



แบบจำลองโครงข่ายประสาทเทียมแบบลึกสำหรับการทำนายปริมาณความหนาแน่น
ของฝุ่นละออง PM2.5 บริเวณพื้นที่จังหวัดกรุงเทพมหานครชั้นใน

A LONG SHORT-TERM MEMORY DEEP NEURAL NETWORK MODEL FOR PM2.5
PREDICTION IN THE URBAN BANGKOK AREA

กานต์กมล ทวีผล

บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ

2562

แบบจำลองโครงข่ายประสาทเทียมแบบลึกสำหรับการทำนายปริมาณความหนาแน่น
ของฝุ่นละออง PM2.5 บริเวณพื้นที่จังหวัดกรุงเทพมหานครชั้นใน



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2562
ลิขสิทธิ์ของมหาวิทยาลัยศรีนครินทรวิโรฒ

A LONG SHORT-TERM MEMORY DEEP NEURAL NETWORK MODEL FOR
PM2.5 PREDICTION IN THE URBAN BANGKOK AREA



A Thesis Submitted in partial Fulfillment of Requirements
for MASTER OF SCIENCE (Information Technology)
Faculty of Science Srinakharinwirot University

2019

Copyright of Srinakharinwirot University

ปริญญานิพนธ์

เรื่อง

แบบจำลองโครงข่ายประสาทเทียมแบบลึกสำหรับการทำนายปริมาณความหนาแน่น
ของฝุ่นละออง PM2.5 บริเวณพื้นที่จังหวัดกรุงเทพมหานครชั้นใน

ของ

กานต์กมล ทวีผล

ได้รับอนุมัติจากบัณฑิตวิทยาลัยให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ของมหาวิทยาลัยศรีนครินทรวิโรฒ

คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ นายแพทย์ฉัตรชัย เอกปัญญาสกุล)

คณะกรรมการสอบปากเปล่าปริญญานิพนธ์

ที่ปรึกษาหลัก

ประธาน

(ผู้ช่วยศาสตราจารย์ ดร.นุรีย์ วิวัฒน์วัฒนา)

(ดร.สุทธิพงศ์ ธีชัยพงษ์)

กรรมการ

(อาจารย์ ดร.วีรยุทธ เจริญเรืองกิจ)

ชื่อเรื่อง	แบบจำลองโครงข่ายประสาทเทียมแบบลึกสำหรับการทำนายปริมาณ ความหนาแน่น ของฝุ่นละออง PM2.5 บริเวณพื้นที่จังหวัดกรุงเทพมหานครชั้นใน
ผู้วิจัย	กานต์กมล ทวีผล
ปริญญา	วิทยาศาสตร์มหาบัณฑิต
ปีการศึกษา	2562
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. นุรีย์ วิวัฒน์วัฒนา

การทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ได้อย่างแม่นยำนั้นช่วยให้สามารถจัดการกับปัญหามลพิษทางอากาศได้อย่างมีประสิทธิภาพ อีกทั้งยังสามารถช่วยแจ้งเตือนประชาชนและสื่อต่าง ๆ ที่มีความเกี่ยวข้องกับปัญหาดังกล่าวได้อย่างทันท่วงที ในการวิจัยนี้ ได้ใช้แบบจำลองโครงข่ายประสาทเทียมเชิงลึกแบบ Long Short-Term Memory (LSTM) และแบบจำลองอนุกรมเวลา Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) โดยใช้ข้อมูลฝุ่นละออง ข้อมูลสารก่อมลพิษทางอากาศ ข้อมูลอุตุนิยมวิทยาและสภาพอากาศในปี.ศ.2017 และ 2018 จากสถานีตำรวจนครบาลโชคชัย ซึ่งถูกเลือกเป็นตัวแทนข้อมูลของพื้นที่จังหวัดกรุงเทพมหานครชั้นใน งานวิจัยมุ่งหวังในการแสดงผลสมรรถนะของแบบจำลอง LSTM เปรียบเทียบกับแบบจำลอง SARIMAX ในการทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า และจากการทดลองพบว่าแบบจำลอง LSTM นั้นให้ค่า RMSE และ MAE ของแต่ละช่วงเวลาในการทำนายออกมาดีกว่าแบบจำลอง SARIMAX ซึ่งการทำนายในอีก 1 ชั่วโมงข้างหน้าแบบจำลอง LSTM ได้ค่าเฉลี่ยของ RMSE = 3.11 ไมโครกรัมต่อลูกบาศก์เมตร และ MAE = 2.36 ไมโครกรัมต่อลูกบาศก์เมตร ในขณะที่ค่าความผิดพลาด (error) ของแบบจำลอง SARIMAX นั้นมีค่าสูงกว่าเป็นเท่าตัว จากการทดลองจะสังเกตได้ว่ายิ่งจำนวนชั่วโมงในการทำนายเพิ่มมากขึ้น ค่าความผิดพลาดที่ได้จากการทำนายของทั้งสองแบบจำลองก็จะยิ่งสูงขึ้นตาม

คำสำคัญ : การทำนายฝุ่นละออง PM2.5, โครงข่ายประสาทเทียมแบบลึก, เทคนิคการเรียนรู้ของเครื่อง

Title	A LONG SHORT-TERM MEMORY DEEP NEURAL NETWORK MODEL FOR PM2.5 PREDICTION IN THE URBAN BANGKOK AREA
Author	KANKAMON THAWEEPHOL
Degree	MASTER OF SCIENCE
Academic Year	2019
Thesis Advisor	Assistant Professor Dr. Nuwee Wiwatwattana

The accurately forecasting of fine particulate matter of a less than 2.5 micrometer diameter (PM2.5) concentration levels is important to better manage the air pollution situation and to give advance warnings to residents and officials. In this research, a Long Short-Term Memory (LSTM) deep neural network model and a Seasonal Auto Regressive Integrated Moving Average with eXogenous regressor (SARIMAX) were used for air quality and meteorological time series data at the Chokchai metropolitan police station area in Bangkok from 2017 to 2018. After figuring out the best configuration of both algorithms, the performance of the LSTM model to predict PM2.5 concentrations for twenty-four hours was evaluated and compared against the SARIMAX model. The experiments indicated that LSTM had better prediction accuracy as indicated by the RMSE and MAE values of each of the time steps. LSTM could forecast one hour ahead at a very low RMSE of 3.11 micrograms per cubic meter on average, and a MAE of 2.36 micrograms per cubic meter on average, while SARIMAX errors were more than doubled. When the time steps were farther apart, the number of errors were higher for both models.

Keyword : PM2.5 forecasting, Long short-term memory, Neural network, Deep learning

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยความช่วยเหลือจาก ผศ.ดร.นุรีย์ วิวัฒน์วัฒนา อาจารย์ที่ปรึกษาที่ให้คำปรึกษา คำแนะนำในการทำปริญญานิพนธ์ ตลอดจนสนับสนุนข้อมูลทางวิชาการ และข้อมูลสำหรับทำปริญญานิพนธ์นี้

ขอกราบขอบพระคุณกรมควบคุมมลพิษ ที่ให้ความช่วยเหลือข้อมูลฝุ่นละออง PM2.5 และข้อมูลอุตุนิยมวิทยาที่ถูกจัดเก็บในพื้นที่กรุงเทพมหานครชั้นในมาใช้ในการทำปริญญานิพนธ์นี้

ขอกราบขอบพระคุณคณะกรรมการสอบปริญญานิพนธ์ ที่ได้ให้คำแนะนำและข้อเสนอแนะสำหรับการปรับปรุงปริญญานิพนธ์ และการใช้เทคนิคการเรียนรู้ของเครื่องโดยใช้แบบจำลองโครงข่ายประสาทเทียมเชิงลึกและแบบจำลองอนุกรมเวลาเป็นเครื่องมือในการวิเคราะห์และแก้ไขปัญหาทางข้อมูลต่อไป

ขอกราบขอบพระคุณบัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ สำหรับทุนสนับสนุนการนำเสนอผลงานวิจัยของนิสิตบัณฑิตศึกษา ทำให้ได้รับประสบการณ์นำเสนอปริญญานิพนธ์ รวมถึงได้แลกเปลี่ยนความรู้ทางด้านเทคโนโลยีต่าง ๆ

กานต์กมล ทวีผล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญตาราง.....	ญ
สารบัญภาพประกอบ	ฎ
บทที่ 1 บทนำ.....	1
ภูมิหลัง	1
ความมุ่งหมายของงานวิจัย.....	2
ความสำคัญของการวิจัย	2
ขอบเขตของการวิจัย	2
ประชากรที่ใช้ในการวิจัย.....	2
กลุ่มตัวอย่างที่ใช้ในการวิจัย.....	3
ตัวแปรที่ศึกษา	3
กรอบแนวคิดในงานวิจัย	4
สมมุติฐานในการวิจัย.....	4
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	5
ทฤษฎีเกี่ยวกับ Long Short-Term Memory (LSTM).....	5
Forget gate layer	7
Input gate layer.....	8
Output gate layer.....	9

ทฤษฎีเกี่ยวกับ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX)	10
Integrated (I)	11
Autoregressive model (AR)	11
Moving Average model (MA)	12
Seasonal ARIMA Model with External Regressors	12
ทฤษฎีเกี่ยวกับการประเมินวัดความแม่นยำของแต่ละอัลกอริทึม	13
งานวิจัยที่เกี่ยวข้อง	14
บทที่ 3 วิธีดำเนินการวิจัย.....	20
การกำหนดประชากรและการสุ่มกลุ่มตัวอย่าง.....	20
ประชากร	20
การเลือกกลุ่มตัวอย่าง	20
การสร้างเครื่องมือที่ใช้ในการวิจัย.....	21
การเก็บรวบรวมข้อมูล	22
การจัดกระทำและการวิเคราะห์ข้อมูล.....	23
บทที่ 4 ผลการดำเนินการวิจัย	36
ผลลัพธ์ของการสร้างแบบจำลอง LSTM	36
ผลลัพธ์ของการสร้างแบบจำลอง SARIMAX.....	38
ผลลัพธ์จากการเปรียบเทียบการทำนายความหนาแน่นของฝุ่นละออง PM2.5 ระหว่าง แบบจำลอง LSTM และ SARIMAX	41
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ	48
สรุปผลการวิจัย.....	48
อภิปรายผลการวิจัย	50
ข้อเสนอแนะ	51

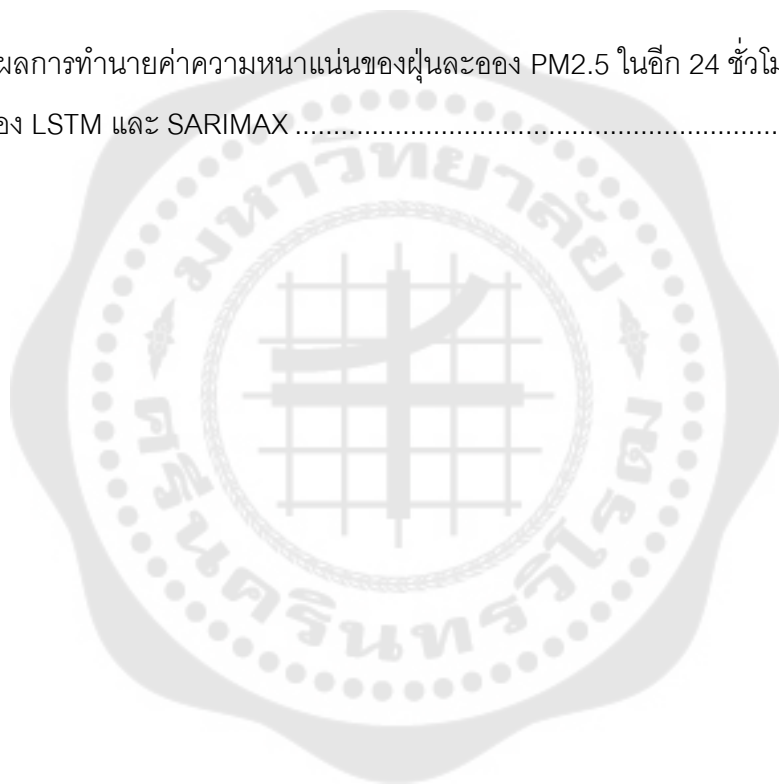
บรรณานุกรม	52
ภาคผนวก.....	55
ประวัติผู้เขียน.....	62



สารบัญตาราง

หน้า

ตาราง 1 แสดงรายละเอียดของแต่ละ Feature	23
ตาราง 2 ผลการทำนายค่าฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าโดยใช้แบบจำลอง LSTM37	
ตาราง 3 ผลการทำนายค่าฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าโดยใช้แบบจำลอง SARIMAX	38
ตาราง 4 ผลการทำนายค่าความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าระหว่างแบบจำลอง LSTM และ SARIMAX	49



สารบัญภาพประกอบ

หน้า

ภาพประกอบ 1 แสดงโครงสร้างการทำงานของ RNN	5
ภาพประกอบ 2 แสดงโครงสร้างการทำงานของ LSTM	6
ภาพประกอบ 3 แสดงโครงสร้าง Forget gate layer	7
ภาพประกอบ 4 แสดงโครงสร้าง Input gate layer	8
ภาพประกอบ 5 แสดงโครงสร้าง Output gate layer.....	9
ภาพประกอบ 6 แสดงตารางสรุปผลงานวิจัยที่ศึกษา (ส่วนที่ถูกไฮไลต์ไว้เป็นแบบจำลองที่ให้ค่าออกมาดีที่สุด)	19
ภาพประกอบ 7 Flowchart กระบวนการทำงานของแบบจำลอง LSTM และ แบบจำลอง SARIMAX	21
ภาพประกอบ 8 แสดงตัวอย่างข้อมูลของ Dataset สถานีตำรวจนครบาลโชคชัย	25
ภาพประกอบ 9 แสดงตัวอย่างข้อมูลของ Dataset สถานีตำรวจนครบาลโชคชัย ในโปรแกรม	26
ภาพประกอบ 10 แสดงค่าฝุ่นละออง PM2.5 ช่วงปี 2017-2018	26
ภาพประกอบ 11 แสดงค่าฝุ่นละออง PM2.5 ของปี 2017	26
ภาพประกอบ 12 แสดงค่าฝุ่นละออง PM2.5 ของปี 2018	27
ภาพประกอบ 13 แสดงรายละเอียดข้อมูลทั้งหมดของแต่ละ feature	27
ภาพประกอบ 14 แสดงความสัมพันธ์ระหว่างข้อมูลฝุ่นละออง PM2.5 กับ feature อื่น ในรูปแบบ Heatmap	28
ภาพประกอบ 15 แสดงค่าความสัมพันธ์ระหว่าง feature อื่นกับฝุ่นละออง PM2.5.....	29
ภาพประกอบ 16 แสดงค่าความสัมพันธ์ระหว่าง feature อื่นกับฝุ่นละออง PM2.5 (ต่อ).....	30
ภาพประกอบ 17 แสดงภาพตัวอย่างการ shift ข้อมูล เพื่อใช้ในการทำนาย.....	32
ภาพประกอบ 18 แสดงภาพตัวอย่างข้อมูลที่ได้จากการใช้ฟังก์ชัน shift ในการขยับข้อมูล	32

ภาพประกอบ 19 แสดงค่า Trend Seasonal และ Residual ของ train data	33
ภาพประกอบ 20 แสดงกราฟ ACF ที่พล็อตค่า lag แบบช่วงเวลาสั้นกับช่วงเวลายาว.....	34
ภาพประกอบ 21 แสดงตัวอย่างค่าการ Tuning Parameter	34
ภาพประกอบ 22 แสดงภาพตัวอย่างข้อมูลที่ได้จากการทำนายโดยขยับทีละหนึ่งชั่วโมง	35
ภาพประกอบ 23 กราฟแสดงค่า RMSE เปรียบเทียบระหว่างแบบจำลอง LSTM กับ SARIMAX	40
ภาพประกอบ 24 กราฟแสดงค่า MAE เปรียบเทียบระหว่างแบบจำลอง LSTM กับ SARIMAX ..	40
ภาพประกอบ 25 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 1 - 2 ชั่วโมง	41
ภาพประกอบ 26 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 3 - 5 ชั่วโมง	42
ภาพประกอบ 27 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 6 - 10 ชั่วโมง	43
ภาพประกอบ 28 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 11 - 15 ชั่วโมง	44
ภาพประกอบ 29 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 16 - 19 ชั่วโมง	45
ภาพประกอบ 30 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 20 - 23 ชั่วโมง	46
ภาพประกอบ 31 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 24 ชั่วโมง	47

บทที่ 1

บทนำ

ภูมิหลัง

ปัจจุบันมลพิษทางอากาศในกรุงเทพมหานครนั้นเป็นปัญหาที่มีความสำคัญอันดับต้น ๆ สืบเนื่องจากจำนวนประชากรที่เพิ่มมากขึ้นและการเจริญเติบโตทางด้านเศรษฐกิจที่เป็นไปอย่างรวดเร็ว ส่งผลให้มีการใช้พลังงาน ทรัพยากรที่เพิ่มมากขึ้น ทั้งในด้านอุตสาหกรรม การคมนาคมขนส่ง ซึ่งเป็นสาเหตุหลักที่ทำให้เกิดปัญหามลพิษทางอากาศ อย่างฝุ่นละอองขนาดเล็ก PM2.5 และ PM10 รวมถึงสารอันตรายประเภทต่าง ๆ ที่เจือปนอยู่ในอากาศ ปัญหามลพิษทางอากาศดังกล่าวส่งผลกระทบต่อสุขภาพของประชาชนเป็นอย่างมาก ดังปรากฏในรายงานวิจัย “ปกป้องสุขภาพคนไทย ถึงเวลาเปลี่ยนมาตรฐานฝุ่น PM2.5” (บัวคำศรี, 2562)

ฝุ่นละอองที่มีอยู่ในอากาศนั้นอาจเป็นได้ทั้งของแข็งหรือของเหลว ซึ่งมีขนาดตั้งแต่ 0.002 ไมครอน ไปจนถึงขนาดใหญ่กว่า 500 ไมครอน โดยฝุ่นละอองขนาดไม่เกิน 2.5 ไมครอนหรือเรียกอีกชื่อว่า “Particulate Matter 2.5 (PM2.5)” นั้นเป็นฝุ่นที่เกิดขึ้นจากการเผาไหม้ต่าง ๆ เช่น การเผาในที่โล่ง อุตสาหกรรมการผลิต การขนส่ง และการผลิตไฟฟ้า ซึ่งเป็นสาเหตุหลักที่ทำให้อากาศของกรุงเทพฯ ในปัจจุบันนี้มีปริมาณฝุ่นละออง PM2.5 เพิ่มขึ้นมากกว่าเมื่อก่อน และยังเป็นอันตรายมากกว่าฝุ่นละออง PM10 ที่สามารถพบได้ทั่วไปอีกด้วย เพราะ PM2.5 นั้นเป็นฝุ่นละอองที่มีขนาดค่อนข้างเล็ก เมื่อฝุ่นละอองดังกล่าวผ่านเข้าสู่ร่างกายแล้วจะสามารถแพร่กระจายสู่ทางเดินหายใจ กระแสเลือด และแทรกซึมกระบวนการทำงานในอวัยวะต่าง ๆ ซึ่งหากได้รับหรือสัมผัสกับสภาพอากาศดังกล่าวนานเข้าอาจก่อให้เกิดผลเสียต่อร่างกายจนถึงขั้นเป็นอันตรายต่อชีวิตได้

ผู้วิจัยเล็งเห็นความสำคัญในการศึกษาปริมาณความหนาแน่นของฝุ่นละออง การทำนายหาปริมาณฝุ่นล่วงหน้าว่าจะมีปริมาณที่เพิ่มมากขึ้นหรือลดน้อยลง โดยได้ทำการขอความอนุเคราะห์ข้อมูลที่จัดเก็บโดยกรมควบคุมมลพิษในพื้นที่กรุงเทพมหานครขึ้นในมาใช้ในการศึกษา โดยข้อมูลที่ได้จากการทำนายนั้นสามารถนำมาใช้เป็นข้อมูลพื้นฐานที่ใช้สำหรับวางแผนบริหารจัดการคุณภาพอากาศให้มีประสิทธิภาพมากยิ่งขึ้น และยังสามารถนำไปใช้กำหนดมาตรการในการแก้ไขปัญหาฝุ่นละอองที่เกิดจากหลาย ๆ สาเหตุ ตลอดจนถึงอาจยังช่วยลดผลกระทบที่อาจเกิดจากปัญหามลพิษทางอากาศในพื้นที่บริเวณกรุงเทพมหานครขึ้นได้อีกด้วย

ความมุ่งหมายของงานวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้ตั้งความมุ่งหมายไว้ดังนี้

1. เพื่อศึกษาการสร้างแบบจำลองและวิเคราะห์เปรียบเทียบสมรรถนะของแบบจำลองโครงข่ายประสาทเทียมแบบลึก Long Short-Term Memory (LSTM) กับแบบจำลองอนุกรมเวลาแบบ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) ในการทำนายปริมาณความหนาแน่นฝุ่นละออง PM2.5 ล่วงหน้า โดยใช้มาตรวัดเพื่อการวิเคราะห์แบบถดถอย (Regression Analysis)
2. เพื่อให้ทราบถึงแนวโน้มของปริมาณฝุ่นละออง PM2.5 ที่อาจจะเกิดขึ้นในช่วงเวลาของการทำนาย
3. เพื่อให้ได้ข้อมูลที่สามารถนำมาใช้ต่อยอดและปรับปรุงแบบจำลองให้ดียิ่งขึ้นในอนาคต
4. เพื่อให้ทราบถึงหลักการของอัลกอริทึม LSTM และแบบจำลองอนุกรมเวลา SARIMAX และนำมาปรับใช้งานได้อย่างเหมาะสม

ความสำคัญของการวิจัย

งานวิจัยนี้ศึกษาปริมาณความหนาแน่นของฝุ่นละอองและทำนายหาปริมาณฝุ่นล่วงหน้าว่าจะมีปริมาณที่เพิ่มมากขึ้นหรือลดน้อยลง โดยได้ทำการขอความอนุเคราะห์ข้อมูลที่จัดเก็บโดยกรมควบคุมมลพิษในพื้นที่กรุงเทพมหานครชั้นในมาใช้ในการศึกษา โดยหากกล่าวถึงเทคนิคการทำนายข้อมูลมลพิษทางอากาศที่เป็นแบบอนุกรมเวลารายชั่วโมงแล้วย่อมนึกถึงเทคนิคอนุกรมเวลาแบบ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) ซึ่งเป็นแบบจำลองดั้งเดิมที่รู้จักกันดี แต่ในปัจจุบันนี้แบบโครงข่ายประสาทเทียมแบบลึกนั้นกำลังเป็นที่นิยมค่อนข้างมากโดยเฉพาะเทคนิค Long Short-Term Memory Model (LSTM) ดังนั้นผู้วิจัยจึงสนใจและอยากศึกษาแบบจำลองทั้งสองแบบโดยใช้มาตรวัดเพื่อการวิเคราะห์แบบถดถอย (Regression Analysis) เป็นการเปรียบเทียบสมรรถนะ

ขอบเขตของการวิจัย

ประชากรที่ใช้ในการวิจัย

ใช้ข้อมูลที่ถูกเก็บมาเรียบร้อยแล้วจากสถานีตำรวจนครบาลโชคชัยโดยใช้ข้อมูลเป็นรายชั่วโมงตั้งแต่ปี 2017 - 2018 ประกอบด้วย feature ทั้งหมด 16 คอลัมน์ ได้แก่ year (ปี), month (เดือน), day (วันที่), hour (ชั่วโมง), ฝุ่นละออง PM2.5, CO (คาร์บอนมอนอกไซด์), NO

(ไนตริกออกไซด์), NO₂ (ไนโตรเจนไดออกไซด์), NO_x (ไนโตรเจนออกไซด์), ฝุ่นละออง PM₁₀, Wind speed (ความเร็วลม), Wind direction (ทิศทางลม), Temperature (อุณหภูมิ), Relative humidity (ค่าความชื้นสัมพัทธ์ในอากาศ), Pressure (ความกดอากาศ), Rain (ปริมาณน้ำฝน) โดยมีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00)

กลุ่มตัวอย่างที่ใช้ในการวิจัย

ใช้ข้อมูลจากสถานีตำรวจนครบาลโชคชัยโดยใช้ข้อมูลเป็นรายชั่วโมง ประกอบด้วย feature ทั้งหมด 16 คอลัมน์ ได้แก่ year (ปี), month (เดือน), day (วันที่), hour (ชั่วโมง), ฝุ่นละออง PM_{2.5}, CO (คาร์บอนมอนอกไซด์), NO (ไนตริกออกไซด์), NO₂ (ไนโตรเจนไดออกไซด์), NO_x (ไนโตรเจนออกไซด์), ฝุ่นละออง PM₁₀, Wind speed (ความเร็วลม), Wind direction (ทิศทางลม), Temperature (อุณหภูมิ), Relative humidity (ค่าความชื้นสัมพัทธ์ในอากาศ), Pressure (ความกดอากาศ), Rain (ปริมาณน้ำฝน) โดยมีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00) เป็นจำนวน 9,336 แถว

ตัวแปรที่ศึกษา

1. ตัวแปรอิสระ แบ่งได้ดังนี้

1.1 ข้อมูลเวลา

1.1.1 Year (ปี)

1.1.2 Month (เดือน)

1.1.3 Day (วันที่)

1.1.4 Hour (ชั่วโมง)

1.2 ข้อมูลสารก่อมลพิษทางอากาศ

1.2.1 CO (คาร์บอนมอนอกไซด์)

1.2.2 NO (ไนตริกออกไซด์)

1.2.3 NO₂ (ไนโตรเจนไดออกไซด์)

1.2.4 NO_x (ไนโตรเจนออกไซด์)

1.2.5 ฝุ่นละออง PM₁₀

1.3 ข้อมูลอุตุนิยมวิทยาและสภาพอากาศ

1.3.1 Wind speed (ความเร็วลม)

1.3.2 Wind direction (ทิศทางลม)

1.3.3 Temperature (อุณหภูมิ)

1.3.4 Relative humidity (ค่าความชื้นสัมพัทธ์ในอากาศ)

1.3.5 Pressure (ความกดอากาศ)

1.3.6 Rain (ปริมาณน้ำฝน)

2. ตัวแปรตาม ได้แก่ ฝุ่นละออง PM2.5

กรอบแนวคิดในงานวิจัย

งานวิจัยนี้พัฒนาแบบจำลองการทำนายโดยใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ที่ทำงานในรูปแบบของโปรแกรมคอมพิวเตอร์ โดยใช้ข้อมูลที่ถูกเก็บมาเรียบร้อยแล้วจากสถานีตำรวจนครบาลโชคชัยโดยใช้ข้อมูลเป็นรายชั่วโมงในการทำนาย ประกอบด้วย feature ทั้งหมด 16 คอลัมน์ ได้แก่ year (ปี), month (เดือน), day (วันที่), hour (ชั่วโมง), ฝุ่นละออง PM2.5, CO (คาร์บอนมอนอกไซด์), NO (ไนตริกออกไซด์), NO2 (ไนโตรเจนไดออกไซด์), NOX (ไนโตรเจนออกไซด์), ฝุ่นละออง PM10, Wind speed (ความเร็วลม), Wind direction (ทิศทางลม), Temperature (อุณหภูมิ), Relative humidity (ค่าความชื้นสัมพัทธ์ในอากาศ), Pressure (ความกดอากาศ), Rain (ปริมาณน้ำฝน) โดยมีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00) ในการทำนายปริมาณความหนาแน่น 24 ชั่วโมงข้างหน้า

เทคนิคการเรียนรู้ของเครื่องแบบโครงข่ายประสาทเทียมแบบลึกที่จะประยุกต์ใช้ในงานวิจัยนี้คือเทคนิค Long Short-Term Memory Model (LSTM) โดยจะศึกษาความแม่นยำในการทำนายเปรียบเทียบกับเทคนิคอนุกรมเวลาแบบ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) ด้วย Root Mean Squared Error (RMSE) และ Mean Absolute Error (MAE)

สมมุติฐานในการวิจัย

1. แบบจำลอง LSTM และแบบจำลอง SARIMAX สามารถทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ล่วงหน้าได้ 24 ชั่วโมง
2. แบบจำลอง Long Short-Term Memory Model มีประสิทธิภาพดีกว่าแบบจำลองอนุกรมเวลาแบบ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX)
3. การใช้เทคนิค Grid search ในการ Tuning Parameter จะช่วยเพิ่มประสิทธิภาพให้แบบจำลองสามารถทำนายด้วยความแม่นยำที่ดียิ่งขึ้น

บทที่ 2

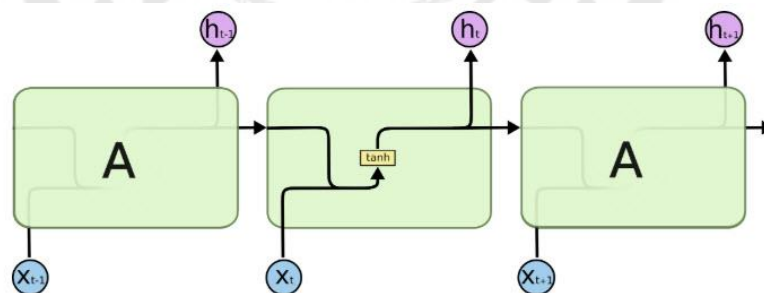
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และได้นำเสนอตามหัวข้อต่อไปนี้

1. ทฤษฎีเกี่ยวกับ Long Short-Term Memory (LSTM)
2. ทฤษฎีเกี่ยวกับ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX)
3. ทฤษฎีเกี่ยวกับการประเมินวัดความแม่นยำของแต่ละอัลกอริทึม
4. งานวิจัยที่เกี่ยวข้อง

การวิเคราะห์การถดถอย (Regression Analysis) เป็นวิธีการทางสถิติที่ศึกษาความสัมพันธ์ระหว่างตัวแปรตั้งแต่ 2 ตัวขึ้นไป ซึ่งได้แก่ ตัวแปรอิสระหรือเรียกว่าตัวพยากรณ์ (Predictor, X) และตัวแปรตาม (Dependent, y) ซึ่งความสัมพันธ์ของตัวแปรทั้งสองจะอยู่ในรูปแบบใด ๆ เช่นเส้นตรง เส้นโค้ง ฯลฯ (Sarkar, Bali, & Sharma, 2018)

ทฤษฎีเกี่ยวกับ Long Short-Term Memory (LSTM)

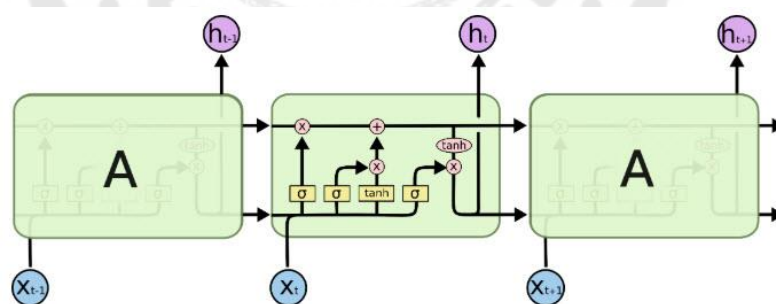


ภาพประกอบ 1 แสดงโครงสร้างการทำงานของ RNN

ที่มา : Olah, C. (2015). Understanding LSTM Networks -- colah's blog. Retrieved from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Long Short-Term Memory Model (LSTM) เป็นเทคนิคหนึ่งที่ถูกพัฒนาจาก Recurrent neural network (RNN) ซึ่ง RNN นั้นมีหลักการทำงาน คือ การนำ output (ผลลัพธ์) ที่ได้จากการคำนวณจากโหนดก่อนหน้านี้กลับมาใช้เป็นข้อมูล input ของโหนดถัดไป ซึ่งแต่ละโหนดของ RNN นั้นจะมีข้อมูลที่เข้ามา 2 ส่วน คือ ข้อมูล input ของโหนดนั้น ๆ กับ output ที่ผ่านการคำนวณจากโหนดก่อนหน้า โดยข้อมูลทั้ง 2 ชุดที่เข้ามาในโหนดจะถูกรวมเข้าด้วยกัน ก่อนจะถูกแยกผลลัพธ์ออกเป็น 2 ส่วน คือ ผลลัพธ์ที่ได้จากโหนดนั้น ๆ และผลลัพธ์ที่จะถูกนำไปเป็นข้อมูล input ของโหนดถัดไป เทคนิค RNNs นั้นเหมาะนำมาใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (sequence) หรือข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลอนุกรมเวลา (Time Series), ข้อมูลเสียง, ข้อมูลประเภทข้อความ, ข้อมูลรูปภาพและวิดีโอ เป็นต้น (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017)

ข้อดีของ RNN คือ สามารถนำข้อมูลก่อนหน้า (ในอดีต) มาใช้ในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคตได้ ส่วนข้อเสียของ RNN คือ จะสามารถดูข้อมูลย้อนหลังได้แค่เพียงระยะสั้น ๆ เท่านั้น ซึ่งทำให้เกิดปัญหาในการทำ Backpropagation หรือการคำนวณหาความผิดพลาดย้อนหลังของแต่ละโหนดเมื่อสิ้นสุดการทำงาน เพราะการทำ Backpropagation นั้นจะต้องทำย้อนกลับไปหลายขั้นตอนและหลายโหนด จึงทำให้เกิดปัญหา Vanishing Gradient Problem ดังนั้นเพื่อแก้ปัญหาดังกล่าวจึงทำให้เกิดเทคนิค LSTM ขึ้น

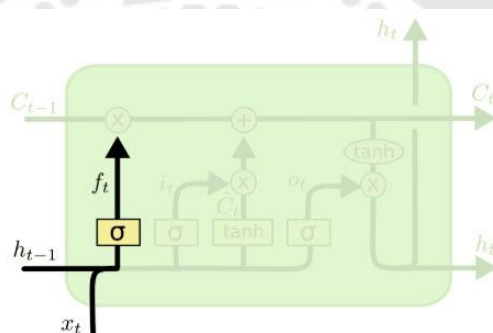


ภาพประกอบ 2 แสดงโครงสร้างการทำงานของ LSTM

ที่มา : Olah, C. (2015). Understanding LSTM Networks -- colah's blog. Retrieved from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Long short-term memory (LSTM) เป็นโครงข่ายประเภท RNNs รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น LSTM เริ่มเป็นที่รู้จักในปี ค.ศ.1997 โดย Hochreiter และ Schmidhuber (Hochreiter & Schmidhuber, 1997) โดยมีหลักการทำงาน คือ สามารถเก็บ 'สถานะ' หรือข้อมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับไปดูจะได้ทราบถึงที่มาของข้อมูลค่าดังกล่าวว่าเดิมเป็นค่าอะไร และจุดเด่นของเทคนิค LSTM คือฟังก์ชันพิเศษที่มีหน้าที่เหมือน 'ประตู (Gate)' ที่คอยควบคุมข้อมูลที่จะเข้ามาในแต่ละโหนด ซึ่งประกอบไปด้วย Forget gate layer, Input gate layer และ Output gate layer (Jozefowicz, Zaremba, & Sutskever, 2015)

Forget gate layer



ภาพประกอบ 3 แสดงโครงสร้าง Forget gate layer

ที่มา : Olah, C. (2015). Understanding LSTM Networks -- colah's blog. Retrieved from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน cell state นั้นควรจะถูกรักษาไว้ หรือควรที่จะทิ้งไป ซึ่งข้อมูลที่ถูกลบทิ้งจะถูกรักษาไว้จะถูกระบุจากข้อมูล input ที่เข้ามาใน โหนดนั้น ๆ รวมกับผลลัพธ์ที่ได้จากการคำนวณของโหนดก่อนหน้า ผ่านฟังก์ชัน sigmoid ดังสมการที่ (1)

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

จากสมการ

f_t คือ Forget gate

σ คือ ฟังก์ชัน sigmoid

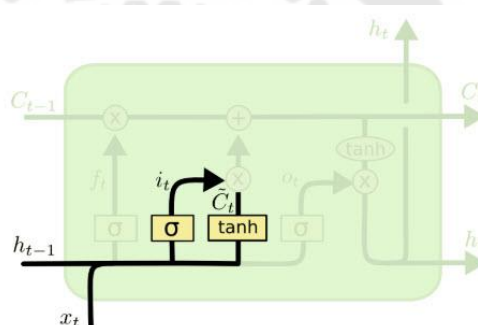
W_f คือ ค่าน้ำหนักของ matrices

h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp t-1)

x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t และ b_f คือ ค่า bias

ผลลัพธ์ที่ได้จาก Forget gate layer จะอยู่ระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้นหมายถึงให้ลบค่า cell state เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า cell state นี้ต่อไป

Input gate layer



ภาพประกอบ 4 แสดงโครงสร้าง Input gate layer

ที่มา : Olah, C. (2015). Understanding LSTM Networks -- colah's blog. Retrieved from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

เป็น Gate ที่มีหน้าที่รับข้อมูล input เข้ามาใหม่แล้วจึงทำการบันทึกหรือ 'เขียน (write)' ข้อมูลลงไปในแต่ละโหนด โดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ ถ้าต้องการ update cell state เมื่อทำการรับข้อมูล input เข้ามาแล้วฟังก์ชัน sigmoid ที่เป็นตัวควบคุมจะเรียกใช้ input gate เพื่อเลือกว่าจะให้ update cell state หรือไม่ ต้อง และในส่วนที่สอง ถ้า input gate เลือกที่จะทำการ update cell state ฟังก์ชัน tanh ก็จะมีการสร้าง candidate values ($\tilde{C}_t$) ขึ้นมาใน state ดังสมการที่ (2) และ (3)

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$C_t = \tanh (W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

จากสมการ

i_t คือ Input gate

σ คือ ฟังก์ชัน sigmoid

$\tilde{C}_t$ คือ ค่า candidate ของ cell state ที่เวลา t

$\tanh$ คือ ฟังก์ชัน tanh

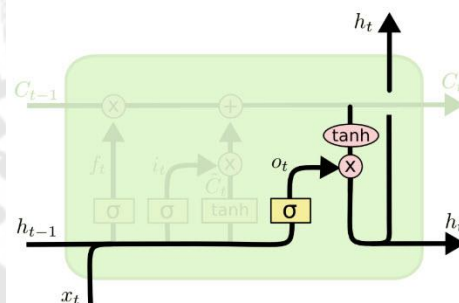
W_i, W_c คือ ค่าน้ำหนักของ matrices

h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp $t-1$)

x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t

b_i, b_c คือ ค่า bias

Output gate layer



ภาพประกอบ 5 แสดงโครงสร้าง Output gate layer

ที่มา : Olah, C. (2015). Understanding LSTM Networks -- colah's blog. Retrieved from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

เป็น Gate ที่มีหน้าที่เตรียมทำการส่งออกข้อมูล (output data) โดยข้อมูลที่จะทำการ output นั้นจะดูจาก cell state ที่ผ่านกระบวนการคำนวณต่าง ๆ แล้ว โดยฟังก์ชัน sigmoid จะเป็นตัวเลือกว่าข้อมูลส่วนไหนใน cell state ที่จะถูก output จากนั้นก็นำค่า cell state เข้าฟังก์ชัน

tanh (เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1) แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า output ที่ได้จาก sigmoid gate จากนั้นก็จะได้ค่า output ที่ต้องการดังสมการที่ (4)

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

จากสมการ

- o_t คือ Output gate
- σ คือ ฟังก์ชัน sigmoid
- W_o คือ ค่าน้ำหนักของ matrices
- h_{t-1} คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp t-1)
- x_t คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t
- b_o คือ ค่า bias

ซึ่งค่า output ที่ได้ออกมานั้นจะถูกแบ่งออกเป็น 2 ส่วน คือ ค่า output ที่ได้จากโหนดนั้น ๆ กับค่า output ที่จะถูกส่งไปเป็นข้อมูล input ของโหนดถัดไป

ตัวอย่างพารามิเตอร์ของอัลกอริทึม LSTM

- Number of neurons, nodes คือ จำนวน neurons ที่ใช้ในแต่ละ hidden layer ของ LSTM
- Epoch คือ จำนวนรอบที่โมเดลจะทำ forward และ back propagation วนไปเรื่อย ๆ จนครบ training sample ทั้งหมด
- Batch size คือ จำนวนของ sample ที่ถูก training ในหนึ่ง batch
- Window size คือ จำนวน data point ที่เราจะนำมาใช้ในการทำนายหา data point ถัดไป

ทฤษฎีเกี่ยวกับ Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX)

แบบจำลอง SARIMAX นั้นเป็นการรวมกันของการทำนายอนุกรมเวลา 3 เทคนิคประกอบไปด้วย Autoregressive (AR), Integrated (I), Moving Average (MA) (Hyndman & Athanasopoulos, 2018) ซึ่งในแต่ละเทคนิคนั้นจะมีหน้าที่ในการช่วยกรองข้อมูลหรือพยายามลด White noise pattern ให้เหลือน้อยที่สุด และส่วนที่ทำให้แบบจำลอง SARIMAX แตกต่างกับแบบจำลอง ARIMA คือ แบบจำลอง SARIMAX นั้นจะสามารถจัดการกับข้อมูลที่มีลักษณะเป็น

Seasonal ได้ดี (Shumway & Stoffer, 2017) และยังสามารถเพิ่มตัวแปรอื่นหรือ Exogenous variables เข้าไปในแบบจำลองได้ เพื่อช่วยเพิ่มประสิทธิภาพในการทำนายให้ดียิ่งขึ้น โดยมีพารามิเตอร์ $p, d, q, P, D, Q, s, \text{exog}$ โดย $AR = p, P$ (Seasonal), $I = d, D$ (Seasonal), $MA = q, Q$ (Seasonal), s = ตัวกำหนดช่วงเวลาของอนุกรมเวลา และ exog = exogenous variables

Integrated (I)

เทคนิค Integrated (I) นั้นใช้พารามิเตอร์ d เป็นตัวทำให้ข้อมูลที่เป็น Trend หรือ Seasonality ซึ่งเป็นข้อมูลที่มีแนวโน้มเป็น Non-Stationary ให้กลายเป็น Stationary โดยใช้เทคนิค Differencing คือ การหาค่าความแตกต่างระหว่างค่าปัจจุบันกับค่าก่อนหน้า (ค่าในอดีต) ดังสมการที่ (5)

$$\text{No Differencing } (d = 0) | y'_t = y_t \quad (5)$$

$$1^{st} \text{ Differencing } (d = 1) | y'_t = y_t - y_{t-1} \quad (6)$$

$$2^{nd} \text{ Differencing } (d = 2) | y'_t = y_t - y_{t-1} - (y_{t-1} - y_{t-2}) = y_t - 2 \times y_{t-1} + y_{t-2} \quad (7)$$

Autoregressive model (AR)

เทคนิค Autoregressive (AR) นั้นจะใช้หลังจากทำการ Differencing ข้อมูลให้เป็น Stationary เรียบร้อยแล้ว โดย AR นั้นใช้พารามิเตอร์ p ซึ่งเป็นค่า lag (ค่าความล่าช้า) ในอนุกรมเวลา ซึ่งจะนำข้อมูลก่อนหน้านี้นี้มาใช้ในการทำนายค่าในอนาคตหรือก็คือค่าปัจจุบัน ดังสมการที่ (8)

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (8)$$

จากสมการ

c คือ ค่าคงที่

ϕ_p คือ order ของ Autoregressive

y_{t-p} คือ ค่าของ time series ที่เวลา $t-p$

ε_t คือ ค่า error ของโมเดล (ค่า white noise)

Moving Average model (MA)

ในเทคนิค Moving Average (MA) มีหลักการทำงานค่อนข้างคล้ายกับ Autoregressive (AR) แต่แตกต่างกันตรงที่ Moving Average จะสนใจความสัมพันธ์ระหว่าง error ของช่วงเวลาก่อนหน้ากับ error ของช่วงเวลาปัจจุบัน ซึ่งใช้พารามิเตอร์ q ในการหา error ดังสมการที่ (9)

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (9)$$

จากสมการ

c คือ ค่าคงที่

ε_t คือ ค่า error ของโมเดล (ค่า white noise)

θ_q คือ order ของ Moving Average

ε_{t-q} คือ ผลรวมระหว่างค่าปัจจุบันกับค่าก่อนหน้าของ error term

Seasonal ARIMA Model with External Regressors

แบบจำลอง SARIMAX นั้นมีความแตกต่างกับแบบจำลอง ARIMA ตรงที่แบบจำลอง SARIMAX นั้นสามารถจัดการกับข้อมูลที่มีลักษณะเป็น Seasonal ได้ดี โดยใช้พารามิเตอร์ s เป็นตัวกำหนดช่วงเวลาและยังสามารถเพิ่มตัวแปรนอกหรือ Exogenous variables เข้าไปในแบบจำลองได้ โดยใช้พารามิเตอร์ $exog$ เพื่อช่วยให้การทำนายมีประสิทธิภาพมากขึ้น (Vagropoulos, Chouliaras, Kardakos, Simoglou, & Bakirtzis, 2016) ดังสมการที่ (10) และสมการที่ (11)

SARIMA Model

$$\varphi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \quad (10)$$

SARIMAX Model

$$\varphi_p(B)\Phi_P(B^s)\nabla^d\nabla_s^D y_t = \beta_k x_{k,t}' + \theta_q(B)\Theta_Q(B^s)\varepsilon_t \quad (11)$$

จากสมการ

y_t คือ ตัวแปรที่ใช้สำหรับทำนาย

$\varphi_p(B)$ คือ order ที่เป็น non-seasonal ของ Autoregressive

$\Phi_p(B^s)$ คือ order ที่เป็น seasonal ของ Autoregressive

$\theta_q(B)$ คือ order ที่เป็น non-seasonal ของ Moving Average

$\Theta_Q(B^s)$ คือ order ที่เป็น seasonal ของ Moving Average

∇^d คือ order ของการ differencing ที่เป็น non-seasonal ของ Integrated

∇_s^D คือ order ของการ differencing ที่เป็น seasonal ของ Integrated

ε_t คือ ค่า error ของโมเดล (ค่า white noise)

β_k คือ ค่าสัมประสิทธิ์ของตัวแปร exogenous ที่ใส่เข้ามา k^{th}

$x_{k,t}'$ คือ ค่า vector ที่ประกอบด้วยตัวแปรที่ถูก input เข้ามา k^{th} ที่เวลา t

ตัวอย่างพารามิเตอร์ที่ใช้ในอัลกอริทึม SARIMAX :

- p คือ order ของ Autoregressive (lag order), ตัวแปรล่าช้าของผลต่างของตัวแปร
- d คือ จำนวนในการทำ Differencing ทั้งหมด
- q คือ order ของ Moving Average
- P คือ order ของ Autoregressive (lag order) ที่เป็น seasonal
- D คือ จำนวนในการทำ Differencing ทั้งหมด ที่เป็น seasonal
- Q คือ order ของ Moving Average ที่เป็น seasonal
- exog คือ การเพิ่มตัวแปรตัวอื่นเข้าไปในแบบจำลอง ARIMA เพื่อเพิ่มความแม่นยำในการช่วยทำนาย
- s คือ ตัวกำหนดช่วงเวลาของอนุกรมเวลาและจะขึ้นอยู่กับข้อมูลดิบที่ถูกนำมาใช้ เช่น $s = 4$ (คือ 4 ไตรมาส = 1 ปี) หรือ $s = 12$ (คือ 12 เดือน = 1 ปี) เป็นต้น

ทฤษฎีเกี่ยวกับการประเมินวัดความแม่นยำของแต่ละอัลกอริทึม

ในส่วนของการทำนายแบบถดถอย (Regression) ในงานวิจัยนี้จะใช้ค่า Root Mean Squared Error (RMSE) และ Mean Absolute Error (MAE) ในการวัดค่าความแม่นยำของอัลกอริทึม

1. รากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (Root Mean Squared Error: RMSE) คือ การวัดความแตกต่างระหว่างค่าจริงและค่าพยากรณ์ (ค่าทำนาย) ถ้าหากค่า RMSE นั้นมีค่าน้อย แสดงว่าค่าทำนายนั้นประมาณค่าได้ใกล้เคียงกับค่าจริง ดังสมการที่ (12)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (12)$$

จากสมการ

Y_i คือ ค่าจริงหรือ ค่า actual value ของ samples test ที่ i

$\hat{Y}_i$ คือ ค่าที่ประมาณได้หรือ ค่า predict ของ samples test ที่ i

n คือ จำนวน samples ทั้งหมด

2. ค่าคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error: MAE) คือ การหาค่าเฉลี่ยของความแตกต่างสัมบูรณ์ระหว่างค่าพยากรณ์(ค่าทำนาย)และค่าจริง ถ้าหากค่า MAE นั้นมีค่าน้อย แสดงว่าค่าทำนายนั้นมีค่าใกล้เคียงกับค่าจริง ดังสมการที่ (13)

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (13)$$

จากสมการ

Y_i คือ ค่าจริงหรือ ค่า actual value ของ samples test ที่ i

$\hat{Y}_i$ คือ ค่าที่ประมาณได้หรือ ค่า predict ของ samples test ที่ i

n คือ จำนวน samples ทั้งหมด

งานวิจัยที่เกี่ยวข้อง

การทบทวนวรรณกรรมของงานวิจัยนี้ได้ทำการศึกษางานวิจัยที่เกี่ยวข้องกับทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 โดยงานวิจัยที่เกี่ยวข้องมีรายละเอียดดังต่อไปนี้

(1) บทความวิจัยเรื่อง Prediction of PM10 using Support Vector Regression โดย Soawalak Arampongsanuwat และ Phayung Meesad (Arampongsanuwat & Meesad, 2011)

งานวิจัยนี้ได้ทำการพัฒนาแบบจำลอง Support Vector Regression (SVR) เพื่อใช้สำหรับการทำนายค่าฝุ่นละออง PM10 ในกรุงเทพฯ โดยใช้พารามิเตอร์ที่มีความสัมพันธ์กับฝุ่นละออง PM10 เป็นตัวแปรในการเพิ่มประสิทธิภาพของโมเดล ซึ่งประกอบด้วย globe radiation, net radiation, air pressure, rainfall, relative humidity, temperature, wind direction และ wind

speed รวมถึงความหนาแน่นของปริมาณสารต่าง ๆ ที่ตรวจวัดได้ในอากาศ คือ carbon monoxide, ozone, nitrogen dioxide และ sulfur dioxide โดยใช้ข้อมูลตั้งแต่ปี 2007 – 2009 ซึ่งผลที่ได้จากการทดลองในการสร้างแบบจำลองโดยการกำหนดให้ค่า $C = 5.000$, $\epsilon = 0.001$ และ $\sigma = 0.1$ ซึ่งได้ค่า MSE ออกมาดีที่สุดคือ 1.0588×10^{-10}

(2) บทความวิจัยเรื่อง A Hybrid ARIMA and Neural Networks Model for PM-10 Pollution Estimation: The Case of Chiang Mai City Moat Area โดย Rati Wongsathan และ Isaravuth Seedadan (Wongsathan & Seedadan, 2016)

งานวิจัยนี้ได้ทำการนำเอาแบบจำลอง Autoregressive Integrated Moving Average (ARIMA) และ Neural networks (NNs) มาปรับปรุงพัฒนาโดยนำทั้งสองแบบจำลองมาทำการ hybrid กัน ซึ่งได้เป็นแบบจำลอง hARIMA-NNs ซึ่งใช้ข้อมูลจากกรมอุตุนิยมวิทยาและข้อมูลฝุ่นละออง PM10 ของจังหวัดเชียงใหม่ ตั้งแต่ปี 2011-2014 ประกอบด้วย wind speed, wind direction, humidity, temperature และฝุ่นละออง PM10 เป็นต้น โดยใช้แบบจำลอง ARIMA, NNs และ hARIMA-NNs ในการทำนายค่าความแม่นยำของฝุ่นละออง PM10 ผลลัพธ์ที่ได้จากการกำหนดค่าให้ ARIMA(5, 1, 5), NNs(10, 5) และ hARIMA(5, 1, 5)NNs([5,5], 5) สรุปผลได้ว่าแบบจำลอง hARIMA-NNs นั้นให้ค่าออกมาดีที่สุด

(3) บทความวิจัยเรื่อง Long Short-Term Memory Neural Network for Air Pollutant Concentration Predictions: Method Development and Evaluation โดย Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, Yuan Hu, Chengzeng You, Tianhe Chi (Li et al., 2017)

งานวิจัยนี้ได้ใช้แบบจำลอง Long Short-Term Memory Neural Network Extended (LSTME) ซึ่งมีการใช้ spatiotemporal correlations มาช่วยเพิ่มประสิทธิภาพในการทำนายค่าความแม่นยำของฝุ่นละออง PM2.5 โดยเปรียบเทียบกับแบบจำลอง Long Short-term Memory (LSTM NN), The Spatiotemporal Deep Learning (STDL), The Time Delay Neural Network (TDNN), Autoregressive Moving Average (ARMA) และ Support Vector Regression (SVR) ซึ่งใช้ข้อมูลเป็นรายชั่วโมงตั้งแต่วันที่ 1 มกราคม 2014 ถึง 2 พฤษภาคม 2016 ทั้งหมด 12 สถานีในเมืองปักกิ่ง ประเทศจีน โดยเลือกใช้ตัวแปรจากข้อมูลกรมอุตุนิยมวิทยา (Meteorological Dataset) ที่มีความสัมพันธ์กับฝุ่นละออง PM2.5 ประกอบด้วย temperature, humidity, wind speed และ visibility ในงานวิจัยนี้มีการนำ spatiotemporal มาวิเคราะห์เพื่อหา correlations ของข้อมูลฝุ่นละออง PM2.5 ทั้ง 12 สถานี ซึ่งผลลัพธ์ที่ได้จากการนำข้อมูลของหลาย ๆ สถานีมาใช้ในการทำนายค่าความแม่นยำนั้นให้ผลออกมาดีกว่าการใช้ข้อมูลจากสถานีเดียว

จากการเปรียบเทียบความแม่นยำด้วยค่า RMSE, MAE และ MAPE ของแต่ละแบบจำลอง สรุปผลได้ว่าแบบจำลอง LSTM นั้นให้ค่า RMSE, MAE และ MAPE ออกมาดีที่สุด

(4) บทความวิจัยเรื่อง PM10 Density Forecast Model Using Long Short-Term Memory โดย Jung-Hwan Park, Seong-Joon Yoo, Kyung-Joong Kim, Yeong-Hyeon Gu, Keon-Hoon Lee และ U-Hyon Son (Park et al., 2017)

งานวิจัยนี้ได้ทำการเปรียบเทียบแบบจำลองระหว่าง Linear Regression กับ Recurrent Neural Network ที่ใช้ LSTM (RMSProp), LSTM (Adagrad) และ RNN ในการทำนายค่าความแม่นยำของฝุ่นละออง PM10 โดยมีการใช้เทคนิค Moving Average (MA) เพื่อช่วยลดค่า Noise หลังจากกำจัดข้อมูลที่เป็น NaN ออกเรียบร้อยแล้ว ซึ่งใช้ข้อมูลจากเว็บไซต์ AirKorea บริเวณกรุงโซล ประเทศเกาหลี ตั้งแต่วันที่ 1 มกราคม 2005 ถึง 31 มีนาคม 2016

จากการเปรียบเทียบค่าความแม่นยำด้วยค่า RMSE และ MSE ของแต่ละแบบจำลอง สรุปผลได้ว่า LSTM ที่ถูกกำหนด Optimizer เป็น RMSProp นั้นให้ค่า RMSE และ MSE ออกมาดีที่สุด

(5) บทความวิจัยเรื่อง Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality โดย Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li และ Zhongfei (Mark) Zhang (Qi et al., 2017)

งานวิจัยนี้ได้ทำการพัฒนาแบบจำลอง The Deep Air Learning (DAL) ขึ้นเพื่อใช้แก้ปัญหาต่าง ๆ โดยความคิดรวบยอดหลักของแบบจำลอง DAL นั้นคือได้นำเอา feature selection และ spatiotemporal มาปรับใช้ในแบบจำลองเพื่อช่วยเพิ่มประสิทธิภาพในการทำนายค่าฝุ่นละออง PM2.5 ให้มีเพิ่มมากขึ้น ซึ่งใช้ข้อมูลตั้งแต่ช่วงเดือนพฤศจิกายนถึงธันวาคม 2015 (60 วัน) บริเวณปักกิ่ง ประเทศจีน ในการทำนายค่าฝุ่นละออง PM2.5 ล่วงหน้า 49 ชั่วโมง (0-48 ชม.) ข้อมูลนั้นประกอบด้วย temperature, humidity, barometric pressure, precipitation, wind direction, wind strength, PM2.5, PM10, SO2, NO2, CO และ O3 โดยทำการเปรียบเทียบกับแบบจำลอง Logistic Regression, L_1 -regularized Logistic Regression, L_1 -regularized Logistic Regression (ST), Neural Network, Autoencoder, Laplacian Regression, ARIMA และ RNN

ผลลัพธ์ที่ได้จากการเปรียบเทียบความแม่นยำด้วยค่า RMSE ของแต่ละแบบจำลอง ในแต่ละช่วงเวลานั้นสรุปได้ว่าแบบจำลอง DAL นั้นให้ค่า RMSE ออกมาดีที่สุด

(6) บทความวิจัยเรื่อง DeepAirNet: Applying Recurrent Networks for Air Quality Prediction โดย Athira V., Greetha P., Vinayakumar R. และ Soman K P. (V, P, R, & K P, 2018)

งานวิจัยนี้ได้ใช้แบบจำลอง deep learning ในการทำนายความแม่นยำของค่าฝุ่นละออง PM10 โดยใช้แบบจำลอง Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), และ Gated Recurrent Unit (GRU) ซึ่งใช้ข้อมูลทั้งหมด 1,498 สถานีจาก China National Environmental Monitoring Center (CNEMC) และ Global Forecasting System (GFS) ตั้งแต่วันที่ 1 เมษายน 2015 ถึง 1 กันยายน 2017 ข้อมูลที่นำมาใช้ประกอบด้วย PM10, PM2.5, NO2, CO, O3, SO2, AQI, temperature, relative humidity, u-wind component, v-wind component, precipitation rate และ total cloud cover

จากการเปรียบเทียบค่าความแม่นยำด้วยค่า RMSE และ MAPE ของแต่ละแบบจำลองสรุปผลได้ว่าแบบจำลอง GRU นั้นให้ค่า RMSE และ MAPE ออกมาดีที่สุด

งานวิจัยเหล่านี้ได้ศึกษาและทำการวิเคราะห์เกี่ยวกับฝุ่นละออง PM2.5 และได้ใช้แบบจำลองตัวอื่นในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ผู้วิจัยเห็นว่างานวิจัยเหล่านี้มีประโยชน์ในการใช้อ้างอิงจึงเขียนรายละเอียดสรุปออกมา ซึ่งในส่วนที่ถูกต้องไว้่นั้นเป็นแบบจำลองที่ให้ค่าออกมาดีที่สุดได้ดังภาพประกอบ 6

(7) Prediction of PM2.5 Based on Elman Neural Network with Chaos Theory แบบจำลอง Elman-chaos ให้ค่าออกมาดีที่สุด (Hu, Li, & Qiao, 2016)

(8) Comparative Analysis of different Statistical Methods for Prediction of PM2.5 and PM10 Concentrations in Advance for Several Hours แบบจำลอง Random Forest (RF) ให้ค่าออกมาดีที่สุด (Saeed, Hussain, Awan, & Idris, 2017)

(9) A Novel Hybrid-Garch Model Based on ARIMA and SVM for PM 2.5 Concentrations Forecasting แบบจำลอง Hybrid-Garch ให้ค่าออกมาดีที่สุด (Wang, Zhang, Qin, & Zhang, 2017)

(10) An Empirical Study of PM2.5 Forecasting Using Neural Network แบบจำลอง Neural Network Autoregression (NNAR) ให้ค่าออกมาดีที่สุด (Mahajan, Chen, & Tsai, 2017)

(11) Double Layer Back Propagation Neural Network Based on Restricted Boltzmann Machines for Forecasting Daily Particulate Matter 2.5 แบบจำลอง Restricted

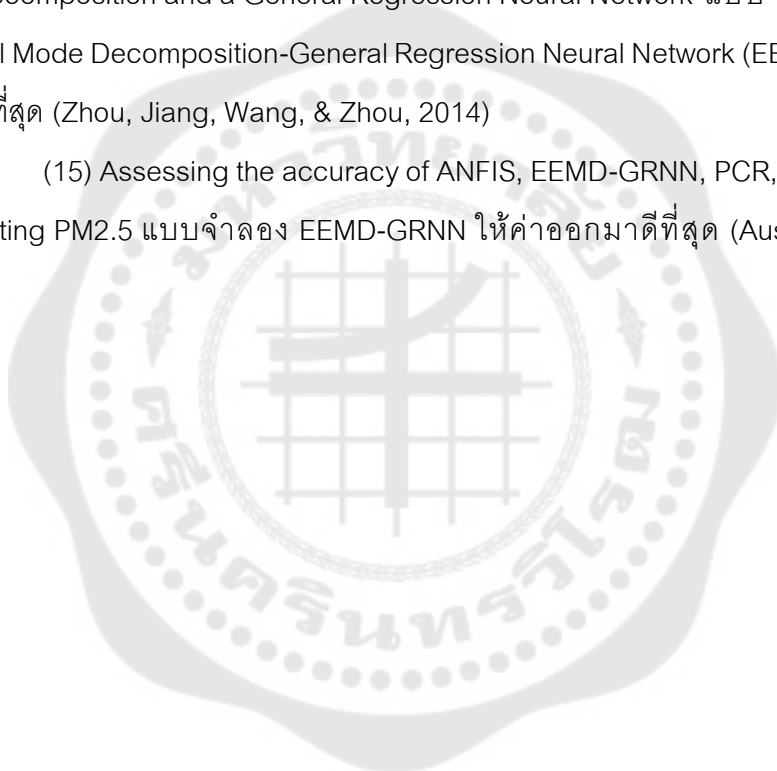
Boltzmann Machines Double Layer Back Propagation Neural Network Model (RBM-DL-BPNN) ให้ค่าออกมาดีที่สุด (Fu, Wang, Le, & Manko, 2017)

(12) Deep Distributed Fusion Network for Air Quality Prediction แบบจำลอง DeepAir ให้ค่าออกมาดีที่สุด (Yi, Zhang, Wang, Li, & Zheng, 2018)

(13) Deep Neural Network for PM2.5 Pollution Forecasting Based on Manifold Learning แบบจำลอง M-DNN ให้ค่าออกมาดีที่สุด (Xie, 2017)

(14) A Hybrid Model for PM2.5 Forecasting Based on Ensemble Empirical Mode Decomposition and a General Regression Neural Network แบบจำลอง Ensemble Empirical Mode Decomposition-General Regression Neural Network (EEMD-GRNN) ให้ค่าออกมาดีที่สุด (Zhou, Jiang, Wang, & Zhou, 2014)

(15) Assessing the accuracy of ANFIS, EEMD-GRNN, PCR, and MLR models in predicting PM2.5 แบบจำลอง EEMD-GRNN ให้ค่าออกมาดีที่สุด (Ausati & Amanollahi, 2016)



สรุปงานวิจัยที่ศึกษาทั้งหมด

No.	The name of researcher	Year	Country	Model					
				ARIMA	LSTM	Hybrid	Etc.		
1	Prediction of PM10 using Support Vector Regression	2011	Thailand				SVR		
2	A Hybrid ARIMA and Neural Networks Model for PM ₁₀ Pollution Estimation: The Case of Chiang Mai City Moist Area	2016	Thailand	ARIMA		h-ARIMA-NN%	NN%		
3	Long Short-Term Memory Neural Network for Air Pollutant Concentration Prediction: Method Development and Evaluation	2017	China	ARMA	LSTM NN LSTM		TDNN SVR	STD	
4	PM10 Density Forecast Model Using Long Short-Term Memory	2017	Korea		LSTM(RMSEProp) LSTM(adagrad)		Linear Regression RNN		
5	Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality	2018	China	ARIMA			Logistic Regression L1-regularized LR L1-regularized LR(ST)	NN RNN	Laplacian R. Autoencoder DAL
6	DeepAirNet: Applying Recurrent Networks for Air Quality Prediction	2018	China		LSTM GRU		RNN		
7	Prediction of PM _{2.5} Based on Elman Neural Network with Chaos Theory	2016	China			Elman-chaos RBF-chaos MLP-chaos	RBF Elman	MLP	
8	Comparative Analysis of Different Statistical Methods: for Prediction of PM _{2.5} and PM10 Concentration: in Advance for Several Hours	2017	Pakistan				linear radial SVR%	RF	
9	A Novel Hybrid-Garch Model Based on ARIMA and SVN for PM _{2.5} Concentration: Forecasting	2017	China	ARIMA		Hybrid-Garch	SVN		
10	An Empirical Study of PM _{2.5} Forecasting Using Neural Network	2017	Taiwan	ARIMA		NNAR	Holt-Winters		
11	Double Layer Back Propagation Neural Network Based on Restricted Boltzmann Machines for Forecasting Daily Particulate Matter 2.5	2017	China			FEM-DL-BPNN DL-BPNN	BPNN		
12	Deep Distributed Fusion Network for Air Quality Prediction	2018	China	ARIMA	LSTM LSTM-STC		Lasso	GBDT DeepST	DeepFM DeepAir (DNN)
13	Deep Neural Network for PM _{2.5} Pollution Forecasting Based on Manifold Learning	2017	China			M-DNN	DNN	BPNN	
14	A Hybrid Model for PM _{2.5} Forecasting Based on Ensemble Empirical Mode Decomposition and a General Regression Neural Network	2014	China	ARIMA		EEMD-GRNN	MLR	PCR GRNN	
15	Assessing the accuracy of ANFIS, EEMD-GRNN, PCR and MLR Model in Predicting PM _{2.5}	2016	Iran			EEMD-GRNN	MLR	PCR ANFIS	

ภาพประกอบ 6 แสดงตารางสรุปผลงานวิจัยที่ศึกษา
(ส่วนที่ถูกไฮไลท์ไว้เป็นแบบจำลองที่ได้ออกมาดีที่สุด)

บทที่ 3

วิธีดำเนินการวิจัย

ในการวิจัยครั้งนี้ ผู้วิจัยได้ดำเนินการตามขั้นตอนดังนี้

1. การกำหนดประชากรและการสุ่มกลุ่มตัวอย่าง
2. การสร้างเครื่องมือที่ใช้ในการวิจัย
3. การเก็บรวบรวมข้อมูล
4. การจัดกระทำและการวิเคราะห์ข้อมูล

การกำหนดประชากรและการสุ่มกลุ่มตัวอย่าง

ประชากร

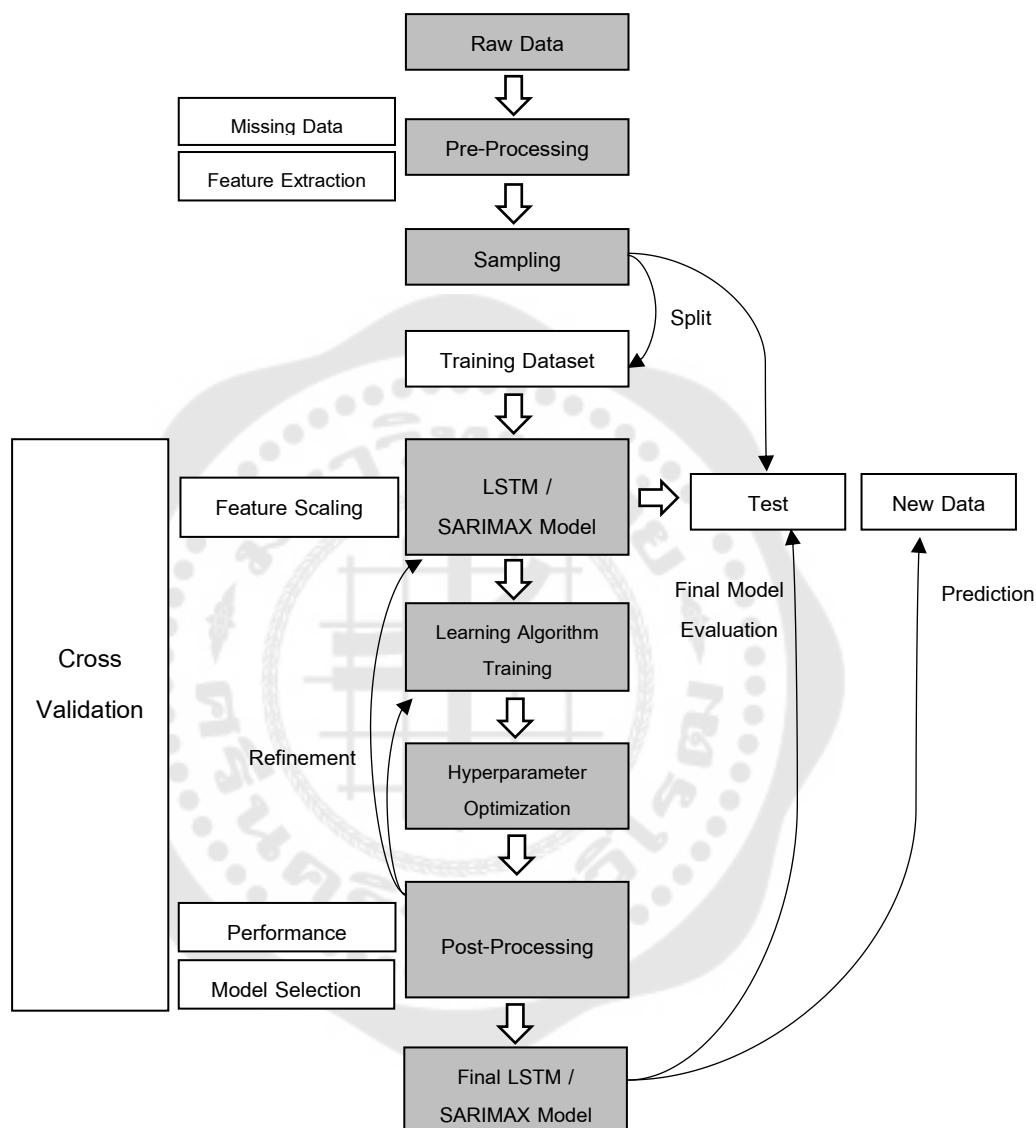
ใช้ข้อมูลที่ถูกเก็บมาเรียบร้อยแล้วจากสถานีตำรวจนครบาลโชคชัยโดยใช้ข้อมูลเป็นรายชั่วโมงตั้งแต่ปี 2017 - 2018 ประกอบด้วย feature ทั้งหมด 16 คอลัมน์ ได้แก่ year (ปี), month (เดือน), day (วันที่), hour (ชั่วโมง), ฝุ่นละออง PM2.5, CO (คาร์บอนมอนนอกไซด์), NO (ไนตริกออกไซด์), NO2 (ไนโตรเจนไดออกไซด์), NOX (ไนโตรเจนออกไซด์), ฝุ่นละออง PM10, Wind speed (ความเร็วลม), Wind direction (ทิศทางลม), Temperature (อุณหภูมิ), Relative humidity (ค่าความชื้นสัมพัทธ์ในอากาศ), Pressure (ความกดอากาศ), Rain (ปริมาณน้ำฝน) โดยมีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00)

การเลือกกลุ่มตัวอย่าง

ข้อมูลจะถูกแบ่งออกเป็น 2 ชุด คือ ชุดสำหรับสร้างแบบจำลอง (Train Dataset) และชุดสำหรับทดสอบแบบจำลอง (Test Dataset)

การสร้างเครื่องมือที่ใช้ในการวิจัย

กระบวนการทำงานของแบบจำลอง LSTM และ SARIMAX



ภาพประกอบ 7 Flowchart กระบวนการทำงานของแบบจำลอง LSTM และ
แบบจำลอง SARIMAX

ที่มา : Supervised Learning - a workflow chart : Machine Learning. (2019).
Retrieved from https://www.reddit.com/r/MachineLearning/comments/2e3vcs/supervised_learning_a_workflow_chart/

จากรูป Flowchart ด้านบนได้อธิบายถึงกระบวนการทำงานของแบบจำลอง LSTM และแบบจำลอง SARIMAX โดยเริ่มตั้งแต่ขั้นตอนแรกที่น่าข้อมูลเข้ามา ทำความสะอาดข้อมูล แบ่งข้อมูลออกเป็น Train, Test แล้วจึงนำข้อมูล Train ใส่เข้าแบบจำลอง ให้แบบจำลองทำการเรียนรู้ข้อมูล จากนั้นก็ทำการเลือกพารามิเตอร์ที่เหมาะสม (Grid search) เพื่อให้แบบจำลองทำนายค่าออกมาดีที่สุด และสุดท้ายก็นำข้อมูลที่ได้จากการทำนายมาเทียบกับข้อมูล test เพื่อประเมินความแม่นยำของแบบจำลอง

การเก็บรวบรวมข้อมูล

งานวิจัยนี้ได้ขอความอนุเคราะห์นำข้อมูลที่เก็บโดยกรมควบคุมมลพิษในพื้นที่กรุงเทพมหานครชั้นในมาใช้ในการศึกษา ซึ่งได้เลือกใช้ Dataset ที่มีข้อมูลฝุ่นละออง PM2.5 บริเวณพื้นที่สถานีตำรวจนครบาลโชคชัย

- มีข้อมูล Feature ทั้งหมด 16 คอลัมน์ ประกอบด้วย year, month, day, hour, PM2.5, CO, NO, NO2, NOX, PM10, Wind speed, Wind direction, Temperature, Relative humidity, Pressure, Rain

- มีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00)

- ข้อมูลของปี 2017 มีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 31-12-2017 (เวลา 23:00) จำนวน 7 เดือน

- ข้อมูลของปี 2018 มีข้อมูลตั้งแต่วันที่ 01-01-2018 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00) จำนวน 6 เดือน

ตาราง 1 แสดงรายละเอียดของแต่ละ Feature

Variable	Description
CO (ppm)	คาร์บอนมอนนอกไซด์
NO (ppb)	ไนตริกออกไซด์
NOX (ppb)	ไนโตรเจนออกไซด์
NO2 (ppb)	ไนโตรเจนไดออกไซด์
PM10 ($\mu\text{g}/\text{m}^3$)	ฝุ่นละออง PM10
PM2.5 ($\mu\text{g}/\text{m}^3$)	ฝุ่นละออง PM2.5
Wind speed (m/s)	ความเร็วลม
Wind direction (degrees)	ทิศทางลม
Temperature	อุณหภูมิ
Relative humidity (%RH)	ค่าความชื้นสัมพัทธ์ในอากาศ
Rain (mm)	ปริมาณน้ำฝน
Pressure (mmHg)	ความกดอากาศ
Year	ปี
Month	เดือน
Day	วันที่
Hour	ชั่วโมง

การจัดกระทำและการวิเคราะห์ข้อมูล

1. ทำการเลือกข้อมูล Dataset บริเวณพื้นที่สถานีตำรวจนครบาลโชคชัย และทำการจัดรูปแบบข้อมูลให้เหมาะสม เพื่อให้สามารถนำเข้ามาใช้งานได้

2. ศึกษาและทำการวิเคราะห์ feature ใน Dataset โดยทำการ clean ข้อมูลที่เป็น missing value และ outliers ด้วยวิธีการ drop ข้อมูลทิ้งทั้งแถว

3. ศึกษาความสัมพันธ์ระหว่างค่าฝุ่นละออง PM2.5 กับ feature ตัวอื่น (Explore data)

4. สร้างแบบจำลองในการทำนายข้อมูล ด้วยภาษา Python และทำการปรับค่าพารามิเตอร์ให้เหมาะสมโดยใช้เทคนิค Hyperparameter Selection หรือ Grid Search

4.1 สร้างแบบจำลองโดยใช้อัลกอริทึม Long Short-Term Memory (LSTM)

4.2 สร้างแบบจำลองโดยใช้อัลกอริทึม Seasonal Autoregressive Integrated Moving Averages with Exogenous Regressors (SARIMAX)

5. เครื่องมือที่ใช้วัดประสิทธิภาพของแบบจำลอง

5.1 RMSE

5.2 MAE



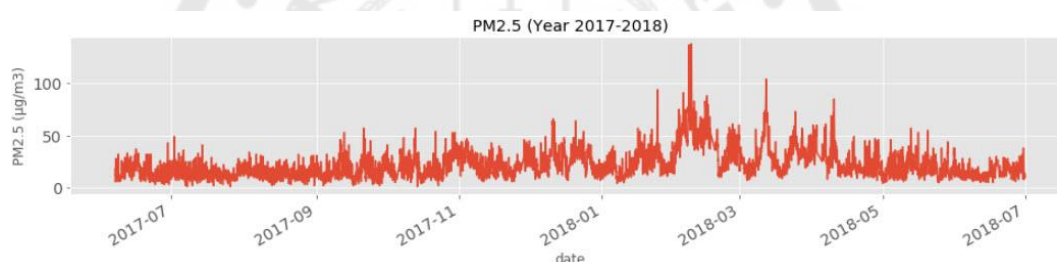
year	month	day	hour	PM2.5	CO	NO	NO2	NOX	PM10	Wind speed	Wind dir	Temp	Rel hum	Pressure	Rain
2017	6	7	0	6	0.4	23	16	39	18	0.1	243	27.8	77	754	0
2017	6	7	1	7	0.6	37	15	52	22	0	167	27.8	80	754	0
2017	6	7	2	8	0.7	53	16	69	25	0.2	121	27.7	84	753	0
2017	6	7	3	14	0.6	48	16	64	37	0.2	93	27.1	87	753	0
2017	6	7	4	19	0.6	30	13	43	33	0.3	86	26.8	88	753	0
2017	6	7	5	11	0.3	24	12	37	27	0.3	87	26.7	89	753	0
2017	6	7	6	12	0.6	35	13	48	32	0.3	78	26.8	89	754	0
2017	6	7	7	26	0.9	48	15	62	42	0.3	69	27.9	84	754	0
2017	6	7	8	28	0.5	39	18	57	44	0.5	66	29.2	72	755	0
2017	6	7	9	11	0.4	25	19	45	29	0.3	67	30	67	755	0
2017	6	7	10	10	0.1	14	16	30	25	0.5	50	30.7	62	755	0
2017	6	7	11	14	0	10	16	26	19	0.4	298	31.2	60	754	0
2017	6	7	12	11	NaN	NaN	NaN	NaN	22	0.3	315	31.6	59	754	0
2017	6	7	13	9	0.1	6	14	20	15	0.2	320	31.8	59	753	0
2017	6	7	14	10	0.1	11	16	27	15	0.3	270	31.7	60	752	0
2017	6	7	15	9	0.2	9	15	24	19	0.3	31	31.4	61	752	0
2017	6	7	16	6	0.7	7	14	21	15	0.4	21	30.8	65	752	0
2017	6	7	17	8	0.9	7	14	21	21	0.6	59	30.4	70	752	0
2017	6	7	18	6	1	6	15	21	20	0.3	31	29.8	71	752	0.6
2017	6	7	19	8	0.6	28	21	49	18	0.2	86	24.9	94	753	4
2017	6	7	20	7	0.2	23	21	44	14	0.3	72	25	94	754	0.4
2017	6	7	21	7	0.2	15	23	37	16	0.4	85	25.2	94	754	0.2
2017	6	7	22	8	0.2	24	21	45	14	0.3	85	25.3	94	755	0
2017	6	7	23	9	0.3	28	22	49	16	0.4	64	25.3	93	754	0

ภาพประกอบ 8 แสดงตัวอย่างข้อมูลของ Dataset สถานีตำรวจนครบาลโชคชัย

date	PM_density	CO	NO	NO2	NOX	PM10	Wind speed	Wind dir	Temp	Rel hum	Pressure	Rain
2017-06-07 00:00:00	6.0	0.4	23.0	16.0	39.0	18.0	0.1	243.0	27.8	77.0	754.0	0.0
2017-06-07 01:00:00	7.0	0.6	37.0	15.0	52.0	22.0	0.0	167.0	27.8	80.0	754.0	0.0
2017-06-07 02:00:00	8.0	0.7	53.0	16.0	69.0	25.0	0.2	121.0	27.7	84.0	753.0	0.0
2017-06-07 03:00:00	14.0	0.6	48.0	16.0	64.0	37.0	0.2	93.0	27.1	87.0	753.0	0.0
2017-06-07 04:00:00	19.0	0.6	30.0	13.0	43.0	33.0	0.3	86.0	26.8	88.0	753.0	0.0

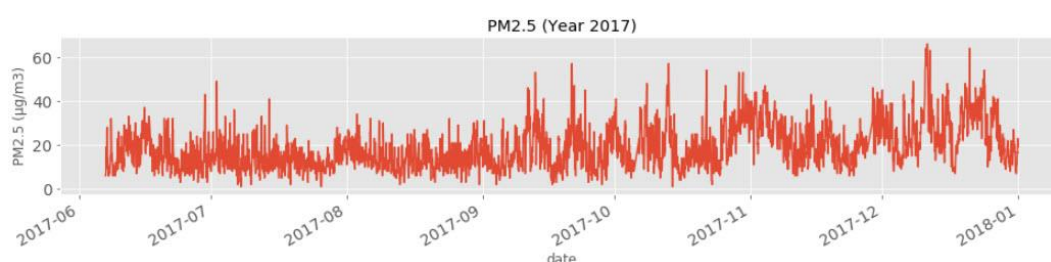
ภาพประกอบ 9 แสดงตัวอย่างข้อมูลของ Dataset สถานีตำรวจนครบาลโชคชัย ในโปรแกรม

ทำการวิเคราะห์ปริมาณของฝุ่นละออง PM2.5 ในช่วง 2 ปี ซึ่งเริ่มตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00) จะเห็นได้ว่าช่วงต้นปี 2018 ปริมาณฝุ่นละออง PM2.5 มีปริมาณที่เพิ่มสูงขึ้น ดังภาพประกอบ 10



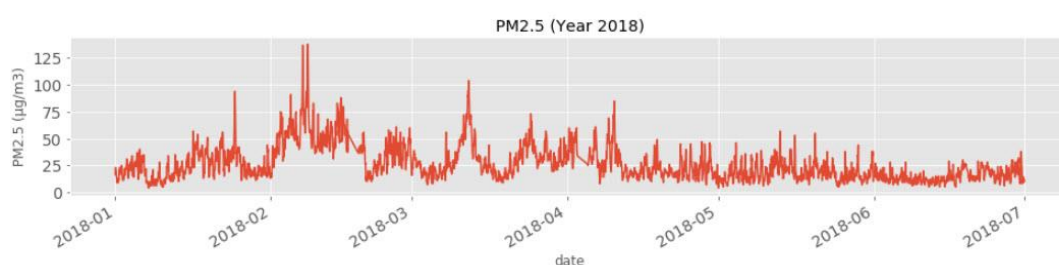
ภาพประกอบ 10 แสดงค่าฝุ่นละออง PM2.5 ช่วงปี 2017-2018

ทำการวิเคราะห์ปริมาณของฝุ่นละออง PM2.5 ในช่วงปี 2017 มีข้อมูลตั้งแต่วันที่ 07-06-2017 (เวลา 00:00) ถึง 31-12-2017 (เวลา 23:00) จำนวน 7 เดือน จะเห็นได้ว่าในช่วงปลายปีค่าปริมาณฝุ่นละออง PM2.5 ค่อย ๆ เพิ่มมากขึ้น ดังภาพประกอบ 11



ภาพประกอบ 11 แสดงค่าฝุ่นละออง PM2.5 ของปี 2017

ทำการวิเคราะห์ปริมาณของฝุ่นละออง PM2.5 ในช่วงปี 2018 มีข้อมูลตั้งแต่วันที่ 01-01-2018 (เวลา 00:00) ถึง 30-06-2018 (เวลา 23:00) จำนวน 6 เดือน จะเห็นได้ว่าในช่วงต้นปี 2018 ค่าปริมาณฝุ่นละออง PM2.5 เพิ่มมากขึ้นและก็ค่อย ๆ ลดลง ดังภาพประกอบ 12



ภาพประกอบ 12 แสดงค่าฝุ่นละออง PM2.5 ของปี 2018

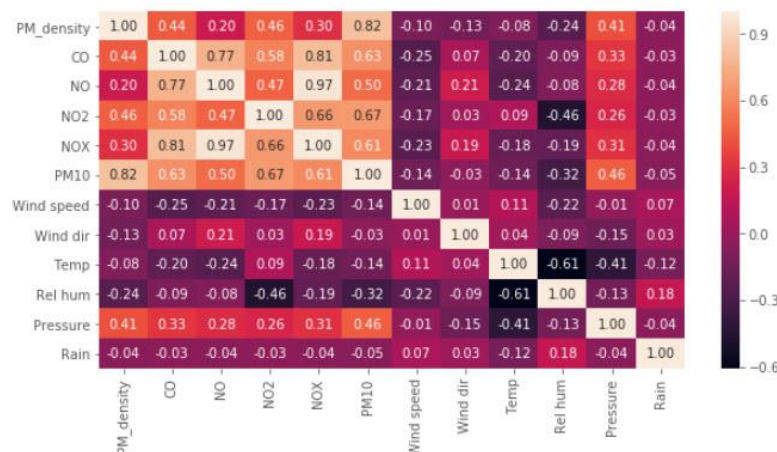
ทำการวิเคราะห์ feature ทั้งหมดที่มีอยู่ในชุดข้อมูล ดังภาพประกอบ 13 จะเห็นได้ว่าหลังจากทำความสะอาดข้อมูลเรียบร้อยแล้ว ข้อมูลที่ตอนแรกมีทั้งหมด 9,336 แถว เหลืออยู่ 8,681 แถว ซึ่ง count คือ จำนวนแถวทั้งหมด mean คือ ค่าเฉลี่ย std คือ ค่าส่วนเบี่ยงเบนมาตรฐาน min คือ ค่าที่น้อยสุด และ max คือค่าที่มากที่สุด

	PM_density	CO	NO	NO2	NOX	PM10	Wind speed	Wind dir	Temp	Rel hum	Pressure	Rain
count	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000	8681.000000
mean	22.046193	0.888388	37.338671	22.722267	59.694160	47.073609	0.508582	157.598433	28.805495	66.471029	754.945974	0.246746
std	13.666167	0.526600	40.175394	12.986972	46.939508	26.478732	0.411196	94.604541	2.908120	14.210210	2.110094	2.122963
min	1.000000	0.000000	0.000000	5.000000	7.000000	3.000000	0.000000	0.000000	17.200000	23.000000	749.000000	0.000000
25%	13.000000	0.500000	7.000000	14.000000	26.000000	28.000000	0.200000	73.000000	27.000000	56.000000	754.000000	0.000000
50%	19.000000	0.720000	22.000000	19.000000	44.000000	40.000000	0.400000	121.000000	28.700000	67.000000	755.000000	0.000000
75%	28.000000	1.100000	54.000000	28.000000	79.000000	60.000000	0.700000	253.000000	30.800000	77.000000	756.000000	0.000000
max	138.000000	5.200000	311.000000	97.000000	366.000000	196.000000	3.000000	359.000000	36.600000	98.000000	765.000000	52.800000

ภาพประกอบ 13 แสดงรายละเอียดข้อมูลทั้งหมดของแต่ละ feature

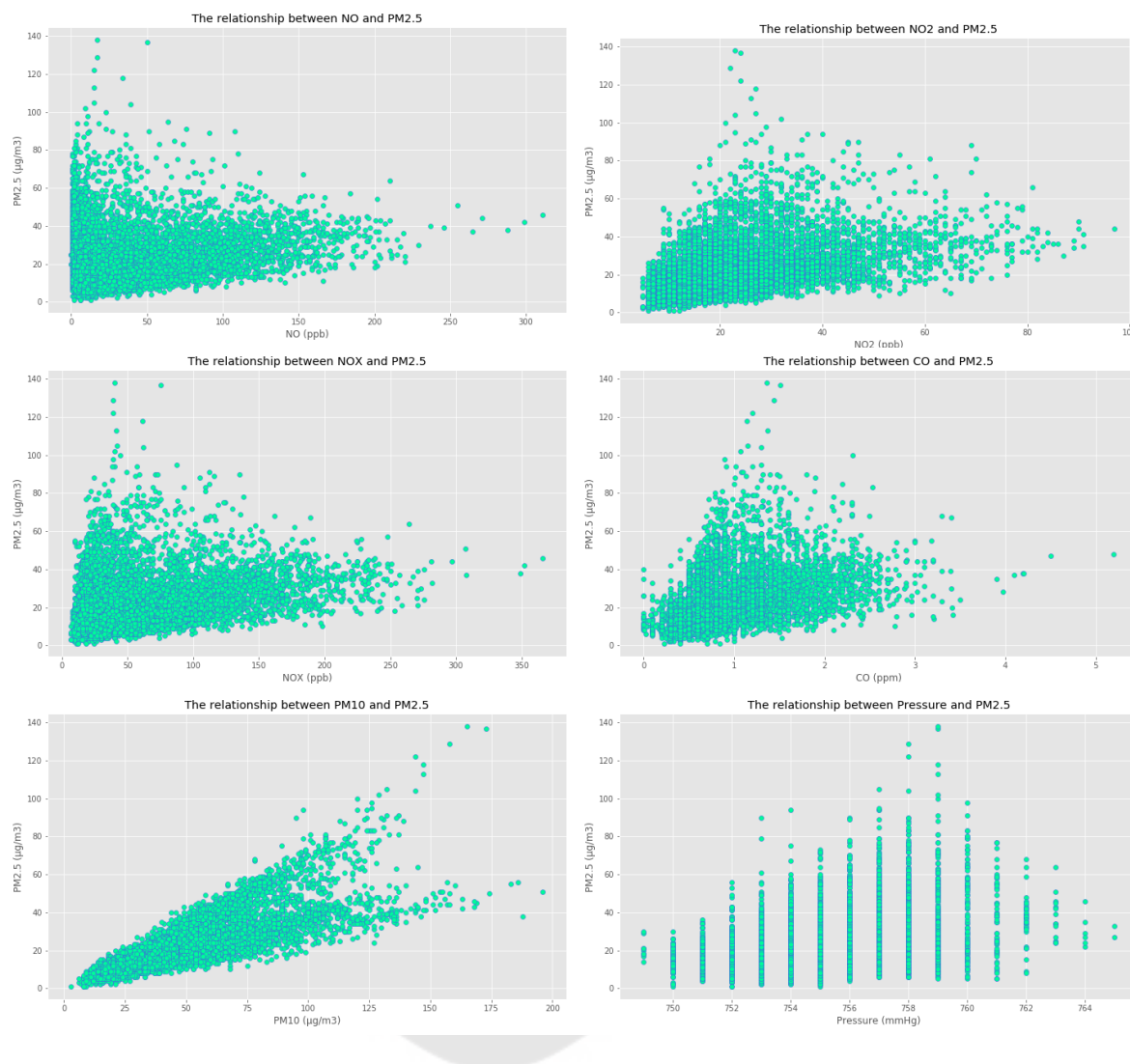
แสดงการวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลฝุ่นละออง PM2.5 กับ feature ตัวอื่น ประกอบด้วย CO, NO, NO2, NOX, PM10, Wind speed, Wind direction, Temperature, Relative humidity, Pressure, Rain จากรูปที่ 10 ถ้าเรียงลำดับความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ feature ตัวอื่นจากมากไปน้อยทางบวกจะได้ดังนี้ ฝุ่นละออง PM10 > NO2 > CO >

Pressure > NOX > NO และจากความสัมพันธ์มากไปในทางลบจะได้ Relative humidity > Wind direction > Wind speed > Temperature > Rain



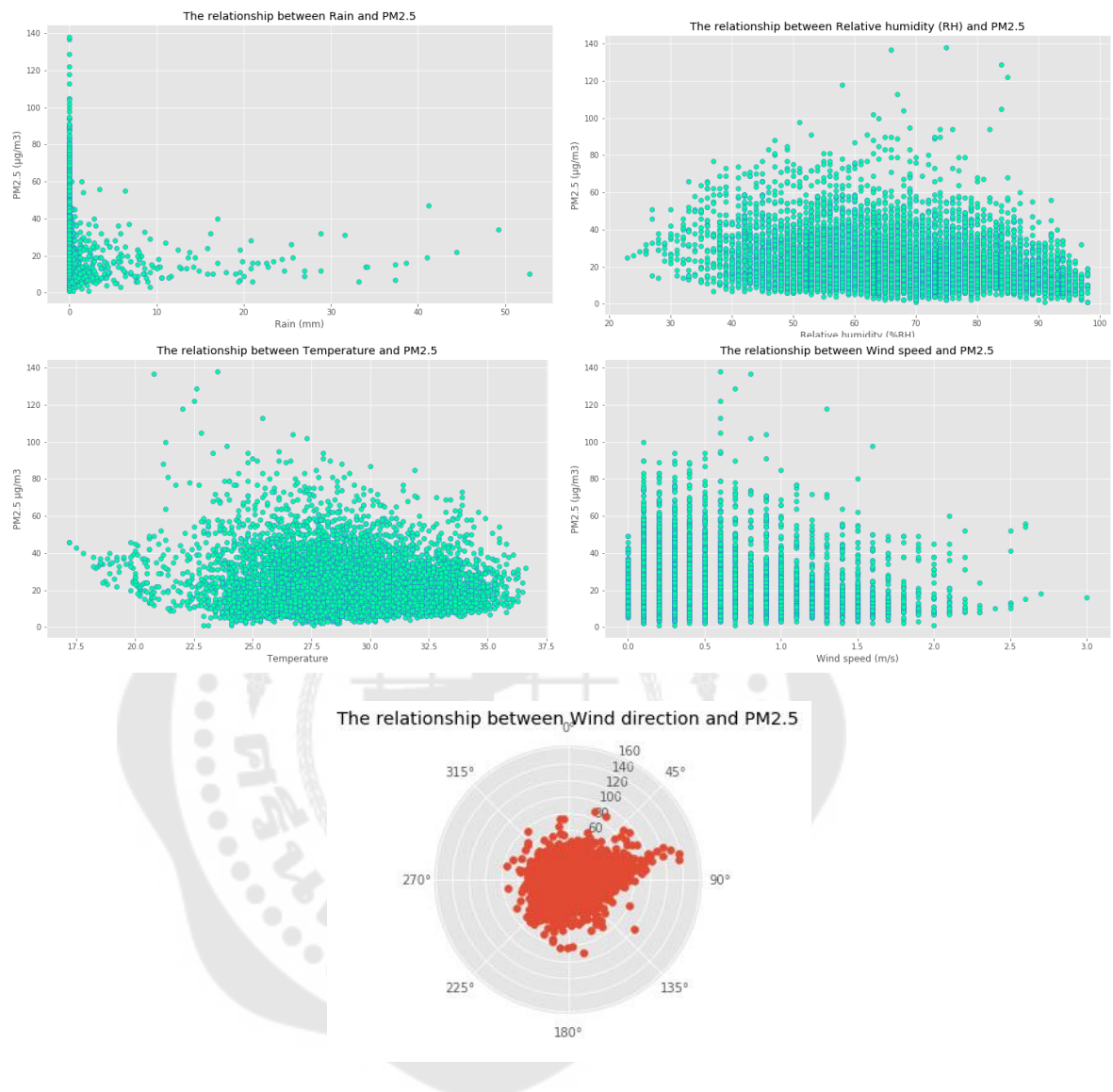
ภาพประกอบ 14 แสดงความสัมพันธ์ระหว่างข้อมูลฝุ่นละออง PM2.5 กับ feature อื่น
ในรูปแบบ Heatmap

จากภาพประกอบ 15 ซ้ายบนแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ NO (ไนตริกออกไซด์) รูปขวบนแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ NO2 (ไนโตรเจนไดออกไซด์) รูปซ้ายกลางแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ NOX (ไนโตรเจนออกไซด์) รูปขวากลางแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ CO (คาร์บอนมอนอกไซด์) และรูปซ้ายล่างแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ ฝุ่นละออง PM10 รูปขวาล่างแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ ความกดอากาศ



ภาพประกอบ 15 แสดงค่าความสัมพันธ์ระหว่าง feature อื่นกับฝุ่นละออง PM2.5

จากภาพประกอบ 16 รูปซ้ายบนแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ ปริมาณน้ำฝน รูปขวามบนแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ ค่าความชื้นสัมพัทธ์ใน อากาศ รูปซ้ายล่างแสดงความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ อุณหภูมิ รูปขวาล่างแสดง ความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ ความเร็วลม และรูปสุดท้ายแสดงความสัมพันธ์ ระหว่างฝุ่นละออง PM2.5 กับ ทิศทางลม จะเห็นได้ว่าทิศทางลมนั้นไม่ได้ส่งผลกับปริมาณค่าฝุ่น ละออง PM2.5 เท่าไรนัก



ภาพประกอบ 16 แสดงค่าความสัมพันธ์ระหว่าง feature อื่นกับฝุ่นละออง PM2.5 (ต่อ)

จากภาพประกอบ 14 - 16 นั้นได้ทำการศึกษาและวิเคราะห์หาความสัมพันธ์ระหว่างข้อมูลฝุ่นละออง PM2.5 กับ feature อื่น เพื่อที่จะได้ทำการเลือก feature ดังกล่าวมาใส่เป็นข้อมูล input ในแบบจำลอง เพื่อช่วยเพิ่มประสิทธิภาพในการทำนายปริมาณฝุ่นละออง PM2.5 ในอนาคตให้มีความแม่นยำมากขึ้น ซึ่งจากภาพประกอบจะเห็นได้ว่าแต่ละ feature นั้นมีความสัมพันธ์กับข้อมูลฝุ่นละออง PM2.5 ทั้งหมด ไม่ว่าจะเป็นทางบวกและทางลบ เช่น ในภาพประกอบ 16 ตัวอย่างภาพที่แสดงความสัมพันธ์ระหว่างความเร็วลมกับฝุ่นละออง PM2.5 จากภาพจะเห็นได้ว่าแนวโน้ม

ของปริมาณฝุ่นละออง PM2.5 ดูเพิ่มสูงขึ้น เมื่อความเร็วลมนั้นมีค่าน้อย และเมื่อความเร็วลมมีค่าเพิ่มสูงขึ้น ปริมาณฝุ่นละออง PM2.5 ดูจะมีแนวโน้มลดลง เป็นต้น ดังนั้นในการวิจัยนี้จึงได้เลือกใช้ feature ทั้งหมดมาใช้เป็นข้อมูล input เพื่อช่วยเพิ่มประสิทธิภาพของแบบจำลองในการทำนายปริมาณฝุ่นละออง PM2.5 ล่วงหน้าให้มีความแม่นยำมากยิ่งขึ้น

อัลกอริทึมและแบบจำลองของการทำนาย

ในงานวิจัยนี้ได้ใช้แบบจำลอง LSTM และ SARIMAX เพื่อทำการเปรียบเทียบสมรรถนะโดยใช้ค่า Root Mean Squared Error (RMSE) และ Mean Absolute Error (MAE) เป็นตัวชี้วัดความแม่นยำของแต่ละแบบจำลอง ซึ่งข้อมูลใน Dataset สถานีตำรวจนครบาลโชคชัย มี feature ทั้งหมด 16 feature จำนวน 9,336 แถว เก็บข้อมูลเป็นรายชั่วโมงตั้งแต่วันที่ 07/06/2017 – 30/06/2018 โดยแบ่งข้อมูลเป็น train 90% และ test 10% ดังนี้

- Train ใช้ข้อมูลช่วง 07-06-2017 00:00:00 ถึง 22-05-2018 23:00:00 เป็นระยะเวลา 350 วัน จำนวน 8,400 ชั่วโมง

- Test ใช้ข้อมูลช่วง 23-05-2018 00:00:00 ถึง 30-06-2018 23:00:00 เป็นระยะเวลา 39 วัน จำนวน 936 ชั่วโมง

ซึ่งจากการวิเคราะห์ความสัมพันธ์ระหว่างฝุ่นละออง PM2.5 กับ feature อื่นเรียบร้อยแล้ว จะเห็นได้ว่า feature อื่นนั้นมีความสัมพันธ์กับฝุ่นละออง PM2.5 ทั้งในทางบวกและทางลบ ดังนั้นในการวิจัยนี้จึงใช้ข้อมูล feature อื่นทั้งหมดเป็นข้อมูล input เพื่อช่วยเพิ่มประสิทธิภาพในการทำนายค่าความแม่นยำในแบบจำลอง LSTM และแบบจำลอง SARIMAX

แบบจำลอง LSTM :

ในการทำนายค่าปริมาณความหนาแน่นของ PM2.5 นั้นใช้ทั้งหมด 12 feature ที่ได้จากการทำการวิเคราะห์ค่าความสัมพันธ์ก่อนหน้า ซึ่งรวม PM2.5 มาใช้ในการทำนายด้วย โดยข้อมูลดังกล่าวจะใช้ฟังก์ชัน shift เข้ามาช่วยในการขยับข้อมูลย้อนหลังหรือขยับข้อมูลไปข้างหน้าดังตัวอย่างภาพประกอบ 17 ซึ่งข้อมูล input ที่ใช้ในแบบจำลองนี้เป็นข้อมูลที่ถูกรับค่าย้อนหลังหรือก็คือใช้ข้อมูลของวันก่อนหน้าเพื่อใช้ในการทำนาย โดยในการวิจัยนี้ได้กำหนดค่า lag observation เป็น 7 วัน (168 ชั่วโมง) ส่วนข้อมูล output หรือ forecast time คือข้อมูลที่ต้องการจะทำนายนั้นได้กำหนดค่า ahead เป็น 24 ชั่วโมง เพราะต้องการจะหาค่าความหนาแน่นของ PM2.5 ในอีก 24 ชั่วโมงข้างหน้า

t-...	NO (t-1)	NO2 (t-1)	PM2.5 (t-1)	PM2.5 (t)	PM2.5 (t+1)	PM2.5 (t+...)
....	23	16	12	6	7	
....	37	15	6	7	8	
t-168 (lasts 7 days) ...	53	16	7	8	14	...
....	48	16	8	14	19	...
....	30	13	14	19	11	...
....	24	12	19	11	20	...

... PM2.5 (t+24)

ภาพประกอบ 17 แสดงภาพตัวอย่างการ shift ข้อมูล เพื่อใช้ในการทำนาย

หลังจากใช้ฟังก์ชัน shift แล้วจะได้ข้อมูลออกมาเป็น t-n, t และ t+n ซึ่งข้อมูลที่เป็น t-n นั้นเป็นข้อมูลที่ถูก shift ย้อนหลัง เช่น Rain Temp Wind dir เป็นต้น เพื่อที่จะนำข้อมูลในช่วงเวลาดังกล่าว รวมถึง t ซึ่งเป็นเวลาปัจจุบันมาเป็น input ของแบบจำลอง และสุดท้าย t+n เป็นข้อมูล output ที่ต้องการจะทำนายล่วงหน้าอย่าง PM_density (PM2.5) ซึ่งข้อมูลดังกล่าวจะถูก shift ไปข้างหน้า ดังตัวอย่างภาพประกอบ 18

Wind speed(t-1)	Wind dir(t-1)	Temp(t-1)	Rel hum(t-1)	Pressure(t-1)	Rain(t-1)	PM_density(t-1)	PM_density(t)	PM_density(t+1)	PM_density(t+2)	PM_density(t+3)
0.2	201	33.7	46	755	0	23	14	10	21	15
0.4	93	33	48	754	0	14	10	21	15	17
0.8	70	31.9	53	754	0	10	21	15	17	14
0.5	75	30.2	58	754	0	21	15	17	14	11
0.5	54	30.1	62	755	0	15	17	14	11	14
0.1	321	29.9	64	755	0	17	14	11	14	15
0.3	279	29.9	64	756	0	14	11	14	15	25
0.4	278	30	66	756	0	11	14	15	25	20
0.8	269	30.1	62	756	0	14	15	25	20	18
0.9	258	29.9	55	756	0	15	25	20	18	16
0.5	251	29.5	55	756	0	25	20	18	16	29
0.3	246	29.1	57	756	0	20	18	16	29	31
0.6	259	28.7	59	755	0	18	16	29	31	26
0.7	256	28.2	63	755	0	16	29	31	26	23
0.8	258	28.2	63	756	0	29	31	26	23	27
0.6	258	30.3	52	757	0	31	26	23	27	30
1.4	260	31.1	48	757	0	26	23	27	30	32

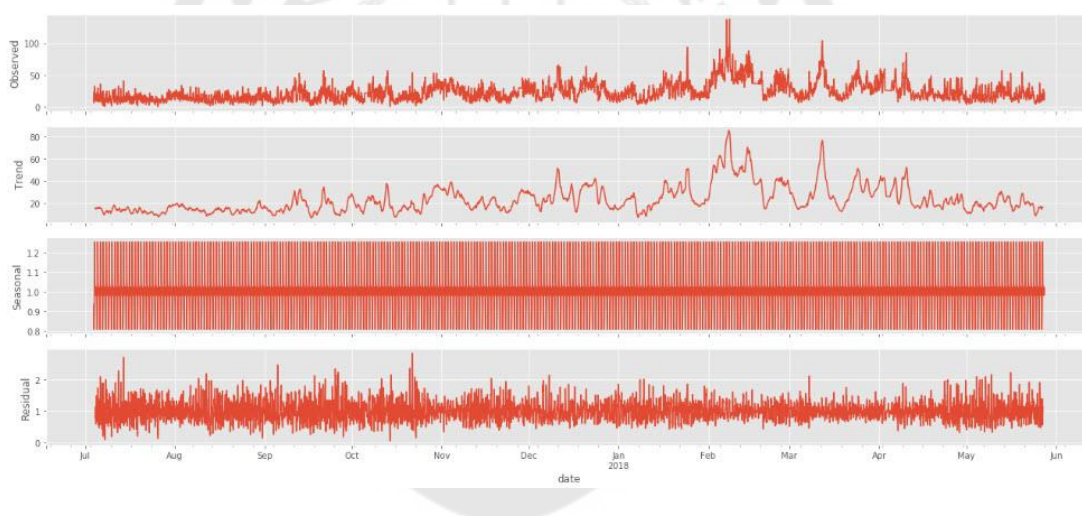
ภาพประกอบ 18 แสดงภาพตัวอย่างข้อมูลที่ได้จากการใช้ฟังก์ชัน shift ในการขยับข้อมูล

จากนั้นทำการแบ่งข้อมูลออกเป็น 2 ชุด คือ train และ test แล้วจึงแบ่งชุดข้อมูลของ train และ test ออกเป็นข้อมูลที่จะใช้ในการ input และ output ของแบบจำลอง โดยแปลงค่าของชุดข้อมูลให้อยู่ในรูปแบบของอาร์เรย์ 3 มิติ ซึ่งถูกแบ่งออกเป็น [sample, timesteps, feature] เพื่อใช้ในแบบจำลอง LSTM แล้วทำการใช้ Grid search ในการ tuning parameter เพื่อหาค่า epochs hidden neuron และ batch size ที่ดีที่สุด ซึ่งได้ค่า epochs = 50 ค่า hidden neuron (n_neurons) = 100 และค่า batch size (n_batch) = 27 โดยได้กำหนดค่าของ Optimizer เป็น RMSprop เพื่อ

เพิ่มประสิทธิภาพในการช่วยทำนายผลของแบบจำลอง แล้วจึงทำการ fit ข้อมูลกับแบบจำลอง เพื่อให้แบบจำลองนั้นทำการทำนายหาค่าความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมง

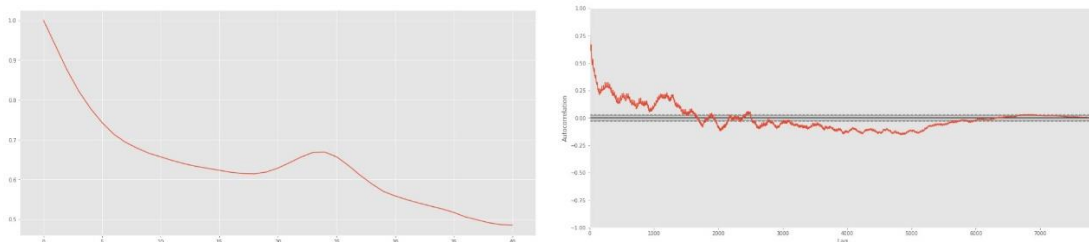
แบบจำลอง SARIMAX :

แบบจำลอง SARIMAX นั้นมีวิธีการทำนายที่แตกต่างกับแบบจำลอง LSTM โดยหลักการทำงานของ SARIMAX นั้นจะนำข้อมูลที่ต้องการจะทำนายอย่างเดียวนำมาใช้ในการ train ดังนั้นเพื่อให้สามารถเปรียบเทียบกับ LSTM ได้ จึงใช้วิธีการ 'ขยับทีละชั่วโมง' และทำการเก็บค่าที่ได้จากการทำนายของการ train ในแต่ละครั้ง โดยกำหนดค่า lag observation เป็น 7 วัน (168 ชั่วโมง) และ forecast time เป็น 24 ชั่วโมงเหมือนกับแบบจำลอง LSTM จากนั้นนำข้อมูลฝุ่นละออง PM2.5 ที่ทำการแบ่งเป็น train set มาตรวจสอบดูค่า Trend Seasonal และ Residual จากภาพประกอบ 19 จะเห็นได้ว่าข้อมูลฝุ่นละออง PM2.5 นั้นไม่เป็น Trend แต่มีความเป็น Seasonal



ภาพประกอบ 19 แสดงค่า Trend Seasonal และ Residual ของ train data

ทำการหาความสัมพันธ์ (correlation) ระหว่างข้อมูลที่ต้องการจะศึกษากับข้อมูลก่อนหน้าว่ามีค่า lag เป็นเท่าไร โดยใช้ Autocorrelation Function (ACF) เพื่อที่จะนำเอาค่าที่ได้มาเป็นตัวกำหนดในตัวแปร s ในแบบจำลอง SARIMAX ซึ่งจากภาพประกอบ 20 ทางด้านซ้ายจะเห็นได้ว่ากราฟสูงขึ้นตรง lag = 24 ดังนั้นจึงกำหนดให้ $s = 24$



ภาพประกอบ 20 แสดงกราฟ ACF ที่พล็อตค่า lag แบบช่วงเวลายสั้นกับช่วงเวลายาว

จากนั้นใช้ Grid Search ในการ Tuning Parameter เพื่อหาค่า p, d, q ที่ดีที่สุด ซึ่งดูจากค่า Akaike Information Criterion (AIC) โดยกำหนดให้ $p = \text{range}(0, 4)$, $d = \text{range}(0, 2)$, $q = \text{range}(0, 4)$ และ $s = 24$ (รายชั่วโมง) ดังตัวอย่างภาพประกอบด้านล่าง

$(p, d, q) \times (P, D, Q, s)$	AIC
$(0, 0, 0) \times (0, 0, 0, 24)$	6989.0920
$(0, 0, 0) \times (0, 0, 1, 24)$	6264.5451
$(0, 0, 0) \times (0, 0, 2, 24)$	5641.1697
.....	
.....	
$(3, 1, 3) \times (3, 1, 3, 24)$	4224.0026

ภาพประกอบ 21 แสดงตัวอย่างค่าการ Tuning Parameter

ซึ่งได้ค่าพารามิเตอร์ออกมา คือ SARIMAX(3, 1, 3)x(3, 1, 3, 24) มีค่า AIC น้อยสุด และนำค่าที่ได้ไป fit กับแบบจำลอง SARIMAX ซึ่ง Akaike Information Criterion (AIC) เป็นตัววัดว่าข้อมูลที่ใช้ fit กับแบบจำลองได้ดีมากน้อยแค่ไหน ซึ่งแบบจำลองที่ fit กับข้อมูลได้ดีนั้นจะต้องใช้ feature จำนวนมาก ซึ่งจะทำให้ค่า AIC score ที่ออกมามีตัวเลขที่ค่อนข้างมากกว่าแบบจำลองที่ใช้ข้อมูลน้อย ๆ ในท้ายที่สุดแบบจำลองที่มีค่า AIC น้อยสุดก็คือแบบจำลองที่ดีที่สุด จากนั้นทำการเพิ่ม Exogenous variable (exog) เข้าไปในแบบจำลอง โดยกำหนดตัวแปรเป็น exo_data ประกอบด้วยคอลัมน์ CO, NO, NO2, NOX, PM10, Wind speed, Wind dir, Temp, Rel hum, Pressure และ Rain

ทำการทำนายค่าฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า โดยค่อย ๆ ขยับจำนวน ชั่วโมงไปข้างหน้าทีละหนึ่งชั่วโมงจนหมดข้อมูล test ที่แบ่งไว้ในตอนแรกและใช้ exogenous variable (exog) ของ 24 ชั่วโมงก่อนหน้าเข้าไปช่วยในการทำนายแบบจำลอง แล้วจึงทำการเก็บ ข้อมูลที่ได้จากการเปรียบเทียบกับ test (ข้อมูลจริง) และนำค่าที่ได้จากการทำนายค่าฝุ่นละออง PM2.5 ที่ค่อย ๆ ขยับทีละชั่วโมงทั้งหมดมาหาค่า RMSE และ MAE ของชั่วโมงที่ 1 ไปจนถึง ชั่วโมงที่ 24 ดังตัวอย่างภาพตารางประกอบด้านล่าง

year	month	day	hour	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_10	t_11	t_12	t_13	t_14
2018	5	26	0	9.896963	7.502174	20.33516	20.45465	14.84932	31.69565	23.32079	11.79496	12.89776	13.52439	20.25918	14.1781	14.71989	14.3611
2018	5	26	1	6.842603	19.58086	19.76162	14.42729	31.01231	23.10062	11.76268	12.6495	13.51583	20.20782	13.97639	14.50737	14.06868	13.09905
2018	5	26	2	6.842603	19.58086	19.76162	14.42729	31.01231	23.10062	11.76268	12.6495	13.51583	20.20782	13.97639	14.50737	14.06868	13.09905
2018	5	26	3	20.1983	14.88896	31.457	23.18107	11.89593	12.98664	13.71507	20.17099	14.22514	14.94122	14.42168	13.4726	13.10345	12.64077
2018	5	26	4	18.75216	34.95306	26.74356	15.49286	16.41725	17.45395	23.96575	17.86149	18.58803	18.11836	16.96114	16.54937	15.95467	19.95235
2018	5	26	5	37.89213	30.00369	18.52957	19.15	20.15217	26.85844	20.29567	21.23164	20.9785	19.47564	19.13482	18.28748	22.22974	18.91279
2018	5	26	6	32.70056	21.79647	22.69072	23.84264	31.14468	24.18431	24.47629	24.1202	22.48899	21.86425	20.94972	24.90872	22.4309	27.10725
2018	5	26	7	25.97648	26.85145	28.39177	35.50135	28.69214	28.49765	27.97071	26.48077	25.68866	24.96631	28.64272	26.60821	31.10781	33.65357
2018	5	26	8	22.35803	23.71304	31.01659	24.3655	24.41397	23.61946	22.07939	20.99939	20.26848	23.42972	21.55652	26.43112	28.80921	25.54374
2018	5	26	9	21.93606	29.78755	23.25713	23.14255	22.38748	20.86077	19.80026	19.11601	22.09487	20.34003	24.88988	27.39654	24.21219	21.92297
2018	5	26	10	29.83157	23.28863	23.17038	22.48493	20.79968	19.62354	19.00418	21.9623	20.14076	24.71728	27.10909	23.95135	21.74336	24.21172
2018	5	26	11	22.25973	22.50496	21.7191	20.27957	19.20602	18.62766	21.34824	19.50051	23.8504	26.63829	23.47252	21.23416	23.84318	26.22048
2018	5	26	12	24.31537	23.50556	21.75697	20.5727	20.30148	22.96366	21.41832	25.32924	27.94229	24.70833	22.67035	25.44336	27.66546	26.50773
2018	5	26	13	23.62692	21.8467	20.68156	20.3666	23.2684	21.56952	25.53695	28.15733	24.82504	22.74134	25.58174	27.90683	26.90897	27.94603
2018	5	26	14	20.1562	19.30373	18.65627	21.26437	19.57702	24.11632	26.87998	23.7689	21.42501	24.10549	26.60517	25.65487	26.74486	19.18665
2018	5	26	15	18.27467	17.66977	20.57339	18.75103	23.3922	25.92994	22.82077	20.61835	23.07237	25.50719	24.78182	25.91204	18.38316	22.63478
2018	5	26	16	19.12356	21.64473	19.9034	24.07247	26.83974	23.69994	21.54755	24.31229	26.7248	25.65683	26.89877	19.62398	23.84462	21.39674
2018	5	26	17	19.69039	18.22114	22.4248	25.0884	21.81084	19.61307	22.63308	25.12957	23.83884	25.11164	17.81524	21.79647	19.49733	23.35919
2018	5	26	18	15.84951	20.5612	23.25219	20.13815	18.03618	20.7536	23.21074	22.22577	23.36597	15.9806	20.36589	18.02771	21.88752	25.17976
2018	5	26	19	21.39169	23.91653	20.76419	18.66021	21.32444	23.86342	22.83498	23.91625	16.68835	20.8346	18.5742	22.33124	25.51141	19.74479
2018	5	26	20	21.75718	18.65308	16.728	19.84941	22.17038	21.20329	22.25677	15.4596	19.19265	17.05741	20.88991	23.86417	17.94104	14.29399
2018	5	26	21	14.17951	12.85575	15.46877	17.82698	16.65448	17.73371	11.10146	15.09878	12.91361	16.93438	19.68332	13.98396	10.28522	9.679944

ภาพประกอบ 22 แสดงภาพตัวอย่างข้อมูลที่ได้จากการทำนายโดยขยับทีละหนึ่งชั่วโมง

บทที่ 4

ผลการดำเนินการวิจัย

ในการวิจัยการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า ซึ่งใช้ข้อมูลฝุ่นละออง PM2.5 และข้อมูลทางอุตุนิยมวิทยาบริเวณพื้นที่สถานีตำรวจนครบาลโชคชัย โดยใช้เทคนิคการเรียนรู้ของเครื่อง ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษาตามขอบข่ายและขั้นตอนต่าง ๆ ตลอดจนการวัดประสิทธิภาพ เพื่อให้บรรลุจุดประสงค์ของการวิจัยที่ได้กำหนดไว้ ได้ดังนี้

1. ผลลัพธ์ของการสร้างแบบจำลอง LSTM
2. ผลลัพธ์ของการสร้างแบบจำลอง SARIMAX
3. ผลลัพธ์จากการเปรียบเทียบการทำนายความหนาแน่นของฝุ่นละออง PM2.5

ระหว่างแบบจำลอง LSTM และ SARIMAX

ผลลัพธ์ของการสร้างแบบจำลอง LSTM

จากการสร้างแบบจำลอง LSTM โดยใช้ข้อมูลที่ประกอบด้วยฝุ่นละออง PM2.5 และข้อมูลอุตุนิยมวิทยาจากสถานีตำรวจนครบาลโชคชัย นั้นมีการกำหนดข้อมูลที่ใช้อย่อนหลังเป็นจำนวน 7 วันหรือ 168 ชั่วโมง และข้อมูลฝุ่นละออง PM2.5 ที่ต้องการจะทำนายในอีก 24 ชั่วโมงข้างหน้า ซึ่งใช้ RMSE และ MAE เป็นตัววัดประสิทธิภาพของแบบจำลอง LSTM จากผลลัพธ์ของการทำนายจะเห็นได้ว่าในช่วงชั่วโมงแรก ๆ นั้นค่า RMSE และ MAE มีค่าออกมาค่อนข้างต่ำ ต่างกับช่วงชั่วโมงหลัง ๆ จะเห็นได้ว่า RMSE และ MAE มีค่าสูงขึ้น ซึ่งค่าดังกล่าวสามารถอธิบายได้ว่าผลลัพธ์จากการทำนายของแบบจำลองในช่วงชั่วโมงแรก ๆ นั้นมีความแม่นยำค่อนข้างดีกว่าค่าความแม่นยำของช่วงชั่วโมงหลัง ๆ ดังตัวอย่างตารางแสดงผลของการทำนายฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าด้านล่าง

ตาราง 2 ผลการทำนายค่าฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าโดยใช้แบบจำลอง LSTM

Time Step	แบบจำลอง LSTM	
	RMSE	MAE
t+1	3.11	2.36
t+2	4.14	3.16
t+3	4.87	3.74
t+4	5.36	4.11
t+5	5.60	4.36
t+6	5.84	4.55
t+7	5.98	4.70
t+8	6.09	4.80
t+9	6.48	5.29
t+10	6.32	5.07
t+11	6.47	5.24
t+12	6.44	5.20
t+13	6.69	5.56
t+14	6.52	5.25
t+15	6.46	5.26
t+16	6.42	5.27
t+17	6.52	5.37
t+18	6.45	5.24
t+19	6.71	5.54
t+20	6.60	5.46
t+21	6.30	5.08
t+22	6.59	5.50
t+23	6.54	5.44
t+24	6.37	5.20
Average	6.04	4.86

ผลลัพธ์ของการสร้างแบบจำลอง SARIMAX

จากการสร้างแบบจำลอง SARIMAX โดยใช้ข้อมูลของฝุ่นละออง PM2.5 อย่างเดียวในการ train และใช้ตัวแปร exogenous variable ซึ่งประกอบด้วย CO, NO, NO2, NOX, PM10, Wind speed, Wind dir, Temp, Rel hum, Pressure และ Rain มาช่วยเพิ่มประสิทธิภาพในการทำนายค่าความหนาแน่นของฝุ่นละออง PM2.5 โดยใช้ RMSE และ MAE เป็นตัววัดประสิทธิภาพของแบบจำลอง จากผลการทดลองจะเห็นได้ว่าค่า RMSE และ MAE ของแบบจำลอง SARIMAX นั้นมีค่าความแม่นยำในช่วงชั่วโมงแรก ๆ ค่อนข้างต่ำและเพิ่มสูงขึ้นในช่วงชั่วโมงหลัง ๆ เหมือนกับแบบจำลอง LSTM แต่ผลลัพธ์ที่ได้จากแบบจำลอง SARIMAX นั้นมีค่าค่อนข้างสูงกว่าผลลัพธ์ที่ได้จากการทดลองของแบบจำลอง LSTM ซึ่งสังเกตได้จากค่า RMSE และ MAE ของแต่ละชั่วโมง ดังตาราง 3

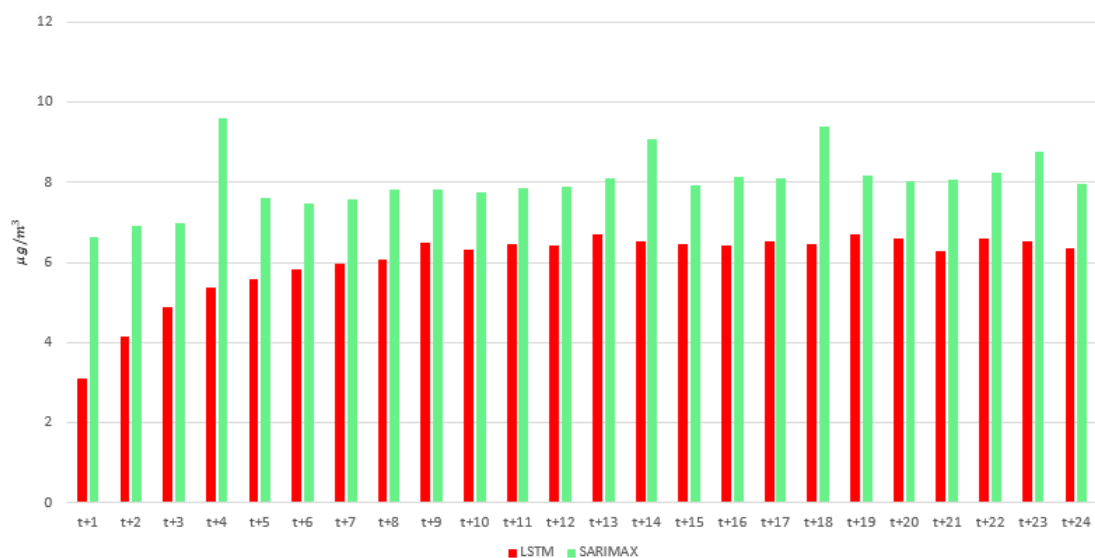
ตาราง 3 ผลการทำนายค่าฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าโดยใช้แบบจำลอง SARIMAX

Time Step	แบบจำลอง SARIMAX	
	RMSE	MAE
t+1	6.65	5.13
t+2	6.91	5.25
t+3	6.98	5.31
t+4	9.59	5.86
t+5	7.63	5.83
t+6	7.46	5.61
t+7	7.58	5.66
t+8	7.83	5.88
t+9	7.82	5.84
t+10	7.76	5.76
t+11	7.87	5.80
t+12	7.88	5.69
t+13	8.10	5.89
t+14	9.07	6.10

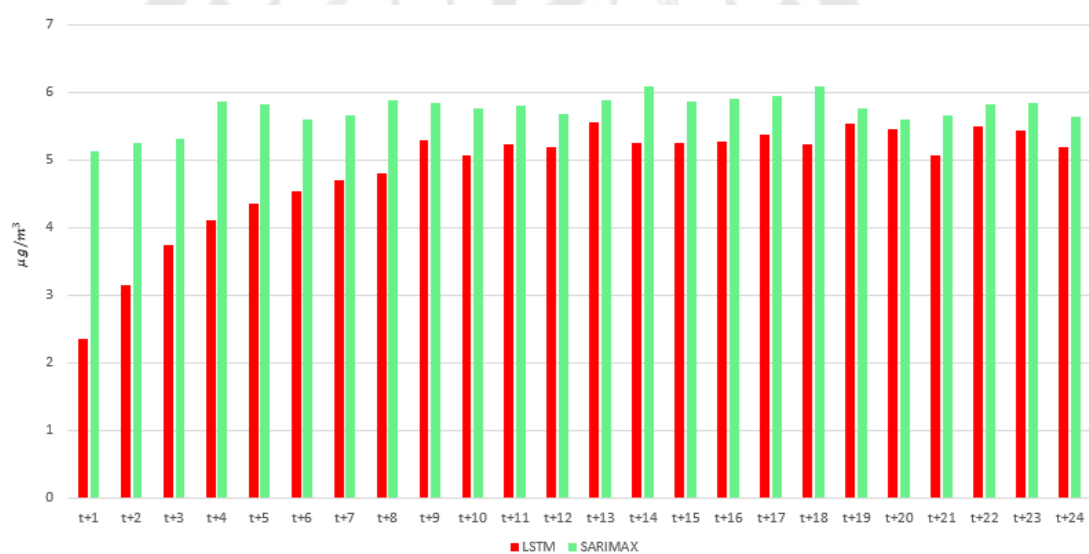
ตาราง 3 (ต่อ)

Time Step	แบบจำลอง SARIMAX	
	RMSE	MAE
t+15	7.93	5.86
t+16	8.13	5.91
t+17	8.12	5.94
t+18	9.40	6.09
t+19	8.17	5.77
t+20	8.04	5.61
t+21	8.08	5.67
t+22	8.25	5.83
t+23	8.77	5.85
t+24	7.96	5.64
Average	7.99	5.74

ทำการพล็อตกราฟค่า RMSE กับ MAE ตั้งแต่ชั่วโมงที่ t+1 ถึง t+24 เปรียบเทียบระหว่างแบบจำลอง LSTM กับ SARIMAX เพื่อให้เห็นประสิทธิภาพของทั้งสองแบบจำลองเพิ่มมากขึ้น จากภาพประกอบ 23 และ 24 กราฟแท่งแสดงค่า RMSE และ MAE ซึ่งกราฟแท่งสีแดงนั้นเป็นของแบบจำลอง LSTM และกราฟแท่งสีเขียวอ่อนนั้นเป็นของแบบจำลอง SARIMAX สำหรับ RMSE และ MAE ที่เป็นตัววัดประสิทธิภาพของแบบจำลองแบบ Regression หากค่าที่ได้ออกมานั้นยิ่งต่ำมากหรือเข้าใกล้ศูนย์มากเท่าไร นั้นหมายความว่าแบบจำลองมีประสิทธิภาพในการทำนายผลออกมาได้มีความแม่นยำมากขึ้นเท่านั้น โดยวัดจากการนำค่าทำนายที่ได้ไปเทียบกับค่าจริง ซึ่งจะเห็นได้ว่าค่า RMSE และค่า MAE ของแบบจำลอง LSTM นั้นมีค่าต่ำกว่าของแบบจำลอง SARIMAX ซึ่งหมายความว่าแบบจำลอง LSTM นั้นมีประสิทธิภาพในการทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ล่วงหน้าได้มีความแม่นยำกว่าแบบจำลอง SARIMAX



ภาพประกอบ 23 กราฟแสดงค่า RMSE เปรียบเทียบระหว่างแบบจำลอง LSTM กับ SARIMAX

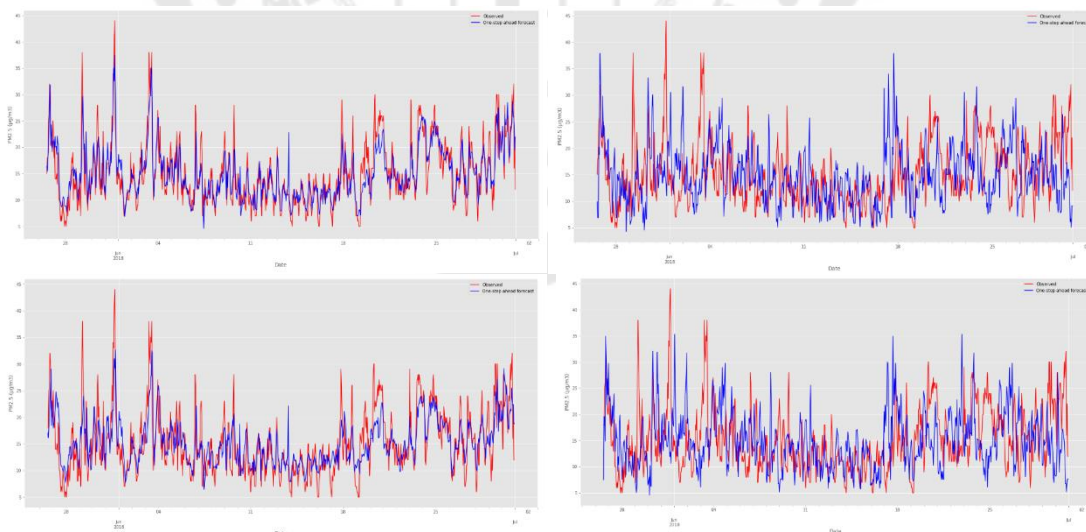


ภาพประกอบ 24 กราฟแสดงค่า MAE เปรียบเทียบระหว่างแบบจำลอง LSTM กับ SARIMAX

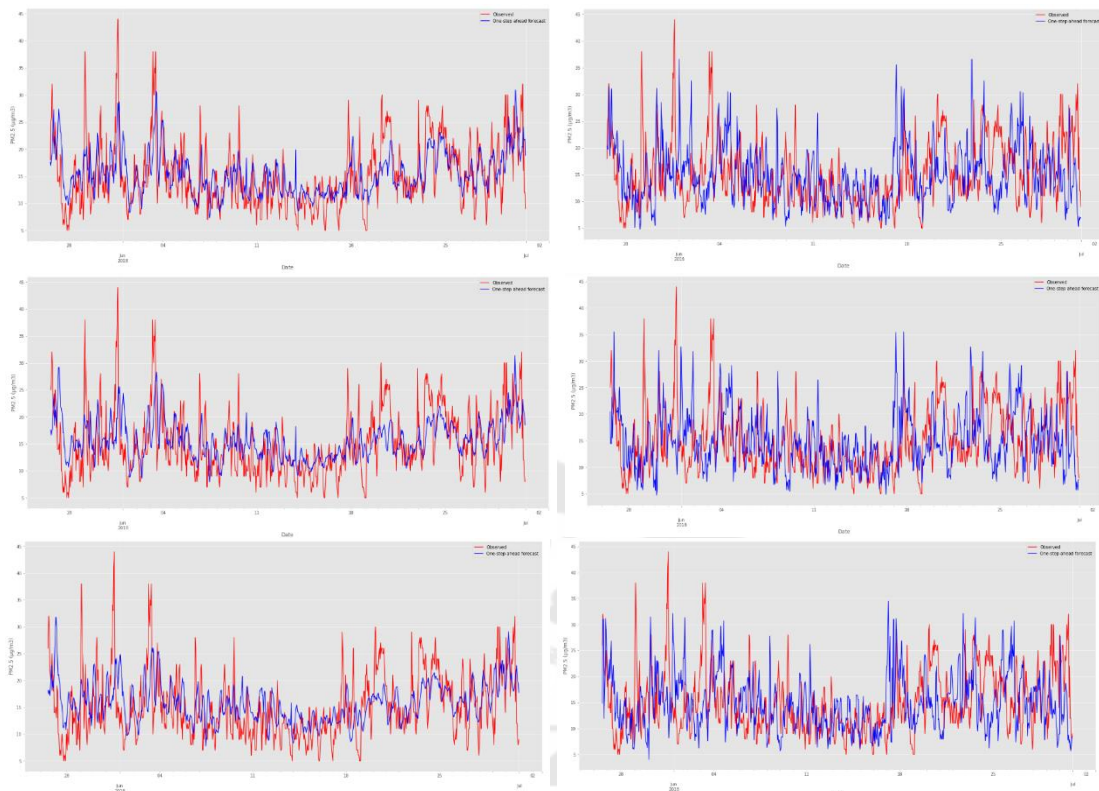
ผลลัพธ์จากการเปรียบเทียบการทำนายความหนาแน่นของฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX

ผลลัพธ์ที่ได้จากการทดลองของแบบจำลอง LSTM และ SARIMAX สามารถนำค่าที่ได้จากการทำนายมาพล็อตกราฟเปรียบเทียบผลลัพธ์ของแต่ละชั่วโมง โดยเริ่มตั้งแต่ชั่วโมงที่ 1 จนถึงชั่วโมงที่ 24 ดังนี้

จากภาพประกอบ 25 และ 26 นั้นแสดงผลการทำนายล่วงหน้า 1 ชั่วโมงถึง 5 ชั่วโมงโดยเรียงลำดับจากบนลงล่างระหว่างแบบจำลอง LSTM และ SARIMAX โดยเส้นกราฟสีน้ำเงินคือกราฟที่พล็อตจากค่าที่ได้จากการทำนายของแบบจำลอง ส่วนเส้นกราฟสีแดงนั้นคือกราฟที่พล็อตจากค่าจริง ซึ่งรูปทางด้านซ้ายนั้นเป็นรูปกราฟที่ถูกพล็อตมากจากการทำนายของแบบจำลอง LSTM ส่วนรูปทางด้านขวานั้นเป็นรูปกราฟที่ถูกพล็อตมากจากการทำนายของแบบจำลอง SARIMAX จากรูปจะเห็นได้ว่ากราฟของแบบจำลอง LSTM เมื่อนำมาพล็อตเทียบกับค่าจริงแล้วเส้นกราฟดูมีความแม่นยำมากกว่าเส้นกราฟของแบบจำลอง SARIMAX ได้อย่างชัดเจน ซึ่งในชั่วโมงที่ 1 - 2 นั้นจะเห็นได้ว่าเส้นกราฟที่ถูกพล็อตของแบบจำลอง LSTM นั้นดูแม่นยำที่สุดและตั้งแต่ชั่วโมงที่ 3 ขึ้นไป ค่าความแม่นยำนั้นค่อย ๆ ลดลงโดยสังเกตได้จากเส้นกราฟ เช่นเดียวกับเส้นกราฟที่ถูกพล็อตจากการทำนายของแบบจำลอง SARIMAX ดังภาพประกอบด้านล่าง

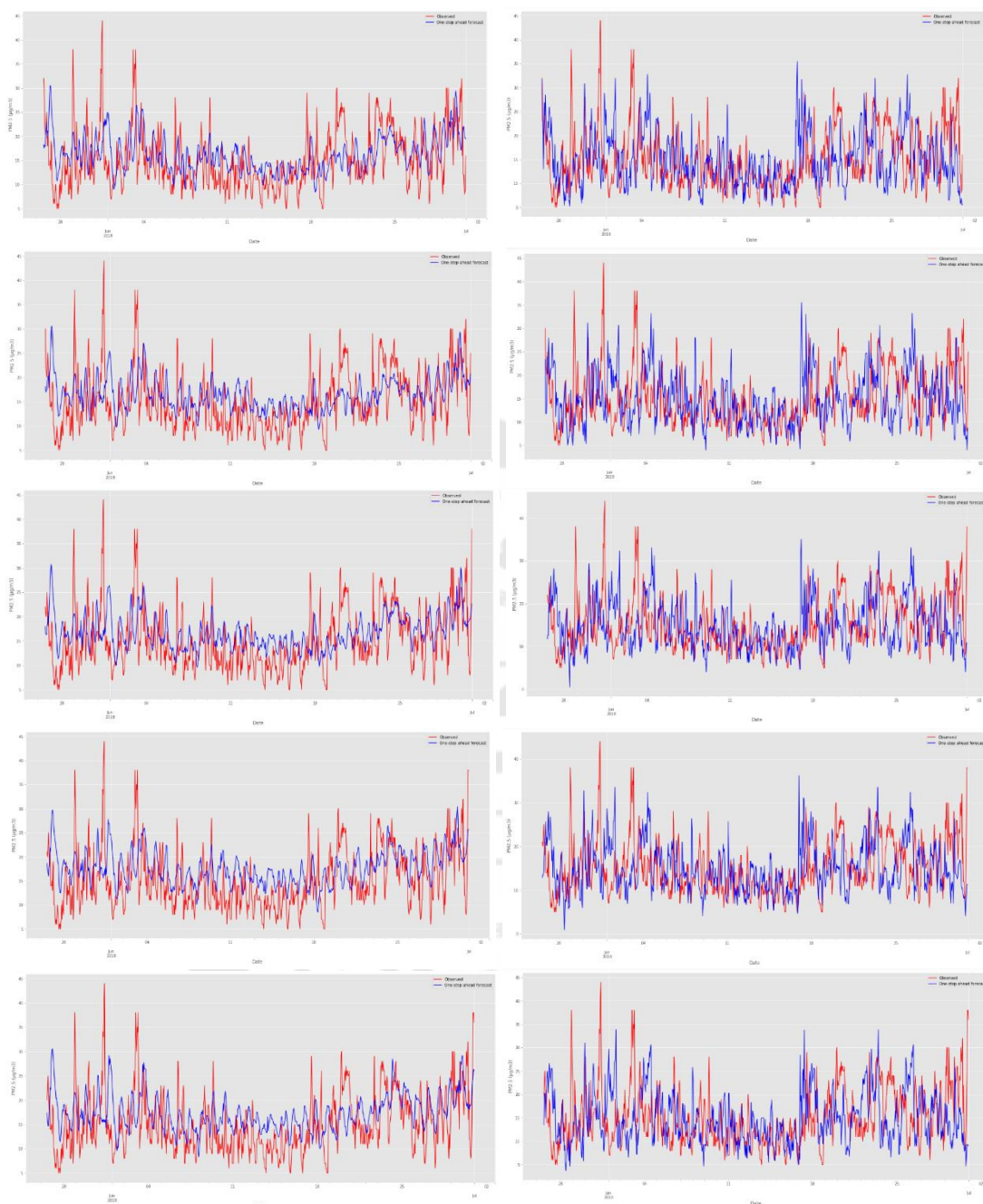


ภาพประกอบ 25 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 1 - 2 ชั่วโมง



ภาพประกอบ 26 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 3 - 5 ชั่วโมง

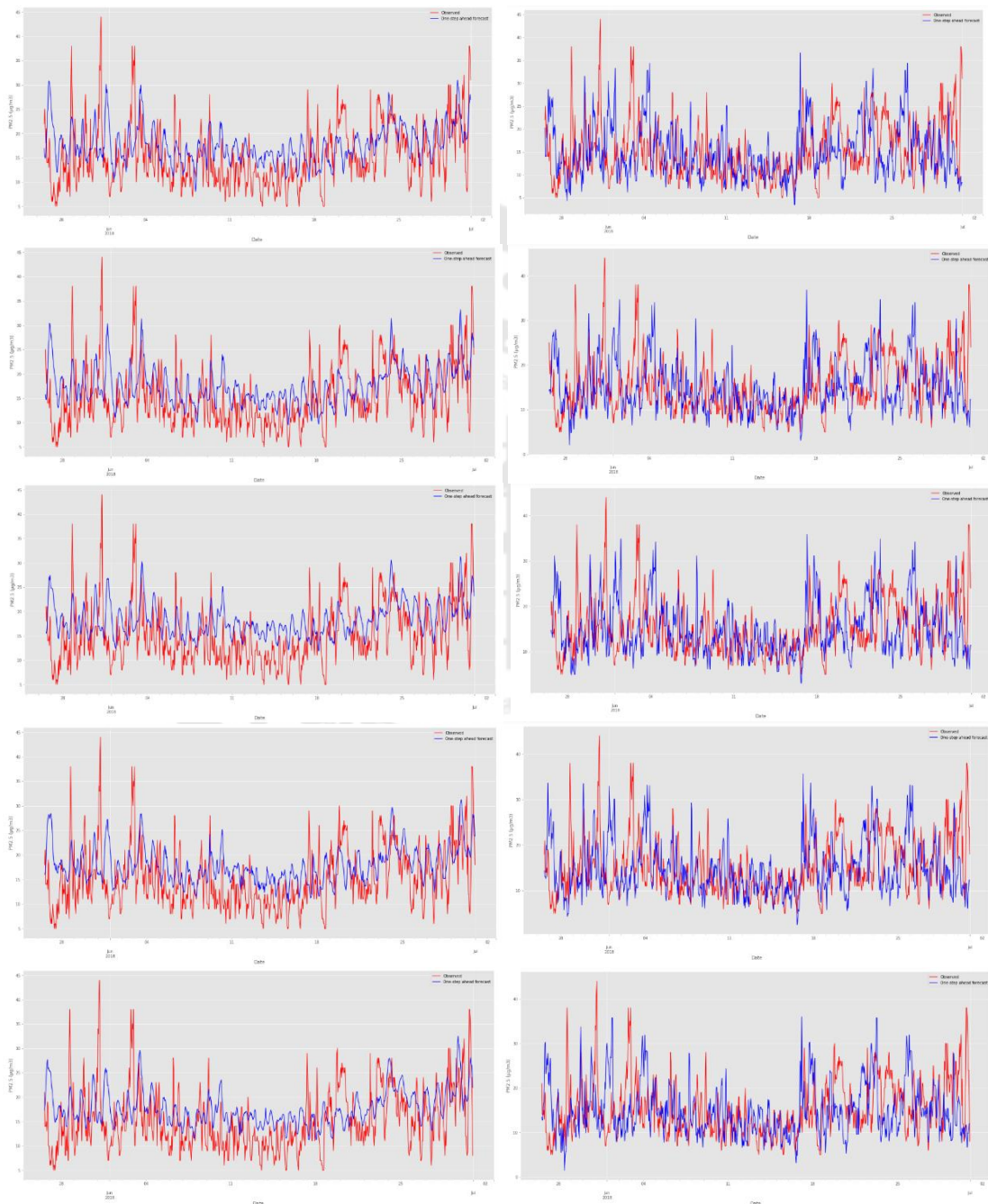
จากภาพประกอบ 27 นั้นแสดงผลการทำนายล่วงหน้า 6 ชั่วโมงถึง 10 ชั่วโมง โดยเรียงลำดับจากบนลงล่างระหว่างแบบจำลอง LSTM (รูปทางด้านซ้าย) และ SARIMAX (รูปทางด้านขวา) จากรูปจะเห็นได้ว่ากราฟของแบบจำลอง LSTM เมื่อนำมาพล็อตเทียบกับค่าจริงแล้วเส้นกราฟดูมีความแม่นยำมากกว่าเส้นกราฟของแบบจำลอง SARIMAX ซึ่งตั้งแต่ชั่วโมงที่ 6 ขึ้นไปนั้นจะเห็นได้ว่าเส้นกราฟที่ได้จากการทำนายของทั้งสองแบบจำลองนั้นมีค่าความแม่นยำลดลง เมื่อเทียบกับเส้นกราฟของค่าจริง ดังภาพประกอบด้านล่าง



ภาพประกอบ 27 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 6 - 10 ชั่วโมง

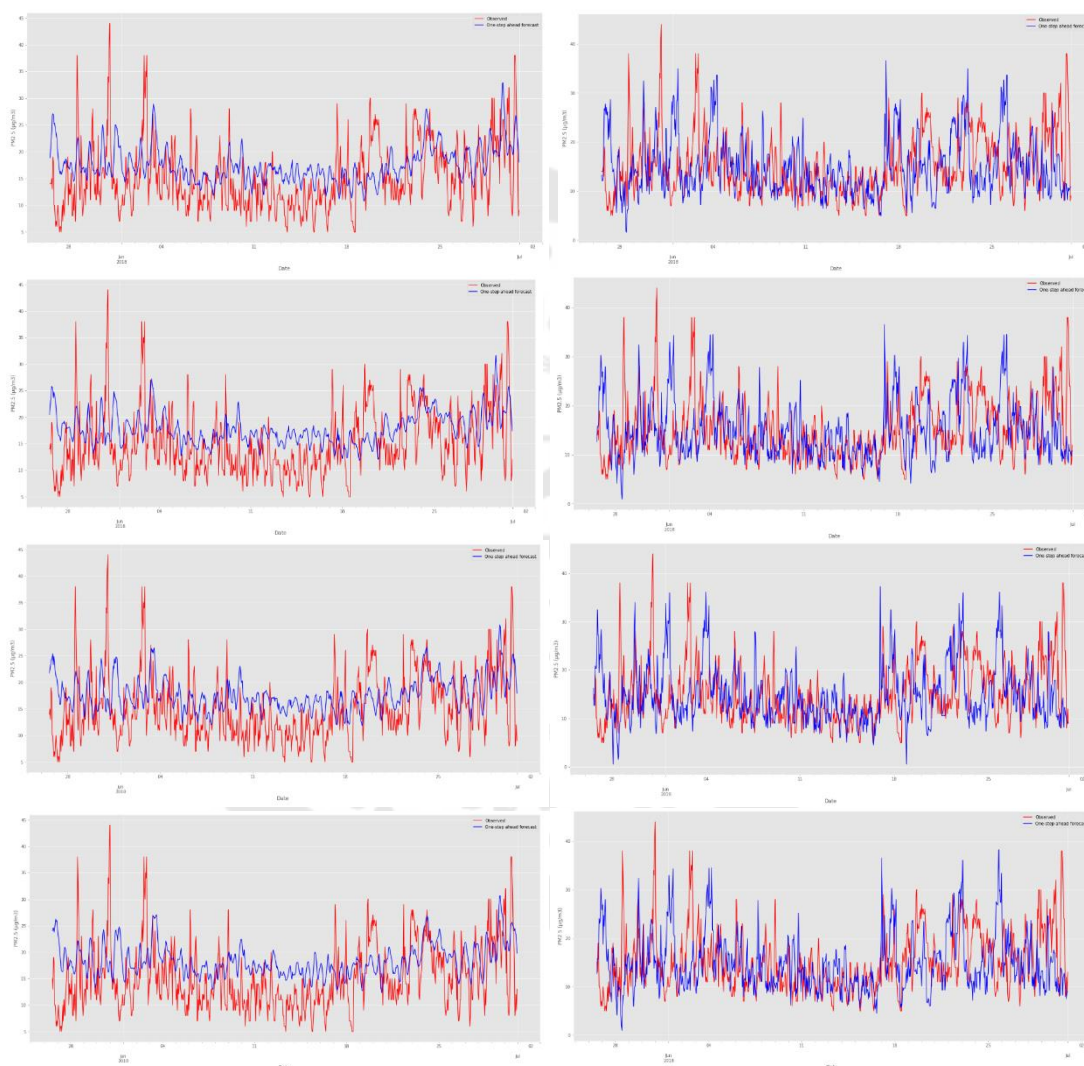
จากภาพประกอบ 28 นั้นแสดงผลการทำนายล่วงหน้า 11 ชั่วโมงถึง 15 ชั่วโมง โดยเรียงลำดับจากบนลงล่างระหว่างแบบจำลอง LSTM (รูปทางด้านซ้าย) และ SARIMAX (รูปทางด้านขวา) จากรูปจะเห็นได้ว่ากราฟของแบบจำลอง LSTM เมื่อนำมาพล็อตเทียบกับค่าจริง

แล้วเส้นกราฟดูมีความแม่นยำมากกว่าเส้นกราฟของแบบจำลอง SARIMAX ซึ่งตั้งแต่ชั่วโมงที่ 11 ขึ้นไปนั้นจะเห็นได้ว่าเส้นกราฟที่ได้จากการทำนายของทั้งสองแบบจำลองนั้นไม่ค่อยมีความแม่นยำเท่ากับค่าที่ได้จากการทำนายในช่วงชั่วโมงแรก ๆ ดังภาพประกอบด้านล่าง



ภาพประกอบ 28 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 11 - 15 ชั่วโมง

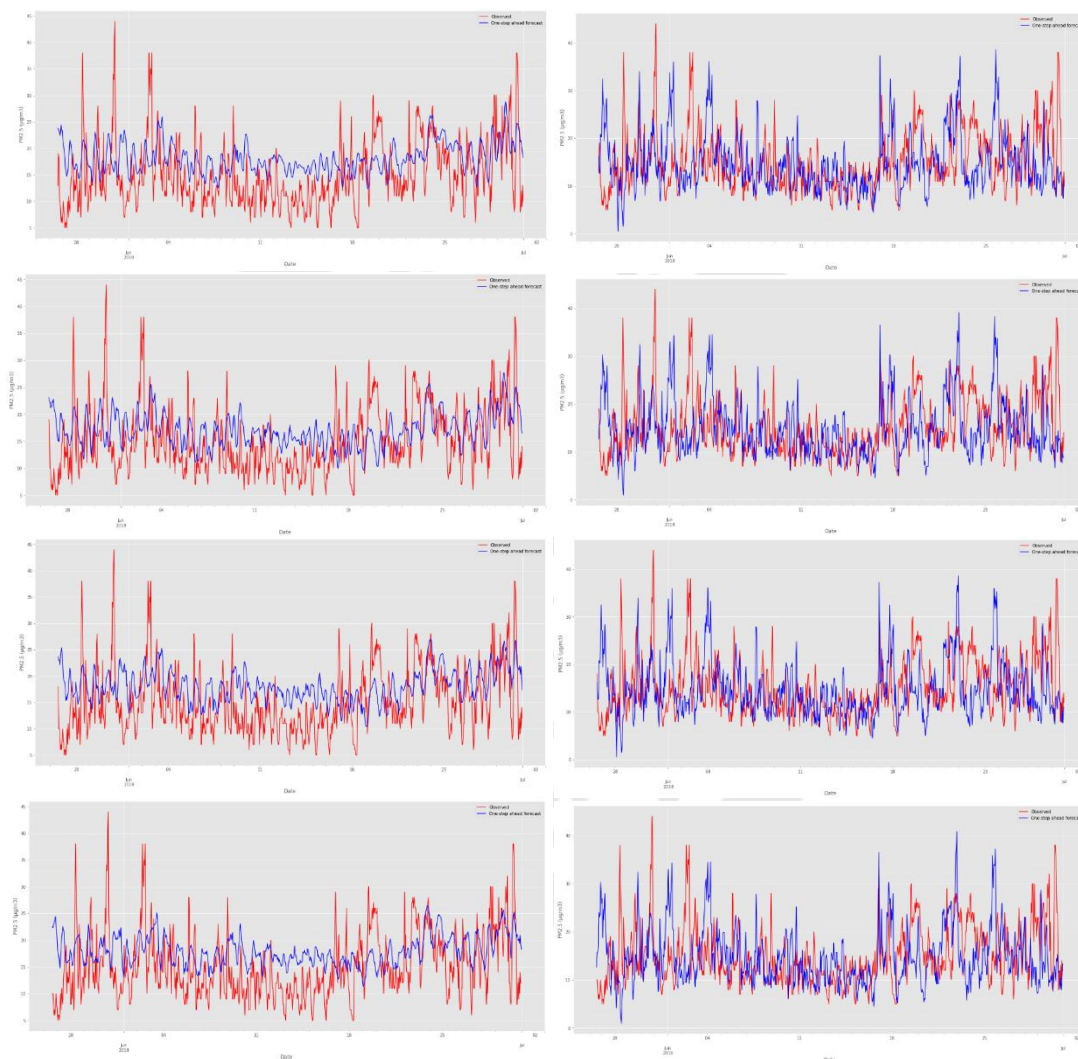
จากภาพประกอบ 29 นั้นแสดงผลพยากรณ์การดำเนินงานล่วงหน้า 16 ชั่วโมงถึง 19 ชั่วโมง โดยเรียงลำดับจากบนลงล่างระหว่างแบบจำลอง LSTM (รูปทางด้านซ้าย) และ SARIMAX (รูปทางด้านขวา) จากรูปจะเห็นได้ว่าเส้นกราฟของแบบจำลองทั้งสองเมื่อนำมาพล็อตเทียบกับค่าจริงแล้วเส้นกราฟดูไม่ค่อยมีความแม่นยำนักหากเทียบกับชั่วโมงแรก ๆ ดังภาพประกอบด้านล่าง



ภาพประกอบ 29 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 16 - 19 ชั่วโมง

จากภาพประกอบ 30 นั้นแสดงผลพยากรณ์การดำเนินงานล่วงหน้า 20 ชั่วโมงถึง 23 ชั่วโมง โดยเรียงลำดับจากบนลงล่างระหว่างแบบจำลอง LSTM (รูปทางด้านซ้าย) และ SARIMAX (รูป

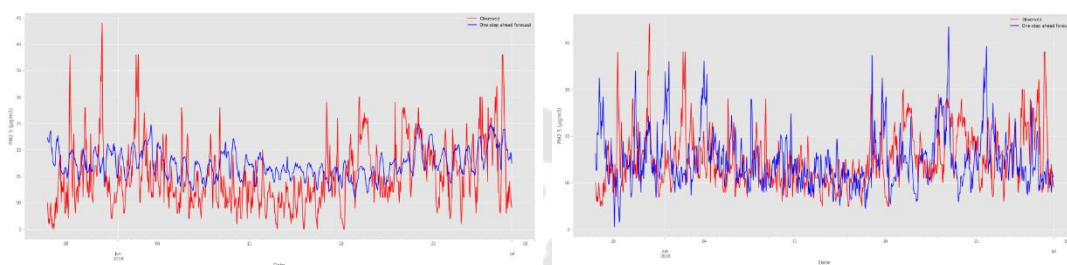
ทางด้านขวา) จากรูปจะเห็นได้ว่าเส้นกราฟของแบบจำลองทั้งสองนั้นเมื่อนำมาพล็อตเทียบกับค่าจริงแล้วเส้นกราฟดูไม่ค่อยมีความแม่นยำนัก แต่หากเทียบจากค่า RMSE และ MAE แล้วนั้นแบบจำลอง LSTM ยังคงมีค่าความแม่นยำมากกว่าแบบจำลอง SARIMAX ดังภาพประกอบด้านล่าง



ภาพประกอบ 30 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 20 - 23 ชั่วโมง

จากภาพประกอบ 31 นั้นแสดงผลลัพธ์การทำนายล่วงหน้า 24 ชั่วโมงระหว่างแบบจำลอง LSTM (รูปทางด้านซ้าย) และ SARIMAX (รูปทางด้านขวา) จากรูปจะเห็นได้ว่าเส้นกราฟของแบบจำลองทั้งสองนั้นเมื่อนำมาพล็อตเทียบกับค่าจริงแล้วเส้นกราฟดูไม่ค่อยมีความแม่นยำนัก

แต่เส้นกราฟสีน้ำเงินที่เป็นค่าที่ได้จากการทำนายของแบบจำลอง LSTM นั้นเส้นกราฟจะค่อนข้างเกาะกับค่าจริงมากกว่าถ้าเทียบกับเส้นกราฟที่ได้จากการทำนายของแบบจำลอง SARIMAX แต่หากเทียบจากค่า RMSE และ MAE แล้วแบบจำลอง LSTM นั้นมีค่าความแม่นยำมากกว่าแบบจำลอง SARIMAX ซึ่งในช่วงชั่วโมงหลัง ๆ จะเห็นได้ว่ายิ่งจำนวนชั่วโมงเพิ่มมากขึ้นเท่าไร ผลลัพธ์ที่ได้จากการทำนายค่าความแม่นยำก็จะลดลงตามไปด้วย ดังภาพประกอบด้านล่าง



ภาพประกอบ 31 แสดงกราฟเปรียบเทียบผลการทำนายฝุ่นละออง PM2.5 ระหว่างแบบจำลอง LSTM และ SARIMAX ล่วงหน้า 24 ชั่วโมง

จากภาพรวมสรุปได้ว่าผลการทำนายล่วงหน้าตั้งแต่ชั่วโมงที่ 1 จนถึงชั่วโมงที่ 24 นั้นจะเห็นได้ว่าค่าที่ได้จากการทำนายในช่วงชั่วโมงแรก ๆ นั้นจะมีค่าความแม่นยำมากกว่าชั่วโมงหลัง ๆ ดังนั้นกราฟที่ถูกพล็อตในชั่วโมงแรก ๆ นั้นจึงดูใกล้เคียงกับค่าจริงมากกว่ากราฟของชั่วโมงหลัง ๆ จึงสรุปได้ว่าประสิทธิภาพในการทำนายของแบบจำลอง LSTM และแบบจำลอง SARIMAX นั้นมีค่าความแม่นยำมากที่สุดในช่วงต้นชั่วโมงแรก ๆ ในการทำนายและจะค่อย ๆ แย่ลงเมื่อจำนวนชั่วโมงในการทำนายเพิ่มมากขึ้น

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

ในการวิจัยการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า ซึ่งใช้ข้อมูลฝุ่นละออง PM2.5 และข้อมูลทางอุตุนิยมวิทยาบริเวณพื้นที่สถานีตำรวจนครบาลโชคชัย โดยใช้เทคนิคการเรียนรู้ของเครื่อง ผู้วิจัยได้วัดประสิทธิภาพของแบบจำลองแต่ละอัลกอริทึมเพื่อนำมาเปรียบเทียบและสรุปผล โดยสามารถแบ่งหัวข้อในการสรุปผลได้ดังต่อไปนี้

1. สรุปผลการวิจัย
2. อภิปรายผลการวิจัย
3. ข้อเสนอแนะ

สรุปผลการวิจัย

ในปัจจุบันกรุงเทพมหานครนั้นกำลังเผชิญกับปัญหามลพิษทางอากาศ โดยเฉพาะปัญหาเรื่องฝุ่นละออง PM2.5 ที่เพิ่มมากขึ้นจากเมื่อก่อน จนส่งผลกระทบต่อสุขภาพของประชาชนที่ได้สัมผัสหรือสูดอากาศในบริเวณพื้นที่ที่มีค่าฝุ่นละออง PM2.5 สูงเกินเกณฑ์มาตรฐาน ซึ่งถ้าปัญหาดังกล่าวนั้นสามารถที่จะประเมินผลหรือคาดการณ์ล่วงหน้าได้ว่าจะมีปริมาณฝุ่นละออง PM2.5 เพิ่มมากขึ้นหรือลดน้อยลง และอาจนำข้อมูลที่ได้จากการทำนายนั้นมาใช้เป็นข้อมูลพื้นฐานในการวางแผนจัดการคุณภาพอากาศให้มีประสิทธิภาพมากยิ่งขึ้น

ในการวิจัยนี้ ได้ทำการศึกษาและวิเคราะห์เปรียบเทียบสมรรถนะระหว่างแบบจำลอง LSTM กับแบบจำลอง SARIMAX โดยผู้วิจัยได้เลือกศึกษาข้อมูลจากบริเวณพื้นที่สถานีตำรวจนครบาลโชคชัยและนำชุดข้อมูลดังกล่าวมาใช้ในการสร้างแบบจำลอง LSTM และ SARIMAX โดยใช้เทคนิคการเรียนรู้ของเครื่อง เพื่อเป็นแนวทางในการพัฒนาการสร้างแบบจำลองในการทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า โดยวัดประสิทธิภาพของแต่ละแบบจำลองได้ตามตาราง 4

ตาราง 4 ผลการทำนายค่าความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้าระหว่าง
แบบจำลอง LSTM และ SARIMAX

Time Step	แบบจำลอง LSTM		แบบจำลอง SARIMAX	
	RMSE	MAE	RMSE	MAE
t+1	3.11	2.36	6.65	5.13
t+2	4.14	3.16	6.91	5.25
t+3	4.87	3.74	6.98	5.31
t+4	5.36	4.11	9.59	5.86
t+5	5.60	4.36	7.63	5.83
t+6	5.84	4.55	7.46	5.61
t+7	5.98	4.70	7.58	5.66
t+8	6.09	4.80	7.83	5.88
t+9	6.48	5.29	7.82	5.84
t+10	6.32	5.07	7.76	5.76
t+11	6.47	5.24	7.87	5.80
t+12	6.44	5.20	7.88	5.69
t+13	6.69	5.56	8.10	5.89
t+14	6.52	5.25	9.07	6.10
t+15	6.46	5.26	7.93	5.86
t+16	6.42	5.27	8.13	5.91
t+17	6.52	5.37	8.12	5.94
t+18	6.45	5.24	9.40	6.09
t+19	6.71	5.54	8.17	5.77
t+20	6.60	5.46	8.04	5.61
t+21	6.30	5.08	8.08	5.67
t+22	6.59	5.50	8.25	5.83
t+23	6.54	5.44	8.77	5.85
t+24	6.37	5.20	7.96	5.64
Average	6.04	4.86	7.99	5.74

จากผลการทดลองสรุปได้ว่าแบบจำลองที่มีประสิทธิภาพดีที่สุ่มสุ่มนั้นคือ แบบจำลองโครงข่ายประสาทเทียมแบบลึก LSTM โดยค่าที่ได้จากการเฉลี่ยรวมค่าการทำนายความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า ซึ่งได้ค่า $RMSE = 6.04$ และ $MAE = 4.86$

อภิปรายผลการวิจัย

เนื่องจากในการวิจัยนี้ได้สร้างแบบจำลองแบบ Regression เพื่อใช้ในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ล่วงหน้า ดังนั้นจึงเลือกใช้ $RMSE$ และ MAE มาเป็นตัววัดประสิทธิภาพของแบบจำลองหรือการหาค่า error ซึ่งได้จากการหาค่าความแตกต่างระหว่างค่าที่ได้จากการทำนายกับค่าจริง โดย $RMSE$ นั้นแตกต่างกับ MAE ตรงที่ $RMSE$ นั้นจะนำผลรวมค่าเฉลี่ยของ error ที่ได้มายกกำลังสองแล้วค่อยทำการถอดรากที่สอง ส่วน MAE นั้นจะนำผลรวมค่าเฉลี่ยของ error ที่ได้มาถอดค่าสัมบูรณ์ โดยค่า $RMSE$ นั้นมีแนวโน้มที่จะมีผลต่อค่า error ที่ค่อนข้างสูงหรือเป็น outliers ซึ่งได้จากการทำนายมากกว่าค่า MAE ดังนั้นค่า $RMSE$ ที่ได้ออกมานั้นจึงมีค่ามากกว่าค่า MAE ดังผลลัพธ์ที่แสดงในตาราง 4 จะเห็นได้ว่าค่า $RMSE$ และ MAE ของแบบจำลอง LSTM นั้นมีค่าต่ำกว่าค่าที่ได้จากแบบจำลอง SARIMAX และได้ลองนำเอาค่า $RMSE$ มาทำการลบกับค่า MAE ที่ $t+1$ ทั้งของแบบจำลอง LSTM และแบบจำลอง SARIMAX ซึ่งได้ออกมาเท่ากับ 0.75 และ 1.52 ส่วนค่าที่ $t+24$ เมื่อนำค่ามาลบกันเรียบร้อยแล้วจะได้ค่าของแบบจำลอง LSTM เท่ากับ 1.17 และค่าของแบบจำลอง SARIMAX เท่ากับ 2.32 ซึ่งหลังจากนำค่า $RMSE$ มาทำการลบกับค่า MAE แล้วจะเห็นได้ว่าค่าของแบบจำลอง LSTM นั้นมีค่าต่ำกว่าค่าที่ได้จากแบบจำลอง SARIMAX นั้นแสดงว่าค่า $RMSE$ และ MAE ของแบบจำลอง LSTM นั้นมีค่า error ไม่ค่อยสูงมากนักเมื่อเทียบกับแบบจำลอง SARIMAX ดังนั้นเมื่อนำค่าที่ได้จากการทำนายไปทำการพล็อตกราฟเปรียบเทียบกับค่าจริงก็จะเห็นได้ว่ากราฟของ LSTM นั้นจะค่อนข้างเกาะกับกราฟของค่าจริงมากกว่ากราฟที่ถูกพล็อตจากค่าที่ได้จากแบบจำลอง SARIMAX และจะเห็นได้ว่าผลลัพธ์ที่ได้จากแบบจำลองที่ถูกสร้างด้วยอัลกอริทึม LSTM นั้นมีประสิทธิภาพในการทำนายค่าในอีก 24 ชั่วโมงข้างหน้าออกมาได้มีความแม่นยำกว่าแบบจำลองที่ถูกสร้างด้วยอัลกอริทึม SARIMAX โดยทั้งสองแบบจำลองนั้นได้มีการใช้ feature อื่น เช่น PM10, CO, NO เป็นต้น มาช่วยเพิ่มประสิทธิภาพในการทำนายค่าความแม่นยำให้ดียิ่งขึ้น และจากผลลัพธ์ที่ได้จากการเปรียบเทียบค่า $RMSE$ และ MAE ของทั้งสองแบบจำลอง จะเห็นว่าค่าที่ได้ออกมานั้นค่อนข้างต่ำหรือก็คือมีประสิทธิภาพในการทำนายค่าความแม่นยำค่อนข้างดี เพียงแต่ผลการทำนายค่าความ

แม่นยำของแบบจำลอง LSTM นั้นให้ผลออกมามีประสิทธิภาพที่ดีกว่า เนื่องจากแบบจำลอง LSTM นั้นเป็นแบบจำลองโครงข่ายประสาทเทียมแบบเชิงลึก (Deep Neural Network) จึงทำให้ตัวแบบจำลองนั้นสามารถเรียนรู้ชุดข้อมูลที่ผู้วิจัยใส่เข้าไปได้ดีกว่า รวมถึงสามารถตั้งค่าหรือปรับจูนค่าพารามิเตอร์ต่าง ๆ ในแบบจำลองได้มากกว่าการใช้แบบจำลองอนุกรมเวลาแบบ SARIMAX ซึ่งเป็นแบบจำลองดั้งเดิม ดังนั้นในการตั้งค่าหรือปรับจูนค่าพารามิเตอร์ต่าง ๆ เลยค่อนข้างจำกัด จึงทำให้ประสิทธิภาพในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า นั้นมีค่าความแม่นยำน้อยกว่าแบบจำลอง LSTM ซึ่งในอนาคตหากมีการจูนหรือปรับค่าพารามิเตอร์ในแบบจำลอง LSTM ให้ดีขึ้นนี้อาจช่วยเพิ่มประสิทธิภาพในการทำนายของแบบจำลองให้มีความแม่นยำมากยิ่งขึ้น ซึ่งหากมีการนำเอาแบบจำลองดังกล่าวไปประยุกต์ใช้งานจริง โดยใช้ในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ล่วงหน้า 24 ชั่วโมง แบบจำลอง LSTM นั้นถือว่าเป็นอีกหนึ่งแบบจำลองที่น่าสนใจและน่านำมาประยุกต์ใช้งานจริง เพียงแต่ในการวิจัยนี้เราได้ทำการทดลองโดยวัดประสิทธิภาพจากการเปรียบเทียบค่าความแม่นยำจากสองอัลกอริทึมเท่านั้น ดังนั้นอาจมีอัลกอริทึมอื่นที่อาจมีประสิทธิภาพมากกว่าและสามารถทำนายค่าความแม่นยำของปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า ได้ดีกว่าทั้งสองอัลกอริทึมนี้

ข้อเสนอแนะ

1. เนื่องจากในการวิจัยนี้ ในขั้นตอนการเตรียมข้อมูลนั้นได้ทำการ Drop ข้อมูลที่เป็น missing value หรือ outliers ทั้งทั้งแถว ทำให้ข้อมูลที่จะต้องใช้งานบางส่วนนั้นถูกลบทิ้งไป ดังนั้นหากมีวิธีที่ไม่จำเป็นจะต้อง drop ข้อมูลดังกล่าวทิ้ง อาจช่วยเพิ่มประสิทธิภาพในการทำนายของแบบจำลองให้ดียิ่งขึ้น
2. ในการวิจัยนี้ได้ใช้ข้อมูลในการศึกษาเพียงบริเวณสถานีเดียว หากใช้ข้อมูลที่ถูกเก็บจากบริเวณสถานีอื่นด้วย อาจช่วยให้แบบจำลองนั้นสามารถเรียนรู้ข้อมูลได้มากขึ้น รวมถึงสามารถทำนายค่าออกมาได้มีความแม่นยำและมีประสิทธิภาพมากกว่าแบบจำลองที่ใช้ข้อมูลจากบริเวณพื้นที่เดียวหรือสถานีเดียว
3. เนื่องจากในการวิจัยนี้ได้ใช้อัลกอริทึมเพียงสองแบบ คือ LSTM และ SARIMAX ดังนั้นอาจจะมีอัลกอริทึมอื่นที่สามารถเรียนรู้ข้อมูลที่มีจำนวนมากและทำนายค่าออกมาได้มีความแม่นยำและมีประสิทธิภาพดีกว่า

บรรณานุกรม

- Arampongsanuwat, S., & Meesad, P. (2011). *Prediction of PM_{10} using Support Vector Regression*. Paper presented at the 2011 International Conference on Information and Electronics Engineering.
- Ausati, S., & Amanollahi, J. (2016). Assessing the accuracy of ANFIS, EEMD-GRNN, PCR, and MLR models in predicting $PM_{2.5}$. *Atmospheric Environment*, 142, 465-474. doi:<https://doi.org/10.1016/j.atmosenv.2016.08.007>
- Fu, M., Wang, C., Le, Z., & Manko, D. (2017). *Double layer back propagation neural network based on restricted Boltzmann machines for forecasting daily particulate matter 2.5*.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222-2232. doi:10.1109/TNNLS.2016.2582924
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735
- Hu, Z., Li, W., & Qiao, J. (2016, 27-29 July 2016). *Prediction of $PM_{2.5}$ based on Elman neural network with chaos theory*. Paper presented at the 2016 35th Chinese Control Conference (CCC).
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting : principles and practice* (2nd ed.). Melbourne, Australia: OTexts.
- Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). *An empirical exploration of recurrent network architectures*. Paper presented at the Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, Lille, France.
- Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231, 997-1004. doi:<https://doi.org/10.1016/j.envpol.2017.08.114>

- Mahajan, S., Chen, L., & Tsai, T. (2017, 4-8 Aug. 2017). *An empirical study of PM_{2.5} forecasting using neural network*. Paper presented at the 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI).
- Park, J., Yoo, S., Kim, K., Gu, Y., Lee, K., & Son, U. (2017, 4-7 July 2017). *PM₁₀ density forecast model using long short term memory*. Paper presented at the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN).
- Qi, Z., Wang, T., Song, G., Hu, W., Li, X., Zhongfei, & Zhang. (2017). *Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality* (Vol. PP).
- Saeed, S., Hussain, L., Awan, I., & Idris, A. (2017). *Comparative Analysis of different Statistical Methods for Prediction of PM_{2.5} and PM₁₀ Concentrations in Advance for Several Hours* (Vol. 17).
- Sarkar, D., Bali, R., & Sharma, T. (2018). *Practical Machine Learning with Python A Problem-Solver's Guide to Building Real-World Intelligent Systems* (1st ed.). United States: Apress.
- Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications* (4th ed.). Cham: Springer International Publishing AG.
- V, A., P, G., R, V., & K P, S. (2018). DeepAirNet: Applying Recurrent Networks for Air Quality Prediction. *Procedia Computer Science*, 132, 1394-1403.
doi:<https://doi.org/10.1016/j.procs.2018.05.068>
- Vagropoulos, S. I., Chouliaras, G. I., Kardakos, E. G., Simoglou, C. K., & Bakirtzis, A. G. (2016, 4-8 April 2016). *Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting*. Paper presented at the 2016 IEEE International Energy Conference (ENERGYCON).
- Wang, P., Zhang, H., Qin, Z., & Zhang, G. (2017). A novel hybrid-Garch model based on ARIMA and SVM for PM_{2.5} concentrations forecasting. *Atmospheric Pollution Research*, 8(5), 850-860. doi:<https://doi.org/10.1016/j.apr.2017.01.003>

- Wongsathan, R., & Seedadan, I. (2016). A Hybrid ARIMA and Neural Networks Model for PM-10 Pollution Estimation: The Case of Chiang Mai City Moat Area. *Procedia Computer Science*, 86, 273-276. doi:<https://doi.org/10.1016/j.procs.2016.05.057>
- Xie, J. (2017, 16-18 Aug. 2017). *Deep Neural Network for PM2.5 Pollution Forecasting Based on Manifold Learning*. Paper presented at the 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC).
- Yi, X., Zhang, J., Wang, Z., Li, T., & Zheng, Y. (2018). *Deep Distributed Fusion Network for Air Quality Prediction*. Paper presented at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom.
- Zhou, Q., Jiang, H., Wang, J., & Zhou, J. (2014). A hybrid model for PM2.5 forecasting based on ensemble empirical mode decomposition and a general regression neural network. *Science of The Total Environment*, 496, 264-274. doi:<https://doi.org/10.1016/j.scitotenv.2014.07.051>
- บัวคำศรี, ธ. (2562). ปกป้องสุขภาพคนไทย ถึงเวลาเปลี่ยนมาตรฐานฝุ่น PM2.5. Retrieved from <http://www.greenpeace.org/seasia/th/news/blog1/pm25/blog/62098/>



ภาคผนวก

รายละเอียดโค้ดในการสร้างแบบจำลอง LSTM เพื่อใช้ในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า

```
# ทำการ Import libraries ที่ใช้ใน LSTM
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler, RobustScaler, MinMaxScaler, Normalizer
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('ggplot')
from math import sqrt
from sklearn import metrics
from numpy import concatenate
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM, Dropout, GRU, Bidirectional
from datetime import datetime
from pandas import datetime
from pandas import ExcelWriter
from pandas import ExcelFile
from keras.models import load_model
from pandas import read_csv
from tensorflow import set_random_seed
set_random_seed(36)

# ทำการนำเข้าไฟล์ข้อมูล
def parse(x):
    return datetime.strptime(x, '%Y %m %d %H')
df = pd.read_excel('53t.xlsx', parse_dates = [['year', 'month', 'day', 'hour']], index_col=0, date_parser=parse)
# เปลี่ยนชื่อ index ให้เป็น date
df.index.name = 'date'
# แสดงค่า 5 แถวแรกของข้อมูล
print(df.head(5))
# ให้ชื่อข้อมูลของแต่ละคอลัมน์
df.columns

# ทำการลบช่องว่างที่อยู่ระหว่างชื่อของแต่ละคอลัมน์
df.columns = map(lambda x: x.strip(), df.columns.values)
# เปลี่ยนชื่อคอลัมน์ PM2.5 ให้เป็น PM_density
df = df.rename(columns = {'PM2.5': 'PM_density'})
df.columns

# เช็คจำนวนแถวและคอลัมน์ในชุดข้อมูล
df.shape
# ทำการแทนที่ช่องข้อมูลที่ว่าง
df.replace(r'^\s*$', np.nan, regex=True, inplace=True)
# ทำการเช็คข้อมูลว่ามีค่าเป็น null
len(df[df.isnull().any(axis=1)])
r_index = df[df.isnull().any(axis=1)].index.tolist()
len(r_index)
# ทำการตัดข้อมูลที่มีค่า null ทั้ง
df_drop1 = df.dropna()
df_drop1.shape
# เช็คประเภทของข้อมูลในแต่ละคอลัมน์
df_drop1.dtypes
# ดูรายละเอียดของข้อมูล
df_drop1.describe()
```



```

# function สำหรับ preprocess ข้อมูล time series หลายตัวแปร
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True, feat_name=None):

    n_vars = 1 if type(data) is list else data.shape[1]
    df = pd.DataFrame(data)
    cols, names = list(), list()

    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += ['{feat_name[j]}(t-{i})' for j in range(n_vars)]

    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += ['{feat_name[j]}(t)' for j in range(n_vars)]
        else:
            names += ['{feat_name[j]}(t+{i})' for j in range(n_vars)]

    # put it all together
    agg = pd.concat(cols, axis=1)
    agg.columns = names

    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg

# function สำหรับ reshape ข้อมูลและทำการ fit ข้อมูลเข้ากันแบบจำลอง
def fit_lstm(train, n_lag, n_ahead, n_batch, nb_epoch, n_neurons):
    # reshape training into [samples, timesteps, features]
    X, y = train[:, :-n_ahead], train[:, -n_ahead:]
    X = X.reshape(X.shape[0], n_lag, int(X.shape[1]/n_lag))

    # design network
    model = Sequential()
    model.add(LSTM(n_neurons, batch_input_shape=(n_batch, X.shape[1], X.shape[2]), stateful=True))
    model.add(Dense(n_ahead))
    model.compile(loss='mean_squared_error', optimizer='RMSprop')

    # fit network
    for i in range(nb_epoch):
        model.fit(X, y, epochs=10, batch_size=n_batch, verbose=2, shuffle=False)
        model.reset_states()
    return model

# function สำหรับ ทำนายข้อมูล
def forecast_lstm(model, X, n_batch, n_lag):
    # reshape input pattern to [samples, timesteps, features]
    X = X.reshape(1, n_lag, int(len(X)/n_lag))
    # make forecast
    forecast = model.predict(X, batch_size=n_batch)
    # convert to array
    return [x for x in forecast[0, :]]

def make_forecasts(model, n_batch, train, test, n_lag, n_ahead):
    forecasts = list()
    for i in range(len(test)):
        X = test[i, :-n_ahead]
        # make forecast
        forecast = forecast_lstm(model, X, n_batch, n_lag)
        # store the forecast
        forecasts.append(forecast)
    return forecasts

```

```

# function สำหรับประเมินประสิทธิภาพของแบบจำลองโดยใช้ RMSE
def evaluate_forecasts1(y, forecasts, n_lag, n_seq):
    for i in range(n_seq):
        actual = [row[i] for row in y]
        predicted = [forecast[i] for forecast in forecasts]
        rmse = sqrt(mean_squared_error(actual, predicted))
        print('t+%d RMSE: %f' % ((i+1), rmse))
# function สำหรับประเมินประสิทธิภาพของแบบจำลองโดยใช้ MAE
def evaluate_forecasts2(y, forecasts, n_lag, n_seq):
    for i in range(n_seq):
        actual = [row[i] for row in y]
        predicted = [forecast[i] for forecast in forecasts]
        mae = mean_absolute_error(actual, predicted)
        print('t+%d MAE: %f' % ((i+1), mae))

# เปลี่ยนชื่อตัวแปรที่ใช้เรียก
df = df_drop1
# ทำ scaling ข้อมูลด้วยวิธี min max scaling
min_dict = dict()
max_dict = dict()
for col in df.columns:
    min_dict[col] = df[col].min()
    max_dict[col] = df[col].max()
    df[col] = (df[col] - df[col].min()) / (df[col].max() - df[col].min())

# กำหนดค่า lag และ ahead
lag = 168
ahead = 24
# ใช้ function supervised learning ในการ shift ข้อมูล
reframed = series_to_supervised(df, lag, ahead, feat_name=df.columns)
print(reframed.head())

# drop คอลัมน์ที่ไม่ต้องการใช้ในการทำนาย
reframed.drop(['NO(t)', 'NO2(t)', 'NOX(t)', 'PM10(t)', 'Wind speed(t)', 'Wind dir(t)', 'Temp(t)', 'Rel hum(t)', 'Pressure(t)', 'Rain(t)', 'CO(t)',
               'NO(t+1)', 'NO2(t+1)', 'NOX(t+1)', 'PM10(t+1)', 'Wind speed(t+1)', 'Wind dir(t+1)', 'Temp(t+1)', 'Rel hum(t+1)', 'Pressure(t+1)', 'Rain(t+1)', 'CO(t+1)',
               'NO(t+2)', 'NO2(t+2)', 'NOX(t+2)', 'PM10(t+2)', 'Wind speed(t+2)', 'Wind dir(t+2)', 'Temp(t+2)', 'Rel hum(t+2)', 'Pressure(t+2)', 'Rain(t+2)', 'CO(t+2)',
               'NO(t+3)', 'NO2(t+3)', 'NOX(t+3)', 'PM10(t+3)', 'Wind speed(t+3)', 'Wind dir(t+3)', 'Temp(t+3)', 'Rel hum(t+3)', 'Pressure(t+3)', 'Rain(t+3)', 'CO(t+3)',
               'NO(t+4)', 'NO2(t+4)', 'NOX(t+4)', 'PM10(t+4)', 'Wind speed(t+4)', 'Wind dir(t+4)', 'Temp(t+4)', 'Rel hum(t+4)', 'Pressure(t+4)', 'Rain(t+4)', 'CO(t+4)',
               'NO(t+5)', 'NO2(t+5)', 'NOX(t+5)', 'PM10(t+5)', 'Wind speed(t+5)', 'Wind dir(t+5)', 'Temp(t+5)', 'Rel hum(t+5)', 'Pressure(t+5)', 'Rain(t+5)', 'CO(t+5)',
               'NO(t+6)', 'NO2(t+6)', 'NOX(t+6)', 'PM10(t+6)', 'Wind speed(t+6)', 'Wind dir(t+6)', 'Temp(t+6)', 'Rel hum(t+6)', 'Pressure(t+6)', 'Rain(t+6)', 'CO(t+6)',
               'NO(t+7)', 'NO2(t+7)', 'NOX(t+7)', 'PM10(t+7)', 'Wind speed(t+7)', 'Wind dir(t+7)', 'Temp(t+7)', 'Rel hum(t+7)', 'Pressure(t+7)', 'Rain(t+7)', 'CO(t+7)',
               'NO(t+8)', 'NO2(t+8)', 'NOX(t+8)', 'PM10(t+8)', 'Wind speed(t+8)', 'Wind dir(t+8)', 'Temp(t+8)', 'Rel hum(t+8)', 'Pressure(t+8)', 'Rain(t+8)', 'CO(t+8)',
               'NO(t+9)', 'NO2(t+9)', 'NOX(t+9)', 'PM10(t+9)', 'Wind speed(t+9)', 'Wind dir(t+9)', 'Temp(t+9)', 'Rel hum(t+9)', 'Pressure(t+9)', 'Rain(t+9)', 'CO(t+9)',
               'NO(t+10)', 'NO2(t+10)', 'NOX(t+10)', 'PM10(t+10)', 'Wind speed(t+10)', 'Wind dir(t+10)', 'Temp(t+10)', 'Rel hum(t+10)', 'Pressure(t+10)', 'Rain(t+10)', 'CO(t+10)',
               'NO(t+11)', 'NO2(t+11)', 'NOX(t+11)', 'PM10(t+11)', 'Wind speed(t+11)', 'Wind dir(t+11)', 'Temp(t+11)', 'Rel hum(t+11)', 'Pressure(t+11)', 'Rain(t+11)', 'CO(t+11)',
               'NO(t+12)', 'NO2(t+12)', 'NOX(t+12)', 'PM10(t+12)', 'Wind speed(t+12)', 'Wind dir(t+12)', 'Temp(t+12)', 'Rel hum(t+12)', 'Pressure(t+12)', 'Rain(t+12)', 'CO(t+12)',
               'NO(t+13)', 'NO2(t+13)', 'NOX(t+13)', 'PM10(t+13)', 'Wind speed(t+13)', 'Wind dir(t+13)', 'Temp(t+13)', 'Rel hum(t+13)', 'Pressure(t+13)', 'Rain(t+13)', 'CO(t+13)',
               'NO(t+14)', 'NO2(t+14)', 'NOX(t+14)', 'PM10(t+14)', 'Wind speed(t+14)', 'Wind dir(t+14)', 'Temp(t+14)', 'Rel hum(t+14)', 'Pressure(t+14)', 'Rain(t+14)', 'CO(t+14)',
               'NO(t+15)', 'NO2(t+15)', 'NOX(t+15)', 'PM10(t+15)', 'Wind speed(t+15)', 'Wind dir(t+15)', 'Temp(t+15)', 'Rel hum(t+15)', 'Pressure(t+15)', 'Rain(t+15)', 'CO(t+15)',
               'NO(t+16)', 'NO2(t+16)', 'NOX(t+16)', 'PM10(t+16)', 'Wind speed(t+16)', 'Wind dir(t+16)', 'Temp(t+16)', 'Rel hum(t+16)', 'Pressure(t+16)', 'Rain(t+16)', 'CO(t+16)',
               'NO(t+17)', 'NO2(t+17)', 'NOX(t+17)', 'PM10(t+17)', 'Wind speed(t+17)', 'Wind dir(t+17)', 'Temp(t+17)', 'Rel hum(t+17)', 'Pressure(t+17)', 'Rain(t+17)', 'CO(t+17)',
               'NO(t+18)', 'NO2(t+18)', 'NOX(t+18)', 'PM10(t+18)', 'Wind speed(t+18)', 'Wind dir(t+18)', 'Temp(t+18)', 'Rel hum(t+18)', 'Pressure(t+18)', 'Rain(t+18)', 'CO(t+18)',
               'NO(t+19)', 'NO2(t+19)', 'NOX(t+19)', 'PM10(t+19)', 'Wind speed(t+19)', 'Wind dir(t+19)', 'Temp(t+19)', 'Rel hum(t+19)', 'Pressure(t+19)', 'Rain(t+19)', 'CO(t+19)',
               'NO(t+20)', 'NO2(t+20)', 'NOX(t+20)', 'PM10(t+20)', 'Wind speed(t+20)', 'Wind dir(t+20)', 'Temp(t+20)', 'Rel hum(t+20)', 'Pressure(t+20)', 'Rain(t+20)', 'CO(t+20)',
               'NO(t+21)', 'NO2(t+21)', 'NOX(t+21)', 'PM10(t+21)', 'Wind speed(t+21)', 'Wind dir(t+21)', 'Temp(t+21)', 'Rel hum(t+21)', 'Pressure(t+21)', 'Rain(t+21)', 'CO(t+21)',
               'NO(t+22)', 'NO2(t+22)', 'NOX(t+22)', 'PM10(t+22)', 'Wind speed(t+22)', 'Wind dir(t+22)', 'Temp(t+22)', 'Rel hum(t+22)', 'Pressure(t+22)', 'Rain(t+22)', 'CO(t+22)',
               'NO(t+23)', 'NO2(t+23)', 'NOX(t+23)', 'PM10(t+23)', 'Wind speed(t+23)', 'Wind dir(t+23)', 'Temp(t+23)', 'Rel hum(t+23)', 'Pressure(t+23)', 'Rain(t+23)', 'CO(t+23)'],
               axis=1, inplace=True)

# เช็คว่าข้อมูลหลังจากการ drop ข้อมูลที่ไม่ใช่แล้ว
print(reframed.head())

# กำหนด train / test โดยแบ่งเป็น train 90% และ test 10%
values = reframed.values
n_train = int(len(values) * 0.9)
train = values[:n_train, :]
test = values[n_train:, :]
# แบ่ง train / test ออกเป็น input และ outputs เพื่อใส่ในแบบจำลอง
train_X, train_y = train[:, :-ahead], train[:, -ahead:]
test_X, test_y = test[:, :-ahead], test[:, -ahead:]
# reshape ข้อมูล input ให้เป็น 3D (samples, timesteps, features)
train_X = train_X.reshape(train_X.shape[0], lag, int(train_X.shape[1]/lag))
test_X = test_X.reshape(test_X.shape[0], lag, int(test_X.shape[1]/lag))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)

# กำหนดค่า batch size กับ hidden neurons
n_batch = 27
n_neurons = 100

```

```

# สร้าง LSTM network โดยใช้ library ของ Keras
model = Sequential()
model.add(LSTM(n_neurons, batch_input_shape=(n_batch, train_X.shape[1], train_X.shape[2]), stateful=True))
model.add(Dense(ahead))
model.compile(loss='mean_squared_error', optimizer='RMSprop')
model.summary()

# ทำการ train แบบจำลอง โดยกำหนดให้ epochs = 50
history = model.fit(train_X, train_y, epochs=50, batch_size=n_batch, verbose=2, shuffle=False)
model.reset_states()

# สร้าง LSTM network ขึ้นอีกครั้ง โดยกำหนดให้ batch size = 1 และ copy weights จากการ train ก่อนนำมาใช้ในการสร้างแบบจำลอง
# เพื่อใช้ในการทำนายค่า
n_batch = 1
# re-define model
new_model = Sequential()
new_model.add(LSTM(n_neurons, batch_input_shape=(n_batch, train_X.shape[1], train_X.shape[2]), stateful=True))
new_model.add(Dense(ahead))
# copy weights
old_weights = model.get_weights()
new_model.set_weights(old_weights)
# compile model
new_model.compile(loss='mean_squared_error', optimizer='RMSprop')

# ทำการประเมินประสิทธิภาพของแบบจำลองโดยใช้ RMSE, MAE
forecasts = make_forecasts(new_model, 1, train, test, lag, ahead)
# evaluate forecasts RMSE
actual = [row[-ahead:] for row in test]
evaluate_forecasts1(actual, forecasts, lag, ahead)
forecasts = make_forecasts(new_model, 1, train, test, lag, ahead)
# evaluate forecasts MAE
actual = [row[-ahead:] for row in test]
evaluate_forecasts2(actual, forecasts, lag, ahead)

# แปลงค่า yhat ให้เป็น array
yhat = np.asarray(forecasts)
yhat.shape
print(yhat)

# ทำ rescale ข้อมูลกลับ
inv_yhat = yhat*(max_dict['PM_density']-min_dict['PM_density'])+min_dict['PM_density']
inv_y = test_y*(max_dict['PM_density']-min_dict['PM_density'])+min_dict['PM_density']
# จัดข้อมูลให้อยู่ในรูปของ list
inv_yhat1 = inv_yhat.tolist()
inv_y1 = inv_y.tolist()

# ทำการประเมินประสิทธิภาพของแบบจำลองโดยใช้ RMSE, MAE อีกครั้งหลัง rescale กลับ
forecasts = make_forecasts(new_model, 1, train, test, lag, ahead)
# evaluate forecasts RMSE
actual = [row[-ahead:] for row in test]
evaluate_forecasts1(inv_y, inv_yhat, lag, ahead)
forecasts = make_forecasts(new_model, 1, train, test, lag, ahead)
# evaluate forecasts MAE
actual = [row[-ahead:] for row in test]
evaluate_forecasts2(inv_y, inv_yhat, lag, ahead)

```

รายละเอียดโค้ดในการสร้างแบบจำลอง SARIMAX เพื่อใช้ในการทำนายหาปริมาณความหนาแน่นของฝุ่นละออง PM2.5 ในอีก 24 ชั่วโมงข้างหน้า

```
# ทำการ Import libraries ที่ใช้ใน SARIMAX
import warnings
import itertools
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
plt.rcParams['figure.figsize'] = (20.0, 10.0)
plt.rcParams.update({'font.size': 12})
plt.style.use('ggplot')
from pandas import Series, DataFrame
from datetime import datetime, timedelta
%matplotlib inline
import datetime
from dateutil.relativedelta import relativedelta
import seaborn as sns
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.metrics import mean_squared_error
from math import sqrt
from sklearn import metrics
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.arima_model import ARIMA
from math import sin
from random import random
from datetime import datetime
from pandas import ExcelWriter
from pandas import ExcelFile
from numpy.random import seed
seed(1234)

# ทำการนำเข้าไฟล์ข้อมูล
def parse(x):
    return datetime.strptime(x, '%Y %m %d %H')
df = pd.read_excel('53t.xlsx', parse_dates = [['year', 'month', 'day', 'hour']], index_col=0, date_parser=parse)
# เปลี่ยนชื่อ index ให้เป็น date
df.index.name = 'date'
# แสดงค่า 5 แถวแรกของข้อมูล
print(df.head(5))
# ทำการลบช่องว่างที่อยู่ระหว่างชื่อของแต่ละคอลัมน์
df.columns = map(lambda x: x.strip(), df.columns.values)
# เปลี่ยนชื่อคอลัมน์ PM2.5 ให้เป็น PM_density
df = df.rename(columns = {'PM2.5': 'PM_density'})
df.columns
# แสดงรายละเอียดของข้อมูล
df.info()
df.describe()
# ทำการตัดข้อมูลที่มีค่า null ทั้ง
null_data = df[df.isnull().any(axis=1)]
df = df.dropna()
df.shape
# แสดงความสัมพันธ์ของแต่ละ feature
df.corr()

# พล็อตกราฟเพื่อดูค่า Trend, Seasonal และ Residual จากข้อมูล train
res = sm.tsa.seasonal_decompose(df.PM_density.dropna(), freq=365)
fig = res.plot()
fig.set_figheight(10)
fig.set_figwidth(20)
plt.show()
```

```

# ทำการแบ่ง train/ test โดยกำหนดช่วงเวลา
tr_start, tr_end = '2017-06-07 00:00:00', '2018-05-22 23:00:00'
te_start, te_end = '2018-05-23 00:00:00', '2018-06-30 23:00:00'
train_data = df['PM_density'][tr_start:tr_end].dropna().resample('H').mean()
test_data = df['PM_density'][te_start:te_end].dropna().resample('H').mean()
train_data = train_data.fillna(train_data.bfill())
test_data = test_data.fillna(test_data.bfill())
# กำหนด feature ที่จะใช้เป็น exog เพื่อใช้ในการเพิ่มประสิทธิภาพในการทำนายของแบบจำลอง
exo_data = df.drop(['PM_density'], axis=1)[tr_start:tr_end].dropna().resample('H').mean()
exo_data = exo_data.fillna(exo_data.bfill())

# กำหนดค่าพารามิเตอร์ p, d, q ในแบบจำลอง SARIMAX และใช้ grid search หาพารามิเตอร์ที่ให้ค่า AIC อ่อนนาคที่สุด
p = range(0, 4)
d = range(0, 2)
q = range(0, 4)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 24) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

# ทำการ train แบบจำลอง SARIMAX โดยกำหนดพารามิเตอร์ดังนี้
model = SARIMAX(train_data, exog=exo_data, order=(3, 1, 1), seasonal_order=(3, 1, 1, 24), enforce_stationarity=False, enforce_invertibility=False)
model_fit = model.fit()
print(model_fit.summary().tables[1])

# กำหนด exog forecast เพื่อใช้ในการทำนาย
tr_start1, tr_end1 = '2018-05-22 00:00:00', '2018-05-22 23:00:00'
exog_forecast = df.drop(['PM_density'], axis=1)[tr_start1:tr_end1].dropna().resample('H').mean()
exog_forecast = exog_forecast.fillna(exog_forecast.bfill())

# กำหนดว่าจะทำนายค่าในอีก 24 ชั่วโมงข้างหน้า
pred_uc = model_fit.get_forecast(steps=24, exog=exog_forecast)
pred_ci = pred_uc.conf_int()
print(pred_ci)

# พล็อตกราฟเปรียบเทียบระหว่างค่าที่ได้จากการทำนายกับค่าจริง
ax = train_data['2018-05-20:'].plot(label='observed', figsize=(25, 10), color='indianred')
pred_uc.predicted_mean.plot(ax=ax, label='Forecast', color='midnightblue')
test_data['2018-05-23 00:00:00: 2018-05-23 23:00:00'].plot(label='test_data', color='indianred')

ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel('PM2.5 (µg/m3)')

plt.legend()
plt.show()

# ทำการประเมินประสิทธิภาพของแบบจำลองโดยใช้ MSE, RMSE และ MAE
y_forecasted = pred_uc.predicted_mean
y_truth = test_data['2018-05-23 00:00:00:']
mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2)))
print('The Root Mean Squared Error of our forecasts is {}'.format(round(np.sqrt(mse), 2)))
mae = np.mean(np.abs(y_truth - y_forecasted))
print('The Mean Absolute Error of our forecasts is {}'.format(mae))

```

ประวัติผู้เขียน

ชื่อ-สกุล นางสาวกานต์กมล ทวีผล
วัน เดือน ปี เกิด 30 ธันวาคม 2535
สถานที่เกิด กรุงเทพมหานคร
วุฒิการศึกษา พ.ศ. 2554
วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ประยุกต์-มัลติมีเดีย
จาก มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
พ.ศ. 2560
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
จาก มหาวิทยาลัยศรีนครินทรวิโรฒ

