



WEB APPLICATION FOR CLOUD COVER ESTIMATION AND NOWCASTING

MS. JINJUTHA THAPANAPOOMPONG

MS. PATCHARAPORN SIRIMOM

MS. SAROCHA CHONMAHATRAKOO

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2022

Web application for cloud cover estimation and nowcasting

Ms. Jinjutha Thapanapoompong

Ms. Patcharaporn Sirimom

Ms. Sarocha Chonmahatrakool

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2022

Project Committee

Project Advisor

(Asst.Prof. Suthathip Maneewongvatana, Ph.D.)

J. Harnsomburana

Committee Member

(Jaturon Harnsomburana, Ph.D.)

Committee Member

(Asst.Prof. Marong Phadoongsidhi, Ph.D)

กิตติปงค์ พิyawanno

Committee Member

(Kittipong Piyawanno, Ph.D.)

| | |
|-----------------|---|
| Project Title | Web application for cloud cover estimation and nowcasting |
| Credits | 3 |
| Member(s) | Ms. Jinjutha Thapanapoompong Ms. Patcharaporn Sirimom Ms. Sarocha Chonmahatrakool |
| Project Advisor | Asst.Prof. Suthathip Maneewongvatana, Ph.D. |
| Program | Bachelor of Engineering |
| Field of Study | Computer Engineering |
| Department | Computer Engineering |
| Faculty | Engineering |
| Academic Year | 2022 |

Abstract

Nowadays, in Thailand, renewable energy and clean energy are increasingly important. The use of solar energy as a renewable energy is something to study. and developing tools to convert light energy into electrical energy for a long time until solar energy can be used as electrical energy By using an electronic device called Photovoltaic Cell. The KMUTT research team, Bang Khun Thian Campus, wants to calculate the amount of electricity produced from photovoltaic cells that can be produced each day. to forecast the efficiency of power generation of the photovoltaic cell. So, The cloud cover in the sky is a very important variable.

By estimating the cloud cover from sky imagers, there are 2 methods, The first method of estimating cloud cover by an expert specifying one by one. This will cause delays due to the large number of images and error-prone. Because each expert has different criteria for estimating cloud cover. And the second method is to use the technique of image segmentation of clouds, sky, sunlight and irrelevant areas. This method must calculate the background image of the sky for the same day and time. and estimate cloud cover from the proportion of clouds to the sky in the image, making this method take a long time to prepare large amounts of image data to be used to estimate cloud cover.

From the problems mentioned above, in this project, we will study 2 parts, The first part is to estimate cloud cover from the sky imagers. Before estimate cloud cover from the sky imagers. There must be a way to detect clouds. The authors want to study and apply deep learning with the U-Net model to detect clouds. and compared with Otsu's Method, the traditional image segmentation method, The second part is to predict the cloud cover in the future 5-30 minutes with the LSTM model compared to the Prophet model to see which model performs best. The team will then develop a web application that can calculate the amount of clouds from images. And predict cloud volume by building a model that has the best performance in detecting clouds and predicting cloud cover in the future 5-30 minutes.

Keywords: Web Application / Cloud Cover / Image Segmentation / Machine Learning / Deep Learning / Convolutional Neural Network / U-Net / Prophet / LSTM

| | |
|--------------------|---|
| หัวข้อปริญญาบัณฑิต | Web Application for cloud cover estimation and nowcasting เว็บแอปพลิเคชันสำหรับคำนวณปริมาณเมฆและพยากรณ์ปริมาณเมฆ |
| หน่วยกิต | 3 |
| ผู้เขียน | นางสาวจินท์จุฑา ฐานกุมิพงศ์ นางสาวพัชราภรณ์ ศิริม่อม นางสาวสโโรชา ชนมหาตรรภุล |
| อาจารย์ที่ปรึกษา | ผศ.ดร.สุรชาติพิย์ มณีวงศ์วัฒนา |
| หลักสูตร | วิศวกรรมศาสตรบัณฑิต |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| ภาควิชา | วิศวกรรมคอมพิวเตอร์ |
| คณะ | วิศวกรรมศาสตร์ |
| ปีการศึกษา | 2565 |

บทคัดย่อ

ปัจจุบันในประเทศไทยให้ความสำคัญทางด้านพลังงานทดแทนและพลังงานสะอาดเพิ่มมากขึ้นการนำพลังงานแสงอาทิตย์มาใช้เป็นพลังงานไฟฟ้าได้ โดยการใช้อุปกรณ์อิเล็กทรอนิกส์ที่เรียกว่า เซลล์โฟโตวอลเทอิก (Photovoltaic cell) ที่มีวิจัยของ วิทยาเขตบางขุนเทียนต้องการคำนวณปริมาณการผลิตไฟฟ้าจากเซลล์โฟโตวอลเทอิกที่ผลิตได้ในแต่ละวัน เพื่อนำไปพยากรณ์ประสิทธิภาพในการผลิตกำลังไฟฟ้าของเซลล์โฟโตวอลเทอิก ซึ่งปริมาณเมฆปกคลุมบนท้องฟ้าที่เป็นตัวแปรสำคัญอย่างยิ่ง

โดยวิธีการประมาณปริมาณเมฆจากรูปภาพถ่ายท้องฟ้าที่มี 2 วิธี ได้แก่ วิธีแรกประมาณปริมาณเมฆโดยผู้เชี่ยวชาญรูปที่ลีรูป ซึ่งวิธีนี้จะทำให้เกิดความล่าช้าเนื่องจากรูปภาพมีจำนวนมากและความผิดพลาดได้ง่าย เพราะผู้เชี่ยวชาญแต่ละบุคคลมีเกณฑ์ในการประมาณปริมาณเมฆไม่เท่ากัน และวิธีที่สองคือใช้เทคนิคการแบ่งช่วงเมฆภาพซึ่งเป็นวิธีการประมาณผลภาพแบบดั้งเดิม ได้แก่ พื้นที่ของเมฆ ท้องฟ้า แสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้อง เช่น อาคารบ้านเรือน เสาไฟฟ้า เป็นต้น ซึ่งวิธีนี้ต้องคำนวณหารูปพื้นหลังของท้องฟ้าในแต่ละวันและเวลาเดียวกัน และคำนวณปริมาณเมฆจากสัดส่วนของเมฆกับท้องฟ้าในรูปภาพทำให้วิธีนี้ใช้เวลานานในการเตรียมช่วงเมฆของภาพจำนวนมากเพื่อนำไปใช้ในการคำนวณปริมาณเมฆ

จากปัญหาที่กล่าวมาข้างต้นในโครงการนี้จึงศึกษาทั้งหมด 2 ส่วนได้แก่ ส่วนแรกคือการคำนวณปริมาณเมฆจากรูปภาพถ่ายท้องฟ้า โดยก่อนที่จะคำนวณปริมาณเมฆจากรูปภาพจะต้องมีวิธีที่สามารถตรวจสอบจับเมฆได้ คณะผู้จัดทำต้องการศึกษาและนำวิธีการเดียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ด้วยโมเดล U-Net มาใช้ตรวจจับเมฆ และเปรียบเทียบกับ Otsu's Method ซึ่งวิธีประมาณผลภาพแบบดั้งเดิม ส่วนที่สองคือการทำนายปริมาณเมฆในช่วงเวลา 5-30 นาทีในอนาคต คณะผู้จัดทำต้องการศึกษาและนำวิธีการเดียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ด้วยโมเดล LSTM เปรียบเทียบกับโมเดล Prophet เพื่อดูประสิทธิภาพของโมเดลใหม่ประสิทธิภาพที่ดีที่สุด จากนั้นคณะผู้จัดทำจะพัฒนาเว็บแอปพลิเคชันที่สามารถคำนวณปริมาณเมฆจากรูปภาพ และทำนายปริมาณเมฆได้จากการสร้างโมเดลที่มีประสิทธิภาพที่สุดในการตรวจจับเมฆและทำนายปริมาณเมฆในช่วง 5-30 นาทีในอนาคต

คำสำคัญ: เว็บแอปพลิเคชัน / ปริมาณเมฆปกคลุมบนท้องฟ้า / การเรียนรู้เชิงลึกโดยโครงข่ายประสาทเทียมคอนโวคุชัน สำหรับกำหนดขอบเขตอัตโนมัติ / การวิเคราะห์อนุกรมเวลาด้วย Prophet / โครงข่ายประสาทเทียมแบบวนกับสถาปัตยกรรม LSTM

กิตติกรรมประกาศ

โครงการนี้ได้รับความช่วยเหลือจาก ผศ.ดร.สุราทิพย์ มนิวงศ์วัฒนา อาจารย์ที่ปรึกษาโครงการ คณบดีคณะมนุษย์ศาสตร์และศิลปศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี (พื้นที่การศึกษาบางขุนเทียน) สำหรับข้อมูลที่ใช้ในการทำวิจัยที่เกี่ยวข้อง กับโครงการ สุดท้ายนี้ ขอขอบพระคุณบิดามารดาและครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน และคอยให้กำลังใจมาโดยตลอด

สารบัญ

หน้า

| | |
|--|-------|
| ABSTRACT | ii |
| บทคัดย่อ | iii |
| กิตติกรรมประกาศ | iv |
| สารบัญ | v |
| สารบัญตาราง | viii |
| สารบัญรูปภาพ | ix |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญ | 1 |
| 1.1.1 ประเภทของโครงงาน | 2 |
| 1.2 วัตถุประสงค์ | 2 |
| 1.3 ขอบเขตของโครงงาน | 2 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ | 2 |
| 1.5 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน | 2 |
| บทที่ 2 ทฤษฎีความรู้และงานวิจัยที่เกี่ยวข้อง | 6 |
| 2.1 ทฤษฎีความรู้ที่เกี่ยวข้อง | 6 |
| 2.1.1 การแบ่งชื่อมูลภาพโดยการหาค่าพิกัด (Image Segmentation using thresholding method) | 6 |
| 2.1.1.1 การทำ threshold ด้วยวิธีของโอซึ (Otsu's method) | 7 |
| 2.1.1.2 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) | 8 |
| 2.1.1.3 U-Net Architecture | 9 |
| 2.1.3.1 U-Net Architecture with ResNet-34 encoder | 10 |
| 2.1.1.4 การประเมินประสิทธิภาพของ U-Net Architecture | 11 |
| 2.1.1.5 อนุกรมเวลา (Time Series) | 11 |
| 2.1.1.6 Prophet | 14 |
| 2.1.1.7 Long Short Term Memory (LSTM) | 15 |
| 2.1.1.8 การประเมินประสิทธิภาพของโมเดล Prophet และ LSTM | 16 |
| 2.1.1.9 การทำ Data Normalization | 16 |
| 2.1.1.10 การแบ่งชุดข้อมูล (Train Test Split Data) | 16 |
| 2.1.1.11 ระบบฐานข้อมูล (Database System) | 18 |
| 2.1.11.1 Relational Database | 18 |
| 2.1.11.2 SQL | 18 |
| 2.2 เครื่องมือที่ใช้ในการพัฒนา | 19 |
| 2.2.1 Python | 19 |
| 2.2.2 OpenCV | 19 |
| 2.2.3 Scikit-Learn | 19 |
| 2.2.4 Prophet | 19 |
| 2.2.5 Tensorflow | 19 |
| 2.2.6 Keras | 19 |
| 2.2.7 Segmentation Models API | 19 |
| 2.2.8 Albumentations | 20 |
| 2.2.9 Django | 20 |
| 2.2.10 Django REST Framework | 20 |
| 2.2.11 PostgreSQL | 20 |

| | |
|---|-----------|
| 2.3 งานวิจัยที่เกี่ยวข้อง | 21 |
| 2.3.1 Cloud Coverage Prediction to Improve Solar Power Management | 21 |
| 2.3.2 A Deep Learning Model to Forecast Solar Irradiance Using a Sky Camera | 21 |
| 2.3.3 SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation | 21 |
| บทที่ 3 วิธีการดำเนินงาน | 22 |
| 3.1 การออกแบบระบบ (System Design) | 22 |
| 3.1.1 Use Case Diagram | 22 |
| 3.1.2 โครงสร้างระบบของเว็บแอปพลิเคชัน (Web Application Architecture) | 23 |
| 3.1.3 Database Design | 24 |
| 3.1.3.1 พจนานุกรมข้อมูล (Data dictionary) | 24 |
| 3.1.3.2 Entity Relationship Diagram (ER-Diagram) | 25 |
| 3.2 การเตรียมข้อมูล | 26 |
| 3.2.1 การเตรียมข้อมูลสำหรับการตรวจสอบเมฆติดโน้มตี | 26 |
| 3.2.1.1 ชุดข้อมูลภายนอก | 26 |
| 3.2.1.2 ชุดข้อมูลภายนอก | 26 |
| 3.2.1.3 การทำ Data Augmentation กับชุดข้อมูล WSISEG-database | 28 |
| 3.2.2 การเตรียมข้อมูลสำหรับการพยากรณ์ปริมาณเมฆ | 30 |
| 3.2.3 การออกแบบ Data Modeling | 31 |
| 3.2.3.1 การออกแบบโมเดลสำหรับตรวจสอบเมฆอัตโนมัติ | 31 |
| 3.2.3.2 การออกแบบชั้นตอนการคำนวณปริมาณเมฆจากรูปภาพ | 32 |
| 3.2.3.3 การออกแบบโมเดลสำหรับการพยากรณ์ปริมาณเมฆ | 33 |
| 3.3 การออกแบบการประเมินประสิทธิภาพของโมเดล | 35 |
| 3.3.1 การออกแบบการประเมินและเปรียบเทียบประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ | 35 |
| 3.3.2 การออกแบบการประเมินประสิทธิภาพของโมเดลตรวจสอบเมฆด้วย U-Net | 36 |
| 3.3.3 การออกแบบการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวณได้จากการผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น | 36 |
| 3.4 การออกแบบอัลกอริทึมที่ใช้ในการคำนวณปริมาณเมฆ | 37 |
| 3.5 การออกแบบ Mockup User Interface | 38 |
| 3.5.1 หน้าจอ Welcome | 38 |
| 3.5.2 หน้าจอ Nowcasting | 39 |
| 3.5.3 หน้าจอ Calculate | 40 |
| 3.5.4 หน้าจอ View image data | 41 |
| 3.5.5 หน้าจอ View weather data | 42 |
| 3.6 การออกแบบ Function | 43 |
| 3.6.1 Function Login | 43 |
| 3.6.2 Function Logout | 43 |
| 3.6.3 Function Dashboard | 43 |
| 3.6.4 Function Prediction | 43 |
| 3.6.5 Function Calculate | 43 |
| 3.6.6 Function Insert to database | 43 |
| 3.6.7 Function Table | 43 |
| 3.6.8 Function Show popup image in Table | 43 |
| บทที่ 4 ผลการดำเนินงาน | 44 |

| | |
|---|-----|
| 4.1 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆ (Cloud detection) | 44 |
| 4.1.1 ผลการทดลองของวิธี Otsu's method ตรวจจับเมฆ | 44 |
| 4.1.2 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net without Data augmentation | 44 |
| 4.1.3 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net using encoder ResNet34 without Data augmentation | 46 |
| 4.1.4 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net with Data augmentation | 47 |
| 4.1.5 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net using encoder ResNet34 with Data augmentation | 48 |
| 4.1.6 สรุปผลประสิทธิภาพของโมเดลตรวจจับเมฆ | 49 |
| 4.1.7 ตัวอย่างผลการตรวจจับเมฆจากชุดข้อมูลจร.วิทยาเขตบางเขนเที่ยนด้วยโมเดล U-Net | 50 |
| 4.2 ผลการทดลองของโมเดลสำหรับพยากรณ์ปริมาณเมฆ (Cloud cover nowcasting) | 51 |
| 4.2.1 ผลทดลองการพยากรณ์ปริมาณเมฆด้วย Prophet | 51 |
| 4.2.1.1 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นนาที | 51 |
| 4.2.1.2 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นชั่วโมง | 53 |
| 4.2.1.3 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นรายวัน | 55 |
| 4.2.1.4 การแบ่งชุดข้อมูลหลายตัวแปรโดยใช้ตัวแปร Cloud Cover, Relative humidity, Temperature และ Wind speed ที่เป็นนาที | 55 |
| 4.2.2.2 ผลทดลองของการพยากรณ์ปริมาณเมฆด้วย LSTM | 57 |
| 4.2.2.1 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นนาที | 57 |
| 4.2.2.2 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นชั่วโมง | 76 |
| 4.2.2.3 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นรายวัน | 78 |
| 4.2.2.4 การแบ่งชุดข้อมูลหลายตัวแปรโดยใช้ตัวแปร Cloud Cover, Relative humidity, Temperature และ Wind speed ที่เป็นนาที | 79 |
| 4.3 สรุปผลประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ | 90 |
| 4.4 การพัฒนา User Interface | 92 |
| 4.4.1 หน้าจอ Welcome | 92 |
| 4.4.2 หน้าจอ Nowcasting | 93 |
| 4.4.3 หน้าจอ Estimate Cloud Cover | 94 |
| 4.4.4 หน้าจอ View image data | 96 |
| 4.4.5 หน้าจอ View weather data | 98 |
| 4.5 ผลการทดสอบการทำงาน | 99 |
| บทที่ 5 บทสรุป | 100 |
| 5.1 สรุปผลการทดลองโมเดลตรวจจับเมฆ | 100 |
| 5.1.1 สรุปผลการทดลองตรวจจับเมฆด้วย Otsu's method | 100 |
| 5.1.2 สรุปผลการทดลองโมเดลตรวจจับเมฆด้วย U-Net | 100 |
| 5.2 สรุปผลการทดลองโมเดลการพยากรณ์ปริมาณเมฆ | 100 |
| 5.2.1 สรุปผลการทดลองของ Prophet | 100 |
| 5.2.2 สรุปผลการทดลองของ LSTM | 100 |
| 5.2.3 สรุปผลการเรียนเที่ยบระหว่าง Prophet และ LSTM | 101 |
| 5.2.4 สรุปผลการพัฒนาเว็บแอปพลิเคชัน | 101 |
| 5.3 ปัญหาที่พบและการแก้ไข | 101 |
| 5.4 แนวทางในการพัฒนาโครงการ | 101 |
| หนังสืออ้างอิง | 102 |

สารบัญตาราง

| ตารางที่ | หน้า |
|---|------|
| 1.1 ตารางแสดงขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงานภาคเรียนที่ 1 | 4 |
| 1.2 ตารางแสดงขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงานภาคเรียนที่ 2 | 5 |
| 3.1 User Table | 24 |
| 3.2 TimeCloudCover Table | 24 |
| 3.3 ImageCloudCover Table | 24 |
| 3.4 ตารางอธิบายชื่อย่อและชื่อเต็มของฟีเจอร์ที่ใช้ในการพยากรณ์ปริมาณเมฆ | 35 |
| 3.5 ตารางประเมินประสิทธิภาพของโมเดล LSTM ในแต่ละ lags | 35 |
| 3.6 ตารางประเมินประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ | 35 |
| 3.7 ตารางประเมินประสิทธิภาพของโมเดลของโมเดลตรวจจับเมฆด้วย U-Net | 36 |
| 3.8 ตารางการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวนได้จากผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น | 36 |
| 4.1 ตารางประเมินประสิทธิภาพของโมเดลตรวจจับเมฆ | 49 |
| 4.2 ตารางการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวนได้จากผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น | 49 |
| 4.3 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 5 นาที | 57 |
| 4.4 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 5 นาที | 59 |
| 4.5 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 10 นาที | 60 |
| 4.6 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 10 นาที | 62 |
| 4.7 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 15 นาที | 63 |
| 4.8 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 15 นาที | 65 |
| 4.9 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 20 นาที | 66 |
| 4.10 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 20 นาที | 68 |
| 4.11 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 25 นาที | 69 |
| 4.12 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 25 นาที | 71 |
| 4.13 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 30 นาที | 72 |
| 4.14 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 30 นาที | 74 |
| 4.15 ตารางแสดงโมเดล LSTM ที่มีประสิทธิภาพดีที่สุดของการแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ในแต่ละ lags | 90 |
| 4.16 ตารางแสดงโมเดล LSTM ที่มีประสิทธิภาพดีที่สุดของการแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น Test set ในแต่ละ lags | 91 |
| 4.17 ตารางประเมินประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ | 91 |
| 4.18 ตารางการแสดงผลการทดสอบการทำงานของหน้าจอต่างๆ ภายในเว็บแอปพลิเคชัน | 99 |

สารบัญ

| รูปที่ | หน้า |
|--|------|
| 1.1 การเปรียบเทียบของการตัดกรายทบทองปริมาณแสงในเมฆแต่ละระดับ | 1 |
| 2.1 Histogram ของค่า threshold ของแบคทีเรียชนิดหนึ่ง | 6 |
| 2.2 ภาพด้านซ้ายเป็นภาพต้นฉบับ และภาพด้านขวาเป็นภาพที่ทำ threshold | 7 |
| 2.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) | 8 |
| 2.4 สถาปัตยกรรม U-Net (U-Net Architecture) | 9 |
| 2.5 โครงข่ายสถาปัตยกรรม U-Net โดยใช้ ResNet-34 เป็น encoder | 10 |
| 2.6 โครงสร้าง ResNet-34 | 10 |
| 2.7 กราฟแสดงค่าแนวโน้มของผลผลิตเคมีภัณฑ์ชนิดหนึ่ง | 12 |
| 2.8 กราฟแสดงยอดขายรายเดือนของห้างสรรพสินค้าแห่งหนึ่ง | 12 |
| 2.9 กราฟแสดงวัฏจักรธุรกิจ | 13 |
| 2.10 โครงสร้างสถาปัตยกรรมของ LSTM | 15 |
| 2.11 กราฟแสดงการเกิด Overfitting และ Underfitting | 17 |
| 2.12 รูปการแบ่งชุดข้อมูลในการทำ Machine Learning และ Deep Learning | 17 |
| 3.1 Use Case Diagram | 22 |
| 3.2 Web application architecture diagram | 23 |
| 3.3 แสดง ER-Diagram ของทั้ง 3 ตาราง | 25 |
| 3.4 ตัวอย่างชุดข้อมูลใน WSSEG-database | 26 |
| 3.5 ตัวอย่างภาพถ่ายห้องฟ้าจากชุดข้อมูลรูปภาพถ่ายห้องฟ้า | 27 |
| 3.6 ตัวอย่างปริมาณเมฆจากรูปภาพที่คำนวณโดยผู้เชี่ยวชาญ | 27 |
| 3.7 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย RandomBrightnessContrast | 28 |
| 3.8 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย CLAHE | 28 |
| 3.9 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย Flip | 29 |
| 3.10 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย RandomBrightnessContrast+ Flip | 29 |
| 3.11 ขั้นตอนการทำโมเดลสำหรับตรวจสอบเมฆอัตโนมัติ | 32 |
| 3.12 ขั้นตอนการทำโมเดลตรวจสอบเมฆอัตโนมัติในการหาปริมาณเมฆ | 33 |
| 3.13 ขั้นตอนการทำโมเดลสำหรับการพยากรณ์ปริมาณเมฆ | 34 |
| 3.14 อัลกอริทึมในการคำนวณปริมาณเมฆจากรูปภาพ | 37 |
| 3.15 หน้าหลักก่อนเข้าสู่ระบบของเว็บแอปพลิเคชัน | 38 |
| 3.16 หน้าหลักหลังการเข้าสู่ระบบหรือหน้าการแสดง Cloud Cover Nowcasting | 39 |
| 3.17 หน้าการแสดงผลลัพธ์ของการบริมาณเมฆด้วยรูปภาพ | 40 |
| 3.18 หน้าของการแสดงข้อมูลจากการคำนวณปริมาณเมฆ | 41 |
| 3.19 หน้าของการแสดงข้อมูลปริมาณเมฆกับตัวแปรสภาพอากาศ | 42 |
| 4.1 ผลการ prediction ของวิธี Otsu's method เปรียบเทียบกับรูปต้นฉบับ และ ground truth | 44 |
| 4.2 กราฟแสดง loss and jaccard curve ของ U-Net without data augmentation | 45 |
| 4.3 ผลการ prediction ของโมเดล U-Net without data augmentation เปรียบเทียบกับ Ground truth | 45 |
| 4.4 กราฟแสดง loss and jaccard curve ของ U-Net using encoder ResNet34 without Data augmentation | 46 |
| 4.5 ผลการ prediction ของโมเดล U-Net using ResNet34 without Data augmentation เปรียบเทียบกับ Ground truth | 46 |
| 4.6 กราฟแสดง loss and jaccard curve ของ U-Net with data augmentation | 47 |
| 4.7 ผลการ prediction ของโมเดล U-Net with data augmentation เปรียบเทียบกับ Ground truth | 47 |
| 4.8 กราฟแสดง loss and jaccard curve ของ U-Net using encoder ResNet34 with Data augmentation | 48 |
| 4.9 ผลการ prediction ของโมเดล U-Net using ResNet34 with data augmentation เปรียบเทียบกับ Ground truth | 48 |
| 4.10 ตัวอย่างรูปผลการตรวจจับเมฆจากชุดข้อมูล骂.วิทยาเขตบางขุนเทียนตัวอย่างโมเดล U-Net | 50 |
| 4.11 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20% | 51 |
| 4.12 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูลที่เลือกวันสุดท้ายเป็น test set | 52 |

| | | |
|------|---|----|
| 4.58 | กราฟแสดงผลการทำนายเป็นนาที Lags 30 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20% | 89 |
| 4.59 | การขยายกราฟ 4.58 | 89 |
| 4.60 | กราฟแสดงผลการทำนายเป็นนาที Lags 30 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set | 90 |
| 4.61 | หน้าหลักก่อนเข้าสู่ระบบของเว็บแอปพลิเคชัน | 92 |
| 4.62 | การแสดงผลของข้อความพิเศษเฉพาะเมื่อเข้าสู่ระบบไม่สำเร็จ | 92 |
| 4.63 | หน้าหลักหลังการเข้าสู่ระบบหรือหน้าการแสดง Cloud Cover Nowcasting | 93 |
| 4.64 | หน้าการทำปริมาณเมฆด้วยรูปภาพ | 94 |
| 4.65 | หน้าการทำปริมาณเมฆด้วยรูปภาพเมื่อกดปุ่ม Browse | 94 |
| 4.66 | หน้าการแสดงผลภาพต้นฉบับของการทำปริมาณเมฆด้วยรูปภาพ | 95 |
| 4.67 | หน้าการแสดงผลลักษณะของการทำปริมาณเมฆด้วยรูปภาพ | 95 |
| 4.68 | หน้าการแสดงผลเมื่อบันทึกข้อมูลลงระบบฐานข้อมูลสำเร็จ | 96 |
| 4.69 | หน้าของการแสดงข้อมูลจากการคำนวนปริมาณเมฆ | 96 |
| 4.70 | การแสดงผลรูปภาพต้นฉบับเมื่อกดที่ปุ่มของรูปภาพ | 97 |
| 4.71 | การแสดงผลรูปภาพที่ทำการ Segmentation แล้วเมื่อกดที่ปุ่มของรูปภาพ | 97 |
| 4.72 | หน้าของการแสดงข้อมูลปริมาณเมฆกับตัวแปรสภาพอากาศ | 98 |

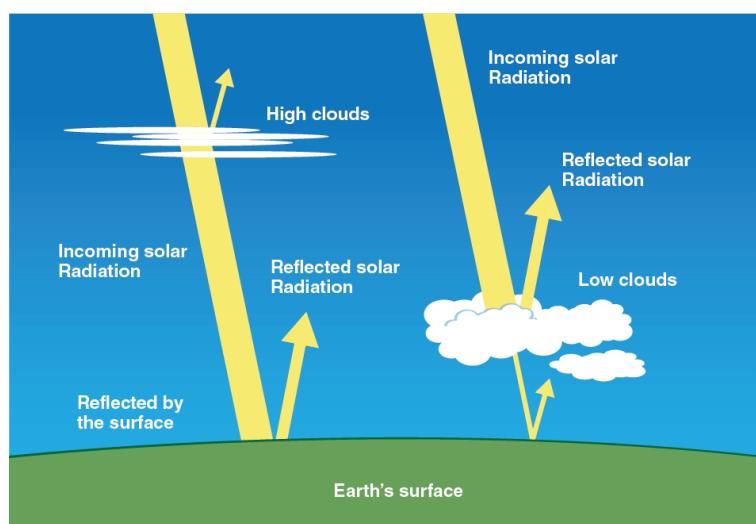
บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันในประเทศไทยให้ความสำคัญทางด้านพลังงานทดแทนและพลังงานสะอาดเพิ่มมากขึ้นการนำพลังงานแสงอาทิตย์มาใช้เป็นพลังงานทดแทนเป็นสิ่งที่นักวิจัยศึกษา และพัฒนาเครื่องมือในการเปลี่ยนพลังงานแสงมาเป็นพลังงานไฟฟ้ามาอย่างยาวนานจนสามารถนำพลังงานแสงอาทิตย์มาใช้เป็นพลังงานไฟฟ้าได้โดยการใช้อุปกรณ์อิเล็กทรอนิกส์ที่เรียกว่าเซลล์โฟโตวอลเทอิก (Photovoltaic Cell) ที่มีวิถีของวงจร วิทยาเขตบางขุนเทียนต้องการคำนวณปริมาณการผลิตไฟฟ้าจากเซลล์โฟโตวอลเทอิกที่ผลิตได้ในแต่ละวัน เพื่อนำไปพยากรณ์ประสิทธิภาพในการผลิตกำลังไฟฟ้าของเซลล์โฟโตวอลเทอิก แต่ในการพยากรณ์ประสิทธิภาพนี้ จำเป็นจะต้องคำนึงถึงสภาพอากาศต่าง ๆ เช่นมาเกียร์วัชช์ เช่น อุณหภูมิ ความชื้น ความเข้มของแสงจากดวงอาทิตย์ และปริมาณเมฆปกคลุมบนท้องฟ้า เป็นต้น ปัจจัยเหล่านี้มีผลต่อการพยากรณ์ประสิทธิภาพในการผลิตกำลังไฟฟ้าของเซลล์โฟโตวอลเทอิกในการผลิตไฟฟ้าเป็นอย่างมาก ซึ่งปริมาณการผลิตไฟฟ้าของเซลล์โฟโตวอลเทอิกแปรผันตรงกับปริมาณแสงอาทิตย์ที่ได้รับหากเซลล์โฟโตวอลเทอิกรับพลังงานแสงอาทิตย์ได้มาก ทำให้เซลล์โฟโตวอลเทอิกผลิตพลังงานได้อย่างเต็มประสิทธิภาพ แต่ถ้าเซลล์โฟโตวอลเทอิกถูกเมฆคลอยด์ตัวบดบังตามรูปที่ 1.1 พลังงานแสงอาทิตย์ก็จะไปไม่ถึงเซลล์โฟโตวอลเทอิกทำให้เซลล์โฟโตวอลเทอิกผลิตพลังงานได้น้อยลงปริมาณครึ่งหนึ่งของเซลล์โฟโตวอลเทอิกทั้งหมด

โดยครูงานนี้จะศึกษาการประมาณปริมาณเมฆจากรูปภาพถ่ายท้องฟ้า เพราะเป็นตัวแปรหนึ่งที่สำคัญในการพยากรณ์ประสีทวีภพ ในการผลิตกำลังไฟฟ้าของเซลล์โฟโตวอลเทอิก ในการถ่ายรูปภาพท้องฟ้าจะใช้กล้องลักษณะเลนส์ฟิชอาย (Fisheye Lens) ชนิดเดียวกัน กับที่ใช้ในการถ่ายภาพดาว太空 วิธีการประมาณปริมาณเมฆจากรูปภาพถ่ายท้องฟ้ามี 2 วิธี ได้แก่ ประมาณปริมาณเมฆโดยผู้เชี่ยวชาญ ระบุที่ละรูป ซึ่งวิธีนี้จะทำให้เกิดความล่าช้าเนื่องจากรูปภาพมีจำนวนมากและความผิดพลาดได้ง่าย เพราะผู้เชี่ยวชาญแต่ละบุคคลมีภูมิคุณที่ในการประมาณปริมาณเมฆไม่เท่ากัน และวิธีที่สองคือใช้เทคนิคการแบ่งชั้นมุลภาพซึ่งเป็นวิธีการประมาณผลภาพแบบดั้งเดิม ได้แก่ พื้นที่ของเมฆ ท้องฟ้า แสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้อง เช่น อาคารบ้านเรือน เสาไฟฟ้า เป็นต้น ซึ่งวิธีนี้จะต้องคำนวณหารูปพื้นหลังของ ท้องฟ้าในแต่ละวันและเวลาเดียวกัน และคำนวณปริมาณเมฆจากสัดส่วนของเมฆกับท้องฟ้าในรูปภาพทำให้วิธีนี้ใช้เวลานานในการเตรียม ข้อมูลของภาพจำนวนมากเพื่อนำไปใช้ในการคำนวณปริมาณเมฆ

ดังนั้นกลุ่มของคนผู้จัดทำจึงต้องการใช้วิธีการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Deep Learning) เพื่ามาช่วยแบ่งข้อมูลภาพส่วนของเมฆ ท้องฟ้า แสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้องโดยไม่คำนวณหารูปพื้นหลังของท้องฟ้าในแต่ละวันและเวลาเดียวกันในแต่ละรูปเพื่อมาช่วยการตรวจจับเมฆแบบอัตโนมัติ (Cloud Detection) เพื่อนำไปคำนวณปริมาณเมฆจากรูปภาพ และนำข้อมูลที่ได้ในลักษณะของลำดับเวลา มาพยากรณ์ปริมาณเมฆในช่วงเวลา 5 ถึง 30 นาทีในอนาคต (Cloud Cover Nowcasting) จากนั้นนำโมเดลที่ตรวจจับเมฆแบบอัตโนมัติ และโมเดลที่ใช้สำหรับพยากรณ์ปริมาณเมฆนำไปให้ทีมนักวิจัยมาระดับ วิทยาเขตบางขุนเทียนใช้งานบนเว็บแอปพลิเคชัน



รูปที่ 1.1 การเปรียบเทียบของการตอกกระดาษของปริมาณแสงในเมฆและระดับ

[ที่มา : <https://sciijinks.gov/solar-energy-and-clouds/>]

1.1.1 ประเภทของโครงการ

โครงการประเภทการวิจัย (Research Oriented)

1.2 วัตถุประสงค์

- เพื่อศึกษาและสร้างโมเดลที่สามารถตรวจสอบเมฆอัตโนมัติ (Cloud Detection) จากรูปภาพ ด้วยโมเดล U-Net และนำไปคำนวณหาปริมาณเมฆจากรูปภาพ
- เพื่อศึกษาและเปรียบเทียบโมเดลสำหรับการพยากรณ์ปริมาณเมฆ (Cloud cover nowcasting) ในช่วงเวลา 5 ถึง 30 นาทีในอนาคต ระหว่างโมเดล Prophet และโมเดล Long Short-Term Memory (LSTM)
- เพื่อจัดทำเว็บแอปพลิเคชันที่ใช้ในการคำนวณปริมาณเมฆจากรูปภาพ และพยากรณ์ปริมาณเมฆในช่วงเวลา 5 ถึง 30 นาทีในอนาคต

1.3 ขอบเขตของโครงการ

- พัฒนาเว็บแอปพลิเคชัน โดยแบ่งออกเป็น 2 ฟีเจอร์ ได้แก่ ผู้ใช้งานสามารถคำนวณปริมาณเมฆจากรูปภาพและผู้ใช้สามารถดูการพยากรณ์ปริมาณเมฆที่จะเกิดขึ้นได้ภายในช่วงเวลา 5 ถึง 30 นาทีในอนาคต
- ระบบสามารถคำนวณปริมาณเมฆจากรูปภาพ โดยรูปภาพที่ใช้จะในการคำนวณปริมาณเมฆต้องเป็นรูปภาพชนิด .jpg เท่านั้น
- ในการตรวจจับเมฆอัตโนมัติเพื่อนำไปหาปริมาณเมฆจากรูปภาพ ขนาดของรูปที่อัปโหลดจะต้องมีขนาดกว้างและยาว 2072x1411 pixels เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

เว็บแอปพลิเคชันที่สามารถให้ผู้ใช้สามารถคำนวณปริมาณเมฆจากรูปภาพ และสามารถดูผลลัพธ์การพยากรณ์ปริมาณเมฆที่จะเกิดขึ้นได้ภายในช่วงเวลา 5 ถึง 30 นาทีในอนาคต เพื่อให้ทีมนักวิจัยมจธ.วิทยาเขตบางขุนเทียนใช้งานได้สะดวกมากที่สุด

1.5 ขั้นตอนการทำงานและระยะเวลาการดำเนินงาน

- กำหนดหัวข้อและขอบเขตของโครงการ
- ศึกษาค้นคว้าความรู้ ทฤษฎี และงานวิจัยที่เกี่ยวข้อง
 - ศึกษาวิธีการที่ใช้คำนวณปริมาณเมฆจากกาวิจัยที่มีจด.วิทยาเขตบางขุนเทียน และศึกษางานวิจัยภายนอกที่ทำกับงานที่เกี่ยวข้อง เช่น การทำงานสภาพอากาศ การคำนวณปริมาณเมฆจากรูปภาพ เป็นต้น
 - ศึกษาทฤษฎีอื่น ๆ ที่เคยถูกนำมาใช้ในขอบเขตการวิจัยเดียว กับ ศึกษาทฤษฎีเดียวกันนี้เพื่อใช้เป็นข้อมูลในการนำมาพิจารณาที่จะสามารถนำมาประยุกต์ใช้ในโครงการได้
 - ศึกษารูปแบบการพัฒนาเว็บแอปพลิเคชันสำหรับการคำนวณเมฆจากรูปภาพ และการพยากรณ์ปริมาณเมฆ
- จัดทำแบบเสนอโครงการ (Project Proposal)
- การรวบรวมข้อมูล เตรียมข้อมูล และการศึกษางานที่เกี่ยวข้องกับส่วนของโมเดล
 - รวบรวมข้อมูล (Data Gathering) จากนักวิจัยมจธ.วิทยาเขตบางขุนเทียนสำหรับการคำนวณปริมาณเมฆด้วยรูปภาพ และสำหรับการพยากรณ์ปริมาณเมฆ
 - หาชุดข้อมูลภายนอกที่เป็นแหล่ง open source ที่เป็นภาพถ่ายท้องฟ้าที่มีการ Label ในรูปภาพที่ระบุส่วนของเมฆ ท้องฟ้า แสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้องไว้แล้วเพื่อมาใช้тренโมเดล U-Net และนำไปใช้กับชุดข้อมูลของนักวิจัยมจธ.วิทยาเขตบางขุนเทียน

4.3 เตรียมข้อมูล (Data Preparation) เพื่อปรับให้ข้อมูลอยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ได้ และการทำให้ข้อมูลมีความเหมาะสมที่ถูกนำไปใช้ได้ เช่น การจัดการกับข้อมูลที่ขาดหายไป การแก้ไขข้อมูลที่ผิดพลาด การปรับสเกลของข้อมูลให้อยู่ในรูปแบบเดียวกัน เป็นต้น

4.4 ศึกษาการพยากรณ์ปริมาณเมฆด้วยโมเดล Prophet และโมเดล LSTM

4.5 ศึกษาการทำตรวจจับเมฆอัตโนมัติ (Cloud Detection) ด้วยโมเดล U-Net

4.6 ออกแบบ User Interface ของเว็บแอปพลิเคชัน

4.7 ศึกษาการสร้างเว็บแอปพลิเคชันด้วย Django

5. จัดทำรายงานภาคเรียนที่ 1

6. จัดทำโมเดลตรวจจับเมฆอัตโนมัติ (Cloud Detection)

7. จัดทำโมเดลพยากรณ์ปริมาณเมฆ (Cloud Cover Nowcasting)

8. ทดสอบโมเดลตรวจจับเมฆอัตโนมัติ และโมเดลพยากรณ์ปริมาณเมฆ

9. ปรับปรุงโมเดลตรวจจับเมฆอัตโนมัติ และโมเดลพยากรณ์ปริมาณเมฆ

10. พัฒนาและทดสอบเว็บแอปพลิเคชัน

11. สรุปผลการดำเนินงาน และจัดทำรายงานภาคเรียนที่ 2

จากขั้นตอนการดำเนินงานข้างต้น สามารถแสดงขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงานได้ดังตารางที่ [1.1](#) และตารางที่ [1.2](#)

ตารางที่ 1.1 ตารางแสดงขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงานภาคเรียนที่ 1

| Task/Week | AUGUST | | | | SEPTEMBER | | | | OCTOBER | | | | NOVEMBER | | | | DECEMBER | | | | |
|---|--------|---|---|---|-----------|---|---|---|---------|---|---|---|----------|---|---|---|----------|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | |
| กำหนดวิวัฒนาและขอบเขตของโครงการ | ● | | | | | | | | | | | | | | | | | | | | |
| จัดทำ Idea Document | | ● | ● | | | | | | | | | | | | | | | | | | |
| จัดทำ Proposal Document | | | | ● | | | | | | | | | | | | | | | | | |
| ศึกษางานจากนักวิจัย รวบรวมข้อมูล และเตรียมข้อมูล | ● | ● | ● | ● | | ● | ● | ● | | | | | | | | | | | | | |
| ศึกษางานวิจัยที่เกี่ยวข้อง | | | | | | | | | | | | | ● | ● | ● | ● | | | | | |
| ศึกษาเครื่องมือที่ใช้ในการทำโครงการ | | | | | | | | | | | | | ● | ● | ● | ● | | | | | |
| ศึกษาการพยากรณ์ปริมาณเมฆด้วยโนมแอล Prophet และโนมแอล LSTM | | | | | | | | | | | | | | | | | ● | ● | ● | ● | |
| ศึกษาการท่าตรวจจับเมฆอุดโน้ม (Cloud Detection) ด้วยโนมแอล | | | | | | | | | | | | | | | | | ● | ● | ● | ● | |
| ออกแบบ User Interface ของเว็บแอปพลิเคชัน | | | | | | | | | | | | | | | | | ● | ● | ● | ● | |
| ศึกษาการสร้างเว็บแอปพลิเคชันด้วย Django | | | | | | | | | | | | | | | | | ● | ● | ● | ● | |
| จัดทำรายงานภาคเรียนที่ 1 | | | | | | | | | | | | | | | | | | ● | ● | ● | ● |
| นำเสนอบรรيجาภาคเรียนที่ 1 | | | | | | | | | | | | | | | | | | ● | ● | ● | ● |

ตารางที่ 1.2 ตารางแสดงขั้นตอนการดำเนินงานและระยะเวลาการดำเนินงานภาคเรียนที่ 2

บทที่ 2 ทฤษฎีความรู้และงานวิจัยที่เกี่ยวข้อง

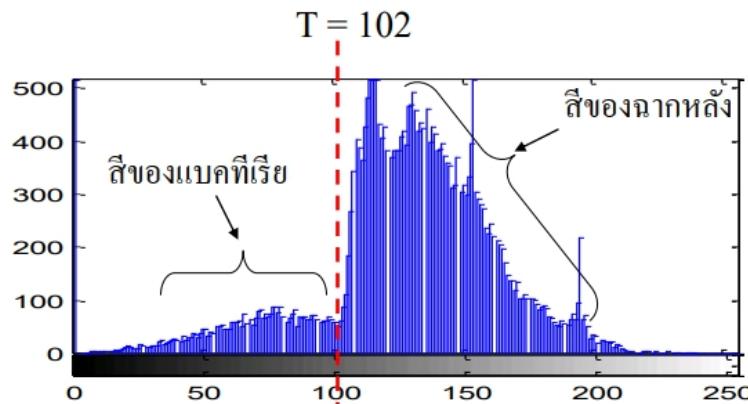
2.1 ทฤษฎีความรู้ที่เกี่ยวข้อง

2.1.1 การแบ่งข้อมูลภาพโดยการหาค่าพิกัด (Image Segmentation using thresholding method)

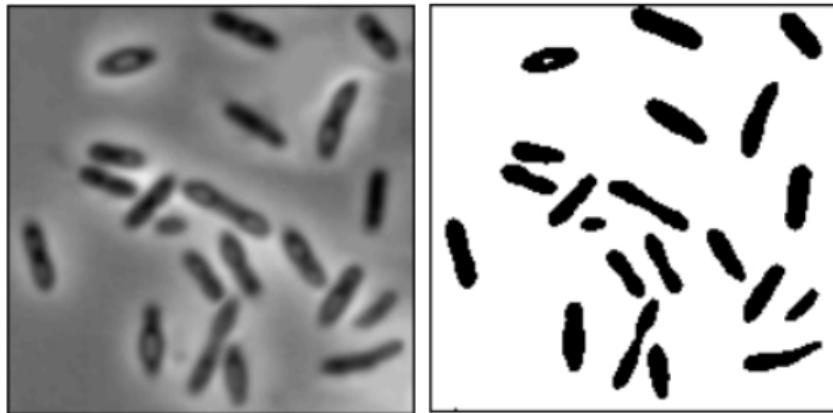
การแบ่งข้อมูลภาพ (Image Segmentation) เป็นการแบ่งข้อมูลภาพออกเป็นส่วนย่อย ๆ ที่แต่ละส่วนจะมีพื้นที่ต่อ กันไปแต่ละส่วนจะเป็นพื้นที่ซึ่งอาจจะแทนวัตถุที่อยู่ในภาพเช่นเมื่อวัตถุที่ต้องการถูกแบ่งออกได้สมบูรณ์ ผลลัพธ์ที่ได้จากการเชือกเมื่น์จะเป็นตัวชี้วัดความสำเร็จในขั้นตอนการวิเคราะห์ภาพด้วยคอมพิวเตอร์ โดยเทคนิคที่ใช้การแบ่งข้อมูลภาพมีหลายวิธีและนำไปใช้กับสายงานด้านต่าง ๆ เช่น Edge-based, Motion-based, Document images, Medical images เป็นต้น [1] การเชือกเมื่น์ข้อมูลภาพเกรย์สเกลโดยหาค่าพิกัด (Grey-Scale Thresholding) เป็นวิธีแยกส่วนของภาพจากการนำภาพสีมาทำการแปลงให้เป็นภาพสีเทา แล้วนำภาพสีเทามาทำให้เป็นภาพใบหน้าหรือภาพขาวดำความสำคัญของกระบวนการทำ threshold คือ การเลือกค่าของ threshold วิธีง่ายสุดคือการเลือกเฉลี่ยหรือค่ากลางคือค่าของ threshold เท่ากับ 128 หากพิเศษของวัตถุมีความสว่างกว่าพื้นหลัง ในภาพไม่มีสัญญาณrgb กวน และมีพื้นหลังที่สงบเรียบร้อย ค่าของ threshold ที่เป็นค่าเฉลี่ยหรือค่ากลางจะทำงานได้ดี โดยการหาค่า threshold จากกราฟ Histogram ดังรูปที่ 2.1 โดยมีเงื่อนไขคือ

- พิกเซลที่มีค่ามากกว่าเท่ากับค่า threshold มีค่าเท่ากับ 1 หรือสีขาว
- พิกเซลที่มีค่าน้อยกว่าค่า threshold มีค่าเท่ากับ 0 หรือสีดำ

$$f_{thr}(x, y) = \begin{cases} 1, & f_f(x, y) < \text{Threshold} \\ 0, & f_f(x, y) \geq \text{Threshold} \end{cases}$$



รูปที่ 2.1 Histogram ของค่า threshold ของแบบที่เรียบชนิดหนึ่ง



รูปที่ 2.2 ภาพด้านข่ายเป็นภาพต้นฉบับ และภาพด้านขวาเป็นภาพที่ทำ threshold

[ที่มา : <http://staff.cs.psu.ac.th/sathit/344-671/Image%20Segmentation.pdf>]

2.1.1.1 การทำ threshold ด้วยวิธีของอ้อชี (Otsu's method)

การทำ threshold ด้วยวิธีของอ้อชีใช้ในการทำไชส์โตแกรมและทำการ threshold โดยอัดโนมัติ [2] หรือการลดความเช้มของสีเทาลง เพื่อแปลงเป็นภาพในนารี วิธีการทำงานโดยสมมติฐานว่าภาพที่จะทำ threshold มีพิกเซล 2 ชั้น หรือเป็นรูปแบบบิสโตแกรม เช่น มีพื้นหน้าและพื้นหลังนั้นคือการทำการคำนวณแยกค่า threshold ทั้ง 2 ชั้นให้เหมาะสมแล้วรวมการแพร่กระจายของภาพในระดับต่าง มีอัลกอริทึมค้นหาเกณฑ์อย่างละเอียดถี่ถ้วนเพื่อลดความแปรปรวนภายในคลาสกำหนดเป็นผลรวมของความแปรปรวนของทั้งสองคลาสแบบถ่วงน้ำหนักตามสมการที่ 2.1

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad (2.1)$$

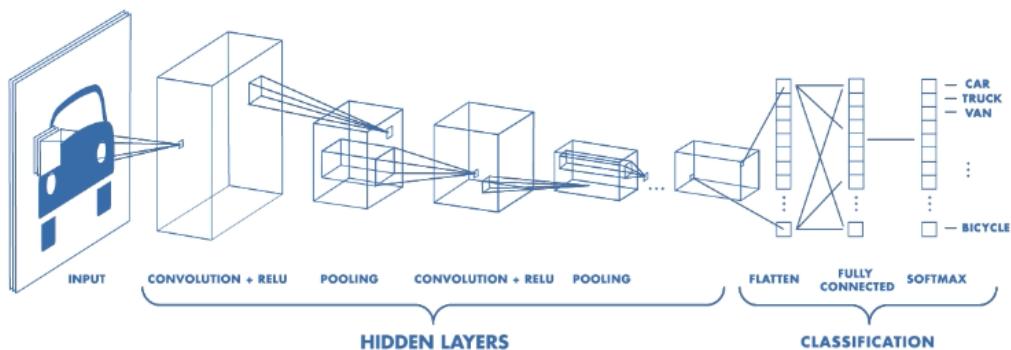
- อัลกอริทึมในการทำ threshold ด้วยวิธีของอ้อชี

1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds $t = 1, \dots$ maximum intensity
 - (a) Update $\omega_i(0)$ and $\mu_i(0)$
 - (b) Compute $\sigma_b^2(t)$
4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$

2.1.2 โครงข่ายประสาทแบบconvอลูชัน (Convolutional Neural Network)

โครงข่ายประสาทเทียมconvอลูชัน คือ Neural Network แบบหนึ่งที่มักถูกนำมาใช้ในงาน Computer Vision หรือวิเคราะห์รูปภาพ เช่น การจำแนกรูปภาพ การตรวจสอบจัตุรัส การเรียนรู้จัดลำดับในภาพ เป็นต้น และเป็นโครงข่ายประสาทเทียมชนิดหนึ่งที่มีการดำเนินการประมวลผลทางคณิตศาสตร์ที่เรียกว่า convอลูชัน โดยโครงข่ายประสาทเทียมนิดนี้ประมวลผลได้ดีกับข้อมูลที่มีความลับซ่อนอยู่ เช่น ภาพใบหน้า ภาพคน ภาพสัตว์ ฯลฯ ซึ่งมีความสามารถในการจำแนกและจัดอันดับของข้อมูลนี้ได้ดีกว่าโครงข่ายประสาทแบบเดาๆ เช่น Perceptron หรือ Multilayer Perceptron ที่ต้องคำนึงถึง特徵ที่สำคัญของข้อมูลทุกประการ แต่ conv สามารถจับจังหวะของข้อมูลที่สำคัญได้โดยอัตโนมัติ ไม่ต้องคำนึงถึง特徵ที่สำคัญของข้อมูลทุกประการ ทำให้ conv สามารถประมวลผลได้เร็วและแม่นยำกว่าโครงข่ายประสาทแบบเดาๆ

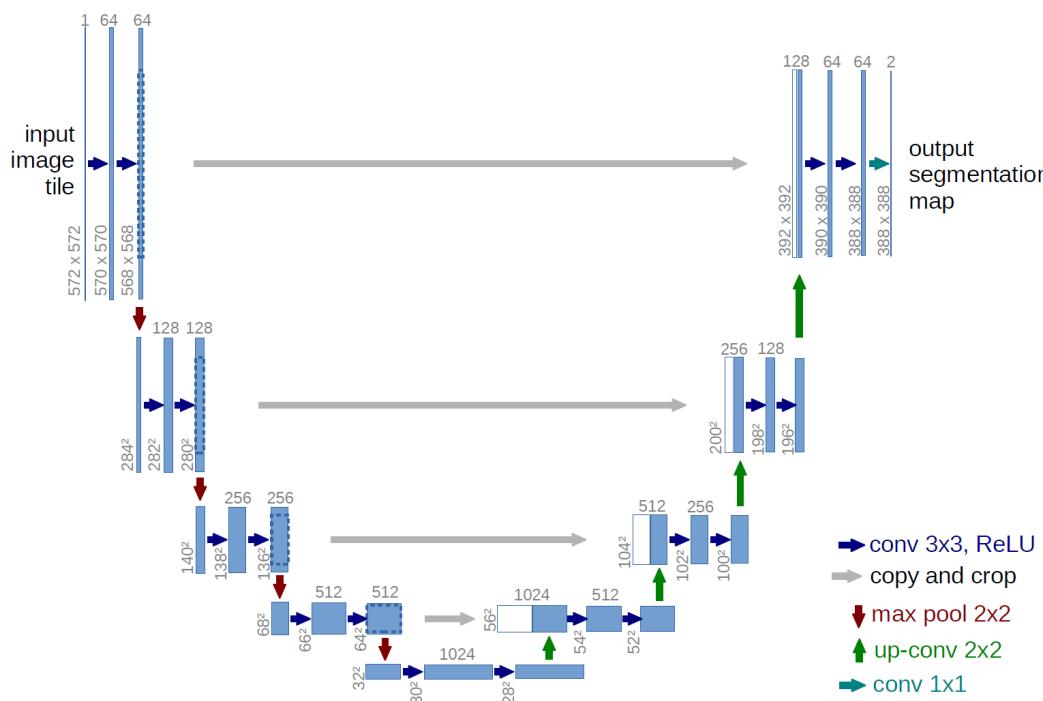
1. convolution (convolution) เป็นการทำให้รูปภาพมีความลึกซึ้งมากขึ้นโดยนำรูปภาพมาผ่าน filter จำนวนมาก ทำให้รูปภาพมีความลึกซึ้งมากขึ้น เช่น โดยปกติภาพจะมีชีวลักษณะ 3 ชีว คือ สีแดง สีเขียว และสีฟ้า แบ่งเป็น 3 ชั้น (Channel) รูปทรงของสีนั้นจะมีค่าตั้งแต่ 0-255 ในรูปภาพที่มีความลึกซึ้ง 1 ชั้นจะมีค่าตั้งแต่ 0-255 ในการทำ convolution จะใช้ filter ที่มีขนาด 3x3 ทำการ convolution กับภาพขาวดำ เพื่อเก็บค่าไว้ใน feature map (feature map)
2. Activation layer เป็นขั้นตอนในการนำผลลัพธ์ของ convolution มาผ่านฟังก์ชัน activation เช่น ReLU สำหรับขั้นตอน normalization ฟังก์ชัน softmax สำหรับขั้นตอน classification ที่มีค่าตั้งแต่ 0-1 แสดงถึงความน่าจะเป็นของผลลัพธ์ เช่น 0.8 หมายความว่าภาพนี้มีความน่าจะเป็น 80% ว่าเป็นรถ
3. pooling (pooling) เป็นการลดมิติของ feature map ให้มีขนาดเล็กลงแต่ยังคงรักษารายละเอียด ข้อมูลสำคัญของภาพไว้ การ pooling มีประโยชน์ในการลดจำนวน parameter และแก้ปัญหา overfitting การ pooling สามารถจำแนกเป็นประเภทต่างๆ ได้ เช่น pooling ด้วยค่าสูงสุด (Max Pooling) ค่าเฉลี่ย (Mean Pooling) และการ pooling ด้วยค่าสูงสุด (Max Pooling)
4. fully connected layer (fully connected layer) เป็นขั้นตอนในการนำข้อมูลที่ได้จากการ pooling ไปผ่าน fully connected layer ที่มีจำนวน node ที่กำหนด เช่น 5 หรือ 10 หรือ 20 หรือมากกว่านั้น แล้วนำผลลัพธ์มาผ่าน softmax ที่มีค่าตั้งแต่ 0-1 แสดงถึงความน่าจะเป็นของข้อมูลที่ได้รับ เช่น 0.9 หมายความว่าภาพนี้มีความน่าจะเป็น 90% ว่าเป็นรถ



รูปที่ 2.3 โครงข่ายประสาทแบบconvอลูชัน (Convolutional Neural Network)

2.1.3 U-Net Architecture

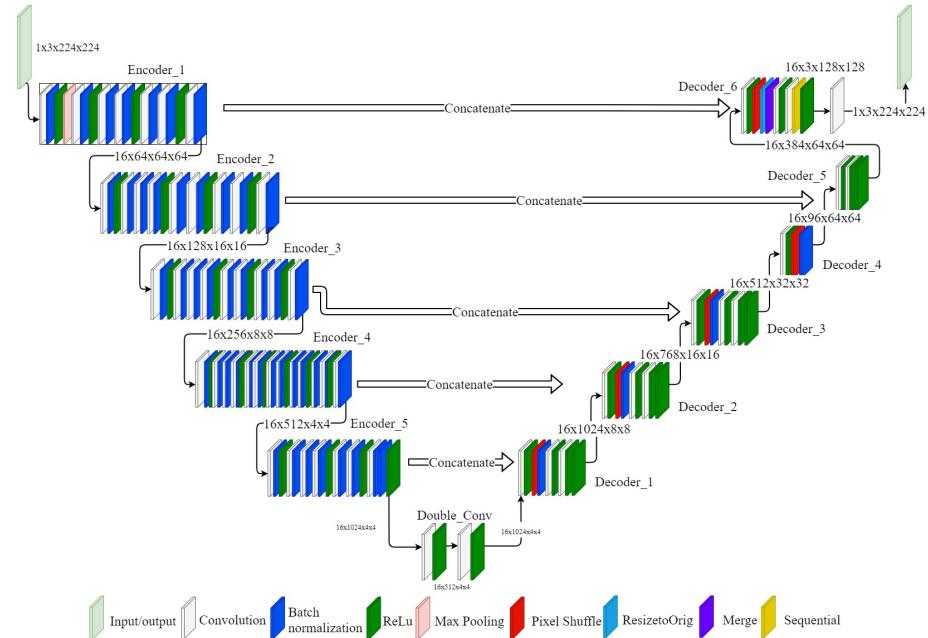
สถาปัตยกรรม U-Net พัฒนาจากสถาปัตยกรรมคอนโวลูชันแบบเต็มรูปแบบ Fully Convolutional Neural Networks (FCNN) โดยทำการปรับปรุงและขยายบางส่วนของสถาปัตยกรรม FCNN ทำให้สถาปัตยกรรม U-Net ทำงานได้แม้จะใช้ข้อมูลรูปภาพในการสอนจำนวนน้อยได้ และยังคงให้ผลลัพธ์ที่ถูกต้องแม่นยำสำหรับงานด้านการแบ่งส่วนภาพ โดยรูปแบบสถาปัตยกรรมแสดงดังรูปที่ 2.4 ประกอบไปด้วยส่วน contracting (ด้านซ้าย) และส่วน expansive (ด้านขวา) โดยส่วน contracting เป็นสถาปัตยกรรมโครงข่ายแบบคอนโวลูชัน ประกอบด้วยการประยุกต์ใช้ชั้นคอนโวลูชันขนาด 3×3 สองชั้นแต่ละชั้นตามด้วยฟังก์ชันกระตุน Rectified linear unit (ReLU) และชั้น max pooling ขนาด 2×2 โดย stride เท่ากับ 2 สำหรับการดำเนินการ downsampling โดยที่แต่ละชั้นตอนของการ downsampling มีการเพิ่มจำนวนของ feature channels เป็น 2 เท่า ซึ่งในส่วนนี้มีเป้าหมายเพื่อทำการค้นหาองค์ประกอบภาพด้วยการเข้ารหัส (Encoder) ถึงส่วนหนึ่งคือส่วน expansive ทุกชั้นตอนในส่วนนี้ประกอบด้วยการ upsampling ของ feature map โดยชั้นคอนโวลูชันขนาด 2×2 จำนวนครึ่งหนึ่งของ feature channels มีการเชื่อมต่อกัน crop feature map ที่สัมพันธ์กับส่วน contracting และชั้นคอนโวลูชันขนาด 3×3 สองชั้น แต่ละชั้นตามด้วยฟังก์ชันกระตุน ReLU ในส่วนของการ cropping เป็นสี่เหลี่ยมคางหมู เนื่องจากมีการสูญเสียพิกเซลบริเวณขอบสำหรับการคัดลอกคอนโวลูชันแต่ละครั้ง และในชั้นสุดท้ายของสถาปัตยกรรมเป็นชั้นคอนโวลูชันขนาด 1×1 ถูกนำมาใช้เพื่อทำการ map แต่ละ feature vector จำนวน 64 องค์ประกอบกับจำนวนคลาสที่ต้องการ โดยรวมแล้วทั้งโครงข่ายมีชั้นคอนโวลูชัน 23 ชั้น



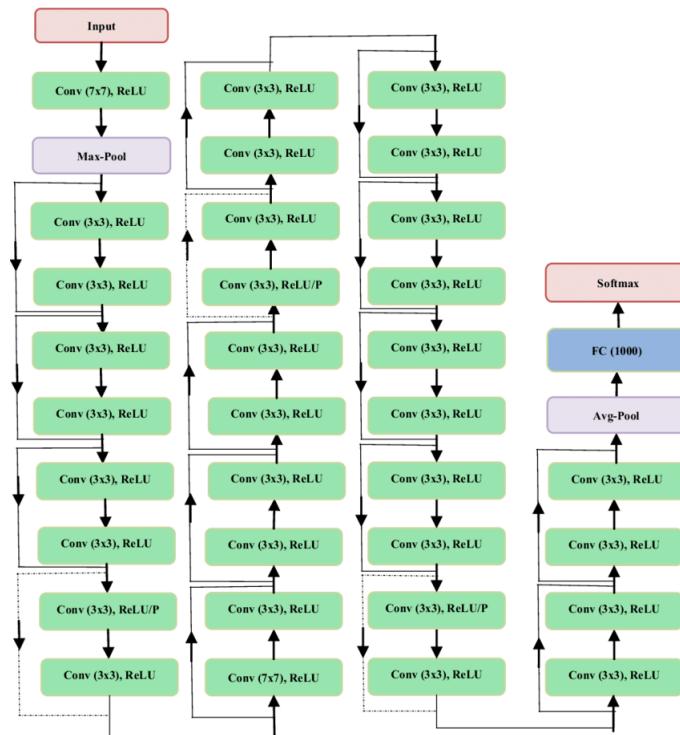
รูปที่ 2.4 สถาปัตยกรรม U-Net (U-Net Architecture)

2.1.3.1 U-Net Architecture with ResNet-34 encoder

ResNet หรือ Residual Network เป็น Deep Convolutional Neural Networks ที่ทาง Microsoft เผยแพร่ออกสู่สาธารณะตั้งแต่ปี 2015 และเป็นทางการในปี 2016 ที่ CVPR ซึ่งเป็นโมเดลที่ใช้กันยาวนานจนถึงปี 2020 (ในปัจจุบันปี 2023 ก็ยังสามารถใช้งานได้สำหรับบางงาน) เนื่องจาก U-Net เป็นสถาปัตยกรรมที่พัฒนาจากสถาปัตยกรรม Fully Convolutional Neural Networks (FCNN) ทำให้สามารถใช้ ResNet ในส่วนองค์ประกอบภาพด้วยการเข้ารหัส (Encoder) ของ U-Net ได้โดยมีรูปร่างสถาปัตยกรรมตามรูปที่ 2.5



รูปที่ 2.5 โครงข่ายสถาปัตยกรรม U-Net โดยใช้ ResNet-34 เป็น encoder



รูปที่ 2.6 โครงสร้าง ResNet-34

2.1.4 การประเมินประสิทธิภาพของ U-Net Architecture

ในส่วนนี้จะเป็นเทคนิคทั่วไปในการประเมินประสิทธิภาพของการแบ่งแยกเมฆ ซึ่งแบ่งเทคนิคออกเป็น 3 ส่วน คือ Confusion Matrix, Intersection over Union (IoU), Dice Similarity Index (DSC)

- Confusion Matrix คือการประเมินผลของการทำนาย เปรียบเทียบกับผลลัพธ์จริง ๆ ซึ่งมี attributes ดังนี้

TP คือ True Positive เป็นการทำนายว่าจริง และสิ่งที่เกิดขึ้นคือจริง

TN คือ True Negative เป็นการทำนายว่าไม่จริง และสิ่งที่เกิดขึ้นคือไม่จริง

FP คือ False Positive เป็นการทำนายว่าจริง แต่สิ่งที่เกิดขึ้นคือไม่จริง

FN คือ False Negative เป็นการทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้นคือจริง

จากนั้นจะนำ attributes เหล่านี้ไปหาค่า Accuracy, Precision, Recall และ F1-Score

- Accuracy คือ ค่าความแม่นยำเป็นตัวชี้วัดว่าโมเดลทำนายถูกต้องมากแค่ไหน ดังสมการที่ 2.2

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.2)$$

- Precision คือ ค่าความเที่ยงตรงเป็นการเปรียบเทียบการทำนายที่ถูกต้องว่าจริง และก็เกิดขึ้นจริง (TP) กับการทำนายว่าจริง แต่สิ่งที่เกิดขึ้นคือไม่จริง (FP) ดังสมการที่ 2.3

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

- Recall คือ ความถูกต้องของการทำนายว่าจะเป็นจริง เทียบกับจำนวนครั้งของเหตุการณ์ทั้งหมด แล้วเกิดขึ้นว่าเป็นจริง ดังสมการที่ 2.4

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

- F1-Score เป็นค่าเฉลี่ยแบบ harmonic mean ระหว่าง precision และ recall จุดประสงค์ของการสร้าง F1 ขึ้นมาเพื่อเป็น single metric ที่วัดความสามารถของโมเดล ดังสมการที่ 2.5

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (2.5)$$

- Intersection over Union (IoU, Jaccard index) เป็นสถิติที่ใช้เปรียบเทียบค่าความคล้ายคลึงกันของพิกเซลที่ทำนาย และพิกเซลจริง

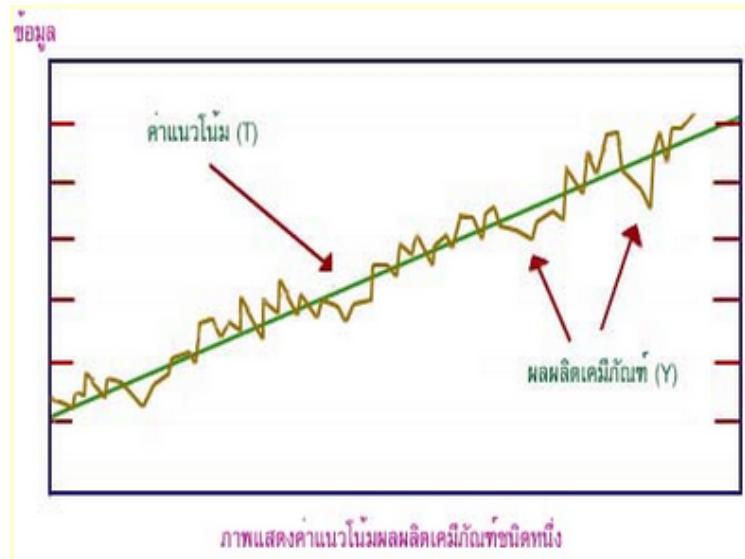
$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (2.6)$$

- Dice Similarity Coefficient เป็นสถิติที่ใช้ในการเปรียบเทียบความคล้ายคลึงกันของสองตัวอย่าง ซึ่งสองตัวอย่างในที่นี้คือพิกเซลที่ทำนาย และพิกเซลจริง

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (2.7)$$

2.1.5 อนุกรมเวลา (Time Series)

อนุกรมเวลา คือ เซตของข้อมูลเชิงปริมาณที่จัดเก็บในช่วงเวลาหนึ่ง ตัวอย่างเช่น ดัชนีตลาดหลักทรัพย์ในแต่ละวัน เมื่อปิดทำการซื้อขายในแต่ละวัน รายได้ประชาชาติ (GNP) รายได้รวม รายรับในแต่ละปีของบริษัทแห่งหนึ่ง เป็นต้น ในกรณีเคราะห์อนุกรมเวลา ผู้วิเคราะห์จะแยกองค์ประกอบต่าง ๆ ที่ประกอบกันขึ้นเป็นอนุกรมเวลา โดยจะมีการเปลี่ยนแปลงไปตามอิทธิพลต่าง ๆ เช่น การเปลี่ยนแปลงการผลิต เทคโนโลยี สภาพอากาศ เป็นต้น ในการหากลุ่มลักษณะของอนุกรมเวลาสามารถใช้แบบจำลองได้หลายแบบ แบบจำลองที่ใช้โดยนักเศรษฐศาสตร์แบบหนึ่ง คือ แบบจำลองแบบคลาสสิก (classical model) เป็นการอธิบายถึงองค์ประกอบของการแปรผันของอนุกรมเวลา 4 ส่วน ดังนี้ [3]



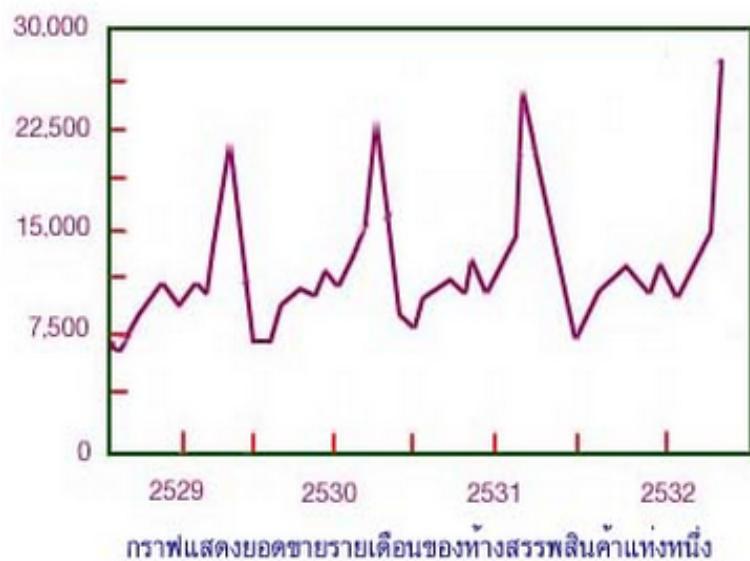
รูปที่ 2.7 กราฟแสดงค่าแนวโน้มของผลผลิตเคมีภัณฑ์ชนิดหนึ่ง

1. ค่าแนวโน้ม (Long Term Trend : T)

ค่าแนวโน้มเป็นการแสดงถึงการเคลื่อนไหวหรือเปลี่ยนแปลงของข้อมูลในระยะยาว เช่น ปริมาณการใช้ไฟฟ้าของประเทศไทย ปริมาณการนำเข้าสินค้ามันดิบ เป็นต้น ลักษณะของกราฟแนวโน้มตามรูปที่ 2.7

2. ค่าการผันแปรตามฤดูกาล (Seasonal Variation : S)

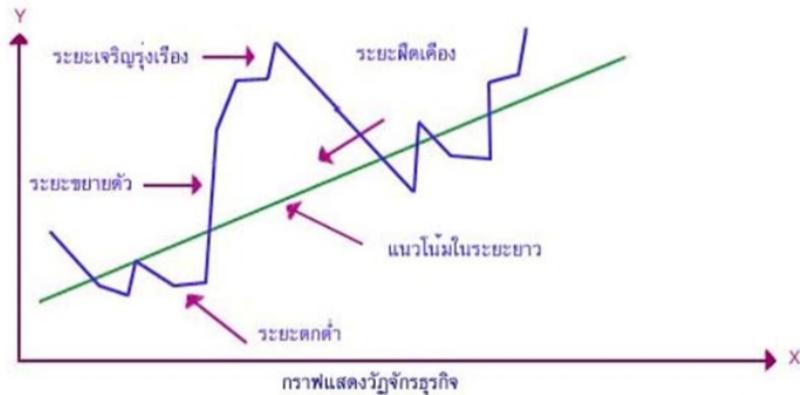
ค่าการผันแปรตามฤดูกาล หมายถึงการเปลี่ยนแปลงตามฤดูกาล โดยเกิดขึ้นซ้ำ ๆ กันในรอบ 1 ปี จนกลายเป็นแบบแผนเดียวกัน เช่น ผลผลิตหัวใจจะสูงในช่วงไตรมาสแรกของปี ยอดขายของห้างสรรพสินค้าจะสูงในช่วงปลายปี เป็นต้น ในการวิเคราะห์การผันแปรตามฤดูกาลนี้จะวัดอุกมาในรูปของตัวบ่งคัดฤดูกาล (Seasonal Index) ลักษณะของกราฟการเปลี่ยนแปลงตามฤดูกาลตามรูปที่ 2.8



รูปที่ 2.8 กราฟแสดงยอดขายรายเดือนของห้างสรรพสินค้าแห่งหนึ่ง

3. ค่าการผันแปรตามวัฏจักร (Cyclical Variation : C)

ค่าการผันแปรตามวัฏจักร หมายถึงการเคลื่อนไหวที่เป็นไปตามวัฏจักร ซึ่งการเคลื่อนไหวตามวัฏจักรนี้จะมีลักษณะคล้ายกับการผันแปรตามฤดูกาล แต่จะมีระยะเวลาที่ยาวนานกว่า ตัวอย่างเช่น วัฏจักรธุรกิจ (Business Cyclical) คือ ข้อมูลที่มีการเปลี่ยนแปลงตามวัฏจักรในทางธุรกิจโดยทั่วไปประกอบด้วยระยะเรืองรุ่งเรือง (prosperity) ระยะฝืดเคือง (recession) ระยะตกต่ำ (depression) และระยะขยายตัว (recovery) ซึ่งตัวอย่างของข้อมูลที่มีความเป็นวัฏจักรแสดงวัฏจักรธุรกิจ ลักษณะของกราฟวัฏจักรธุรกิจตามรูปที่ 2.9



รูปที่ 2.9 กราฟแสดงวัฏจักรธุรกิจ

4. การผันแปรเนื่องจากเหตุการณ์ไม่ปกติ (Irregular Variation : I)

ความผันแปรเนื่องจากเหตุการณ์ผิดปกติ คือ ลักษณะการเปลี่ยนแปลงเคลื่อนไหวของข้อมูลไม่มีรูปแบบที่แน่นอนเป็นเหตุการณ์ที่ไม่มีการคาดการณ์ล่วงหน้าซึ่งมีการเปลี่ยนแปลงที่เป็นเชิงสุ่ม (random variation) เช่น เหตุการณ์ภัยพิบัติ การเกิดอุบัติเหตุ

2.1.6 Prophet

Prophet เป็น open source library พัฒนาโดยทีม Facebook Research ซึ่งสร้างมาเพื่อการทำวิเคราะห์ Time Series และการทำนายเป็นโมเดลที่ใช้งานง่าย สะดวกในการปรับจูนและทำการ optimization โดยมีสมการการทำงานตามสมการที่ 2.8 [4]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2.8)$$

โดยที่

$g(t)$: piecewise linear or logistic growth curve for modelling non-periodic changes in time series

$s(t)$: periodic changes (e.g. weekly/yearly seasonality)

$h(t)$: effects of holidays (user provided) with irregular schedules

ϵ_t : error term accounts for any unusual changes not accommodated by the model

จากสมการที่ 2.8 ทำการ decompose ข้อมูล time series ออกเป็นองค์ประกอบ 3 ส่วนคือ Trend, seasonality และ holidays กะทัดรัดนี้เน้นไปที่ Trend เป็นส่วนสำคัญการวิเคราะห์และพยากรณ์ Equity Curve ที่เกิด เพื่อใช้สร้าง Downside Protection Model ซึ่ง Prophet ทำการโมเดล Trend ด้วย Nonlinear, Saturating Growth (logistic growth model) ตามสมการที่ 2.9

$$g(t) = \frac{C}{1 + (e^{-k(t-m)})} \quad (2.9)$$

โดยที่

C is the carrying capacity (that is the curve's maximum value).

k is the growth rate (which represents "the steepness" of the curve).

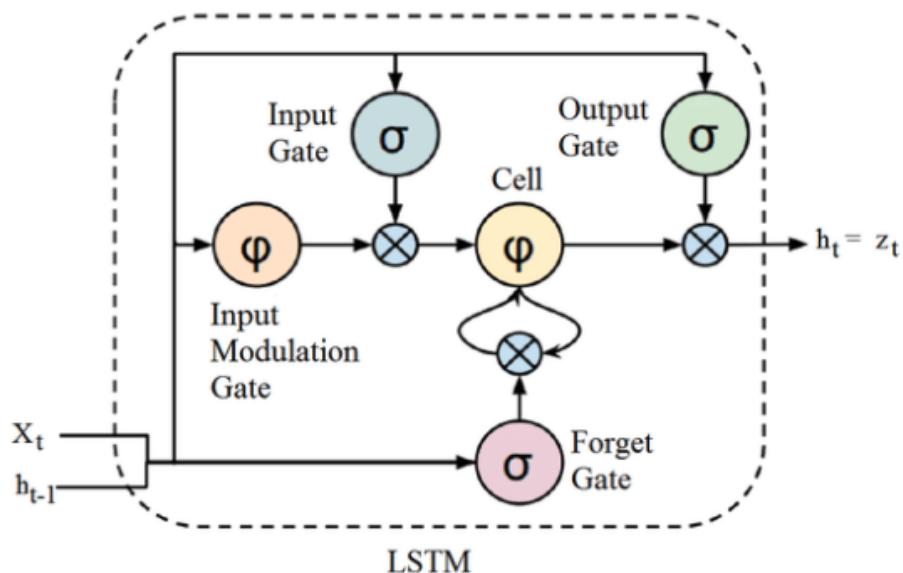
m is an offset parameter.

นอกจากสกัด Trend จากข้อมูล time series แล้ว Prophet ยังสามารถทำนาย forecasting ทิศทางของ Time Series , คำนวณหาค่า Uncertainty จากข้อมูล และสกัดเอา outliers ที่ไม่ปกติจากชุดข้อมูลได้อีกด้วย

2.1.7 Long Short Term Memory (LSTM)

LSTM หรือ Long Short Term Memory เป็นโครงข่ายประจํา RNN รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียร และมีประสิทธิภาพมากขึ้น สามารถจำลองรูปแบบความจำของคน (memory) ที่มีความจุของความทรงจำอยู่จำกัด เมื่อมีเหตุการณ์ใหม่ ๆ เข้ามายังความทรงจำ สามารถเลือกที่จะรับ หรือไม่รับเหตุการณ์ใหม่เข้ามายังความทรงจำตามความสำคัญของเหตุการณ์ และเมื่อสมองเลือกที่จะรับเหตุการณ์ใหม่ ๆ ที่มีความสำคัญเข้ามาเก็บไว้ในระบบความทรงจำแล้ว (memorize) ก็จะเป็นจะต้องมีเหตุการณ์บางอย่างในอดีตที่ถูกลืมไป (forget) และจุดเด่นของแบบจำลอง LSTM คือพังก์ชันพิเศษที่มีหน้าที่เสมือนประตู (Gate) ที่คอยควบคุมข้อมูลที่จะเข้ามาในแต่ละโหนด ซึ่งประกอบด้วย 3 โหนด ดังนี้

1. Forget gate layer เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามายัง Cell นั้นควรจะถูกเก็บไว้หรือควรจะทิ้งไป ซึ่งข้อมูลที่ถูกตัดสินว่าควรเก็บไว้นั้นจะถูกประเมินจากข้อมูล Input ที่เข้ามายังโหนดนั้นๆ รวมกับผลลัพธ์ที่จะได้จากการคำนวณของโหนดก่อนหน้า ผ่านพังก์ชัน ReLU ผลลัพธ์ที่ได้จาก Forget gate layer อุ่ร่วงระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้นหมายถึงให้ลบค่า Cell state เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า Cell state นี้ต่อไป
2. Input gate layer เป็น Gate ที่มีหน้าที่รับข้อมูล Input เข้ามายังแล้วจึงทำการบันทึกหรือเขียนข้อมูลลงไปในแต่ละโหนดโดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ ถ้าต้องการ Update cell state เมื่อทำการรับข้อมูล Input เข้ามายังแล้วพังก์ชันที่เป็นตัวควบคุมจะเรียกว่า Input gate เพื่อเลือกว่าจะให้ Update cell state หรือไม่ต้อง และในส่วนที่สองถ้า Input gate เลือกที่ทำการ Update cell state พังก์ชัน tanh ก็จะทำการสร้าง Candidate values ขึ้นมาใน State
3. Output gate layer เป็น Gate ที่มีหน้าที่เตรียมทำการส่งออกข้อมูล (Output data) โดยข้อมูลที่จะทำการ Output นั้นจะดูจาก Cell state ที่ผ่านกระบวนการคำนวณต่าง ๆ แล้ว โดยพังก์ชัน ReLU จะเป็นตัวเลือกว่าข้อมูลส่วนไหน Cell state ที่จะถูก Output จากนั้นก็นำค่า Cell state เข้าพังก์ชัน tanh (เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1) และนำค่าที่ได้จากพังก์ชัน tanh มาทำการคำนวณกับค่า Output ที่ได้จาก ReLU ของ Output จากนั้นจะได้ค่า Output ที่ต้องการ



รูปที่ 2.10 โครงสร้างสถาปัตยกรรมของ LSTM

2.1.8 การประเมินประสิทธิภาพของโมเดล Prophet และ LSTM

ในการประเมินประสิทธิภาพของโมเดล Prophet และ LSTM จะประเมินจาก Cost Function ซึ่งเป็นฟังก์ชันวัดความคลาดเคลื่อนในการเรียบสูตรของเครื่อง และแบบจำลองโดยใช้ตัวแปรทางเศรษฐศาสตร์ที่มีอยู่ เช่น จำนวนคน เศรษฐกิจ ฯลฯ ซึ่งถูกใช้ในขั้นตอนเรียบแบบจำลองทางเศรษฐศาสตร์ ซึ่งในโครงงานนี้จะเป็นการเรียบแบบจำลองโดยใช้ตัวแปรตามที่เป็นการคาดการณ์ปริมาณ เช่น จำนวนผู้เชื้อไวรัส โดยมีฟังก์ชันที่ใช้คือ Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) และ Mean Absolute Percentage Error (MAPE)

1. Mean Square Error (MSE) คือฟังก์ชันเรียบแบบจำลองที่ใช้ตัวแปรตามที่มีค่าจริง กับค่าคาดคะหนึ่งของตัวพยากรณ์ โดยแสดงเป็นผลเฉลี่ยของค่าคลาดเคลื่อนของทุก ๆ จุดเวลาที่แบบจำลองทำการพยากรณ์บนหนึ่งช่วงข้อมูล ตามสมการที่ 2.10

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.10)$$

2. Root Mean Square Error (RMSE) คือฟังก์ชันเรียบแบบจำลองที่ใช้ตัวแปรตามที่มีค่าจริง กับค่าคาดคะหนึ่งของตัวพยากรณ์ เช่นเดียวกัน แต่มีการเพิ่ม Square Root ในสมการเพื่อให้สะท้อนค่าเฉลี่ยของค่าคลาดเคลื่อนแต่ละจุดที่มีขนาดใหญ่ ซึ่งจะให้น้ำหนักของค่าคลาดเคลื่อนดังกล่าวมากกว่า จึงช่วยในการเรียบแบบจำลองให้ดีมากขึ้น ตามสมการที่ 2.11 ด้วย RMSE นี้จะประเมินร่วมกับ MSE ด้วย

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad (2.11)$$

3. Mean Absolute Error (MAE) คือฟังก์ชันเรียบแบบจำลองที่ใช้ตัวแปรตามที่มีค่าจริง กับค่าคาดคะหนึ่งของตัวพยากรณ์ เช่นเดียวกัน เพื่อวัดค่าส่วนบุรุษของค่าความคลาดเคลื่อนเฉลี่ยของผลพยากรณ์ ซึ่งใช้วัดค่าคลาดเคลื่อนที่เกิดขึ้นบนข้อมูลที่มีการเปลี่ยนแปลงเพียงเล็กน้อยได้ดี ถูกใช้ในขั้นตอนประเมินความแม่นยำของแบบจำลองตามสมการที่ 2.12

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2.12)$$

4. Mean Absolute Percentage Error (MAPE) เป็นวิธีวัดความแม่นยำโดยคำนวณเปอร์เซ็นต์ความผิดพลาดในการพยากรณ์ โดยไม่คำนึงถึงเครื่องหมาย ค่าที่ได้ต่างจะแสดงให้เห็นว่าโมเดลนั้นมีความแม่นยำสูง มีรูปแบบสมการที่ 2.13

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right| \quad (2.13)$$

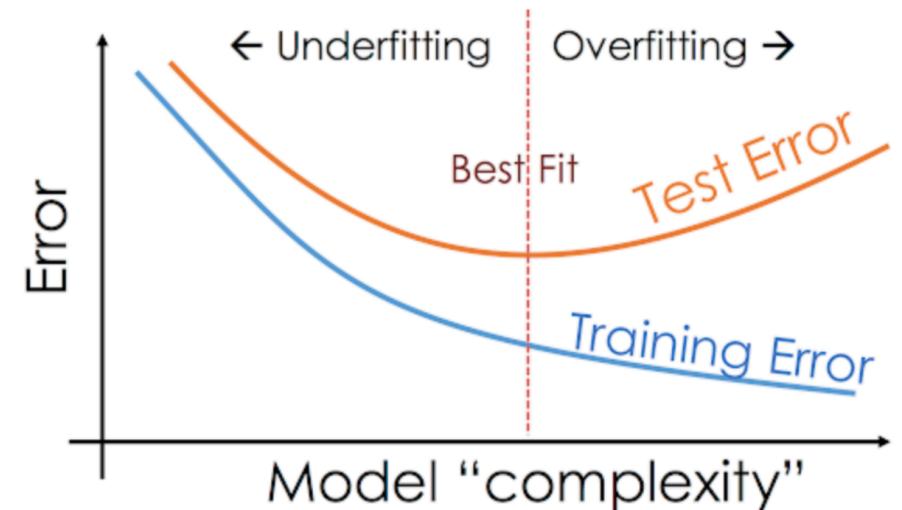
2.1.9 การทำ Data Normalization

เป็นเทคนิคส่วนหนึ่งในการจัดเตรียมข้อมูลก่อนการสร้างแบบจำลองที่เรียกว่า Machine Learning (Machine Learning) ซึ่งเป้าหมายของการทำ Data Normalization เป็นการเปลี่ยนข้อมูลตัวแปรต้นที่เป็นตัวเลขให้อยู่ในช่วงความถี่ที่เป็นมาตรฐานเดียวกันทั้งหมดเพื่อเพิ่มความแม่นยำในการพยากรณ์ เช่น ข้อมูลราคาซื้อขายของหุ้นอยู่ในช่วง 200 - 500 หรือ ข้อมูลสินทรัพย์ในงบการเงินอยู่ในช่วง 100,000,000 – 1,000,000,000 ซึ่งจะถูกปรับให้อยู่ในช่วงความถี่มาตรฐานเดียวกัน เช่น ช่วง 0 - 1 เป็นต้น ทั้งนี้ทำให้แบบจำลองเรียนรู้ข้อมูลได้ดีขึ้น การทำ Data Normalization นั้นทำได้หลายวิธี แต่ในงานวิจัยนี้ใช้สูตรการปรับความถี่ของข้อมูลโดยพิจารณาจากความทั่งของค่าต่อสุ่ล เป็นสัดส่วนต่อค่าของช่วงข้อมูลทั้งหมด ตามสมการที่ 2.14

$$x_{new} = \frac{x - min(x)}{max(x) - min(x)} \quad (2.14)$$

2.1.10 การแบ่งชุดข้อมูล (Train Test Split Data)

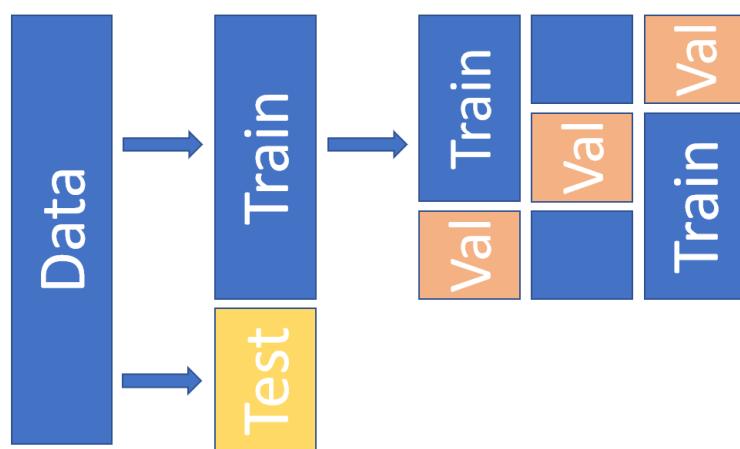
การแบ่งชุดข้อมูลเป็นเรื่องที่สำคัญมากในการทำ Machine Learning และ Deep Learning สมมติว่ามีชุดข้อมูลอยู่ 10,000 record หากนำข้อมูลทั้งหมดมาเทรนโมเดล อาจทำให้โมเดลเกิด overfitting ได้ ตามรูปที่ 2.11 และมีความสำคัญในการประเมินประสิทธิภาพของโมเดลมากที่สุด เพราะไม่มีข้อมูลมาทดสอบโมเดลการทำนาย แล้วจะรู้ได้อย่างไรว่าโมเดลทำงานได้แม่นยำแค่ไหนกับข้อมูลที่มันไม่เคยเห็นมาก่อน การที่ไม่สามารถทำงานได้แม่นยำกับข้อมูลที่ไม่เคยเห็นมาก่อน เรียกว่า Generalization เป็นคุณสมบัติสำคัญของ การพัฒนาระบบ Machine Learning และ Deep Learning เพราะถ้ามีระบบที่ทำงานได้แม่นยำเฉพาะกับข้อมูลที่มันเคยเห็นมาก่อน



รูปที่ 2.11 กราฟแสดงการเกิด Overfitting และ Underfitting

เหมือนกับเป็นนักเรียนที่จำข้อสอบ ทำถูกเฉพาะโจทย์ที่เหมือนเป็นเท่านั้น ไม่สามารถพลิกแพลงกับโจทย์ที่แตกต่างกันได้แม้แต่เล็กน้อยก็ตาม เมื่อเวลาไปใช้งานจริง เจอข้อมูลจริง ๆ ไม่เดลก็จะมีประสิทธิภาพความแม่นยำต่างกันไป เรียกว่า Overfit ตามรูปที่ 2.11 โดยปกติควรจะแบ่งข้อมูลออกเป็น 2 ส่วน คือ Training Set และ Test Set เช่น 90 % เป็น Training Set และ 10 % เป็น Test Set หรือ 80 % เป็น Training Set และ 20 % เป็น Test Set ก็ได้ ขึ้นอยู่กับความเหมาะสมของข้อมูลที่มี เมื่อแบ่งข้อมูลเป็น 2 ส่วนนี้ ในการเรียนรู้การทดสอบและการปรับ Hyperparameter เพื่อเลือกโมเดลที่ทำงานได้แม่นยำที่สุดกับ Test Set เพื่อที่จะได้โมเดลที่ทำงานได้ดีกับข้อมูลที่ไม่เคยเห็นมาก่อน แต่ก็มีอีกปัญหาคือ Bias เป็นการใช้ Test Set หลาย ๆ ครั้ง ทำให้โมเดลที่เลือกว่าทำงานได้ดีที่สุดนั้น Bias ไปทาง Test Set เมื่อเทียบกับบางส่วนของโมเดลนั้นจำ Test Set เข้าไปด้วย ทำเกิดปัญหา Generalization เหมือนเดิม ทำให้ไม่สามารถมั่นใจได้ว่า โมเดลของเราจะทำงานได้ดีกับข้อมูลที่มั่นใจไม่เคยเห็นมาก่อน ดังนั้นจึงควรแบ่งข้อมูลออกเป็น 3 ส่วน คือ Training Set, Validation Set และ Test Set เช่น 80% เป็น Training Set แบ่งอีก 10% เป็น Validation Set และ 10% เป็น Test Set ตามรูปที่ 2.12

1. Training Set ใช้สำหรับป้อนให้โมเดลใช้เทรน
2. Validation Set ใช้สำหรับทดสอบหากประสิทธิภาพโมเดล เพื่อนำ weight ไปปรับปรุงและจุนในโมเดลระหว่างการเทรนโมเดล
3. Test Set ใช้สำหรับทดสอบหากประสิทธิภาพโมเดล (Evaluation Model)



รูปที่ 2.12 รูปการแบ่งชุดข้อมูลในการทำ Machine Learning และ Deep Learning

2.1.11 ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล (Database System) คือ ระบบที่ร่วบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ซัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวกับข้อมูลกันเข้าไว้ด้วยกันอย่างเป็นระบบ และเปิดโอกาสให้ผู้ใช้งานสามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งค่าตามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้เมื่อจำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภัยในโครงสร้างของฐานข้อมูล สามารถแก้ไขฐานข้อมูล (File) ระเบียน (Record) และ เขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวทั้งหมด โปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ต้องการได้อย่างรวดเร็ว

2.1.11.1 Relational Database

Relational Database คือ ระบบฐานข้อมูลเชิงสัมพันธ์ เป็นการเก็บข้อมูลในรูปของตาราง (table) ในแต่ละตารางแบ่งออกเป็น列 ๆ และในแต่ละ列จะแบ่งเป็นคอลัมน์ (Column) ซึ่งในการเขียนโปรแกรมระหว่างข้อมูลในตารางต่าง ๆ จะเขียนโดยใช้การอ้างอิงจากข้อมูลในคอลัมน์ที่กำหนดไว้ที่ใช้model เชิงสัมพันธ์ (Relational Database Model) เนื่องด้วยแนวคิดของแบบจำลองแบบนี้มีลักษณะที่คนใช้กันทั่วโลกคือ มีการเก็บเป็นตาราง ทำให้ง่ายต่อการเข้าใจและการประยุกต์ใช้งาน

2.1.11.2 SQL

Structured Query Language (SQL) คือภาษาที่ใช้ในการจัดการฐานข้อมูล (Database) ซึ่งได้รับการยอมรับมากที่สุดในโลกภาษาหนึ่ง และได้รับการยอมรับในมาตรฐาน American National Standards Institute (ANSI) SQL สามารถใช้งานร่วมกับเว็บไซต์ ระบบฐานข้อมูล SQL Server ไปจนถึงการสร้างระบบบริเคราะห์ข้อมูลด้วยตนเอง

การทำงานของ SQL สามารถใช้ทำงานได้หลายรูปแบบ โดยจะมีการทำงานหลัก ๆ ด้วยกัน 4 ประเภท ดังนี้

1. Select query ใช้สำหรับเลือกข้อมูล
2. Update query ใช้สำหรับเปลี่ยนแปลงข้อมูล
3. Insert query ใช้สำหรับการเพิ่มข้อมูล
4. Delete query ใช้สำหรับการลบข้อมูล

2.2 เครื่องมือที่ใช้ในการพัฒนา

2.2.1 Python

Python คือหนึ่งในภาษาโปรแกรมระดับสูงที่ใช้กันอย่างแพร่หลาย ถูกออกแบบเพื่อให้มีโครงสร้างและไวยากรณ์ของภาษาที่ไม่ซับซ้อน เข้าใจง่าย ถูกออกแบบมาให้เรียนรู้ได้ง่าย มีความสามารถใช้ชนิดข้อมูลแบบเดนามิก จัดการหน่วยความจำอัตโนมัติ สนับสนุนระบบหัศน์ การเขียนโปรแกรม (Programming paradigms) ประกอบด้วย การเขียนโปรแกรมเชิงวัตถุ (OOP : Object Oriented Programming) การเขียนโปรแกรมเชิงคำสั่ง (Imperative Programming) การเขียนโปรแกรมเชิงฟังก์ชัน (Functional) และการเขียนโปรแกรมเชิงกระบวนการ มีไลบรารีมาตรฐานมากมาย และใช้อินเตอร์เฟซต่อร์เปลภาษาโปรแกรมให้ทำงานบนระบบปฏิบัติการได้หลากหลาย มีการใช้พัฒนาแอปพลิเคชัน เว็บไซต์ รวมถึงแอปบันมือถือ หรืออุปกรณ์เคลื่อนที่ด้วย

2.2.2 OpenCV

OpenCV หรือ OpenSource Computer Vision คือเครื่องมือคอมพิวเตอร์ ซึ่งเป็นไลบรารีในภาษา Python ที่ถูกสร้างไว้ด้วยเทคนิคและอัลกอริธึมที่จำเป็นทั้งหมดเพื่อการประมวลผลภาพ ซึ่งสามารถช่วยให้เราจัดการวิดีโอหรือรูปภาพ เพื่อให้เกิดการเปลี่ยนแปลงกับรูปภาพให้เป็นตามที่ต้องการ เช่นการทำ Image Transformation และ Image Segmentation

2.2.3 Scikit-Learn

Scikit-Learn เป็นไลบรารีการเรียนรู้ของเครื่องสำอาง Python มันมีอัลกอริทึมการจัดหมวดหมู่ การคัดค้อย และการจัดกลุ่มที่หลากหลาย รวมถึงสนับสนุนแก้ตัวเลขและวิทยาศาสตร์ของ Python NumPy และ SciPy

2.2.4 Prophet

Prophet เป็นไลบรารีที่พัฒนาโดย Facebook สำหรับการคาดการณ์ข้อมูลอนุกรมเวลาตามแบบ Additive model ซึ่งแนวโน้มที่ไม่เป็นเชิงเส้นเหมาะสมกับคุณภาพประจำปี รายสัปดาห์ และรายวัน บวกกับเอฟเฟกต์วันหยุด สามารถจัดการกับข้อมูลที่ขาดหายไป จัดการกับค่าที่ผิดปกติ และการเปลี่ยนแปลงในแนวโน้มได้ดี

2.2.5 Tensorflow

Tensorflow เป็นไลบรารีสำหรับสร้าง machine learning models แบบ open source จาก Google สามารถใช้งานได้กับภาษา Python แต่ก็สามารถใช้ภาษาอื่นๆ เช่น C, Java หรือ Go ได้เช่นกัน และยังมี community ขนาดใหญ่ ทำให้สามารถค้นหาข้อมูล หรือสอบถามเวลาเจอบัญญาได้ง่าย

2.2.6 Keras

Keras คือ High-level interface ของ TensorFlow เป็น Low-level framework เป็น Opensource neural network ที่เขียนด้วยภาษา Python ออกแบบเนื้อเพื่อสร้างกับ Deep Learning Neural Network ได้อย่างรวดเร็ว สามารถ Run ได้ทั้งบน TensorFlow, Theano และ CNTK Keras อยู่ภายใต้ MIT license ที่หมายความว่าเราสามารถใช้งานมันได้อย่างอิสระเพื่อธุรกิจที่หวังกำไรได้ โดยไม่ต้องจ่ายเงินให้กับองค์กรต่าง ๆ

2.2.7 Segmentation Models API

Segmentation model เป็น Python library โครงข่ายประสาทเทียมสำหรับงาน Image segmentation ที่เขียนด้วย Keras/Tensorflow หรือ PyTorch เป็นพื้นฐาน library นี้สามารถสร้างโมเดลที่เป็นงาน Image segmentation ได้ด้วย API ซึ่งใน segmentation model จะประกอบไปด้วยโมเดล ดังนี้ Unet, FPN, Linknet และ PSPNet ซึ่งโมเดลเหล่านี้สามารถใช้งานร่วมกับ Backbones pre-trained ที่เป็น State of art ของ CNN และสามารถเรียกใช้ weight ของ imangenet มาช่วยในการเรียนรู้ได้อีกด้วย

2.2.8 Albulmentations

Albulmentations เป็น Python library สำหรับการเพิ่มจำนวนรูปภาพที่รวมเร็วและยืดหยุ่น สามารถใช้คำสั่งเปลี่ยนแปลงภาพที่หลากหลายได้อ่ายมีประสิทธิภาพ เช่น การหมุนภาพ การบีบอัด การครอบภาพ เป็นต้น หมายสำหรับงานประมวล Image Classification, Segmentation และ Object Detection

2.2.9 Django

เป็น framework ที่ใช้ในการสร้าง Web Application ในฝั่งของ Back End ที่พัฒนาด้วยภาษา Python โดยในตัว framework จะมีส่วนประกอบทุกอย่างที่จำเป็นตั้งแต่การเชื่อมต่อฐานข้อมูล ไปจนถึงการ render ข้อมูลออกมาให้ฝั่ง Front End แสดงผลข้อมูลเหล่านั้นได้ ซึ่ง framework ในรูปแบบนี้ในภาษาอื่น ๆ

2.2.10 Django REST Framework

Django REST Framework คือ toolkit หรือไลบรารีของ Python ในการสร้าง RESTful APIs และ REST : REpresentational State Transfer คือ สถาปัตยกรรมอย่างหนึ่งของซอฟต์แวร์ที่อยู่บนพื้นฐานของ HTTP Protocol ที่เกี่ยวข้องกับการส่งข้อมูลระหว่าง Client และ Server

2.2.11 PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลโอเพ่นซอร์สระดับองค์กรที่ทันสมัยที่สุดในโลกที่พัฒนาโดย PostgreSQL Global Development Group เป็นระบบฐานข้อมูล SQL (Structured Query Language) เงิงวัตถุสัมพันธ์ที่มีประสิทธิภาพและขยายได้สูงซึ่งได้รับความนิยมในด้านความนำร่องด้วยความสามารถในการจัดการฐานข้อมูลที่มีประสิทธิภาพสูง สามารถปรับขนาดได้สูงทั้งในด้านปริมาณข้อมูลที่สามารถจัดเก็บและจัดการได้ และในจำนวนผู้ใช้งานพร้อมกันที่สามารถรองรับได้ ซึ่ง PostgreSQL พร้อมให้บริการและเผยแพร่ภายใต้สิทธิการใช้งาน PostgreSQL ซึ่งเป็นสิทธิ์การใช้งานโอเพนซอร์ส ซึ่งหมายความว่าคุณสามารถดาวน์โหลดซอฟต์แวร์ใช้งานแก้ไขและแจกจ่ายได้โดยไม่เสียค่าใช้จ่ายเพื่อวัตถุประสงค์ใด ๆ นอกจากนี้ยังเป็นขั้มแพลตฟอร์มทำงานบน Linux, Windows, macOS และระบบปฏิบัติการอื่น ๆ

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 Cloud Coverage Prediction to Improve Solar Power Management

งานวิจัยนี้มีจุดประสงค์เพื่อสร้างโมเดลทำนายปริมาณเมฆเพื่อใช้ในการปรับปรุงการจัดการพลังงานแสงอาทิตย์ โดยในงานวิจัยนี้จะทำการหาค่าปริมาณเมฆ (Cloud Coverage) จากรูปภาพโดยใช้เทคนิคของ RBR Threshold และการแบ่งพื้นที่ของพิกเซลด้วยการเรียนรู้โครงข่ายประสาทเทียม (UNet) และใช้ค่าปริมาณเมฆที่ทำนายปริมาณเมฆโดยใช้เทคนิคของการเรียนรู้ของเครื่อง (Machine Learning) ในช่วงเวลา 5 10 และ 30 นาที นาทีในอนาคต มีการใช้เทคนิคของ vanilla Neural Network ในการทำนายปริมาณเมฆจากภาพในอีก 5 นาทีข้างหน้า โดยทำนายจากการแพร่ของสภาพอากาศเป็นปัจจัยในการทำนายเพิ่มเติม (Predictor) และใช้ MAPE เป็นตัวชี้วัดหลักในการคำนวณค่าความผิดพลาดของโมเดล ผลจากการทดลองในงานวิจัยนี้พบว่าโมเดลของการเรียนรู้ของเครื่อง (Machine Learning) ที่ประกอบไปด้วยฟีเจอร์ของค่าปริมาณเมฆ (Cloud Cover) และการฉายรังสีในแนวอนุทั่วโลก (Global Horizontal Irradiance) มีผลในการทำนายปริมาณเมฆมากที่สุด มีค่า MAPE อยู่ที่ 7.7603% [5]

2.3.2 A Deep Learning Model to Forecast Solar Irradiance Using a Sky Camera

งานวิจัยนี้มีจุดประสงค์เพื่อการเรียนรู้แบบโครงข่ายประสาทเทียม (Deep Learning) มาใช้ในการพยากรณ์เมฆปกคลุมโดยใช้ภาพท้องฟ้า การศึกษานี้พยายามใช้อัลกอริธึมหน่วยความจำระยะสั้นระยะยาวในการเรียนรู้เชิงลึก (LSTM) ซึ่งมีข้อมูลเมฆปกคลุมที่รวบรวมโดยการประมวลผลภาพของภาพท้องฟ้า ซึ่งใช้เทคนิคในของ RBR Thersholt และใช้สำหรับการพัฒนาแบบจำลองการเรียนรู้เชิงลึก เพื่อพยากรณ์เมฆปกคลุมล่วงหน้า 10 นาที ข้อมูลการปกคลุมของเมฆที่คาดการณ์ไว้จะมีค่าตัวแปรของพังงานแสงอาทิตย์ แบบจำลองการแพร่งสี (GHI) เป็นอินพุตเพื่อทำนายการแพร่งสีในแนวอนุทั่วโลก โดยผลลัพธ์ที่คาดการณ์ไว้แบ่งออกเป็น 3 ประเภทตามสภาพท้องฟ้า ได้แก่ ท้องฟ้าแจ่มใส และเมฆครึ่ง เมื่อเปรียบเทียบกับการวัดการแพร่งสีแสงอาทิตย์ที่สถานีภาคพื้นดิน แบบจำลองที่เสนอคือ ประเมินแบบจำลองที่เสนอันมีประสิทธิภาพดีกว่าแบบจำลองการคงอยู่ภายใต้ความแปรปรวนของแสงอาทิตย์สูง การฉายรังสี เช่น วันที่เมฆครึ่งเป็นบางส่วน โดยมีค่าความแตกต่างของรูทเฉลี่ยเป็นกำลังสองเป็นเวลา 10 นาทีข้างหน้า การคาดการณ์คือ 25.10% และ 39.95% ตามลำดับ ในที่สุดการศึกษาครั้งนี้ก็แสดงให้เห็นว่าโมเดลนี้สามารถเรียนรู้ความสามารถพยากรณ์เมฆที่ปกคลุมจากภาพท้องฟ้าได้ ดังนั้นจึงเป็นประโยชน์ในการพยากรณ์สุริยะ และการฉายรังสีภายใต้ความแปรปรวนสูง [6]

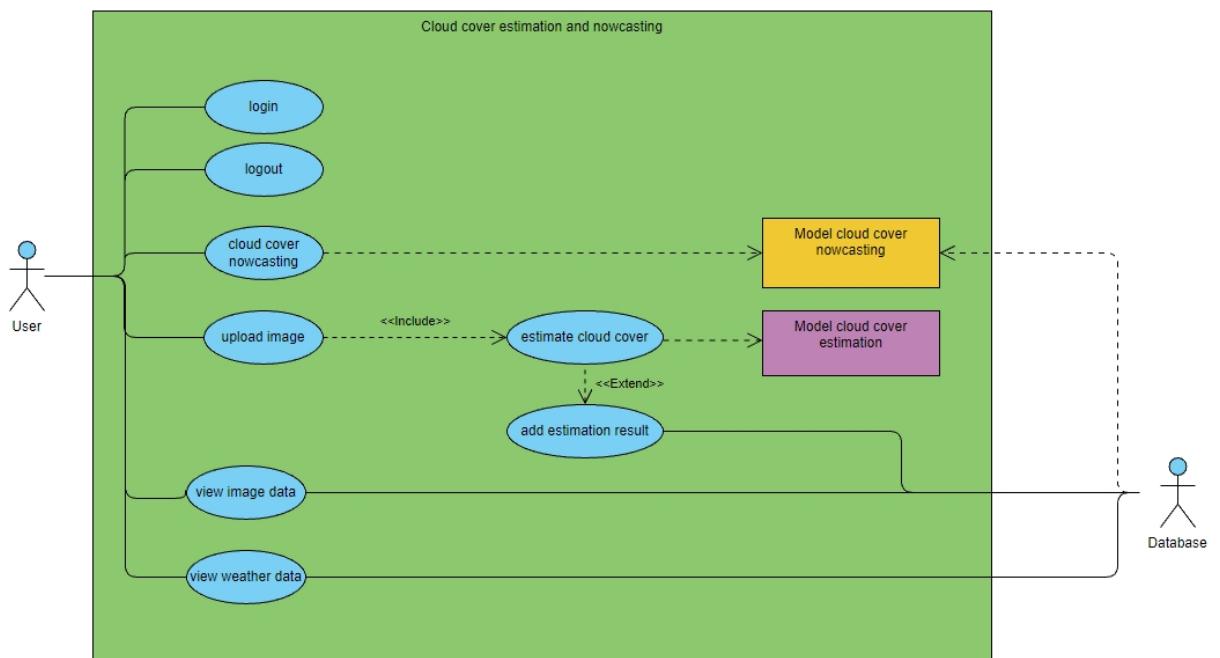
2.3.3 SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation

งานวิจัยมีจุดประสงค์เพื่อสร้างโมเดลที่สามารถตรวจจับพิกเซลของเมฆได้ โดยใช้การเรียนรู้แบบโครงข่ายประสาทเทียมสถาปัตยกรรมคอนโวลูชัน (CNN) มาใช้ในการแบ่งพิกเซลบนรูปภาพเมฆ โดยชุดข้อมูลที่ใช้คือ WSISEG Database เป็นชุดข้อมูลที่มีรูปถ่ายท้องฟ้า 400 รูป ประกอบไปด้วยรูปถ่ายท้องฟ้าต้นฉบับ และรูปถ่ายท้องฟ้าที่มีการทำ Ground truth ที่ทำการ label pixel เพื่อแยกว่า pixel ในหนึ่ง Clouds, Clear sky และ Underdefined Area ซึ่ง label ตัวเลข 255 (สีขาว), 100 (สีเทา), 0 (สีดำ) ตามลำดับ ซึ่งก่อนหน้าที่จะใช้การเรียนรู้แบบโครงข่ายประสาทเทียมสถาปัตยกรรมคอนโวลูชัน (CNN) นักวิจัยในงานนี้ได้มีการใช้วิธีของ Traditional segmentation methods ในการแบ่งพิกเซลบนรูปภาพเมฆ ได้แก่ R/B Threshold และ Otsu algorithm ซึ่งวิธีแบ่งดังเดิมเหล่านี้ยังไม่สามารถแยกขอบเขตระดับพิกเซลได้มากนัก นักวิจัยเลยคิดใช้วิธีของการเรียนรู้แบบโครงข่ายประสาทเทียมสถาปัตยกรรมคอนโวลูชัน (CNN) มาช่วยแก้ไขปัญหานี้ โดยโมเดลที่ใช้ในงานนี้คือ SegClouds Model ซึ่งภายในจะประกอบไปด้วย Encoder network, Decoder network ซึ่งมีการใช้ pre-trained ของ VGG16 และขั้นสุดท้ายมีการใช้ activation เป็น softmax เพื่อแยกพิกเซลแต่ละจุดว่าอยู่ในคลาสใด จากรезультатทดลองพบว่า Average accuracy ทั้ง 3 คลาสพบว่าวิธีของ Otsu method มี Average accuracy อยู่ที่ 56.86% R/B Threshold มี Average accuracy อยู่ที่ 81.17% และ SegClouds Model มี Average accuracy อยู่ที่ 96.24% จากผลการทดลองข้างต้นจะเห็นว่า SegClouds Model สามารถตรวจจับพิกเซลของ Clouds, Clear sky และ Underdefined Area ได้ดีที่สุด [7]

บทที่ 3 วิธีการดำเนินงาน

3.1 การออกแบบระบบ (System Design)

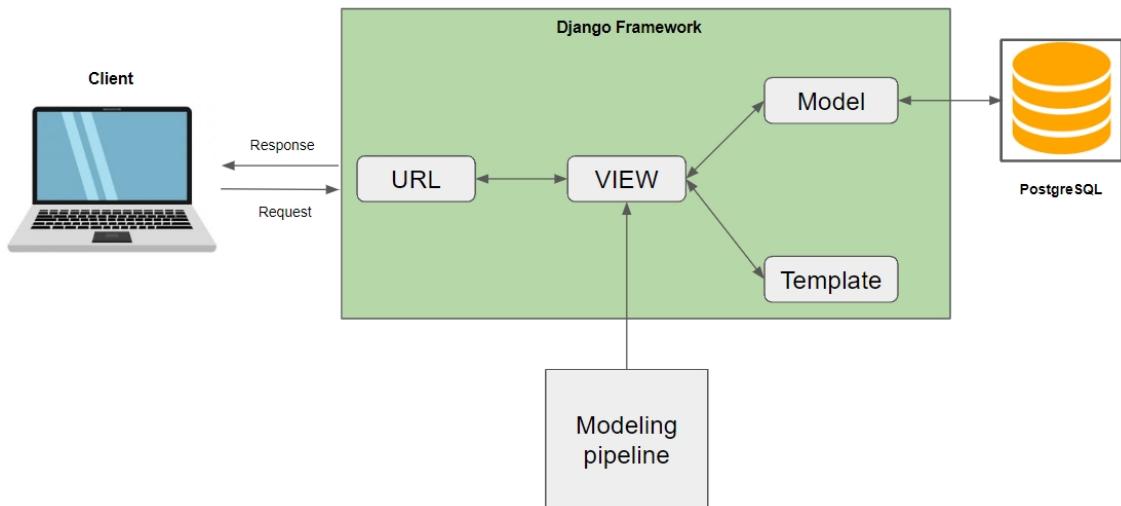
3.1.1 Use Case Diagram



รูปที่ 3.1 Use Case Diagram

จากรูปที่ 3.1 ผู้ใช้งานจะต้องมี username และ password เพื่อเข้าใช้งานระบบก่อนเข้าใช้งานภายในเว็บแอปพลิเคชัน โดย username และ password ที่ใช้เข้าใช้งานระบบผู้ใช้มีความสามารถสร้างเองได้ จะต้องให้ผู้ดูแลระบบสร้าง username และ password ให้เท่านั้น เมื่อผู้ใช้งานเข้าระบบมาได้แล้ว ก็จะสามารถใช้งานที่เจอร์การคำนวณปริมาณเมฆจากรูปภาพ และดูการพยากรณ์เมฆได้ โดยที่เจอร์เหล่านี้จะเชื่อมต่อการทำงานกับระบบ Model system ประกอบไปด้วย Model cloud cover nowcasting และ Model cloud cover estimation ซึ่งเป็นการทำงานของโมเดลที่ใช้ในการคำนวณปริมาณเมฆและพยากรณ์เมฆ โดยอธิบายเพิ่มเติมในส่วนของการออกแบบ Data Modeling

3.1.2 โครงสร้างระบบของเว็บแอปพลิเคชัน (Web Application Architecture)



รูปที่ 3.2 Web application architecture diagram

จากรูปที่ 3.2 เป็นภาพรวมโครงสร้างระบบของเว็บแอปพลิเคชัน มีการทำงานด้วยกันอยู่ 4 ส่วนคือ

1. Client จะเป็นส่วนเรียกใช้เว็บแอปพลิเคชันจากผู้ใช้งานโดยเรียกผ่าน URL ซึ่งเป็นการทำงานภายใน Django Framework
2. Django Framework คือส่วนที่ควบคุมการทำงานของเว็บแอปพลิเคชันเป็นหลัก ภายในจะประกอบไปด้วยพื้นที่ทำงานดังนี้
 - URL เป็นส่วนรับการเรียกใช้จากผู้ใช้งาน
 - View เป็นส่วนควบคุมการทำงานหลักของระบบหลังบ้าน (Backend) เพื่อเรียกใช้งาน Model เมื่อผู้ใช้งานต้องเพิ่มข้อมูลภายใน Database และ Modeling pipeline ซึ่งอธิบายอยู่ในส่วนของการออกแบบ Data Modeling เมื่อผู้ใช้งานมีการเรียกใช้การทำงานของโมเดลในการคำนวณปริมาณเมฆและพยากรณ์ปริมาณเมฆ แล้วก็ส่งผลการคำนวณหรือพยากรณ์ออกไปส่วนของ Template
 - Template คือส่วนที่แสดงผลหน้าเว็บและข้อมูลต่าง ๆ ที่ได้มาจากการ View โดยในเว็บแอปพลิเคชันนี้จะสร้างระบบหน้าบ้าน (Frontend) ด้วย HTML, CSS และ Javascript
3. PostgreSQL เป็นระบบฐานข้อมูลที่ใช้เก็บข้อมูลต่าง ๆ ที่ผู้ใช้งานเพิ่มเข้ามา และทำงานร่วมกับ Model ใน Django
4. Modeling pipeline เป็นส่วนการทำงานของโมเดลต่าง ๆ คือโมเดลตรวจสอบจับเมฆเพื่อนำไปคำนวณปริมาณเมฆจากรูปภาพ และโมเดลในการพยากรณ์ปริมาณเมฆ และในส่วนนี้มีการเชื่อมต่อการทำงานกับ View ในภาษา Django

3.1.3 Database Design

3.1.3.1 พจนานุกรมข้อมูล (Data dictionary)

อธิบายถึงชื่อคอลัมน์ ความหมาย และประเภทของข้อมูลที่เก็บอยู่ในตาราง โดยที่ตารางประกอบไปด้วย 3 ตาราง คือ

1. User Table เป็นตารางที่ใช้เก็บข้อมูลของผู้ใช้งานทั้งหมด
2. TimeCloudCover Table เป็นตารางที่ใช้เก็บข้อมูลของปริมาณเมฆกับตัวแปรสภาพอากาศ
3. ImageCloudCover Table เป็นตารางที่ใช้เก็บข้อมูลของการคำนวณปริมาณเมฆด้วยรูปภาพ

ตารางที่ 3.1 User Table

| ชื่อคอลัมน์ | ความหมาย | ประเภท |
|-------------|---|-------------------------------|
| user_id | id สำหรับบ่งบอกผู้ใช้งาน (นักวิจัยหรือผู้เชี่ยวชาญ) (Primary Key) | Big integer auto_increment |
| username | ชื่อสำหรับผู้ใช้งานที่ใช้สำหรับเข้าสู่ระบบ | varchar(255) Unique |
| password | รหัสผ่านสำหรับผู้ใช้งานที่ใช้เข้าสู่ระบบ | varchar(255) |
| firstname | ชื่อจริงของผู้ใช้งาน | varchar(255) |
| lastname | นามสกุลของผู้ใช้งาน | varchar(255) |

ตารางที่ 3.2 TimeCloudCover Table

| ชื่อคอลัมน์ | ความหมาย | ประเภท |
|-------------|---|----------------------------|
| tcc_id | id สำหรับบ่งบอกลำดับการเพิ่มของข้อมูลอนุกรรมาදา (Primary Key) | Big integer auto_increment |
| datetime | วัน/เดือน/ปี เวลา ที่เก็บข้อมูล | datetime |
| cloud_cover | ปริมาณเมฆปกคลุมท้องฟ้า | float |
| eeg | การผลิตไฟฟ้าหูนูนเรียบ | float |
| temp | อุณหภูมิ | float |
| rh | ค่าความชื้นสมพองในวันเวลาหนึ่ง ๆ | float |
| ws | ค่าความเร็วลมในวันเวลาหนึ่ง ๆ | float |

ตารางที่ 3.3 ImageCloudCover Table

| ชื่อคอลัมน์ | ความหมาย | ประเภท |
|----------------|--|----------------------------|
| icc_id | สำหรับบ่งบอกลำดับการเพิ่มของข้อมูลรูปภาพ (Primary Key) | Big integer auto_increment |
| datetime | วัน/เดือน/ปี เวลา ที่เก็บข้อมูล | datetime |
| cloud_cover | ปริมาณเมฆปกคลุมในวันเวลาหนึ่ง ๆ | float |
| original_image | รูปภาพต้นฉบับที่อัปโหลด | path of picture |
| segment_image | รูปภาพที่ผ่านการตรวจสอบจับเมฆ | path of picture |

3.1.3.2 Entity Relationship Diagram (ER-Diagram)

แสดงถึงความสัมพันธ์ระหว่าง User Table, TimeCloudCover Table และ ImageCloudCover Table

| User | | timeCloudCover | | imageCloudCover | |
|------|-----------|----------------|-------------|-----------------|----------------|
| PK | | PK | | PK | |
| | user_id | | tcc_id | | icc_id |
| | username | | datetime | | datetime |
| | password | | cloud_cover | | cloud_cover |
| | email | | eeg | | original_image |
| | firstname | | Temperature | | segment_image |
| | lastname | | rh | | |
| | | | ws | | |

รูปที่ 3.3 แสดง ER-Diagram ของทั้ง 3 ตาราง

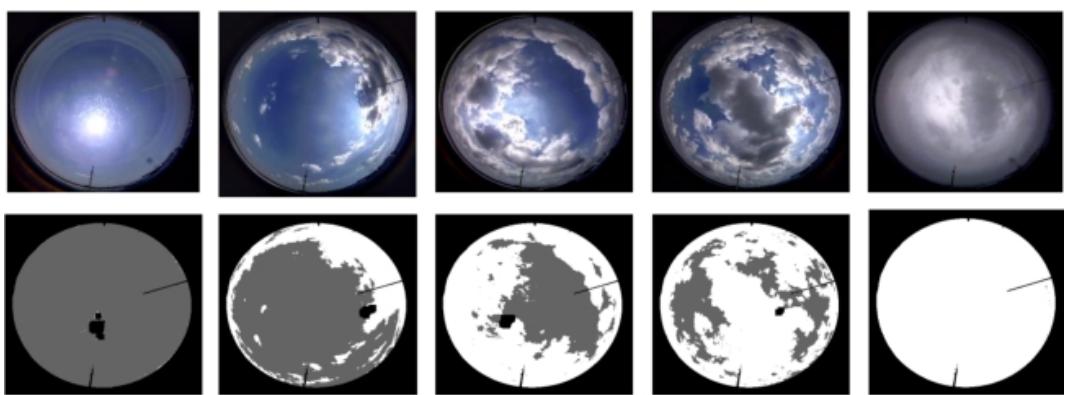
3.2 การเตรียมข้อมูล

3.2.1 การเตรียมข้อมูลสำหรับการตรวจสอบเมฆอัตโนมัติ

3.2.1.1 ชุดข้อมูลภายนอก

ชุดข้อมูลภายนอกที่ทำเพิ่มเติมนี้ มาจาก Dataset ที่มีชื่อว่า WSISEG-database จากงานวิจัยที่เกี่ยวข้องที่ [7] ซึ่งเปิดให้ใช้งานข้อมูลแบบ Open Source คณผู้จัดทำโครงการจะนำข้อมูลส่วนนี้ไปใช้เทรนโมเดลสำหรับการทำ Cloud detection เนื่องจากชุดข้อมูลมี Ground truth image เพื่อให้โมเดลสามารถเรียนรู้แบบ Supervised learning ได้ ชุดข้อมูลนี้มีภาพถ่ายท้องฟ้าทั้งหมด 400 รูป ขนาดรูปตันฉบับ 2000x1944 พิกเซลและทำการครอบภาพให้เหลือขนาด 480x450 พิกเซล ประกอบไปด้วย 2 โฟลเดอร์ ได้แก่

1. whole sky images เป็นโฟลเดอร์ที่เป็นรูปถ่ายท้องฟ้าตันฉบับ
2. annotation เป็นโฟลเดอร์ที่ทำการ label รูปถ่ายท้องฟ้าตันฉบับมาแล้ว ในส่วนนี้จะใช้เพื่อเป็น groung truth ในการเรียนรู้ของโมเดล



รูปที่ 3.4 ตัวอย่างชุดข้อมูลใน WSISEG-database

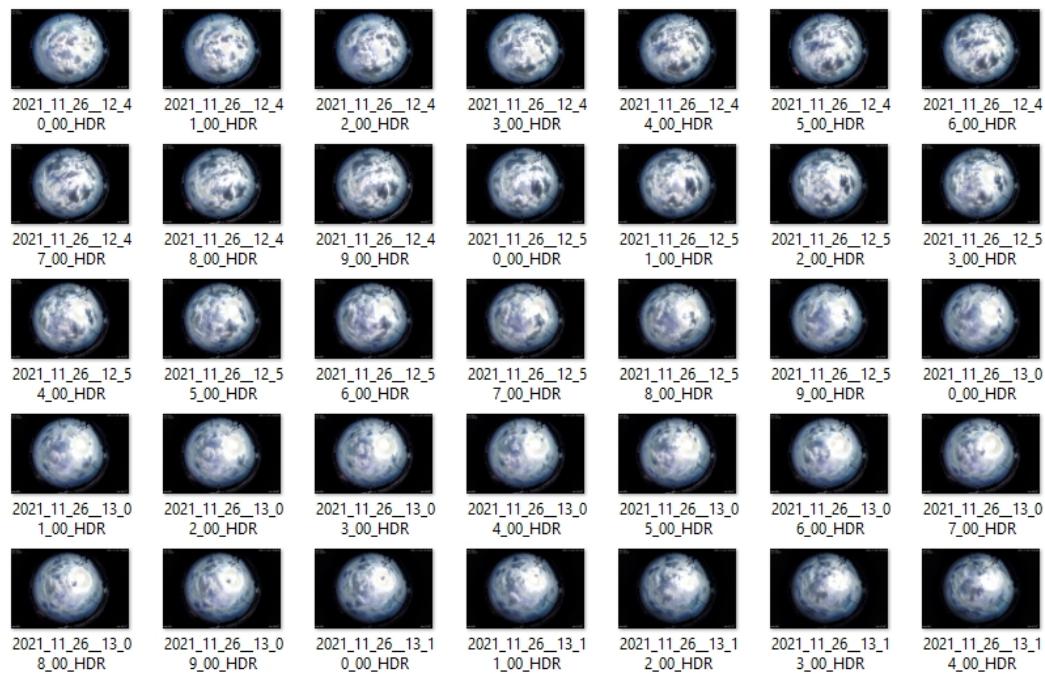
[ที่มา : <https://github.com/CV-Application/WSISEG-Database>]

โดยจำนวนชุดข้อมูลที่ใช้ในการเรียนรู้ของโมเดลจะแบ่งชุดข้อมูลเป็น Train set, Validation set และ Test set ดังนี้

1. Training set : 320 รูป
2. Validation set : 56 รูป
3. Test set : 24 รูป

3.2.1.2 ชุดข้อมูลภายใน

เป็นชุดข้อมูลรูปถ่ายท้องฟ้ามีข้อมูลตั้งแต่วันที่ 2021-11-16 ถึง 2021-12-16 รวมทั้งหมด 22,027 ภาพ ขนาดรูปตันฉบับ 2072x1411 พิกเซล ซึ่งถ่ายที่มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี วิทยาเขตบางขุนเทียน โดยข้อมูลชุดนี้ไม่มี groung truth มาให้ ซึ่งแต่ละรูปภาพ มีการหาระบีร์มาณ์เมฆจากรูปภาพไว้แล้วโดยผู้เชี่ยวชาญ ซึ่งเป็นไฟล์ .csv ตามรูปที่ 3.6 เมื่อเทรนโมเดลจากชุดข้อมูลภายนอกเสร็จเรียบร้อย แล้ว จะเอามาใช้งานกับชุดข้อมูลภายในนี้ และคำนวณหาระบีร์มาณ์เมฆปกคลุม แล้วเทียบ mean squared error ระหว่างระบีร์มาณ์เมฆที่ทำได้จากผู้เชี่ยวชาญ และระบีร์มาณ์เมฆที่คำนวณได้จากโมเดล U-Net



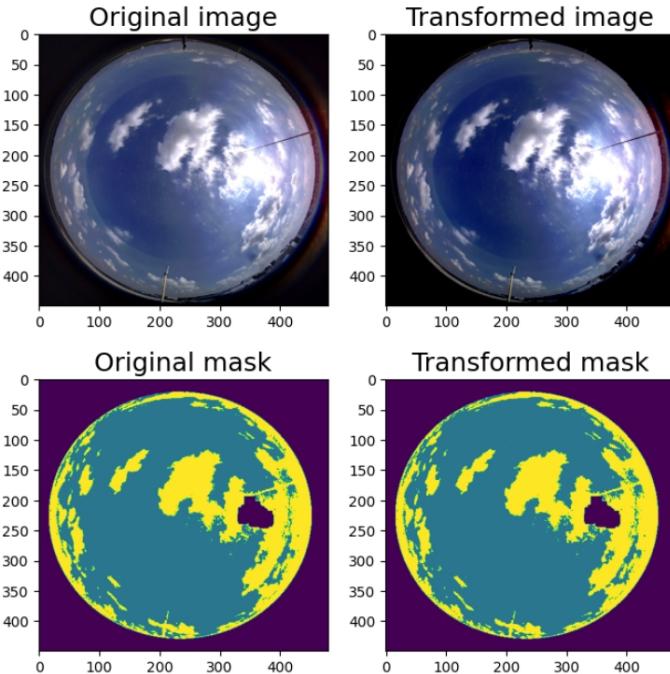
รูปที่ 3.5 ตัวอย่างภาพถ่ายท้องฟ้าจากชุดข้อมูลรูปภาพถ่ายท้องฟ้า

| Year | Month | Day | Hour | Minute | Second | CloudCover |
|------|-------|-----|------|--------|--------|------------|
| 2021 | 11 | 16 | 11 | 15 | 59 | 0.99 |
| 2021 | 11 | 16 | 11 | 16 | 20 | 0.989 |
| 2021 | 11 | 16 | 11 | 16 | 40 | 0.989 |
| 2021 | 11 | 16 | 11 | 17 | 1 | 0.988 |
| 2021 | 11 | 16 | 11 | 17 | 21 | 0.988 |
| 2021 | 11 | 16 | 11 | 17 | 42 | 0.987 |
| 2021 | 11 | 16 | 11 | 18 | 2 | 0.987 |
| 2021 | 11 | 16 | 11 | 18 | 22 | 0.987 |
| 2021 | 11 | 16 | 11 | 18 | 43 | 0.987 |
| 2021 | 11 | 16 | 11 | 19 | 3 | 0.987 |
| 2021 | 11 | 16 | 11 | 19 | 24 | 0.987 |
| 2021 | 11 | 16 | 11 | 19 | 45 | 0.987 |
| 2021 | 11 | 16 | 11 | 20 | 5 | 0.988 |
| 2021 | 11 | 16 | 11 | 20 | 26 | 0.99 |
| 2021 | 11 | 16 | 11 | 20 | 46 | 0.991 |
| 2021 | 11 | 16 | 11 | 21 | 6 | 0.991 |
| 2021 | 11 | 16 | 11 | 21 | 27 | 0.992 |
| 2021 | 11 | 16 | 11 | 21 | 48 | 0.993 |
| 2021 | 11 | 16 | 11 | 22 | 8 | 0.993 |
| 2021 | 11 | 16 | 11 | 22 | 29 | 0.993 |
| 2021 | 11 | 16 | 11 | 22 | 49 | 0.993 |
| 2021 | 11 | 16 | 11 | 23 | 9 | 0.993 |
| 2021 | 11 | 16 | 11 | 23 | 30 | 0.993 |
| 2021 | 11 | 16 | 11 | 23 | 51 | 0.993 |

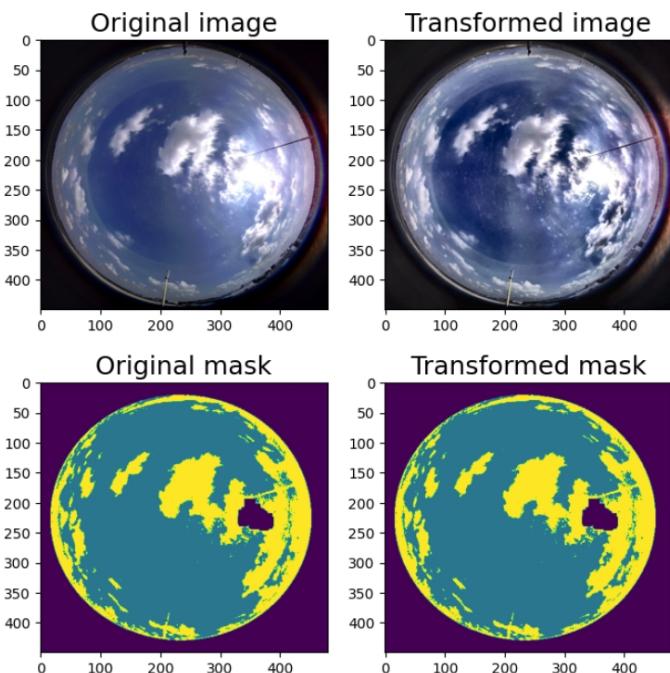
รูปที่ 3.6 ตัวอย่างปริมาณเมฆจากรูปภาพที่คำนวณโดยผู้เชี่ยวชาญ

3.2.1.3 การทำ Data Augmentation กับชุดข้อมูล WSISEG-database

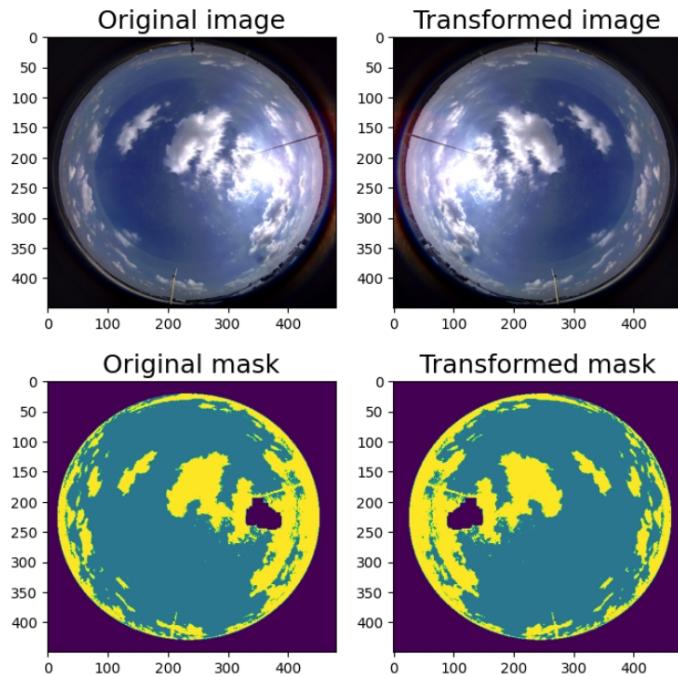
เนื่องจากชุดข้อมูล WSISEG-database มีจำนวนชุดข้อมูลเพียง 400 รูปเท่านั้น หากนำไปเทรนโน้มเดลสำหรับตรวจจับเมืองอัดโนมต้องทำให้ประสิทธิภาพไม่เสื่อมกันมาก จึงทำให้การทำ Data Augmentation เพื่อเพิ่มจำนวนข้อมูลเป็นวิธีที่ดีที่สุด ทำให้ข้อมูลจากเพิ่มจาก 400 รูปเป็นทั้งหมด 2000 รูป โดยจะทำการเปลี่ยนแปลงข้อมูลรูปภาพ ได้แก่ RandomBrightnessContrast ตามรูปที่ 3.7, CLAHE ตามรูปที่ 3.8, Flip ตามรูปที่ 3.9, RandomBrightnessContrast+Flip ตามรูปที่ 3.10 โดยจำนวนชุดข้อมูลที่ใช้ในการเรียนรู้ของโน้มเดลจะแบ่งเป็น



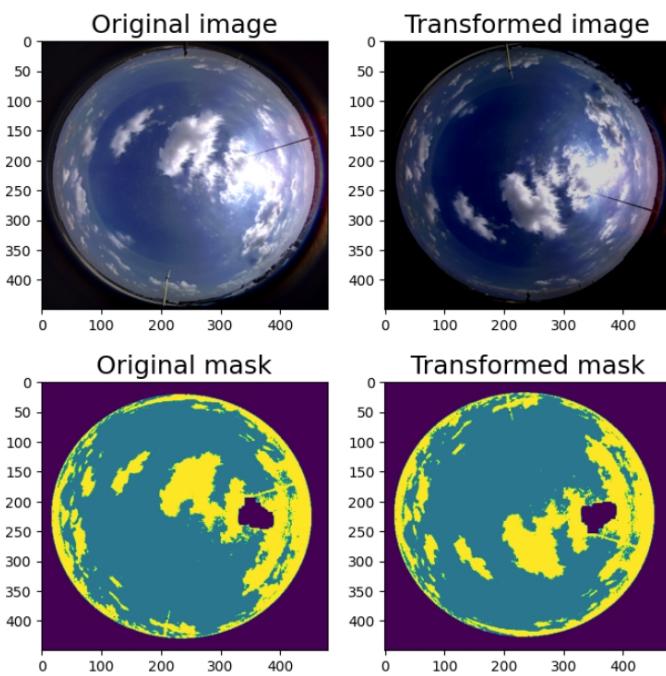
รูปที่ 3.7 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย RandomBrightnessContrast



รูปที่ 3.8 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย CLAHE



รูปที่ 3.9 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย Flip



รูปที่ 3.10 ตัวอย่างภาพการเปลี่ยนแปลงข้อมูลภาพด้วย RandomBrightnessContrast+ Flip

ข้อมูลเป็น Train set, Validation set และ Test set โดย Test set จะเป็นชุดข้อมูลที่ไม่ได้ทำ Data Augmentation ดังนี้

1. Train set: 1280
2. Validation set: 320
3. Test set: 400

3.2.2 การเตรียมข้อมูลสำหรับการพยากรณ์ปริมาณเมฆ

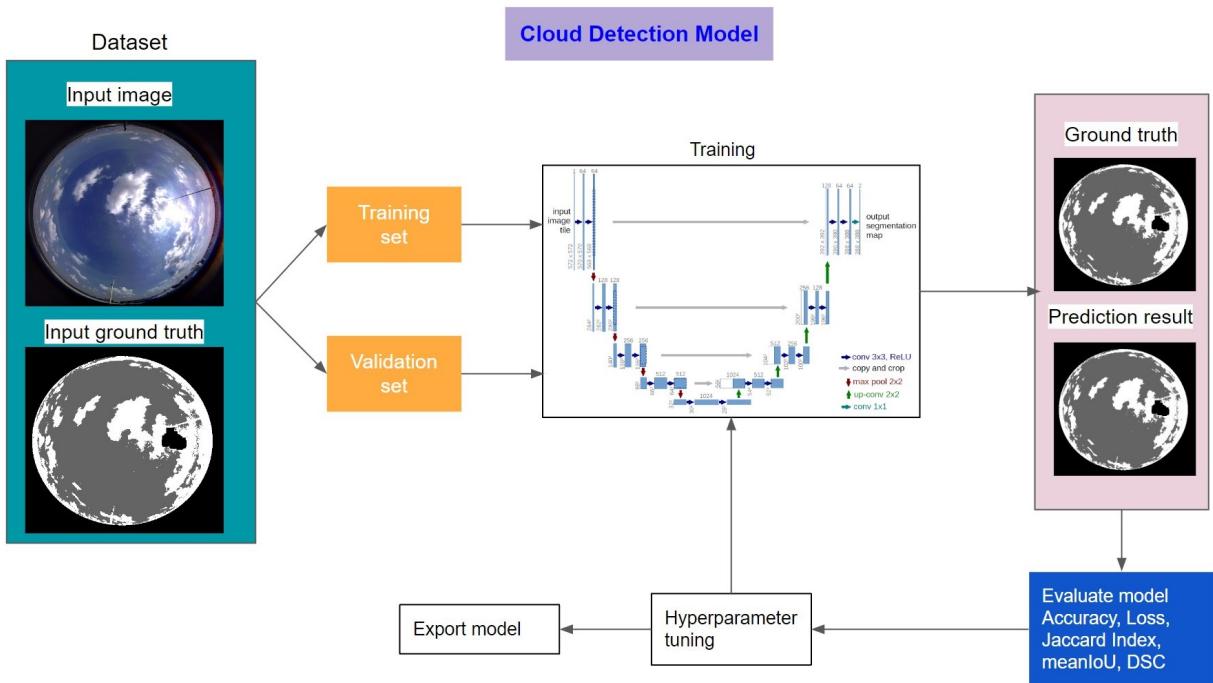
1. เตรียมชุดข้อมูลตัวแปรเดียวเป็นนาที ชั่วโมง วัน และหลายตัวแปรเป็นนาที
2. ในการทำ Prophet Model ก่อนการแบ่งชุดข้อมูล ควรเปลี่ยนชื่อคอลัมน์เวลาเป็น ds และตัวแปรที่ต้องการทำนายให้เป็น y
3. แบ่งชุดข้อมูลเป็น Train set และ Test set โดยใช้วิธีการแบ่งชุดข้อมูลเป็น 2 วิธี คือ
 - (a) แบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ซึ่งมีดังนี้
 - การแบ่งชุดข้อมูลตัวแปรเดียวเป็นนาทีโดยใช้ตัวแปร Cloud Cover มีจำนวนดังนี้
 - Train set มีจำนวน 12048 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:15:00 ถึงวันที่ 10-12-2021 เวลา 15:23:00
 - Test set มีจำนวน 3013 ข้อมูล โดยเริ่มตั้งแต่วันที่ 10-12-2021 เวลา 15:24:00 ถึงวันที่ 16-12-2021 เวลา 14:02:00
 - การแบ่งชุดข้อมูลตัวแปรเดียวเป็นชั่วโมงโดยใช้ตัวแปร Cloud Cover มีจำนวนดังนี้
 - Train set มีจำนวน 204 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:00:00 ถึงวันที่ 10-12-2021 เวลา 15:00:00
 - Test set มีจำนวน 52 ข้อมูล โดยเริ่มตั้งแต่วันที่ 10-12-2021 เวลา 16:00:00 ถึงวันที่ 16-12-2021 เวลา 14:00:00
 - การแบ่งชุดข้อมูลตัวแปรเดียวเป็นรายวันโดยใช้ตัวแปร Cloud Cover มีจำนวนดังนี้
 - Train set มีจำนวน 24 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 ถึงวันที่ 10-12-2021
 - Test set มีจำนวน 6 ข้อมูล โดยเริ่มตั้งแต่วันที่ 11-12-2021 ถึงวันที่ 16-12-2021
 - การแบ่งชุดข้อมูลหลายตัวแปรเป็นนาทีโดยใช้ตัวแปร Cloud Cover, Temperature, Relative humidity และ Wind speed จากนั้นทำให้ค่าอยู่ในช่วงเดียวกันโดยทำการ normalization ข้อมูลตัววิธี MinMaxScaler มีจำนวนดังนี้
 - Train set มีจำนวน 2384 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:15:00 ถึงวันที่ 10-12-2021 เวลา 15:05:00
 - Test set มีจำนวน 596 ข้อมูล โดยเริ่มตั้งแต่วันที่ 10-12-2021 เวลา 15:15:00 ถึงวันที่ 16-12-2021 เวลา 14:00:00
- (b) แบ่งชุดข้อมูล โดยกำหนดให้ชุดข้อมูลสำหรับ Test set เป็นวันสุดท้ายของชุดข้อมูลทั้งหมด ซึ่งมีดังนี้
 - การแบ่งชุดข้อมูลตัวแปรเดียวเป็นนาทีโดยใช้ตัวแปร Cloud Cover มีจำนวนดังนี้
 - Train set มีจำนวน 14768 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:15:00 ถึงวันที่ 15-12-2021 เวลา 16:59:00
 - Test set มีจำนวน 293 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-12-2021 เวลา 09:10:00 ถึงวันที่ 16-12-2021 เวลา 14:02:00
 - การแบ่งชุดข้อมูลตัวแปรเดียวเป็นชั่วโมงโดยใช้ตัวแปร Cloud Cover มีจำนวนดังนี้
 - Train set มีจำนวน 250 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:00:00 ถึงวันที่ 15-12-2021 เวลา 16:00:00
 - Test set มีจำนวน 6 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-12-2021 เวลา 9:00:00 ถึงวันที่ 16-12-2021 เวลา 14:00:00
 - การแบ่งชุดข้อมูลหลายตัวแปรเป็นนาทีโดยใช้ตัวแปร Cloud Cover, Temperature, Relative humidity, Wind speed จากนั้นทำให้ค่าอยู่ในช่วงเดียวกันโดยทำการ normalization ข้อมูลตัววิธี MinMaxScaler มีจำนวนดังนี้
 - Train set มีจำนวน 2921 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-11-2021 เวลา 11:15:00 ถึงวันที่ 16-12-2021 เวลา 15:55:00
 - Test set มีจำนวน 59 ข้อมูล โดยเริ่มตั้งแต่วันที่ 16-12-2021 เวลา 09:10:00 ถึงวันที่ 16-12-2021 เวลา 14:00:00

3.2.3 การออกแบบ Data Modeling

3.2.3.1 การออกแบบโมเดลสำหรับตรวจจับเมฆอัตโนมัติ

ขั้นตอนการสร้างวิธีการตรวจจับเมฆอัตโนมัติด้วยโมเดล U-Net เป็นสถาปัตยกรรมลักษณะต้องการผู้สอน (Supervised Learning) โดยในโครงงานนี้จะทำการสอนรูปที่เป็นต้นฉบับและรูปที่ทำการแบ่งส่วนของพื้นที่เมฆและส่วนของพื้นที่ท้องฟ้า โดยเราจะใช้คัพเพ็ทเทคนิคดังนี้ คือ รูปต้นฉบับ เรียกว่า Original และรูปที่ทำการแบ่งส่วนของพื้นที่เพื่อใช้สอนโมเดลให้รู้จักพื้นที่ของเมฆและพื้นที่ท้องฟ้า จะเรียกว่า Ground truth จากนั้นจะประเมินผลการตรวจจับเมฆของโมเดลจากการวัดค่าความถูกต้องด้วยวิธีการวัดรูปแบบต่าง ๆ ได้แก่ Accuracy, Categorical Loss Entropy, Jaccard Index(IoU) และ meanIoU หลักจากนั้นจะบันทึกน้ำหนักของโมเดลที่เหมาะสมไปใช้งานบนเว็บแอปพลิเคชัน จากรูปที่ 3.11 มีขั้นตอนการเทรนโมเดลในการตรวจจับเมฆดังนี้

1. นำชุดข้อมูล WSISEG-Database ที่มีขนาดรูปต้นฉบับ 2000x1944 พิกเซล ครอบคลุมให้เหลือขนาด 480x450 พิกเซล
2. แบ่งข้อมูลของ train set, validation set และ test set ตามชุดข้อมูลที่ทำแล้วได้ทำ Data Augmentation
3. ทำการปรับขนาดรูปภาพของ 480x450 พิกเซล เป็น 224x224 พิกเซลเพื่อใช้เป็น input shape ในการเทรนโมเดล U-Net
4. ทำการ normalize รูปภาพด้วย $\text{image} = \text{image}/255.0$ หรือเรียกวิธีนี้ว่า Min-Max Normalization
5. ทำการเทรนโมเดล U-Net ทั้งหมด 4 แบบที่ได้ออกแบบไว้ ดังนี้
 - (a) U-Net without data augmentation : โมเดลนี้จะเทรนด้วยชุดข้อมูลที่ไม่ได้ทำการเพิ่มจำนวนชุดข้อมูล กล่าวคือโมเดลนี้จะถูกเทรนด้วยชุดข้อมูลจำนวน 400 รูป
 - (b) U-Net with data augmentation : โมเดลนี้จะเทรนด้วยชุดข้อมูลที่ทำการเพิ่มจำนวนชุดข้อมูล กล่าวคือโมเดลนี้จะถูกトレนด้วยชุดข้อมูลจำนวน 2,000 รูป
 - (c) U-Net+ResNet34 without data augmentation : โมเดลนี้จะถูกปรับแต่งส่วนของ encoder ด้วยโครงสร้าง ResNet34 ซึ่งโมเดลนี้จะเทรนด้วยชุดข้อมูลที่ไม่ได้ทำการเพิ่มจำนวนชุดข้อมูล กล่าวคือโมเดลนี้จะถูกเทรนด้วยชุดข้อมูลจำนวน 400 รูป
 - (d) U-Net+ResNet34 with data augmentation : โมเดลนี้จะถูกปรับแต่งส่วนของ encoder ด้วยโครงสร้าง ResNet34 ซึ่งโมเดลนี้จะเทรนด้วยชุดข้อมูลที่ทำการเพิ่มจำนวนชุดข้อมูล กล่าวคือโมเดลนี้จะถูกเทรนด้วยชุดข้อมูลจำนวน 2,000 รูป
6. เทรนโมเดลทั้ง 4 แบบด้วย epoch=60, optimizer=AdamW
7. ประเมินประสิทธิภาพของโมเดลด้วย metrics : Accuracy, Categorical Loss Entropy, Jaccard Index(IoU) และ meanIoU
8. ทำการปรับ batch size และ learning rate เพื่อเพิ่มประสิทธิภาพของโมเดล
9. เลือกโมเดลที่ดีที่สุดเพื่อนำไปใช้งานบนเว็บแอปพลิเคชัน



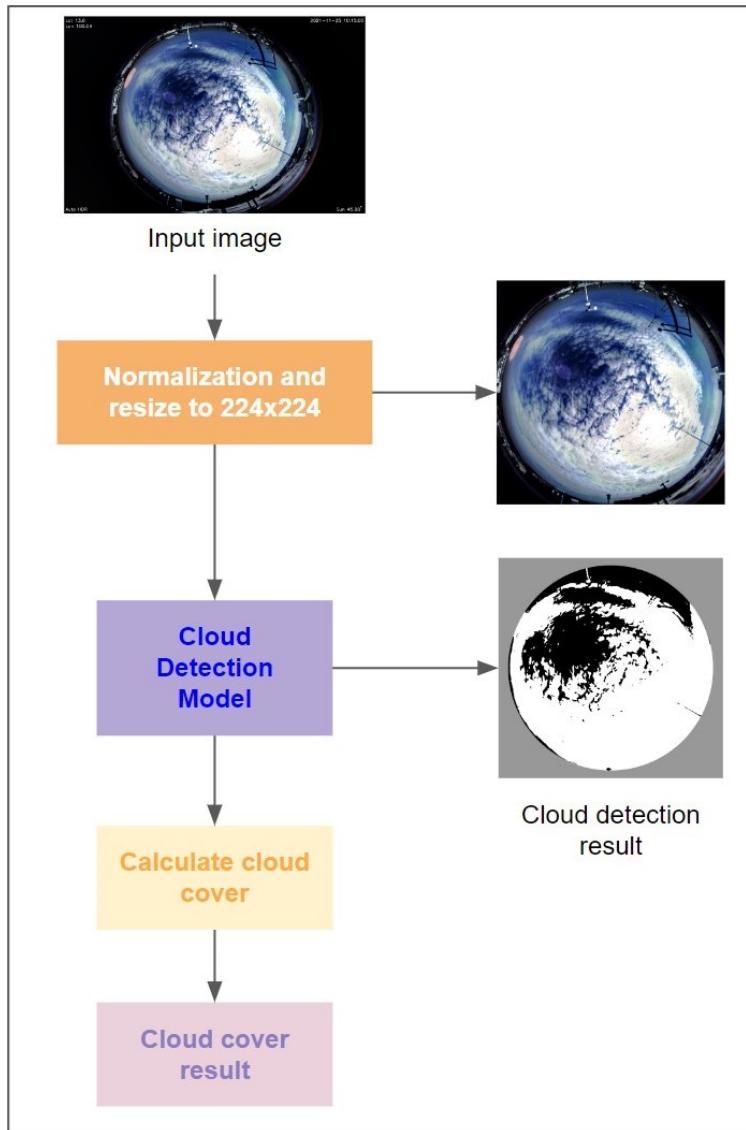
รูปที่ 3.11 ขั้นตอนการทำโมเดลสำหรับตรวจจับเมฆอัตโนมัติ

3.2.3.2 การออกแบบขั้นตอนการคำนวณปริมาณเมฆจากการปีกษา

ในส่วนนี้เป็นขั้นตอนการคำนวณปริมาณเมฆจากรูปภาพโดยรับภาพต้นฉบับจากผู้ใช้งานบนเว็บแอปพลิเคชัน หรือใช้ในการประเมินค่าความคลาดเคลื่อนของการคำนวณเริงากอนๆ เช่นที่คำนวณได้จากการผู้ใช้เวลาอ่านกับโน้ตเดลที่สร้างขึ้น

จารุ | ๓๑๒ มีข้อเสนอแนะในการคำนวณ ริมานูเมต์จากวงล้อภาพ ดังนี้

- รับภาพถ่ายท้องฟ้าจากผู้ใช้งาน
 - ครอบภาพที่รับเข้ามาจากผู้ใช้งานที่มีขนาดรูปด้านบน 2072x1411 พิกเซล เป็นขนาด 1170x1170 พิกเซล
 - ปรับขนาดภาพจาก 1170x1170 พิกเซล เป็น 224x224 พิกเซล ที่เป็น input shape ในขั้นตอนการเทรนโมเดลตรวจจับเมฆ
 - ทำการ normalize รูปภาพด้วย $\text{image} = \text{image}/255.0$
 - ตรวจจับเมฆจากรูปภาพด้วยโมเดลที่ได้ที่สุดจากขั้นตอนการออกแบบโมเดลสำหรับตรวจจับเมฆอัตโนมัติ ผลลัพธ์การตรวจจับเมฆ จะได้รูปที่ทำ Image Segmentation พื้นที่ของเมฆแทนด้วยพิกเซลสีขาว พื้นที่ของท้องฟ้าแทนด้วยพิกเซลสีเทา และพื้นที่ของแสงอาทิตย์หรือพื้นที่ที่ไม่เกี่ยวข้องแทนด้วยพิกเซลสีดำ โดยลักษณะของรูปที่ได้จากโมเดลจะเป็น Gray-Scale
 - นำรูปที่ได้จากการตรวจจับเมฆมาคำนวณปริมาณเมฆด้วยอัลกอริทึมที่ 3.14
 - ผลลัพธ์จากการคำนวณจะได้ปริมาณเมฆปกคลุมท้องฟ้าจากรูปภาพที่รับเข้ามาในขั้นตอนที่ 1



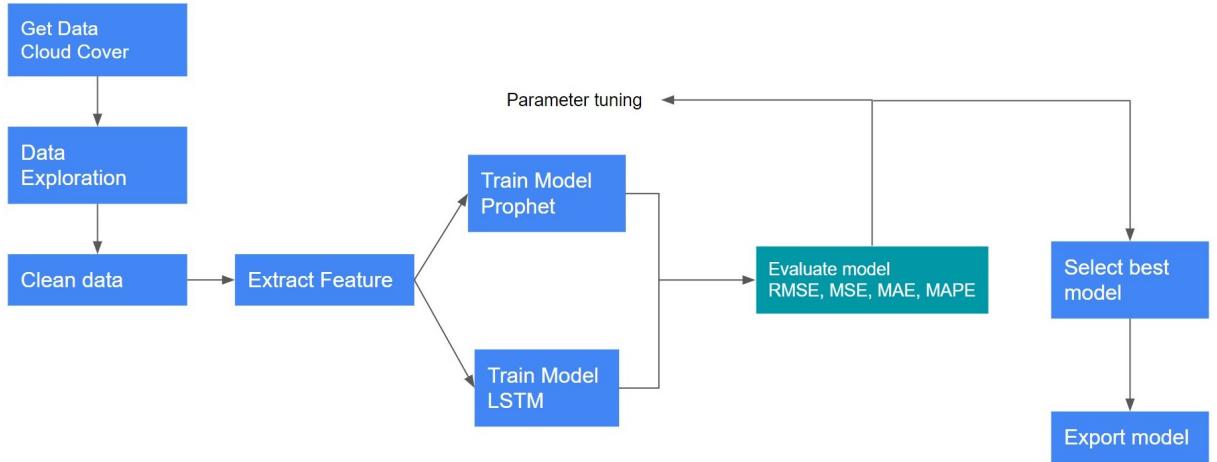
รูปที่ 3.12 ขั้นตอนการนำโมเดลตรวจจับเมฆอัตโนมัติมาใช้ในการทำปริมาณเมฆ

3.2.3.3 การออกแบบโมเดลสำหรับการพยากรณ์ปริมาณเมฆ

ขั้นตอนการสร้างแบบจำลองหรือวิธีการพยากรณ์ที่เหมาะสมกับข้อมูลโครงงานนี้จะเปรียบเทียบการใช้การพยากรณ์อนุกรมเวลาด้วย Prophet และ LSTM โดยการประเมินผลประสิทธิภาพการพยากรณ์ของโมเดลจากการวัดค่าความคลาดเคลื่อนการพยากรณ์ ได้แก่ RMSE, MSE, MAE, MAPE เพื่อเลือกโมเดลที่เหมาะสมที่สุด หลักจากนั้นจะบันทึกโมเดลที่เหมาะสมเพื่อนำโมเดลไปใช้งานบนเว็บแอปพลิเคชัน สามารถอธิบายขั้นตอนการออกแบบโมเดลสำหรับการพยากรณ์ปริมาณเมฆได้ตามรูปที่ 3.13 ดังนี้

1. รวบรวมข้อมูลที่ใช้ในการพยากรณ์ปริมาณเมฆ
2. ทำ Data exploration เพื่อดูลักษณะของข้อมูล เช่น ค่าเฉลี่ย ค่าสัมประสิทธิ์สหสัมพันธ์ในแต่ละตัวแปร
3. ทำความสะอาดข้อมูล เช่น การกำจัดค่าว่างในแต่ละแคว (NA) และการปรับช่วงของข้อมูลให้อยู่สเกลเดียวกัน (Feature scaling)
4. เลือก Feature ที่ใช้ในการพยากรณ์ปริมาณเมฆ
5. นำข้อมูลที่ได้เข้าไปเทรนโมเดล Prophet และโมเดล LSTM
6. ประเมินประสิทธิภาพของโมเดลด้วยวิธีการวัด RMSE, MSE, MAE, MAPE

7. เลือกโมเดลที่ดีที่สุดเพื่อนำไปใช้บนเว็บแอปพลิเคชัน



รูปที่ 3.13 ขั้นตอนการทำโมเดลสำหรับการพยากรณ์ปริมาณเมฆ

จากรูปที่ 3.13 ส่วนของโมเดลที่จะทำการทดลองและเปรียบเทียบประสิทธิภาพในการทำนายปริมาณเมฆในช่วงเวลา 5-30 นาทีในอนาคต โดยประเภทของ Feature ที่จะทดลองแบ่งเป็น 2 ชุดได้แก่ univariate feature ประกอบไปด้วยค่า cloud cover และ multivariate features ประกอบไปด้วย Cloud cover, Temperature, Relative humidity, Wind speed ซึ่งลักษณะของโครงสร้างและการปรับจุน Optimizer, Learning rate, Batch size และ Epochs ที่แตกต่างกันตามความเหมาะสมดังนี้

1. Prophet : โมเดลนี้จะทำนายเป็นช่วงเวลาของนาที ชั่วโมง และรายวัน ซึ่งชุดข้อมูลที่ใช้เทรนจะมีลักษณะการแบ่งข้อมูลที่แตกต่างกัน คือ เトレนด้วยชุดข้อมูลที่ทำการแบ่ง Train set 80%, Test set 20% และเทรนด้วยชุดข้อมูลที่ทำการแบ่งให้วันสุดท้ายเป็น Test set
2. LSTM : โมเดลนี้จะทำนายเป็นช่วงเวลาของนาที ชั่วโมง และรายวัน ซึ่งชุดข้อมูลที่ใช้トレนจะมีลักษณะการแบ่งข้อมูลที่แตกต่างกัน คือ เトレนด้วยชุดข้อมูลที่ทำการแบ่ง Train set 80%, Test set 20% และトレนด้วยชุดข้อมูลที่ทำการแบ่งให้วันสุดท้ายเป็น Test set

3.3 การออกแบบการประเมินประสิทธิภาพของโมเดล

3.3.1 การออกแบบการประเมินและเปรียบเทียบประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ

ในการประเมินผลประสิทธิภาพของโมเดลจะใช้ Mean Absolute Percentage Error (MAPE) เป็น metrics หลักในการวัดความถูกต้องของโมเดลพยากรณ์ปริมาณเมฆ โดยในตารางที่ 3.6 จะเป็นการเปรียบเทียบประสิทธิภาพระหว่างโมเดล Prophet และโมเดล LSTM ซึ่งแต่ละโมเดลจะมีฟีเจอร์ที่ใช้ในการพยากรณ์ปริมาณเมฆตามตารางที่ ดังนี้

ตารางที่ 3.4 ตารางอธิบายชื่อย่อและชื่อเต็มของฟีเจอร์ที่ใช้ในการพยากรณ์ปริมาณเมฆ

| คำย่อ | ชื่อเต็ม | ความหมาย |
|-------|-------------------|--------------------------|
| CC | Cloud Cover | บริเวณเมฆปกคลุมบนท้องฟ้า |
| Temp | Temperature | อุณหภูมิ |
| RH | Relative humidity | ค่าความชื้นสัมพัทธ์ |
| WS | Wind speed | ความเร็วลม |

ตารางที่ 3.5 ตารางประเมินประสิทธิภาพของโมเดล LSTM ในแต่ละ lags

| Lags | Optimizer | Batch size | Learning rate | First layer | Dropout | Second layer | Output layer | Epochs | MAPE |
|------|-----------|------------|---------------|-------------|---------|--------------|--------------|--------|------|
| 5 | | | | | | | | | |
| 10 | | | | | | | | | |
| 15 | | | | | | | | | |
| 20 | | | | | | | | | |
| 25 | | | | | | | | | |
| 30 | | | | | | | | | |

ตารางที่ 3.6 ตารางประเมินประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ

| Model | รูปแบบของข้อมูล | การแบ่งชุดข้อมูล | Feature | Evaluation metric | | | |
|---------|-----------------|------------------------------|---------------|-------------------|------|-----|------|
| | | | | MAE | MAPE | MSE | RMSE |
| Prophet | นาที | train set 80 % test set 20 % | CC | | | | |
| | | | CC Temp RH WS | | | | |
| | ชั่วโมง | วันสุดท้ายเป็น test set | CC | | | | |
| | | | CC Temp RH WS | | | | |
| | รายวัน | train set 80 % test set 20% | CC | | | | |
| | | วันสุดท้ายเป็น test set | CC | | | | |
| LSTM | นาที | train set 80 % test set 20 % | CC | | | | |
| | | | CC Temp RH WS | | | | |
| | ชั่วโมง | train set 80 % test set 20% | CC | | | | |
| | | | CC Temp RH WS | | | | |
| | รายวัน | train set 80 % test set 20% | CC | | | | |
| | | วันสุดท้ายเป็น test set | CC | | | | |

3.3.2 การออกแบบการประเมินประสิทธิภาพของโมเดลตรวจจับเมฆด้วย U-Net

ในการประเมินประสิทธิภาพโมเดล U-Net ที่ใช้ในการตรวจจับเมฆอัตโนมัติ จากการเรียนรู้ของภาพต้นฉบับ (Original) และภาพที่ใช้ในการสอน (Ground truth) ตามตารางที่ 3.7

ตารางที่ 3.7 ตารางประเมินประสิทธิภาพของโมเดลของโมเดลตรวจจับเมฆด้วย U-Net

| Model | Pre-trained as encoder | Dataset | MeanIOU | Accuracy | Jaccard coefficient | Loss |
|-------|---------------------------|----------------|---------|----------|---------------------|------|
| U-Net | - | Train set | | | | |
| | | Validation set | | | | |
| | | Test set | | | | |

3.3.3 การออกแบบการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวณได้จากผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น

ในส่วนนี้ผู้จัดทำต้องการประเมินค่าความคลาดเคลื่อนจากการคำนวณปริมาณเมฆจากการตรวจจับเมฆอัตโนมัติ ด้วยวิธีของ Otsu's Method และ U-Net เปรียบเทียบกับค่าปริมาณเมฆที่คำนวณได้จากผู้เชี่ยวชาญ เพื่อดูประสิทธิภาพในการทำงานของโมเดลตรวจจับเมฆอัตโนมัติที่สร้างขึ้นกับชุดข้อมูลจากมจธ.วิทยาเขตบางขุนเทียน (ชุดข้อมูลภายใน)

ตารางที่ 3.8 ตารางการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวณได้จากผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น

| Method | จำนวนข้อมูล | RMSE | MSE | MAE | MAPE | R2 |
|--|-------------|------|-----|-----|------|----|
| Otsu's method | | | | | | |
| U-Net without data augmentation | | | | | | |
| U-Net + ResNet34 without data augmentation | | | | | | |
| U-Net with data augmentation | | | | | | |
| U-Net + ResNet34 with data augmentation | | | | | | |

3.4 การออกแบบอัลกอริทึมที่ใช้ในการคำนวณปริมาณเมฆ

ในส่วนนี้จะเป็นการคำนวณปริมาณเมฆ ซึ่งจะคำนวณจากพิกเซลในรูปภาพ Ground truth ที่ได้จากการทำ Cloud detection โดยสีขาวจะเป็นพื้นที่ของเมฆจะมีค่าพิกเซลเท่ากับ 255 พิกเซลสีเทาจะเป็นพื้นที่ของฟ้ามีค่าพิกเซลเท่ากับ 100 และสีดำจะเป็นพื้นที่ของดวงอาทิตย์และพื้นที่ส่วนที่ไม่เกี่ยวข้องในการคำนวณปริมาณเมฆมีค่าพิกเซลเท่ากับ 0 โดยการคำนวณปริมาณเมฆจะสามารถหาจากสัดส่วนพิกเซลของเมฆและผลรวมของพิกเซลท้องฟ้ากับเมฆ มีอัลกอริทึมตามรูปที่ [3.14](#)

Algorithm 1 Calculate Cloud Cover from image

Require: Gray scale image array

Ensure: Cloud cover percentage

```

cloudcover = 0
totalblackpx = 0 // undefined region pixel
totalwhitepx = 0 // cloud pixel
totalpx = 0
totalgraypx = 0 // sky pixel
for pixel in imgArr do
    if pixel == 255 then
        totalwhitepx += 1
    else if pixel == 100 then
        totalgraypx += 1
    else if pixel == 0 then
        totalblackpx += 1
    end if
end for
totalpx = totalgraypx + totalwhitepx + totalblackpx
cloudcover = (totalwhitepx/totalpx)*100

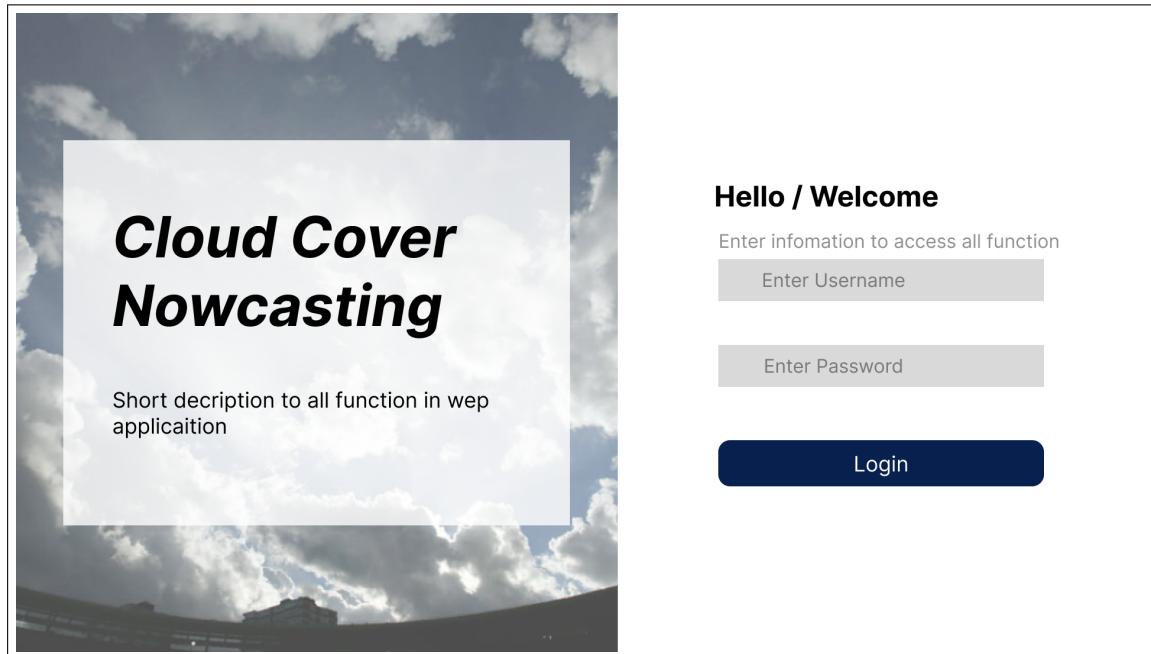
```

รูปที่ 3.14 อัลกอริทึมในการคำนวณปริมาณเมฆจากรูปภาพ

3.5 การออกแบบ Mockup User Interface

ในส่วนนี้จะเป็นการออกแบบหน้า User Interface เมื่อผู้ใช้งานเข้ามาในเว็บแอปพลิเคชัน โดยประกอบไปด้วยหน้าจอดังต่อไปนี้

3.5.1 หน้าจอ Welcome



รูปที่ 3.15 หน้าหลักก่อนเข้าสู่ระบบของเว็บแอปพลิเคชัน

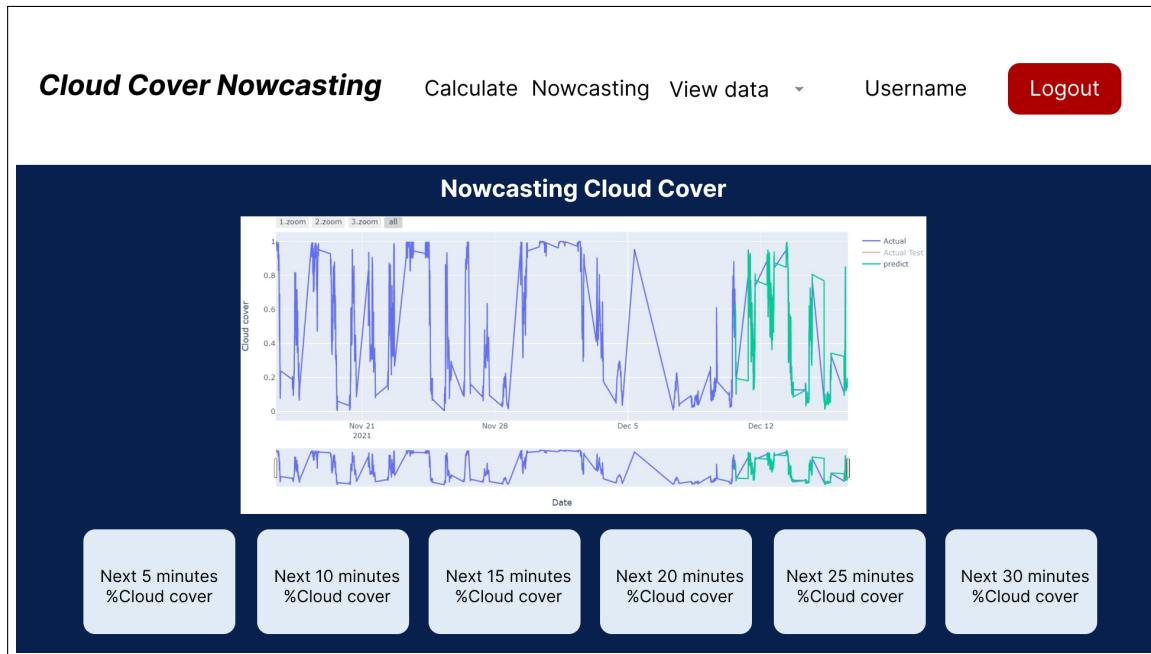
วัตถุประสงค์: เพื่อให้ผู้ใช้เข้าสู่ระบบเพื่อเข้าถึงฟีเจอร์การทำงานของเว็บแอปพลิเคชัน

Components: หน้าจอ Welcome ประกอบด้วย

- ส่วนแสดงผลข้อมูลเว็บแอปพลิเคชัน และ Login form
- Text box เพื่อรับข้อมูล "username" และ "password" ของผู้ใช้งานสำหรับการเข้าสู่ระบบ
- ปุ่ม "Login"

การทำงาน: เมื่อผู้ใช้งานกรอก "username" และ "password" แล้วกดปุ่ม "Login" ระบบจะส่งข้อมูลเพื่อให้ผู้ใช้เข้าสู่ระบบและสามารถเข้าถึงการทำงานของฟีเจอร์ต่างๆ รวมไปถึงแสดงชื่อ "username" ของผู้ใช้งานในส่วนของหน้าจอ Nowcasting ดังรูปที่ 3.16

3.5.2 หน้าจอ Nowcasting



รูปที่ 3.16 หน้าหลักหลังการเข้าสู่ระบบหรือหน้าการแสดง Cloud Cover Nowcasting

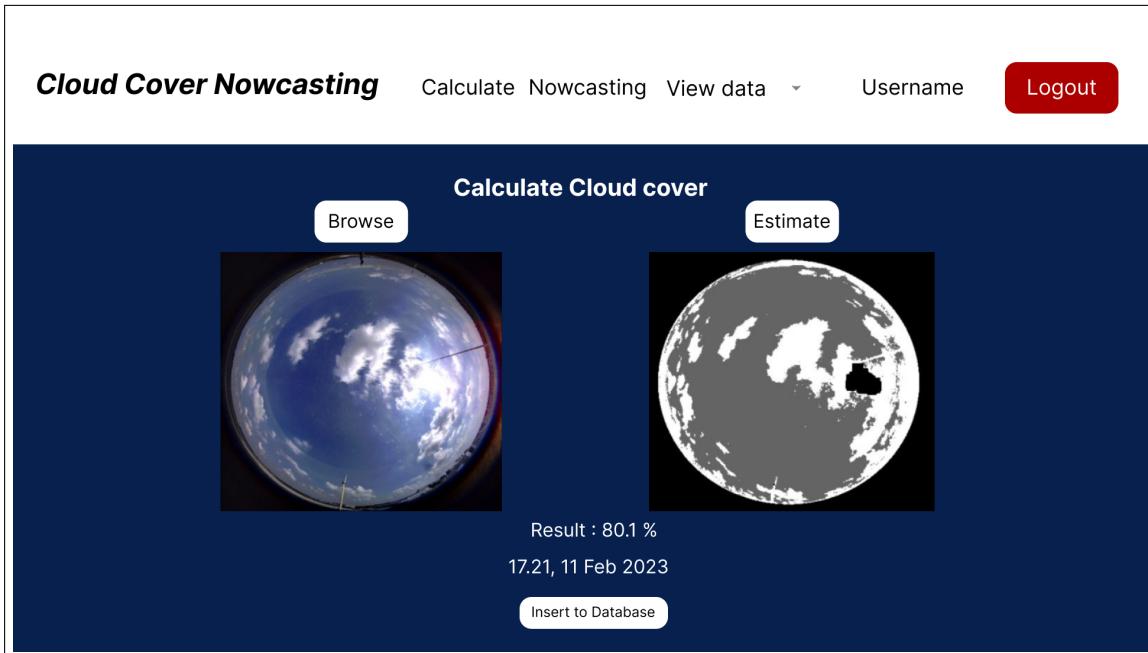
วัตถุประสงค์: เพื่อให้ผู้ใช้งานเลือกฟีเจอร์การทำงานอื่นๆ จากแท็บเมนู รวมไปถึงแสดงผลการ nowcasting และผลการ prediction cloud cover

Components: หน้าจอ Nowcasting ประกอบด้วย

- แท็บเมนู ซึ่งเป็นส่วนประกอบของทุกหน้าจอเมื่อผู้ใช้เดินทางสู่ระบบ จะมีฟีเจอร์การทำงานให้ใช้อยู่ทั้งหมด 4 ฟีเจอร์ โดย
 - Calculate เป็นการทำปริมาณเมฆจากรูปภาพ
 - Nowcasting เป็น Dashboard แสดงผลการพยากรณ์ปริมาณเมฆ
 - View cloud cover image data เป็นฟีเจอร์แรกเมื่อขยายเมนู View data ซึ่งเป็นการดูข้อมูลทั้งหมดจากฐานข้อมูล และการเพิ่มข้อมูลจากการคำนวนปริมาณเมฆด้วยรูปภาพ
 - View cloud cover with weather data เป็นฟีเจอร์ที่สองเมื่อขยายเมนู View data ซึ่งเป็นการดูข้อมูลทั้งหมดจากฐานข้อมูล และการเพิ่มข้อมูลของข้อมูลอนุกรมเวลาของปริมาณเมฆและตัวแปรทางสภาพอากาศ
- Text box แสดงชื่อหัวข้อของหน้าจอ
- Chart box แสดงผลการ Nowcasting ในลักษณะกราฟเส้น
- Text box แสดงผลการ Prediction ค่า Cloud cover

การทำงาน: เมื่อผู้ใช้งานเข้าสู่หน้าจอนี้จะเป็นหน้าของการแสดงผลการพยากรณ์ในช่วงเวลาอีก 5-60 นาทีในอนาคต โดยใช้ข้อมูลในอดีต จากระบบฐานข้อมูล โดยจะแสดง Dashboard ที่ประกอบไปด้วยข้อมูลปริมาณเมฆในช่วงเวลาต่าง ๆ รวมไปถึงผลการพยากรณ์ปริมาณเมฆทั้งหมด 5 - 30 นาทีในอนาคต

3.5.3 หน้าจอ Calculate



รูปที่ 3.17 หน้าการแสดงผลลัพธ์ของการปริมาณเมฆด้วยรูปภาพ

วัตถุประสงค์: เพื่อคำนวนปริมาณเมฆจากรูปภาพและบันทึกผลลงฐานข้อมูล

Components: หน้าจอ Calculate ประกอบด้วย

- Text box แสดงชื่อหัวข้อของหน้าจอ
- ปุ่ม "Browse", "Estimate", และ "Insert to database"
- Image box เพื่อแสดงผลภาพต้นฉบับและภาพที่ทำการแยกพื้นที่แล้ว
- Text box แสดงผลลัพธ์การคำนวนปริมาณเมฆและชื่อของรูปภาพต้นฉบับ

การทำงาน: เมื่อผู้ใช้งานเข้าสู่หน้าจอ Calculate จะเห็นมาสู่หน้าของการหารปริมาณเมฆจากรูปภาพ โดยในหน้านี้ผู้ใช้สามารถอัปโหลดรูปที่ต้องการหารปริมาณเมฆได้โดยการกดปุ่ม "Browse" ซึ่งจะแสดงผลภาพที่ผู้ใช้งานเลือก จากนั้นผู้ใช้งานกดปุ่ม "Estimate" ระบบจะแสดงผลภาพที่ทำการแยกพื้นที่ของเมฆและท้องฟ้าที่ตรวจสอบได้ รวมไปถึงผลลัพธ์การคำนวนหาปริมาณเมฆจากรูปภาพและวันเวลาของรูปภาพต้นฉบับ จากตัวอย่างในรูป 3.17 นี้มี Result อยู่ที่ 80.1 % และหากผู้ใช้งานต้องการบันทึกข้อมูลการหารปริมาณเมฆด้วยรูปภาพนี้ลงในฐานข้อมูล สามารถทำได้โดยการกดปุ่ม "Insert to Database"

3.5.4 หน้าจอ View image data

| Datetime | Cloud Cover | Original image | Segment image |
|---------------------|-------------|-------------------------------|------------------------------|
| 2021_11_16_11_15_00 | 0.781 | original_path | segment_path |
| 2021_11_16_11_16_00 | 0.799 | original_path | segment_path |
| 2021_11_16_11_17_00 | 0.780 | original_path | segment_path |
| 2021_11_16_11_18_00 | 0.787 | original_path | segment_path |
| 2021_11_16_11_19_00 | 0.786 | original_path | segment_path |
| 2021_11_16_11_20_00 | 0.788 | original_path | segment_path |
| 2021_11_16_11_21_00 | 0.780 | original_path | segment_path |
| 2021_11_16_11_22_00 | 0.800 | original_path | segment_path |
| 2021_11_16_11_23_00 | 0.801 | original_path | segment_path |
| 2021_11_16_11_24_00 | 0.799 | original_path | segment_path |
| 2021_11_16_11_25_00 | 0.800 | original_path | segment_path |

รูปที่ 3.18 หน้าของการแสดงข้อมูลจากการคำนวณปริมาณเมฆ

วัตถุประสงค์: เพื่อแสดงผลข้อมูลทั้งหมดจากการคำนวณปริมาณเมฆด้วยรูปภาพ

Components: หน้าจอ View image data ประกอบด้วย

- Text box แสดงชื่อหัวข้อของหน้าจอ
- Table box เพื่อแสดงผลข้อมูลทั้งหมดจากการคำนวณปริมาณเมฆด้วยรูปภาพ

การทำงาน: เมื่อผู้ใช้งานเข้าสู่หน้าจอ โดยเลือกพิเจอร์ "View cloud cover image data" จากภายในเมนู "View data" จะเป็นการดูข้อมูลทั้งหมดจากฐานข้อมูลของการคำนวณปริมาณเมฆด้วยรูปภาพ และการเพิ่มข้อมูลจากการคำนวณปริมาณเมฆด้วยรูปภาพ

3.5.5 หน้าจอ View weather data

The screenshot shows a web-based application titled "Cloud Cover Nowcasting". At the top, there is a navigation bar with links for "Calculate Nowcasting", "View data", "Username", and a red "Logout" button. Below the navigation bar, the main content area has a dark blue header with the title "Cloud cover with weather data". The main body contains a table with the following data:

| Datetime | Eeg | Relative Humidity | Wind Speed | Cloud Cover |
|---------------------|-----|-------------------|------------|-------------|
| 2021_11_16_11_15_00 | 0 | 88.272 | 1.7196 | 0.781 |
| 2021_11_16_11_16_00 | 0 | 88.646 | 1.6868 | 0.799 |
| 2021_11_16_11_17_00 | 0 | 88.874 | 1.7732 | 0.780 |
| 2021_11_16_11_18_00 | 0 | 88.9374 | 1.7358 | 0.787 |
| 2021_11_16_11_19_00 | 0 | 88.5572 | 1.7244 | 0.786 |
| 2021_11_16_11_20_00 | 0 | 87.7722 | 1.7124 | 0.788 |
| 2021_11_16_11_21_00 | 0 | 88.5572 | 1.7732 | 0.780 |
| 2021_11_16_11_22_00 | 0 | 87.7722 | 1.7358 | 0.800 |
| 2021_11_16_11_23_00 | 0 | 88.646 | 1.7196 | 0.801 |
| 2021_11_16_11_24_00 | 0 | 88.272 | 1.6868 | 0.799 |

รูปที่ 3.19 หน้าของการแสดงข้อมูลปริมาณเมฆกับตัวแปรสภาพอากาศ

วัตถุประสงค์: เพื่อแสดงข้อมูลทั้งหมดของปริมาณเมฆกับตัวแปรสภาพอากาศ

Components: หน้าจอ View weather data ประกอบด้วย

- Text box แสดงชื่อหัวข้อของหน้าจอนี้
- Table box เพื่อแสดงผลข้อมูลทั้งหมดของปริมาณเมฆกับตัวแปรสภาพอากาศ

การทำงาน: เมื่อผู้ใช้งานเข้าสู่หน้าจอนี้ โดยเลือกไฟล์เจ่อร์ "View cloud cover with weather data" จากภายในเมนู "View data" จะเป็นการดูข้อมูลทั้งหมดจากฐานข้อมูลที่เป็นลำดับอนุกรมเวลาของปริมาณเมฆและตัวแปรทางสภาพอากาศ

3.6 การออกแบบ Function

ในส่วนนี้จะเป็นการออกแบบ Function การทำงานทั้งหมดภายในเว็บแอปพลิเคชัน โดยประกอบไปด้วยหน้าจอต่อไปนี้

3.6.1 Function Login

Input parameter: Username และ Password

Output parameter: Username

Operation: รับ Username และ Password แล้วทำการ Authentication กับฐานข้อมูลเพื่อให้ผู้ใช้เข้าสู่ระบบได้ โดยจะแสดงผลหน้าจอ Nowcasting เมื่อเข้าสู่ระบบสำเร็จ และแสดงผลข้อความผิดพลาดเมื่อเข้าสู่ระบบไม่สำเร็จ

3.6.2 Function Logout

Input parameter: การกดปุ่ม "Logout"

Output parameter: Logout successfull

Operation: ทำการออกจากระบบให้กับผู้ใช้ โดยการลบ session การทำงานของผู้ใช้

3.6.3 Function Dashboard

Input parameter: Datetime, Cloud cover, Irradiation, Relative humidity, Temperature และ Wind speed จาก database

Output parameter: แสดงผลแต่ละค่าในรูปแบบกราฟเส้น

Operation: รับค่า Input จาก database ที่เป็นส่วนของ weather table มาทำการสร้างกราฟเส้น

3.6.4 Function Prediction

Input parameter: Datetime, Cloud cover

Output parameter: ผลการทำนาย Cloud cover ในวิก 5-30 นาทีข้างหน้า

Operation: รับค่า Input เข้าโน้มเดลการทำนายผลปริมาณเมฆและส่งผลอภินิหาร

3.6.5 Function Calculate

Input parameter: Original image

Output parameter: Segment image, Result of calculation, Meta-data ของภาพ

Operation: รับ Input เป็นไฟล์รูปภาพเพื่อนำเข้าฟังก์ชันในการแยกส่วนพื้นที่ของท้องฟ้าและเมฆ จากนั้นคำนวณปริมาณเมฆจากรูปภาพโดยการนับ pixel และส่งออกผลลัพธ์ทั้งหมดรวมไปถึงวันและเวลาของรูปภาพที่ได้จาก Meta-data

3.6.6 Function Insert to database

Input parameter: Original image, Segment image, Result of calculation และ meta-data ของภาพ

Output parameter: Insert to database success

Operation: รับ Input ทั้งหมด เพื่อเพิ่มลงระบบฐานข้อมูลโดยใช้ฟังก์ชัน sql

3.6.7 Function Table

Input parameter: CSV file

Output parameter: Table

Operation: อ่านไฟล์ CSV จากนั้นทำการสร้างตาราง

3.6.8 Function Show popup image in Table

Input parameter: Path of Image

Output parameter: Image

Operation: รับ Path of Image ที่ผู้ใช้งานคลิก และนำมาแสดงผลรูปภาพในรูปแบบ Popup

บทที่ 4 ผลการดำเนินงาน

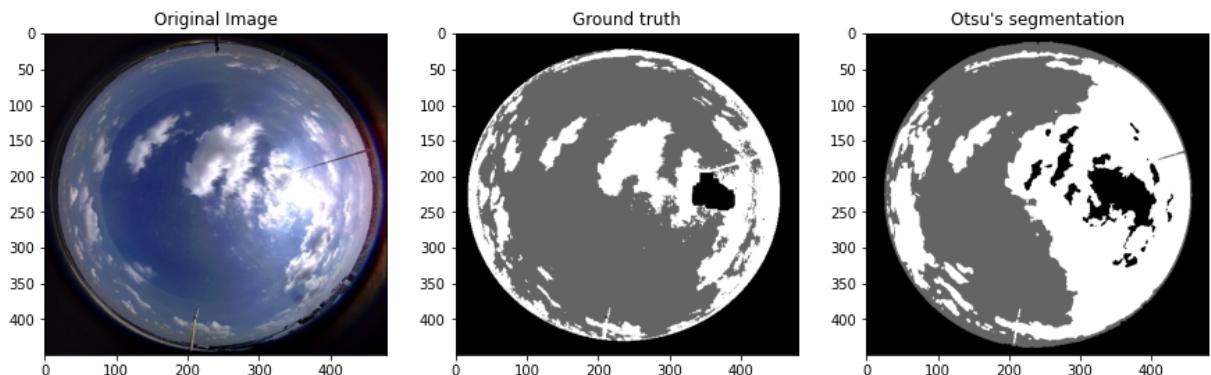
4.1 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆ (Cloud detection)

จากการสร้างโมเดลสำหรับตรวจจับเมฆด้วยวิธีของ Otsu's method และการเรียนรู้โครงข่ายประสาทเทียมโครงสร้างสถาปัตยกรรมแบบ U-Net Architecture ผู้จัดทำได้กำหนดขนาดรูปและช่องสีอยู่ที่ 224x224x3 ซึ่งมีผลการทดลองดังนี้

4.1.1 ผลการทดลองของวิธี Otsu's method ตรวจจับเมฆ

จากการทดลองใช้ Otsu's method ใน การตรวจจับเมฆ โดยใช้ชุดข้อมูลทั้งหมดใน WSISEG-database จำนวน 400 รูป โดยในแต่ละคลาส จะกำหนดค่า threshold ในการทำ otsu's method ดังนี้

- Clear sky class : $T = 0$
- Sun glare class : $T = 240$
- Cloud class : $T = 255$

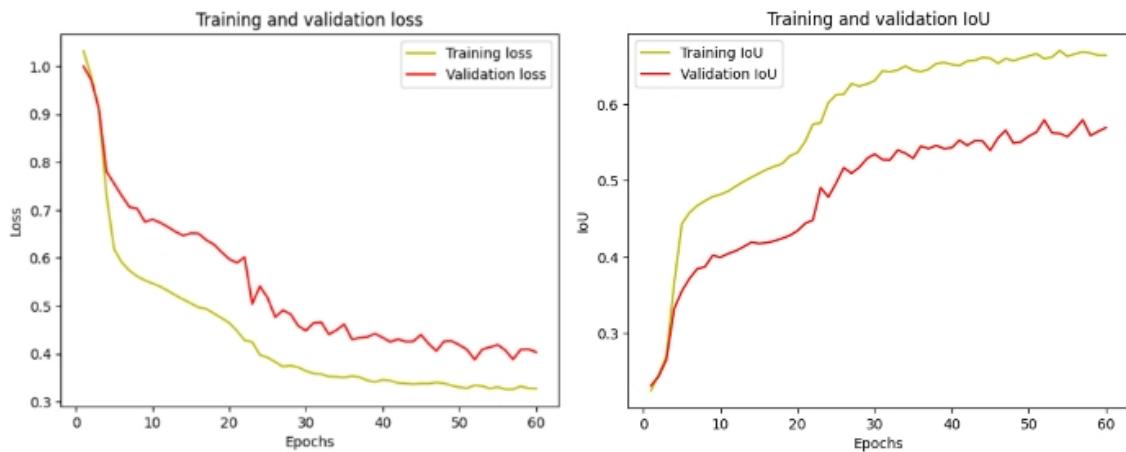


รูปที่ 4.1 ผลการ prediction ของวิธี Otsu's method เปรียบเทียบกับรูปต้นฉบับ และ ground truth

จากรูปที่ 4.1 ผลการทำ segmentation จาก Otsu's method เปรียบเทียบกับ Ground truth พบว่าโมเดลนี้ยังไม่สามารถตรวจจับพิกเซลของเมฆ และพิกเซลของแสงอาทิตย์ได้ละเอียดมากนัก จากรูปจะเห็นได้ว่า Otsu's method บางพิกเซลที่เป็นท้องฟ้า โมเดลนี้กลับตรวจจับว่าเป็นเมฆ และเมื่อจะตรวจจับพิกเซลที่เป็นแสงอาทิตย์ โมเดลนี้กลับตรวจจับบางพิกเซลเป็นเมฆ เนื่องจากค่าพิกเซลของเมฆ ท้องฟ้า และแสงอาทิตย์บางส่วนมีค่าใกล้เคียงกับค่า threshold ที่กำหนดไว้ซึ่งต้น

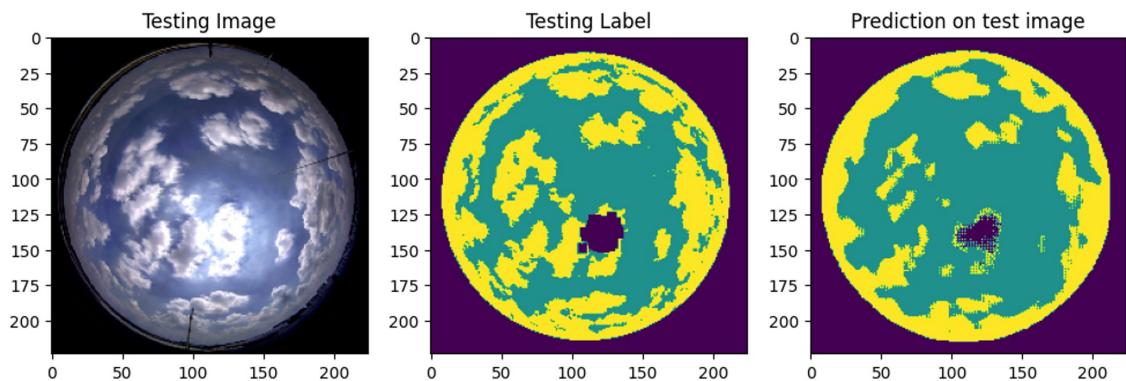
4.1.2 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net without Data augmentation

- Total params: 1,941,139
- Trainable params: 1,941,139
- Non-trainable params: 0
- Optimizer: AdamW
- Learning rate: 0.0001
- batch_size: 32
- epochs: 60



รูปที่ 4.2 กราฟแสดง loss and jaccard curve ของ U-Net without data augmentation

จากรูปที่ 4.2 เส้นสีเหลืองแสดงส่วนของ training และเส้นสีแดงแสดงส่วนของ validation จากกราฟ training loss และ validation loss จะเห็นได้ว่าการเรียนรู้ของโมเดลกับชุดข้อมูลนี้กราฟมีลักษณะเหมือนไปมา อาจเป็นเพราะว่าไม่ได้ตั้งค่า learning rate ที่สูงมากเกินไป และจากกราฟ training IoU และ validation IoU จะเห็นได้ว่า training IoU มีแนวโน้มที่จะสามารถเพิ่ม accuracy ได้อีกเมื่อมีค่า epoch ที่สูงขึ้น

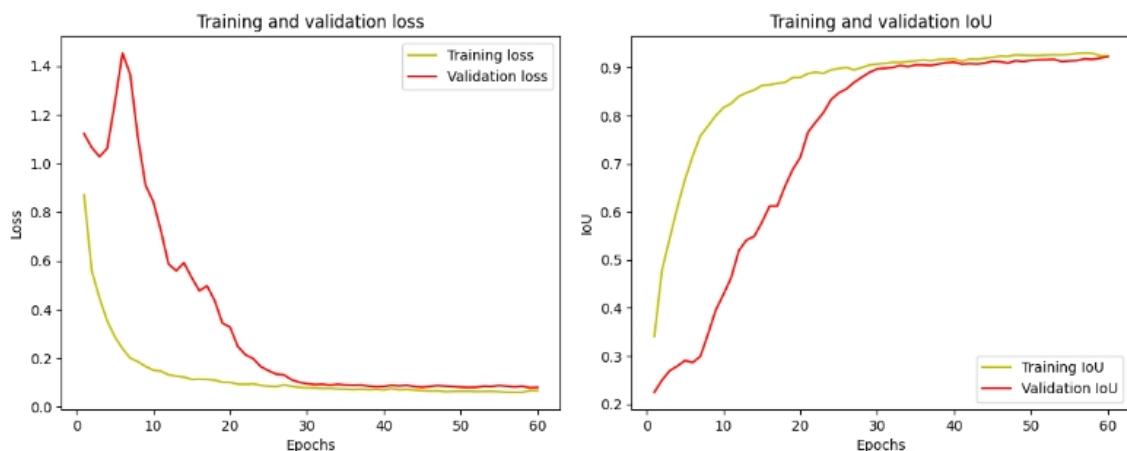


รูปที่ 4.3 ผลการ prediction ของโมเดล U-Net without data augmentation เปรียบเทียบกับ Ground truth

จากรูป 4.3 สีม่วงเป็นพิกเซลของแสงอาทิตย์ และพื้นที่ไม่เกี่ยวข้อง สีเขียวเป็นพิกเซลของท้องฟ้า และสีเหลืองเป็นพิกเซลของเมฆ เมื่อเปรียบเทียบกับ prediction ที่เป็น ground truth พบว่าไม่ได้สามารถตรวจจับเมฆได้โดยถูกต้องกับ ground truth แต่ยังมีบางส่วนในรูปที่เป็น prediction ยังไม่สามารถตรวจจับพิกเซลเมฆ และพิกเซลแสงอาทิตย์ได้ครบถ้วนมากนัก

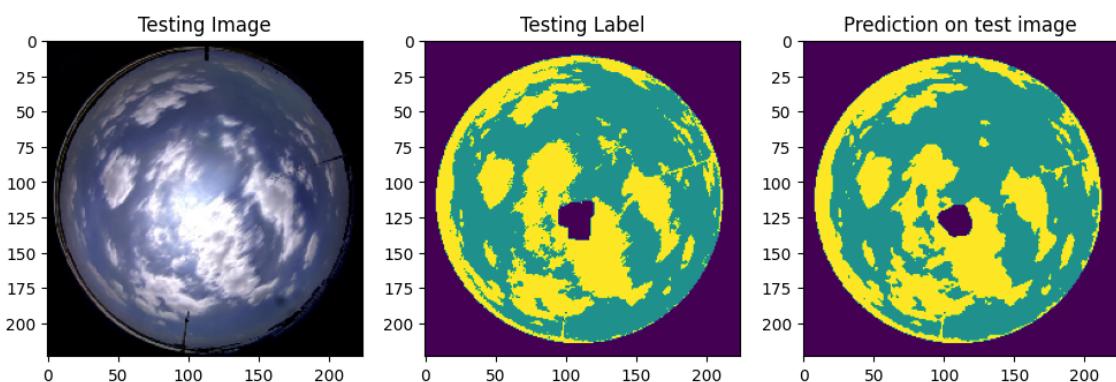
4.1.3 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net using encoder ResNet34 without Data augmentation

- Total params: 24,456,444
- Trainable params: 3,167,350
- Non-trainable params: 21,289,094
- Optimizer: AdamW
- Learning rate: 0.0001
- batch_size: 8
- epochs: 60



รูปที่ 4.4 กราฟแสดง loss and jaccard curve ของ U-Net using encoder ResNet34 without Data augmentation

จากรูปที่ 4.4 เส้นสีเหลืองแสดงส่วนของ training และเส้นสีแดงแสดงส่วนของ validation จากกราฟ training loss และ validation loss จะเห็นได้ว่าในช่วง epoch ที่ 0-30 training loss คู่เข้าสู่ค่า 0 ได้เร็วกว่า validation loss และ training IoU คู่เข้าสู่ค่า 0.9 ได้เร็วกว่า validation IoU ช่วง epoch ที่ 30 ขึ้นไปจะเห็นได้ว่าทั้งกราฟ training, validation loss และ IoU ทั้ง 2 เส้นจะเริ่มคู่เข้าหากัน แสดงให้เห็นว่าโมเดลนี้สามารถเริ่มเรียนรู้กับชุดข้อมูลนี้ได้มากขึ้น โดยรวมกราฟ loss ไม่มีแนวโน้มที่จะเกิด overfitting ได้เลย

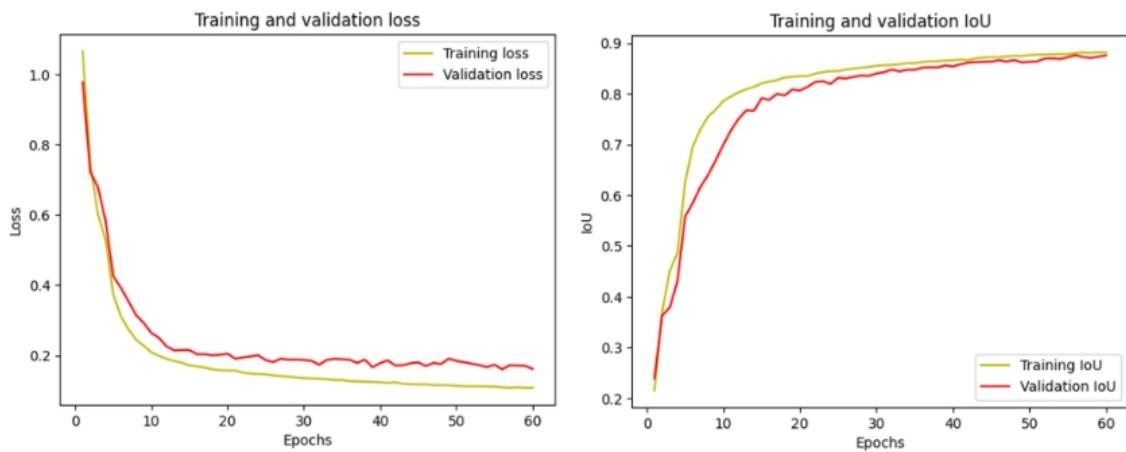


รูปที่ 4.5 ผลการ prediction ของโมเดล U-Net using ResNet34 without Data augmentation เปรียบเทียบกับ Ground truth

จากรูป 4.5 สีม่วงเป็นพิกเซลของแสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้อง สีเขียวเป็นพิกเซลของท้องฟ้า และสีเหลืองเป็นพิกเซลของเมฆ เมื่อเปรียบเทียบรูปที่เป็น prediction กับ ground truth พบร่วมกันนี้สามารถตรวจจับพิกเซลของเมฆ และพิกเซลของแสงอาทิตย์ได้ใกล้เคียงกับ ground truth

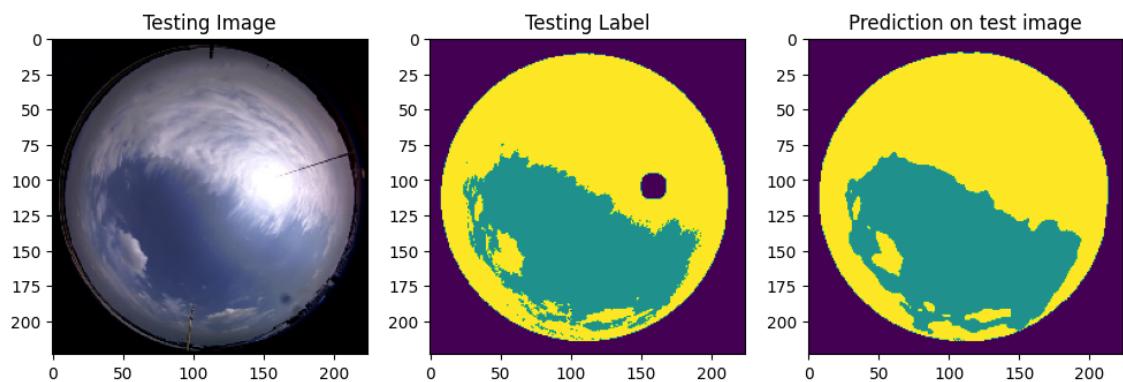
4.1.4 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net with Data augmentation

- Total params: 1,941,139
- Trainable params: 1,941,139
- Non-trainable params: 0
- Optimizer: AdamW
- Learning rate: 0.0001
- batch_size: 32
- epochs: 60



รูปที่ 4.6 กราฟแสดง loss and jaccard curve ของ U-Net with data augmentation

จากรูปที่ 4.6 เส้นสีเหลืองแสดงส่วนของ training และเส้นสีแดงแสดงส่วนของ validation จากกราฟ training loss และ validation loss จะเห็นได้ว่ากราฟค่อนข้างที่จะ good fit ถึงแม้ลักษณะของกราฟมีการเหวี่ยงเล็กน้อย จากกราฟ Validation IoU จะเห็นได้ว่ากราฟเริ่มสู่เข้าสู่ 0.9 ตั้งแต่ epoch ที่ 40

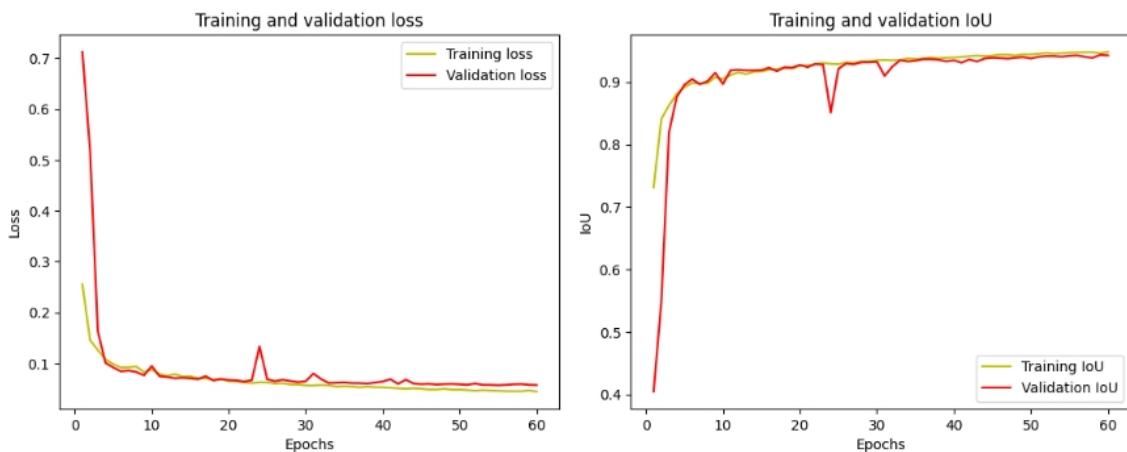


รูปที่ 4.7 ผลการ prediction ของโมเดล U-Net with data augmentation เปรียบเทียบกับ Ground truth

จากรูปที่ 4.7 สม่วงเป็นพิกเซลของแสงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้อง สีเขียวเป็นพิกเซลของท้องฟ้า และสีเหลืองเป็นพิกเซลของเมฆ เมื่อเปรียบเทียบรูปที่เป็น prediction กับ ground truth พบร้าโมเดลนี้สามารถตรวจจับพิกเซลของเมฆได้ค่อนข้างดี แต่การตรวจจับพิกเซลของดวงอาทิตย์นั้นยังไม่ดีมากนัก

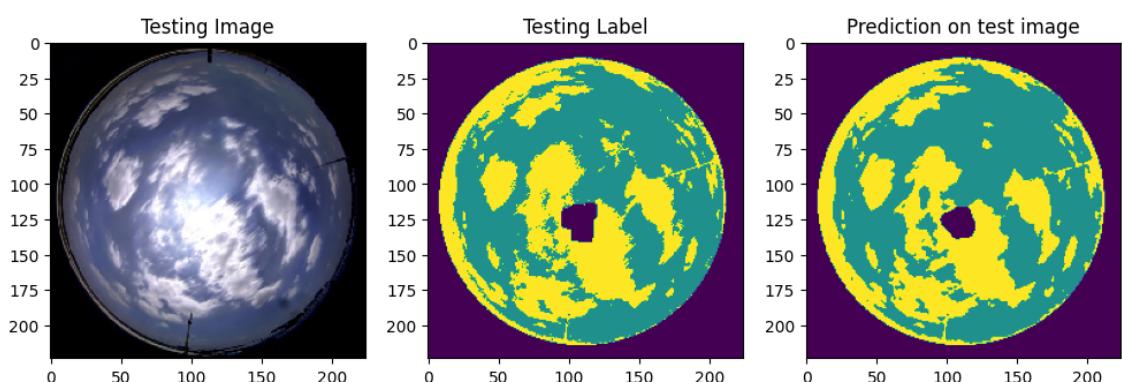
4.1.5 ผลการทดลองของโมเดลสำหรับตรวจจับเมฆด้วย U-Net using encoder ResNet34 with Data augmentation

- Total params: 24,456,444
- Trainable params: 3,167,350
- Non-trainable params: 21,289,094
- Optimizer: AdamW
- Learning rate: 0.0003
- batch_size: 4
- epochs: 60



รูปที่ 4.8 กราฟแสดง loss and jaccard curve ของ U-Net using encoder ResNet34 with Data augmentation

จากรูปที่ 4.8 เส้นสีเหลืองแสดงส่วนของ training และเส้นสีแดงแสดงส่วนของ validation จากกราฟ training loss และ validation loss จะเห็นได้ว่า training loss ลุ่เข้าใกล้ 0 ได้เร็วกว่า validation loss จาก training IoU ลุ่เข้าใกล้ 0.9 ได้เร็วกว่า validation IoU เมื่อโมเดลเรียนรู้กับข้อมูลไปเรื่อย ๆ จะเห็นว่าช่วง epoch ที่ 5 ขึ้นไปทั้ง validation loss และ validation IoU จะเริ่มคงที่และมีค่า IoU เพิ่มขึ้นและค่า loss น้อยลงเมื่อ epoch สูงขึ้น



รูปที่ 4.9 ผลการ prediction ของโมเดล U-Net using ResNet34 with data augmentation เปรียบเทียบกับ Ground truth

จากรูปที่ 4.9 สามารถเห็นว่า prediction ของ模型 U-Net using ResNet34 with data augmentation สามารถตรวจจับเมฆได้ดีเยี่ยม โดย prediction ของ模型มีความคล้ายคลึงกับ ground truth มาก แม้จะมีความไม่แม่นยำในบางจุด เช่น บริเวณกลางเมฆที่มีสีเหลืองใน testing label แต่ prediction ของ模型มีสีเหลืองที่คล้ายคลึงกัน แสดงถึงความสามารถในการจัดแบ่งพิกเซลของเมฆได้ดี

4.1.6 สรุปผลประสิทธิภาพของโมเดลตรวจจับเมฆ

ตารางที่ 4.1 ตารางประเมินประสิทธิภาพของโมเดลตรวจจับเมฆ

| Model+Hyperparameter | Pre-trained as Encoder | Dataset | MeanIOU | Accuracy | Jaccard coefficient | Loss |
|--|------------------------|----------------|---------|----------|---------------------|---------|
| Otsu's method | - | All dataset | 0.5324 | 0.6748 | 0.5155 | 88.1226 |
| U-Net without data augmentation + lr=0.0001 + batch size=32 | - | Train set | - | 0.8659 | 0.6640 | 0.3275 |
| | | Validation set | - | 0.8274 | 0.5692 | 0.4036 |
| | | Test set | 0.6801 | 0.8299 | 0.5730 | 0.4011 |
| U-Net using without data augmentation + lr=0.0001 + batch size=8 | resnet34 | Train set | - | 0.9526 | 0.8715 | 0.1118 |
| | | Validation set | - | 0.8579 | 0.7297 | 0.6107 |
| | | Test set | 0.7724 | 0.8698 | 0.7469 | 0.5189 |
| U-Net with data augmentation + lr=0.0001 + batch size=32 | - | Train set | - | 0.9568 | 0.8813 | 0.1078 |
| | | Validation set | - | 0.9476 | 0.8759 | 0.1614 |
| | | Test set | 0.9099 | 0.9530 | 0.8879 | 0.1488 |
| U-Net with data augmentation + lr=0.0003 + batch size=4 | resnet34 | Train set | - | 0.9815 | 0.9483 | 0.0445 |
| | | Validation set | - | 0.9780 | 0.9427 | 0.0573 |
| | | Test set | 0.9642 | 0.9824 | 0.9507 | 0.0429 |

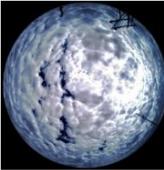
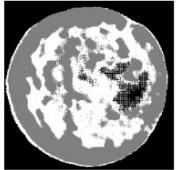
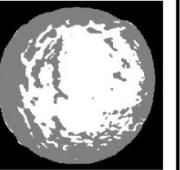
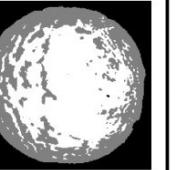
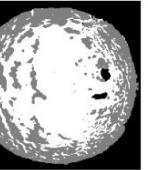
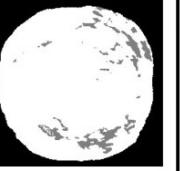
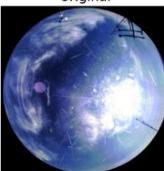
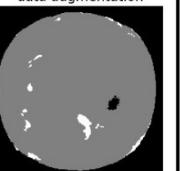
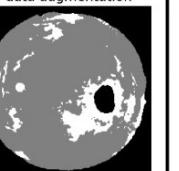
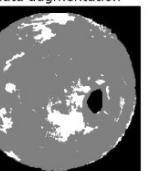
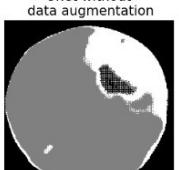
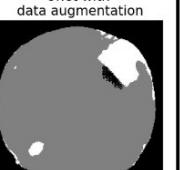
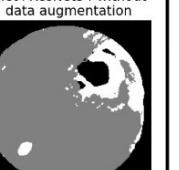
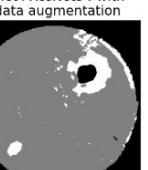
จากตารางที่ 4.1 เมื่อพิจารณาที่ Test set ที่ค่า meanIOU จะเห็นได้ว่า Otsu's method มีค่า meanIOU น้อยที่สุดอยู่ที่ 53.24% และเมื่อพิจารณาการเทรนโมเดลด้วยการไม่เพิ่มชุดข้อมูล พบว่าการตรวจจับเมฆด้วยโมเดล U-Net ที่ใช้ encoder:ResNet34 มีค่า meanIOU อยู่ที่ 77.24% ในทางกลับกันเมื่อพิจารณาการเทรนโมเดลด้วยการเพิ่มชุดข้อมูล พบว่าการตรวจจับเมฆด้วยโมเดล U-Net ที่ใช้ encoder:ResNet34 มีค่า meanIOU อยู่ที่ 96.42% ซึ่งมีประสิทธิภาพดีที่สุด โดยมีค่า Loss อยู่ที่ 0.0429 Accuracy อยู่ที่ 98.24% และ Jaccard coefficient อยู่ที่ 95.07% เมื่อเทียบกับโมเดลอื่น ๆ

ตารางที่ 4.2 ตารางการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆที่คำนวนได้จากผู้เชี่ยวชาญกับโมเดลที่สร้างขึ้น

| Method | จำนวนข้อมูล | RMSE | MSE | MAE | MAPE | R2 |
|--|-------------|--------|--------|--------|----------|---------|
| Otsu's method | 22,026 | 0.5403 | 0.1159 | 0.2919 | 64321.11 | -0.0826 |
| U-Net without data augmentation | 22,026 | 0.5018 | 0.1058 | 0.2518 | 57846.86 | 0.0108 |
| U-Net + ResNet34 without data augmentation | 22,026 | 0.4739 | 0.0862 | 0.2246 | 6064.26 | 0.1943 |
| U-Net with data augmentation | 22,026 | 0.5006 | 0.1099 | 0.2506 | 17846.32 | -0.0269 |
| U-Net + ResNet34 with data augmentation | 22,026 | 0.4874 | 0.0962 | 0.2375 | 54970.76 | 0.1006 |

จากตารางที่ 4.2 เป็นตารางการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณปริมาณเมฆ เพื่อพิจารณาว่าโมเดลใดที่สามารถตรวจจับเมฆและคำนวณปริมาณเมฆได้ใกล้เคียงกับชุดข้อมูลที่นักวิจัยมาช.วิทยาเขตบางเขนเทียนมากที่สุด ซึ่งโมเดลที่มีภาพรวมค่า Error เยอะที่สุดคือ Otsu's method และโมเดลที่มีภาพรวมค่า Error น้อยที่สุดคือ U-Net+ResNet34 without data augmentation แต่เมื่อพิจารณาประสิทธิภาพโมเดลตรวจจับเมฆจากตารางที่ 4.1 ระหว่างโมเดล U-Net+ResNet34 without data augmentation และ U-Net+ResNet34 with data augmentation จะพบว่าโมเดลที่เทรนด้วยการเพิ่มของชุดข้อมูลมี Accuracy ที่สูงกว่ามาก แต่พบว่าเมื่อพิจารณาค่า Error ที่ metric ต่าง ๆ พิจารณาที่ค่า MSE และ MAE พบว่าโมเดลนี้มีค่า Error มากกว่า U-Net+ResNet34 without data augmentation ต่างกันเพียงเล็กน้อย หากวิเคราะห์จากรูปที่ 4.10 พิจารณาที่ Overcast sky จะเห็นได้ว่าโมเดล U-Net+ResNet34 with data augmentation สามารถเก็บรายละเอียดพิกเซลของดวงอาทิตย์ได้ดีมากกว่า U-Net+ResNet34 without data augmentation ทำให้ตอนคำนวณปริมาณเมฆที่เป็นลักษณะของเมฆปากคลุนเต็มท้องฟ้าและมีแสงจากดวงอาทิตย์อยู่ในรูปถ่ายท้องฟ้าด้วยเกิดความคลาดเคลื่อนได้ เพราะพิกเซลตรงจุดนั้นไม่สามารถบ่งบอกได้อย่างชัดเจนว่าเป็นพิกเซลของเมฆหรือดวงอาทิตย์

4.1.7 ตัวอย่างผลการตรวจจับเมฆจากชุดข้อมูลจร.วิทยาเขตบางขุนเทียนด้วยโมเดล U-Net

| Types | Models | | | | |
|------------------------------|---|---|--|---|---|
| Overcast sky |  |  |  |  |  |
| Overcast sky with dark cloud |  |  |  |  |  |
| Partial cloud |  |  |  |  |  |
| Clear sky |  |  |  |  |  |

รูปที่ 4.10 ตัวอย่างรูปผลการตรวจจับเมฆจากชุดข้อมูลจร.วิทยาเขตบางขุนเทียนด้วยโมเดล U-Net

จากรูปที่ 4.10 เป็นตัวอย่างรูปผลการตรวจจับเมฆจากชุดข้อมูลจร.วิทยาเขตบางขุนเทียนด้วยโมเดล U-Net ทั้งหมด โดยรูปที่เป็นตัวอย่างนี้มี 4 หมวดที่จะทำการวิเคราะห์ได้แก่ Overcast sky คือรูปที่มีปริมาณเมฆปกคลุมเต็มท้องฟ้า Overcast sky with dark cloud คือรูปที่มีปริมาณเมฆปกคลุมเต็มท้องฟ้าที่ประกอบไปด้วยเมฆสีดำ Partial cloud คือรูปที่มีปริมาณเมฆบนท้องฟ้าบางส่วน และ Clear sky คือรูปที่ท้องฟ้า晴朗 (ไม่มีเมฆปกคลุม หรือมีเมฆปกคลุมที่ปริมาณน้อยมาก) โดยพิกเซลสีขาวคือพิกเซลของเมฆ พิกเซลสีเทาคือพิกเซลของท้องฟ้า และพิกเซลสีดำคือพิกเซลของดวงอาทิตย์และพื้นที่ที่ไม่เกี่ยวข้อง

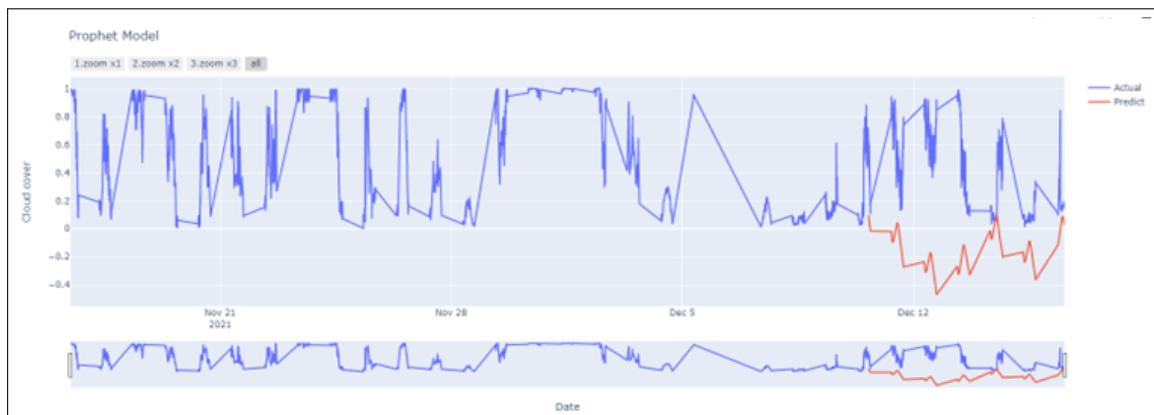
- เมื่อพิจารณาที่โมเดล U-Net without data augmentation พบว่าโมเดลนี้สามารถตรวจจับพิกเซลของดวงอาทิตย์ และพื้นที่ที่ไม่เกี่ยวข้องได้ค่อนข้างแม่นยำ ในทางกลับกันโมเดลนี้ยังไม่สามารถตรวจจับพิกเซลของเมฆได้ดีมากนัก โดยเฉพาะพิกเซลของเมฆที่มีลักษณะที่เบาบางกว่าจุดอื่น และเมื่อพิจารณาที่โมเดล U-Net with data augmentation พบว่าโมเดลนี้มีความสามารถในการตรวจจับของพิกเซลเมฆได้ดีขึ้น พิจารณาได้ที่รูปของ Overcast sky และ Overcast sky with dark cloud
- เมื่อพิจารณาที่โมเดล U-Net+ResNet34 without data augmentation พบว่าโมเดลนี้สามารถตรวจจับพิกเซลของเมฆและพิกเซลของดวงอาทิตย์ได้แม่นยำขึ้น และหากพิจารณาที่รูป Clear Sky กับโมเดล U-Net+ResNet34 with data augmentation พบว่าโมเดลนี้สามารถเก็บรายละเอียดพิกเซลของท้องฟ้าได้แม่นยำกว่า U-Net+ResNet34 without data augmentation

4.2 ผลการทดลองของโมเดลสำหรับพยากรณ์ปริมาณเมฆ (Cloud cover nowcasting)

4.2.1 ผลทดลองการพยากรณ์ปริมาณเมฆด้วย Prophet

4.2.1.1 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นนาที

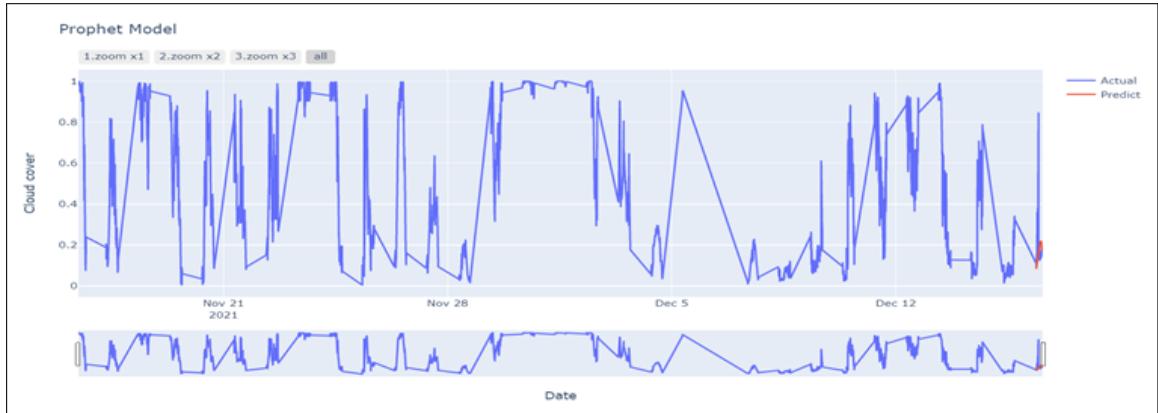
1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้



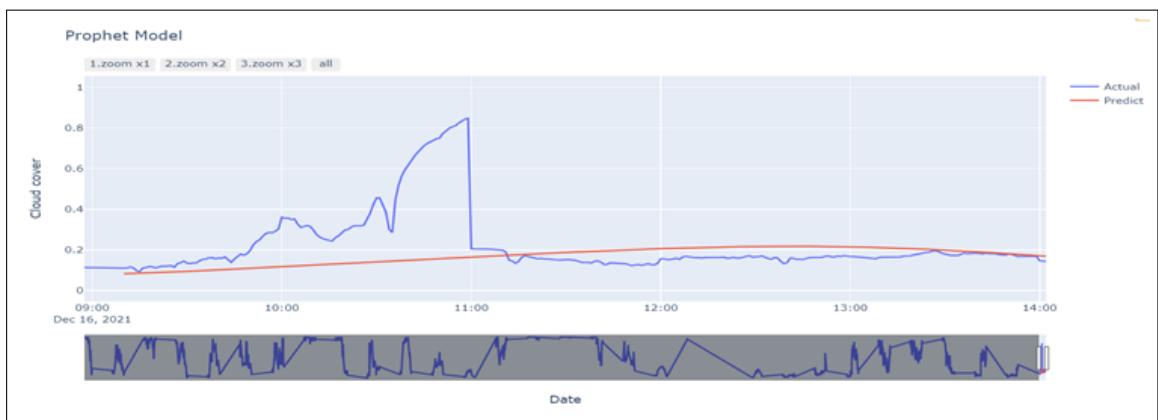
รูปที่ 4.11 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%

จากราฟ 4.11 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เคลื่อนทำนายปริมาณเมฆ (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict น้อยกว่าเส้น Actual โดยเส้นที่ Predict นั้นอยู่ในช่วงติดลบ ซึ่งในความเป็นจริงแล้วปริมาณเมฆที่ทำนายออกมากไม่ควรติดลบ แต่เส้น Actual นั้นอยู่ในช่วงที่เป็นบวก ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual ทำให้มีค่า MAPE อยู่ที่ 1.79844

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้



รูปที่ 4.12 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูลที่เลือกวันสุดท้ายเป็น test set

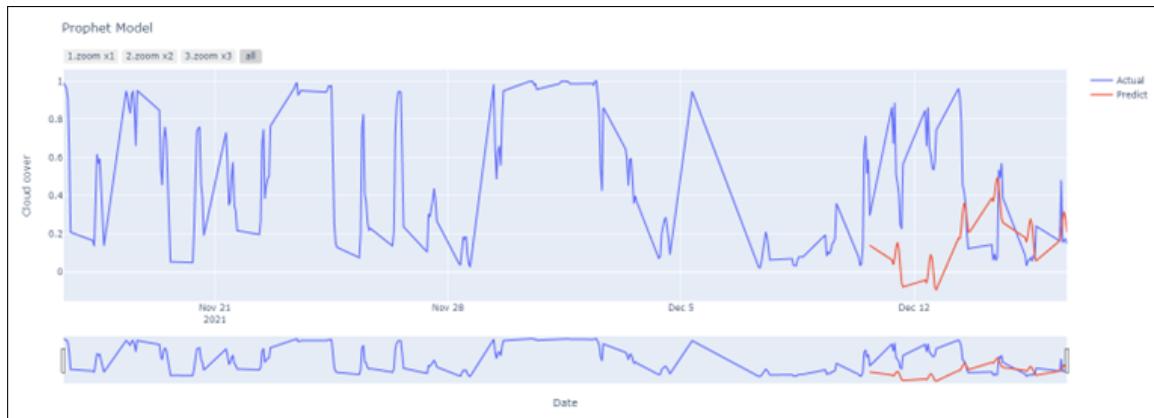


รูปที่ 4.13 การขยายกราฟ 4.12

จากราฟ 4.13 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เดลทำนายปริมาณเมฆ (Predict) โดยพิจารณา ช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict มีช่วงที่น้อยกว่าเส้น Actual โดยเส้นที่ Predict นั้นอยู่ในช่วงที่เป็นบวก และเส้น Actual นั้น อยู่ในช่วงที่เป็นบวกเช่นเดียวกัน ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual ทำให้มีค่า MAPE อยู่ที่ 0.35754

4.2.1.2 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นช่วงโมง

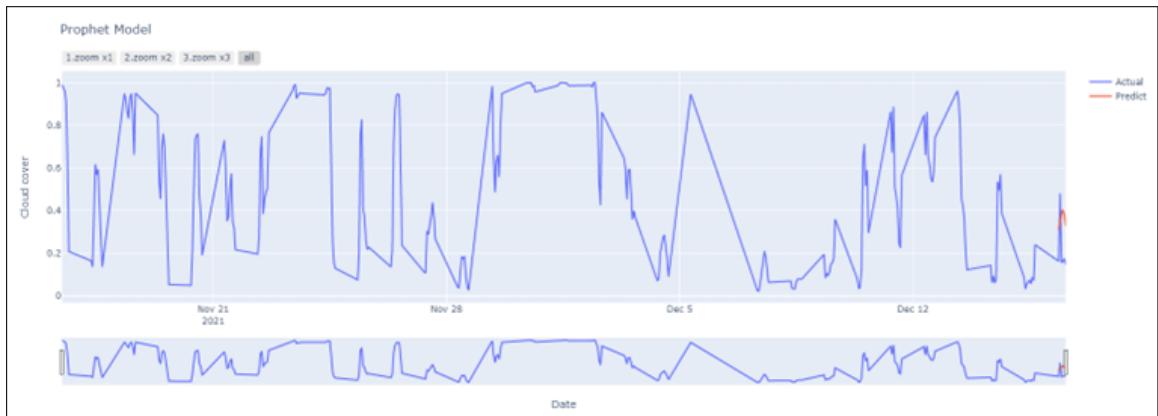
- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้



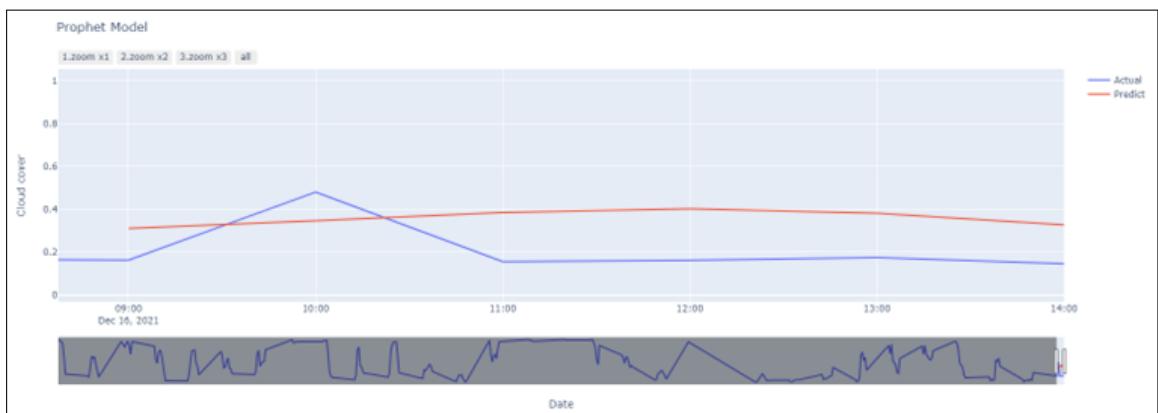
รูปที่ 4.14 กราฟแสดงผลการทำนายเป็นช่วงโมงของ Prophet ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%

จากราฟ 4.14 เส้นสีน้ำเงินคือบริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เคลื่อนทำนายบริมาณเมฆ (Predict) พบว่าเส้น Predict มีบางช่วงที่ติดลบ ซึ่งเป็นช่วงที่มีปริมาณเมฆน้อยกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 1.29596

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้



รูปที่ 4.15 กราฟแสดงผลการทำนายเป็นช่วงโมฆะของ Prophet ของชุดข้อมูลที่เลือกวันสุดท้ายเป็น test set

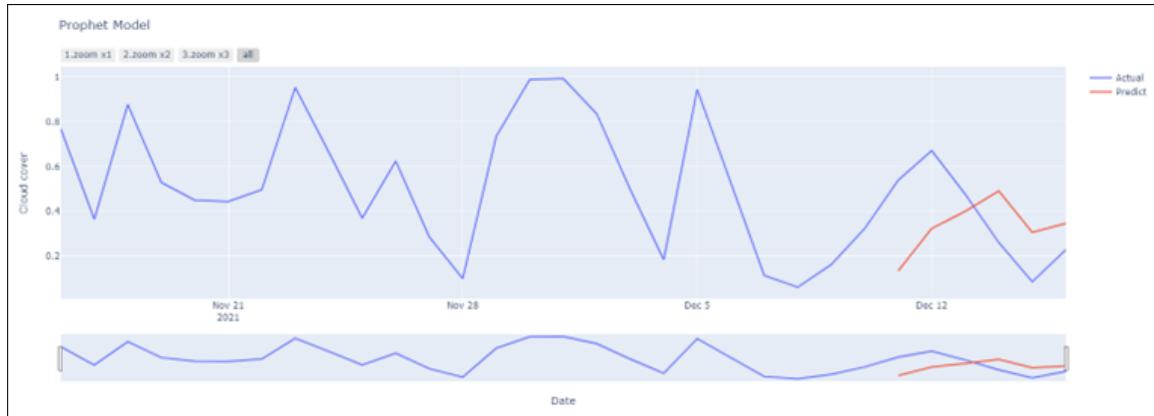


รูปที่ 4.16 การขยายกราฟ 4.15

จากราฟ 4.16 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เคลื่อนทำนายปริมาณเมฆ (Predict) พบร่วมกับเส้นที่ Predict มากกว่าเส้น Actual โดยเส้นที่ Predict นั้นอยู่ในช่วงที่เป็นบวก และเส้น Actual นั้นอยู่ในช่วงที่เป็นบวก ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 1.10121

4.2.1.3 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นรายวัน

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

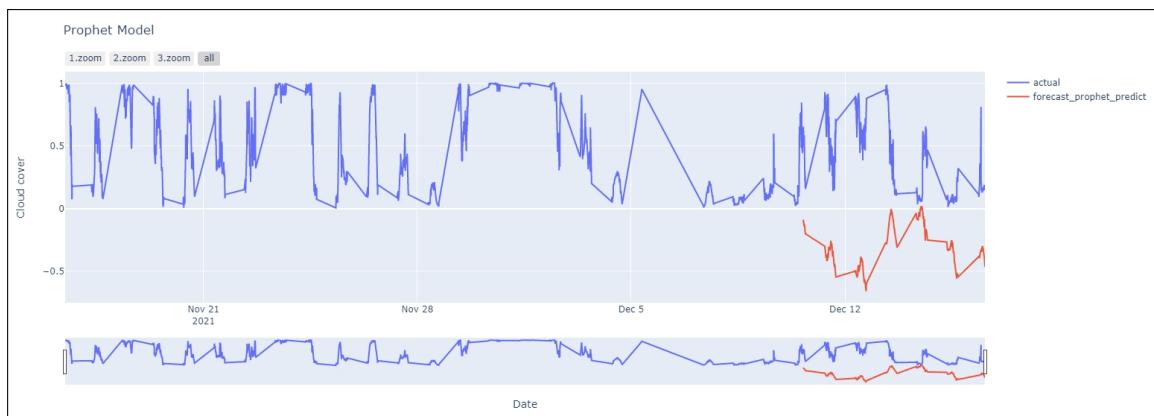


รูปที่ 4.17 กราฟแสดงผลการทำนายเป็นรายวันของ Prophet ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%

จากราฟ 4.17 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เดลการทำนายปริมาณเมฆ (Predict) พบว่าเส้นที่ Predict มีการทำนายที่ช่วงที่น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.90479

4.2.1.4 การแบ่งชุดข้อมูลหลายตัวแปรโดยใช้ตัวแปร Cloud Cover, Relative humidity, Temperature และ Wind speed ที่เป็นนาที

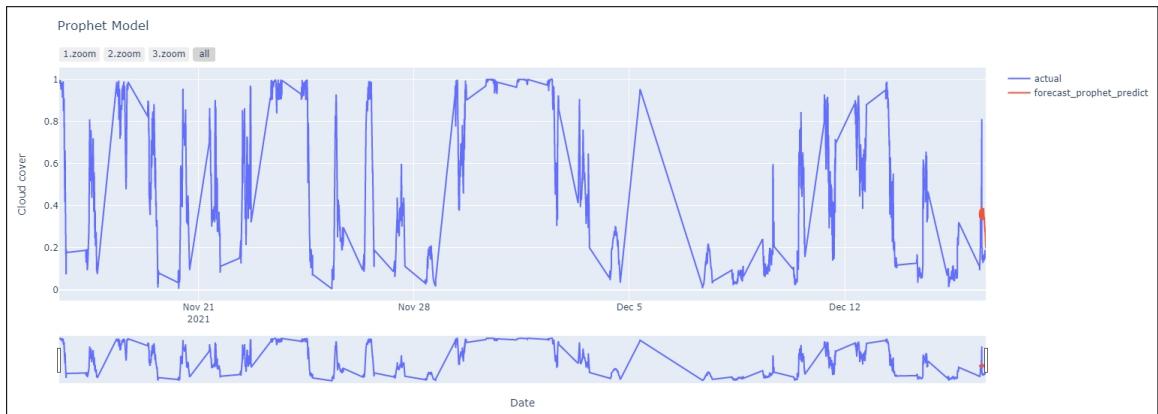
- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้



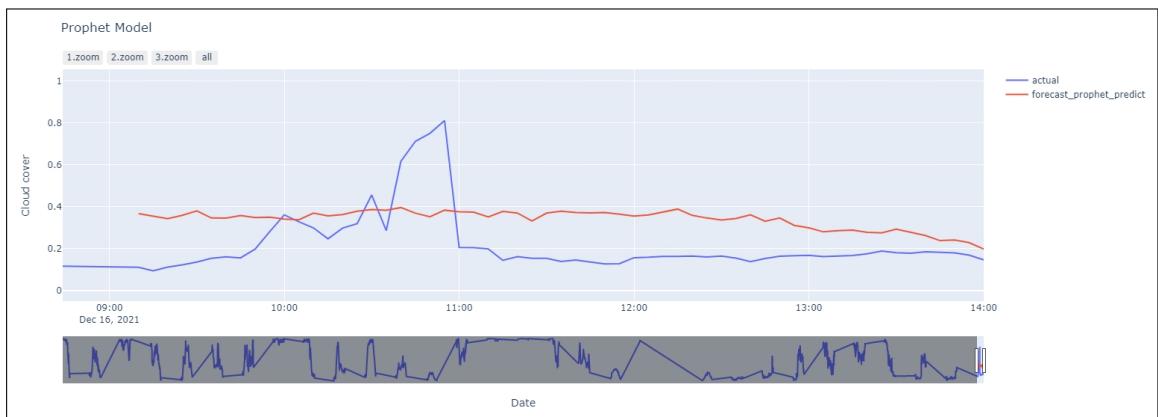
รูปที่ 4.18 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%

จากราฟ 4.18 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เดลการทำนายปริมาณเมฆ (Predict) พบว่าเส้นที่ Predict น้อยกว่าเส้น Actual โดยเส้นที่ Predict นั้นอยู่ในช่วงติดลบ แต่เส้น Actual นั้นอยู่ในช่วงที่เป็นบวก ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 2.72634

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้



รูปที่ 4.19 กราฟแสดงผลการทำนายเป็นนาทีของ Prophet ของชุดข้อมูลที่เลือกวันสุดท้ายเป็น test set



รูปที่ 4.20 การขยายกราฟ 4.19

จากกราฟ 4.20 เส้นสีน้ำเงินคือปริมาณเมฆข้อมูลจริง (Actual) และเส้นสีแดงคือเส้นที่ไม่เดลการทำนายปริมาณเมฆ (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบว่าเส้นที่ Predict มีการทำนายที่มีลักษณะคล้ายกับเส้น Actual แต่เส้น Predict ยังไม่มีความใกล้เคียงกับเส้นActual มีค่า MAPE อยู่ที่ 0.97149

4.2.2 ผลทดลองของการพยากรณ์ปริมาณเมฆด้วย LSTM

4.2.2.1 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นนาที

- Lags 5 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

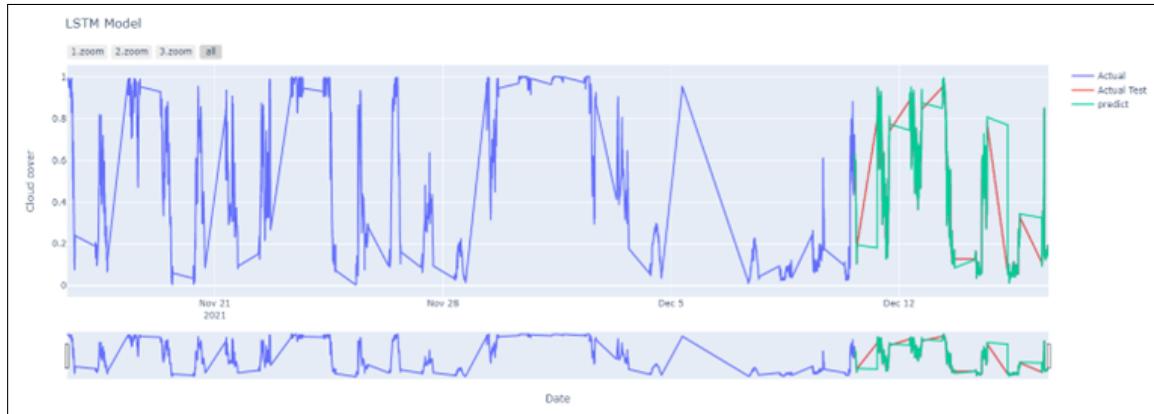
ตารางที่ 4.3 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 5 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.023304 | 0.08622 | 0.002084 | 0.045656 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.026051 | 0.099542 | 0.00229 | 0.047852 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.026095 | 0.104324 | 0.00244 | 0.049398 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.029741 | 0.122874 | 0.002586 | 0.050855 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.03241 | 0.109154 | 0.002627 | 0.051257 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.022388 | 0.092431 | 0.002054 | 0.045323 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.025692 | 0.123222 | 0.002101 | 0.045832 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.022388 | 0.092431 | 0.002054 | 0.045323 |
| Adam | 32 | | 100 | | | 1 | 10 | 0.022172 | 0.078861 | 0.002032 | 0.045083 |
| Adam | 32 | | 100 | 0.1 | | 1 | 10 | 0.022633 | 0.083288 | 0.002084 | 0.045654 |
| Adam | 32 | | 100 | | 50 | 1 | 10 | 0.0229 | 0.11058 | 0.002047 | 0.045243 |
| Adam | 32 | | 100 | | | 1 | 20 | 0.020476 | 0.074577 | 0.002057 | 0.045356 |
| Adam | 32 | | 100 | | | 1 | 30 | 0.023905 | 0.102613 | 0.002147 | 0.046339 |

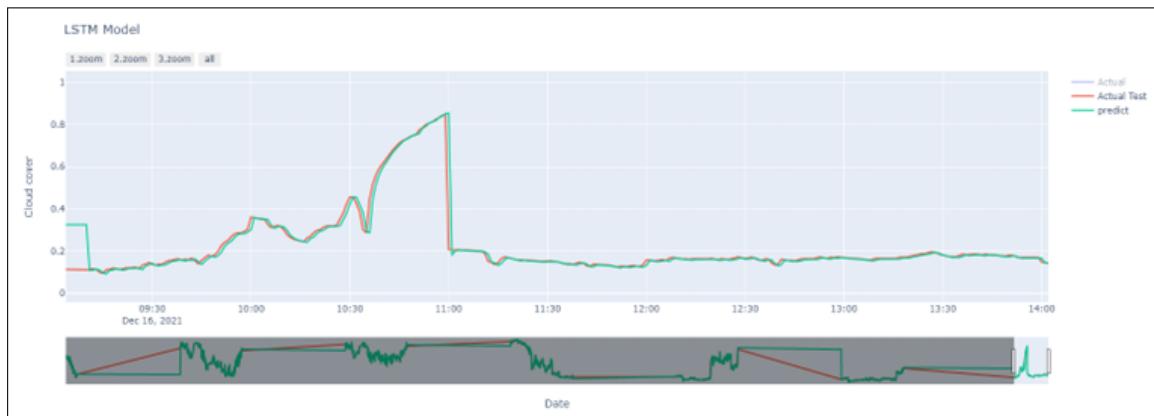
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ทำให้ค่า MAE error และ MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.21



รูปที่ 4.21 กราฟแสดงผลการทำนายเป็นนาที Lags 5 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.22 การขยายกราฟ 4.21

จากราฟ 4.22 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict น้อยกว่าเส้นแดง ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.074577

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

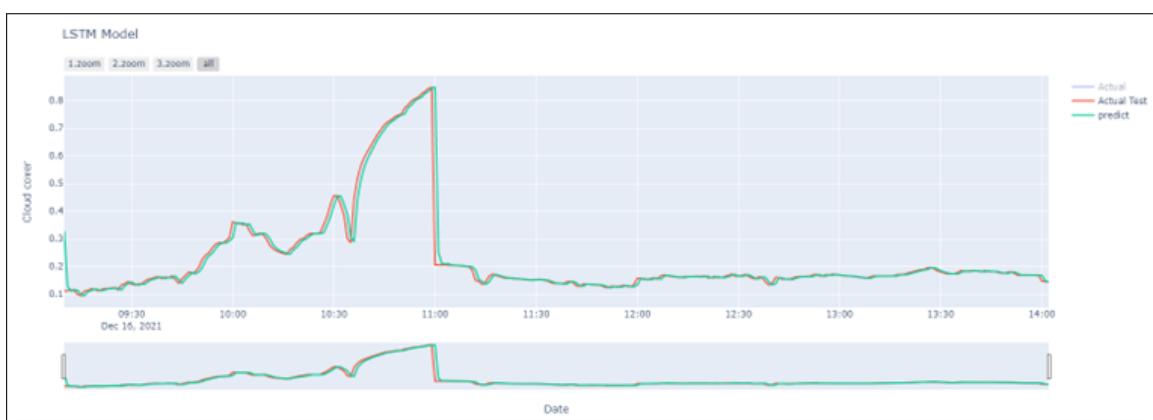
ตารางที่ 4.4 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 5 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error | |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|----------|
| Adam | 32 | | 50 | | | | 1 | 10 | 0.011943 | 0.057024 | 0.001834 | 0.042828 |
| RMSprop | 32 | | 50 | | | | 1 | 10 | 0.014134 | 0.061505 | 0.001822 | 0.042686 |
| Adam | 64 | | 50 | | | | 1 | 10 | 0.016937 | 0.082864 | 0.002659 | 0.05157 |
| Adam | 32 | 0.1 | 50 | | | | 1 | 10 | 0.019179 | 0.106886 | 0.002075 | 0.045556 |
| Adam | 32 | 0.05 | 50 | | | | 1 | 10 | 0.020574 | 0.104792 | 0.001988 | 0.04459 |
| Adam | 32 | 0.01 | 50 | | | | 1 | 10 | 0.012338 | 0.062135 | 0.00183 | 0.042782 |
| Adam | 32 | 0.005 | 50 | | | | 1 | 10 | 0.010836 | 0.051084 | 0.001823 | 0.042695 |
| Adam | 32 | 0.001 | 50 | | | | 1 | 10 | 0.012338 | 0.062135 | 0.00183 | 0.042782 |
| Adam | 32 | 0.005 | 100 | | | | 1 | 10 | 0.011804 | 0.05511 | 0.001828 | 0.04275 |
| Adam | 32 | 0.005 | 50 | 0.1 | | | 1 | 10 | 0.011749 | 0.058326 | 0.001831 | 0.042784 |
| Adam | 32 | 0.005 | 50 | | 50 | | 1 | 10 | 0.01358 | 0.071594 | 0.001902 | 0.043608 |
| Adam | 32 | 0.005 | 50 | | | | 1 | 20 | 0.015791 | 0.077961 | 0.0019 | 0.043592 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error และ MAPE น้อยที่สุด
- Optimizer เป็น RMSprop Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.23



รูปที่ 4.23 กราฟแสดงผลการทำนายเป็นนาที Lags 5 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.23 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลการทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.051084

- Lags 10 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

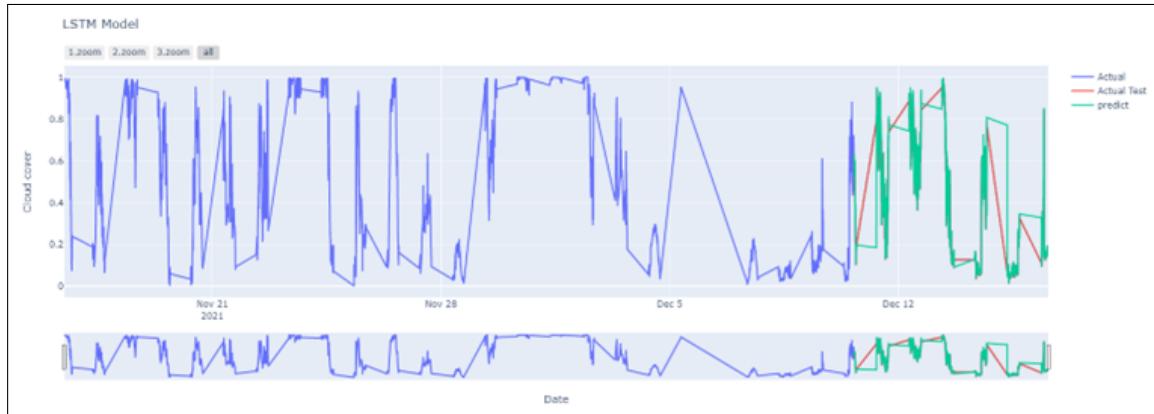
ตารางที่ 4.5 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 10 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.020345 | 0.07531 | 0.00198 | 0.044498 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.024135 | 0.106225 | 0.002155 | 0.046424 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.026757 | 0.106641 | 0.002565 | 0.050645 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.025646 | 0.106002 | 0.002185 | 0.046746 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.022393 | 0.077619 | 0.002132 | 0.046169 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.024583 | 0.11002 | 0.002183 | 0.046722 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.02536 | 0.111802 | 0.002194 | 0.04684 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.024583 | 0.11002 | 0.002183 | 0.046722 |
| Adam | 32 | | 100 | | | 1 | 10 | 0.02033 | 0.077081 | 0.002003 | 0.044754 |
| Adam | 32 | | 50 | 0.1 | | 1 | 10 | 0.022268 | 0.087595 | 0.00209 | 0.045721 |
| Adam | 32 | | 50 | | 50 | 1 | 10 | 0.020372 | 0.077518 | 0.002004 | 0.04477 |
| Adam | 32 | | 50 | | | 1 | 20 | 0.020381 | 0.074959 | 0.002049 | 0.045265 |
| Adam | 32 | | 50 | | | 1 | 30 | 0.021787 | 0.07896 | 0.002027 | 0.045024 |

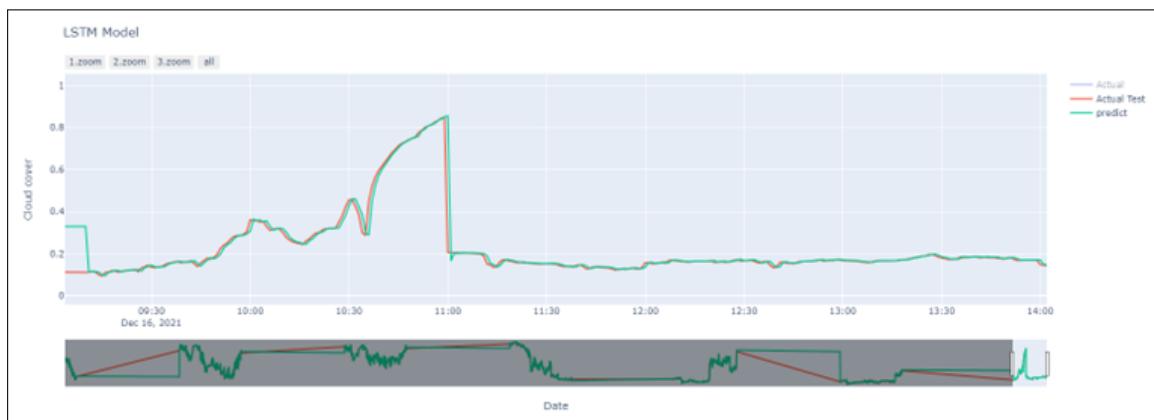
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ทำให้ค่า MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.24



รูปที่ 4.24 กราฟแสดงผลการทำนายเป็นนาที Laes 10 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.25 การขยายกราฟ 4.24

จากกราฟ 4.25 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่ว่าเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.074959

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

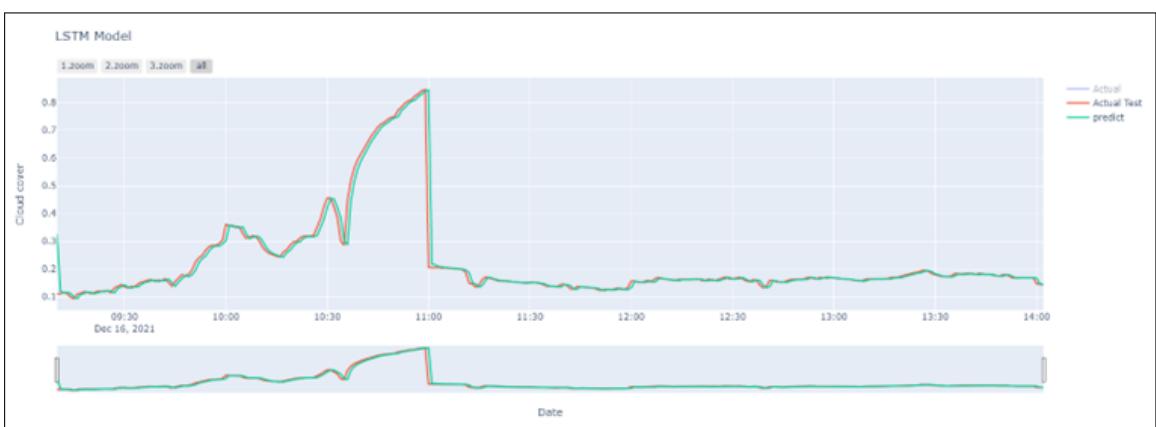
ตารางที่ 4.6 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 10 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.012136 | 0.054429 | 0.001818 | 0.04264 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.015239 | 0.071949 | 0.001859 | 0.043111 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.017186 | 0.087572 | 0.00238 | 0.048786 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.167481 | 0.911509 | 0.035691 | 0.18892 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.024616 | 0.124375 | 0.002104 | 0.045864 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.013826 | 0.067467 | 0.001801 | 0.042441 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.011539 | 0.053332 | 0.001807 | 0.04251 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.013826 | 0.067467 | 0.001801 | 0.042441 |
| Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.018319 | 0.084075 | 0.00191 | 0.043706 |
| Adam | 32 | 0.005 | 50 | 0.1 | | 1 | 10 | 0.011366 | 0.053539 | 0.001819 | 0.042652 |
| Adam | 32 | 0.005 | 50 | | 50 | 1 | 10 | 0.013061 | 0.066181 | 0.001969 | 0.044371 |
| Adam | 32 | 0.005 | 50 | | | 1 | 20 | 0.017468 | 0.09482 | 0.002041 | 0.045172 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Drop out เท่ากับ 0.1 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.01 และ 0.001 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.26



รูปที่ 4.26 กราฟแสดงผลการทำนายเป็นนาที Lag 10 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.26 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลการทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual ทำให้เส้น Predict มีค่า MAPE อยู่ที่ 0.053332

- Lags 15 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

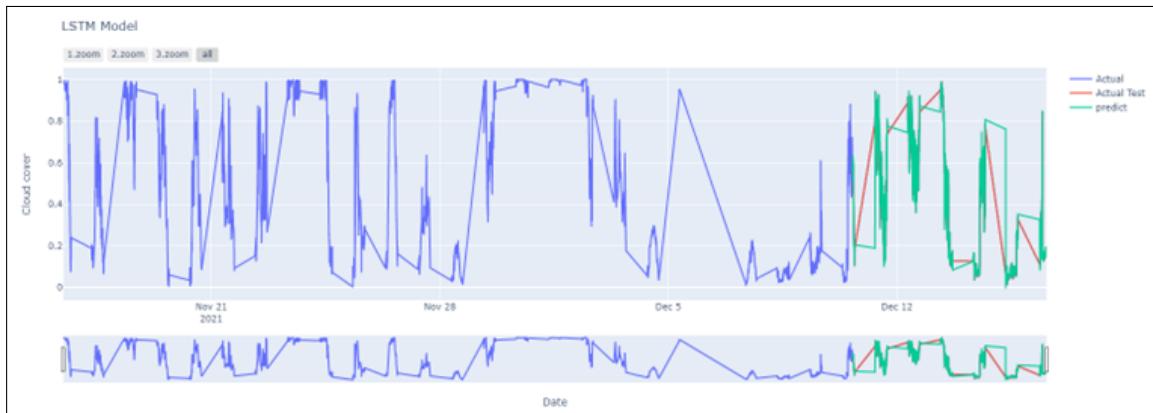
ตารางที่ 4.7 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 15 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.021963 | 0.086015 | 0.002031 | 0.045063 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.023073 | 0.097727 | 0.002102 | 0.045846 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.026016 | 0.100981 | 0.002505 | 0.050054 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.023612 | 0.106149 | 0.002249 | 0.047422 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.028634 | 0.099055 | 0.002412 | 0.049107 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.030749 | 0.111999 | 0.002459 | 0.049589 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.024594 | 0.082993 | 0.002169 | 0.04657 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.030749 | 0.111999 | 0.002459 | 0.049589 |
| Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.033338 | 0.122054 | 0.002651 | 0.051491 |
| Adam | 32 | 0.005 | 50 | 0.1 | | 1 | 10 | 0.020882 | 0.081045 | 0.002047 | 0.045248 |
| Adam | 32 | 0.005 | 50 | 0.1 | 50 | 1 | 10 | 0.02124 | 0.080831 | 0.002124 | 0.046085 |
| Adam | 32 | 0.005 | 50 | 0.1 | 100 | 1 | 10 | 0.039621 | 0.178691 | 0.003032 | 0.055062 |
| Adam | 32 | 0.005 | 50 | 0.2 | | 1 | 10 | 0.02953 | 0.103907 | 0.0024 | 0.048985 |
| Adam | 32 | 0.005 | 50 | 0.1 | 50 | 1 | 10 | 0.029741 | 0.180933 | 0.002365 | 0.048627 |
| Adam | 32 | 0.005 | 50 | 0.1 | 50 | 1 | 20 | 0.025559 | 0.105548 | 0.002371 | 0.048692 |

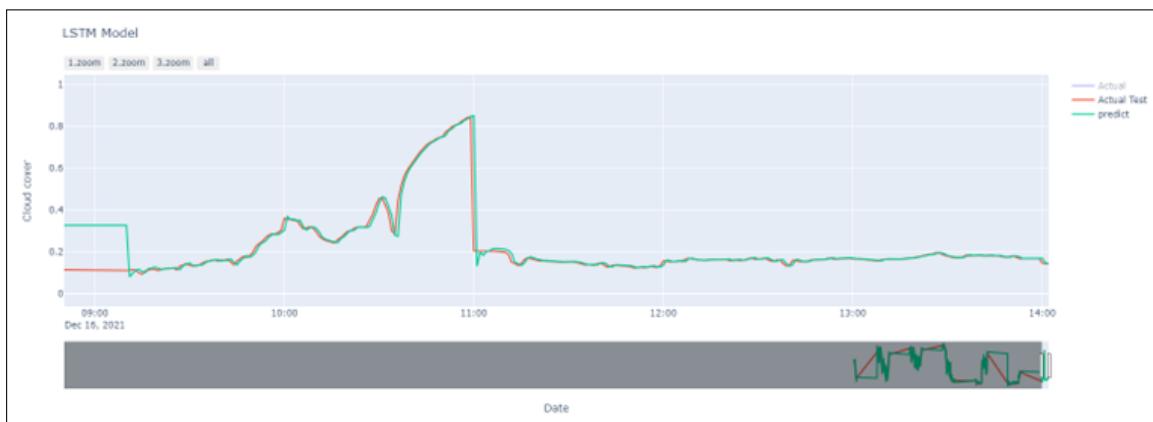
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Drop out เท่ากับ 0.1 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Drop out เท่ากับ 0.1 Second layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.27



รูปที่ 4.27 กราฟแสดงผลการทำนายเป็นนาที Lag 15 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.28 การขยายกราฟ 4.27

จากราฟ 4.28 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่โมเดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.080831

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

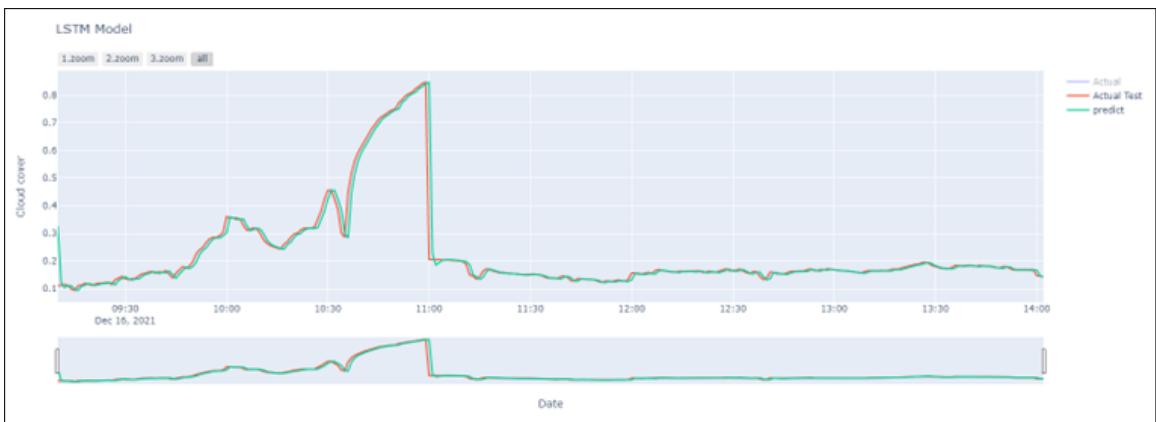
ตารางที่ 4.8 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 15 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.012933 | 0.058176 | 0.001824 | 0.042712 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.013506 | 0.06436 | 0.001847 | 0.042978 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.016758 | 0.081087 | 0.002445 | 0.049449 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.036101 | 0.195138 | 0.00281 | 0.053013 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.016697 | 0.085019 | 0.002133 | 0.046184 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.02336 | 0.128703 | 0.002301 | 0.047968 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.010777 | 0.051141 | 0.001808 | 0.042519 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.02336 | 0.128703 | 0.002301 | 0.047968 |
| Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.014637 | 0.077302 | 0.001852 | 0.043038 |
| Adam | 32 | 0.005 | 50 | 0.1 | | 1 | 10 | 0.015182 | 0.075445 | 0.001899 | 0.043575 |
| Adam | 32 | 0.005 | 50 | | 50 | 1 | 10 | 0.010689 | 0.052357 | 0.001894 | 0.043518 |
| Adam | 32 | 0.005 | 50 | | | 1 | 20 | 0.014184 | 0.076167 | 0.001907 | 0.043674 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAPE error, MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.29



รูปที่ 4.29 กราฟแสดงผลการทำนายเป็นนาที Lags 15 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.29 เสนสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลการทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้นที่ Predict น้อยกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.051141

- Lags 20 นาที
 1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

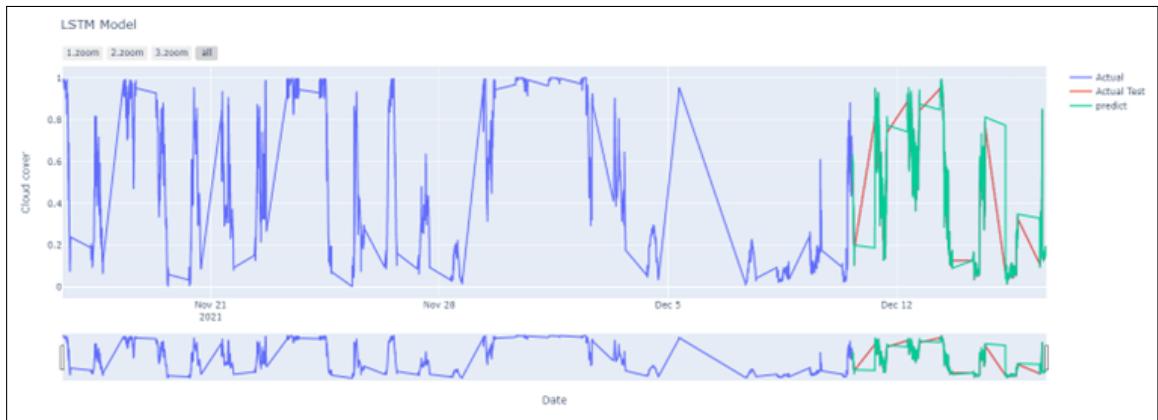
ตารางที่ 4.9 ตารางแสดง Hyperparameter ของชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% Lags 20 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.020944 | 0.07768 | 0.001977 | 0.044458 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.022332 | 0.103089 | 0.002051 | 0.045292 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.024254 | 0.090601 | 0.002132 | 0.04617 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.021808 | 0.077436 | 0.002142 | 0.046281 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.025341 | 0.107421 | 0.002387 | 0.048852 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.024354 | 0.081011 | 0.00214 | 0.046255 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.020603 | 0.075586 | 0.002082 | 0.045629 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.024354 | 0.081011 | 0.00214 | 0.046255 |
| Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.025168 | 0.129331 | 0.002161 | 0.046489 |
| Adam | 32 | 0.005 | 50 | 0.1 | | 1 | 10 | 0.022006 | 0.094634 | 0.002075 | 0.045557 |
| Adam | 32 | 0.005 | 50 | | 50 | 1 | 10 | 0.028265 | 0.174193 | 0.002307 | 0.048028 |
| Adam | 32 | 0.005 | 50 | | | 1 | 20 | 0.023093 | 0.114096 | 0.002086 | 0.045674 |

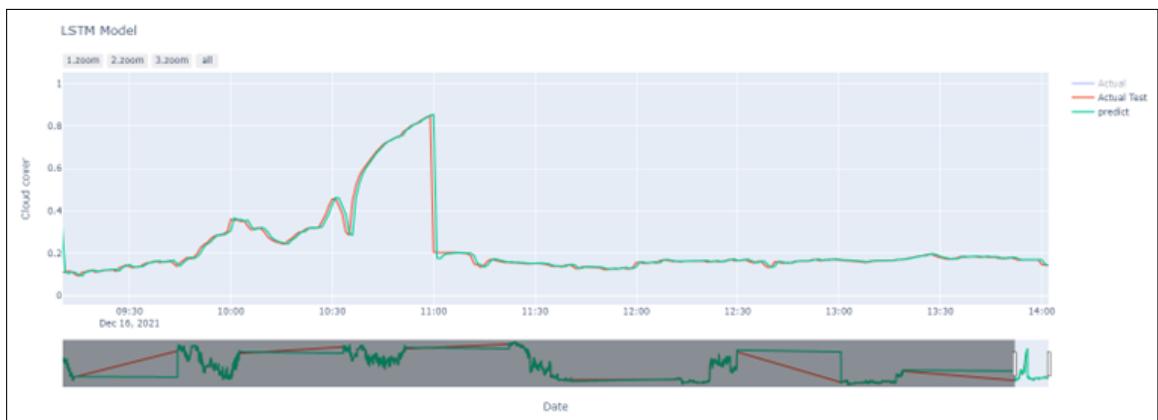
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error และ MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.30



รูปที่ 4.30 กราฟแสดงผลการทำนายเป็นนาที Lag 20 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.31 การขยายกราฟ 4.30

จากกราฟ 4.31 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้น Actual มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.075586

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

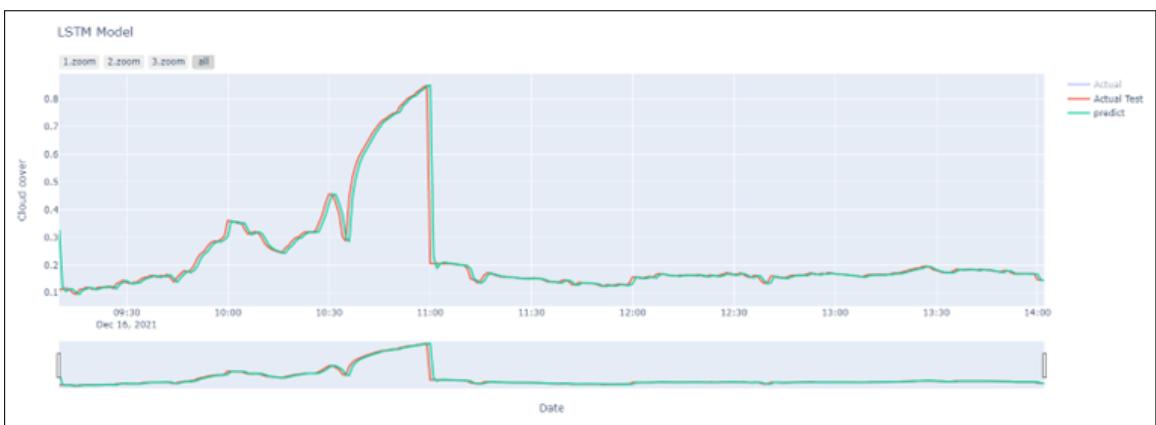
ตารางที่ 4.10 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 20 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error | |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|----------|
| Adam | 32 | | 50 | | | | 1 | 10 | 0.012457 | 0.057628 | 0.001861 | 0.043136 |
| RMSprop | 32 | | 50 | | | | 1 | 10 | 0.013382 | 0.068889 | 0.001943 | 0.044082 |
| Adam | 64 | | 50 | | | | 1 | 10 | 0.012772 | 0.061331 | 0.002049 | 0.045265 |
| Adam | 32 | 0.1 | 50 | | | | 1 | 10 | 0.01227 | 0.05878 | 0.001795 | 0.042368 |
| Adam | 32 | 0.05 | 50 | | | | 1 | 10 | 0.120803 | 0.669086 | 0.016434 | 0.128196 |
| Adam | 32 | 0.01 | 50 | | | | 1 | 10 | 0.014821 | 0.078361 | 0.001971 | 0.044394 |
| Adam | 32 | 0.005 | 50 | | | | 1 | 10 | 0.01074 | 0.051085 | 0.001817 | 0.042621 |
| Adam | 32 | 0.001 | 50 | | | | 1 | 10 | 0.014821 | 0.078361 | 0.001971 | 0.044394 |
| Adam | 32 | 0.005 | 100 | | | | 1 | 10 | 0.01698 | 0.092862 | 0.001951 | 0.044173 |
| Adam | 32 | 0.005 | 50 | 0.1 | | | 1 | 10 | 0.021325 | 0.120427 | 0.002108 | 0.045917 |
| Adam | 32 | 0.005 | 50 | | 50 | | 1 | 10 | 0.010519 | 0.051351 | 0.001876 | 0.043318 |
| Adam | 32 | 0.005 | 50 | | | | 1 | 20 | 0.012345 | 0.061195 | 0.001862 | 0.043152 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Second layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.1 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.32



รูปที่ 4.32 กราฟแสดงผลการทำนายเป็นนาที Laggs 20 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.32 เสน่สีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้นที่ Predict น้อยกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.051085

- Lags 25 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

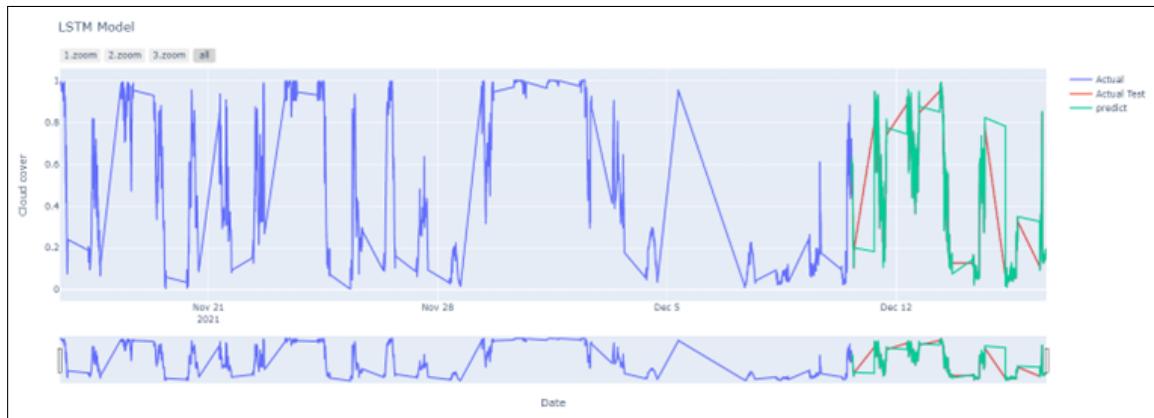
ตารางที่ 4.11 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยTrain set เป็น 80% และ Test set เป็น 20% Lags 25 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.021512 | 0.085131 | 0.002007 | 0.044801 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.023064 | 0.08598 | 0.002084 | 0.045648 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.024111 | 0.08767 | 0.002244 | 0.047368 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.020791 | 0.078015 | 0.002096 | 0.045784 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.02733 | 0.105585 | 0.002285 | 0.047799 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.022327 | 0.078284 | 0.002064 | 0.045436 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.021713 | 0.083822 | 0.002074 | 0.04554 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.022327 | 0.078284 | 0.002064 | 0.045436 |
| Adam | 32 | 0.1 | 100 | | | 1 | 10 | 0.027932 | 0.109917 | 0.002377 | 0.048754 |
| Adam | 32 | 0.1 | 50 | 0.1 | | 1 | 10 | 0.031217 | 0.186418 | 0.00247 | 0.049701 |
| Adam | 32 | 0.1 | 50 | | 50 | 1 | 10 | 0.028495 | 0.125518 | 0.002446 | 0.049452 |
| Adam | 32 | 0.1 | 50 | | | 1 | 20 | 0.021766 | 0.083192 | 0.002136 | 0.046214 |

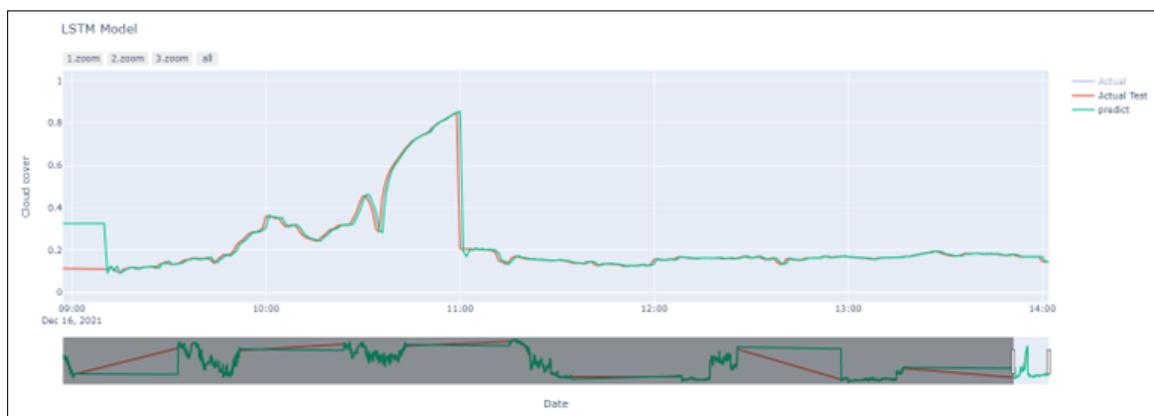
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.1 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error และ MAPE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.33



รูปที่ 4.33 กราฟแสดงผลการทำนายเป็นนาที Lag 25 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.34 การขยายกราฟ 4.33

จากราฟ 4.34 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.078015

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

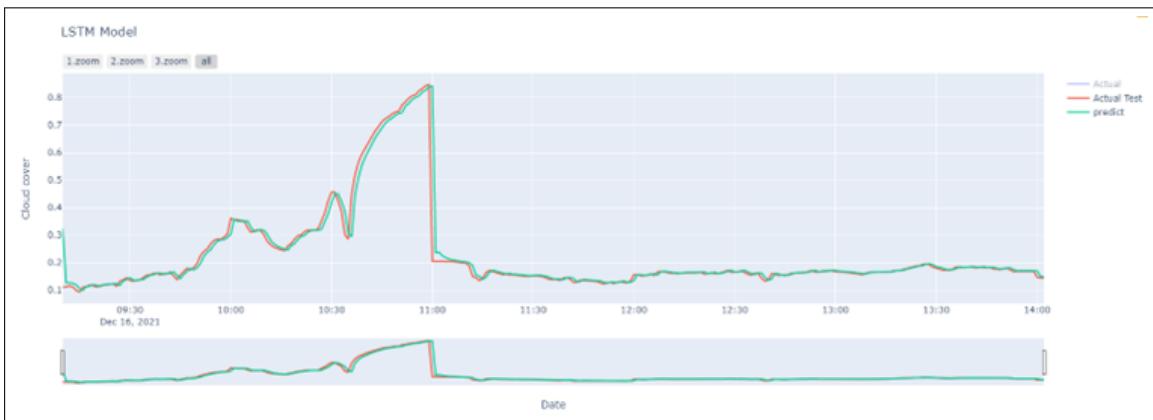
ตารางที่ 4.12 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 25 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error | |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|----------|
| Adam | 32 | | 50 | | | | 1 | 10 | 0.012478 | 0.056204 | 0.001835 | 0.042836 |
| RMSprop | 32 | | 50 | | | | 1 | 10 | 0.012514 | 0.056939 | 0.001816 | 0.042613 |
| Adam | 64 | | 50 | | | | 1 | 10 | 0.013487 | 0.060221 | 0.00193 | 0.043927 |
| Adam | 32 | 0.1 | 50 | | | | 1 | 10 | 0.020206 | 0.113982 | 0.002136 | 0.046219 |
| Adam | 32 | 0.05 | 50 | | | | 1 | 10 | 0.020668 | 0.114666 | 0.00219 | 0.046794 |
| Adam | 32 | 0.01 | 50 | | | | 1 | 10 | 0.012911 | 0.062051 | 0.001843 | 0.042933 |
| Adam | 32 | 0.005 | 50 | | | | 1 | 10 | 0.013141 | 0.064064 | 0.001844 | 0.042937 |
| Adam | 32 | 0.001 | 50 | | | | 1 | 10 | 0.012911 | 0.062051 | 0.001843 | 0.042933 |
| Adam | 32 | | 100 | | | | 1 | 10 | 0.011594 | 0.055983 | 0.001813 | 0.042577 |
| Adam | 32 | | 100 | 0.1 | | | 1 | 10 | 0.013066 | 0.063178 | 0.001825 | 0.04272 |
| Adam | 32 | | 100 | | 50 | | 1 | 10 | 0.017154 | 0.083024 | 0.001912 | 0.043728 |
| Adam | 32 | | 100 | | | | 1 | 20 | 0.011638 | 0.054141 | 0.001805 | 0.042484 |
| Adam | 32 | | 100 | | | | 1 | 30 | 0.012958 | 0.059748 | 0.001834 | 0.042821 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error น้อยที่สุด
- Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ทำให้ค่า MAPE error MSE error และ RMSE error น้อยที่สุด

กราฟแสดงผลได้รูปที่ 4.35



รูปที่ 4.35 กราฟแสดงผลการทำนายเป็นนาที Lags 25 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.35 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลการทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict มากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.055983

- Lags 30 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ได้ผลดังนี้

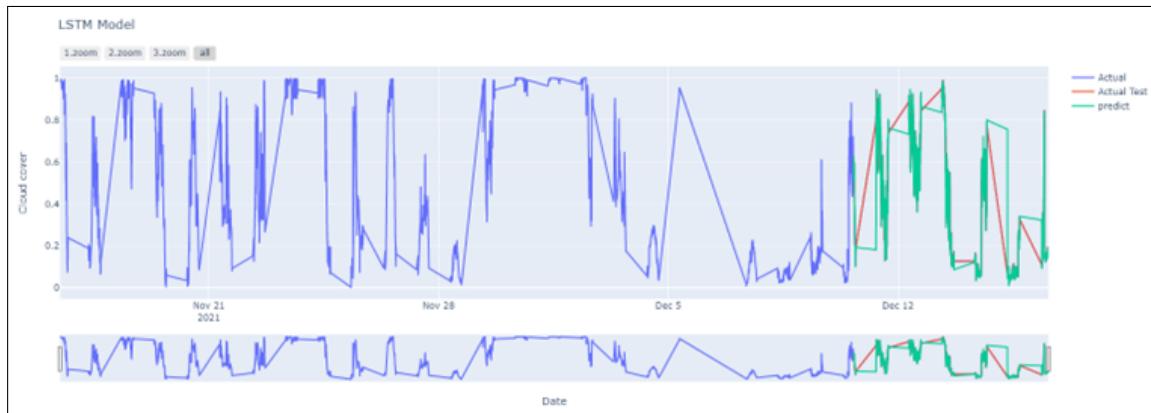
ตารางที่ 4.13 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยTrain set เป็น 80% และ Test set เป็น 20% Lags 30 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|-------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.026799 | 0.09603 | 0.002205 | 0.046956 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.026451 | 0.112134 | 0.002275 | 0.047702 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.025145 | 0.106561 | 0.002341 | 0.048383 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.0353 | 0.180074 | 0.003722 | 0.061011 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.055625 | 0.32054 | 0.00459 | 0.067749 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.022734 | 0.104956 | 0.002059 | 0.045379 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.021521 | 0.081069 | 0.002045 | 0.045226 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.022734 | 0.104956 | 0.002059 | 0.045379 |
| Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.020973 | 0.077932 | 0.002028 | 0.045035 |
| Adam | 32 | 0.005 | 100 | 0.1 | | 1 | 10 | 0.022035 | 0.089694 | 0.002071 | 0.045506 |
| Adam | 32 | 0.005 | 100 | | 50 | 1 | 10 | 0.025626 | 0.141868 | 0.002191 | 0.046804 |
| Adam | 32 | 0.005 | 50 | | | 1 | 20 | 0.022289 | 0.092985 | 0.002119 | 0.046036 |

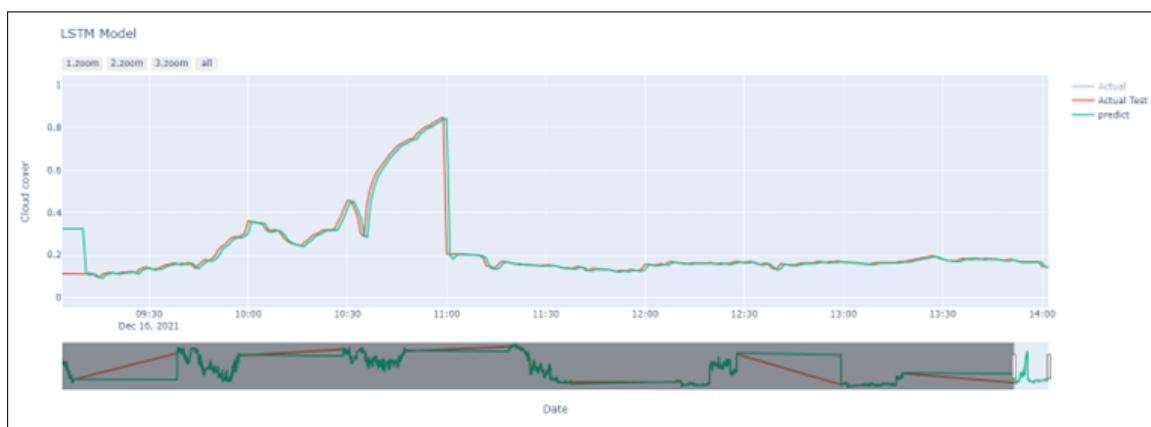
จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ทำให้ค่า MAE error, MAPE error, MSE error และ RMSE error น้อยที่สุด และกราฟแสดงผลได้ดังนี้

กราฟแสดงผลได้รูปที่ 4.36



รูปที่ 4.36 กราฟแสดงผลการทำนายเป็นนาที Lag 30 นาทีของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.37 การขยายกราฟ 4.36

จากราฟ 4.37 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict น้อยกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.081069

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ได้ผลดังนี้

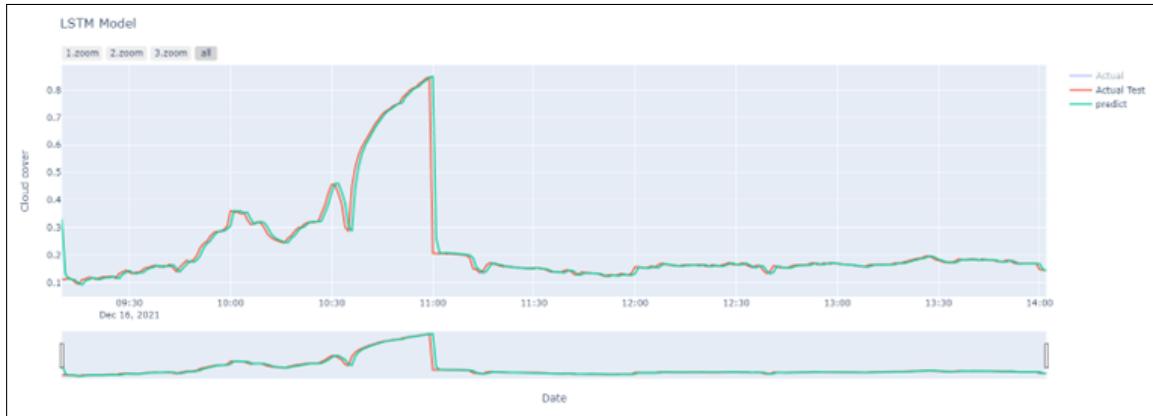
ตารางที่ 4.14 ตารางแสดง Hyperparameter ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set Lags 30 นาที

| Optimizer | Batch_size | Learning rate | Frist layer | Drop out | Second layer | Output layer | Epochs | MAE error | MAPE error | MSE error | RMSE error |
|-----------|------------|---------------|-------------|----------|--------------|--------------|--------|-----------|------------|-----------|------------|
| Adam | 32 | | 50 | | | 1 | 10 | 0.012892 | 0.059515 | 0.001843 | 0.042926 |
| RMSprop | 32 | | 50 | | | 1 | 10 | 0.017057 | 0.077014 | 0.001873 | 0.043275 |
| Adam | 64 | | 50 | | | 1 | 10 | 0.013222 | 0.06426 | 0.002055 | 0.045337 |
| Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.018787 | 0.083944 | 0.002249 | 0.047424 |
| Adam | 32 | 0.05 | 50 | | | 1 | 10 | 0.010813 | 0.053302 | 0.001868 | 0.043221 |
| Adam | 32 | 0.01 | 50 | | | 1 | 10 | 0.020407 | 0.112591 | 0.002015 | 0.044892 |
| Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.011762 | 0.05992 | 0.001841 | 0.042906 |
| Adam | 32 | 0.001 | 50 | | | 1 | 10 | 0.020407 | 0.112591 | 0.002015 | 0.044892 |
| Adam | 32 | 0.05 | 100 | | | 1 | 10 | 0.293096 | 1.726908 | 0.090901 | 0.301498 |
| Adam | 32 | 0.05 | 50 | 0.1 | | 1 | 10 | 0.03017 | 0.171165 | 0.00301 | 0.054864 |
| Adam | 32 | 0.05 | 50 | | 50 | 1 | 10 | 0.014864 | 0.070726 | 0.001857 | 0.043098 |
| Adam | 32 | 0.05 | 50 | | | 1 | 20 | 0.010618 | 0.051346 | 0.001831 | 0.042792 |
| Adam | 32 | 0.05 | 50 | | | 1 | 30 | 0.01831 | 0.088785 | 0.001895 | 0.043537 |

จากการปรับ hyperparameter พบว่า

- Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.05 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ทำให้ค่า MAE error, MAPE error, MSE error และ RMSE error น้อยที่สุด และกราฟแสดงผลได้ดังนี้

กราฟแสดงผลได้รูปที่ 4.38

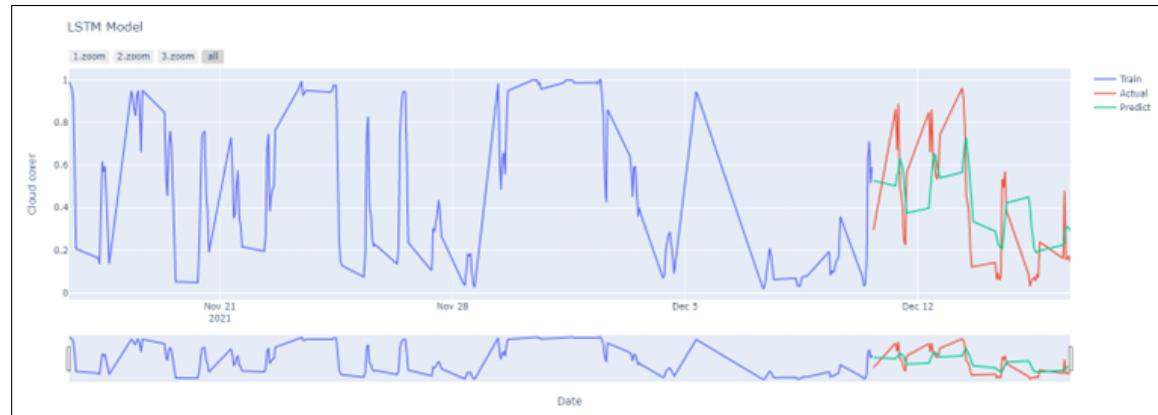


รูปที่ 4.38 กราฟแสดงผลการทำนายเป็นนาที Lagts 30 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

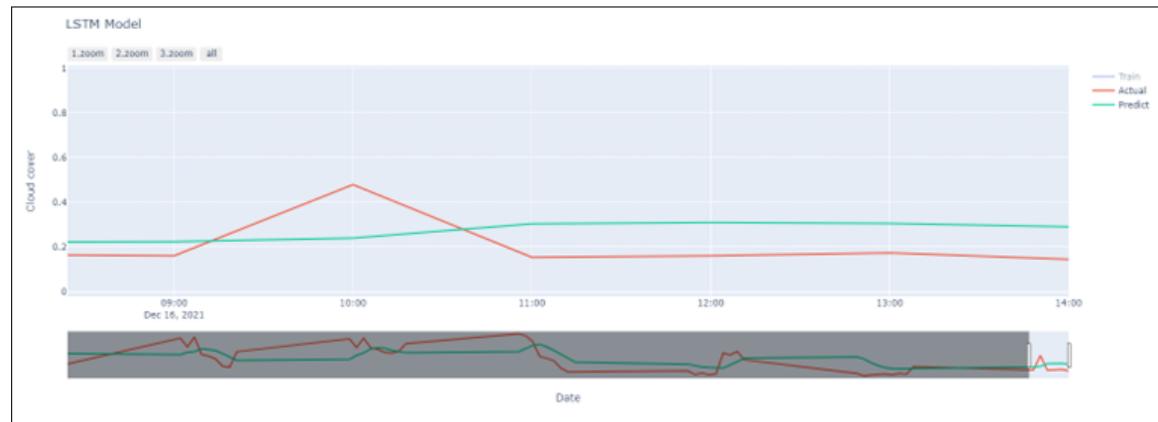
จากกราฟ 4.38 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้นที่ Predict มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.051346

4.2.2.2 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นช่วงโหมด

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



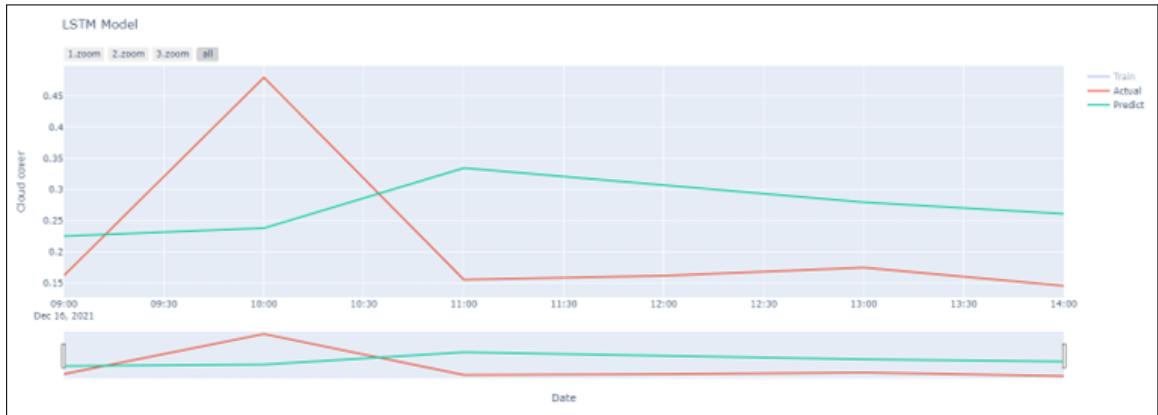
รูปที่ 4.39 กราฟแสดงผลการทำนายเป็นช่วงโหมดของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.40 การขยายกราฟ 4.39

จากราฟ 4.40 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่делทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 1.20508

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้

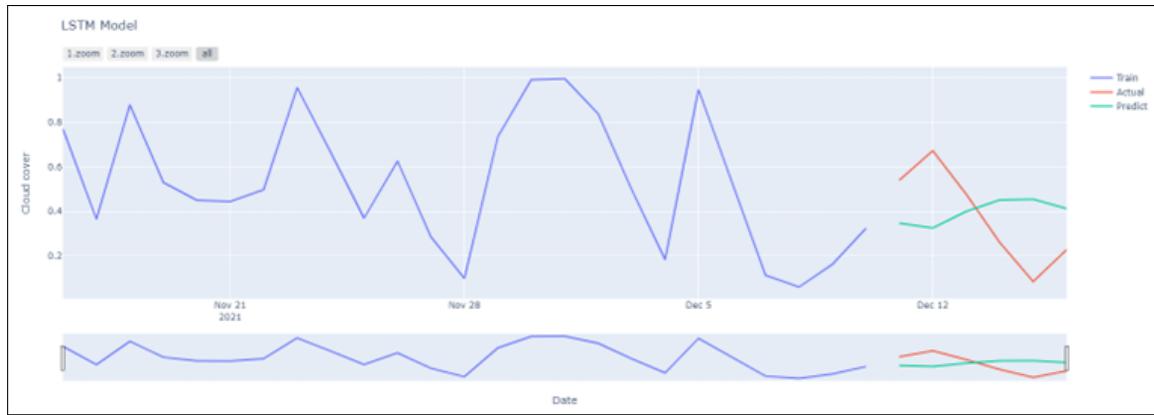


รูปที่ 4.41 กราฟแสดงผลการทำนายเป็นขั้วมองของ LSTM ของชุดข้อมูลที่เลือกวันสุดท้ายเป็น test set

จากราฟ 4.41 เสนอสีแดงคือข้อมูลที่ใช้ test model และเสนอสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict ไม่มีความใกล้เคียงกับเส้น Actual เนื่องจากเส้น Predict มากกว่าเส้น Actual มีค่า MAPE อยู่ที่ 0.72599

4.2.2.3 การแบ่งชุดข้อมูลตัวแปรเดียวโดยใช้ตัวแปร Cloud Cover ที่เป็นรายวัน

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



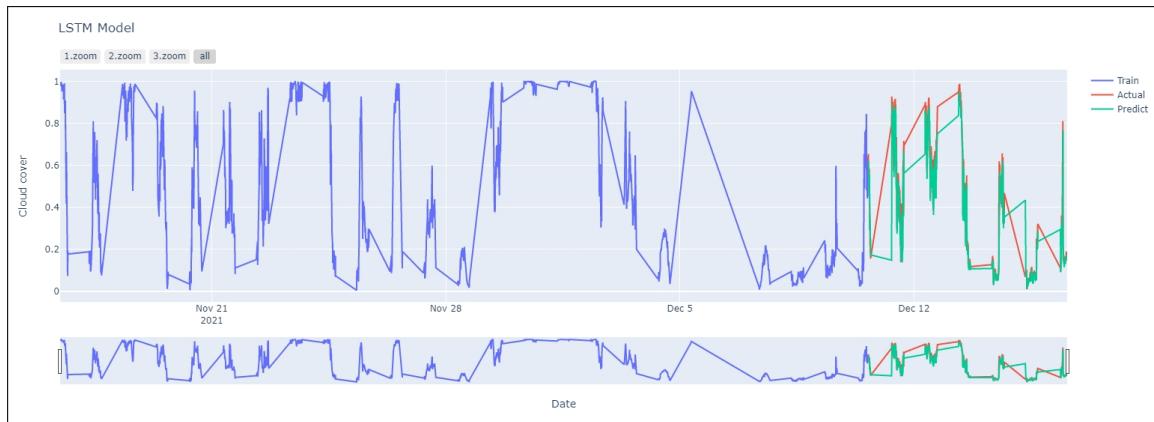
รูปที่ 4.42 กราฟแสดงผลการทำนายเป็นรายวันของ LSTM ของชุดข้อมูล Train set เป็น 80% และ Test set เป็น 20%

จากการ [4.42](#) เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่делทำนาย (Predict) พบร้าเส้นที่ Predict ไม่มีความใกล้เคียงกับเส้น Actual เนื่องจากเส้น Predict น้อยกว่าและมากกว่าเส้น Actual มีค่า MAPE 1.36781

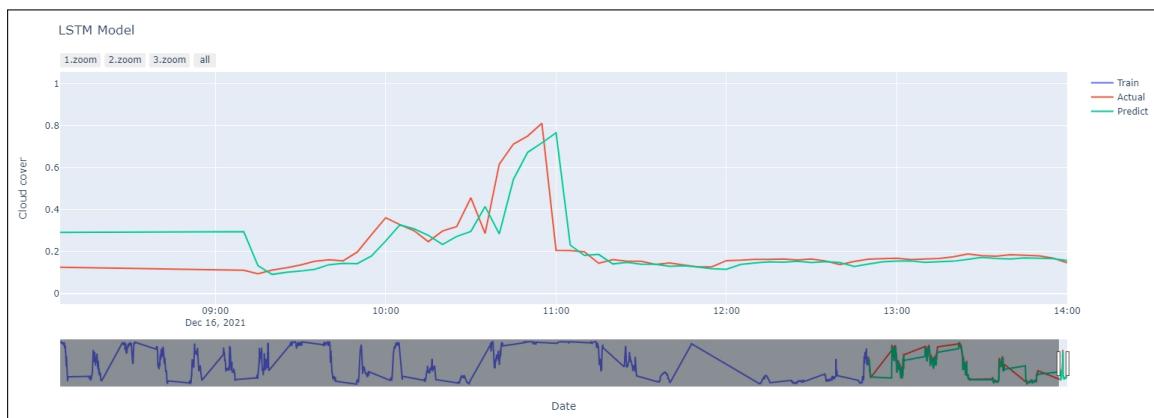
4.2.2.4 การแบ่งชุดข้อมูลหลายตัวแปรโดยใช้ตัวแปร Cloud Cover, Relative humidity, Temperature และ Wind speed ที่เป็นนาที

- Lags 5 นาที

1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



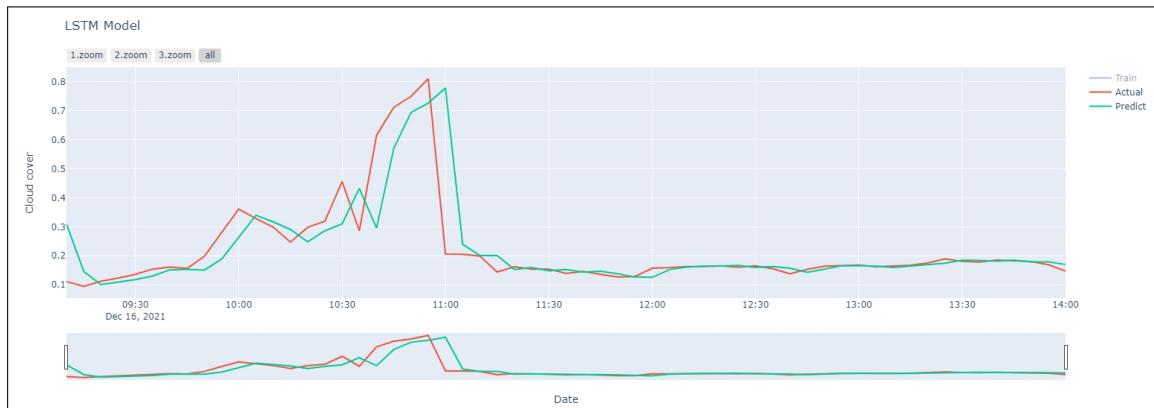
รูปที่ 4.43 กราฟแสดงผลการทำนายเป็นนาที Lags 5 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.44 การขยายกราฟ 4.43

จากการ 4.44 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เคลื่อนทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ 0.18229

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้

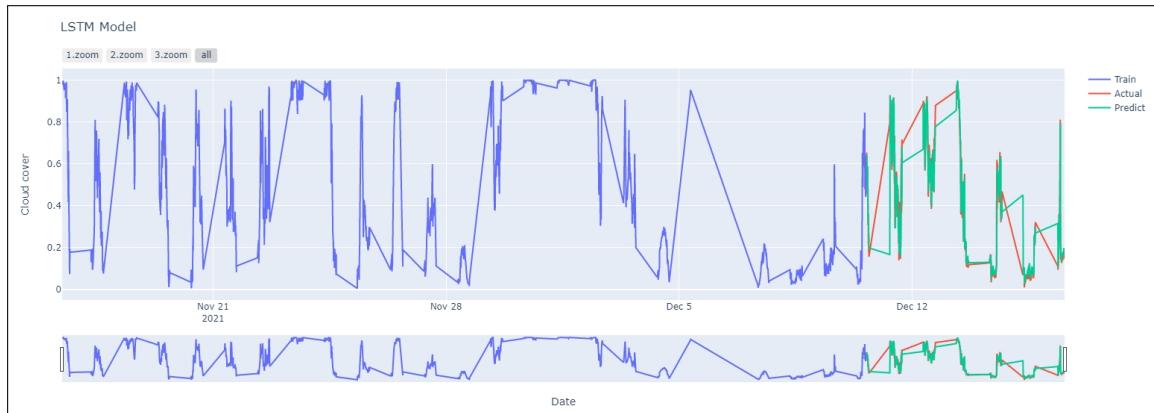


รูปที่ 4.45 กราฟแสดงผลการทำนายเป็นนาที Lags 5 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

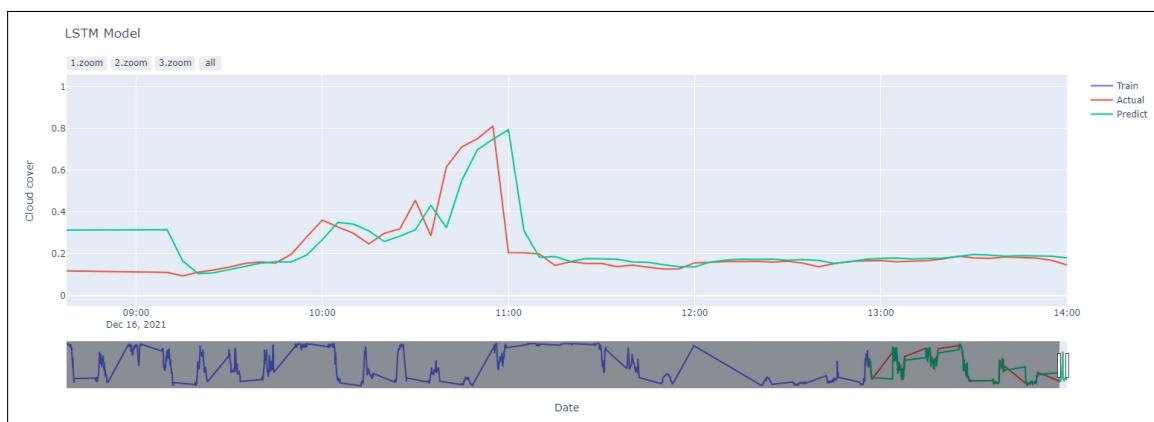
จากราฟ 4.45 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual นีค่า MAPE อยู่ที่ 0.18602

- Lags 10 นาที

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



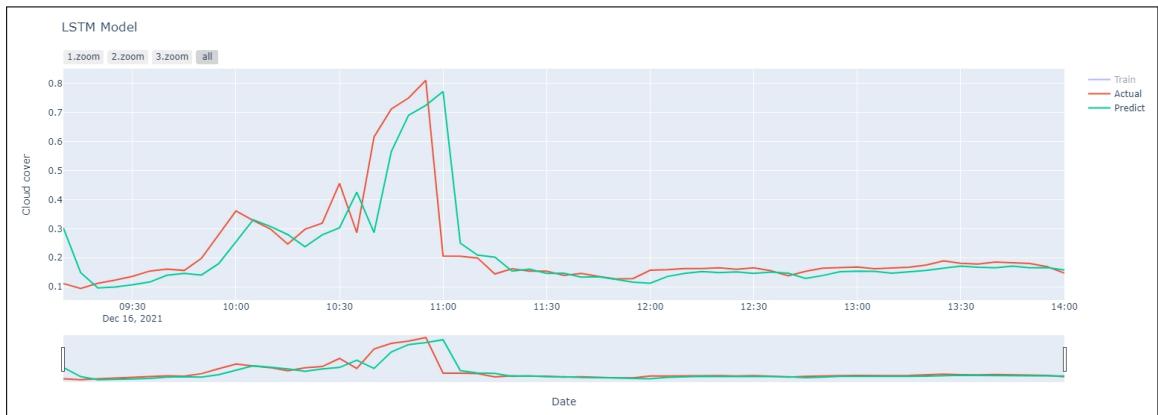
รูปที่ 4.46 การแสดงผลการทำนายเป็นนาที Lags 10 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.47 การขยายกราฟ 4.46

จากการ [รูปที่ 4.47](#) เน้นสื้น้ำเงินคือข้อมูลที่ใช้ train model และสีแดงคือข้อมูลที่ใช้ test model และสีเขียวคือสีที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าseenที่ Predict น้อยกว่าและมากกว่าseen Actual ทำให้seen Predict ไม่มีความใกล้เคียงกับseen Actual มีค่า MAPE อยู่ที่ 0.20244

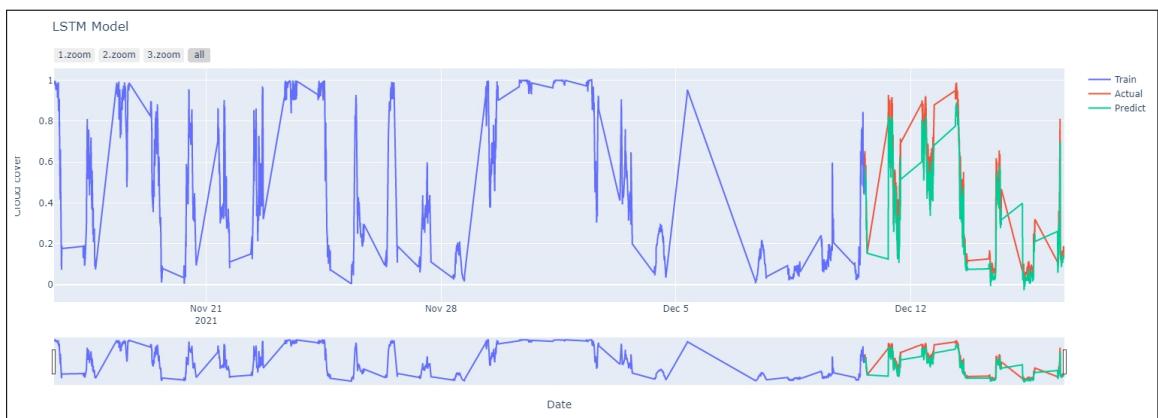
2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้



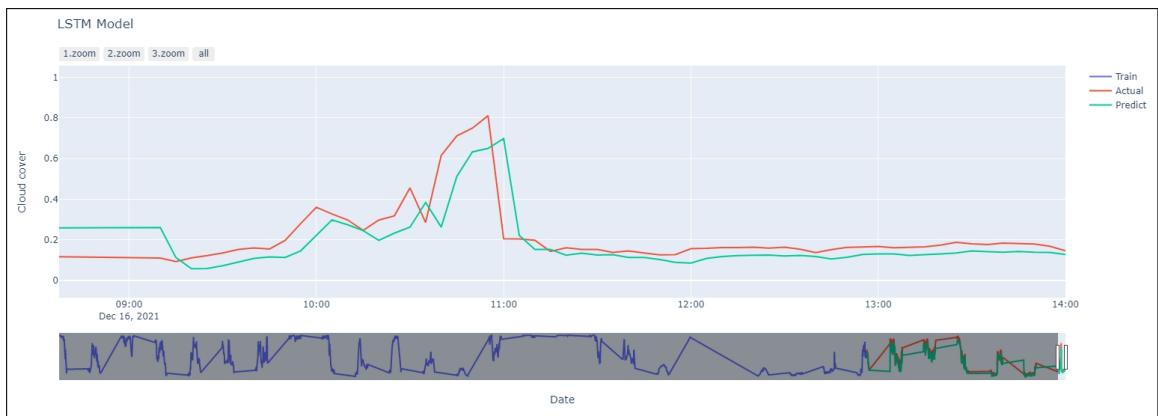
รูปที่ 4.48 กราฟแสดงผลการทำนายเป็นนาที Lags 10 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากกราฟ 4.48 เสนนสีแดงคือข้อมูลที่ใช้ test model และเสนสีเขียวคือเสนที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict มากกว่าเสน Actual ทำให้เสน Predict ไม่มีความใกล้เคียงกับเสน Actual มีค่า MAPE อยู่ที่ 0.21580

- Lags 15 นาที
 1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Drop out เท่ากับ 0.1 Second layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้



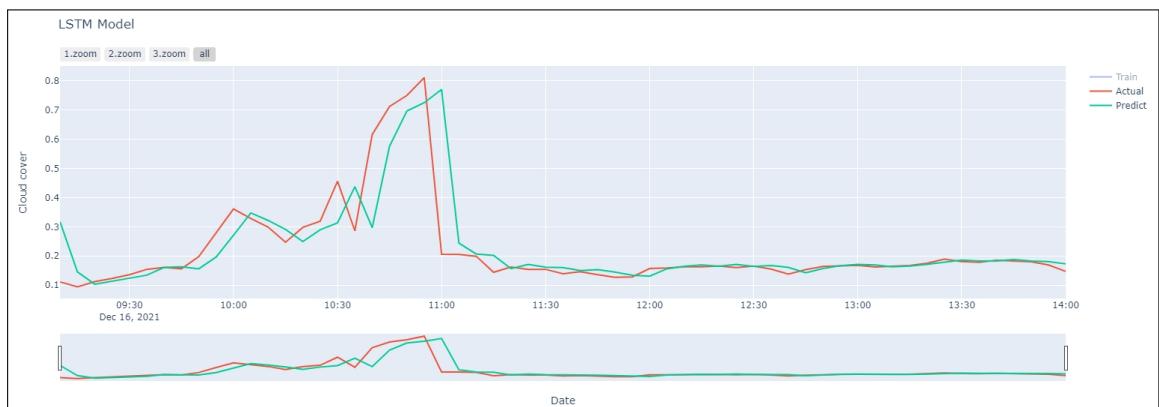
รูปที่ 4.49 กราฟแสดงผลการทำนายเป็นนาที Lags 15 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.50 การขยายกราฟ 4.49

จากกราฟ 4.50 เส้นสีเขียวคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกับเส้นที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.30476

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้

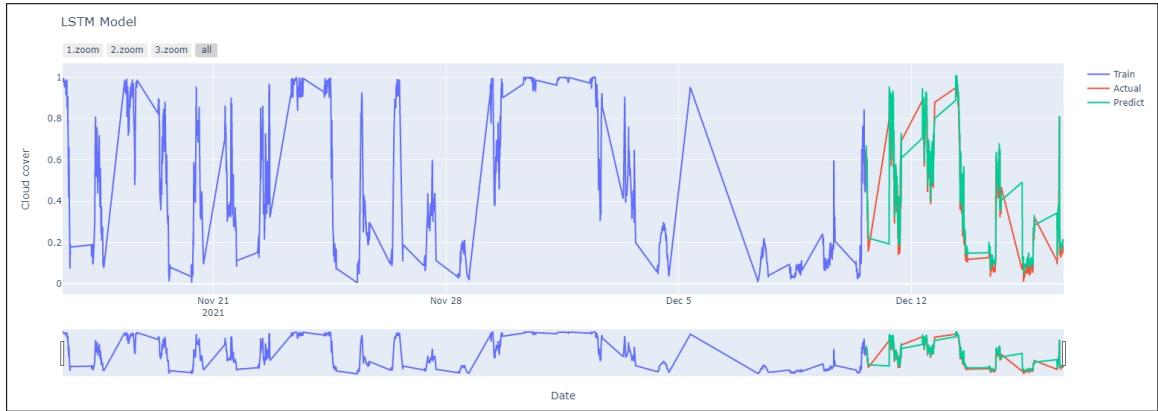


รูปที่ 4.51 กราฟแสดงผลการทำนายเป็นนาที Lagts 15 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

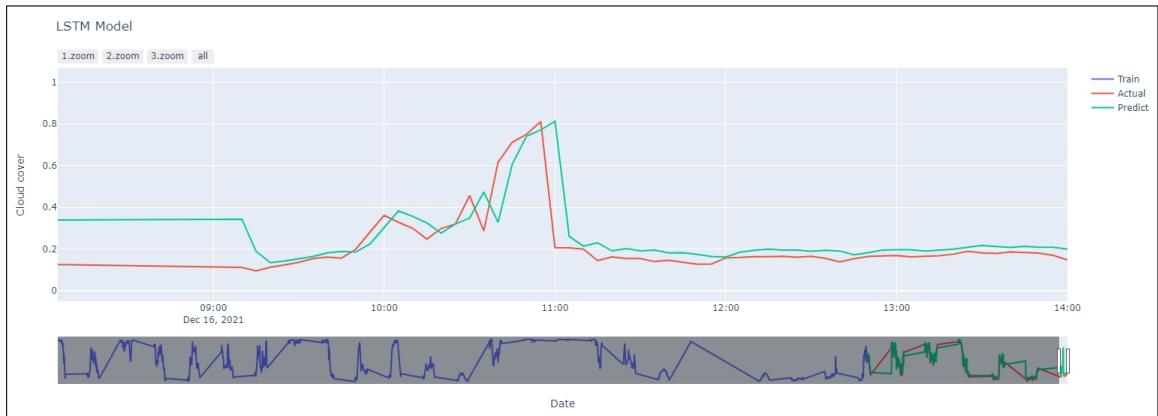
จากการ [ภาพ 4.51](#) เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบว่าเส้นที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.18848

- Lags 20 นาที

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้



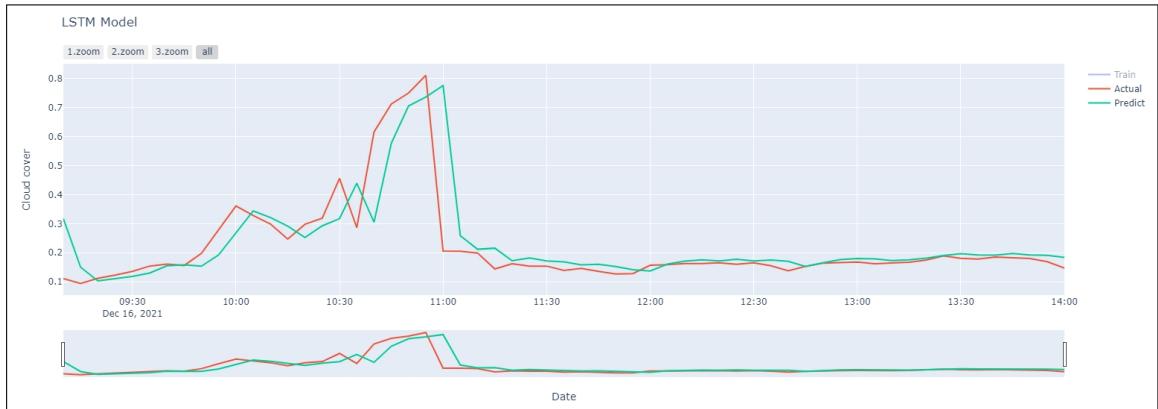
รูปที่ 4.52 กราฟแสดงผลการทำนายเป็นนาที Lags 20 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.53 การขยายกราฟ 4.52

จากราฟ 4.53 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.30851

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้

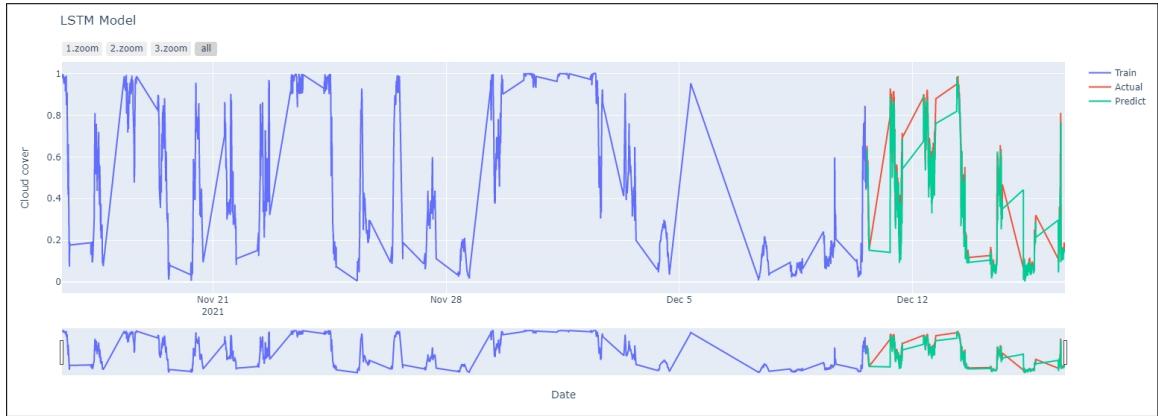


รูปที่ 4.54 กราฟแสดงผลการทำนายเป็นนาที Laggs 20 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

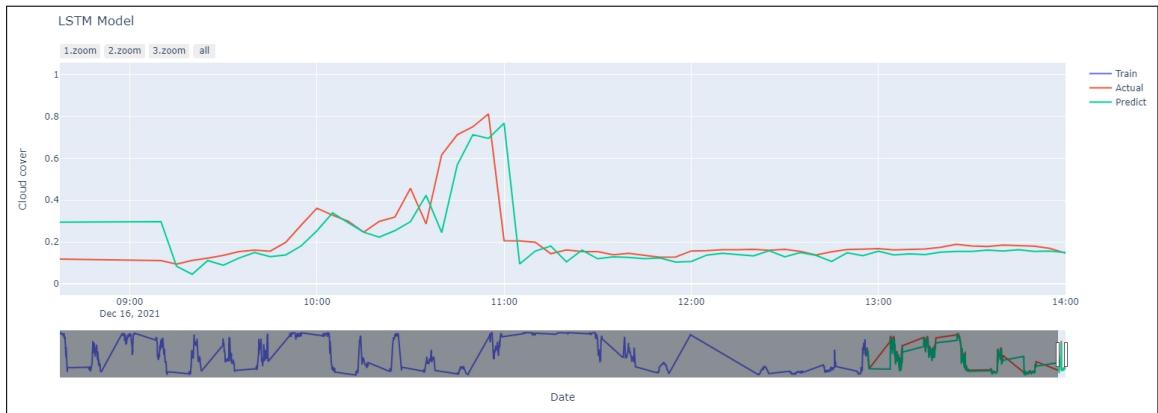
จากราฟ 4.54 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่โมเดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่เรียบเกล้าเดียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.21496

- Lags 25 นาที

- การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.1 Frist layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้



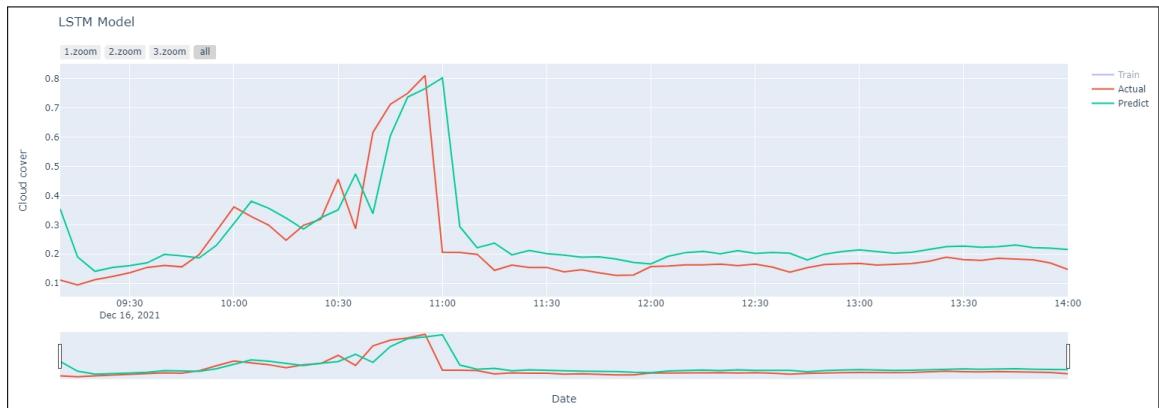
รูปที่ 4.55 กราฟแสดงผลการทำนายเป็นนาที Lags 25 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.56 การขยายกราฟ 4.55

จากราฟ 4.56 เส้นสีน้ำเงินคือข้อมูลที่ใช้ train model และเส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict น้อยกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.20747

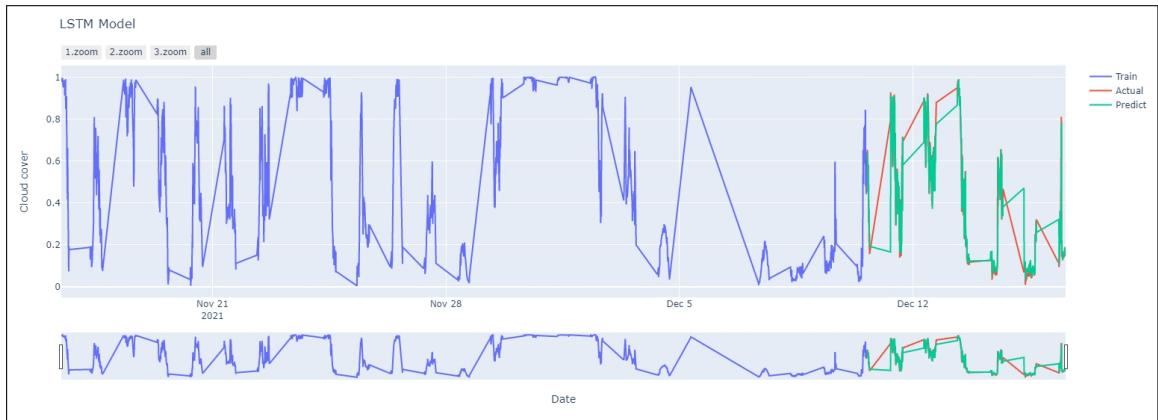
2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



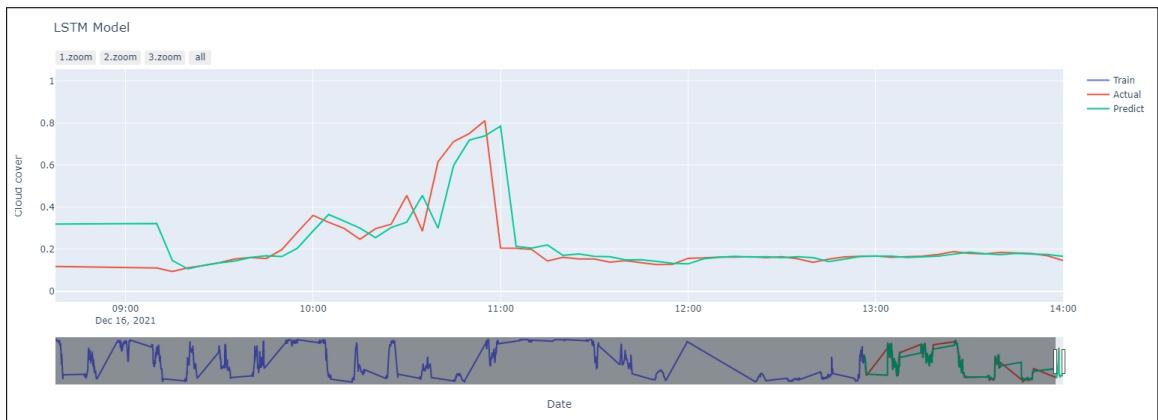
รูปที่ 4.57 กราฟแสดงผลการทำนายเป็นนาที Lags 25 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.57 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร่วมกันที่ Predict มากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.34690

- Lags 30 นาที
 1. การแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.005 Frist layer เท่ากับ 100 Output layer เท่ากับ 1 Epochs เท่ากับ 10 ได้ผลดังนี้



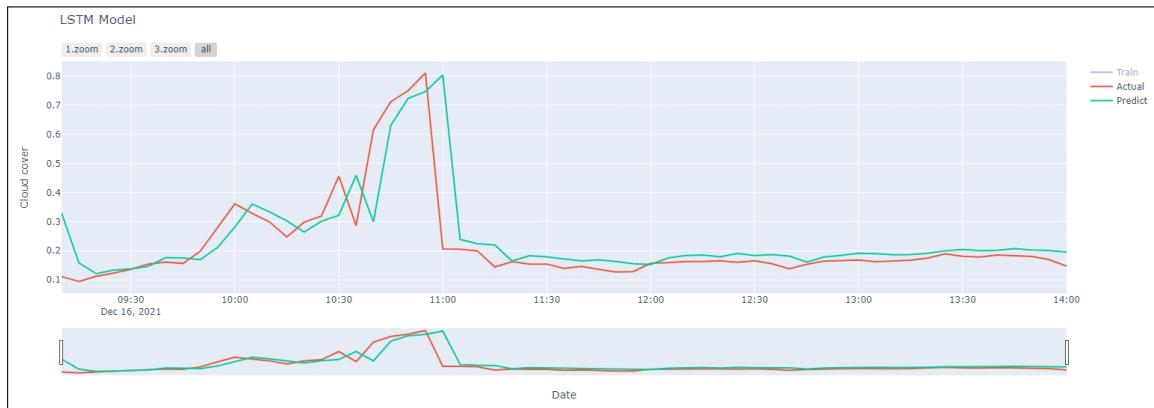
รูปที่ 4.58 กราฟแสดงผลการทำนายเป็นนาที Lags 30 นาทีของ LSTM ของชุดข้อมูลหลายตัวแปร Train set เป็น 80% และ Test set เป็น 20%



รูปที่ 4.59 การขยายกราฟ 4.58

จากราฟ 4.59 เสนอสีน้ำเงินคือข้อมูลที่ใช้ train model และเสนอสีแดงคือข้อมูลที่ใช้ test model และเสนอสีเขียวคือเสนอที่ไม่เดลทำนาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าseenที่ Predict น้อยกว่าและมากกว่าseen Actual ทำให้seen Predict ไม่มีความใกล้เคียงกับseen Actual มีค่า MAPE อยู่ที่ 0.21125

2. การแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ใช้ Optimizer เป็น Adam Batch_size เท่ากับ 32 Learning rate เท่ากับ 0.05 First layer เท่ากับ 50 Output layer เท่ากับ 1 Epochs เท่ากับ 20 ได้ผลดังนี้



รูปที่ 4.60 กราฟแสดงผลการทํานายเป็นนาที Lags 30 นาทีของ LSTM ของชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set

จากราฟ 4.60 เส้นสีแดงคือข้อมูลที่ใช้ test model และเส้นสีเขียวคือเส้นที่ไม่เดลทํานาย (Predict) โดยพิจารณาช่วงวันที่ 16-12-2021 พบร้าเส้นที่ Predict น้อยกว่าและมากกว่าเส้น Actual ทำให้เส้น Predict ไม่มีความใกล้เคียงกับเส้น Actual มีค่า MAPE อยู่ที่ 0.24564

4.3 สรุปผลประสิทธิภาพของโมเดลพยากรณ์ปริมาณเมฆ

ตารางที่ 4.15 ตารางแสดงโมเดล LSTM ที่มีประสิทธิภาพดีที่สุดของการแบ่งชุดข้อมูลโดย Train set เป็น 80% และ Test set เป็น 20% ในแต่ละ lags

| Lags | Optimizer | Batch size | Learning rate | First layer | Dropout | Second layer | Output layer | Epochs | MAPE |
|------|-----------|------------|---------------|-------------|---------|--------------|--------------|--------|----------|
| 5 | Adam | 32 | | 100 | | | 1 | 20 | 0.074577 |
| 10 | Adam | 32 | | 50 | | | 1 | 20 | 0.074959 |
| 15 | Adam | 32 | 0.005 | 50 | 0.1 | 50.0 | 1 | 10 | 0.080831 |
| 20 | Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.075586 |
| 25 | Adam | 32 | 0.1 | 50 | | | 1 | 10 | 0.078015 |
| 30 | Adam | 32 | 0.005 | 100 | | | 1 | 10 | 0.077932 |

จากตารางที่ 4.15 เมื่อพิจารณาจากค่า MAPE พบร้าโมเดล LSTM Lags 5 มีค่า MAPE น้อยที่สุดอยู่ที่ 0.074577 รองลงมาค่า Lags 10, 20, 30, 25, 15 ตามลำดับ

ตารางที่ 4.16 ตารางแสดงโมเดล LSTM ที่มีประสิทธิภาพดีที่สุดของการแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น Test set ในแต่ละ lags

| Lags | Optimizer | Batch size | Learning rate | First layer | Dropout | Second layer | Output layer | Epochs | MAPE |
|------|-----------|------------|---------------|-------------|---------|--------------|--------------|--------|----------|
| 5 | Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.051084 |
| 10 | Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.053332 |
| 15 | Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.051141 |
| 20 | Adam | 32 | 0.005 | 50 | | | 1 | 10 | 0.051085 |
| 25 | Adam | 32 | | 100 | | | 1 | 20 | 0.054141 |
| 30 | Adam | 32 | 0.05 | 50 | | | 1 | 20 | 0.051346 |

จากตารางที่ 4.16 เมื่อพิจารณาจากค่า MAPE พบร่วมกับ LSTM Lags 5 มีค่า MAPE น้อยที่สุดอยู่ที่ 0.051084 รองลงมาค่า Lags 20, 15, 30, 10, 25 ตามลำดับ

ในการทดลองของ LSTM มีการ lags ข้อมูล 5, 10, 15, 20, 25 และ 30 นาที จึงได้ทำการเลือก Model ที่ดีที่สุดในแต่ละ lags มาหาค่าเฉลี่ยเพื่อเปรียบเทียบ Prophet และ LSTM

ตารางที่ 4.17 ตารางประเมินประสิทธิภาพของโมเดลพยากรณ์ปริมาณแม่น

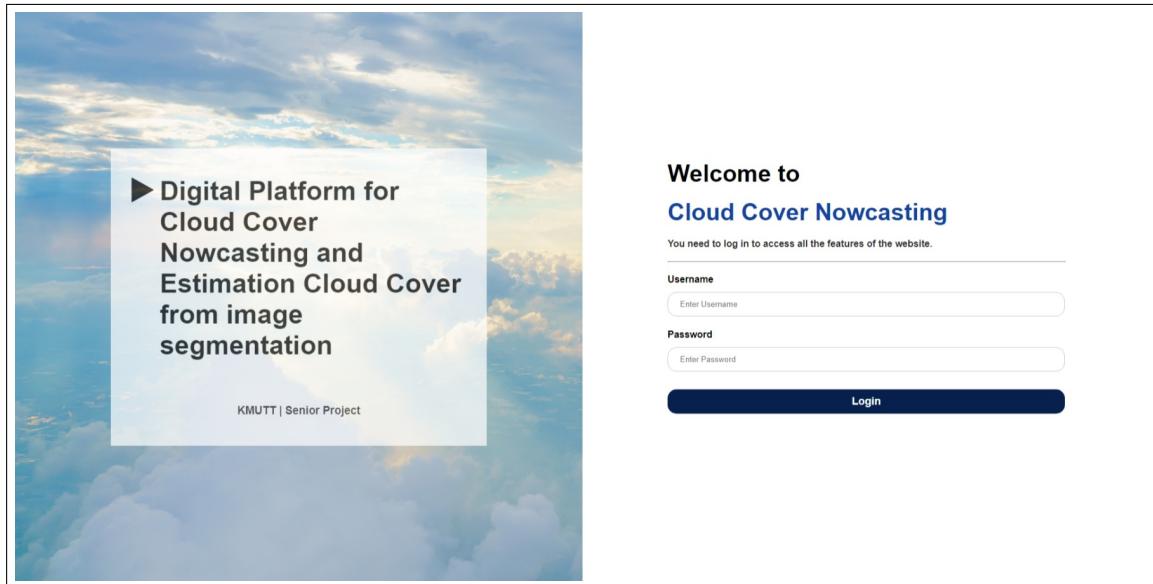
| Model | รูปแบบของชุดข้อมูล | การแบ่งชุดข้อมูล | Feature | Evaluation metric | | | |
|---------|-------------------------|------------------------------|------------------|-------------------|---------|---------|---------|
| | | | | MAE | MAPE | MSE | RMSE |
| Prophet | นาที | train set 80 % test set 20 % | CC | 0.52741 | 1.79844 | 0.39959 | 0.63213 |
| | | | CC, Temp, RH, WS | 0.68843 | 2.72634 | 0.61059 | 0.78140 |
| | วันสุดท้ายเป็น test set | CC | 0.10543 | 0.35754 | 0.03359 | 0.18327 | |
| | | CC, Temp, RH, WS | 0.16830 | 0.97149 | 0.03550 | 0.18841 | |
| | ชั่วโมง | train set 80 % test set 20% | CC | 0.33522 | 1.29596 | 0.18310 | 0.42790 |
| | | วันสุดท้ายเป็น test set | CC | 0.19039 | 1.10121 | 0.03781 | 0.19444 |
| | รายวัน | train set 80 % test set 20% | CC | 0.23312 | 0.90479 | 0.06782 | 0.26042 |
| LSTM | นาที | train set 80 % test set 20 % | CC | 0.02068 | 0.07698 | 0.00201 | 0.04482 |
| | | | CC, Temp, RH, WS | 0.05547 | 0.23612 | 0.00756 | 0.08675 |
| | วันสุดท้ายเป็น test set | CC | 0.01094 | 0.05202 | 0.00181 | 0.04253 | |
| | | CC, Temp, RH, WS | 0.04937 | 0.23297 | 0.01020 | 0.10096 | |
| | ชั่วโมง | train set 80 % test set 20% | CC | 0.19347 | 1.20508 | 0.04797 | 0.21903 |
| | | วันสุดท้ายเป็น test set | CC | 0.14199 | 0.72599 | 0.02345 | 0.15315 |
| | รายวัน | train set 80 % test set 20% | CC | 0.24643 | 1.36781 | 0.07708 | 0.27764 |

จากตารางที่ 4.17 คือการที่นำค่าเฉลี่ยของ MAE, MAPE, MSE และ RMSE ของ lags5, lags10, lags15, lags20, lags25, lags30 ของโมเดลที่ดีที่สุดมาเฉลี่ยกัน เพื่อเปรียบเทียบว่าการแบ่งชุดข้อมูลแบบใด ทำให้โมเดลมีค่าความคลาดเคลื่อนน้อยที่สุด เมื่อพิจารณาจากค่า MAPE พบร่วมกับ Prophet มีค่า MAPE น้อยที่สุดอยู่ที่ 0.35754 ซึ่งเป็นการเทรนในชุดข้อมูลที่ทำการแบ่งวันสุดท้ายแบบนาทีเป็น Test set และโมเดล LSTM มีค่า MAPE น้อยที่สุดอยู่ที่ 0.05202 ซึ่งเป็นการเทรนในชุดข้อมูลที่ทำการแบ่งวันสุดท้ายแบบนาทีเป็น Test set

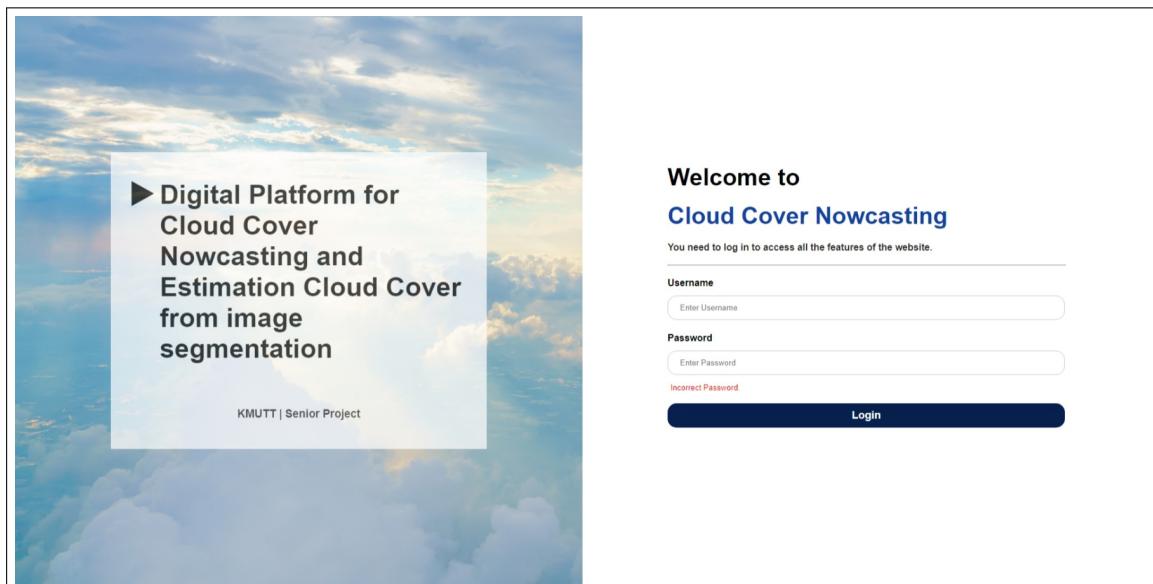
4.4 การพัฒนา User Interface

ในส่วนนี้จะเป็นผลการพัฒนาเว็บแอปพลิเคชันที่ได้ทำการออกแบบไว้โดยมีผู้ใช้งานเข้ามาในเว็บแอปพลิเคชันนี้ จะต้องทำการเข้าสู่ระบบ ก่อนตามรูปที่ [4.61](#) และเมื่อเข้าสู่ระบบเสร็จเรียบร้อยแล้ว ผู้ใช้งานจะสามารถใช้งานฟีเจอร์ของเว็บแอปพลิเคชันได้

4.4.1 หน้าจอ Welcome



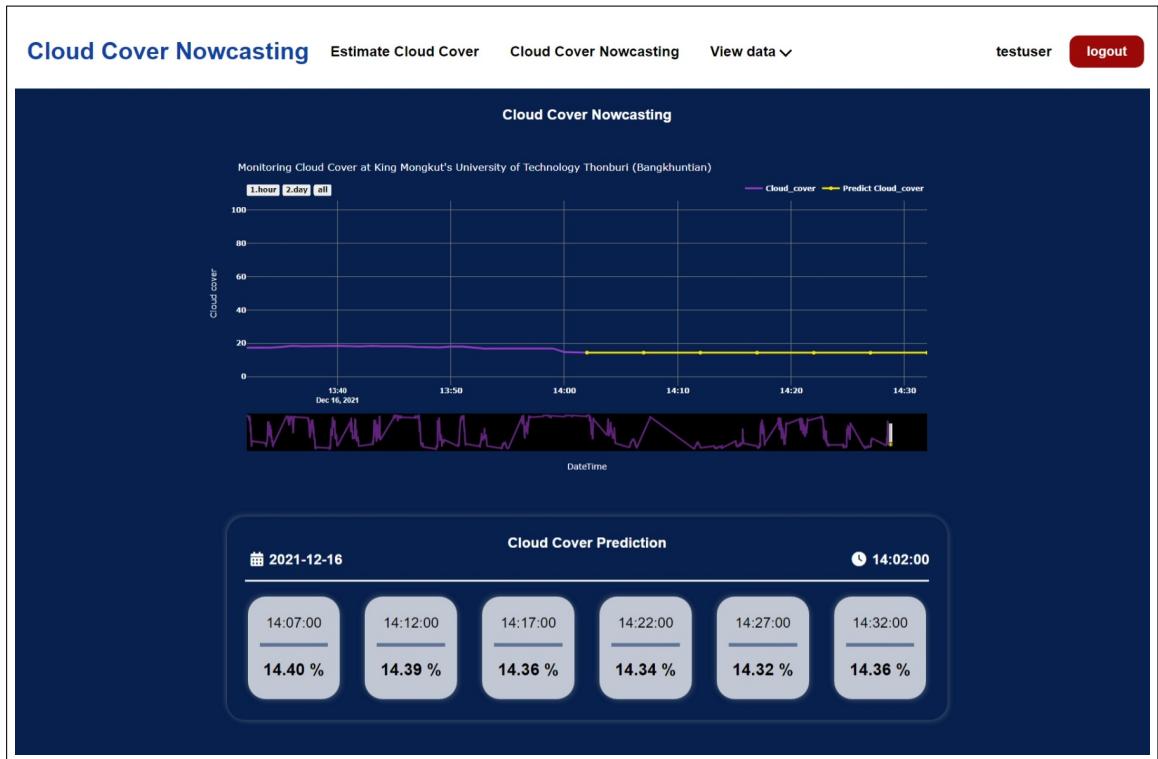
รูปที่ 4.61 หน้าหลักก่อนเข้าสู่ระบบของเว็บแอปพลิเคชัน



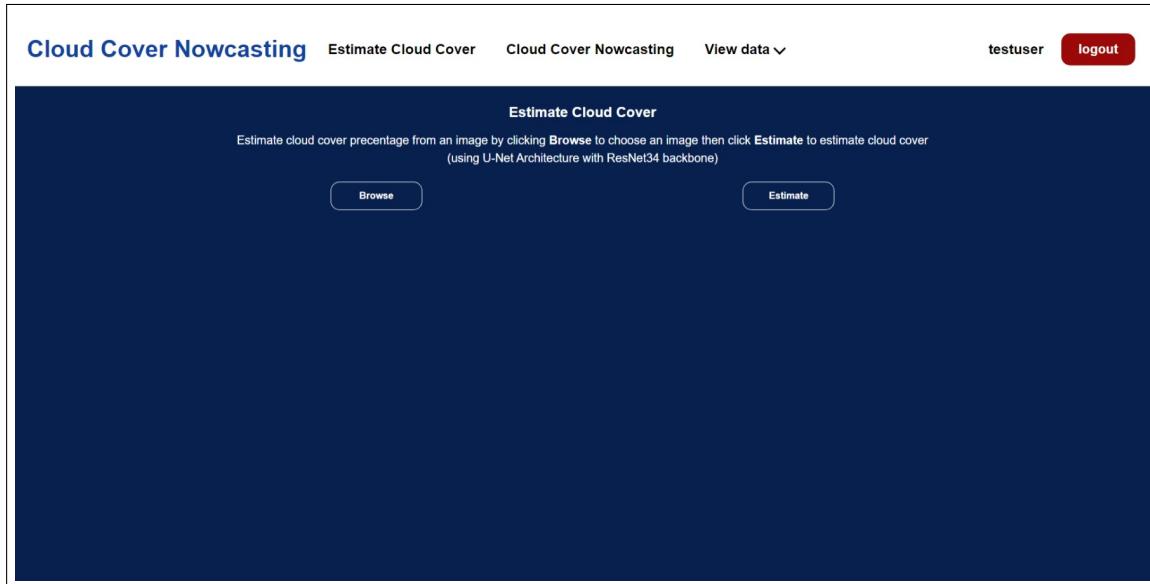
รูปที่ 4.62 การแสดงผลของข้อความผิดพลาดเมื่อเข้าสู่ระบบไม่สำเร็จ

4.4.2 หน้าจอ Nowcasting

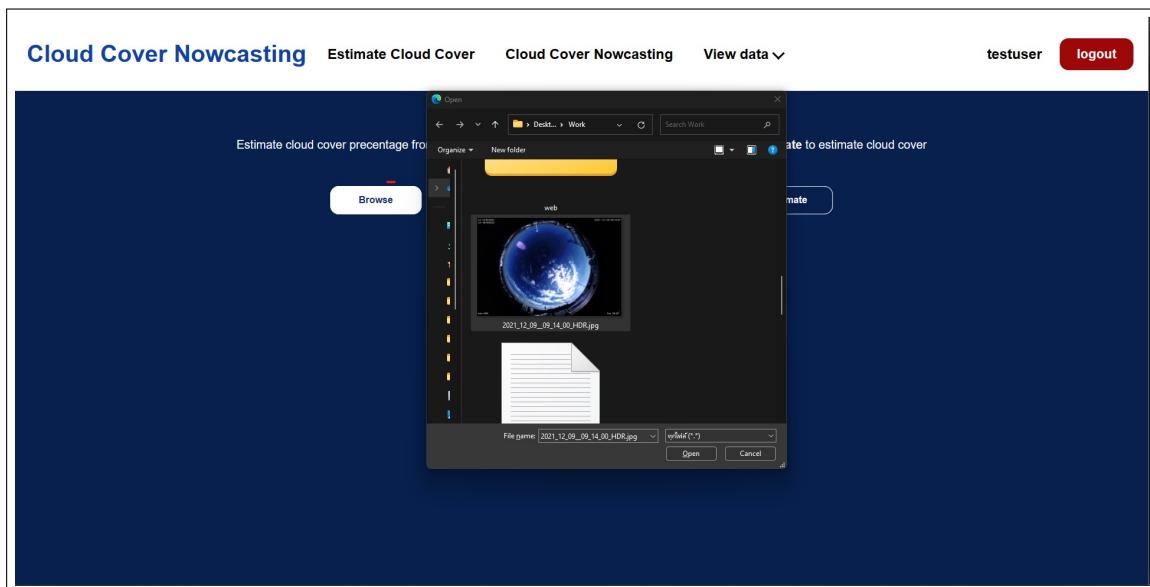
ในส่วนนี้จะเป็นหน้าหลักหลังจากการเข้าสู่ระบบ โดยจะประกอบไปด้วยกราฟเส้นที่แสดงผลของปริมาณเมฆทั้งหมดและผลของการทำนายปริมาณเมฆจากโมเดล LSTM ซึ่งได้ทำนายปริมาณเมฆทุก 5 นาที เป็นเวลาทั้งหมด 30 นาทีดังรูปที่ 4.63



รูปที่ 4.63 หน้าหลักหลังจากการเข้าสู่ระบบหรือหน้าการแสดง Cloud Cover Nowcasting



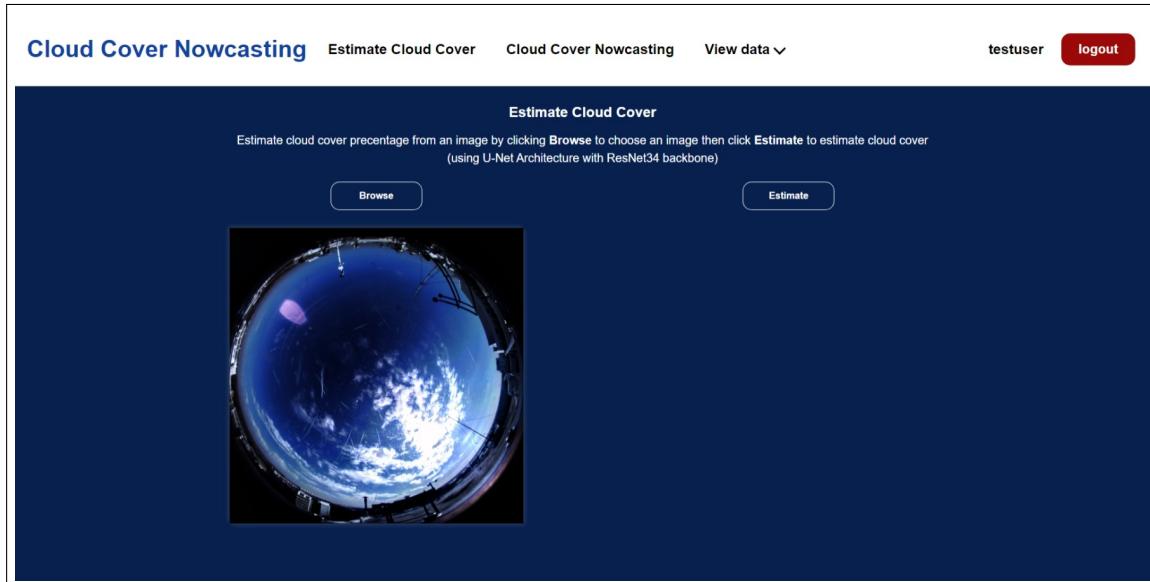
รูปที่ 4.64 หน้าการหาปริมาณเมฆด้วยรูปภาพ



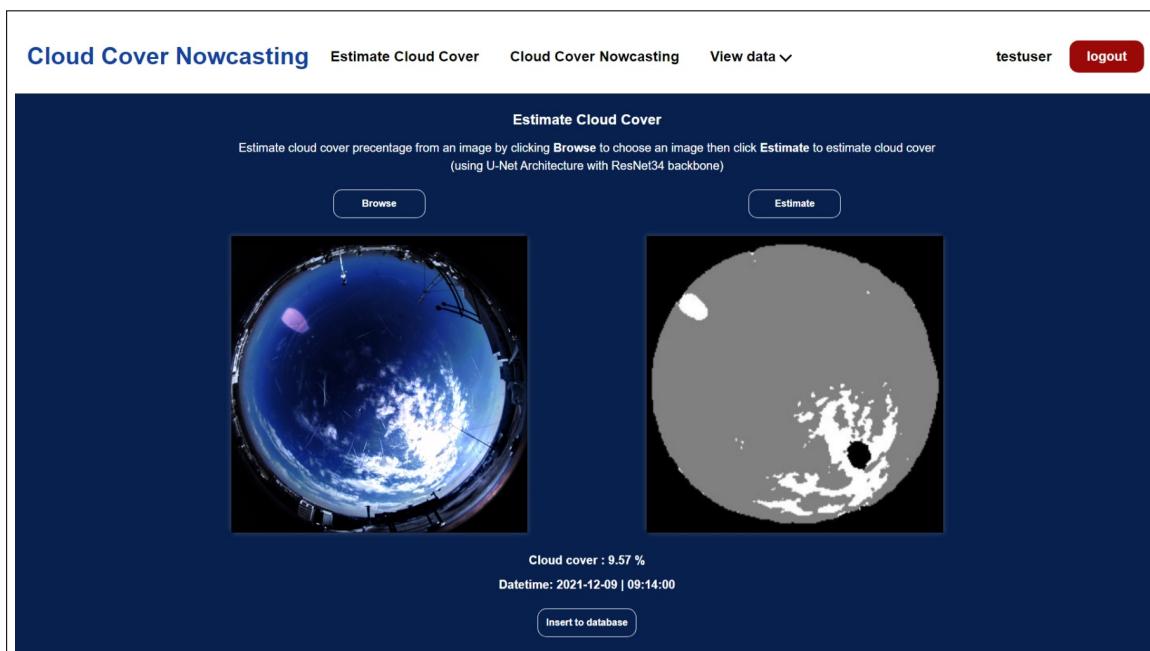
รูปที่ 4.65 หน้าการหาปริมาณเมฆด้วยรูปภาพเมื่อกดปุ่ม Browse

4.4.3 หน้าจอ Estimate Cloud Cover

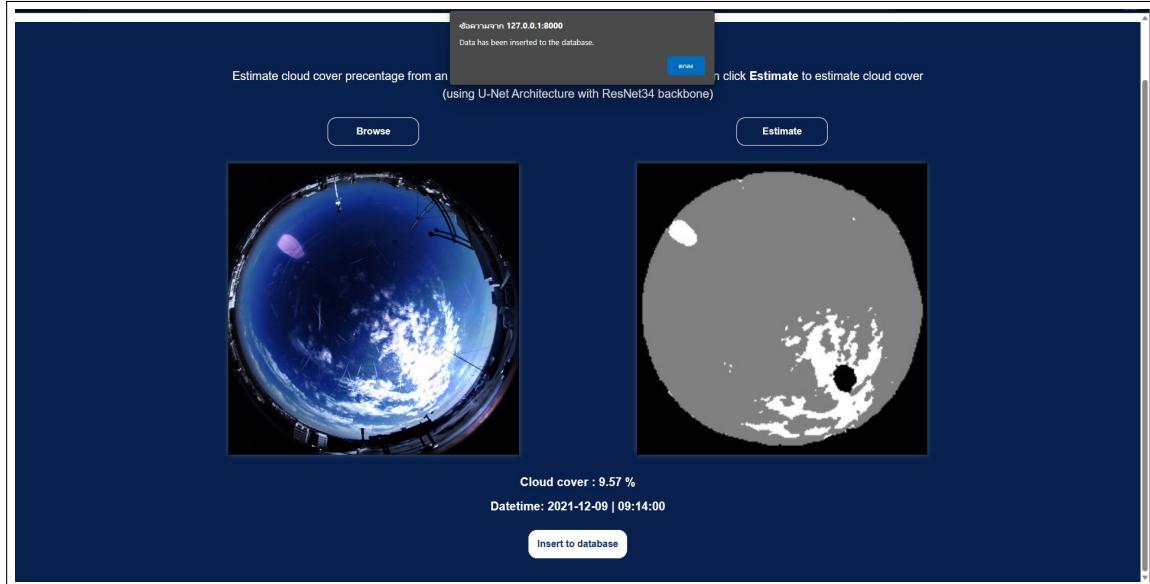
ผลการดำเนินงานต่อไปนี้จะเป็นส่วนของการคำนวณปริมาณเมฆจากรูปภาพ โดยเริ่มตั้งแต่การเข้าสู่หน้าของการหาปริมาณเมฆด้วยรูปภาพ ไปจนถึงการแสดงผลลัพธ์ ซึ่งจะเป็นดังรูปด้านล่าง



รูปที่ 4.66 หน้าการแสดงผลภาพต้นฉบับของการหาปริมาณเมฆด้วยรูปภาพ



รูปที่ 4.67 หน้าการแสดงผลลัพธ์ของการหาปริมาณเมฆด้วยรูปภาพ



รูปที่ 4.68 หน้าการแสดงผลเมื่อบันทึกข้อมูลลงระบบฐานข้อมูลสำเร็จ

| Cloud Cover Nowcasting | | | | Estimate Cloud Cover | Cloud Cover Nowcasting | View data ▾ | testuser | logout |
|-----------------------------|-----------------|--------------------------------------|--|--------------------------------------|--|-----------------------------|----------|--------|
| View cloud cover image data | | | | | | | | |
| Datetime | Cloud Cover (%) | Original Image | Segment Image | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_Yi6whdI.jpg | segment_2021_12_09_09_14_00_HDR_Yi6whdI.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_d53zydr.jpg | segment_2021_12_09_09_14_00_HDR_d53zydr.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_JGEKJ4g.jpg | segment_2021_12_09_09_14_00_HDR_JGEKJ4g.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_9pWtmpJ.jpg | segment_2021_12_09_09_14_00_HDR_9pWtmpJ.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_9pWtmpJ.jpg | segment_2021_12_09_09_14_00_HDR_9pWtmpJ.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_IBQithK1.jpg | segment_2021_12_09_09_14_00_HDR_IBQithK1.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_IBQithK1.jpg | segment_2021_12_09_09_14_00_HDR_IBQithK1.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_IHzelf9.jpg | segment_2021_12_09_09_14_00_HDR_IHzelf9.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_V6duimF.jpg | segment_2021_12_09_09_14_00_HDR_V6duimF.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_JsqEVm.jpg | segment_2021_12_09_09_14_00_HDR_JsqEVm.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_BMb0z2h.jpg | segment_2021_12_09_09_14_00_HDR_BMb0z2h.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_1yrG9xM.jpg | segment_2021_12_09_09_14_00_HDR_1yrG9xM.jpg | | | | | |
| 2021-12-09 09:14:00 | 9.57 | 2021_12_09_09_14_00_HDR_t3LUJsU.jpg | segment_2021_12_09_09_14_00_HDR_t3LUJsU.jpg | | | | | |

รูปที่ 4.69 หน้าข้อมูลการแสดงข้อมูลจากการคำนวณปริมาณเมฆ

4.4.4 หน้าจอ View image data

ผลการดำเนินงานต่อไปนี้จะเป็นส่วนของการการแสดงข้อมูลจากการคำนวณปริมาณเมฆจากรูปภาพทั้งหมดที่ผู้ใช้งานได้ทำการบันทึกข้อมูลลงระบบฐานข้อมูลจากส่วนของหน้าจอ Estimate Cloud Cover

Cloud Cover Nowcasting Estimate Cloud Cover Cloud Cover Nowcasting View data **testuser** logout

View cloud cover image data

| Datetime | Lat: 18.854306 Lon: 98.958222 | Original Image | Segment Image |
|---------------------|----------------------------------|--------------------------------------|---|
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_Yl6whdI.jpg | 4_00_HDR_Yl6whdI.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_d53zy0r.jpg | 4_00_HDR_d53zy0r.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_JGEKJ4g.jpg | 4_00_HDR_JGEKJ4g.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_9pWtmpJ.jpg | 4_00_HDR_9pWtmpJ.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_9pWtmpJ.jpg | 4_00_HDR_9pWtmpJ.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_IBQithK1.jpg | 4_00_HDR_IBQithK1.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_IBQithK1.jpg | 4_00_HDR_IBQithK1.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_lHzelf9.jpg | 4_00_HDR_lHzelf9.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_V6dulmF.jpg | 4_00_HDR_V6dulmF.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_jsqEVlm.jpg | 4_00_HDR_jsqEVlm.jpg |
| 2021-12-09 09:14:00 | | 2021-12-09_09_14_00_HDR_BMb02h.jpg | 4_00_HDR_BMb02h.jpg |
| 2021-12-09 09:14:00 | Auto HDR | 2021-12-09_09_14_00_HDR_t3LUJsU.jpg | segment_2021_12_09_09_14_00_HDR_t3LUJsU.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_t3LUJsU.jpg | segment_2021_12_09_09_14_00_HDR_t3LUJsU.jpg |

« ⏴ Page 7 of 8 ⏵ »

รูปที่ 4.70 การแสดงผลรูปภาพต้นฉบับเมื่อถูกตัดที่ชื่อของรูปภาพ

Cloud Cover Nowcasting Estimate Cloud Cover Cloud Cover Nowcasting View data **testuser** logout

View cloud cover image data

| Datetime | Cloud Cover (%) | Original Image | Segment Image |
|---------------------|-----------------|--------------------------------------|--|
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_Yl6whdI.jpg | segment_2021_12_09_09_14_00_HDR_Yl6whdI.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_d53zy0r.jpg | segment_2021_12_09_09_14_00_HDR_d53zy0r.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_JGEKJ4g.jpg | segment_2021_12_09_09_14_00_HDR_JGEKJ4g.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_9pWtmpJ.jpg | segment_2021_12_09_09_14_00_HDR_9pWtmpJ.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_9pWtmpJ.jpg | segment_2021_12_09_09_14_00_HDR_9pWtmpJ.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_IBQithK1.jpg | segment_2021_12_09_09_14_00_HDR_IBQithK1.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_IBQithK1.jpg | segment_2021_12_09_09_14_00_HDR_IBQithK1.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_lHzelf9.jpg | segment_2021_12_09_09_14_00_HDR_lHzelf9.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_V6dulmF.jpg | segment_2021_12_09_09_14_00_HDR_V6dulmF.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_jsqEVlm.jpg | segment_2021_12_09_09_14_00_HDR_jsqEVlm.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_BMb02h.jpg | segment_2021_12_09_09_14_00_HDR_BMb02h.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_1yrG9xM.jpg | segment_2021_12_09_09_14_00_HDR_1yrG9xM.jpg |
| 2021-12-09 09:14:00 | 9.57 | 2021-12-09_09_14_00_HDR_t3LUJsU.jpg | segment_2021_12_09_09_14_00_HDR_t3LUJsU.jpg |

« ⏴ Page 7 of 8 ⏵ »

รูปที่ 4.71 การแสดงผลรูปภาพที่ทำการ Segmentation และเมื่อถูกตัดที่ชื่อของรูปภาพ

4.4.5 หน้าจอ View weather data

ผลการดำเนินงานต่อไปนี้จะเป็นส่วนของการการแสดงผลข้อมูลปริมาณเมฆกับตัว预报สภาพอากาศทั้งหมดที่ใช้ในการสร้างโมเดล LSTM และทำนายผลของปริมาณเมฆ



The screenshot shows a web-based application titled "Cloud Cover Nowcasting". At the top, there are navigation links: "Estimate Cloud Cover", "Cloud Cover Nowcasting", "View data", and user information "testuser" with a "logout" button. Below the header is a dark blue section titled "View cloud cover with weather data". This section contains a table with the following data:

| Datetime | EEG | Temperature (°C) | Relative humidity (%) | Windspeed (km/hr) | Cloud Cover (%) |
|---------------------|--------|------------------|-----------------------|-------------------|-----------------|
| 2021-11-16 11:15:00 | 389.31 | 29.33 | 68.97 | 1.74 | 99.00 |
| 2021-11-16 11:16:00 | NA | NA | NA | NA | 98.90 |
| 2021-11-16 11:17:00 | NA | NA | NA | NA | 98.77 |
| 2021-11-16 11:18:00 | NA | NA | NA | NA | 98.70 |
| 2021-11-16 11:19:00 | NA | NA | NA | NA | 98.70 |
| 2021-11-16 11:20:00 | 386.47 | 29.26 | 69.64 | 1.72 | 98.97 |
| 2021-11-16 11:21:00 | NA | NA | NA | NA | 99.20 |
| 2021-11-16 11:22:00 | NA | NA | NA | NA | 99.30 |
| 2021-11-16 11:23:00 | NA | NA | NA | NA | 99.30 |
| 2021-11-16 11:24:00 | NA | NA | NA | NA | 99.40 |
| 2021-11-16 11:25:00 | 362.68 | 29.27 | 70.65 | 1.74 | 99.37 |
| 2021-11-16 11:26:00 | NA | NA | NA | NA | 99.30 |
| 2021-11-16 11:27:00 | NA | NA | NA | NA | 99.30 |

Page 1 of 1159 < >

รูปที่ 4.72 หน้าจอการแสดงข้อมูลปริมาณเมฆกับตัว预报สภาพอากาศ

4.5 ผลการทดสอบการทำงาน

ตารางที่ 4.18 ตารางการแสดงผลการทดสอบการทำงานของหน้าจอต่างๆ ภายใต้เว็บแอปพลิเคชัน

| หน้าจอ | การทำงาน | การแสดงผลข้อมูล | ความต้องของ Input | ความต้องของ Output ที่ได้จาก Backend |
|-------------------|---|-----------------|-------------------|--------------------------------------|
| Welcome | การเข้าสู่ระบบ | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Welcome | การแจ้งเตือนเมื่อเข้าสู่ระบบไม่ถูกต้อง | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Welcome | การออกจากระบบ | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Nowcasting | การแสดงผลกราฟเส้น | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Nowcasting | การแสดงผลการท่านาย | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Calculate | การค้นหารูปภาพและแสดงรูปภาพนั้น | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Calculate | การ Segmentation ของรูปภาพ | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Calulate | การแสดงผลลัพธ์ทั้งหมด | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| Calculate | การเพิ่มข้อมูลลงระบบฐานข้อมูล | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| View image data | การแสดงผลข้อมูลการคำนวณปริมาณเมฆจากรูปภาพในรูปแบบตาราง | ถูกต้อง | ถูกต้อง | ถูกต้อง |
| View weather data | การแสดงผลข้อมูลปริมาณเมฆกับตัวแปรสภาพอากาศในรูปแบบตาราง | ถูกต้อง | ถูกต้อง | ถูกต้อง |

บทที่ 5 บทสรุป

5.1 สรุปผลการทดลองไม้เดลตรวจจับเมฆ

5.1.1 สรุปผลการทดลองตรวจจับเมฆด้วย Otsu's method

การตรวจจับเมฆด้วยวิธีนี้เป็นวิธีของ Traditional Image Segmentation โดยวิธีที่แบ่งพิกเซลของเมฆ ห้องฟ้า และดวงอาทิตย์เกิดจาก การตั้งค่า Threshold ให้เหมาะสมต่อความส่วนของแต่ละพิกเซล จากผลการทดลองพบว่า Otsu's method มีค่า meanIOU, Accuracy และ Jaccard coefficient ที่ 50% แต่เมื่อพิจารณา Loss แล้วพบว่าวิธีนี้มีค่า Pixel error ยังเยื่อยู่มาก ทำให้ความถูกต้องจริงในแต่ละคลาสอย่างไม่แน่นอนมากนัก ทำให้ไม่เหมาะสมกับรูปถ่ายห้องฟ้าบางภาพที่มีค่าความส่วนต่างจากค่า Threshold ที่ตั้งค่าไว้ในการทดลอง

5.1.2 สรุปผลการทดลองไม้เดลตรวจจับเมฆด้วย U-Net

การตรวจจับเมฆด้วยวิธีนี้เป็นวิธีของ Deep Learning โดยวิธีที่แบ่งพิกเซลของเมฆ ห้องฟ้า และดวงอาทิตย์เกิดจากการเรียนรู้โดยมีผู้สอน กล่าวคือต้องมีชุดข้อมูลรูปภาพถ่ายห้องฟ้าที่มี Ground truth เพื่อให้โมเดลสามารถเรียนรู้ลักษณะค่าสีในพิกเซลของแต่ละคลาส ได้ดี โดยชุดข้อมูลที่ใช้เห็นโน้มเดลคือชุดข้อมูล WSISEG-Database จำนวน 400 รูปซึ่งไม่เพียงพอสำหรับการเห็นโน้มเดล การทำ Data Augmentation จึงเป็นวิธีที่เพิ่มจำนวนชุดข้อมูลได้ดีที่สุด โดยการเปลี่ยนแปลงชุดข้อมูลในรูปแบบต่าง ๆ ทำให้มีจำนวนข้อมูลเพิ่มขึ้นเป็น 2000 รูป ซึ่งเพียงพอต่อการเห็นโน้มเดล U-Net ในระดับนี้ ในการทดลองมีการเพิ่มปรับเปลี่ยนส่วนของ encoder โดยใช้โครงสร้าง ResNet34 มาช่วยเพิ่มความสามารถในการเรียนรู้ภาพที่เป็น layer ของ CNN ทำให้ผลการทดลองพบว่าการใช้ U-Net ร่วมกับ ResNet34 ที่เห็นกับข้อมูลที่ทำ Data Augmentation และปรับส่วนของ Hyperparameter ทำให้ค่า Accuracy ในการตรวจจับเมฆสูงถึง 98.24%

5.2 สรุปผลการทดลองไม้เดลการพยากรณ์ปริมาณเมฆ

5.2.1 สรุปผลการทดลองของ Prophet

จากการทดลอง การทำตัวแปรเดียวสำหรับการแบ่งชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% ในการทำนายนาที ชั่วโมง รายวัน และรายตัวแปร พบร่วมกับการแบ่งชุดข้อมูลเป็นรายวันนั้น มีลักษณะของเส้น Predict ที่อยู่ในช่วงเป็นบวกมากกว่าการแบ่งชุดข้อมูล เป็นนาที ชั่วโมง และรายตัวแปร เนื่องจากการแบ่งชุดข้อมูลเป็นนาที ชั่วโมง และรายตัวแปรทำให้เส้น Predict มีค่าติดลบ โดยการแบ่งชุดข้อมูลเป็นรายวัน มีการทำนายที่มีลักษณะเส้นการทำนายคล้ายกับเส้น Actual โดยเส้น Predict มีช่วงน้อยกว่าและมากกว่าเส้น Actual แต่ค่า Predict นั้นมีค่าเป็นบวก และการที่เส้น Predict อยู่ในช่วงติดลบ เพราะว่าชุดข้อมูลมีเวลาที่หายไปมาก และ Prophet เป็นการค้นหาชุดของไอยิเปอร์พารามิเตอร์ที่ดีสำหรับแบบจำลองโดยอัตโนมัติ เพื่อทำนายต่อจากข้อมูลที่ไม่ได้มีการ laggs เนื่องจาก Prophet นั้นไม่มี laggs ข้อมูล และจากการทดลอง การทำตัวแปรเดียวสำหรับการแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ในการทำนายนาที ชั่วโมง และรายตัวแปร พบร่วมกับ Graficแสดงเส้น Predict ที่อยู่ในค่าที่เป็นบวกและลักษณะเส้น Predict มีความคล้ายกับเส้น Actual แต่เส้น Predict จะอยู่สูงกว่าเส้น Actual แต่การทำนายโดยใช้หลักการตัวแปรทำให้เส้น Predict คล้ายกับเส้น Actual มากกว่า Model ที่ทำมาก เพราะว่าชุดข้อมูลน้อยและมีตัวแปรอื่นที่ช่วยเสริมทำให้เส้น Predict นั้นคล้ายกับเส้น Actual ดังนั้นการแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ในการทำนายเป็นนาทีนั้นให้ Grafic ที่ดีกว่าการทำตัวแปรเดียวสำหรับการแบ่งชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20%

5.2.2 สรุปผลการทดลองของ LSTM

จากการทดลอง การทำตัวแปรเดียวสำหรับการแบ่งชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% ในการทำนายนาที ชั่วโมง รายวัน และรายตัวแปร พบร่วมกับการทำนายที่เป็นนาทีและรายตัวแปรมีลักษณะของเส้น Predict ที่มีความใกล้เคียงกับเส้น Actual มากกว่าการทำนายที่เป็นชั่วโมงและรายวัน เนื่องจากการทำนายที่เป็นชั่วโมงและรายวันนั้นแสดง Grafic ที่เส้น Predict ที่มีช่วงมากกว่าและน้อยกว่าเส้น Actual เพราะว่าชุดข้อมูลของชั่วโมงและรายวันน้อยเกินไปในการ Train เพื่อเรียนรู้ LSTM และจากการทดลอง การทำตัวแปรเดียวสำหรับการแบ่งชุดข้อมูล โดยเลือกวันสุดท้ายเป็น test set ในการทำนายนาที ชั่วโมง และรายตัวแปร พบร่วมกับการทำนายที่เป็นนาทีและรายตัวแปรมีลักษณะของเส้น Predict ที่มีความใกล้เคียงกับเส้น Actual มากกว่าการทำนายที่เป็นชั่วโมง เนื่องจากการทำนายที่เป็นชั่วโมงนั้นแสดง Grafic ที่เส้น Predict ที่มีช่วงมากกว่าและน้อยกว่าเส้น Actual เพราะว่าชุดข้อมูลของชั่วโมงน้อยเกินไปในการ Train เพื่อเรียนรู้ LSTM ดังนั้นการแบ่งชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% ในการทำนายเป็นนาทีนั้นให้ Grafic ที่ใกล้เคียง

กับการแบ่งชุดข้อมูลโดยเลือกวันสุดท้ายเป็น test set ทำให้เลือกการแบ่งชุดข้อมูล โดย Train set เป็น 80% และ Test set เป็น 20% ในการนี้ไปใช้กับเว็บแอปพลิเคชัน เนื่องจากการแบ่งชุดข้อมูลนั้นมีจำนวน Test set ที่ใช้ประเมินประสิทธิภาพของโมเดลมากกว่าการแบ่งชุดข้อมูลโดยเลือกวันสุดท้ายเป็น Test set

5.2.3 สรุปผลการเปรียบเทียบระหว่าง Prophet และ LSTM

จากตาราง 4.17 การแบ่งชุดข้อมูลเป็นนาทีและชั่วโมงของ LSTM มีค่า MAE , MAPE , MSE และ RMSE น้อยกว่า Prophet และการแบ่งชุดข้อมูลเป็นวันของ LSTM ทำให้ค่า MAE , MAPE , MSE และ RMSE มากกว่า Prophet ทำให้กราฟของ LSTM มีการทำงานข้อมูลที่เป็นนาทีและชั่วโมงได้ดีกว่า Prophet และกราฟของ Prophet สามารถทำงานชุดข้อมูลที่เป็นวันได้ดีกว่า LSTM ตั้งแต่การทำงาน 5, 10, 15, 20, 25 และ 30 นาทีต้องมีการใช้ข้อมูลในอดีตย้อนหลังตามนาทีที่ต้องการทำงานทำให้ LSTM เหมาะสมกับการทำงานในอนาคตที่ต้องการมากกว่า Prophet เนื่องจาก LSTM มีการ lags ข้อมูลตามนาทีที่ต้องการทำงานและทำงานนาทีได้ใกล้เคียงกับค่า Cloud Cover มากกว่า Prophet

5.2.4 สรุปผลการพัฒนาเว็บแอปพลิเคชัน

จากการพัฒนาเว็บแอปพลิเคชันสำหรับการทำงานเบื้องต้นของ LSTM และ Prophet พบว่ามีการพัฒนาได้ตรงตามที่วางแผนไว้โดยโมเดลเลือกนำมา Deploy บนเว็บแอปพลิเคชัน แบ่งเป็น 2 ส่วนได้แก่ ส่วนแรกเป็นส่วนของโมเดลที่ใช้ทำงานปริมาณเมฆในช่วงเวลา 5-30 นาทีในอนาคตคือ โมเดล LSTM ที่ถูกтренด้วยชุดข้อมูลที่แบ่งเป็น Train set เป็น 80% และ Test set เป็น 20% และส่วนที่สอง เป็นส่วนของโมเดลที่ใช้ตรวจสอบเมฆบนรูปภาพที่นำมาใช้คือ โมเดล U-Net + ResNet34 ที่ trened ด้วยชุดข้อมูลที่ทำ Data Augmentation

5.3 ปัญหาที่พบและการแก้ไข

1. การเปรียบเทียบโมเดลระหว่าง LSTM และ Prophet อาจไม่เหมาะสม เนื่องจาก LSTM มีการ lags ข้อมูลในอดีต แต่ Prophet ไม่มี การ lags ข้อมูลในอดีต อาจเปลี่ยนวิธี Prophet เป็นวิธี Neural Prophet ซึ่งพัฒนามาจากพื้นฐานของ Deep Neural Networks และนำมาปรับใช้กับ Facebook prophet ที่มีพื้นฐานมาจาก Machine Learning และ A simple Auto-Regressive Neural Network for time-series หรือ AR-net ที่จะทำให้ปรับชุดไอยเปอร์พารามิเตอร์ให้มีการ lags ได้
2. เนื่องจากชุดข้อมูลจากนักวิจัยมจ.วิทยาเขตบางขุนเทียนไม่มี Ground truth ทำให้คณบุรุษัดทำต้องนำชุดข้อมูลภาพนอกมาใช้ เทเรนโมเดลเพื่อเรียนรู้การตรวจจับเมฆอัตโนมัติ และนำโมเดลที่ได้ไปปรับใช้กับชุดข้อมูลจากนักวิจัยมจ.วิทยาเขตบางขุนเทียน

5.4 แนวทางในการพัฒนาโครงการ

1. ตัวแปรที่ใช้ในการพยากรณ์ปริมาณเมฆในช่วงเวลา 5-30 นาที นอกจากค่าความชื้น อุณหภูมิ ความเร็วลม และปริมาณเมฆ อาจจะ ต้องมีการนำอาค่าแสงมาใช้ในการพยากรณ์ด้วย อาจทำให้โมเดลมีความแม่นยำมากยิ่งขึ้นกว่าการใช้ปริมาณเมฆในการพยากรณ์ เพียงตัวแปรเดียว
2. การเปรียบเทียบ Model ระหว่าง LSTM และ Prophet อาจไม่พอสำหรับการวิจัย ควรเพิ่มโมเดลเพื่อเปรียบเทียบ เช่นใช้ Arima, CNN-LSTM หรือ MLPs เป็นต้น
3. ข้อมูลที่ใช้ในการทำงานปริมาณเมฆในช่วง 5-30 นาทีในอนาคตต้องเก็บไว้ ทำให้ค่า Error ไม่แม่นยำกันนัก หากมีการเก็บข้อมูล เพิ่มมากขึ้นอาจทำให้ประสิทธิภาพในการทำงานปริมาณเมฆเพิ่มขึ้นมากกว่าเดิม
4. หากข้อมูลรูปภาพที่จะคำนวนปริมาณเมฆจากรูปภาพ หากไม่มีรูป Ground truth จริง ๆ อาจทดลองใช้วิธีของ Image Regression ในกรณีคำนวนปริมาณเมฆ และเปรียบเทียบกับปริมาณเมฆจากชุดข้อมูลของนักวิจัยมจ.วิทยาเขตบางขุนเทียน
5. เว็บแอปพลิเคชันในโครงงานนี้ได้อีคอมต่อ กับเว็บแอปพลิเคชันส่วนของ Backend และ Database ที่สร้างขึ้นในโครงงานนี้ยังไม่เหมาะสมมากนัก หากในอนาคตนักวิจัยมีการนำไปใช้จริง อาจต้องปรับโครงสร้างเว็บ แอปพลิเคชันให้เหมาะสมกับภาพที่ถ่ายรูปท้องฟ้าแบบ Realtime

หนังสืออ้างอิง

1. Assoc.Prof. Dr. Sathit Intajag, 2015, ``การแบ่งข้อมูลภาพ Image Segmentation," Available at <http://staff.cs.psu.ac.th/sathit/344-671/Image%20Segmentation.pdf>, [Online; accessed 6-September-2022].
2. กิตติโนย แท้ประสาทสิทธิ์, 2015, ``ความรู้เบื้องต้นเกี่ยวกับภาพและการประมวลผลภาพ," Available at https://webserv.cp.su.ac.th/lecturer/pinyotae/comvision/files/Week2_BasicOfImageAndImageProcessing_fullPage.pdf, [Online; accessed 6-September-2022].
3. Digitalschool, 2019, ``อนุกรมเวลา," Available at http://www.digitalschool.club/digitalschool/math2_2_1/math17_3/more/page8.php, [Online; accessed 17-September-2022].
4. cwayinvestment, 2018, ``การพยากรณ์ด้วยโมเดล Prophet," Available at <http://cway-quantlab.blogspot.com/2018/10/downside-protection-model.html>, [Online; accessed 17-September-2022].
5. David A. Yu, 2021, ``Cloud Coverage Prediction to Improve Solar Power Management," Available at <https://scholarworks.calstate.edu/downloads/xw42nf32n>, [Online; accessed 16-September-2022].
6. Rial Arifin Rajagukguk, 2021, ``A Deep Learning Model to Forecast Solar Irradiance Using a Sky Camera," Available at https://www.researchgate.net/publication/351994096_A_Deep_Learning_Model_to_Forecast_Solar_Irradiance_Using_a_Sky_Camera, [Online; accessed 16-September-2022].
7. Wanyi Xie, 2020, ``SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation," Available at <https://amt.copernicus.org/articles/13/1953/2020/>, [Online; accessed 5-January-2023].
8. Jessica Stoller-Conrad, 2002, ``How Do Clouds Affect Solar Energy?," Available at <https://scijinks.gov/solar-energy-and-clouds/>, [Online; accessed 30-August-2022].
9. Jakkarin Sanuksan, 2018, ``โครงข่ายประสาทเทียมแบบคอนโวจูชันเชิงลึก," Available at <http://olarik.it.msu.ac.th/wp-content/uploads/2019/10/001-Jakkarin-2019.pdf>, [Online; accessed 16-September-2022].
10. รัชพงษ์ นันดาเพร, 2020, ``การเรียนรู้เชิงลึกโดยโครงข่ายประสาทเทียมคอนโวจูชันสำหรับกำหนดขอบเขตอัตโนมัติ," Available at <http://nuir.lib.nu.ac.th/dspace/bitstream/123456789/2323/3/61061932.pdf>, [Online; accessed 16-September-2022].
11. Surapong Kanoktipsatharporn, 2019, ``Image Segmentation คืออะไร," Available at <https://www.bualabs.com/archives/835/what-is-image-segmentation-semantic-segmentation-camvid-machine-learning-unet-deep-learning-image-segmentation-ep-1/>, [Online; accessed 17-September-2022].
12. Perth Ngarmtrakulchol, 2018, ``Time Series Forecasting คืออะไร?," Available at <http://cway-quantlab.blogspot.com/2018/10/downside-protection-model.html>, [Online; accessed 17-September-2022].
13. Sirinart Tangruamsub, 2017, ``Long Short-Term Memory:LSTM," Available at <https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>, [Online; accessed 7-November-2022].
14. Pttexpresso, 2021, ``SQL คืออะไร," Available at <https://blog.pttexpresso.com/what-is-sql/>, [Online; accessed 7-November-2022].
15. Youssef Karout, 2020, ``A supervised deep learning model for cloud/sky image segmentation," Available at <https://sfera3.sollab.eu/wp-content/uploads/2020/08/Karout-2020a.pdf>, [Online; accessed 5-January-2023].