้น ACF จะทะลุออกนอกเส้นประที่เป็นขอบเขตที่ Lag=0 เพียงอย่างเดียว แต่ที่ Lag อื่นๆ เส้น ACF จะอยู่ภายในเส้นประทั้งหมด

ข.2) กราฟ Scatter plot ของค่าเศษเหลือ

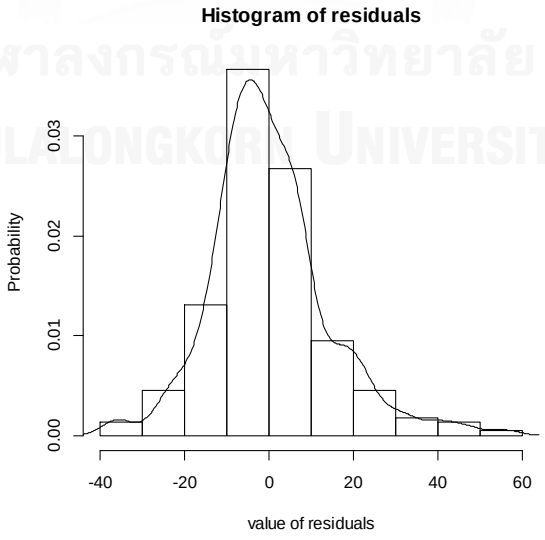
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 3.6 Scatter plot ของค่าเศษเหลือ

Scatter plot ช่วยบอกให้รู้ว่าข้อมูลมีการกระจายตัวเป็นอิสระหรือไม่ ถ้าไม่เป็นอิสระน่าจะมีความสัมพันธ์ในรูปแบบใด เป็นเครื่องมือที่ช่วยตรวจสอบความผิดปกติของความสัมพันธ์ของค่าเศษเหลือ จากภาพที่ 3.6 จัดว่าค่าเศษเหลือมีการกระจายตัวแบบสุ่ม ไม่ได้มีรูปแบบการกระจายตัวที่ระบุได้ชัดเจน

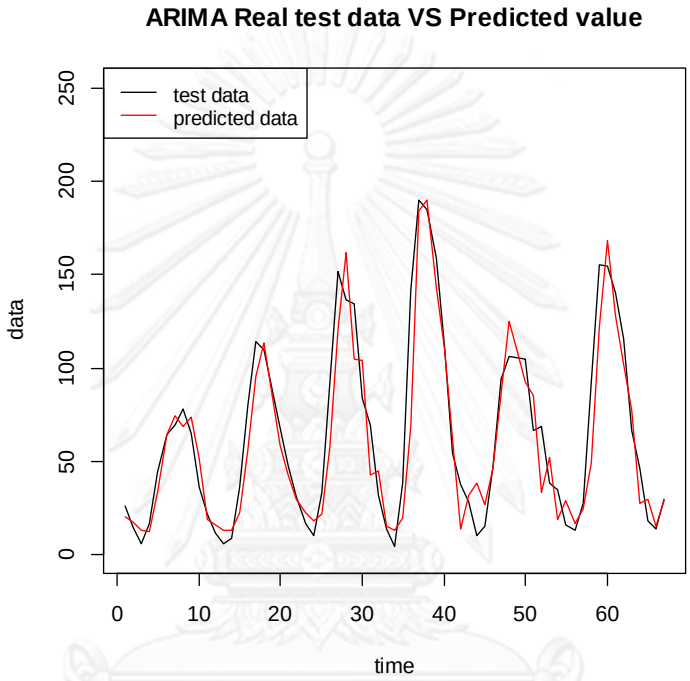
ข.3) Histogram ของค่าเศษเหลือ



ภาพที่ 3.7 Histogram ของค่าเศษเหลือ

Histogram ที่มีการลากเส้นโค้งตามแท่ง Histogram นั้น เป็นการตรวจสอบคุณสมบัติของค่าเศษเหลือว่ามีการแจกแจงแบบปกติหรือไม่ จากภาพที่ 3.7 พบว่าข้อมูลมีลักษณะการแจกแจงแบบปกติ

ขั้นตอนที่ 5 ทำการพยากรณ์ข้อมูลล่วงหน้าในข้อมูลชุดทดสอบทั้ง 67 ข้อมูล ในรูปแบบการพยากรณ์หนึ่งค่าล่วงหน้า ดังภาพ 3.8



ภาพที่ 3.8 เปรียบเทียบค่าจริงและค่าพยากรณ์ ตัวแบบ ARIMA

กราฟเส้นสีแดงแสดงค่าพยากรณ์ ส่วนเส้นสีดำแสดงค่าจริง ค่าพยากรณ์วิ่งใกล้เคียงกับค่าจริง แต่จะคลาดเคลื่อนจากค่าจริงในจุดปลายทั้งบนและล่าง แสดงถึงจุดข้อมูลที่มีความเคลื่อนที่ ARIMA ไม่สามารถจัดการได้

ค่าพารามิเตอร์ที่ดีที่สุดของ ARIMA พร้อมทั้งตัวประเมินประสิทธิภาพตัวแบบ RMSE, MSE, MAE, MAPE และกราฟค่าพยากรณ์เปรียบเทียบกับค่าจริง จะแสดงไว้ในบทที่ 4

3.2 การสร้างตัวแบบ ANN

ANN ทำการทดลองด้วย MATLAB R2012a ด้วย Neural Network time series toolbox โดยเลือกกำหนดส่วนประกอบของ ANN ดังนี้

- โครงสร้างที่ใช้เป็นแบบ Feedforward

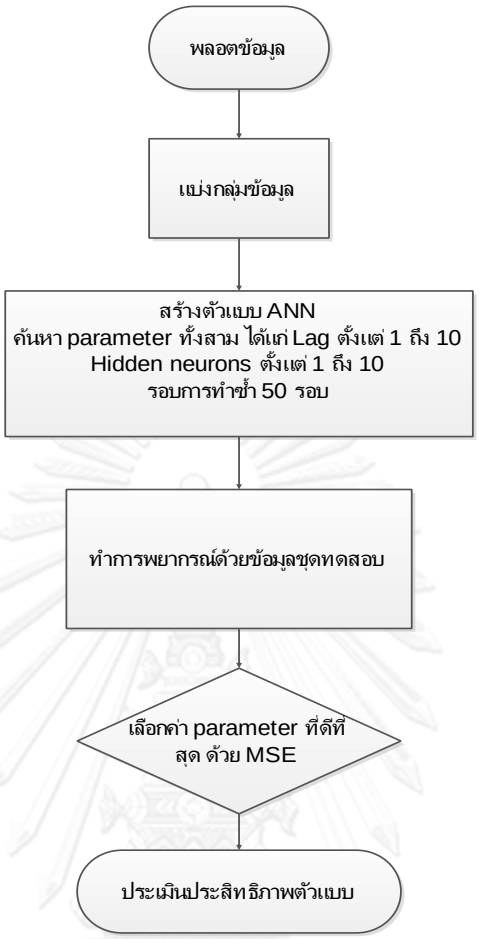
- เทคนิคการเรียนรู้ แบบ Levenberg-Marquardt backpropagation เนื่องจากเป็นเทคนิคการเรียนรู้ที่ใช้เวลาในการคำนวณน้อย ใช้ได้ดีกับการพยากรณ์เชิงปริมาณ
- จำนวน Lag นำเข้า เลือกจำนวนตั้งแต่ 1 ถึง 10 เพิ่มทีละ 1
- จำนวน Hidden layer=1, จำนวน Hidden neurons มีค่าตั้งแต่ 1-10 เพิ่มทีละ 1
- จำนวน Output neuron=1
- Activation function เลือกใช้ Hyperbolic tangent sigmoid transfer function
- Output function เลือกใช้ Linear function เป็นฟังก์ชันที่ใช้โดยทั่วไป ในการพยากรณ์ข้อมูลที่เป็นตัวเลข (Regression and time series)
- ทำการทดลองซ้ำ 50 ครั้งภายใต้ Initial weights ที่ต่างกัน 50 ชุด
- ใช้ MSE เป็นตัวประเมินความแม่นยำของตัวแบบในข้อมูลชุดทดสอบ สำหรับแต่ละชุดของพารามิเตอร์ที่ใส่ลงไป

สรุปมีตัวแปร 3 ตัวที่ต้องค้นหาค่าเอง ได้แก่ จำนวน Lag, จำนวน Hidden neurons และจำนวนการรันซ้ำ ปัจจุบันยังไม่มีกฎหรือทฤษฎีการเลือกพารามิเตอร์ที่เหมาะสม งานวิจัยชิ้นนี้จึงทำการค้นหาพารามิเตอร์ในปริมาณที่มากเพื่อโอกาสที่เจอพารามิเตอร์ที่ทำให้ค่าพยากรณ์มีความแม่นยำสูงยิ่งขึ้น รวมจำนวนครั้งการรันทั้งสิ้น 5,000 ครั้ง

3.2.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ ANN

ขั้นตอนการสร้างตัวแบบ ANN สามารถอธิบายได้ดังภาพด้านล่าง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 3.9 ขั้นตอนการสร้างตัวแบบ ANN

ขั้นตอนที่ 1 plot กราฟ ดูลักษณะข้อมูล ดังภาพที่ 3.1

ขั้นตอนที่ 2 แบ่งกลุ่มข้อมูลออกเป็นสองชุด คือ ข้อมูลชุดเรียนรู้จำนวน 221 ข้อมูล และข้อมูลชุดทดสอบมีจำนวน 67 ข้อมูล ดังภาพที่ 3.1

ขั้นตอนที่ 3 สร้างตัวแบบ ANN โดยค้นหาค่าพารามิเตอร์ทั้งสาม เริ่มจากการใส่ค่า Lag และ Hidden neurons เริ่มตั้งแต่ 1 ถึง 10 โดยชุดแรก คือ Lag=1, Hidden neurons=1 ทำซ้ำ 50 ครั้ง ชุดที่สอง คือ Lag=1, Hidden neurons=2 ทำซ้ำ 50 ครั้งเช่นเดิม ทำเช่นนี้ไปเรื่อยๆ จนกว่าจะถึง Lag=10 และ Hidden neurons=10

ขั้นตอนที่ 4 ตัวแบบที่สร้างจากพารามิเตอร์แต่ละชุด นำไปพยากรณ์ เปรียบเทียบกับข้อมูลชุดทดสอบ และคำนวณหาค่า MSE ดังตารางที่ 3.3 และ 3.4

ตัวอย่างผลลัพธ์ค่า MSE ในข้อมูลชุดทดสอบในแต่ละชุดค่าพารามิเตอร์ของตัวแบบ ANN สำหรับข้อมูล Sunspot

เมื่อ Lag=1, Hidden neurons มีค่าตั้งแต่ 1 ถึง 10

โครงข่าย (Architecture)	MSE
1-1-1	863.667
1-2-1	828.43
1-3-1	826.053
...	...
...	...
1-10-1	810.20

ตารางที่ 3.3 ค่า MSE ในโครงข่ายต่างๆ เมื่อ Lag=1 ทำซ้ำ 50 รอบ

เมื่อ Lag=2, Hidden neurons มีค่าตั้งแต่ 1 ถึง 10

โครงข่าย (Architecture)	MSE
2-1-1	397.58
2-2-1	314.17
2-3-1	299.99
...	...
2-10-1	272.78

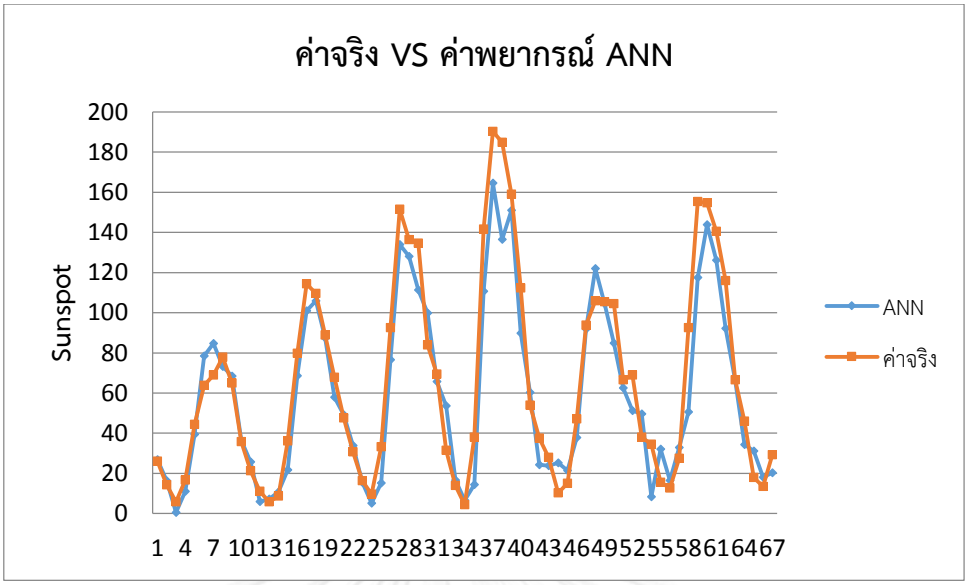
ตารางที่ 3.4 ค่า MSE ในโครงข่ายต่างๆ เมื่อ Lag=2 ทำซ้ำ 50 รอบ

ผลลัพธ์จากทั้งสองตารางข้างต้นเป็นตัวอย่างการสร้างตัวแบบ ANN ที่พารามิเตอร์ต่างๆ และยังผลให้ค่า MSE เท่าไร หลักการทำต่อไปก็ต้องเปลี่ยนเป็น Lag=3, 4, 5,...,10 เมื่อทำการทดลองครบตามกรอบพารามิเตอร์ที่ค้นหาจะได้โครงข่ายที่ดีที่สุดของตัวแบบ ANN ที่ให้ MSE ต่ำสุด

ขั้นตอนที่ 5 เลือกค่าพารามิเตอร์ที่ดีที่สุดที่ให้ค่า MSE ต่ำที่สุด

โครงข่าย ANN 7-9-1 ประกอบด้วย Lag=7, Hidden neurons=9 และ Output neurons=1 ที่รอบการทำซ้ำที่ 35 มีค่า MSE เท่ากับ 236.6

ขั้นตอนที่ 6 ทำการพยากรณ์ข้อมูลล่วงหน้าในข้อมูลชุดทดสอบทั้ง 67 ข้อมูล ในรูปแบบการพยากรณ์หนึ่งค่าล่วงหน้า ดังภาพ 3.10 พร้อมทั้งประเมินประสิทธิภาพตัวแบบด้วย RMSE, MSE, MAE และ MAPE



ภาพที่ 3.10 เปรียบเทียบค่าจริงและค่าพยากรณ์ตัวแบบ ANN

ค่าพารามิเตอร์ที่ดีที่สุดของ ARIMA พร้อมทั้งตัวประเมินประสิทธิภาพตัวแบบ RMSE, MSE, MAE, MAPE และกราฟค่าพยากรณ์เปรียบเทียบกับค่าจริง จะแสดงไว้ในบทที่ 4

3.3 การสร้างตัวแบบ SVM

SVM ทำการทดลองด้วย R version 3.0.1 “Good Sport” package “kernlab” ซึ่งเป็น package “kernlab” นี้ จัดว่า Interface ที่ R นำมาจาก libsvm เป็นเครื่องมือในการพยากรณ์ด้วยตัวแบบ SVMที่เป็นมาตรฐาน ที่โปรแกรมอื่นๆ ทั่วไปนำมาใช้โดยเลือกกำหนดส่วนประกอบของ SVM ดังนี้

- จำนวน Lag ที่ใช้ มีค่าตั้งแต่ 1 ถึง 10 โดยที่ค่าเพิ่มขึ้นทีละ 1
- Kernel function ที่ใช้คือ Radial basis function
- พารามิเตอร์ของ Radial basis function คือ σ ใส่ค่าดังต่อไปนี้ คือ 0.00001, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 1, 5 และ 10 รวมทั้งสิ้น 11 ค่า
- พารามิเตอร์ C ค่าที่ใส่ ได้แก่ 1, 5, 10, 25, 30, 35, 50, 70, 100, 400, 800, 1500, 2500 และ 3500 รวมทั้งสิ้น 14 ค่า
- พารามิเตอร์ ε ค่าที่ใส่ ได้แก่ 0.00001, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 และ 0.99 รวมทั้งสิ้น 10 ค่า
- ใช้ MSE เป็นตัวประเมินความแม่นยำของตัวแบบในข้อมูลชุดทดสอบ สำหรับแต่ละชุดของพารามิเตอร์ที่ใส่ลงไป

สรุปมีตัวแปร 4 ตัวที่ต้องค้นหาค่าเอง ได้แก่ จำนวน Lag, ค่า σ ,ค่า C และค่า ε จำนวนการรันซ้ำ ปัจจุบันยังไม่มีกฎหรือทฤษฎีการเลือกพารามิเตอร์ที่เหมาะสม งานวิจัยชิ้นนี้จึงทำการค้นหา

พารามิเตอร์ในปริมาณที่มากเพื่อโอกาสที่เจอพารามิเตอร์ที่ทำให้ค่าพยากรณ์มีความแม่นยำสูงยิ่งขึ้น รวมจำนวนครั้งการรันทั้งสิ้น 16,940 ครั้ง

3.3.1 ตัวอย่างการพยากรณ์ข้อมูล Sunspot ด้วยตัวแบบ SVM

ขั้นตอนการสร้างตัวแบบ ANN สามารถอธิบายได้ดังภาพที่ 3.11

ภาพที่ 3.11 ขั้นตอนการสร้างตัวแบบ SVM

ขั้นตอนที่ 1 plot กราฟ ดูลักษณะข้อมูล ดังภาพที่ 3.1

ขั้นตอนที่ 2 แบ่งกลุ่มข้อมูลออกเป็นสองชุด คือ ข้อมูลชุดเรียนรู้จำนวน 221 ข้อมูล และข้อมูลชุดทดสอบมีจำนวน 67 ข้อมูล ดังภาพที่ 3.1

ขั้นตอนที่ 3 สร้างตัวแบบ SVM โดยค้นหาค่าพารามิเตอร์ทั้งสี่ เริ่มจากการใส่ค่า Lag, σ, ε และ C ตามลำดับทำการสร้างตัวแบบด้วยทุกชุดของค่าพารามิเตอร์ที่เป็นไปได้ ตัวอย่างเช่น ชุดพารามิเตอร์ที่หนึ่ง คือ Lag=1, σ=0.00001, ε=0.00001 และ C=1 ชุดพารามิเตอร์ที่สอง คือ Lag=1, σ=0.00001, ε=0.00001 และ C=5 ทำเช่นนี้ไปเรื่อยๆจนกว่าจะครบพารามิเตอร์ C แล้วเริ่มเปลี่ยนพารามิเตอร์ ε , σ และ Lag ตามลำดับ จนกระทั่งครบทุกพารามิเตอร์ทั้งหมด

ขั้นตอนที่ 4 ตัวแบบที่สร้างจากพารามิเตอร์แต่ละชุด นำไปพยากรณ์ เปรียบเทียบกับข้อมูลชุดทดสอบ และคำนวณหาค่า MSE

ตัวอย่างผลลัพธ์ค่า MSE ในข้อมูลชุดทดสอบในแต่ละชุดค่าพารามิเตอร์ของตัวแบบ SVM สำหรับข้อมูล Sunspot

เมื่อ Lag=1, σ=0.00001

MSE	Lag	Sigma	Epsilon	C
3166.52	1	1.00E-05	0.00001	1
3088.898	1	1.00E-05	0.00001	5
3015.471	1	1.00E-05	0.00001	10
2822.758	1	1.00E-05	0.00001	25
2758.969	1	1.00E-05	0.00001	30
2695.552	1	1.00E-05	0.00001	35
2475.274	1	1.00E-05	0.00001	50
2225.35	1	1.00E-05	0.00001	70
1996.866	1	1.00E-05	0.00001	100
1121.074	1	1.00E-05	0.00001	400
1000.216	1	1.00E-05	0.00001	800
970.409	1	1.00E-05	0.00001	1500
958.2009	1	1.00E-05	0.00001	2500
957.1852	1	1.00E-05	0.00001	3500

พารามิเตอร์ C เปลี่ยนไป

MSE	Lag	Sigma	Epsilon	C
3166.52	1	1.00E-05	0.0001	1
3088.494	1	1.00E-05	0.0001	5
3015.471	1	1.00E-05	0.0001	10
2822.758	1	1.00E-05	0.0001	25
2758.969	1	1.00E-05	0.0001	30
2695.552	1	1.00E-05	0.0001	35
2475.274	1	1.00E-05	0.0001	50
2225.35	1	1.00E-05	0.0001	70
1999.16	1	1.00E-05	0.0001	100
1121.074	1	1.00E-05	0.0001	400
1000.259	1	1.00E-05	0.0001	800
970.211	1	1.00E-05	0.0001	1500
958.1372	1	1.00E-05	0.0001	2500

พารามิเตอร์ Epsilon เปลี่ยนจาก 0.00001 เป็น 0.0001

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

957.2167	1	1.00E-05	0.0001	3500
...	...	...	...	...
930.612	1	1.00E-05	0.99	3500

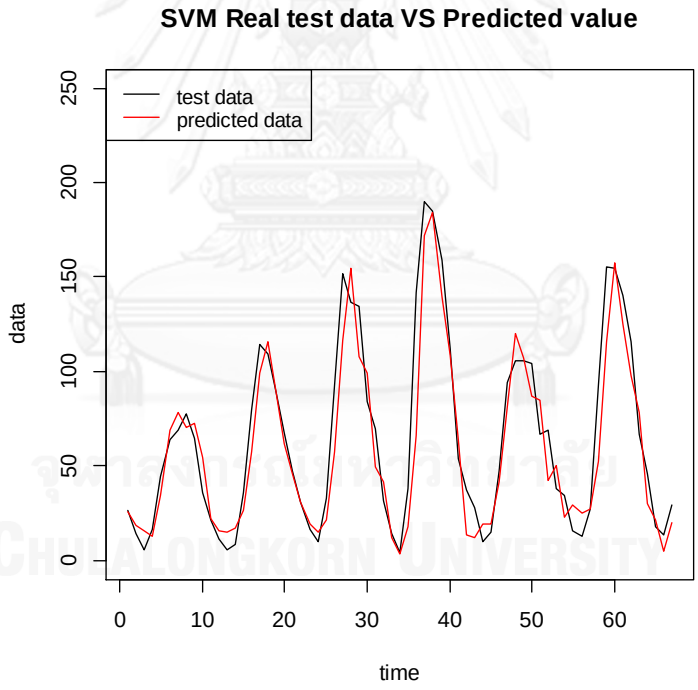
ตารางที่ 3.5 ค่า MSE ที่พารามิเตอร์ต่างๆ ของตัวแบบ SVM เมื่อ Lag=1 และ σ=0.00001

สำหรับกรณีที่ Lag=2,3,...,10 และค่า σ อื่นๆ ก็ทำในลักษณะเดียวกัน จนครบค่าพารามิเตอร์ทั้งหมด

ขั้นตอนที่ 5 เลือกค่าพารามิเตอร์ที่ดีที่สุดที่ให้ค่า MSE ต่ำที่สุด

ประกอบด้วย Lag=9, Sigma=0.001, Epsilon=0.5, C=70 มีค่า MSE เท่ากับ 262.17

ขั้นตอนที่ 6 ทำการพยากรณ์ข้อมูลล่วงหน้าในข้อมูลชุดทดสอบทั้ง 67 ข้อมูล ในรูปแบบการพยากรณ์หนึ่งค่าล่วงหน้า ดังภาพ 3.12 พร้อมทั้งประเมินประสิทธิภาพตัวแบบด้วย RMSE, MSE, MAE และ MAPE



ภาพที่ 3.12 เปรียบเทียบค่าจริงและค่าพยากรณ์ตัวแบบ SVM

ค่าพารามิเตอร์ที่ดีที่สุดของ ARIMA พร้อมทั้งตัวประเมินประสิทธิภาพตัวแบบ RMSE, MSE, MAE, MAPE และกราฟค่าพยากรณ์เปรียบเทียบกับค่าจริง จะแสดงไว้ในบทที่ 4

3.4 การสร้างตัวแบบ Traditional hybrid ARIMAANN และ ARIMASVM

ค่าเศษเหลือจากตัวแบบ ARIMA ของข้อมูล Sunspot แสดงดังตารางที่ 3.6

ค่าจริง	ค่าพยากรณ์ ARIMA	ค่าเศษเหลือ=ค่าจริง-ค่าพยากรณ์ ARIMA
5	20.513	-15.513
11	11.655	-0.655
16	23.992	-7.992
23	28.390	-5.390
36	34.213	1.786
58	47.935	10.064
29	64.730	-35.730
...	...	...
13.4	9.440	3.959
29.2	25.640	3.559

ตารางที่ 3.6 ค่าเศษเหลือที่ได้จากตัวแบบ ARIMA

มาพยากรณ์ด้วย ANN และ SVM โดยยึดขอบเขตและพารามิเตอร์ที่ใช้สร้างตัวแบบเดี่ยวทั้ง ANN และ SVM ขั้นตอนการสร้างตัวแบบก็จะเป็นดังขั้นการสร้างตัวแบบเดี่ยว ANN หรือ SVM เมื่อได้ค่าพยากรณ์ ANN หรือ SVM ก็นำค่าพยากรณ์นั้นรวมกับค่าพยากรณ์ ARIMA ก็จะได้ค่าพยากรณ์รวมทั้งหมด ดังตารางที่ 3.7

ค่าจริง	ค่าพยากรณ์ ARIMA	ค่าพยากรณ์ ANN	ค่าพยากรณ์รวม=ค่าพยากรณ์ ARIMA+ ค่าพยากรณ์ ANN	เศษเหลือรวม=ค่าจริง-ค่าพยากรณ์รวม
5	20.51	-	-	-
11	11.65	-	-	-
16	23.99	-	-	-
23	28.39	-	-	-
36	34.21	0.640	34.854	1.145
58	47.93	2.132	50.067	7.932
29	64.73	4.566	69.296	40.296
20	15.83	8.267	24.106	4.106
10	14.80	0.410	15.219	5.219
8	12.62	-4.623	7.998	0.0011
...	...	...	...	...

13.4	9.44	2.586	12.027	1.372
29.2	25.64	0.166	25.806	3.393

ตารางที่ 3.7 การได้มาซึ่งค่าพยากรณ์รวมและเศษเหลือรวมตัวแบบ ARIMAANN

จากตารางที่ 3.7 จะสังเกตได้ว่า 4 ค่าแรกของคอลัมน์ค่าพยากรณ์ ANN หายไปเนื่องมาจากค่า Lag ที่ดีที่สุดที่ได้จากการค้นหา คือ 4 ทำให้ข้อมูล 4 ค่าแรกหายไปจึงสามารถหาค่าพยากรณ์รวมและค่าเศษเหลือรวมได้ตั้งแต่ค่าที่ 5 เป็นต้นไป

สำหรับตัวแบบ ARIMASVM ก็มีขั้นตอนการผสมเช่นเดียวกับตัวแบบ ARIMAANN ซึ่งแสดงเป็นตัวอย่างด้านบน

3.5 การสร้างตัวแบบ Hybrid combined (ARIMA+ANN+SVM)

ทำการพยากรณ์ด้วย ARIMA, ANN และ SVM แยกส่วนกันทำในข้อมูลชุด Sunspot จะได้ค่าพยากรณ์ดังตารางที่ 3.8

ค่าพยากรณ์ ARIMA	ค่าพยากรณ์ ANN	ค่าพยากรณ์ SVM
20.5133475	-	-
11.6557421	-	-
23.9927874	-	-
28.3901858	-	-
34.2135212	-	-
47.9353797	-	-
64.730019	-	-
15.8388013	17.62951544	-
14.8086087	10.76413951	-
12.6222774	9.072261016	10.12961695
9.4960273	8.841471854	7.755973895
10.176785	8.556602506	7.936703328
7.4258805	9.552422872	8.588090646
14.6034276	7.847576235	10.76932262
17.3241671	12.26789354	18.85742261
...	...	...
9.440	18.089	7.968
25.640	20.271	18.8

ตารางที่ 3.8 ค่าพยากรณ์ของตัวแบบ ARIMA, ANN และ SVM

แล้วนำค่าทั้งสามค่ารวมกันโดยใส่ค่า weights ในแต่ละตัวแบบ ค้นหาค่า Weights ที่ดีที่สุดในข้อมูลชุดทดสอบภายใต้เงื่อนไขค่า MSE มีค่าต่ำสุด

จากตารางที่ 3.8 ค่าพยากรณ์ทั้งสามตัวแบบจะถูกตัดออกไปตามจำนวน Lag ที่ใช้มากที่สุด ซึ่งก็คือจำนวน Lag ของ SVM มีค่าเท่ากับ 9 ดังนั้นจะสามารถเริ่มสร้างตัวแบบ Combined ได้ตั้งแต่ค่าพยากรณ์ในแถวที่ 10 เป็นต้นไป

ค่า Weights ที่ใช้ในตัวแบบ Combined จะขึ้นอยู่กับจำนวนตัวแบบที่นำมาใช้ สำหรับงานวิจัยชิ้นนี้นำเสนอตัวแบบ Combined (ARIMA+ANN+SVM) ดังนั้นจะมีค่า Weights จำนวน 3 ค่าคือ ค่า Weight ของ ARIMA, ค่า Weight ของ ANN และค่า Weight ของ SVM ทำการค้นหาชุดของค่า Weights ทั้งสามที่ดีที่สุดที่ทำให้ค่า MSE ต่ำที่สุด ในรูปแบบสมการผลรวมเชิงเส้นดังสมการที่ 2.28 และ 2.29 การค้นหาค่า Weights เริ่มแรกจะต้องกำหนดตำแหน่งทศนิยมเสียก่อน เช่นกำหนดตำแหน่งทศนิยมระดับที่สาม ค่า Weights จะมีค่าตั้งแต่ 0, 0.001, 0.002, 0.003,...,0.998, 0.999, 1 วิธีการค้นหาในลักษณะนี้เรียกว่าการค้นหาค่า Weights ทุกชุดทั้งหมด (All possible combination) ดังแสดงดังตารางที่ 3.9

ค่า Weight ชุดที่	Weight ARIMA	Weight ANN	Weight SVM	Total weight
1	0	1	0	1
2	0	0.999	0.001	1
3	0	0.998	0.002	1
4	0	0.997	0.003	1
...	...	...	...	...
1000	0	0.001	0.999	1
1001	0	0	1	1

ตารางที่ 3.9 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0

จากตารางที่ 3.9 กำหนดค่า Weight ของ ARIMA=0 ขณะที่ค่า Weight ANN และ SVM จะลดลงและเพิ่มขึ้นไปตามระดับทศนิยมที่กำหนด ในที่นี้กำหนดไว้เป็นสาม โดยที่ผลรวมของค่า Weight ในทุกชุดจะมีค่าเป็น 1

ค่า Weight ชุดที่	Weight ARIMA	Weight ANN	Weight SVM	Total weight
1	0.001	0.999	0	1
2	0.001	0.998	0.001	1
3	0.001	0.997	0.002	1
4	0.001	0.996	0.003	1

...	...	...	...	...
1000	0.001	0.002	0.997	1
1001	0.001	0.001	0.998	1

ตารางที่ 3.10 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.001

จากตารางที่ 3.10 กำหนดค่า Weight ของ ARIMA=0.001 ขณะที่ค่า Weight ANN และ SVM จะลดลงละเพิ่มขึ้นไปตามระดับทศนิยมที่กำหนด ในที่นี้กำหนดไว้เป็นสาม โดยที่ผลรวมของค่า Weight ในทุกชุดจะมีค่าเป็น 1

ค่า Weight ของ ARIMA ก็จะเพิ่มขึ้นครั้งละ 0.001 ตามระดับทศนิยมที่กำหนดไว้ ส่วนค่า Weight ANN และ SVM ก็ลดลงและเพิ่มขึ้นในรูปแบบเดิมครั้งละ 0.001 ดังแสดงในตารางที่ 3.11-3.13

ค่า Weight ชุดที่	Weight ARIMA	Weight ANN	Weight SVM	Total weight
1	0.998	0.002	0	1
2	0.998	0.001	0.001	1
3	0.998	0	0.002	1

ตารางที่ 3.11 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.998

ค่า Weight ชุดที่	Weight ARIMA	Weight ANN	Weight SVM	Total weight
1	0.999	0.001	0	1
2	0.999	0	0.001	1

ตารางที่ 3.12 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=0.999

ค่า Weight ชุดที่	Weight ARIMA	Weight ANN	Weight SVM	Total weight
1	1	0	0	1

ตารางที่ 3.13 ค่า Weight ของตัวแบบ Combined เมื่อค่า Weight ของ ARIMA=1

นำค่าพยากรณ์จากตารางที่ 3.8 ตั้งแต่แถวที่ 10 เป็นต้นไปคูณกับค่า Weight ของแต่ละตัวแบบแล้วนำมาบวกกันทั้งหมดก็จะได้ค่าพยากรณ์รวม เพราะฉะนั้นจะมีชุดของค่าพยากรณ์รวมเท่ากับจำนวนชุดของค่า Weight จากนั้นหาค่าเศษเหลือรวมและคำนวณหาค่า MSE และเลือกชุดของค่า Weight ที่ให้ค่า MSE ต่ำสุดเพื่อนำไปใช้พยากรณ์ข้อมูลในอนาคตต่อไป

บทที่ 4

ผลการทดลองและการวิเคราะห์

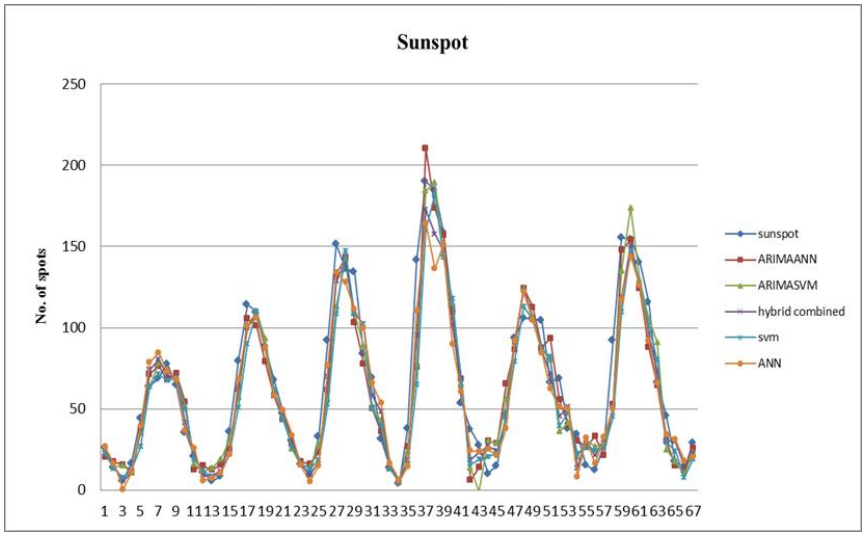
4.1 ผลการทดลอง

เปรียบเทียบค่า RMSE, MSE, MAE, MAPE ในข้อมูลชุดทดสอบ 10 ข้อมูลทั้ง 6 ตัวแบบ รวมทั้งค่าพารามิเตอร์ที่ดีที่สุดในกรอบการค้นหาของแต่ละตัวแบบ ตารางและภาพ 4.1-4.10

1) ข้อมูล Sunspot

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	9	0	0		17.575	308.893	13.046	31.509
ANN	Lag	Hidden neurons	Output neurons					
	7	9	1		15.381	236.6	11.489	26.495
SVM	Lag	Sigma	Epsilon	C				
	9	0.001	0.5	80	18.297	334.81	13.247	31.191
ARIMAANN	Lag	Hidden neurons	Output neurons					
	4	8	1		16.052	257.693	11.843	31.572
ARIMASVM	Lag	Sigma	Epsilon	C				
	1	10	0.99	30	16.697	278.821	11.891	31.372
Hybrid combined					***14.861***	***220.867***	***11.017***	***24.033***

ตารางที่ 4.1 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Sunspot



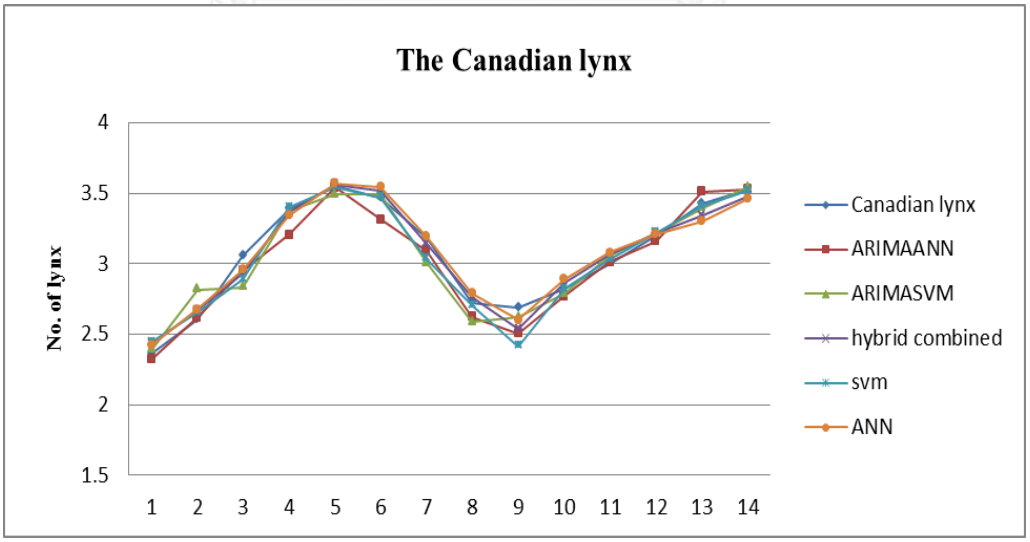
ภาพที่ 4.1 ค่าพยากรณ์ทั้ง 6 ตัวแบบ ของข้อมูล Sunspot

จากตารางที่ 4.1 ตัวแบบ Combined ให้ผลการพยากรณ์ที่มีความแม่นยำมากที่สุดและแม่นยำกว่าตัวแบบ Hybrid ในงานวิจัยของ (Zhang, 2003) มีค่า MSE เท่ากับ 280.15

2) ข้อมูล Canadian lynx

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	12	0	0		0.1522	0.02315	0.1123	3.7181
ANN	Lag	Hidden neurons	Output neurons					
	4	9	1		0.0704	0.00495	0.0628	2.105
SVM	Lag	Sigma	Epsilon	C				
	2	0.5	0.1	80	0.0993	0.00986	0.0607	2.1344
ARIMAANN	Lag	Hidden neurons	Output neurons					
	4	9	1		0.0969	0.0094	0.0768	2.5231
ARIMASVM	Lag	Sigma	Epsilon	C				
	2	0.1	0.00001	2000	0.1076	0.01157	0.0782	2.6758
Hybrid combined					***0.0704***	***0.004949***	***0.0594***	***2.0271***

ตารางที่ 4.2 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Canadian lynx



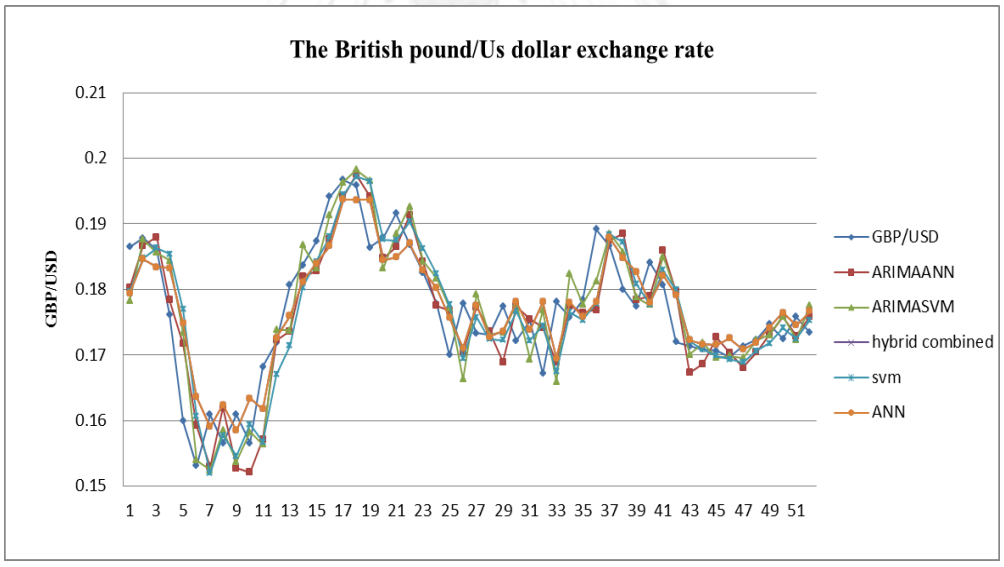
ภาพที่ 4.2 ค่าพยากรณ์ทั้ง 6 ตัวแบบ ของข้อมูล Canadian lynx

จากตารางที่ 4.2 ตัวแบบ Combined ให้ผลการพยากรณ์ที่มีความแม่นยำมากที่สุดและแม่นยำกว่าตัวแบบ Hybrid ในงานวิจัยของ (Zhang, 2003) มีค่า MSE เท่ากับ 0.017

3) ข้อมูล GBP/USD

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	1	0	0		0.005859	3.4329*10-5	0.004761	2.7239
ANN	Lag	Hidden neurons	Output neurons					
	10	9	1		0.005305	2.8146*10-5	0.004192	2.4007
SVM	Lag	Sigma	Epsilon	C				
	6	0.001	0.01	28	0.005879	3.4570*10-5	0.004636	2.651
ARIMAANN	Lag	Hidden neurons	Output neurons					
	9	10	1		0.005438	2.9575*10-5	0.004451	2.5467
ARIMASVM	Lag	Sigma	Epsilon	C				
	5	0.01	0.01	95	0.005754	3.3107*10-5	0.004535	2.5857
Hybrid combined					***0.005305***	***2.8146*10-5***	***0.00419***	***2.4007***

ตารางที่ 4.3 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล GBP/USD



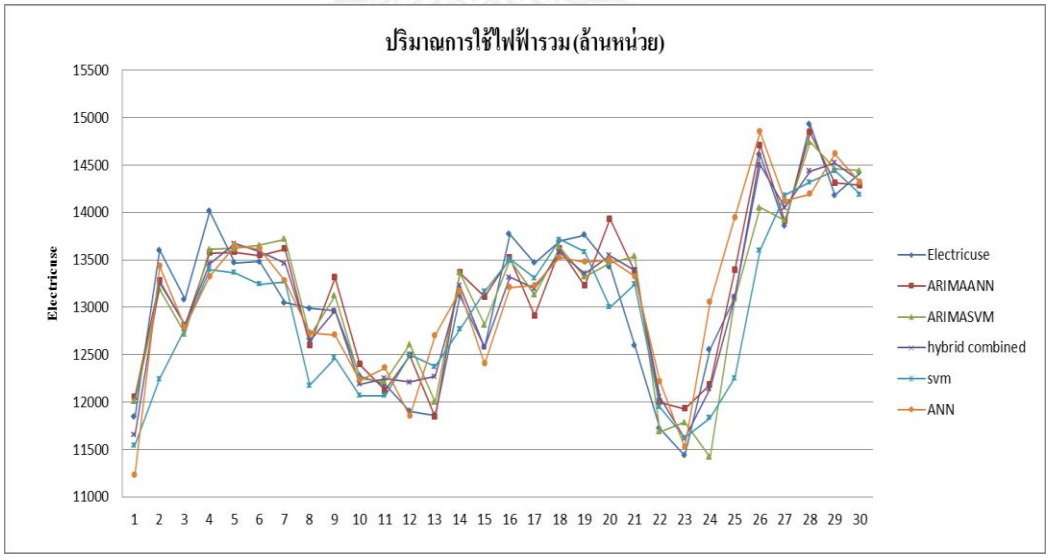
ภาพที่ 4.3 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล GBP/USD

จากตารางที่ 4.3 ตัวแบบ Combined ให้ผลการพยากรณ์ที่มีความแม่นยำมากที่สุดและแม่นยำกว่าตัวแบบ Hybrid ในงานวิจัยของ (Zhang, 2003) มีค่า MSE เท่ากับ $4.359*10^{-5}$

4) ข้อมูล Electricuse

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	13	0	0		423.707	179527.8	328.631	2.516
ANN	Lag	Hidden neurons	Output neurons					
	9	2	1		410.302	168347.7	329.784	2.522
SVM	Lag	Sigma	Epsilon	C				
	10	0.0001	0.5	2500	524.438	275036.1	430.972	3.274
ARIMAANN	Lag	Hidden neurons	Output neurons					
	5	9	1		358.961	128853.3	297.928	2.304
ARIMASVM	Lag	Sigma	Epsilon	C				
	1	0.1	0.001	500	401.952	161565.4	296.811	2.281
Hybrid combined					***315.6592***	***99640.73***	***256.869***	***1.958***

ตารางที่ 4.4 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Electricuse

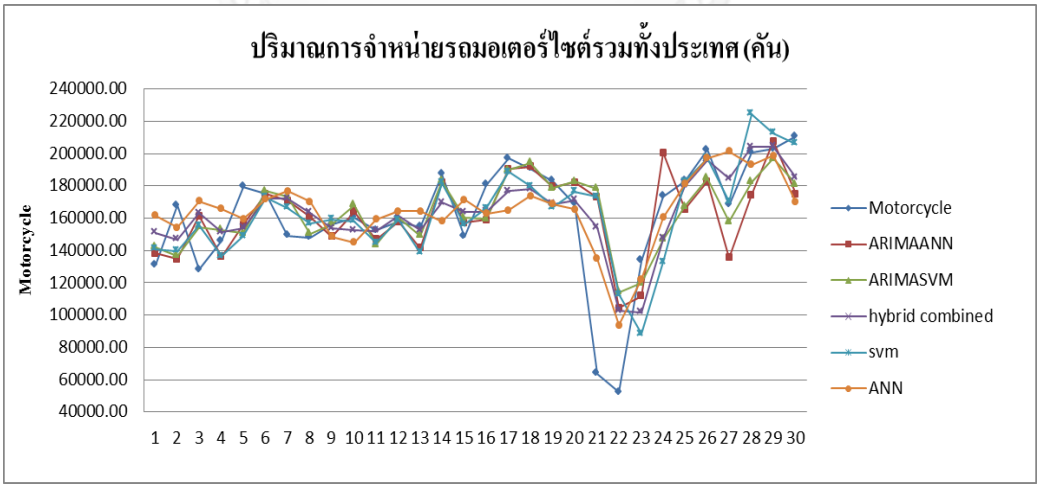


ภาพที่ 4.4 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Electricuse

5) ข้อมูล Motorcycle

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	12	1	0		32115.41	1031400000	20631.15	18.714
ANN	Lag	Hidden neurons	Output neurons					
	8	4	1		24882.83	619155247.7	19691.409	16.158
SVM	Lag	Sigma	Epsilon	C				
	10	0.001	0.1	4000	28433.43	808459800	17841.23	17.127
ARIMAANN	Lag	Hidden neurons	Output neurons					
	9	6	1		28413.65	807335252.1	19276.561	17.397
ARIMASVM	Lag	Sigma	Epsilon	C				
	1	0.5	0.001	1000	28328.32	802493500	17927.2	17.146
Hybrid combined					***24882.59***	***619143533***	***17248.7***	***15.707***

ตารางที่ 4.5 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Motorcycle

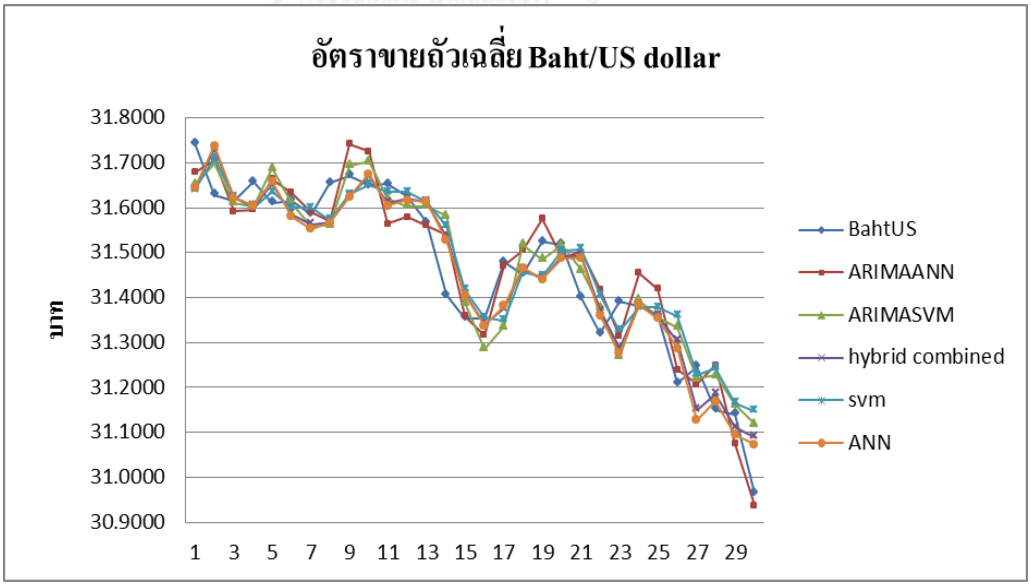


ภาพที่ 4.5 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Motorcycle

6) ข้อมูล USexchange

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	4	1	2		0.07897	0.006237	0.0628	0.2
ANN	Lag	Hidden neurons	Output neurons					
	9	7	1		0.06748	0.004554	0.05607	0.1783
SVM	Lag	Sigma	Epsilon	C				
	2	0.0001	0.001	800	0.0757	0.00573	0.05638	0.1796
ARIMAANN	Lag	Hidden neurons	Output neurons					
	5	9	1		0.06407	0.004105	0.05534	0.1759
ARIMASVM	Lag	Sigma	Epsilon	C				
	2	30	1	1	0.07518	0.005652	0.0593	0.1887
Hybrid combined					***0.06748***	***0.004553***	***0.05507***	***0.1752***

ตารางที่ 4.6 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล USexchange

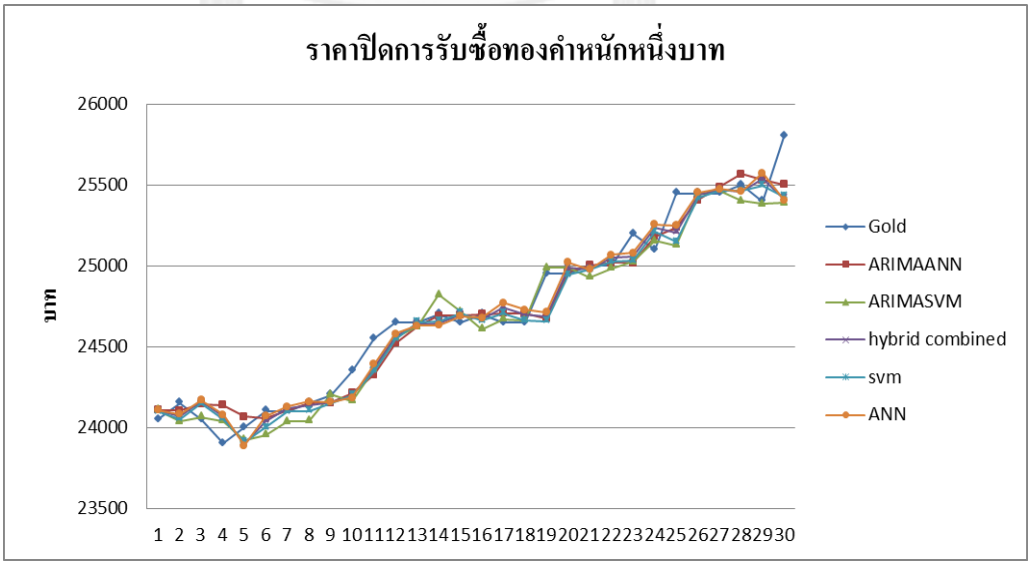


ภาพที่ 4.6 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล USexchange

7) ข้อมูล Gold

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q				104.5346	
	3	1	2		147.3189	21702.86		0.42
ANN	Lag	Hidden neurons	Output neurons					
	8	5	1		127.1756	16173.6319	96.4675	0.3883
SVM	Lag	Sigma	Epsilon	C				
	2	0.1	0.0001	1	131.5889	17315.6404	94.301	0.3797
ARIMAANN	Lag	Hidden neurons	Output neurons		***122.1837***	***14928.859***	***88.1417***	***0.355***
	3	4	1					
ARIMASVM	Lag	Sigma	Epsilon	C				
	3	10	0.00001	30	130.5995	17056.2408	93.3761	0.3762
Hybrid combined					127.1734	16173.08	94.7072	0.3812

ตารางที่ 4.7 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Gold

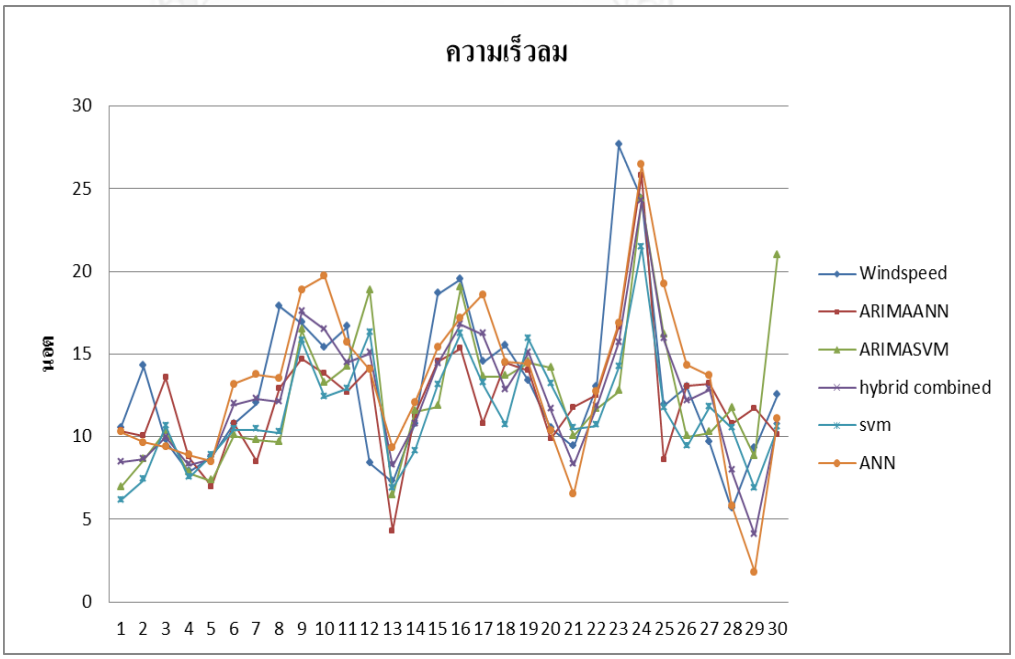


ภาพที่ 4.7 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Gold

8) ข้อมูล Windspeed

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	0	1	6		4.5581	20.7759	3.4757	28.177
ANN	Lag	Hidden neurons	Output neurons					
	8	5	1		3.6725	13.4871	2.6948	21.0637
SVM	Lag	Sigma	Epsilon	C				
	2	0.1	0.0001	1	4.2594	18.1427	3.1649	24.2791
ARIMAANN	Lag	Hidden neurons	Output neurons					
	3	4	1		3.5459	12.5736	2.7564	22.7907
ARIMASVM	Lag	Sigma	Epsilon	C				
	3	10	0.00001	30	4.7172	22.2521	3.1398	25.3531
Hybrid combined					***3.5129***	***12.3403***	***2.4324***	***18.928***

ตารางที่ 4.8 ค่า RMSE, MSE, MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Windspeed



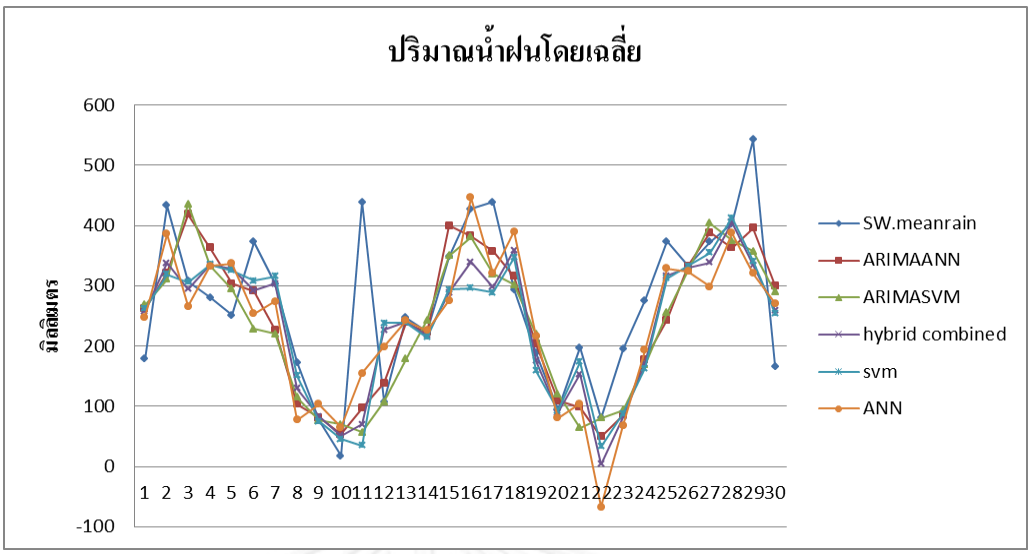
ภาพที่ 4.8 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Windspeed

9) ข้อมูล Meanrain

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	Q					
	5	0	4		112.66	12692.381	82.8018	39.564
ANN	Lag	Hidden neurons	Output neurons					
	6	9	1		97.7607	9557.1494	75.5065	41.7946
SVM	Lag	Sigma	Epsilon	C				
	10	0.1	0.00001	1	106.565	11356.182	69.9311	***31.233***
ARIMAANN	Lag	Hidden neurons	Output neurons					
	8	3	1		***96.8125***	***9372.6604***	70.8839	32.7273
ARIMASVM	Lag	Sigma	Epsilon	C				
	1	0.005	0.01	400	108.19	11705.139	77.0279	36.0095
Hybrid combined					**101.298**	**10261.375**	***69.6915***	**33.5568**

ตารางที่ 4.9 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Meanrain

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

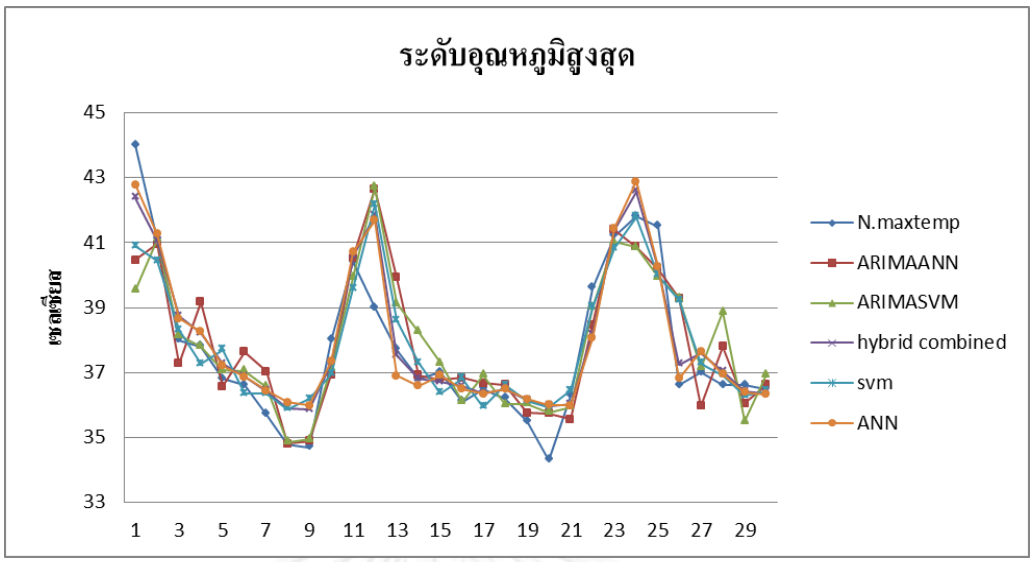


ภาพที่ 4.9 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Meanrain

10) ข้อมูล Maxtemp

	Parameter				RMSE	MSE	MAE	MAPE
ARIMA	p	d	q					
	15	0	0		1.4995	2.24839	1.0467	2.7234
ANN	Lag	Hidden neurons	Output neurons					
	3	4	1		0.9005	0.8109	0.6845	1.8119
SVM	Lag	Sigma	Epsilon	C				
	2	0.1	0.1	500	1.1892	1.4142	0.8744	2.3031
ARIMAANN	Lag	Hidden neurons	Output neurons					
	6	5	1		1.3503	1.82339	0.9626	2.5171
ARIMASVM	Lag	Sigma	Epsilon	C				
	3	0.05	0.00001	50	1.4178	2.01003	0.9193	2.3911
Hybrid combined					***0.9005***	***0.81089***	***0.6644***	***1.7564***

ตารางที่ 4.10 ค่า RMSE,MSE,MAE และ MAPE และค่าพารามิเตอร์ข้อมูล Maxtemp



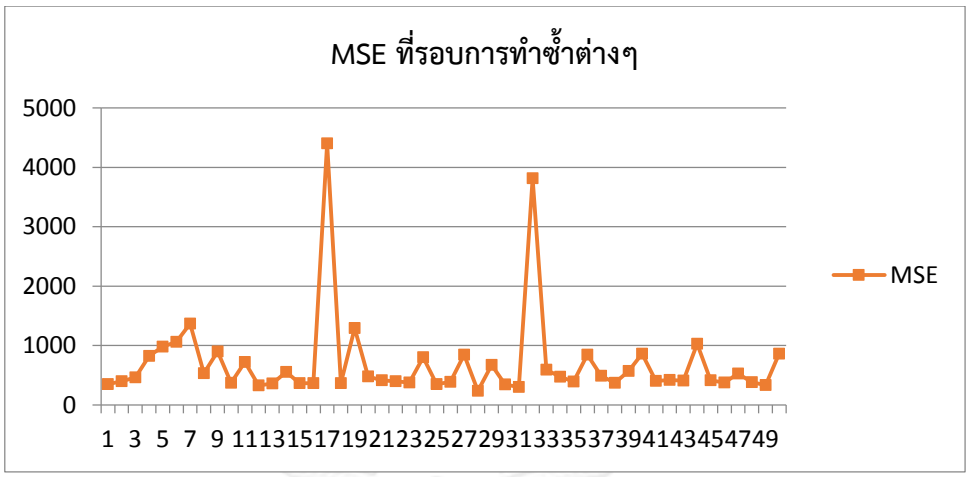
ภาพที่ 4.10 ค่าพยากรณ์ทั้ง 6 ตัวแบบของข้อมูล Meanrain

4.2 การวิเคราะห์ความไวสำหรับตัวแบบ ANN

ตัวแบบ ANN เป็นตัวแบบที่ต้องทดลองแทนค่าพารามิเตอร์เข้าไปในตัวแบบ ปัจจุบันก็ไม่มีวิธีการคัดเลือกพารามิเตอร์เพื่อให้ได้ตัวแบบพยากรณ์ที่ดีที่สุด การเลือกพารามิเตอร์จึงเป็นงานการทดลองเพื่อให้ได้ค่าพารามิเตอร์ที่ทำให้ค่าพยากรณ์มีความแม่นยำ มีความคลาดเคลื่อน (Error) ต่ำ หัวข้อนี้ทำการศึกษาการวิเคราะห์ความไวของข้อมูล Sunspot

ก) ความไวของการทำซ้ำ จำนวนรอบการทำซ้ำเป็นปัจจัยที่สำคัญที่ทำส่งผลต่อประสิทธิภาพตัวแบบ เนื่องจากการทำซ้ำ 1 รอบ จะได้ผลลัพธ์ที่ไม่เท่ากัน เนื่องจากเกี่ยวข้องกับค่า Weights เริ่มต้น ที่ตัวแบบ ANN จะสร้างขึ้นแล้วปรับค่า Weights ในภายหลัง การวิเคราะห์ความไวของรอบการทำซ้ำเป็นเครื่องมือที่แสดงให้เห็นถึงปัญหา Overfitting ได้ดี

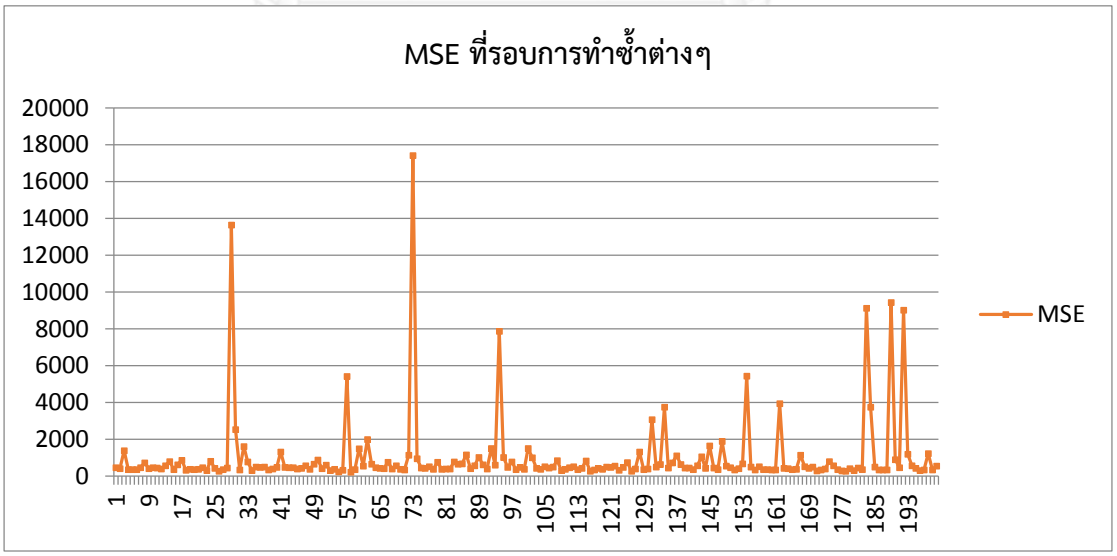
ตัวอย่างการวิเคราะห์ความไวของค่าพารามิเตอร์ในข้อมูล Sunspot ซึ่งมีโครงข่ายที่ดีที่สุด คือ 7-9-1



ภาพที่ 4.11 ค่า MSE ที่รอบการทำซ้ำต่างๆทั้ง 50 รอบ

จากภาพที่ 4.11 แสดงค่า MSE ที่รอบการทำซ้ำ จำนวน 50 รอบ สำหรับตัวแบบ 7-9-1 จะสังเกตได้ว่ารอบการทำซ้ำจะมีผลต่อค่า MSE มีค่าสุดโต่งสองค่า คือที่รอบการทำซ้ำที่ 17 และ 32 แสดงให้เห็นว่าที่รอบที่ 17 และ 32 ตัวแบบเรียนรู้ดีกับข้อมูลชุดเรียนรู้แต่เมื่อพยากรณ์กับข้อมูลชุดทดสอบพบว่าให้ค่าพยากรณ์ที่มีความคลาดเคลื่อนสูงเรียกว่าปัญหา Overfitting ในขณะที่รอบที่ดีที่สุด คือ รอบที่ 28 ซึ่งมีค่า MSE เท่ากับ 236.6 ขณะที่ค่า MSE ในแต่ละรอบก็แกว่งในช่วง 300-600 เป็นส่วนใหญ่ บอกให้ทราบว่าค่า Initial weights ที่มาจากการทำซ้ำในแต่ละรอบมีผลต่อการสร้างตัวแบบและค่าพยากรณ์ที่ได้

เมื่อขยายรอบการทำซ้ำเพิ่มอีก 200 รอบ ได้ผลค่า MSE ดังภาพด้านล่าง

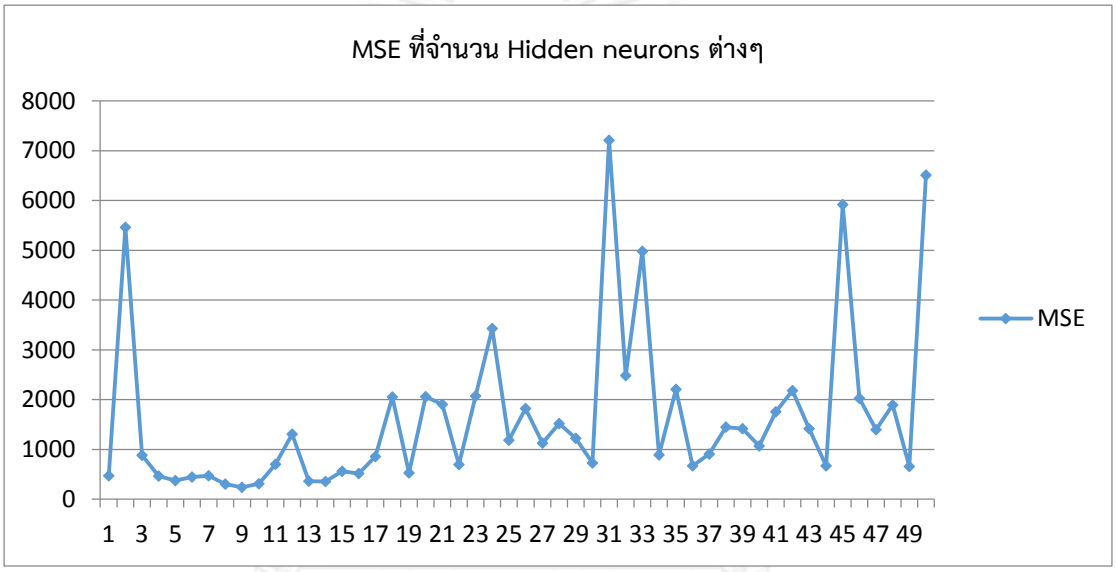


ภาพที่ 4.12 ค่า MSE ที่รอบการทำซ้ำต่างๆตั้งแต่รอบที่ 51 ถึงรอบที่ 250

ค่า MSE ที่ได้จากการทำซ้ำตั้งแต่รอบที่ 51 ถึง 250 นั้นมีค่าต่ำสุด คือ 235.79 น้อยกว่าค่า MSE ที่น้อยที่สุดในรอบ 50 รอบ เพียงเล็กน้อย นั่นแสดงให้เห็นว่าแม้ว่าจะมีจำนวนรอบการทำซ้ำที่มากขึ้นแต่ก็อาจทำให้ค่า MSE ลดลงเพียงเล็กน้อยหรืออาจไม่ลดลง แต่ต้องแลกกับเวลาที่ใช้คำนวณที่มากขึ้น โดยเวลาในการคำนวณสำหรับการทำซ้ำ 50 รอบของโครงข่าย 7-9-1 คือ 35.68 วินาที สำหรับการทำซ้ำ 200 รอบใช้เวลา 125.87 วินาที เพิ่มขึ้น 4 เท่าโดยประมาณ

ข) ความไวของจำนวน Hidden neurons

ทำการศึกษาค่า MSE โดยที่กำหนด Lag และรอบการทำซ้ำเดียวกัน โดยขยายจำนวน Hidden neurons จากเดิมที่ทำการศึกษาไว้ตั้งแต่ 1 ถึง 10 เพิ่มไปจนถึง 1 ถึง 50



ภาพที่ 4.13 ค่า MSE ที่จำนวน Hidden neurons ตั้งแต่ 1-50

จากภาพที่ 4.13 เมื่อจำนวน Hidden neurons เพิ่มขึ้นค่า MSE เพิ่มมากขึ้นโดยเฉพาะตั้งแต่จำนวน Hidden neurons=16 และมีค่าสุดโต่งมากขึ้น แสดงให้เห็นว่าตัวแบบเรียนรู้ดีเกินไปเกิดปัญหา Overfitting นั่นแสดงให้เห็นว่าจำนวน Hidden neurons ที่มีจำนวนมากไม่ได้มีผลต่อค่า MSE ที่ลดลง

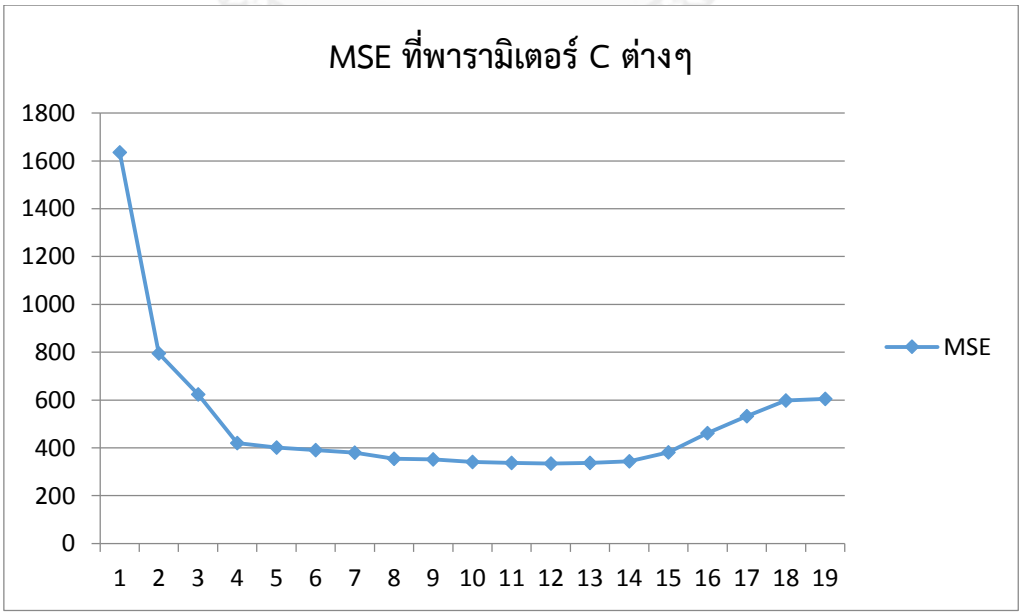
4.3 การวิเคราะห์ความไวของพารามิเตอร์ สำหรับตัวแบบ SVM

ตัวแบบ SVM เป็นตัวแบบที่มีพารามิเตอร์ที่นำมาสร้างตัวแบบ ปัจจุบันก็ไม่มีวิธีการคัดเลือกพารามิเตอร์เพื่อให้ได้ตัวแบบพยากรณ์ที่ดีที่สุด การเลือกพารามิเตอร์จึงเป็นงานการทดลอง เพื่อให้ได้ค่าพารามิเตอร์ที่ทำให้ค่าพยากรณ์มีความแม่นยำมีความคลาดเคลื่อน (Error) ต่ำ หัวข้อนี้ทำการศึกษาการวิเคราะห์ความไวของข้อมูล Sunspot

ก) ความไวของพารามิเตอร์ C เมื่อกำหนดพารามิเตอร์อื่นคงที่ ได้ค่า MSE ดังแสดงดังตาราง 4.11 และภาพที่ 4.14

MSE	Lag	Sigma	Epsilon	C
1636.188	9	0.001	0.5	1
794.4962	9	0.001	0.5	5
624.0148	9	0.001	0.5	10
420.3833	9	0.001	0.5	25
400.6547	9	0.001	0.5	30
390.4284	9	0.001	0.5	35
380.5226	9	0.001	0.5	50
354.5837	9	0.001	0.5	60
352.2213	9	0.001	0.5	65
340.9041	9	0.001	0.5	70
336.4092	9	0.001	0.5	75
334.8158	9	0.001	0.5	80
336.5139	9	0.001	0.5	85
344.2707	9	0.001	0.5	100
380.9676	9	0.001	0.5	400
462.0905	9	0.001	0.5	800
532.0052	9	0.001	0.5	1500
597.74	9	0.001	0.5	2500
605.1981	9	0.001	0.5	3500

ตารางที่ 4.11 ค่า MSE ที่พารามิเตอร์ C มีค่าต่างๆ



ภาพที่ 4.14 ค่า MSE ที่พารามิเตอร์ C ต่างๆ

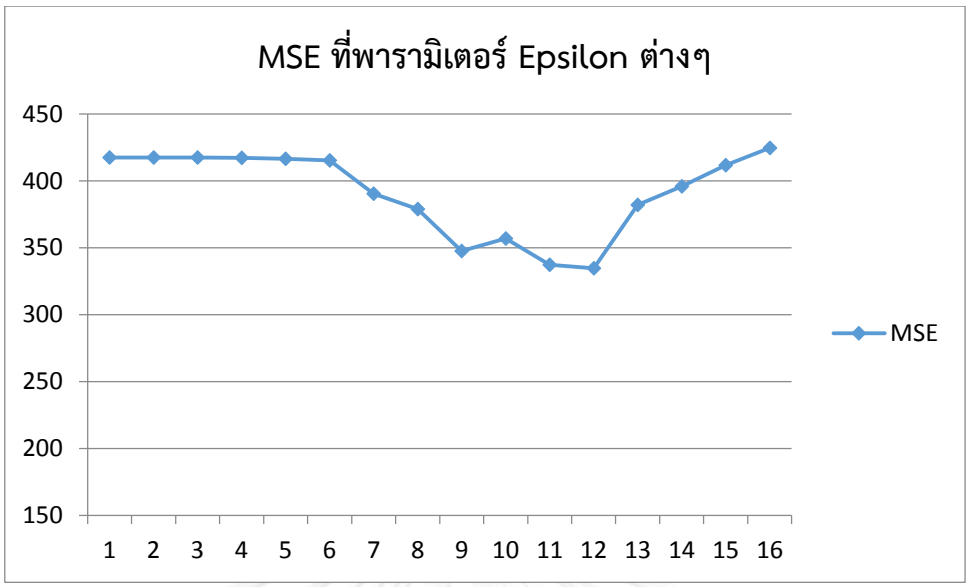
จากตารางที่ 4.11 และภาพที่ 4.14 ค่า MSE มีค่า 336.40, 334.81 และ 336.51 ที่พารามิเตอร์ C เท่ากับ 75, 80 และ 85 ตามลำดับ แม้ว่าค่า MSE ในช่วงแรกจะมีค่ามากแต่เมื่อ C มีค่าตั้งแต่ 25 เป็นต้นไป ค่า MSE จะค่อยๆลดลง จนถึงที่ C เท่ากับ 800 จากนั้นค่า MSE จะเพิ่มขึ้น

ข) ความไวของพารามิเตอร์ Epsilon เมื่อกำหนดพารามิเตอร์อื่นคงที่ ได้ค่า MSE ดังแสดงดังตาราง 4.12 และภาพที่ 4.15

MSE	Lag	Sigma	Epsilon	C
417.5621	9	0.001	0.00001	80
417.5743	9	0.001	0.0001	80
417.5566	9	0.001	0.0005	80
417.2538	9	0.001	0.001	80
416.5314	9	0.001	0.005	80
415.2354	9	0.001	0.01	80
390.4118	9	0.001	0.05	80
378.9117	9	0.001	0.1	80
347.554	9	0.001	0.2	80
356.9612	9	0.001	0.3	80
337.2875	9	0.001	0.4	80
334.8158	9	0.001	0.5	80
382.1395	9	0.001	0.6	80
396.0341	9	0.001	0.7	80
411.8512	9	0.001	0.8	80
424.6278	9	0.001	0.99	80

ตารางที่ 4.12 ค่า MSE ที่พารามิเตอร์ Epsilon มีค่าต่างๆ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 4.15 ค่า MSE ที่พารามิเตอร์ Epsilon ต่างๆ

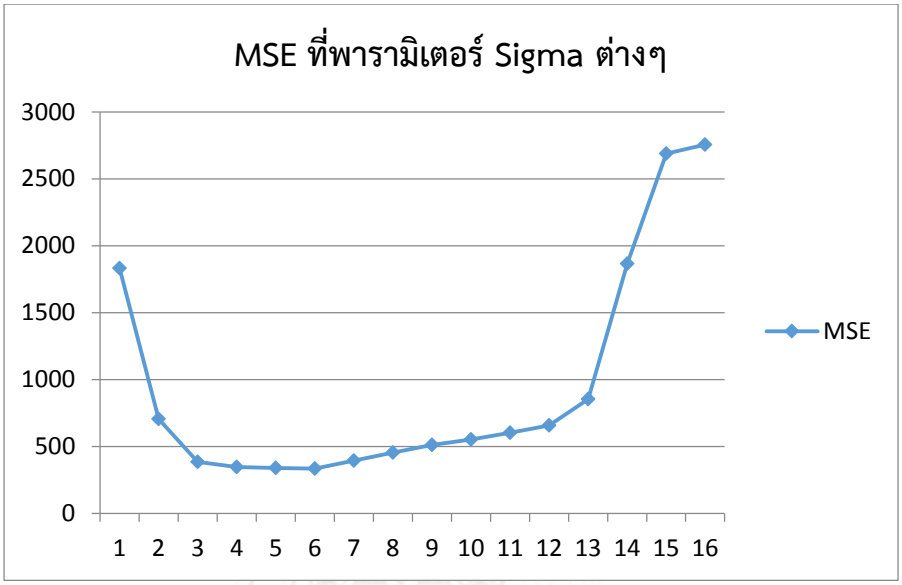
จากตารางที่ 4.12 และภาพที่ 4.15 ค่า MSE มีค่า 337.2875, 334.8158 และ 382.1395 ที่พารามิเตอร์ Epsilon เท่ากับ 0.4, 0.5 และ 0.6 ตามลำดับ ค่า MSE ในแต่ละค่า Epsilon มีผลต่อค่า MSE น้อย ค่า MSE มีค่าคงที่มาตลอดจนถึงค่า Epsilon เท่ากับ 0.4, 0.5 และ 0.6 จากนั้นค่า MSE จะเพิ่มขึ้น

ค) ความไวของพารามิเตอร์ Sigma

MSE	Lag	Sigma	Epsilon	C
1831.716	9	1.00E-05	0.5	80
704.2686	9	1.00E-04	0.5	80
383.9868	9	5.00E-04	0.5	80
345.5464	9	8.00E-04	0.5	80
339.0493	9	9.00E-04	0.5	80
334.8158	9	1.00E-03	0.5	80
395.0046	9	2.00E-03	0.5	80
452.9034	9	3.00E-03	0.5	80
510.585	9	4.00E-03	0.5	80
551.2536	9	5.00E-03	0.5	80
603.2042	9	1.00E-02	0.5	80
657.9245	9	5.00E-02	0.5	80
853.7758	9	1.00E-01	0.5	80
1865.074	9	1.00E+00	0.5	80

2688.126	9	5.00E+00	0.5	80
2754.496	9	1.00E+01	0.5	80

ตารางที่ 4.13 ค่า MSE ที่พารามิเตอร์ Sigma มีค่าต่างๆ



ภาพที่ 4.16 ค่า MSE ที่พารามิเตอร์ Sigma ต่างๆ

เมื่อกำหนดพารามิเตอร์อื่นคงที่ ได้ค่า MSE ดังแสดงดังตาราง 4.13 และภาพที่ 4.16 ค่า MSE มีค่า 339.0493, 334.8158 และ 395.0046 ที่พารามิเตอร์ Sigma เท่ากับ 0.0009, 0.001 และ 0.002 ตามลำดับ ค่า MSE ในแต่ละค่า Sigma มีผลต่อค่า MSE พอสมควร ลักษณะกราฟจะเป็นรูปหม้อช่วงแรกค่า MSE จะสูงและลดลงจนอยู่ในช่วงเกือบจะคงที่และเพิ่มขึ้นในตอนท้าย

เมื่อดูถึงขนาดการเปลี่ยนแปลงค่าของทั้งสามพารามิเตอร์ Sigma, Epsilon และ C พบว่าปัจจัยที่ส่งผลต่อค่า MSE มาก คือ Sigma และ C ขณะที่ Epsilon ไม่ได้มีผลต่อการเปลี่ยนแปลงค่า MSE มากนัก

เมื่อดูความไวของพารามิเตอร์ทั้งตัวแบบ ANN และ SVM พบว่ามีความแตกต่างกัน กล่าวคือ จากภาพที่ 4.11, 4.12 ตัวแบบ ANN ค่า MSE จะเปลี่ยนไปเมื่อพารามิเตอร์เปลี่ยนไปค่อนข้างมาก มองไม่เห็นแนวโน้มของพารามิเตอร์ที่จะให้ค่า MSE ต่ำสุด ขณะที่ตัวแบบ SVM จากภาพที่ 4.14-4.16 พบว่าสามารถมองเห็นแนวโน้ม หรือช่วงของพารามิเตอร์ที่จะทำให้ค่า MSE ต่ำที่สุด ดังนั้นการค้นหาพารามิเตอร์ที่ดีที่สุดของตัวแบบ SVM สามารถทำได้ง่ายกว่า

4.4 เวลาที่ใช้ในการคำนวณของแต่ละตัวแบบ

เวลาในการพยากรณ์ข้อมูลอนุกรมเวลาโดยปกติจะไม่มากนัก มักไม่เกิน 2 ชั่วโมง เนื่องจากข้อมูลอนุกรมเวลาจะไม่มาก อยู่ในหลักร้อยหรือหลักพัน ไม่มีความจำเป็นต้องใช้ข้อมูลจำนวนมาก

เพราะว่าข้อมูลอนุกรมเวลาตัวแรกกับข้อมูลตัวที่หลักพันหรือหลักหมื่นจะไม่มีความสัมพันธ์กันหรือมีความสัมพันธ์กันน้อย พร้อมทั้งยังมีความสัมพันธ์ระหว่าง Lag ซ้ำซ้อนกัน ประกอบกับข้อมูลอนุกรมเวลาในความเป็นจริง เช่นข้อมูลยอดขายสินค้ารายเดือน รายปี ซึ่งมักจะค้นหาข้อมูลในอดีตได้ยาก ตัวอย่างเวลาในการคำนวณของแต่ละตัวแบบ งานวิจัยชิ้นนี้ทำการรันบนคอมพิวเตอร์โนตบุก CPU Intel core i5 4 Core 2.53GHz แรม 6 GB ตัวอย่างเวลาที่ใช้ในกาคำนวณของทั้ง 6 ตัวแบบ สำหรับข้อมูล Sunspot แสดงดังตารางที่ 4.14

ตัวแบบ	เวลาในการคำนวณ (วินาที)	เวลาในการคำนวณ (นาที)
ARIMA	15.55	น้อยกว่า 1 นาที
ANN	3526.29	58.77
SVM	2776.07	46.26
ARIMAANN	2810.95	46.84
ARIMASVM	1612.07	26.86
(ARIMA+ANN+SVM)	6377.91	106.29

ตารางที่ 4.14 เวลาที่ใช้ในการคำนวณของแต่ละตัวแบบ

ตารางที่ 4.14 แสดงเวลาที่ใช้ในการคำนวณ โดยยึดถือจำนวนพารามิเตอร์ที่นำเข้าเพื่อสร้างตัวแบบดังหัวข้อ 3.1, 3.2 และ 3.3 ตัวแบบ ARIMA ใช้เวลาในการคำนวณสั้นที่สุด ขณะที่ตัวแบบ ANN และ SVM ใช้เวลาในการคำนวณที่ยาวนานกว่า เนื่องจากตัวแบบทั้งสอง มีพารามิเตอร์ที่ต้องทำการค้นหาค่ามากกว่า ฟังก์ชันที่ได้จากตัวแบบทั้งสองเป็นฟังก์ชันไม่เป็นเส้นตรง จึงสามารถสร้างฟังก์ชันได้หลายรูปแบบซึ่งมีค่าปรับค่าและค้นหา จนกระทั่งได้ค่าพยากรณ์ที่มีความแม่นยำ ขณะที่ตัวแบบ ARIMA เป็นตัวแบบเส้นตรง สมการการพยากรณ์จะอยู่ในรูปผลรวมเชิงเส้น เมื่อใส่ค่าพารามิเตอร์ในกรอบที่ค้นหา ก็จะได้ค่าสัมประสิทธิ์ออกมา นั่นคือเหตุผลที่ตัวแบบ ARIMA ใช้เวลาในการคำนวณน้อยที่สุด แต่ก็มีข้อเสียที่ความแม่นยำน้อยที่สุด ดังแสดงในตารางที่ 4.1-4.10

เวลาในการคำนวณของตัวแบบ ARIMAANN, ARIMASVM น้อยกว่าตัวแบบเดี่ยว ANN และ SVM เนื่องจากตัวแบบ ARIMAANN, ARIMASVM จะต้องทำการพยากรณ์ด้วยตัวแบบ ARIMA ก่อนแล้วนำค่าเศษเหลือพยากรณ์ต่อด้วยตัวแบบ ANN และ SVM ค่าเศษเหลือนี้ขนาดสเกลของข้อมูลจะน้อยจึงไม่จำเป็นต้องทำการแปลงข้อมูลเพื่อจะพยากรณ์ด้วย ANN หรือ SVM มากนักดังการใช้ข้อมูลจริงเข้ามาพยากรณ์ ทำให้เวลาที่ใช้ในการคำนวณของตัวแบบ ARIMAANN และ ARIMASVM น้อยกว่าตัวแบบเดี่ยวทั้ง ANN และ SVM

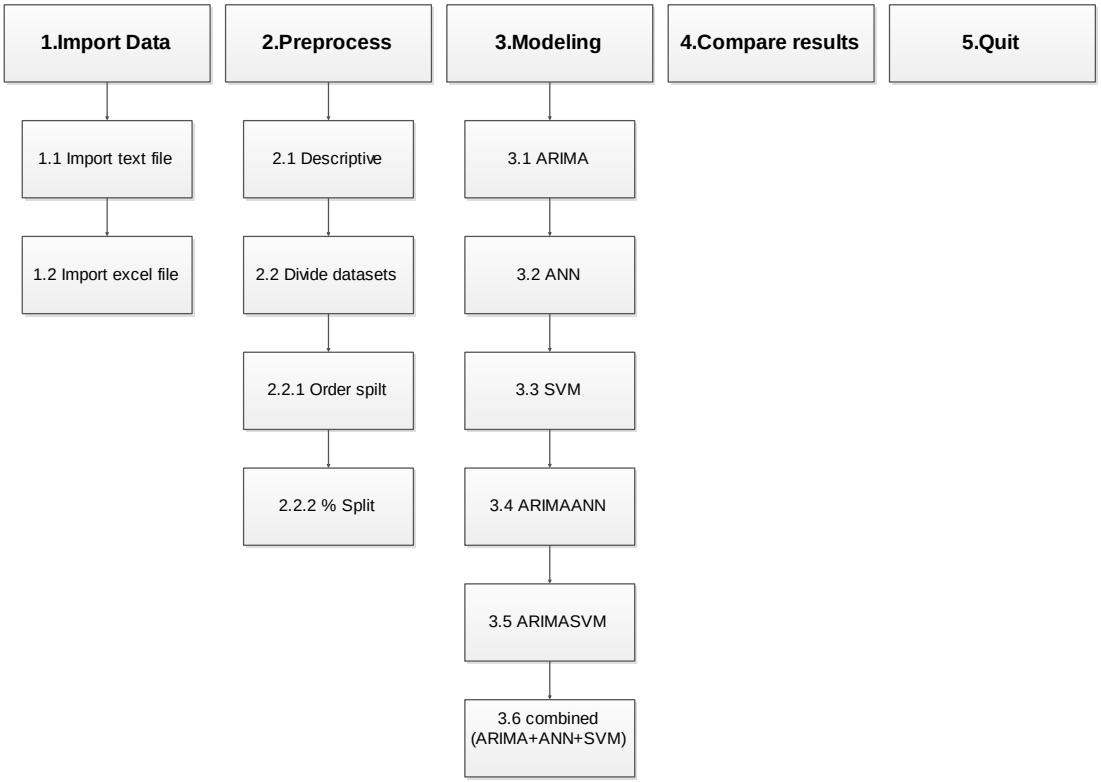
บทที่ 5

โปรแกรม GUI สำหรับพยากรณ์ตัวแบบผสม

โปรแกรม GUI ที่สร้างขึ้นเป็นโปรแกรมที่ตอบโต้กับผู้ใช้งาน ส่วนที่ตอบโต้กับผู้ใช้งาน ได้แก่ ปุ่มกด (push buttons) ตัวเลือกแบบเมนู (pop-up menu) กราฟ (axes) เป็นต้น หากโปรแกรมโต้ตอบกับผู้ใช้ได้รับการออกแบบที่ดีจะทำให้ผู้ใช้สามารถเข้าถึงการทำงานหรือใช้งานได้อย่างไม่ต้องทราบขั้นตอนการทำงานมาก่อนเลย

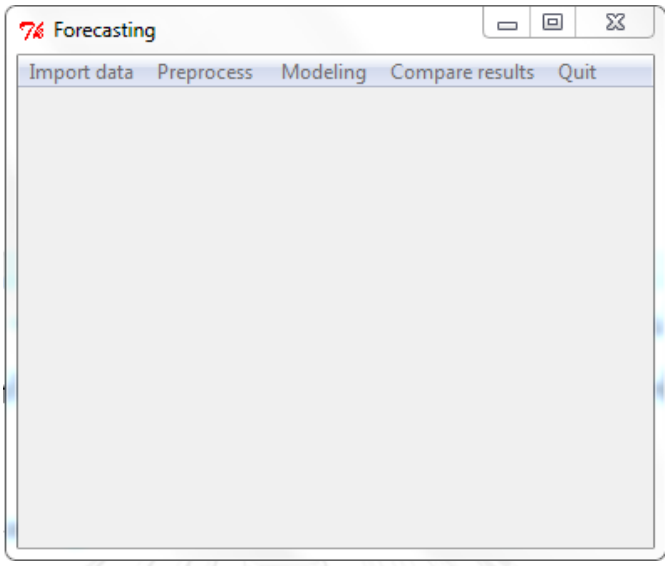
ประโยชน์ที่เห็นได้อย่างชัดเจนของ GUI คือการที่ผู้ใช้สามารถทำงานกับโปรแกรมได้โดยไม่ต้องผ่านทางการเขียนคำสั่งทีละบรรทัดดังเช่นการเขียนโปรแกรมตามปกติ กล่าวคือผู้ใช้เพียงแต่ปฏิบัติตามคำสั่งสำเร็จรูปที่ทางผู้ออกแบบได้จัดหามาให้และทำการใช้งานได้ทันที ดังนั้นจึงเป็นการง่ายกว่าสำหรับผู้ใช้ทั้งในด้านการเรียนรู้และการใช้งานโปรแกรม

โปรแกรมพยากรณ์ตัวแบบผสมที่จัดทำขึ้นมานี้ เป็นโปรแกรมในลักษณะ Graphical User Interface (GUI) สร้างจากโค้ดของโปรแกรม R พัฒนาขึ้นเพื่อใช้พยากรณ์ตัวแบบผสมและตัวแบบเดี่ยวซึ่งยังไม่มีการทำมาก่อน มีชื่อว่า Forecasting สามารถพยากรณ์ข้อมูลอนุกรมเวลาสำหรับตัวแปรเดียว (Univariate model) ประกอบด้วยตัวแบบพยากรณ์ ARIMA, ANN, SVM, ARIMAANN, ARIMASVM และตัวแบบ Combined (ARIMA+ANN+SVM)



ภาพที่ 5.1 แผนผังแต่ของเมนูใหญ่และเมนูย่อยในโปรแกรม Forecasting

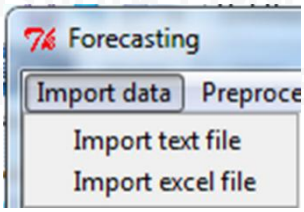
ประกอบด้วย 5 เมนูย่อย คือ Import data, Preprocess, Modeling, Compare results และ Quit ดังภาพที่ 5.1



ภาพที่ 5.2 แถบเมนูหน้าต่างทั้ง 5 เมนู โปรแกรม Forecasting

หน้าตาของโปรแกรม Forecasting เป็นดังภาพที่ 5.1 ด้านมุมบนซ้ายปรากฏชื่อ Forecasting ซึ่งเป็นชื่อโปรแกรมที่ตั้งขึ้น ด้านมุมบนขวาปรากฏช่องสี่เหลี่ยมเล็กๆ 3 ช่องคือ ช่องซ่อนโปรแกรม ช่องย่อหรือขยายโปรแกรมและช่องปิดโปรแกรม เรียงตามลำดับจากซ้ายไปขวา

1. เมนู Import data เป็นเมนูที่นำเข้าข้อมูลเข้าสู่ GUI สามารถนำเข้าข้อมูลสองกลุ่ม คือ ข้อมูลที่เป็นข้อความ Text ได้แก่ไฟล์นามสกุล .txt, .tab, .csv และข้อมูลจากไฟล์ Excel นามสกุล .xls และ .xlsx ดังภาพที่ 5.3



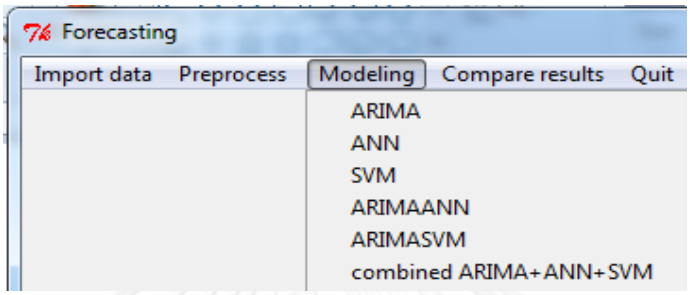
ภาพที่ 5.3 แถบเมนู Import และแถบเมนูย่อย Import text file และ Import excel file ของโปรแกรม Forecasting

2. เมนู Preprocess แบ่งเป็นสองเมนูย่อย คือ Descriptive และ Divide datasets

2.1 เมนู Descriptive เป็นเมนูที่บอกลักษณะข้อมูลขั้น เช่นบอกค่าเฉลี่ย ค่าน้อยสุด ค่าต่ำสุด ค่าข้อมูลในตำแหน่งเปอร์เซนไทล์ที่ 25 และ 75

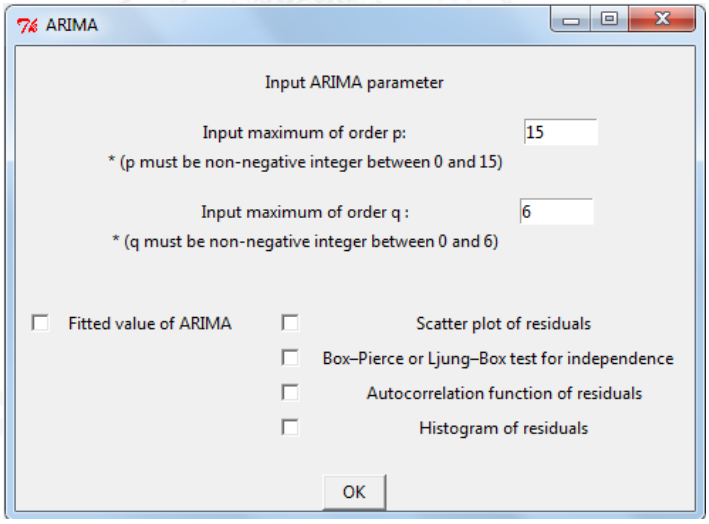
2.2 เมนู Divide datasets เป็นเมนูที่ใช้แบ่งข้อมูลออกเป็นสองกลุ่ม คือ ข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบ มีเมนูย่อยสองเมนู คือ Order split แบ่งข้อมูลตามตำแหน่งข้อมูลที่เลือก ขณะที่ % split แบ่งข้อมูลตามจำนวนเปอร์เซ็นต์ของข้อมูลที่เลือก

3. เมนู Modeling เป็นเมนูที่ใช้พยากรณ์ข้อมูลชุดเรียนรู้ มีเมนูย่อย 6 เมนู คือ ARIMA, ANN, SVM, ARIMAANN, ARIMASVM และ combined ARIMA+ANN+SVM



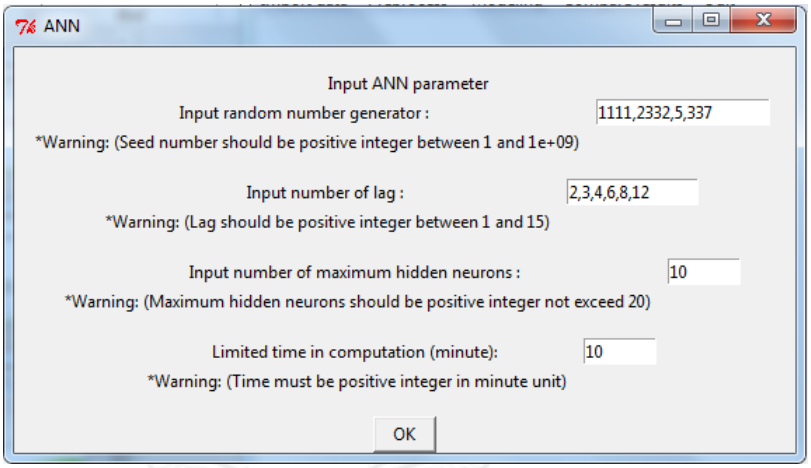
ภาพที่ 5.4 แถบเมนู Modeling และแถบเมนูย่อยของโปรแกรม Forecasting

3.1 เมนู ARIMA ประกอบด้วย ช่องใส่ค่า Parameter และมีช่องให้เลือกว่า จะให้แสดงค่ารายงานผลการวิเคราะห์ค่าเศษเหลือ (Residuals) ใดบ้าง



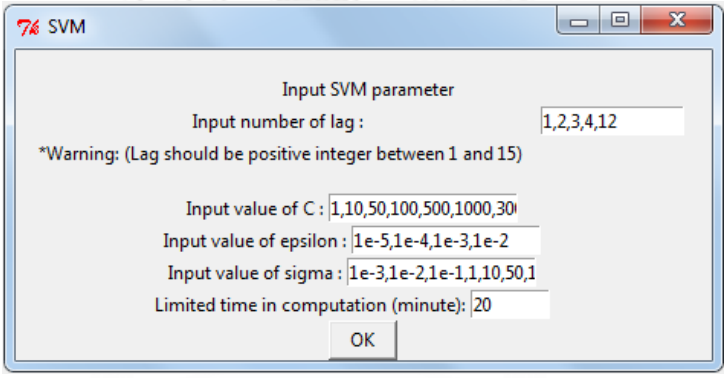
ภาพที่ 5.5 หน้าต่างแถบเมนูย่อย ARIMA

3.2 เมนูย่อย ANN มีช่องใส่ค่า Parameter ทำขึ้นให้อยู่ในรูปการค้นหาตัวแบบที่ดีที่สุดภายใต้กรอบ Parameter ที่ใส่เข้าไป



ภาพที่ 5.6 หน้าต่างแถบเมนูย่อย ANN

3.3 เมนูย่อย SVM มีช่องใส่ค่า Parameter ทำขึ้นให้อยู่ในรูปการค้นหาตัวแบบที่ดีที่สุดภายใต้กรอบ Parameter ที่ใส่เข้าไป

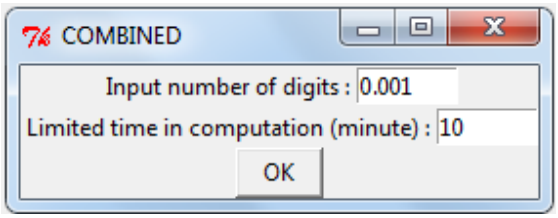


ภาพที่ 5.7 หน้าต่างแถบเมนูย่อย SVM

3.4 เมนู ARIMAANN ทำการพยากรณ์ด้วยตัวแบบ ARIMAANN จะสามารถกดเมนูนี้ได้ก็ต่อเมื่อ มีการกดเมนู ARIMA ทำการพยากรณ์ด้วยตัวแบบ ARIMA ก่อนหน้าแล้ว ช่องใส่ค่า Parameter มีหน้าตาเหมือนกับ เมนู ANN

3.5 เมนู ARIMASVM ทำการพยากรณ์ด้วยตัวแบบ ARIMASVM จะสามารถกดเมนูนี้ได้ก็ต่อเมื่อ มีการกดเมนู ARIMA ทำการพยากรณ์ด้วยตัวแบบ ARIMA ก่อนหน้าแล้ว ช่องใส่ค่า Parameter มีหน้าตาเหมือนกับ เมนู SVM

3.6 เมนูย่อย Combined ARIMA+ANN+SVM เมื่อกดเข้าไป จะมีช่องให้ใส่ค่าพารามิเตอร์ดังภาพที่ 5.8



ภาพที่ 5.8 หน้าต่างแถบเมนูย่อย COMBINED

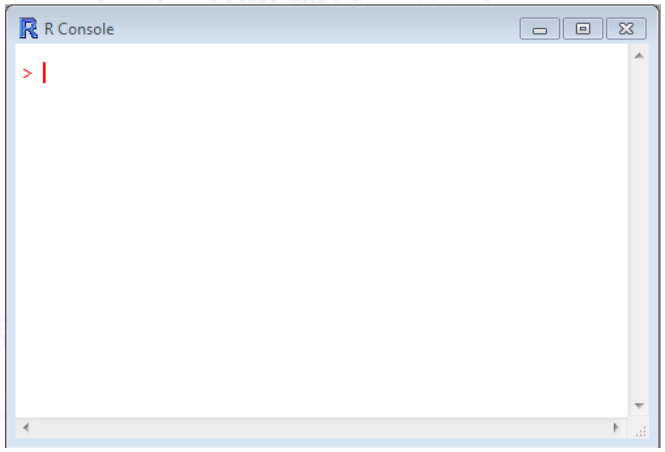
4. เมนู Compare results เป็นเมนูที่เมื่อกดไปแล้วจะทำการแสดงผลลัพธ์ของตัวแบบพยากรณ์ทั้ง 6 ตัวแบบ

5. เมนูสุดท้าย Quit เป็นเมนูที่กดเพื่อออกจาก Forecasting GUI

ตัวอย่างการพยากรณ์ข้อมูลอนุกรมเวลาราคาทองคำ (Gold)

1. เปิดโปรแกรม R ขึ้นมา สำหรับผู้ที่ยังไม่มีโปรแกรม R สามารถดาวน์โหลดฟรีและอ่านรายละเอียดขั้นตอนการติดตั้งจาก http://cran.r-project.org/bin/windows/base/

2. เมื่อติดตั้งเรียบร้อยแล้วก็เปิดหน้าต่าง R ขึ้นมา เป็นดังภาพที่ 5.9



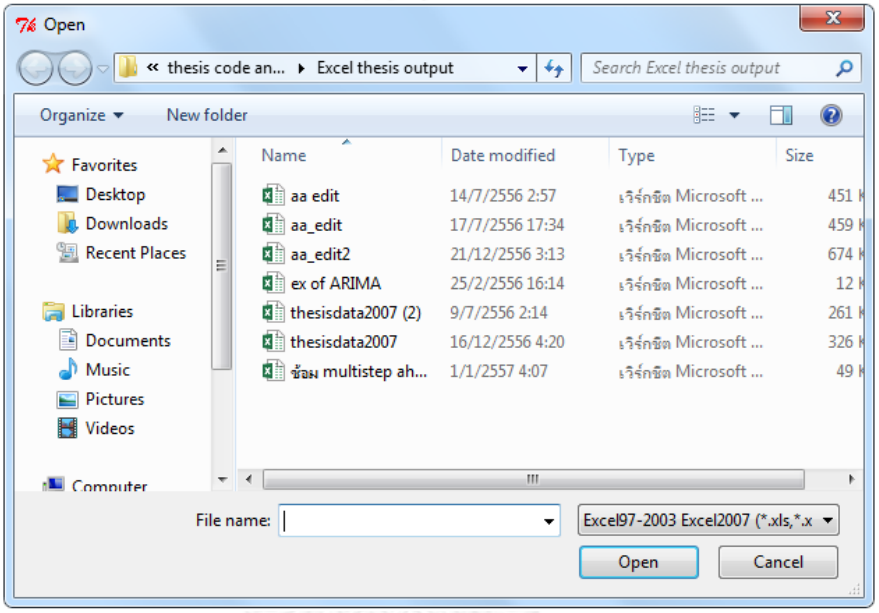
ภาพที่ 5.9 หน้าต่างโปรแกรม R console เมื่อเริ่มเปิดขึ้นใหม่

หน้าต่างนี้เป็นหน้าต่างหลักของโปรแกรม R ใช้สำหรับการคิดคำนวณทางคณิตศาสตร์ตั้งแต่บวก ลบ คูณ หาร ไปจนถึงการคำนวณที่ซับซ้อน

3.รันโค้ดเพื่อสร้าง Forecasting GUI (โค้ดที่ใช้สร้างโปรแกรม Forecasting GUI อยู่ใน**ภาคผนวก** ผู้ใช้สามารถ Copy แล้ว Paste ในหน้าต่าง R console ได้เลย) เมื่อรันเสร็จจะได้หน้าต่าง ดังภาพที่ 5.2

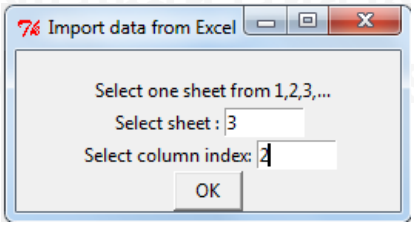
หน้าต่างโปรแกรม Forecasting เป็นหน้าต่างตัวสิ่งที่ผู้ใช้ป้อนข้อมูลเข้าไปและเลือกตัวแบบพยากรณ์ แต่เวลาแสดงผลจะแสดงผลในหน้าต่าง R console ดังภาพที่ 5.8

4. ทำการ Import ข้อมูล สามารถทำได้กับไฟล์จากโปรแกรม Excel version 97-2003 และ 2007 ขึ้นไปโดยคลิก Import data->Import excel file และไฟล์ .txt, .tab และ .csv โดยคลิก Import data->Import text file จากนั้นก็จะปรากฏหน้าต่างดังภาพที่ 5.10 ให้เลือกไฟล์ที่ต้องการในตำแหน่งที่บันทึกไว้ในคอมพิวเตอร์



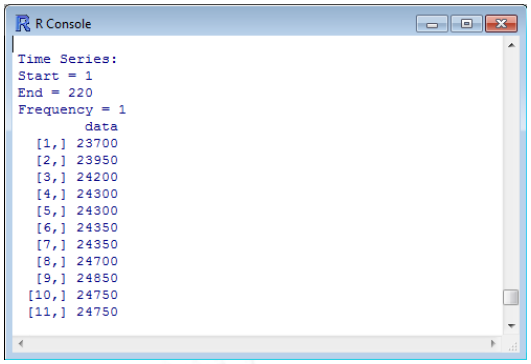
ภาพที่ 5.10 หน้าต่างเลือกไฟล์ในตำแหน่งที่บันทึกไว้

5. เลือกแผ่นงาน (Sheet) และคอลัมน์ (Column) ของข้อมูล ที่ต้องการนำมาพยากรณ์ ดังภาพที่ 5.11



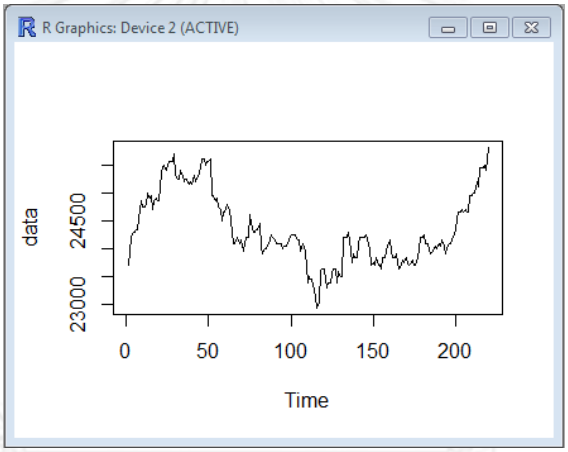
ภาพที่ 5.11 หน้าต่างเลือกไฟล์ในตำแหน่งที่บันทึกไว้

กดปุ่ม OK ข้อมูลจะนำเข้ามายังหน้าต่าง R console พร้อมแสดงกราฟ ของข้อมูลนั้น ดังภาพที่ 5.11



ภาพที่ 5.12 หน้าต่างรายงานผลข้อมูลที่ Import

ข้อมูลที่ Import เข้ามานั้น มีจำนวน 220 ตัว (ดูที่ End=220) เป็นข้อมูลประเภทอนุกรมเวลา (ดูที่ Time Series) และกราฟของข้อมูลจะแสดงดังภาพที่ 5.13



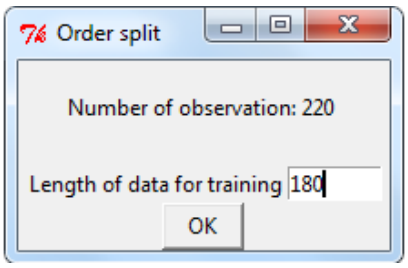
ภาพที่ 5.13 กราฟของข้อมูลภายหลังการ Import

6. ทำการแบ่งข้อมูลเป็นสองชุด คือ ชุดข้อมูลสำหรับสร้างตัวแบบ และชุดข้อมูลสำหรับทดสอบ สามารถแบ่งได้เป็นสองแบบ คือ

แบ่งตามลำดับข้อมูล จะนับข้อมูลตั้งแต่ข้อมูลที่หนึ่งถึงตำแหน่งข้อมูลที่เราแบ่งเป็นข้อมูลชุดเรียนรู้ ส่วนข้อมูลที่เหลือเป็นข้อมูลชุดทดสอบ

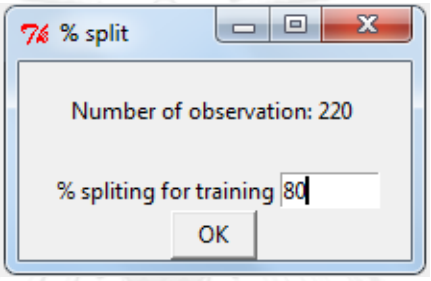
แบ่งตามเปอร์เซ็นต์ที่ใช้ จะนับข้อมูลตั้งแต่ข้อมูลที่หนึ่งถึงตำแหน่งข้อมูลที่เป็นเปอร์เซนต์ที่เราแบ่งเป็นข้อมูลชุดเรียนรู้ ส่วนข้อมูลที่เหลือเป็นข้อมูลชุดทดสอบ

6.1 แบ่งตามลำดับข้อมูลแบบกำหนดจุดแบ่ง คลิก Preprocess->Divide datasets->Order split



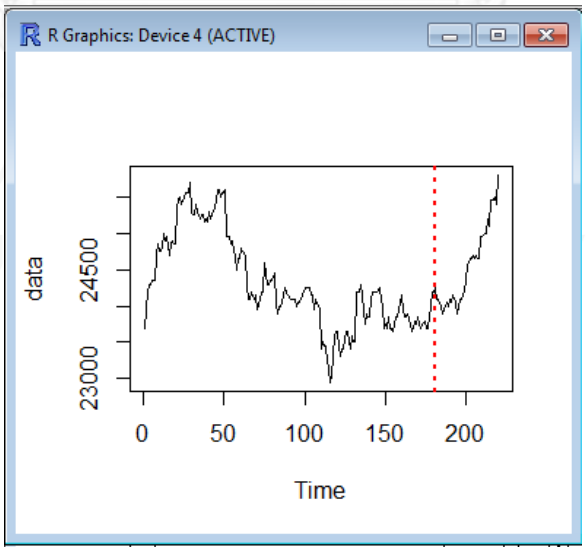
ภาพที่ 5.14 หน้าต่างการแบ่งข้อมูลแบบกำหนดจุดแบ่งกลุ่มข้อมูล

6.2 แบ่งตามลำดับข้อมูลแบบกำหนดเปอร์เซ็นต์ แบ่งกลุ่ม คลิก Preprocess->Divide datasets->% split



ภาพที่ 5.15 หน้าต่างการแบ่งข้อมูลแบบกำหนดเปอร์เซ็นต์การแบ่ง

ในตัวอย่างนี้เลือกแบ่งแบบกำหนดจุด จุดแบ่งอยู่ที่ข้อมูลตัวที่ 180 และที่เหลือ 181-220 เป็นข้อมูลชุดทดสอบ เมื่อแบ่งเสร็จข้อมูลทั้งสองชุดจะแสดงไว้ใน R console และปรากฏกราฟข้อมูลที่แสดงช่วงของข้อมูลที่ใช้สำหรับสร้างตัวแบบและทดสอบ ดังภาพที่ 5.15



ภาพที่ 5.16 กราฟข้อมูลพร้อมเส้นแบ่งช่วงแยกกลุ่มข้อมูล

ด้านซ้ายมือเส้นสีแดงคือ ข้อมูลที่นำมาสร้างตัวแบบเป็นข้อมูลชุดเรียนรู้ ส่วนด้านขวามือคือข้อมูลที่นำมาทดสอบ

จำนวนข้อมูลที่แบ่งเป็นข้อมูลชุดเรียนรู้และข้อมูลชุดทดสอบนั้นไม่มีกฎเกณฑ์ตายตัว แต่ต้องแบ่งให้จำนวนข้อมูลชุดเรียนรู้นั้นมากกว่าครึ่งหนึ่งของข้อมูลทั้งหมด (มากกว่า 50%)

7. เข้าสู่การสร้างตัวแบบ ด้วยเมนู Modeling

หลังจากแบ่งข้อมูลเรียบร้อยแล้วก็จะสามารถสร้างตัวแบบ (ต้องทำตามลำดับ มิฉะนั้นจะไม่สามารถทำต่อได้)

7.1 การสร้างตัวแบบ ARIMA โดยคลิก Modeling->ARIMA จะได้หน้าต่างดังภาพที่ 5.5 ค้นหาอันดับที่ดีที่สุดของตัวแบบภายใต้ Information criteria AICc มีพารามิเตอร์ที่ต้องใส่ 2 ตัว คือ จำนวน p สูงสุด (Input Maximum of Order p) และจำนวน q สูงสุด (Input Maximum of Order q) โดยที่อันดับเพิ่มขึ้นทีละ 1 อันดับ p เริ่มหาจาก 0-p ที่ใส่ค่า และอันดับ q เริ่มจาก 0-q ที่ใส่ค่า โดยกำหนดค่า default Maximum of order p=15 และ Maximum of order q=6 ดังนั้นอันดับ p จะเริ่มค้นหาตั้งแต่ 0-15 และอันดับ q จะเริ่มค้นหาตั้งแต่ 0-6 มีการทดสอบคุณสมบัติ Stationary ภายในด้วย Augmented Dickey and Fuller (ADF) test จึงได้อันดับ q มาภายหลัง สำหรับช่องอื่นๆที่ให้เลือกติ๊ก คือเลือกว่าจะรายงานผลการวิเคราะห์ค่าเศษเหลือตัวใดบ้าง เมื่อเลือกค่าครบถ้วนแล้วก็กดปุ่ม OK ค่ารายงานผลจะแสดงใน R console ดังภาพที่ 5.17

```
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.8422  -0.8597
s.e.  0.5456   0.5147

sigma^2 estimated as 25897:  log likelihood=-1163.48
AIC=2332.96   AICc=2333.1   BIC=2342.52
```

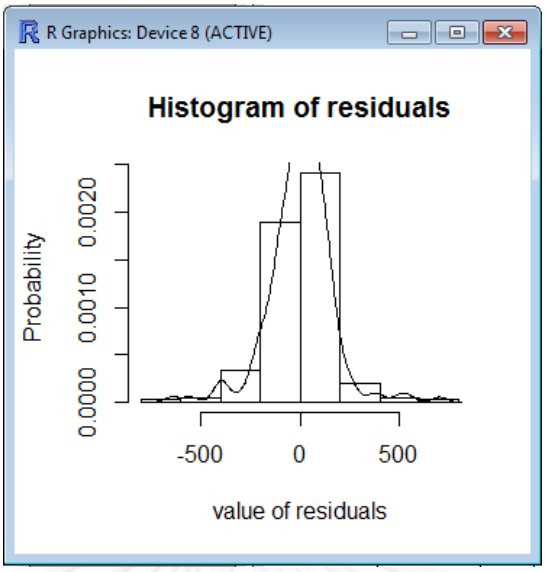
ภาพที่ 5.17 อันดับที่ดีที่สุดจากการค้นหาของตัวแบบ ARIMA

แสดงค่า อันดับตัวแบบ คือ ARIMA(1,1,1) p=1,d=1,q=1 ซึ่งเป็นอันดับที่ดีที่สุดจากการค้นหาและแสดงค่าสัมประสิทธิ์ของตัวแบบ พร้อมทั้งแสดงการทดสอบความเป็นอิสระของ Residuals ด้วย Box-Ljung test ดังภาพที่ 5.18

```
        Box-Ljung test

data:  fit$res
X-squared = 11.2115, df = 9, p-value = 0.2615
```

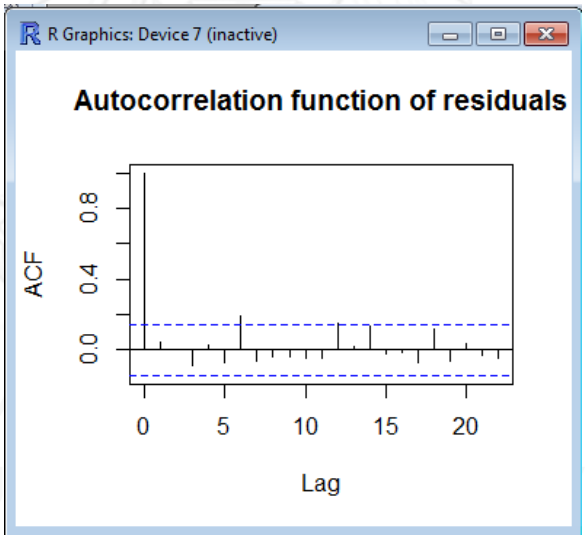
ภาพที่ 5.18 การทดสอบความเป็นอิสระของ Residuals

นอกจากนั้นยังแสดงกราฟของ Residuals ประกอบด้วย Histogram, ACF และกราฟเปรียบเทียบค่าพยากรณ์และค่าจริง ดังภาพที่ 5.19-5.21 ตามลำดับ

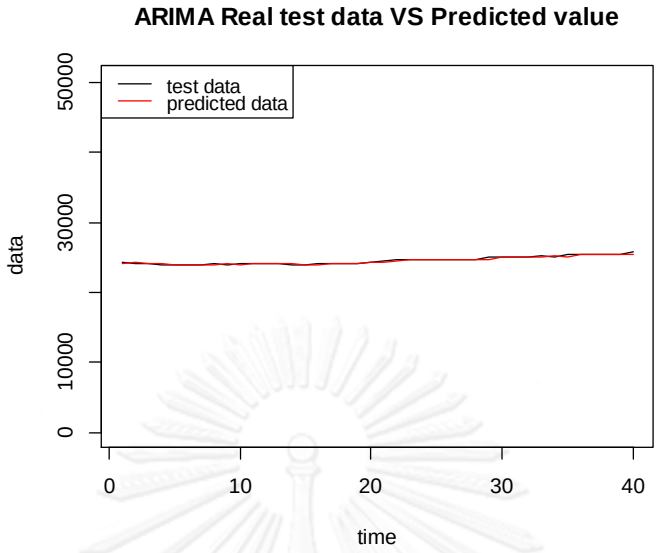


ภาพที่ 5.19 Histogram ของ Residuals

กราฟ Histogram ของ Residuals ที่ได้จากตัวแบบควรจะเป็นรูประฆังคว่ำ รูปแบบเดียวกับ เส้นโค้ง Normal



ภาพที่ 5.20 Autocorrelation ของ Residuals



ภาพที่ 5.21 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ARIMA

7.2 การสร้างตัวแบบ ANN โดยการคลิก Modeling->ANN จะปรากฏหน้าต่างขึ้นมาดังภาพที่ 5.6

ค่าพารามิเตอร์ที่ต้องใส่ได้แก่ ตัวเลขสุ่ม (Input random number generator), จำนวน Lag ที่ใช้ (Input number of lag), จำนวน Hidden neurons สูงสุด (Input number of maximum hidden neurons) และเวลาสูงสุดที่ยอมให้มีการคำนวณหน่วยนาที (Limited time in computation: minute) ค่าพารามิเตอร์ทุกค่าต้องใส่เป็นตัวเลขโดยที่สามารถใส่ทีละหลายๆค่าได้ โดยใช้เครื่องหมาย comma (,) คั่น สำหรับจำนวน Hidden neurons นั้นค่าที่ใส่คือค่าที่มากที่สุดจะทำการค้นหาตั้งจำนวน Hidden neuron=1 จนถึง จำนวน Hidden neuron=ค่าที่ใส่เข้าไป จะทำการค้นหาตัวแบบที่ดีที่สุดจากค่าพารามิเตอร์ที่ใส่ไปทั้งหมด จากนั้นกดปุ่ม OK ก็จะแสดงผลใน R console ดังนี้

- พารามิเตอร์ที่ดีที่สุดจากการค้นหา

```
a 3-7-1 network with 36 weights
options were - linear output units
 b->h1 i1->h1 i2->h1 i3->h1
  0.45  -8.63   6.08   2.54
 b->h2 i1->h2 i2->h2 i3->h2
 -6.05  -8.67  -6.23  -4.21
 b->h3 i1->h3 i2->h3 i3->h3
  1.47  -4.57   0.35   1.95
 b->h4 i1->h4 i2->h4 i3->h4
  1.93 -20.09  -1.77 -14.40
 b->h5 i1->h5 i2->h5 i3->h5
  0.92 -17.98  -2.40  -8.88
 b->h6 i1->h6 i2->h6 i3->h6
  1.08  -4.07  -0.03   2.58
 b->h7 i1->h7 i2->h7 i3->h7
 -1.25  13.29 -15.99   3.91
  b->o  h1->o  h2->o  h3->o  h4->o  h5->o  h6->o  h7->o
  1.52  -1.74   4.82  -8.50   6.05  -7.95   9.21  -0.95
```

โครงข่าย (Architecture) 3-7-1 คือ Input มี 3 ตัว คือ Lag=3, Hidden neurons=7, Output neurons=1 คือโครงข่าย ANN ที่ดีที่สุด พร้อมทั้งแสดงผลค่า Weights ทั้ง 36 ตัว

```
 lag    Seed hidden
   3     337      7
```

ค่า Lag ที่ดีที่สุดในการค้นหาเท่ากับ 3

ค่า Hidden neurons ที่ดีที่สุดคือ 7

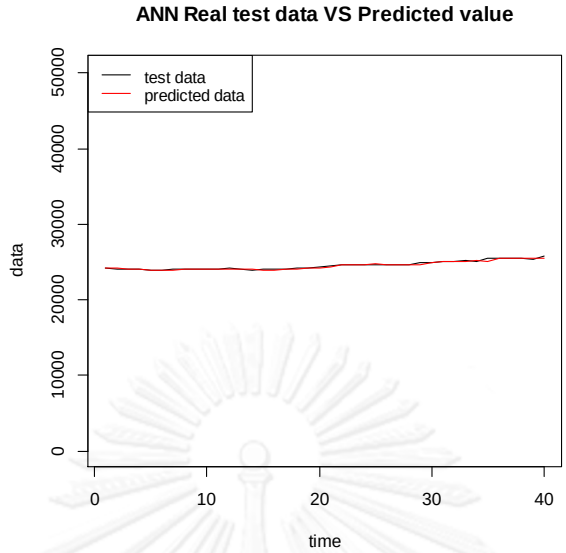
ค่า Seed ซึ่งก็คือ random number generator ที่ใส่ลงไปที่ดีที่สุดในการค้นหาคือ 337

- ข้อมูลจริง ค่าพยากรณ์ และเศษเหลือจากการพยากรณ์ (Residuals)
- ตัววัดประสิทธิภาพของตัวแบบ ค่า RMSE, MSE, MAE, MAPE และค่าพยากรณ์ล่วงหน้า

```
         RMSE          MSE          MAE          MAPE
1.191695e+02 1.420137e+04 8.648658e+01 3.505228e-01
predict.future
      25560.86
```

- กราฟแสดงผลเปรียบค่าจริงและค่าพยากรณ์สำหรับข้อมูลชุดทดสอบ

CHULALONGKORN UNIVERSITY



ภาพที่ 5.22 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ANN

7.3 การสร้างตัวแบบ SVM โดยการคลิก Modeling->SVM จะปรากฏหน้าต่างขึ้นมาดังภาพที่ 5.7

ค่าพารามิเตอร์ที่ต้องใส่ได้แก่ จำนวน Lag ที่ใช้ (Input number of lag), ค่าพารามิเตอร์ C ในตัวแบบ SVM (Input value of C), ค่าพารามิเตอร์ Epsilon ในตัวแบบ SVM, ค่าพารามิเตอร์ Sigma สำหรับ Kernel function (Input value of sigma) และเวลาสูงสุดที่ยอมให้มีการคำนวณหน่วยนาที (Limited time in computation: minute) ค่าพารามิเตอร์ทุกค่าต้องใส่เป็นตัวเลขโดยที่สามารถใส่ทีละหลายๆค่าได้โดยใช้เครื่องหมาย comma (,) คั่น จะทำการค้นหาตัวแบบที่ดีที่สุดจากค่าพารามิเตอร์ที่ใส่ไปทั้งหมด จากนั้นกดปุ่ม OK ก็จะแสดงผลใน R console ดังนี้

- ค่าพารามิเตอร์ที่ดีที่สุดจากการค้นหา

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.001  cost C = 10

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  1

Number of Support Vectors : 176

Objective Function Value : -241.2491
Training error : 0.049508

   Best.mse         Lag      Sigmas    Epsilons           C
14103.93523     3.00000     1.00000     0.00100    10.00000
```

ค่า Lag ที่ดีที่สุดในการค้นหาเท่ากับ 3

ค่า Sigma ที่ดีที่สุดคือ 1

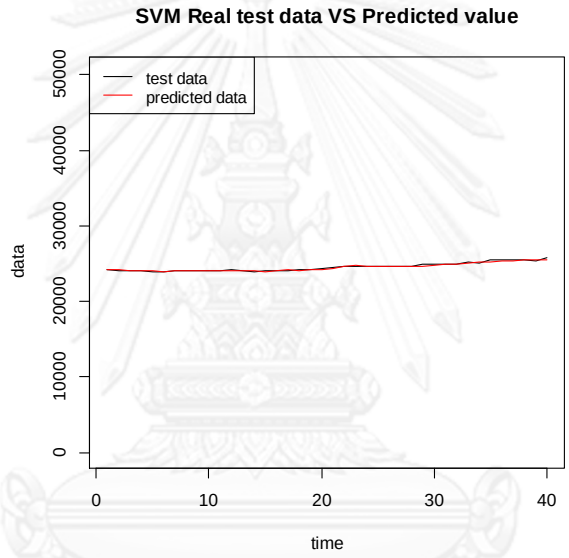
ค่า Epsilon ที่ดีที่สุดคือ 0.001

ค่า C ที่ดีที่สุดคือ 10

- ข้อมูลจริง ค่าพยากรณ์ และเศษเหลือจากการพยากรณ์ (Residuals)
- ตัววัดประสิทธิภาพของตัวแบบ ค่า RMSE, MSE, MAE, MAPE และค่าพยากรณ์ล่วงหน้า

```
            RMSE              MSE              MAE             MAPE
1.187599900e+02 1.410393523e+04 9.123297564e+01 3.694786037e-01
predict.future
    25150.89545
```

- กราฟแสดงผลเปรียบค่าจริงและค่าพยากรณ์สำหรับข้อมูลชุดทดสอบ



ภาพที่ 5.23 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ SVM

7.4 การสร้างตัวแบบ ARIMAANN โดยการคลิก Modeling->ARIMAANN จะปรากฏหน้าต่างขึ้นมาดังภาพที่ 5.8 จะสร้างตัวแบบนี้ได้ก็ต่อเมื่อ ผ่านการสร้างตัวแบบ ARIMA เป็นที่เรียบร้อยแล้ว ช่องใส่ค่าพารามิเตอร์จะเหมือนกับตัวแบบ ANN เมื่อคลิกปุ่ม OK ก็จะได้ค่าแสดงผลใน R console ดังนี้

- ค่าพารามิเตอร์ที่ดีที่สุดจากการค้นหา

```
a 1-10-1 network with 31 weights
options were - linear output units
 b->h1 i1->h1
  2.76   3.65
 b->h2 i1->h2
  1.52  -0.01
 b->h3 i1->h3
  1.56   0.38
 b->h4 i1->h4
  0.75  -0.22
 b->h5 i1->h5
  1.02   1.02
 b->h6 i1->h6
  2.52  -0.13
 b->h7 i1->h7
  1.62   1.04
 b->h8 i1->h8
  2.46   0.63
 b->h9 i1->h9
  1.49  -1.04
 b->h10 i1->h10
   1.04    0.69
  b->o  h1->o  h2->o  h3->o  h4->o  h5->o  h6->o  h7->o  h8->o  h9->o h10->o
 -1.10  -1.66  -0.88   0.13   0.58   0.52   1.58   0.81   1.01   0.08  -0.21
```

โครงข่าย 1-10-1 คือ Input มี 1 ตัว คือ Lag=1, Hidden neurons=10, Output neurons=1 พร้อมทั้งแสดงผลค่า Weights ทั้ง 31 ตัว

```
 lag    Seed hidden
   1      66      10
```

ค่า Lag ที่ดีที่สุดในการค้นหาเท่ากับ 1

ค่า Hidden neurons ที่ดีที่สุดคือ 10

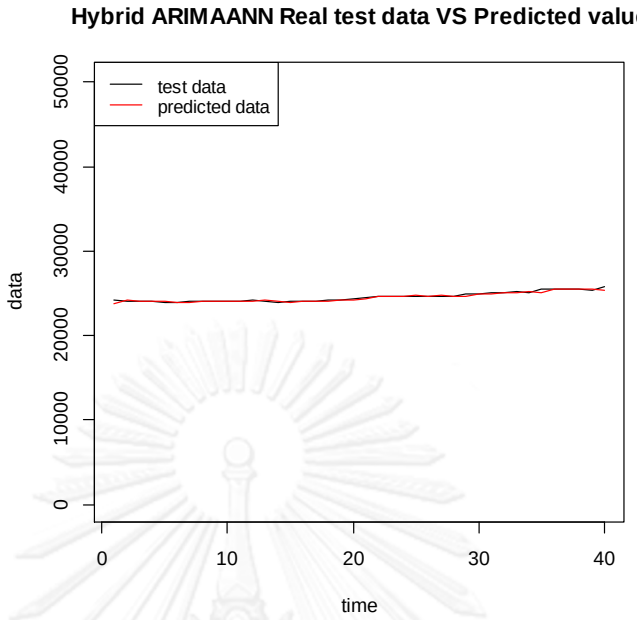
ค่า Seed ซึ่งก็คือ random number generator ที่ใส่ลงไปที่ดีที่สุดในการค้นหาคือ 66

- ข้อมูลจริง ค่าพยากรณ์ และเศษเหลือจากการพยากรณ์ (Residuals)
- ตัววัดประสิทธิภาพของตัวแบบ ค่า RMSE, MSE, MAE, MAPE และค่าพยากรณ์ล่วงหน้า

```
           RMSE             MSE             MAE            MAPE
1.287200206e+02 1.656884370e+04 9.278943672e+01 4.270690338e-01
predict.future
   25828.99842
```

- กราฟแสดงผลเปรียบค่าจริงและค่าพยากรณ์สำหรับข้อมูลชุดทดสอบ

CHULALONGKORN UNIVERSITY



ภาพที่ 5.24 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ ARIMAANN

7.5 การสร้างตัวแบบ ARIMASVM โดยการคลิก Modeling->ARIMASVM จะปรากฏหน้าต่างขึ้นมาดังภาพที่ 5.9 จะสร้างตัวแบบนี้ได้ก็ต่อเมื่อ ผ่านการสร้างตัวแบบ ARIMA เป็นที่เรียบร้อยแล้ว ช่องใส่ค่าพารามิเตอร์จะเหมือนกับตัวแบบ SVM เมื่อคลิกปุ่ม OK ก็จะได้ค่าแสดงผลใน R console ดังนี้

- ค่าพารามิเตอร์ที่ดีที่สุดจากการค้นหา

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.01  cost C = 3000

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  100

Number of Support Vectors : 176

Objective Function Value : -1500.7269
Training error : 0.000524
```

Best.mse	Lag	Sigmas	Epsilons	C
12756.51482	3.00000	100.00000	0.01000	3000.00000

ค่า Lag ที่ดีที่สุดในการค้นหาเท่ากับ 3

ค่า Sigma ที่ดีที่สุดคือ 100

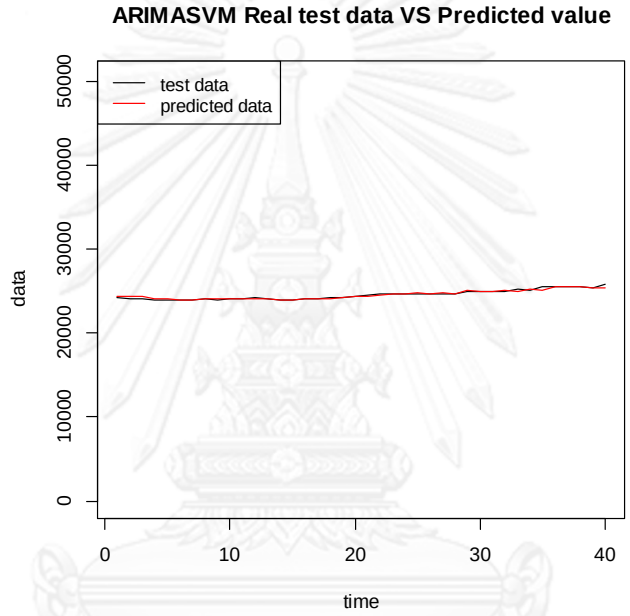
ค่า Epsilon ที่ดีที่สุดคือ 0.01

ค่า C ที่ดีที่สุดคือ 3000

- ข้อมูลจริง ค่าพยากรณ์ และเศษเหลือจากการพยากรณ์ (Residuals)
- ตัววัดประสิทธิภาพของตัวแบบ ค่า RMSE, MSE, MAE, MAPE และค่าพยากรณ์ล่วงหน้า

```
             RMSE              MSE              MAE             MAPE
1.129447423e+02 1.275651482e+04 8.123058614e+01 3.504196151e-01
predict.future
   25829.68133
```

- กราฟแสดงผลเปรียบค่าจริงและค่าพยากรณ์สำหรับข้อมูลชุดทดสอบ



ภาพที่ 5.25 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบของตัวแบบ ARIMASVM

7.6 การสร้างตัวแบบ Combined โดยการคลิก Modeling-> COMBINED จะสร้างตัวแบบนี้ได้ก็ต่อเมื่อ ผ่านการสร้างตัวแบบ ARIMA, ANN และ SVM เป็นที่เรียบร้อยแล้ว ช่องใส่ค่าพารามิเตอร์ คือ Number of digit คือ ตำแหน่งทศนิยมที่ใช้ในการค้นหา การกำหนดทศนิยมที่ละเอียดมาก เช่น ตำแหน่งทศนิยมตำแหน่งที่ 7,8,... จะทำให้การค้นหาใช้เวลานานมาก และมักไม่ได้ให้ค่าพยากรณ์ที่ดีขึ้น ใน Forecasting GUI กำหนดค่าจำนวนตำแหน่งทศนิยมเป็น 3 (default=3) แต่ก็สามารถปรับเปลี่ยนเป็นค่าอื่นได้ตั้งแต่ทศนิยมตำแหน่งที่ 1 ถึงตำแหน่งที่ 6 และมีช่องให้ใส่เวลาสูงสุดที่ยอมให้มีการคำนวณหน่วยนาที (Limited time in computation: minute) ในการคำนวณเอาไว้ในกรณีที่มีเวลาจำกัด หน่วยเป็นนาที ค่าแสดงผลใน R console ได้แก่

- ค่าพารามิเตอร์ที่ดีที่สุดจากการค้นหา

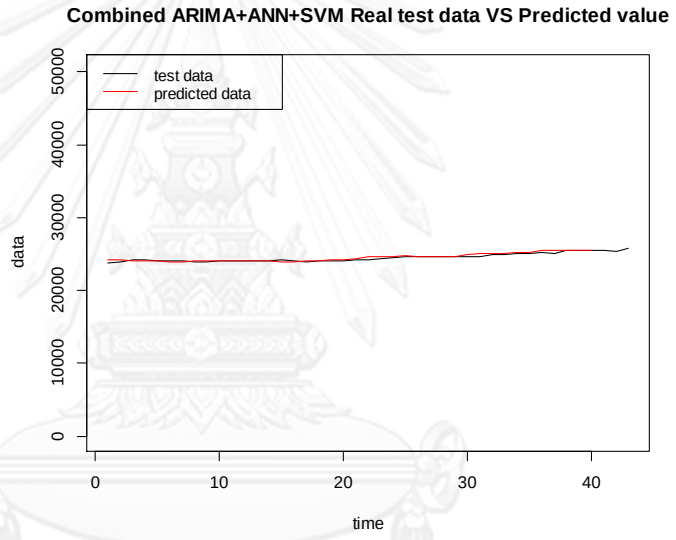
```
   Best.mae Weight.arima   Weight.ann   Weight.svm
85.06283779   0.00000000   0.85800000   0.14200000
```

ภายใต้ตำแหน่งทศนิยมที่กำหนด ในที่นี้ใช้ค่า default ทศนิยมเท่ากับ 3 จะได้ค่า Weight ของทั้ง ARIMA, ANN และ SVM ซึ่งเป็นชุดของค่า Weight ที่ดีที่สุด (หลักการการค้นหาค่า Weight สามารถดูได้ในหัวข้อ 3.5)

- ข้อมูลจริง ค่าพยากรณ์ และเศษเหลือจากการพยากรณ์ (Residuals)
- ตัววัดประสิทธิภาพของตัวแบบ ค่า RMSE, MSE, MAE, MAPE และค่าพยากรณ์ล่วงหน้า

```
          RMSE              MSE              MAE              MAPE
1.178702241e+02 1.389338972e+04 8.506283779e+01 3.448227239e-01
predict.future
   25502.64225
```

- กราฟแสดงผลเปรียบค่าจริงและค่าพยากรณ์สำหรับข้อมูลชุดทดสอบ



ภาพที่ 5.26 เปรียบเทียบค่าจริงกับค่าพยากรณ์ ในข้อมูลชุดทดสอบ ของตัวแบบ Combined (ARIMA+ANN+SVM)

8. เปรียบเทียบผลลัพธ์การพยากรณ์บนข้อมูลชุดทดสอบโดยการคลิกที่เมนู Compare results จะเปรียบเทียบผลลัพธ์ ด้วย RMSE,MSE,MAE และ MAPE พร้อมทั้งแสดงค่าพยากรณ์ล่วงหน้า เมนูนี้จะทำงานได้ก็ต่อเมื่อมีการรันตัวแบบทั้ง 6 ตัวแบบครบถ้วนแล้ว นำค่า RMSE, MSE, MAE และ MAPE พร้อมทั้งค่าพยากรณ์ล่วงหน้า 1 ค่าของแต่ละตัวแบบมาจัดเรียงเป็นตารางเดียวกันทั้งหมดเพื่อให้ผู้ใช้ได้เลือกตัวแบบ ที่ให้ค่า Error ต่ำสุดและนำค่าพยากรณ์นั้นไปใช้

	RMSE	MSE	MAE	MAPE	predict.future
ARIMA	128.2946047	16459.50560	90.97412826	0.3683943514	25803.66361
ANN	119.1694884	14201.36698	86.48657852	0.3505228281	25560.85676
SVM	118.7599900	14103.93523	91.23297564	0.3694786037	25150.89545
ARIMAANN	128.7200206	16568.84370	92.78943672	0.4270690338	25828.99842
ARIMASVM	112.9447423	12756.51482	81.23058614	0.3504196151	25829.68133
COMBINED	117.8702241	13893.38972	85.06283779	0.3448227239	25502.64225

จากตารางข้างต้นผู้ใช้จะต้องพิจารณาว่าควรเลือกใช้ ตัวแบบไหน ตัวแบบ ARIMASVM และตัวแบบ Combined จัดเป็นตัวแบบที่มีค่า Error ต่ำ กรณีที่ผู้ใช้ไม่พอใจผลลัพธ์ ก็สามารถกลับไปรันใหม่ใส่ค่า Parameter ใหม่ทีละหลายๆ ตัว ก็อาจช่วยให้ค่าพยากรณ์ดียิ่งขึ้น

หมายเหตุ: ผลลัพธ์ที่แสดงให้เห็นนี้เป็นเพียงผลลัพธ์ที่แสดงตัวอย่างการใช้งานโปรแกรม Forecasting ที่สร้างขึ้นเท่านั้น ค่าจากการวิเคราะห์จะไม่ตรงกับค่าผลการทดลองในบทที่ 4 เนื่องจากตัวแบบ ANN นั้นใช้ Matlab R2012a Neural network toolbox คำนวณ ขณะที่โปรแกรม Forecasting ที่สร้างขึ้นนี้ใช้ Code R เขียนขึ้นทั้งหมด เพื่อให้สามารถสร้างโปรแกรม GUI ที่ให้ผู้สนใจทดลองพยากรณ์ด้วยตัวแบบผสมและตัวแบบ Combined ได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6

สรุปผลการทดลองและข้อเสนอแนะ

จากผลการทดลองจากตารางที่ 4.1-4.10 สามารถสรุปผลเปรียบเทียบประสิทธิภาพของตัวแบบได้ ดังนี้

6.1 เปรียบเทียบความแม่นยำของตัวแบบ Traditional hybrid กับตัวแบบเดี่ยว

- ตัวแบบ ARIMAANN ดีกว่า ANN และ ARIMA ได้แก่ข้อมูล Electricuse, Usexchange ,Gold และ Meanrain
- ตัวแบบ ARIMAANN มีประสิทธิภาพใกล้เคียงกับANN และ ARIMA ได้แก่ข้อมูล Motorcycle
- ตัวแบบ ARIMAANN แย่กว่า ANN และ ARIMA ได้แก่ข้อมูล Sunspot , Canadian lynx , GBP/USD ,Windspeed และ Maxtemp
- ตัวแบบ ARIMASVM ดีกว่า SVM และ ARIMA ได้แก่ข้อมูล GBP/USD, Electricuse และ Gold
- ตัวแบบ ARIMASVM มีประสิทธิภาพใกล้เคียงกับSVMและ ARIMA ได้แก่ข้อมูล Sunspot, Motorcycle และ Usexchange
- ตัวแบบ ARIMASVM แย่กว่า SVM และ ARIMA ได้แก่ข้อมูล Canadian lynx, Windspeed, Meanrain และ Maxtemp

6.2 เปรียบเทียบความแม่นยำของตัวแบบ Hybrid combined (ARIMA+ANN+SVM) กับตัวแบบ Traditional hybrid

- ตัวแบบ (ARIMA+ANN+SVM) ดีกว่า ARIMAANN และ ARIMASVM ได้แก่ข้อมูล Sunspot, Canadian lynx, Electricuse, USexchange, Gold, Windspeed, Maxtemp และ GBP/USD
- ตัวแบบ (ARIMA+ANN+SVM) มีประสิทธิภาพใกล้เคียงกับ ARIMAANN และ ARIMASVM ได้แก่ข้อมูล Meanrain
- ตัวแบบ (ARIMA+ANN+SVM) แย่กว่า ARIMAANN และ ARIMASVM ได้แก่ข้อมูล Gold แย่กว่า ARIMAANN

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

6.3 เปรียบเทียบความแม่นยำของตัวแบบ Hybrid combined (ARIMA+ANN+SVM) กับตัวแบบเดี่ยว ARIMA, ANN และ SVM

- ตัวแบบ (ARIMA+ANN+SVM) ดีกว่าตัวแบบ ARIMA, ANN และ SVM เกือบทุกข้อมูล ยกเว้น GBP/USD และ USexchange ทั้งสองมีประสิทธิภาพใกล้เคียงกับ ANN
- สำหรับข้อมูลมาตรฐานที่ใช้ในงานวิจัยชิ้นนี้ ได้แก่ ข้อมูล Sunspot, Canadian lynx และ GBP/USD ตัวแบบ Combined ให้ผลความแม่นยำในการพยากรณ์มากกว่าตัวแบบ ARIMAANN และ ARIMASVM และให้ผลความแม่นยำมากกว่าตัวแบบ Hybrid ในงานวิจัยของ (Zhang, 2003)

6.4 เปรียบเทียบประสิทธิภาพตัวแบบตามลักษณะข้อมูล

ข้อมูลทั้ง 10 ชุดที่นำมาใช้ในการพยากรณ์มีลักษณะทางข้อมูลที่แตกต่างกัน สามารถจัดเรียงประสิทธิภาพของแต่ละตัวแบบกับลักษณะข้อมูลที่แตกต่างกันได้ ดังตารางที่ 6.1

ลักษณะข้อมูล	ข้อมูล	เรียงลำดับความแม่นยำตัวแบบ
มีความแปรปรวนไม่คงที่	Sunspot	1.Combined 2.ANN 3.ARIMASVM
มีการเพิ่มขึ้นหรือลดลงเป็นแบบ Exponential	Canadian lynx	1.Combined 2.ANN 3.SVM
มีการขึ้นลงแบบสุ่ม ปริมาณการขึ้นลงคงที่	GBP/USD	1.Combined และ ANN 2.ARIMAANN
	Gold	1.ARIMAANN 2.ARIMASVM 3.Combined
	USexchange	1.Combined 2.ARIMAANN 3.ANN
มีค่าสุดโต่ง	Motorcycle	1.Combined 2.ANN 3.ARIMASVM
มีปัจจัยฤดูกาลที่แน่นอน	Electricuse	1.Combined 2.ARIMAANN

		3.ARIMASVM
เป็นข้อมูลที่ซับซ้อน วุ่นวาย (Chaos series)	Windspeed	1.Combined 2.ANN 3.ARIMAANN
มีปัจจัยฤดูกาลที่ไม่แน่นอน ตำแหน่งที่ต่ำสุด สูงสุด มีความคลาดเคลื่อนข้อมูล	Meanrain	1.ARIMAANN และ Combined 2.SVM
	Maxtemp	1.Combined 2.ANN 3.SVM

ตารางที่ 6.1 เรียงลำดับประสิทธิภาพของตัวแบบตามลักษณะข้อมูล

ตารางที่ 6.1 แสดงให้เห็นว่าเมื่อแยกประเภทข้อมูลตามลักษณะที่แสดงในตาราง ตัวแบบ Combined (ARIMA+ANN+SVM) ที่นำเสนอในงานวิจัยชิ้นนี้ ให้ความแม่นยำของการพยากรณ์มาเป็นอันดับที่หนึ่งในทุกชุดข้อมูล ยกเว้นข้อมูล Gold เท่านั้น จึงเป็นตัวแบบที่เหมาะกับข้อมูลที่มีลักษณะที่หลากหลาย เนื่องมาจากตัวแบบเดี่ยวที่มา Combined กันนั้นเป็นตัวแบบดึงเอาจุดเด่นของตัวแบบเดี่ยวๆ ที่แตกต่างกัน กล่าวคือตัวแบบ ARIMA โดดเด่นในการจับลักษณะข้อมูลที่มีความสัมพันธ์เชิงเส้นตรงได้ดี ขณะที่ตัวแบบ ANN และ SVM ก็เป็นตัวแบบที่ไม่เป็นเชิงเส้นตรง ที่มีวิธีการสร้างตัวแบบที่แตกต่างกันและเป็นตัวแบบที่มีประสิทธิภาพสูง

6.5 บทวิจารณ์และข้อเสนอแนะตัวแบบเดี่ยว

ตัวแบบ ARIMA

ตัวแบบ ARIMA ให้ผลการพยากรณ์ที่ดี ในข้อมูล Electricuse, USexchange และ GBP/USD ลักษณะข้อมูล Electricuse เป็นข้อมูลที่มีปัจจัยฤดูกาลที่แน่นอน ไม่ค่อยแกว่งหรือส่ายไปมา ขณะที่ Usexchange และ GBP/USD ข้อมูลมีการแกว่งขึ้นลงในรูปแบบสุ่ม แต่ขนาดการแกว่งค่อนข้างคงที่

สำหรับข้อมูลที่มีปัจจัยฤดูกาลประกอบ ได้แก่ปริมาณน้ำฝน, ระดับอุณหภูมิ และปริมาณการใช้ไฟฟ้า อันดับ p ของ ARIMA จะสูงใกล้เคียงกับรอบวงของฤดูกาล ขณะที่ข้อมูลทางการเงินที่ขึ้นลงแบบสุ่ม ไม่ได้มีปัจจัยฤดูกาล

สำหรับข้อมูลความเร็วลม ลักษณะข้อมูลมีการขึ้นลงแบบสุ่ม ความแปรปรวนไม่คงที่ สังเกตได้ว่าอันดับ q จะมาก เนื่องจากอันดับ q จะเกี่ยวข้องกับค่า error ใน Lag ก่อนหน้า

อันดับ p และ อันดับ q ที่อธิบายข้างต้น นั้นผู้พยากรณ์อาจกำหนดขอบเขตการค้นหาอันดับที่ดีที่โดยพิจารณาลักษณะข้อมูล เป็นรายข้อมูล ความเข้าใจข้อมูลตรงนี้ จะช่วยให้การค้นหาอันดับเป็นไปได้เร็วขึ้น

ตัวแบบ ANN

ตัวแบบ ANN ให้ผลการพยากรณ์ที่ดีกับข้อมูลแทบทุกชุด ดีกว่าตัวแบบ ARIMA ทุกชุดข้อมูล แสดงให้เห็นว่าข้อมูลที่พบในความเป็นจริงมีลักษณะที่ซับซ้อน มีการส่ายไปมา มีส่วนประกอบที่เป็นเชิงเส้นตรงและไม่เป็นเชิงเส้นตรงปะปนกัน

ข้อเสียของตัวแบบ ANN คือต้องทำการทดลองแทนค่าพารามิเตอร์หลายๆค่า ค่าพยากรณ์ที่ได้จากชุดของค่าพารามิเตอร์ที่นำมาสร้างตัวแบบแต่ละชุด ไม่ค่อยมีความสัมพันธ์กัน ไม่แสดงถึงทิศทางหรือแนวโน้มที่ชัดเจน

ตัวแบบ SVM

ให้ผลการพยากรณ์ดีกว่า ARIMA แสดงให้เห็นว่าข้อมูลที่พบในความเป็นจริงมีลักษณะที่ซับซ้อน มีการส่ายไปมา มีส่วนประกอบที่เป็นเชิงเส้นตรงและไม่เป็นเชิงเส้นตรงปะปนกัน ประสิทธิภาพเป็นรองตัวแบบ ANN บ้างในบางข้อมูล ผลการพยากรณ์มีความแม่นยำกับข้อมูลที่เกี่ยวข้องกับทางการเงิน (Financial datasets) เช่น USexchange, Gold, Motorcycle, GBP/USD ข้อมูลกลุ่มนี้จะมีลักษณะขึ้นลงแบบสุ่มแต่ปริมาณการขึ้นลงจะค่อนข้างคงที่

ข้อเสนอแนะ: ใช้ฮิวริสติกค้นหาค่าพารามิเตอร์ เช่น Genetic algorithm (GA), Particle swarm optimization (PSO) หรือ อื่นๆ มาช่วยค้นหาพารามิเตอร์ที่ทำให้ตัวแบบมีความแม่นยำ ในเวลาที่น้อยลง ตัวแบบ SVM ที่มีงานวิจัยที่นำฮิวริสติกเหล่านี้มาใช้กันมาก เนื่องจากพารามิเตอร์ของ SVM ดังภาพที่ 4.14, 4.15 และ 4.16 สามารถมองเห็นทิศทางแนวโน้มของค่าพารามิเตอร์ที่ส่งผลต่อค่า MSE ได้ชัดเจนกว่าตัวแบบ ANN

6.6 บทวิจารณ์และข้อเสนอแนะตัวแบบ Hybrid

สำหรับตัวแบบ Hybrid ARIMAANN และ ARIMASVM ที่เป็นที่ถกเถียงกันว่าดีกว่าตัวแบบเดี่ยวหรือไม่นั้น งานวิจัยชิ้นนี้สรุปว่า ไม่ได้ดีไปกว่าตัวแบบเดี่ยวในทุกชุดข้อมูล เป็นไปได้ว่าความสัมพันธ์ระหว่างส่วนที่เป็นเชิงเส้นตรงและส่วนที่ไม่เป็นเชิงเส้นตรง อาจไม่เป็นผลบวกในทุกชุดข้อมูล (Taskaya-Temizel and Casey, 2005) อาจทดลองปรับเปลี่ยนในรูปแบบความสัมพันธ์อื่น เช่น การคูณ

ข้อเสนอแนะ: เป็นที่น่าสังเกตว่าตัวแบบผสมนี้ ให้ค่าพยากรณ์ที่ดี เมื่อตัวแบบ ARIMA ให้ค่าพยากรณ์ที่ดีด้วย เช่นข้อมูล Electricuse, Usexchange, GBP/USD, Motorcycle และ Gold โดยเฉพาะ Gold ที่ตัวแบบ ARIMAANN และ ARIMASVM ให้ผลความแม่นยำสูงกว่าตัวแบบเดี่ยว ARIMA, ANN และ SVM ดังนั้นการนำตัวแบบที่เป็นเชิงเส้นตรง อื่นๆ มาประยุกต์ใช้ เช่น Exponential smoothing (ESM), GARCH, E-GARCH หรือตัวแบบอื่นๆ นำมาผสมกับตัวแบบ ANN และ SVM อาจทำให้ตัวแบบผสมมีความแม่นยำเพิ่มมากยิ่งขึ้น

6.7 บทวิจารณ์และข้อเสนอแนะตัวแบบ Hybrid combined

ตัวแบบ Hybrid combined ที่นำเสนอในงานวิจัยชิ้นนี้ให้ผลความแม่นยำเหนือกว่าตัวแบบอื่นทั้งตัวแบบ Hybrid และตัวแบบเดี่ยวเกือบทุกชุดข้อมูล ตัวแบบ Combined ที่นำเสนอนี้ใช้ตัวแบบเดี่ยวๆ คือ ARIMA, ANN และ SVM ซึ่งเป็นตัวแบบเดี่ยวที่โดดเด่นในแง่มุมของแต่ละตัวแบบ ตัวแบบ ARIMA จับลักษณะความสัมพันธ์ที่เป็นเชิงเส้นตรงของข้อมูลได้ดี ขณะที่ตัวแบบ ANN และ SVM เป็นตัวแบบที่จับลักษณะความสัมพันธ์ที่ไม่เป็นเชิงเส้นตรงของข้อมูลได้ดีและมีวิธีการสร้างตัวแบบที่แตกต่างกัน เมื่อนำมา Combined กันก็ช่วยให้จับลักษณะข้อมูลทั้งที่เป็นเชิงเส้นตรงและไม่เป็นเชิงเส้นตรงได้ดียิ่งขึ้น จัดเป็นตัวแบบที่ช่วยลดความผิดพลาดในการเลือกตัวแบบ ซึ่งอาจเลือกตัวแบบที่ไม่เหมาะสม สามารถผสมตัวแบบได้มากกว่าสองตัวแบบ โดยที่กระบวนการไม่ยุ่งยากมากนัก กล่าวคือสามารถพยากรณ์ข้อมูลทั้งสามตัวแบบพร้อมๆ กัน แล้วนำเข้า Combined จึงเหมาะเป็นอย่างยิ่งที่จะนำตัวแบบนี้ไปใช้ในพยากรณ์ข้อมูลที่มีความซับซ้อน ข้อมูลที่ยากแก่การพิจารณาว่าควรใช้ตัวแบบใดและต้องการความแม่นยำในการพยากรณ์สูง

ข้อเสีย คือ ใช้เวลาในการคำนวณที่นานกว่าตัวแบบอื่นๆแต่ก็ไม่ได้สูงมากนักใช้เวลาไม่เกินสองชั่วโมง ซึ่งสามารถทำการพยากรณ์ข้อมูลรายวันหรือรายครึ่งวันได้ทันเวลา แต่ก็มีเทคนิคที่สามารถทำการพยากรณ์ไปพร้อมๆ กันได้ เช่นเปิดหน้าโปรแกรมหลายหน้าจอปล่อยให้รันพร้อมๆกัน การรันโปรแกรมแบบทุกหน่วยประมวลผลพร้อมๆกัน หรือการใช้เครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลที่มีความเร็วสูง

ข้อเสนอแนะ:

1. ตัวแบบ Combined ให้ผลความแม่นยำน้อยกว่า ARIMAANN และ ARIMASVM สำหรับข้อมูล Gold อาจเนื่องมาจากตัวแบบ ARIMA พยากรณ์ข้อมูลชุดนี้ด้วยความแม่นยำที่สูง ดังนั้นเมื่อนำมาผสมกับตัวแบบ ANN หรือ SVM ก็ทำให้ตัวแบบ ARIMAANN และ ARIMASVM มีความแม่นยำที่สูงยิ่งขึ้นและสูงกว่าตัวแบบ Combined

2. ในการค้นหาค่าน้ำหนักของแต่ละตัวแบบสามารถนำเครื่องมือค้นหาค่าน้ำหนัก เช่น GA, PSO อื่นๆ มาช่วยในการค้นหา และน่าจะช่วยลดเวลาอย่างมากถ้า ใช้ตัวแบบมากกว่า 3 ตัวแบบมาทำการ Combined กัน

Chulalongkorn University

รายการอ้างอิง

BEALE, M. H., MARTIN T.HAGAN & HOWARD B. DEMUTH 2013. *Neural Network ToolboxTM: User's Guide R2013b*, TheMathWorks, Inc.

BEKIRKARLIK & A. VEHBI OLGAC 2010. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks *International Journal of Artificial Intelligence And Expert Systems* 1.

CHEN, K.-Y. & WANG, C.-H. 2007. A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Systems with Applications,* 32**,** 254-264.

DE GROOT, C. & WüRTZ, D. 1991. Analysis of univariate time series with connectionist nets: A case study of two classical examples. *Neurocomputing,* 3**,** 177-192.

G.E.P. BOX & G. JENKINS 1970. *Time Series Analysis, Forecasting and Control,* San Francisco,CA, Holden-Day.

GUTIéRREZ-ESTRADA, J. C., SILVA, C., YáñEZ, E., RODRíGUEZ, N. & PULIDO-CALVO, I. 2007. Monthly catch forecasting of anchovy Engraulis ringens in the north area of Chile: Non-linear univariate approach. *Fisheries Research,* 86**,** 188-200.

HONG, W.-C., DONG, Y., CHEN, L.-Y. & LAI, C.-Y. 2010. Taiwanese 3G mobile phone demand forecasting by SVR with hybrid evolutionary algorithms. *Expert Systems with Applications,* 37**,** 4452-4462.

JAMES V. HANSEN & RAY D. NELSON 2010. Time-series analysis with neural networks and ARIMA-neural network hybrids. *Journal of Experimental & Theoretical Artificial Intelligence,* 15**,** 315-330.

KHASHEI, M. & BIJARI, M. 2011. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing,* 11**,** 2664-2675.

NIE, H., LIU, G., LIU, X. & WANG, Y. 2012. Hybrid of ARIMA and SVMs for Short-Term Load Forecasting. *Energy Procedia,* 16, Part C**,** 1455-1460.

ÖMER FARUK, D. 2010. A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence,* 23**,** 586-594.

PAI, P.-F. & LIN, C.-S. 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega,* 33**,** 497-505.

SHI, J., GUO, J. & ZHENG, S. 2012. Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renewable and Sustainable Energy Reviews,* 16**,** 3471-3480.

TASKAYA-TEMIZEL, T. & CASEY, M. C. 2005. A comparative study of autoregressive neural network hybrids. *Neural Networks,* 18**,** 781-789.

TSENG, F.-M., YU, H.-C. & TZENG, G.-H. 2002. Combining neural network model with seasonal time series ARIMA model. *Technological Forecasting and Social Change,* 69**,** 71-87.

VAPNIK V 1995. *The nature of statistic learning theory,* New York, Springer.

WANG, J.-J., WANG, J.-Z., ZHANG, Z.-G. & GUO, S.-P. 2012. Stock index forecasting based on a hybrid model. *Omega,* 40**,** 758-766.

WANG, J., ZHU, S., ZHANG, W. & LU, H. 2010. Combined modeling for electric load forecasting with adaptive particle swarm optimization. *Energy,* 35**,** 1671-1678.

WILLIAMS, G. 2011. *Data mining with rattle and R: The art of excavating data for knowledge discovery*, Springer.

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

ZHANG, G. P., PATUWO, B. E. & HU, M. Y. 2001. A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers & Operations Research,* 28**,** 381-396.

ธนาคารแห่งประเทศไทย. 2555a. เครื่องชี้การใช้จ่ายเพื่อการอุปโภคบริโภคภาคเอกชน [Online]. Available: http://www2.bot.or.th/statistics/BOTWEBSTAT.aspx?reportID=108&language=TH [Accessed 23 กันยายน 2555.

ธนาคารแห่งประเทศไทย. 2555b. อัตราแลกเปลี่ยนเงินตราต่างประเทศย้อนหลัง [Online]. Available: http://www2.bot.or.th/statistics/BOTWEBSTAT.aspx?reportID=108&language=TH [Accessed 23 กันยายน 2555.

สำนักพัฒนาอุตุนิยมวิทยา 2555. รายงานสภาวะอากาศของประเทศไทยตั้งแต่เดือนมกราคม พ.ศ. 2545 ถึงเดือน ตุลาคมคม พ.ศ. 2555. *In:* ศูนย์ภูมิอากาศ & สำนักพัฒนาอุตุนิยมวิทยา (eds.). กรมอุตุนิยมวิทยา.

ห้างขายทองทองใบเยาวราช. 2555. ราคาทองคำย้อนหลัง [Online]. Available: http://www.thongbai.com/modi.php?orderdate1=2011-01-01&orderdate2=2012-09-17&Submit=Submit [Accessed 23 กันยายน 2555.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวกส่วนนี้จะแสดง Code การสร้างโปรแกรม Forecasting เพื่อให้ผู้อ่านสามารถใช้งานตัวแบบผสมได้จริงหรือทำมาใช้ศึกษาต่อยอดได้สะดวกมากยิ่งขึ้น ซึ่ง Code การเขียนโปรแกรมเป็น Code ที่ใช้ในโปรแกรม R ซึ่งแบ่งออกเป็นสองส่วนหลักคือ

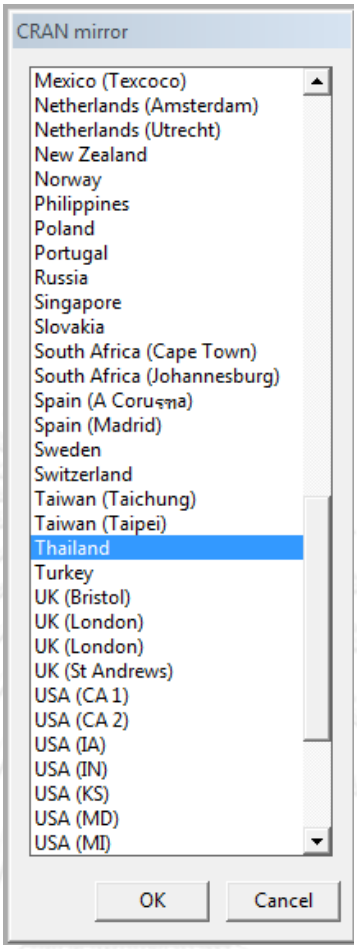
1) ส่วนโปรแกรมเชิงวัตถุ คือ Code ที่มีหน้าที่สร้างหน้าต่างรับเข้าข้อมูลและแสดงผลข้อมูล เช่นเป็นช่องให้ใส่ค่า ช่องให้เลือกหรือไม่เลือก (ติ๊กเครื่องหมายถูกหรือไม่ติ๊ก) แทบเมนูที่เลือกลงไปเป็นชั้นๆ เป็นต้น
2) ส่วนโปรแกรมเชิงการคำนวณ คือ Code ที่มีไว้ในการคำนวณ ตั้งแต่เมื่อรับข้อมูลมาทำการจัดการข้อมูลก่อนเข้าตัวแบบ สร้างตัวแบบ เขียนโดยอาศัยทฤษฎีทางคณิตศาสตร์ของแต่ละตัวแบบ

หลังจากที่ติดตั้งโปรแกรม R ในคอมพิวเตอร์เรียบร้อยแล้วก็ต้อง Install packages ที่ใช้ในโปรแกรม Forecasting โดยทำการ Copy คำสั่งทั้ง 6 บรรทัดด้านล่างนี้แล้วนำไป Paste ในหน้าต่าง R console

```
install.packages('tseries')

install.packages('forecast')

install.packages('nnet')

install.packages('tcltk')

install.packages('xlsx')

install.packages('kernlab')
```

เมื่อ Paste แล้วจะปรากฏหน้าต่างให้เลือกประเทศที่จะดาวน์โหลด Packages ต่างๆที่จำเป็นต้องใช้มาติดตั้งในคอมพิวเตอร์ ก็ให้เลือกประเทศไทย ดังภาพที่ 1

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 1 หน้าต่างให้เลือกประเทศที่จะดาวน์โหลด Packages มาติดตั้ง

จากนั้นทำการโหลด Packages ต่างๆนำมาใช้ โดยการ Copy คำสั่งทั้ง 6 บรรทัดด้านล่างนี้ แล้วนำไป Paste ในหน้าต่าง R console

library(tseries)

library(forecast)

library(nnet)

library(tcltk)

library(xlsx)

library(kernlab)

เมื่อโหลดเสร็จเรียบร้อยก็ทำการ Copy โค้ดที่ใช้เขียนโปรแกรมทุกคำสั่งทุกบรรทัดด้านล่าง และ Paste ในหน้าต่าง R console รอสักครู่ ก็จะได้หน้าต่างโปรแกรม Forecasting ดังภาพที่ 5.2

ส่วนฟังก์ชันที่เขียนขึ้นเพื่อใช้ใน GUI

```
graph<-function(predict,real,type,main2){
plot(time(ts(predict)),realtype="type",ylab="data",xlab="time",main=paste(main2,"Real test data VS Predicted
value"),ylim=c(0,ifelse(max(predict)>max(real),max(predict)+mean(predict),max(real)+mean(predict))))
lines(ts(predict),col="red")
 legend("topleft",c(paste("test data"),paste("predicted data")),lty=c(1,1),col=c("black","red"),cex=0.9)
}
graph2<-function(predict,real,type,main2){
  plot(as.numeric(real),type="l",ylab="data",xlab="time",main=paste(main2,"Real test data VS Predicted
value"),ylim=c(0,ifelse(max(predict)>max(real),max(predict)+mean(predict),max(real)+mean(predict))))
  lines(as.numeric(predict),col="red")
  legend("topleft",c(paste("test data"),paste("predicted data")),lty=c(1,1),col=c("black","red"),cex=0.9)
}
graph3<-function(predict,real,type,main2){
  plot(time(ts(predict)),real,type="type",ylab=paste("trainning data"),xlab="time",main=paste(main2,"Real test data VS Predicted
value"),ylim=c(0,ifelse(max(predict)>max(real),max(predict)+mean(predict),max(real)+mean(predict))))
  lines(ts(predict),col="red")
  legend("topleft",c(paste("train data"),paste("fitted value")),lty=c(1,1),col=c("black","red"),cex=0.9)
}
graph4<-function(predict,real,type,main2){
  plot(as.numeric(real),type="l",ylab=paste("trainning data"),xlab="time",main=paste(main2,"Train data VS fitted
value"),ylim=c(0,ifelse(max(predict)>max(real),max(predict)+mean(predict),max(real)+mean(predict))))
  lines(as.numeric(predict),col="red")
  legend("topleft",c(paste("train data"),paste("fitted value")),lty=c(1,1),col=c("black","red"),cex=0.9)
}
rMSE<-function(res,len) sqrt(sum((res)^2)/len)
mSE<-function(res,len) sum(res^2)/len
mAE<-function(res,len) sum(abs(res))/len
mAPE<-function(res,data.test,len) sum(abs(res/data.test))*100/len
lagmatrix <- function(x,max.lags) embed(c(rep(NA,max.lags), x), max.lags+1)

#2 0-1 transform
l.tran<-function(data) (data-min(data))/(max(data)-min(data));re.l.tran<-function(data.pre,data) data.pre*(max(data)-min(data))+min(data)
#3 standardize transform
l.std<-function(data) (data-mean(data))/sd(data);re.l.std<-function(data.pre,data) data.pre*sd(data)+mean(data)
#4 -1 1 transform
l.tran2<-function(data) ((data-min(data))*2/(max(data)-min(data)))-1;re.l.tran2<-function(data.pre,data) 0.5*((data.pre+1)*(max(data)-
min(data)))+min(data)

#  ฟังก์ชัน Input ค่า
input.entry<-function(x,integer=TRUE){
  x.<-strsplit(paste(strsplit(x," ")[[1]],collapse=""),",")[[1]] # ทำให้เป็น "25""0.4"
  y.<-as.numeric(x.)
  if(all(!is.na(y.))) {
    if(!integer | all(y.==round(y.))) {
      result<-y.
    } else {
      result<-paste("Error: input is not integer '",paste(y.[y.!=round(y.)],collapse=","),"'")
    }
  } else {
    result<-paste0("Error: unexpected symbol in '",paste(x.[is.na(y.)],collapse=","),"'")
  }
  return(result)
```

```
}

# ส่วนฟังก์ชัน GUI ทั้งหมด
# เปิดหน้าต่างใหญ่
tt <- tktoplevel(width=600,height=500)
tkwm.title(tt, "Forecasting")

# 0. กำหนดให้หน้าต่างใหญ่มี tab menu
topMenu <- tkmenu(tt)
tkconfigure(tt, menu = topMenu)

#1. tab: Import แตก tab menu แบบบนลงล่าง
import <- tkmenu(topMenu,tearoff=FALSE)

# 1.1 tab: Import text file
tkadd(import, "command" , label = "Import text file" , command =function() set.source2(tt))

# 1.2 tab: Import excel file
tkadd(import, "command" , label = "Import excel file" , command =function() set.source(tt))
tkadd(topMenu, "cascade" , label = "Import data", menu = import)


# ฟังก์ชันการ import Excel ข้อมูล สำหรับ(1.2)
set.source<-function(tt0){
        file.name<-tclvalue(tkgetOpenFile(filetypes="{{Excel97-2003 Excel2007} {.xls .xlsx}}"))
        if(file.name!=""){
                tt <- tktoplevel()  #เปิดหน้าต่างใหม่
                tkwm.title(tt, "Import data from Excel")  #ตั้งหัวชื่อหน้าต่าง
# tkgetOpenFile สร้างหน้าต่างเปิดไฟล์ที่ใช้เรียกชื่อ file.name

 #เนื้อหาข้างในหน้าต่าง
        ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
        tkgrid(tklabel(ttframe1,text="\n              Select one sheet from 1,2,3,...          "))
 # ใส่ข้อความด้วย tklabel

        ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
        sheet.value<-tclVar("") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
        sheetentry<-tkentry(ttframe2,textvariable=sheet.value,width=8)
        tkgrid(tklabel(ttframe2,text="Select sheet :"),sheetentry)

        ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
        column.value<-tclVar("")
        columnentry<-tkentry(ttframe3,textvariable=column.value,width=8)
        tkgrid(tklabel(ttframe3,text="Select column index:"),columnentry)


        ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
        process.but <-tkbutton(ttframe4,text="   OK   ",command=function() onProcess()) #สร้างปุ่ม "OK" กดแล้วให้ผลยังไงดูที่ command
        tkgrid(process.but)

        tkpack(ttframe1,ttframe2,ttframe3,ttframe4)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม tkpack
        tkfocus(tt) #ปรับความสนใจไปที่หน้าต่างใหม่ ให้พร้อมใช้งาน

        onProcess<-function(){  # ตรงนี้จะลิงค์กับปุ่ม OK ที่อ้างถึงข้างบน
```

```
    sheet<-as.numeric(tclvalue(sheet.value))  # ดึงค่าจาก sheet.value
            column<-as.numeric(tclvalue(column.value))    # column.value

    if(is.na(sheet)){ #เชคเงื่อนไขถ้า sheet ว่างแล้วกดโอเค จะขึ้นข้อความดัง message box
                    tkmessageBox(message="Please input sheet index",icon="error",title="Error")
                    tkfocus(tt)
            } else if(is.na(column)){
                    tkmessageBox(message="Please input column index",icon="error",title="Error")
                    tkfocus(tt)
            } else {
                    real<-na.omit(ts(read.xlsx(file.name,sheetIndex=sheet,colIndex=column)))
                    colnames(real)<-"data"
                    assign("observation",real,envir=.GlobalEnv) # observation จะเป็นตัวแปรที่ใช้ต่อใน R บวกลบคูณหาร ทำต่อได้
                    tkmessageBox(message="Import data successfully")
                    print(real); plot(real)
                    tkdestroy(tt)
                    tkfocus(tt0)
     }
  }
} else tkfocus(tt0)
}

# ฟังก์ชันการ import Text ข้อมูล สำหรับ(1.1)
set.source2<-function(tt0){
            file.name<-tclvalue(tkgetOpenFile(filetypes="{{Text files} {.txt .tab .csv}}"))
            if(file.name!=""){
            file.type<-substr(file.name,nchar(file.name)-3,nchar(file.name))  #ระบุชนิดไฟล์ โดยดูจาก นามสกุลของไฟล์
            real<-read.csv(file.name,sep=switch(file.type,.txt="",.tab="\t",.csv=","))
            if(ncol(real)>1) {
                    tt <- tktoplevel()  #ถ้าไฟล์ที่เรียกมีมากกว่า 1 คอลัมภ์ เปิดหน้าใหม่ ต้องเขียน tktoplevel ไว้บนก่อน
                    tkwm.title(tt, "Import data from text files")

            ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
            column.value<-tclVar("")
            columnentry<-tkentry(ttframe1,textvariable=column.value,width=8)
            tkgrid(tklabel(ttframe1,text="")) # tkgrid จะทำให้เกิดบรรทัดใหม่ "" เป็นการเว้นบรรทัด ให้ดูสวยงาม
            tkgrid(tklabel(ttframe1,text=paste("          Select one column index from 1 to",ncol(real),":")),
                    columnentry,tklabel(ttframe1,text="          "))

            ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
            process.but <-tkbutton(ttframe2,text="   OK   ",command=function() onProcess())
            tkgrid(process.but)

            tkpack(ttframe1,ttframe2)
            tkfocus(tt)

            onProcess<-function(){
            column<-as.numeric(tclvalue(column.value))
            if(is.na(column)){
                    tkmessageBox(message="Please input column index",icon="error",title="Error")
                    tkfocus(tt)
            } else if(!is.element(column,1:ncol(real))) {
              tkmessageBox(message="Incorrect column index",icon="error",title="Error")
              tkfocus(tt)
```

Chulalongkorn University

```
            } else {
                        real<-data.frame(real[,column])
                        tkdestroy(tt)
                        colnames(real)<-"data"
                        assign("observation",na.omit(ts(real)),envir=.GlobalEnv)
                        tkmessageBox(message="Import data successfully")
                        print(observation) ; plot(observation)
                        tkfocus(tt0)

            }
            }
            } else {

            colnames(real)<-"data"
            assign("observation",na.omit(ts(real)),envir=.GlobalEnv)
                        tkmessageBox(message="Import data successfully")
                        print(observation) ; plot(observation)
                        tkfocus(tt0)
            }
            } else tkfocus(tt0)
}

#2. tab: Preprocess
preprocess <- tkmenu(topMenu,tearoff=FALSE)
# 2.1
tkadd(preprocess, "command" , label = "Descriptive" , command =function() {
  if(exists("observation")) {
  print(summary(observation))
  } else {
  tkmessageBox(message="Please import your datasets before show descriptive",icon="error",title="Error")
  tkfocus(tt0)
  }
}
)

# 2.2
divide <- tkmenu(topMenu,tearoff=FALSE)

tkadd(divide, "command" , label = "Order split" , command =function() split.fun(tt,"Order split"))
tkadd(divide, "command" , label = "% split" , command =function() split.fun(tt,"% split"))

split.fun<-function(tt0,Gr)  {
  if(exists("observation")) { # คำสั่ง exists เอาไว้ตรวจดูว่ามีตัวแปรนี้อยู่ใน ls() บอกตป ที่พร้อมเรียก
  tt <- tktoplevel()  #ถ้าไฟล์ที่เรียกมีมากกว่า 1 คอลัมภ์ เปิดหน้าใหม่ ต้องเขียน tktoplevel ไว้บนก่อน
  tkwm.title(tt, Gr)

  ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
  tkgrid(tklabel(ttframe1,text=paste("\n        Number of observation:",length(observation),"        "))) #\n เว้นบรรทัดหนึ่งด้านบน

  ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
  order.value<-tclVar("")  # สำหรับกล่องรับค่าที่ใส่เอง
  orderentry<-tkentry(ttframe2,textvariable=order.value,width=8)
  tkgrid(tklabel(ttframe2,text="")) # tkgrid จะทำให้เกิดบรรทัดใหม่ "" เป็นการเว้นบรรทัด ให้ดูสวยงาม
  tkgrid(tklabel(ttframe2,text=ifelse(Gr=="Order split","Length of data for training","% spliting for training")),orderentry)
```

```
  ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
  process.but <-tkbutton(ttframe3,text="    OK    ",command=function() onProcess())
  tkgrid(process.but)

  tkpack(ttframe1,ttframe2,ttframe3)
  tkfocus(tt)

  onProcess<-function(){
    no.tr<-as.numeric(tclvalue(order.value))
    if(is.na(no.tr)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
      tkmessageBox(message="Please input specific length of training data",icon="error",title="Error")
      tkfocus(tt)
    } else {
      if(Gr=="% split") no.tr<-no.tr*length(observation)/100
        no.tr<-round(no.tr)
      if(no.tr<length(observation)/2) {  # ถ้าจำนวนเทรนที่ใส่น้อยกว่า ครึ่งของข้อมูลทั้งหมด จะแสดงข้อความเตือน
        tkmessageBox(message="Not enough training data",icon="error",title="Error")
        tkfocus(tt)
      } else {
        mwe<-rep(c("Training sets","Test sets"),c(no.tr,length(observation)-no.tr))
        mou<-split(observation,mwe)
        assign("train.observation",mou[["Training sets"]],envir=.GlobalEnv) # observation จะเป็นตัวแปรที่ใช้ต่อใน R
        assign("test.observation",mou[["Test sets"]],envir=.GlobalEnv) # [[ ]] แปลงจาก list เป็น unlist
        assign("ahead",length(test.observation),envir=.GlobalEnv)
        tkmessageBox(message="Split data successfully")
        print(mou); windows()
        plot(observation) ;abline(v=no.tr+0.5,col="red",lty=3,lwd=2)
        tkdestroy(tt)
        tkfocus(tt0)
      }
    }
  }
} else {
  tkmessageBox(message="Please import your datasets before splitting",icon="error",title="Error")
  tkfocus(tt0)
}
}

tkadd(preprocess, "cascade" , label = "Divide datasets", menu = divide)
tkadd(topMenu, "cascade" , label = "Preprocess " , menu =preprocess)


#3. tab: Modeling
model <- tkmenu(topMenu,tearoff=FALSE)

# 3.1 tab: ARIMA
tkadd(model, "command" , label = "ARIMA" , command =function() ARIMA(tt))

ARIMA<-function(tt0){
  if(exists("train.observation")) {
    tt <- tktoplevel()  #เปิดหน้าต่างใหม่
    tkwm.title(tt, "ARIMA")  #ตั้งชื่อหน้าต่างตาม " "
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
#เนื้อหาข้างในหน้าต่าง 2 อันนี้คู่กัน จะแสดงผลหลังฟังก์ชัน tkpack
ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
tkgrid(tklabel(ttframe1,text="\nInput ARIMA Parameter\n"))

ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
maxp.value<-tclVar("15") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
maxpentry<-tkentry(ttframe2,textvariable=maxp.value,width=8)     # ทำช่องใส่ค่า หลัง tclVar
tkgrid(tklabel(ttframe2,text="Input maximum of order p:"),maxpentry,tklabel(ttframe2,text="    "))
tkgrid(tklabel(ttframe2,text="    * (p should be non-negative integer between 0 and 15)    \n"))


ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
maxq.value<-tclVar("6")
maxqentry<-tkentry(ttframe3,textvariable=maxq.value,width=8)
tkgrid(tklabel(ttframe3,text="Input maximum of order q :"),maxqentry,tklabel(ttframe3,text="    "))
tkgrid(tklabel(ttframe3,text="    * (q should be non-negative integer between 0 and 6)    \n"))

# วิธีการทำ Tik box
ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
fittt.arima<-tkcheckbutton(ttframe4)
scatter.plot<-tkcheckbutton(ttframe4)
box.ljung.test<-tkcheckbutton(ttframe4)
ACF<-tkcheckbutton(ttframe4)
Histogram<-tkcheckbutton(ttframe4)

  fitarima.value<-tclVar("0")
  scatter.value<-tclVar("0")
  box.test.value<-tclVar("0")
  ACF.value<-tclVar("0")
  Histogram.value<-tclVar("0")

  tkconfigure(fittt.arima,variable=fitarima.value)
  tkconfigure(scatter.plot,variable=scatter.value)
  tkconfigure(box.ljung.test,variable=box.test.value)
  tkconfigure(ACF,variable=ACF.value)
  tkconfigure(Histogram,variable=Histogram.value)

  tkgrid(tklabel(ttframe4,text=""))

  tkgrid(tklabel(ttframe4,text=""),fittt.arima,tklabel(ttframe4,text="Fitted value of ARIMA"),tklabel(ttframe4,text="
"),scatter.plot,tklabel(ttframe4,text="Scatter plot of residuals"))

tkgrid(tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),box.ljung.test,tklabel(ttframe4,tex
t=" Box-Pierce or Ljung-Box test for independence    "))

tkgrid(tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),ACF,tklabel(ttframe4,text="Autoco
rrelation function of residuals"))

tkgrid(tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),tklabel(ttframe4,text=""),Histogram,tklabel(ttframe4,text="
Histogram of residuals"))

  tkgrid(tklabel(ttframe4,text=""))

ttframe5<-tkframe(tt, borderwidth=0, relief="groove")
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
process.but <-tkbutton(ttframe5,text="   OK   ",command=function() onProcess(train.observation,test.observation))
tkgrid(process.but)

tkpack(ttframe1,ttframe2,ttframe3,ttframe4,ttframe5)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม tkpack
tkfocus(tt) #ปรับความสนใจไปที่หน้าต่างใหม่ ให้พ

onProcess<-function(x.train,x.test){
  max.p<-as.numeric(tclvalue(maxp.value))
  max.q<-as.numeric(tclvalue(maxq.value))
  if(is.na(max.p)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum order of p",icon="error",title="Error")
    tkfocus(tt)
  } else if(!is.element(max.p,0:15)) {
    tkmessageBox(message="p must be non-negative integer between 0 and 15",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(max.q)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum order of q",icon="error",title="Error")
    tkfocus(tt)
  } else if(!is.element(max.q,0:6)) {
    tkmessageBox(message="q must be non-negative integer between 0 and 6",icon="error",title="Error")
    tkfocus(tt)
  } else {
    tkdestroy(tt)   # ลบหน้าจอใหม่ที่เด้งขึ้นมา
    tkfocus(tt0)    # คงหน้าจอเดิมตอนแรก

    # เป็นผลมาจาก Tik box เพื่อนำไปใช้เชคต่อไป
    Fit.aRiMa<-as.numeric(tclvalue(fitarima.value))
    sCaTter<-as.numeric(tclvalue(scatter.value))
    BoXtEsT<-as.numeric(tclvalue(box.test.value))
    aCf<-as.numeric(tclvalue(ACF.value))
    hIsToGrAm<-as.numeric(tclvalue(Histogram.value))

    # fit ARIMA
    fit<-auto.arima(ts(x.train),max.p=max.p,d=NA, max.q=max.q,max.P=5,D=NA,max.Q=5,max.order=40,stepwise=F,trace=T
,parallel=T,sea=T,test="adf",seasonal.test="ocsb",app=T)
    fit.fit<-as.numeric(fit$fit);fit.res<-as.numeric(fit$res)
    table1<-cbind(x.train,fit.fit,fit.res)
    colnames(table1)<-c("train data","fitted Arima","Residuals train data")
    graph4(fit.fit,x.train,"l",main2="ARIMA")

    # Compute test set
    Pr<-c() ;XX.test<-c()
    for(i in 1:(ahead+1)) {
      XX.test<-c(XX.test,x.test[i-1])
      datt<-c(x.train,XX.test)
      pr<-predict(arima(datt,order=c(fit$arm[1],fit$arm[6],fit$arm[2]),method='ML'),n.ahead=1)
      pr<-pr$pr ;Pr<-c(Pr,pr)
    }
    fit2.res<-x.test[1:ahead]-Pr[1:ahead]
    table2<-cbind(x.test[1:ahead],Pr[1:ahead],fit2.res)
    colnames(table2)<-c("test data","predicted Arima","Residuals test data")

    # Show result order of ARIMA,predicted value ,test data and residual # Module diagnostic model
    print(fit);print(table2)
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        print(test.error<-
c(RMSE=rMSE(fit2.res,ahead),MSE=mSE(fit2.res,ahead),MAE=mAE(fit2.res,ahead),MAPE=mAPE(fit2.res,x.test[1:ahead],ahead)))
        windows();if(ahead>=25) {
          graph2(predict=Pr[1:ahead],real=x.test[1:ahead],type="l",main2="ARIMA")
        } else  {
          graph(predict=Pr[1:ahead],real=x.test[1:ahead],type="l",main2="ARIMA")
        }

        # เป็นผลมาจาก Tik box==1 แสดงว่าเลือก ==0 ไม่เลิก
        if(Fit.aRiMa==1)  print(table1)
        if(sCaTter==1) { windows();plot(fit$res,type="p",ylab="Residuals",main="Plot of residuals for checking randomness");abline(h=0) }
        if(BoXtEsT==1) { print(test.res<-Box.test(fit$res,lag=10,type="Ljung",fitdf=1)) # Ho: residuals are random }
        }
        if(aCf==1) { windows();acf(fit$res,main="Autocorrelation function of residuals") }
        if(hIsToGrAm==1) {windows();hist(fit$res,freq=F,main="Histogram of residuals",ylab="Probability",xlab="value of residuals");
lines(density(fit$res,col="red")) }

        # predict future data one-step ahead
        print(pred.fut<-c(predict.future=Pr[ahead+1]))  # เอามาจากการทดสอบในอันดับ

        # นำไปใช้ต่อ
        assign("fitted.res",c(fit.res,fit2.res),envir=.GlobalEnv)
        assign("fitted.tr.res",fit.res,envir=.GlobalEnv)
        assign("fitted.te.res",fit2.res,envir=.GlobalEnv)
        assign("fitted.val",c(fit.fit,Pr[1:ahead]),envir=.GlobalEnv)
        assign("MSE.ARIMA",mSE(fit2.res,ahead),envir=.GlobalEnv)
        assign("pred.future.arima",pred.fut,envir=.GlobalEnv)
        assign("test.result.arima",test.error,envir=.GlobalEnv)
      }
    }
} else {
  tkmessageBox(message="Please split your datasets before modeling ARIMA",icon="error",title="Error")
  tkfocus(tt0)
}
}

# 3.2 tab: ANN
tkadd(model, "command" , label = "ANN" , command =function() ANN(tt))
ANN<-function(tt0){
  if(exists("train.observation")) {
  tt <- tktoplevel(width=350,height=200) #เปิดหน้าต่างใหม่
  tkwm.title(tt, "ANN")  #ตั้งหัวชื่อหน้าต่าง

  #เนื้อหาข้างในหน้าต่าง
  ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
  tkgrid(tklabel(ttframe1,text="\n                Input ANN Parameter              "))

  ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
  lag.value<-tclVar("2,3,4,6,8,12") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
  lagentry<-tkentry(ttframe2,textvariable=lag.value,width=15)
  tkgrid(tklabel(ttframe2,text="Input number of lag :"),lagentry,tklabel(ttframe2,text="    "))
  tkgrid(tklabel(ttframe2,text="    *Warning: (Lag should be positive integer between 1 and 15)   \n"))

  ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
hidden.value<-tclVar("10")
hiddenentry<-tkentry(ttframe3,textvariable=hidden.value,width=8)
tkgrid(tklabel(ttframe3,text="Input number of maximum hidden neurons :"),hiddenentry,tklabel(ttframe3,text="    "))
tkgrid(tklabel(ttframe3,text="    *Warning: (Maximum hidden neurons should be positive integer not exceed 20)   \n"))

ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
seed.value<-tclVar("1111,2332,5,337")
seedentry<-tkentry(ttframe4,textvariable=seed.value,width=20)
tkgrid(tklabel(ttframe4,text="Input random number generator :"),seedentry,tklabel(ttframe4,text="    "))
tkgrid(tklabel(ttframe4,text="    *Warning: (Seed number should be positive integer between 1 and 1e+09)   \n"))

ttframe5<-tkframe(tt, borderwidth=0, relief="groove")
time.value<-tclVar("10")
timeentry<-tkentry(ttframe5,textvariable=time.value,width=8)
tkgrid(tklabel(ttframe5,text="Limited time in computation (minute):"),timeentry,tklabel(ttframe5,text="    "))
tkgrid(tklabel(ttframe5,text="    *Warning: (Time must be positive integer in minute unit)   \n"))

ttframe6<-tkframe(tt, borderwidth=0, relief="groove")
process.but <-tkbutton(ttframe6,text="   OK   ",command=function() onProcess(train.observation,test.observation))
tkgrid(process.but)

tkpack(ttframe1,ttframe4,ttframe2,ttframe3,ttframe5,ttframe6)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม tkpack
tkfocus(tt)

onProcess<-function(x.train,x.test){
  lag.value<-input.entry(as.character(tclvalue(lag.value)),T)
  hidden.value<-as.numeric(tclvalue(hidden.value))
  seed.value<-input.entry(as.character(tclvalue(seed.value)),F)
  time.value<-as.numeric(tclvalue(time.value))
  if(is.na(lag.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input number of lag",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(lag.value)) {
    tkmessageBox(message=lag.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(hidden.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum number of hidden neurons",icon="error",title="Error")
    tkfocus(tt)
  } else if(!is.element(hidden.value,1:20)) {
    tkmessageBox(message="Hidden neurons must be positive integer between 1 and 20",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(seed.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input seed value",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(seed.value)){
    tkmessageBox(message=seed.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(max(abs(seed.value))>2e+9) {
    tkmessageBox(message="supplied seed is not valid ",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(time.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum time executing for ANN: default=20 minutes",icon="error",title="Error")
    tkfocus(tt)
  } else if(time.value<0) {
```

```
  tkmessageBox(message="Maximum executing time must be positive number",icon="error",title="Error")
  tkfocus(tt)
 } else {
  tkdestroy(tt)
  tkfocus(tt0)

  totalprogress<-length(lag.value)*hidden.value*length(seed.value)
  pb <- winProgressBar(title = "progress bar", min = 0, max =totalprogress , width = 300);
  ratio.prog<-0;
  setWinProgressBar(pb, ratio.prog, title=paste( "ANN progress:",round(ratio.prog/totalprogress*100),"% done"))
  init.t<-Sys.time()

  best.mse<-1e15 ;Lagg<-cbind(lagmatrix(observation,max(lag.value))) # นำค่าเมตริกซ์ lag ค่าจริง+ค่าเมตริกซ์ lag ค่า res
    for(i in lag.value) for(j in seq(hidden.value)){
      for(k in seed.value){
      diff.t0<-as.numeric(Sys.time()-init.t)
      ggal<-Lagg[,c(1:(i+1))]
      t.data<-as.data.frame(apply(na.omit(ggal),2,l.tran))
      t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
      set.seed(1111+k)
      net <- nnet(V1~.,data=t.data.tr, size=j,maxit=2e5,abstol = 1.0e-9,reltol=1.0e-9,trace=F,linout=T)
      pr<-re.l.tran(predict(net,t.data.te),ggal[,1]) ;pr.raw<-predict(net,t.data.te)
      MSE<-mSE((x.test-pr),ahead)
      if(MSE<best.mse){
        best.mse<-MSE ;NN<-net ;summary.net<-summary(net) ;Pr<-pr ;Pr.raw<-pr.raw;K<-k  ;nlag<-i ;J<-j ;GGAL<-ggal ;T.data.te<-t.data.te
      }
      # show time tube
      ratio.prog<-ratio.prog+1
      setWinProgressBar(pb, ratio.prog, title=paste( "ANN progress:",round(ratio.prog/totalprogress*100),"% done"))

      # count time
      diff.t<-as.numeric(Sys.time()-init.t)
      if(diff.t<diff.t0 & diff.t>time.value) {
        close(pb)
        stop(paste("Exceed executing time input",time.value,"minutes"))
      }
    }
  }
  # close tube ANN
  close(pb)

  t.error<-x.test-Pr ;table4<-cbind(x.test,Pr,t.error) ;colnames(table4)<-c("test data","Predicted test data","Residuals test data")
  train.data<-x.train[(nlag+1):length(x.train)] ; NN.fit<-re.l.tran(NN$fit,GGAL[,1]) ; NN.res<-train.data-NN.fit
  table3<-cbind(train.data,NN.fit,NN.res) ; colnames(table3)<-c("train data","fitted value","fitted residuals")

  #show result
  print(AA<-c(lag=nlag,Seed=K,hidden=J));print(summary.net);print(table3);print(table4)
  print(test.error<-c(RMSE=rMSE(t.error,ahead),MSE=mSE(t.error,ahead),MAE=mAE(t.error,ahead),MAPE=mAPE(t.error,x.test,ahead)))
  windows();if(ahead>=20) {
    graph2(predict=Pr,real=x.test,type="l",main2="ANN")
  } else  {
    graph(predict=Pr,real=x.test,type="l",main2="ANN")
  }
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    # predict future data one-step ahead
    Pr.outsam.ann<-c()
    out.sam<-1  # one-step ahead forecasting
    pre<-Pr.raw[ahead]
    j4<-seq(1,nlag,1) ; dd<-T.data.te[ahead,j4]
    for(i in 1:out.sam) {
      dd<-unlist(dd) ;dd<-c(pre,dd)[1:(nlag+1)]
      names(dd)<-rep(paste0("V",1:(nlag+1))) ;dd<-as.data.frame(dd)
      pre<-as.numeric(predict(NN,t(dd)))
      Pr.outsam.ann<-c(Pr.outsam.ann,pre)
    }
    # Pr.outsam
    Pr.outsam.ann<-re.l.tran(Pr.outsam.ann,observation); Pr.outsam.ann<-ifelse(Pr.outsam.ann<0,0,Pr.outsam.ann)
    print(pred.fut.ann<-c(predict.future=Pr.outsam.ann))

    # นำไปใช้ต่อ
    assign("nlag",nlag,envir=.GlobalEnv);assign("nn.fit",c(NN.fit,Pr),envir=.GlobalEnv)
    assign("nn.fit",c(NN.fit,Pr),envir=.GlobalEnv)
    assign("MSE.ANN",MSE,envir=.GlobalEnv)
    assign("pred.future.ann",pred.fut.ann,envir=.GlobalEnv)
    assign("test.result.ann",test.error,envir=.GlobalEnv)
    }
   }
  } else {
   tkmessageBox(message="Please split your datasets before modeling ANN",icon="error",title="Error")
   tkfocus(tt0)
}
}

# 3.3 tab: SVM
tkadd(model, "command" , label = "SVM" , command =function() SVM(tt))
  SVM<-function(tt0){
  if(exists("train.observation")) {
    tt <- tktoplevel()  #เปิดหน้าต่างใหม่
    tkwm.title(tt, "SVM")  #ตั้งหัวชื่อหน้าต่าง

    #เนื้อหาข้างในหน้าต่าง
    ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
    tkgrid(tklabel(ttframe1,text="\n                Input SVM Parameter                "))

    ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
    lag.value<-tclVar("1,2,3,4,12") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
    lagentry<-tkentry(ttframe2,textvariable=lag.value,width=15)
    tkgrid(tklabel(ttframe2,text="Input number of lag :"),lagentry,tklabel(ttframe2,text="    "))
    tkgrid(tklabel(ttframe2,text="    *Warning: (Lag should be positive integer between 1 and 15)   \n"))

    ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
    C.value<-tclVar("1,10,50,100,500,1000,3000")
    Centry<-tkentry(ttframe3,textvariable=C.value,width=20)
    tkgrid(tklabel(ttframe3,text="Input value of C :"),Centry,tklabel(ttframe3,text="    "))

    ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
    epsilon.value<-tclVar("1e-5,1e-4,1e-3,1e-2")
    epsilonentry<-tkentry(ttframe4,textvariable=epsilon.value,width=20)
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
tkgrid(tklabel(ttframe4,text="Input value of epsilon :"),epsilonentry,tklabel(ttframe4,text="    "))

ttframe5<-tkframe(tt, borderwidth=0, relief="groove")
sigma.value<-tclVar("1e-3,1e-2,1e-1,1,10,50,100")
sigmaentry<-tkentry(ttframe5,textvariable=sigma.value,width=20)
tkgrid(tklabel(ttframe5,text="Input value of sigma :"),sigmaentry,tklabel(ttframe5,text="    "))

ttframe6<-tkframe(tt, borderwidth=0, relief="groove")
time.value<-tclVar("20")
timeentry<-tkentry(ttframe6,textvariable=time.value,width=8)
tkgrid(tklabel(ttframe6,text="Limited time in computation (minute):"),timeentry,tklabel(ttframe6,text="    "))

ttframe7<-tkframe(tt, borderwidth=0, relief="groove")
process.but <-tkbutton(ttframe7,text="   OK   ",command=function() onProcess(train.observation,test.observation))
tkgrid(process.but)

tkpack(ttframe1,ttframe2,ttframe3,ttframe4,ttframe5,ttframe6,ttframe7)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม
tkfocus(tt)

onProcess<-function(x.train,x.test){
  lag.value<-input.entry(as.character(tclvalue(lag.value)),T)
  C.value<-input.entry(as.character(tclvalue(C.value)),F)
  epsilon.value<-input.entry(as.character(tclvalue(epsilon.value)),F)
  sigma.value<-input.entry(as.character(tclvalue(sigma.value)),F)
  time.value<-as.numeric(tclvalue(time.value))
  if(is.na(lag.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input number of lag",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(lag.value)) {
    tkmessageBox(message=lag.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(C.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input value of C",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(C.value)) {
    tkmessageBox(message=C.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(epsilon.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input value of epsilon",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(epsilon.value)) {
    tkmessageBox(message=epsilon.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(sigma.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input value of sigma",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(sigma.value)) {
    tkmessageBox(message=sigma.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(time.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum executing time: default=20 minutes",icon="error",title="Error")
    tkfocus(tt)
  } else if(time.value<0) {
    tkmessageBox(message="Maximum executing time must be positive number (Minutes)",icon="error",title="Error")
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
    tkfocus(tt)
  } else {
    tkdestroy(tt)   # ลบหน้าจอหลังจากผ่านเงื่อนไข พร้อมรัน ARIMA
    tkfocus(tt0)

    best.mse<-1e15 ;totalprogress<-length(lag.value)*length(sigma.value)*length(C.value)*length(epsilon.value)
    pb <- winProgressBar(title = "progress bar", min = 0, max =totalprogress , width = 300);
    ratio.prog<-0;
    setWinProgressBar(pb, ratio.prog, title=paste("SVM progress:",round(ratio.prog/totalprogress*100),"% done"))
    init.t<-Sys.time()

    for(z in lag.value) {
      for(k in epsilon.value) for(j in sigma.value) for(x in C.value) {
        diff.t0<-as.numeric(Sys.time()-init.t)
        t.data<-as.data.frame(na.omit(lagmatrix(observation,z)))
        t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
        svmm<-ksvm(V1~.,data=t.data.tr,type="eps-svr",epsilon=k,kpar=list(sigma=j),cross=0,tol=0.0001,C=x,paralell=T)
        pr<-predict(svmm,t.data.te)
        MSE<-mSE((t.data.te[,1]-pr),length(t.data.te[,1]))
        if(MSE<best.mse){
          best.mse<-MSE;Svmm<-svmm; Pr<-pr; T.data.tr<-t.data.tr;T.data.te<-t.data.te; Epsilon<-k;SIGma<-j;Cval<-x;slag<-z
        }
        # show time tube
        ratio.prog<-ratio.prog+1
        setWinProgressBar(pb, ratio.prog, title=paste( "SVM progress:",round(ratio.prog/totalprogress*100),"% done"))

        # count time
        diff.t<-as.numeric(Sys.time()-init.t)
        if(diff.t<diff.t0 & diff.t>time.value) {
          close(pb)
          stop(paste("Exceed executing time input",time.value,"minutes"))
        }
      }
    }
    close(pb)
    (out.tune<-c(Best.mse=best.mse,Lag=slag,Sigmas=SIGma,Epsilons=Epsilon,C=Cval))

    zx<-out.tune[2]
    t.data<-as.data.frame(na.omit(lagmatrix(observation,zx)))
    t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
    best.svmm<-ksvm(V1~.,data=t.data.tr,epsilon=out.tune[4],kpar=list(sigma=out.tune[3]),cross=0,tol=0.0001,C=out.tune[5])
    predd<-predict(best.svmm,t.data.te)
    pred<-predict(best.svmm,t.data)

    #ค่าพรีดิกและ error
    options(digits=10)
    tab.svm<-cbind(x.test,predd,x.test-predd);colnames(tab.svm)<-c("test data","predicted svm","Residual test data")

    t.error<-observation[(slag+1):length(train.observation)]-pred ;table1<-
cbind(observation[(slag+1):length(train.observation)],pred,t.error) ;colnames(table1)<-c("Data","Predicted value","Residuals")
    test.error<-c(RMSE=rMSE((x.test-predd),ahead),MSE=mSE((x.test-predd),ahead),MAE=mAE((x.test-predd),ahead),MAPE=mAPE((x.test-
predd),x.test,ahead))
```

```
        # predict future data one-step ahead
        Pr.outsam.svm<-c()
        out.sam<-1  # one-step ahead forecasting
        pre<-predd[ahead]
        j4<-seq(1,slag,1) ; dd<-as.numeric(T.data.te[ahead,j4])
        for(i in 1:out.sam) {
          dd<-c(pre,dd)[1:(slag+1)]
          dd3<-matrix(dd,nrow=1,ncol=(slag+1)) ;tdd<-as.data.frame(dd3)
          pre<-as.numeric(predict(best.svmm,tdd))
          Pr.outsam.svm<-c(Pr.outsam.svm,pre)
        }

        #show result
        print(best.svmm);print(out.tune);print(table1);print(test.error)
        windows() ;graph2(predict=predd,real=x.test,type="l",main2="SVM")
        print(pred.fut<-c(predict.future=Pr.outsam.svm))

        # นำไปใช้ต่อ
        assign("slag",slag,envir=.GlobalEnv);assign("svm.fit",pred,envir=.GlobalEnv)
        assign("tr.svm.res",t.error[(slag+1):length(train.observation)],envir=.GlobalEnv)
        assign("te.svm.res",x.test-predd,envir=.GlobalEnv)
        assign("MSE.SVM",MSE,envir=.GlobalEnv)
        assign("pred.future.svm",pred.fut,envir=.GlobalEnv)
        assign("test.result.svm",test.error,envir=.GlobalEnv)
        assign("pred",pred,envir=.GlobalEnv)
      }
    }
  } else {
    tkmessageBox(message="Please split your datasets before modeling SVM",icon="error",title="Error")
    tkfocus(tt0)
    }
}

# 3.4 tab: ARIMAANN
tkadd(model, "command" , label = "ARIMAANN" , command =function() arimaann(tt))
  arimaann<-function(tt0){
  if(exists("fitted.res")) {
  tt <- tktoplevel()  #เปิดหน้าต่างใหม่
  tkwm.title(tt, "ARIMAANN")  #ตั้งหัวชื่อหน้าต่าง

  #เนื้อหาข้างในหน้าต่าง
  ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
  tkgrid(tklabel(ttframe1,text="\n                Input ANN Parameter              "))

  ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
  lag.value<-tclVar("1,2,3,4,5,6,7,8") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
  lagentry<-tkentry(ttframe2,textvariable=lag.value,width=15)
  tkgrid(tklabel(ttframe2,text="Input number of lag :"),lagentry,tklabel(ttframe2,text="     "))
  tkgrid(tklabel(ttframe2,text="    *Warning: (Lag should be positive integer between 1 and 15)   \n"))

  ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
  hidden.value<-tclVar("10")
  hiddenentry<-tkentry(ttframe3,textvariable=hidden.value,width=8)
  tkgrid(tklabel(ttframe3,text="Input number of maximum hidden neurons :"),hiddenentry,tklabel(ttframe3,text="     "))
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
tkgrid(tklabel(ttframe3,text="    *Warning: (Maximum hidden neurons should be positive integer not exceed 20)   \n"))

ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
seed.value<-tclVar("1111,2332,5,337")
seedentry<-tkentry(ttframe4,textvariable=seed.value,width=20)
tkgrid(tklabel(ttframe4,text="Input random number generator :"),seedentry,tklabel(ttframe4,text="    "))
tkgrid(tklabel(ttframe4,text="    *Warning: (Seed number should be positive integer between 1 and 1e+09)   \n"))

ttframe5<-tkframe(tt, borderwidth=0, relief="groove")
time.value<-tclVar("20")
timeentry<-tkentry(ttframe5,textvariable=time.value,width=8)
tkgrid(tklabel(ttframe5,text="Limited time in computation (minute):"),timeentry,tklabel(ttframe5,text="    "))
tkgrid(tklabel(ttframe5,text="    *Warning: (Time must be positive in minute unit)   \n"))

ttframe6<-tkframe(tt, borderwidth=0, relief="groove")
process.but <-tkbutton(ttframe6,text="    OK    ",command=function() onProcess(fitted.tr.res,fitted.te.res))
tkgrid(process.but)

tkpack(ttframe1,ttframe4,ttframe2,ttframe3,ttframe5,ttframe6)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม tkpack
tkfocus(tt)

onProcess<-function(x.train,x.test){
  lag.value<-input.entry(as.character(tclvalue(lag.value)),T)
  hidden.value<-as.numeric(tclvalue(hidden.value))
  seed.value<-input.entry(as.character(tclvalue(seed.value)),F)
  time.value<-as.numeric(tclvalue(time.value))
  if(is.na(lag.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input number of lag",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(lag.value)) {
    tkmessageBox(message=lag.value,icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(hidden.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum number of hidden neurons",icon="error",title="Error")
    tkfocus(tt)
  } else if(!is.element(hidden.value,1:20)) {
    tkmessageBox(message="Hidden neurons must be positive integer between 1 and 20",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(seed.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input seed value",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.character(seed.value)) {
    tkmessageBox(message="supplied seed is not valid ",icon="error",title="Error")
    tkfocus(tt)
  } else if(is.na(time.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
    tkmessageBox(message="Please input maximum time executing for ANN: default=20 minutes",icon="error",title="Error")
    tkfocus(tt)
  } else if(time.value<0) {
    tkmessageBox(message="Maximum executing time must be positive number (Minutes)",icon="error",title="Error")
    tkfocus(tt)
  } else {
    tkdestroy(tt)
    tkfocus(tt0)
```

```
    # ลูปฟิตตัวแบบ
    totalprogress<-length(lag.value)*hidden.value*length(seed.value)
    pb <- winProgressBar(title = "progress bar", min = 0, max =totalprogress , width = 300);
    ratio.prog<-0;
    setWinProgressBar(pb, ratio.prog, title=paste( "ANN progress:",round(ratio.prog/totalprogress*100),"% done"))
    init.t<-Sys.time()

    best.mse<-1e15 ;Lagg<-cbind(lagmatrix(observation,max(lag.value))) # นำค่าเมตริกซ์ lag ค่าจริง+ค่าเมตริกซ์ lag ค่า res
    Lagg<-cbind(lagmatrix(fitted.res,max(lag.value))) # นำค่าเมตริกซ์ lag ค่าจริง+ค่าเมตริกซ์ lag ค่า res

    best.mse<-1e15 ;  for(i in lag.value) for(j in hidden.value){
      for(k in seed.value){
        diff.t0<-as.numeric(Sys.time()-init.t)
        ggal<-Lagg[,c(1:(i+1))]
        t.data<-as.data.frame(apply(na.omit(ggal),2,l.tran))
        t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
        set.seed(1111+k)
        net <- nnet(V1~.,data=t.data.tr, size=j,maxit=2e5,abstol = 1.0e-9,reltol=1.0e-9,trace=F,linout=T)
        pr<-re.l.tran(predict(net,t.data.te),ggal[,1]) ;pr.raw<-predict(net,t.data.te)
        MSE<-mSE((x.test-pr),ahead)
        if(MSE<best.mse){
          best.mse<-MSE ;NN<-net ;summary.net<-summary(net) ;Pr<-pr ;Pr.raw<-pr.raw;K<-k  ;nlag<-i ;J<-j ;GGAL<-ggal ;T.data.te<-t.data.te
        }

        # show time tube
        ratio.prog<-ratio.prog+1
        setWinProgressBar(pb, ratio.prog, title=paste( "ANN progress:",round(ratio.prog/totalprogress*100),"% done"))

        # count time
        diff.t<-as.numeric(Sys.time()-init.t)
        if(diff.t<diff.t0 & diff.t>time.value) {
          close(pb)
          stop(paste("Exceed executing time input",time.value,"minutes"))
      }
    }
  }
    close(pb)

    # Calculate total forecast
    NN.fit<-re.l.tran(NN$fit,GGAL[,1]) #ค่า fitted value ของ nn ภายหลังการแปลง
    fitted.nn<-c(NN.fit,Pr)
    t.pre<-fitted.nn+fitted.val[(nlag+1):length(fitted.val)]; t.pre2<-t.pre[-((nlag+1):(length(observation)-ahead))]
    t.res<-observation[(nlag+1):length(observation)]-t.pre

  t.error<-x.test-Pr   #ค่า error เทสเซต
  test.error<-c(RMSE=rMSE(t.error,ahead),MSE=mSE(t.error,ahead),MAE=mAE(t.error,ahead),MAPE=mAPE(test.observation-
t.pre2,test.observation,ahead))
    table<-cbind(observation[(nlag+1):length(observation)],t.pre,t.res) ; colnames(table)<-c("Data","ARIMAANN fitted value","ARIMAANN
residuals")

    # predict future data one-step ahead
    Pr.outsam.arimaann<-c()
    out.sam<-1  # one-step ahead forecasting
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    pre<-Pr.raw[ahead]
    j4<-seq(1,nlag,1) ; dd<-T.data.te[ahead,j4]
    for(i in 1:out.sam) {
      dd<-unlist(dd) ;dd<-c(pre,dd)[1:(nlag+1)]
      names(dd)<-rep(paste0("V",1:(nlag+1))) ;dd<-as.data.frame(dd)
      pre<-as.numeric(predict(NN,t(dd)))
      Pr.outsam.arimaann<-c(Pr.outsam.arimaann,pre)
    }
    Pr.outsam.arimaann<-re.l.tran(Pr.outsam.arimaann,fitted.res) ; ARIMAANN.pre<-pred.future.arima+Pr.outsam.arimaann

    #show result
    print(AA<-c(lag=nlag,Seed=K,hidden=J));print(summary.net);print(table);print(test.error)
      # predict one-step ahead
      print(pred.fut<-ARIMAANN.pre)
    windows();if(ahead>=20) {
      graph2(predict=t.pre2,real=test.observation,type="l",main2="Hybrid ARIMAANN")
    } else  {
      graph(predict=t.pre2,real=test.observation,type="l",main2="Hybrid ARIMAANN")
    }

  # เก็บค่าไปใช้ต่อ
  assign("test.result.arimaann",test.error,envir=.GlobalEnv)
  assign("pred.future.arimaann",ARIMAANN.pre,envir=.GlobalEnv)
  assign("Pr.outsam.arimaann",Pr.outsam.arimaann,envir=.GlobalEnv)
  }
  }
 } else {
  tkmessageBox(message="You must run ARIMA before run ARIMAANN",icon="error",title="Error")
  tkfocus(tt0)
 }
}

# 3.5 tab: ARIMASVM
tkadd(model, "command" , label = "ARIMASVM" , command =function() arimasvm(tt))
arimasvm<-function(tt0){
 if(exists("fitted.res")) {
  tt <- tktoplevel()
  tkwm.title(tt, "SVM")

  #เนื้อหาข้างในหน้าต่าง
  ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
  tkgrid(tklabel(ttframe1,text="\n                Input SVM Parameter              "))

  ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
  lag.value<-tclVar("1,2,3,4,12") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
  lagentry<-tkentry(ttframe2,textvariable=lag.value,width=15)
  tkgrid(tklabel(ttframe2,text="Input number of lag :"),lagentry,tklabel(ttframe2,text="    "))
  tkgrid(tklabel(ttframe2,text="    *Warning: (Lag should be positive integer between 1 and 15)   \n"))

  ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
  C.value<-tclVar("1,10,50,100,500,1000,3000")
  Centry<-tkentry(ttframe3,textvariable=C.value,width=20)
  tkgrid(tklabel(ttframe3,text="Input value of C :"),Centry,tklabel(ttframe3,text="    "))
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
ttframe4<-tkframe(tt, borderwidth=0, relief="groove")
epsilon.value<-tclVar("1e-5,1e-4,1e-3,1e-2")
epsilonentry<-tkentry(ttframe4,textvariable=epsilon.value,width=20)
tkgrid(tklabel(ttframe4,text="Input value of epsilon :"),epsilonentry,tklabel(ttframe4,text="    "))

ttframe5<-tkframe(tt, borderwidth=0, relief="groove")
sigma.value<-tclVar("1e-3,1e-2,1e-1,1,10,50,100")
sigmaentry<-tkentry(ttframe5,textvariable=sigma.value,width=20)
tkgrid(tklabel(ttframe5,text="Input value of sigma :"),sigmaentry,tklabel(ttframe5,text="    "))

ttframe6<-tkframe(tt, borderwidth=0, relief="groove")
time.value<-tclVar("20")
timeentry<-tkentry(ttframe6,textvariable=time.value,width=8)
tkgrid(tklabel(ttframe6,text="Limited time in computation (minute):"),timeentry,tklabel(ttframe6,text="    "))

ttframe7<-tkframe(tt, borderwidth=0, relief="groove")
process.but <-tkbutton(ttframe7,text="   OK   ",command=function() onProcess(fitted.tr.res,fitted.te.res))
tkgrid(process.but)

tkpack(ttframe1,ttframe2,ttframe3,ttframe4,ttframe5,ttframe6,ttframe7)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม
tkfocus(tt)

onProcess<-function(x.train,x.test){
lag.value<-input.entry(as.character(tclvalue(lag.value)),T)
C.value<-input.entry(as.character(tclvalue(C.value)),F)
epsilon.value<-input.entry(as.character(tclvalue(epsilon.value)),F)
sigma.value<-input.entry(as.character(tclvalue(sigma.value)),F)
time.value<-as.numeric(tclvalue(time.value))
if(is.na(lag.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
  tkmessageBox(message="Please input number of lag",icon="error",title="Error")
  tkfocus(tt)
} else if(is.character(lag.value)) {
  tkmessageBox(message="Lag must be positive integer between 1 and 15",icon="error",title="Error")
  tkfocus(tt)
} else if(is.na(C.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
  tkmessageBox(message="Please input value of C",icon="error",title="Error")
  tkfocus(tt)
} else if(is.character(C.value)) {
  tkmessageBox(message=C.value,icon="error",title="Error")
  tkfocus(tt)
} else if(is.na(epsilon.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
  tkmessageBox(message="Please input value of epsilon",icon="error",title="Error")
  tkfocus(tt)
} else if(is.character(epsilon.value)) {
  tkmessageBox(message=epsilon.value,icon="error",title="Error")
  tkfocus(tt)
} else if(is.na(sigma.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
  tkmessageBox(message="Please input value of sigma",icon="error",title="Error")
  tkfocus(tt)
} else if(is.na(sigma.value)) {
  tkmessageBox(message=sigma.value,icon="error",title="Error")
  tkfocus(tt)
} else if(is.na(time.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
  tkmessageBox(message="Please input maximum executing time: default=20 minutes",icon="error",title="Error")
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    tkfocus(tt)
  } else if(time.value<0) {
    tkmessageBox(message="Maximum executing time must be positive number(Minutes)",icon="error",title="Error")
    tkfocus(tt)
  } else {
    tkdestroy(tt)   # ลบหน้าจอหลังจากผ่านเงื่อนไข พร้อมรัน ARIMA
    tkfocus(tt0)

    best.mse<-1e15 ;totalprogress<-length(lag.value)*length(sigma.value)*length(C.value)*length(epsilon.value)
    pb <- winProgressBar(title = "progress bar", min = 0, max =totalprogress , width = 300);
    ratio.prog<-0;
    setWinProgressBar(pb, ratio.prog, title=paste( "SVM progress:",round(ratio.prog/totalprogress*100),"% done"))
    init.t<-Sys.time()

    for(z in lag.value) {
      for(k in epsilon.value) for(j in sigma.value) for(x in C.value) {
        diff.t0<-as.numeric(Sys.time()-init.t)
        t.data<-as.data.frame(na.omit(lagmatrix(fitted.res,z)))
        t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
        svmm<-ksvm(V1~.,data=t.data.tr,type="eps-svr",epsilon=k,kpar=list(sigma=j),cross=0,tol=0.0001,C=x,paralell=T)
        pr<-predict(svmm,t.data.te)
        MSE<-mSE((t.data.te[,1]-pr),length(t.data.te[,1]))
        if(MSE<best.mse){
          best.mse<-MSE;Svmm<-svmm; Pr<-pr; T.data.tr<-t.data.tr;T.data.te<-t.data.te; Epsilon<-k;SIGma<-j;Cval<-x;slag<-z
        }
        # show time tube
        ratio.prog<-ratio.prog+1
        setWinProgressBar(pb, ratio.prog, title=paste( "SVM progress:",round(ratio.prog/totalprogress*100),"% done"))
        # count time
        diff.t<-as.numeric(Sys.time()-init.t)
        if(diff.t<diff.t0 & diff.t>time.value) {
          close(pb)
          stop(paste("Exceed executing time input",time.value,"minutes"))
           }
          }
        }
    close(pb)
    (out.tune<-c(Best.mse=best.mse,Lag=slag,Sigmas=SIGma,Epsilons=Epsilon,C=Cval))

    zx<-out.tune[2]
    t.data<-as.data.frame(na.omit(lagmatrix(fitted.res,zx)))
    t.data.tr<-t.data[1:(nrow(t.data)-ahead),]; t.data.te<-t.data[-(1:(nrow(t.data)-ahead)),]
    best.svmm<-ksvm(V1~.,data=t.data.tr,epsilon=out.tune[4],kpar=list(sigma=out.tune[3]),cross=0,tol=0.0001,C=out.tune[5])
    predd<-predict(best.svmm,t.data.te)
    pred<-predict(best.svmm,t.data)

    #ค่าพรีดิกและ error
    options(digits=10)
    tab.svm<-cbind(x.test,predd,x.test-predd);colnames(tab.svm)<-c("test data","predicted svm","Residual test data")
    # Calculate total forecast
    t.pre<-pred+fitted.val[(slag+1):length(fitted.val)]; t.pre2<-t.pre[-((slag+1):(length(observation)-ahead))]
    t.res<-observation[(slag+1):length(observation)]-t.pre
    table<-cbind(observation[(slag+1):length(observation)],t.pre,t.res) ; colnames(table)<-c("Data","ARIMASVM fitted value","ARIMASVM
residuals")
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
      arimasvm.obs<-observation[(slag+1):length(observation)] ;table1<-cbind(arimasvm.obs,t.pre,(arimasvm.obs-t.pre)) ;colnames(table1)<-
c("Data","Predicted value","Residuals")
      test.error<-c(RMSE=rMSE((x.test-predd),ahead),MSE=mSE((x.test-predd),ahead),MAE=mAE((x.test-
predd),ahead),MAPE=mAPE((test.observation-t.pre2),test.observation,ahead))
    # predict future data one-step ahead
    Pr.outsam.arimasvm<-c()
    out.sam<-1  # one-step ahead forecasting
    pre<-predd[ahead]
    j4<-seq(1,slag,1) ; dd<-as.numeric(T.data.te[ahead,j4])
    for(i in 1:out.sam) {
      dd<-c(pre,dd)[1:(slag+1)]
      dd3<-matrix(dd,nrow=1,ncol=(slag+1)) ;tdd<-as.data.frame(dd3)
      pre<-as.numeric(predict(best.svmm,tdd))
      Pr.outsam.arimasvm<-c(Pr.outsam.arimasvm,pre)
    }
    #Find one-step ahead forecasting
    ARIMASVM.pre<-pred.future.arima+Pr.outsam.arimasvm

    #show result
    print(best.svmm);print(out.tune);print(table);print(test.error)
    windows() ;graph2(predict=t.pre2,real=test.observation,type="l",main2="ARIMASVM")
    print(pred.fut<-ARIMASVM.pre)

    #เก็บค่านำไปใช้
    assign("MSE.SVM",mSE((x.test-predd),ahead),envir=.GlobalEnv)
    assign("pred.future.arimasvm",ARIMASVM.pre,envir=.GlobalEnv)
    assign("test.result.arimasvm",test.error,envir=.GlobalEnv)
    assign("x.test",x.test,envir=.GlobalEnv)
    assign("predd",predd,envir=.GlobalEnv)
     }
    }
} else {
      tkmessageBox(message="Please split your datasets before modeling SVM",icon="error",title="Error")
      tkfocus(tt0)
  }
}
# 3.6 tab: Combined
tkadd(model, "command" , label = "combined ARIMA+ANN+SVM" , command =function() conbined(tt))
conbined<-function(tt0){
  if(exists("fitted.val") & exists("nn.fit") & exists("svm.fit")){
  tt <- tktoplevel()  #เปิดหน้าต่างใหม่
  tkwm.title(tt, "COMBINED")  #ตั้งหัวชื่อหน้าต่าง
  ttframe1<-tkframe(tt, borderwidth=0, relief="groove")
  digit.value<-tclVar("3") #แสดงเป็นตัวเก็บค่าแต่ให้เริ่มเป็นค่าว่างไว้ก่อน
  digitentry<-tkentry(ttframe1,textvariable=digit.value,width=8)
  tkgrid(tklabel(ttframe1,text="Input number of digits :"),digitentry)
  ttframe2<-tkframe(tt, borderwidth=0, relief="groove")
  time.value<-tclVar("10")
  timeentry<-tkentry(ttframe2,textvariable=time.value,width=8)
  tkgrid(tklabel(ttframe2,text="Limited time in computation (minute) :"),timeentry)
  ttframe3<-tkframe(tt, borderwidth=0, relief="groove")
  process.but <-tkbutton(ttframe3,text="    OK    ",command=function() onProcess(train.observation,test.observation))
  tkgrid(process.but)
  tkpack(ttframe1,ttframe2,ttframe3)  #เอาส่วนย่อยต่างๆ มาเรียงกัน การทำงานเรียงตาม
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
  tkfocus(tt)
  onProcess<-function(x.train,x.test){
    time.value<-as.numeric(tclvalue(time.value))
    digit.value<-as.numeric(tclvalue(digit.value))
    if(is.na(digit.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
      tkmessageBox(message="Please input number of lag",icon="error",title="Error")
      tkfocus(tt)
    } else if(digit.value<1) {
      tkmessageBox(message="Digit number should not lower than 1 because of poor accuracy",icon="error",title="Error")
      tkfocus(tt)
    } else if(digit.value>6) {
      tkmessageBox(message="Digit number should not greater than 6 because of long time computation and not guarantee better
accuracy",icon="error",title="Error")
      tkfocus(tt)
    } else if(is.na(time.value)){ # ถ้าค่าที่ใส่ว่าง คือไม่ได้ใส่ จะแสดงข้อความเตือนให้ใส่
      tkmessageBox(message="Please input maximum executing time: default=10 minutes",icon="error",title="Error")
      tkfocus(tt)
    } else if(time.value<0) {
      tkmessageBox(message="Maximum executing time must be positive number(Minutes)",icon="error",title="Error")
      tkfocus(tt)
    } else {
      tkdestroy(tt)   # ลบหน้าจอหลังจากผ่านเงื่อนไข พร้อมรัน ARIMA
      tkfocus(tt0)
    options(digits=15)
    StartDivide<-length(x.train)  ; best.mse<-max(MSE.ARIMA,MSE.ANN,MSE.SVM)
    Maxlag=max(nlag,slag)
    length.test<-length(x.test)
    arima.pre<-fitted.val[-(1:(length(fitted.val)-ahead))]
    ann.pre<-nn.fit[(StartDivide-nlag+1):length(nn.fit)]
    svm.pre<-pred[(StartDivide-slag+1):length(pred)]
    #Combine
    best.mae<-1e15 ;digit.value<-10^(-1*(digit.value))
    k<-seq(0,1,digit.value) # search 0 ถึง 1 จำนวน 1000 ค่า
    for(i in k){
      ar.part<-i*arima.pre
      for(j in seq(0,(max(k)-i),digit.value)) {
        ann.part<-j*ann.pre
        svm.part<-(max(k)-i-j)*svm.pre
        combine.pre<-ar.part+ann.part+svm.part
        res.combine.te<-x.test-combine.pre
        MAE<-mAE(res.combine.te,length.test) ;MSE.C<-mSE(res.combine.te,length.test)
        if(MAE<best.mae & MSE.C<=best.mse){
          best.mae<-MAE
          ar.k<-i ;ann.k<-j ;svm.k<-(max(k)-i-j)
        }
      }
    }
    # best mae and all weights
    combine.info<-c(Best.mae=best.mae,Weight.arima=ar.k,Weight.ann=ann.k,Weight.svm=svm.k)

# combined predicted
combine.pre<-ar.k*arima.pre+svm.k*svm.pre+ann.k*ann.pre
res.combine.te<-x.test-combine.pre
MSE<-mSE(res.combine.te,length.test)
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
RMSE<-sqrt(MSE)
MAE<-mAE(res.combine.te,length.test)
MAPE<-mAPE(res.combine.te,x.test,length.test)
options(digits=10)
table4<-cbind(x.test,combine.pre,res.combine.te)
colnames(table4)<-c("test data","combine predicted","residual combine")
test.error<-c(RMSE=RMSE,MSE=MSE,MAE=MAE,MAPE=MAPE)

# show result
print(combine.info) ;print(table4) ;print(test.error)
windows() ;graph2(predict=combine.pre,real=observation[-((Maxlag+1):(length(observation)-ahead))],type="l",main2="Combined
ARIMA+ANN+SVM")
# predict one step ahead
print(combined.pre<-ar.k*pred.future.arima+ann.k*pred.future.ann+svm.k*pred.future.svm)
#เก็บค่านำไปใช้ต่อ
assign("test.result.combined",test.error,envir=.GlobalEnv)
assign("pred.future.combined",combined.pre,envir=.GlobalEnv)
    }
  }
} else {
  tkmessageBox(message="You must run ARIMA,ANN and SVM before run combined model",icon="error",title="Error")
  tkfocus(tt0)
}
}
tkadd(topMenu, "cascade" , label = "Modeling", menu = model)

#4. tab: Compare result
tkadd(topMenu, "command" , label = "Compare results" , command =function() compare(tt))
    compare<-function(tt0){
    tt <- tktoplevel()  #เปิดหน้าต่างใหม่
    tkwm.title(tt, "Compare results")  #ตั้งหัวชื่อหน้าต่าง
  if(!exists("test.result.arima")) {
    tkmessageBox(message="Please run arima before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else if(!exists("test.result.ann")) {
    tkmessageBox(message="Please run ann before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else if(!exists("test.result.svm")) {
    tkmessageBox(message="Please run svm before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else if(!exists("test.result.arimaann")) {
    tkmessageBox(message="Please run arimaann before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else if(!exists("test.result.arimasvm")) {
    tkmessageBox(message="Please run arimasvm before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else if(!exists("test.result.combined")) {
    tkmessageBox(message="Please run combined model before compare all results",icon="error",title="Error")
    tkfocus(tt)
  } else {
    tkdestroy(tt)   # ลบหน้าจอใหม่ที่เด้งขึ้นมา
    tkfocus(tt0)    # คงหน้าจอเดิมตอน
    compare.table<-
rbind(ARIMA=c(test.result.arima,pred.future.arima),ANN=c(test.result.ann,pred.future.ann),SVM=c(test.result.svm,pred.future.svm),ARIMAAN
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
N=c(test.result.arimaann,pred.future.arimaann),ARIMASVM=c(test.result.arimasvm,pred.future.arimasvm),COMBINED=c(test.result.combined
,pred.future.combined))
    print(compare.table)
  }
}

#5. tab: Quit
tkadd(topMenu, "command" , label = "Quit" , command =function() quit.gui(tt))
quit.gui<-function(txt.display){
  tt<-tktoplevel()
  tkwm.title(tt,"Quit")
  tkgrid(tklabel(tt,text="\nAre you sure you want to quit program?\n"),columnspan=2) # "\n ให้แสดงผลเว้นหนึ่งบรเว้นบนล่าง เปิดปิดรทัด
  yes.but <-tkbutton(tt,text="Yes",width=10,command=function(){ # สร้างปุ่มและข้ อความcolumnspan=2 คือ yes noปุ่ม
    tkdestroy(tt)
    tkdestroy(txt.display)
  }
  )
  no.but <-tkbutton(tt,text="No",width=10,command=function(){
    tkdestroy(tt)
    tkfocus(txt.display)
  }
  )
  tkgrid(yes.but,no.but)
  tkfocus(tt)
}
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นายภัทร วรภู เกิดเมื่อวันที่ 28 สิงหาคม พ.ศ.2528 ที่จังหวัดสุราษฎร์ธานี สำเร็จการศึกษาปริญญาตรีหลักสูตรสถิติศาสตร์บัณฑิต คณะพาณิชยศาสตร์และการบัญชี สาขาสถิติคณิตศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยปี พ.ศ.2552 ได้ศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University