

แบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ  
โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสติง



การศึกษาค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
แขนงวิชาเทคโนโลยีดิจิทัล สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช

พ.ศ. 2566

**Floating Solar Power Generation Forecasting Model**  
**Based on Long Short-Term Memory and Extreme Gradient Boosting Methods**



**Mr. Adirek Panboon**

An Independent Study Submitted in Partial Fulfillment of the Requirements for  
the Degree of Master of Science (Digital Technology)

School of Science and Technology

Sukhothai Thammathirat Open University

2023

หัวข้อการศึกษาค้นคว้าอิสระ      แบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์  
บนท่อนลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว  
และเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง

ชื่อและนามสกุล      นายอดิเรก ปั่นบุญ

วิชาเอก      เทคโนโลยีดิจิทัล

สาขาวิชา      วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช

อาจารย์ที่ปรึกษา      รองศาสตราจารย์ ดร. วุฒิชัย ร่มสายหยุด

การศึกษาค้นคว้าอิสระนี้ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรระดับปริญญาโท เมื่อวันที่ 28 กุมภาพันธ์ พ.ศ. 2567

คณะกรรมการสอบการศึกษาค้นคว้าอิสระ

..... ประธานกรรมการ  
(รองศาสตราจารย์ ดร. วุฒิชัย ร่มสายหยุด)

..... กรรมการ  
(ศาสตราจารย์ ดร. ปิยะ โควินท์ทวีวัฒน์)



.....  
(ผู้ช่วยศาสตราจารย์ ดร. ภูมิ เจือศิริภักดี)  
ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

**ชื่อการศึกษาค้นคว้าอิสระ แบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ  
โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกร  
เดียนต์บูสตีง**

**ผู้ศึกษา** นายอดิเรก ปั้นบุญ รหัสนักศึกษา 2659600445 **ปริญญา** วิทยาศาสตรมหาบัณฑิต  
(เทคโนโลยีดิจิทัล) **อาจารย์ที่ปรึกษา** รองศาสตราจารย์ ดร.วฤษาย์ ร่มสายหยุด  
**ปีการศึกษา** 2566

**บทคัดย่อ**

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ (1) ศึกษาวิธีการและสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสตีง และ (2) ประเมินประสิทธิภาพของแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสตีง

วิธีการวิจัยดำเนินการตามกระบวนการเรียนรู้เชิงลึก (Deep Learning) ซึ่งประกอบด้วย 3 ขั้นตอนหลัก ดังนี้ ขั้นตอนแรก เก็บรวบรวมข้อมูลการผลิตพลังงานไฟฟ้าและข้อมูลจากเซนเซอร์ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ ในช่วงเดือนกุมภาพันธ์ ถึง ตุลาคม พ.ศ. 2566 จำนวน 6,511 ระเบียบ และ 11 คุณลักษณะ ขั้นตอนที่สอง การคัดเลือกคุณลักษณะและการจำแนกประเภท โดยประยุกต์ใช้วิธีการแบบผสมผสานด้วยอัลกอริทึมการเรียนรู้เชิงลึก 2 อัลกอริทึม ได้แก่ 1) หน่วยความจำระยะสั้นแบบยาว (LSTM) สำหรับการจดจำข้อมูลในระยะยาวตามอนุกรมเวลา และ 2) เอ็กซ์ตรีมเกรเดียนต์บูสตีง (XGBoost) สำหรับการเรียนรู้จากชุดข้อมูลที่ไม่แน่นอนและการพยากรณ์ที่มีประสิทธิภาพสูง และขั้นตอนที่สาม การประเมินประสิทธิภาพของแบบจำลองโดยใช้ตัวชี้วัด ได้แก่ การวัดค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และรากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) สำหรับค่าผลลัพธ์ที่ได้จากแบบจำลองการพยากรณ์ ได้ค่า MAE เฉลี่ย 0.0577, MSE เฉลี่ย 0.0143 และ RMSE เฉลี่ย 0.1196 ซึ่งแสดงถึงค่าที่เหมาะสมในสถานการณ์จริง

**คำสำคัญ** การพยากรณ์, พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ, หน่วยความจำระยะสั้นแบบยาว, เอ็กซ์ตรีมเกรเดียนต์บูสตีง, วิธีการพยากรณ์แบบผสม

**Independent Study title:** Floating Solar Power Generation Forecasting Model  
Based on Long Short-Term Memory and Extreme  
Gradient Boosting Methods

**Author:** Mr. Adirek Panboon; **ID:** 2659600445;

**Degree:** Master of Science (Digital Technology);

**Independent Study Advisor:** Assoc. Prof. Dr. Walisa Romsaiyud;

**Academic year:** 2023

### **Abstract**

The purposes of this research were to (1) build a forecasting model for floating solar energy generation based on Long Short-Term Memory and Extreme Gradient Boosting methods and (2) to evaluate the performance of a model based on Long Short-Term Memory and Extreme Gradient Boosting methods.

The research methodology follows the deep learning pipeline, which consists of three main steps: In the first step, the data were collected from the electricity generation records and sensor readings obtained from a floating solar power facility with a capacity of 45 megawatts during February–October 2023, which consisted of 6,511 examples and 11 features. In the second step, feature extraction and classification were applied using the hybrid model by using two deep learning algorithms: 1) Long Short-Term Memory (LSTM) for memorizing long-term dependencies of time-series data, and 2) Extreme Gradient Boosting (XGBoost) for learning from uncertain data and predicting high performance; and In the third step, model evaluation was assessed using metrics including the mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE) for the indicated values from the forecasting model. The experiment result shows an average MAE of 0.0577, an average MSE of 0.0143, and an average RMSE of 0.1196 which represent suitable values in a real situation.

**Keywords :**Forecasting, Floating Solar Power, Long Short-Term Memory,  
Extreme Gradient Boosting, Hybrid Forecasting Method

## กิตติกรรมประกาศ

การศึกษาครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความกรุณาอย่างสูงจาก รองศาสตราจารย์ ดร.วราชัย รมสายหยุด อาจารย์ที่ปรึกษางานวิจัย ที่ได้กรุณาให้คำปรึกษา คำแนะนำ ในการทำงานวิจัย ตลอดจนสนับสนุนข้อมูลทางวิชาการและปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่ง ทำให้ผู้วิจัยมีความรู้ ความเข้าใจทฤษฎีและปฏิบัติมากยิ่งขึ้น ผู้วิจัยตระหนักถึงความตั้งใจจริงและความทุ่มเทของอาจารย์และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณพระคุณ ศาสตราจารย์ ดร.ปิยะ โควินท์ทวีวัฒน์ ที่กรุณาสละเวลามาเป็นกรรมการสอบการศึกษาครั้งนี้ตลอดจนให้คำแนะนำและข้อเสนอแนะเพิ่มเติมเกี่ยวกับการดำเนินงานวิจัยที่เป็นประโยชน์และช่วยตรวจสอบ ซึ่งข้อบกพร่องของงานวิจัย ทางผู้วิจัยได้นำข้อเสนอแนะเหล่านั้นมาปรับปรุงงานวิจัยและเก็บไว้เป็นความรู้ในการศึกษาต่อไป

นอกจากนี้ ผู้วิจัยขอขอบคุณแรงสนับสนุนจากครอบครัว กำลังใจจากมิตรสหาย รวมถึงคณาจารย์และเจ้าหน้าที่สาขาวิชาวิทยาศาสตร์และเทคโนโลยีทุกท่าน ที่ได้ให้ความรู้ ให้คำปรึกษา และอำนวยความสะดวก ตลอดระยะเวลาการศึกษา จนสามารถทำให้การศึกษาครั้งนี้สำเร็จลุล่วงสมบูรณ์



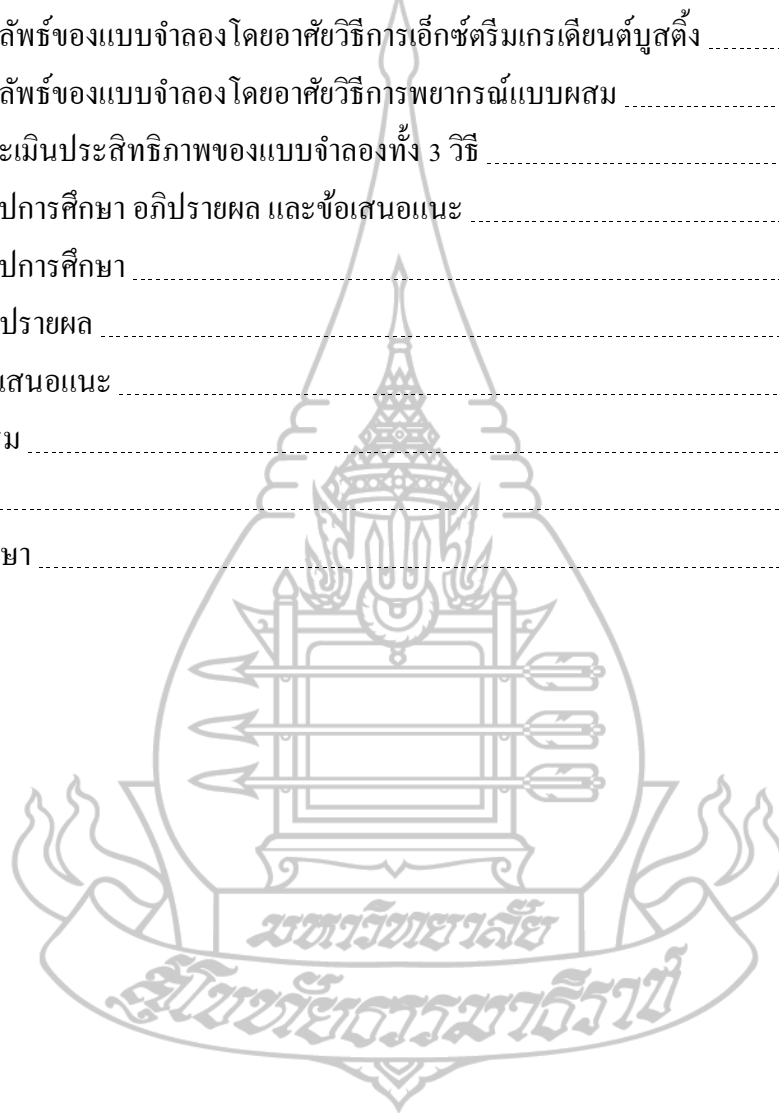
นายอดิเรก ปั่นบุญ

กุมภาพันธ์ 2567

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญตาราง .....	ณ
สารบัญภาพ .....	ญ
บทที่ 1 บทนำ .....	1
ความเป็นมาและความสำคัญของปัญหา .....	1
วัตถุประสงค์การศึกษา .....	3
กรอบแนวคิดการวิจัย .....	3
ขอบเขตการวิจัย .....	5
นิยามศัพท์เฉพาะ .....	5
ประโยชน์ที่คาดว่าจะได้รับ .....	6
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง .....	7
พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ (Floating Solar) .....	7
การพยากรณ์พลังงานแสงอาทิตย์ (Solar Power Forecasting) .....	10
หน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory) .....	12
เอ็กซ์ตรีมเกรเดียนต์บูสต์ติง (Extreme Gradient Boosting) .....	18
วิธีการพยากรณ์แบบผสม (Hybrid Forecasting Method) .....	20
การคัดเลือกคุณลักษณะ (Feature Selection) .....	22
การปรับขนาดข้อมูลให้สมมาตร (Scaling Data) .....	22
การแบ่งข้อมูลเพื่อประเมินประสิทธิภาพแบบจำลอง (Data Partition) .....	23
การประเมินประสิทธิภาพแบบจำลอง (Performance Evaluation) .....	25
งานวิจัยที่เกี่ยวข้อง .....	27
บทที่ 3 วิธีดำเนินการศึกษา .....	33
ประชากรและกลุ่มตัวอย่าง .....	33
เครื่องมือที่ใช้ในการวิจัย .....	33
การศึกษาและวิเคราะห์ปัญหา .....	34

การเก็บรวบรวมข้อมูลและสร้างแบบจำลอง .....	36
การทดสอบประสิทธิภาพของแบบจำลอง .....	57
บทที่ 4 ผลการวิเคราะห์ข้อมูล .....	59
ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว .....	59
ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูตส์ตริง .....	64
ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการพยากรณ์แบบผสม .....	67
ประเมินประสิทธิภาพของแบบจำลองทั้ง 3 วิธี .....	69
บทที่ 5 สรุปการศึกษา อภิปรายผล และข้อเสนอแนะ .....	72
สรุปการศึกษา .....	72
อภิปรายผล .....	74
ข้อเสนอแนะ .....	75
บรรณานุกรม .....	76
ภาคผนวก .....	80
ประวัติผู้ศึกษา .....	90





สารบัญตาราง

	หน้า
ตารางที่ 2.1	เทคนิคการพยากรณ์และคุณลักษณะเฉพาะ ..... 10
ตารางที่ 2.2	สรุปงานวิจัยที่เกี่ยวข้อง ..... 31
ตารางที่ 3.1	รายละเอียดของคุณลักษณะ ..... 37
ตารางที่ 3.2	รหัสโปรแกรมนำเข้าชุดข้อมูล CSV ..... 38
ตารางที่ 3.3	การอธิบายรหัสโปรแกรมนำเข้าชุดข้อมูล CSV ..... 39
ตารางที่ 3.4	รหัสโปรแกรมการแปลงข้อมูล ..... 40
ตารางที่ 3.5	การอธิบายรหัสโปรแกรมการแปลงข้อมูล ..... 40
ตารางที่ 3.6	รหัสโปรแกรมการเลือกคุณลักษณะ ..... 41
ตารางที่ 3.7	การอธิบายรหัสโปรแกรมการเลือกคุณลักษณะ ..... 41
ตารางที่ 3.8	รหัสโปรแกรมการคัดเลือกคุณลักษณะที่ต้องการนำไปใช้งาน ..... 43
ตารางที่ 3.9	การอธิบายรหัสโปรแกรมคัดเลือกคุณลักษณะที่ต้องการนำไปใช้งาน ..... 43
ตารางที่ 3.10	รหัสโปรแกรมการแสดงผล Heatmap ..... 44
ตารางที่ 3.11	การอธิบายรหัสโปรแกรมการแสดงผล Heatmap ..... 44
ตารางที่ 3.12	รหัสโปรแกรมการปรับค่าให้เป็นมาตรฐาน ..... 45
ตารางที่ 3.13	การอธิบายรหัสโปรแกรมการปรับค่าให้เป็นมาตรฐาน ..... 45
ตารางที่ 3.14	รหัสโปรแกรมการแบ่งข้อมูลสำหรับการสร้างแบบจำลอง ..... 46
ตารางที่ 3.15	การอธิบายรหัสโปรแกรมการแบ่งข้อมูลสำหรับการสร้างแบบจำลอง ..... 47
ตารางที่ 3.16	รหัสโปรแกรมการสร้างแบบจำลอง ..... 49
ตารางที่ 3.17	การอธิบายรหัสโปรแกรมการสร้างแบบจำลอง ..... 51
ตารางที่ 3.18	รหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง ..... 52
ตารางที่ 3.19	การอธิบายรหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง ..... 52
ตารางที่ 3.20	รหัสโปรแกรมแบบจำลองวิธีการพยากรณ์แบบผสม ..... 54
ตารางที่ 3.21	การอธิบายรหัสโปรแกรมแบบจำลองวิธีการพยากรณ์แบบผสม ..... 55
ตารางที่ 3.22	รหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง วิธีการพยากรณ์แบบผสม ..... 56
ตารางที่ 3.23	การอธิบายรหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง วิธีการพยากรณ์แบบผสม ..... 56
ตารางที่ 4.1	ไฮเปอร์พารามิเตอร์ที่ใช้กับแบบจำลองหน่วยความจำระยะสั้นแบบยาว ..... 61

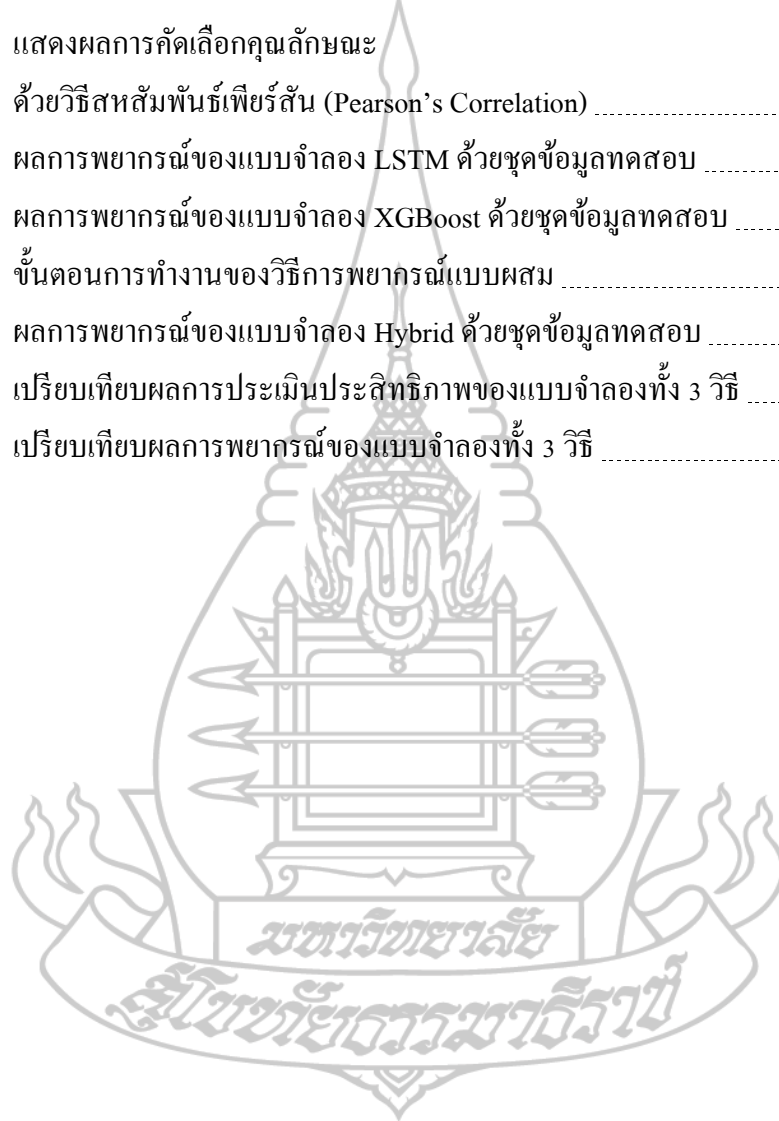
ตารางที่ 4.2	ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง LSTM.....	62
ตารางที่ 4.3	ผลการประเมินประสิทธิภาพของแบบจำลองโดยอาศัยวิธีการ หน่วยความจำระยะสั้นแบบยาว.....	63
ตารางที่ 4.4	ไฮเปอร์พารามิเตอร์ที่ใช้กับแบบจำลองเอ็กซ์ตรีมเกรเดียนต์บูตส์ตริง.....	64
ตารางที่ 4.5	ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง XGBoost.....	65
ตารางที่ 4.6	ผลการประเมินประสิทธิภาพของแบบจำลองโดยอาศัยวิธีการ เอ็กซ์ตรีมเกรเดียนต์บูตส์ตริง.....	66
ตารางที่ 4.7	ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง Hybrid.....	68
ตารางที่ 4.8	ผลการประเมินประสิทธิภาพแบบจำลอง โดยอาศัยวิธีการพยากรณ์แบบผสม.....	69
ตารางที่ 4.9	ตารางเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 3 วิธี.....	70
ตารางที่ 5.1	เปรียบเทียบผลการพยากรณ์และค่าพลังงานแสงอาทิตย์ ในแบบจำลองแต่ละแบบ.....	73
ตารางที่ 5.2	ตารางแสดงประสิทธิภาพประสิทธิภาพของแบบจำลองทั้งหมด.....	74



## สารบัญรูปภาพ

	หน้า
ภาพที่ 1.1 กรอบแนวคิดการวิจัย .....	4
ภาพที่ 2.1 โครงสร้างการทำงานของระบบพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ .....	9
ภาพที่ 2.2 แบบจำลองการพยากรณ์บนพื้นฐานของระยะทางการเชื่อมโยงข้อมูล และระยะเวลา .....	11
ภาพที่ 2.3 โครงสร้างหน่วยความจำระยะสั้นแบบยาว .....	13
ภาพที่ 2.4 ส่วนประกอบหน่วยความจำระยะสั้นและหน่วยความจำระยะยาวของ LSTM .....	14
ภาพที่ 2.5 ประติรูป Forget Gate Input Gate และ Output Gate ของ LSTM .....	14
ภาพที่ 2.6 ลักษณะการทำงานของ Forget Gate .....	15
ภาพที่ 2.7 ลักษณะการทำงานของ Input Gate .....	16
ภาพที่ 2.8 ลักษณะการทำงานของ Update Cell State .....	16
ภาพที่ 2.9 ลักษณะการทำงานของ Output Gate .....	17
ภาพที่ 2.10 โครงสร้างและหลักการการทำงานของ XGBoost .....	19
ภาพที่ 2.11 ความแตกต่างระหว่างวิธีการพยากรณ์แบบผสมและวิธีการพยากรณ์แบบรวม .....	20
ภาพที่ 2.12 Flow Chart วิธีการพยากรณ์แบบผสม .....	21
ภาพที่ 2.13 การแบ่งข้อมูลรูปแบบ Sliding Window .....	24
ภาพที่ 2.14 ตัวอย่างชุดข้อมูลที่ได้รับการแบ่งด้วยวิธี Sliding Window .....	24
ภาพที่ 2.15 ตัวอย่างการแบ่งข้อมูลประเมินประสิทธิภาพแบบจำลอง .....	25
ภาพที่ 2.16 ตัวอย่างผลการคำนวณด้วยวิธีการวัดค่าความเคลื่อนด้วยค่าสมมติ .....	26
ภาพที่ 3.1 ปริมาณค่ากำลังไฟฟ้าที่ผลิตได้จากพลังงานแสงอาทิตย์ .....	34
ภาพที่ 3.2 ค่ากำลังไฟฟ้าพลังงานแสงอาทิตย์เฉลี่ยแบ่งกลุ่มตามวัน สัปดาห์ และเดือน .....	35
ภาพที่ 3.3 ขั้นตอนการวิจัย .....	36
ภาพที่ 3.4 ตัวอย่างชุดข้อมูลในรูปแบบไฟล์ CSV .....	37
ภาพที่ 3.5 ตัวอย่างการอ่านชุดข้อมูล .....	39
ภาพที่ 3.6 ตัวอย่างข้อมูลที่ทำกรแปลง .....	40
ภาพที่ 3.7 ผลการหาค่าความสัมพันธ์ของข้อมูล .....	42
ภาพที่ 3.8 ผลการหาเลือกคุณลักษณะที่ต้องการนำไปใช้งาน .....	43
ภาพที่ 3.9 การแสดงผลภาพ Heatmap ของข้อมูลที่เลือกใช้งาน .....	44
ภาพที่ 3.10 ผลการแบ่งข้อมูลสำหรับสร้างแบบจำลอง .....	48

ภาพที่ 3.11	ผลการประเมินประสิทธิภาพแบบจำลอง LSTM ด้วยชุดข้อมูลทดสอบ	53
ภาพที่ 3.12	ผลการประเมินประสิทธิภาพแบบจำลอง XGBoost ด้วยชุดข้อมูลทดสอบ	53
ภาพที่ 3.13	ขั้นตอนการทำงานของแบบจำลองขั้นสุดท้ายด้วยวิธีการพยากรณ์แบบผสม	54
ภาพที่ 3.14	ผลการประเมินประสิทธิภาพของแบบจำลองวิธีการพยากรณ์แบบผสม	57
ภาพที่ 4.1	แสดงผลการคัดเลือกคุณลักษณะ ด้วยวิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation)	60
ภาพที่ 4.2	ผลการพยากรณ์ของแบบจำลอง LSTM ด้วยชุดข้อมูลทดสอบ	63
ภาพที่ 4.3	ผลการพยากรณ์ของแบบจำลอง XGBoost ด้วยชุดข้อมูลทดสอบ	66
ภาพที่ 4.4	ขั้นตอนการทำงานของวิธีการพยากรณ์แบบผสม	67
ภาพที่ 4.5	ผลการพยากรณ์ของแบบจำลอง Hybrid ด้วยชุดข้อมูลทดสอบ	62
ภาพที่ 4.5	เปรียบเทียบผลการประเมินประสิทธิภาพของแบบจำลองทั้ง 3 วิธี	69
ภาพที่ 4.6	เปรียบเทียบผลการพยากรณ์ของแบบจำลองทั้ง 3 วิธี	70



# บทที่ 1

## บทนำ

### 1. ความเป็นมาและความสำคัญของปัญหา

ประเทศไทยใช้ก๊าซธรรมชาติเป็นเชื้อเพลิงหลักในการผลิตพลังงานไฟฟ้า รองลงมาคือถ่านหิน ซึ่งเชื้อเพลิงจากฟอสซิล เป็นพลังงานที่มีอยู่อย่างจำกัด หากใช้อย่างต่อเนื่อง อาจไม่เพียงพอต่อความต้องการที่กำลังเพิ่มขึ้นในอนาคต แม้ว่าการใช้เชื้อเพลิงฟอสซิลยังคงเป็นที่ต้องการในปัจจุบัน แต่กลับส่งผลต่อการเปลี่ยนแปลงสภาพภูมิอากาศ เกิดเป็นสภาวะโลกร้อน จากการปล่อยก๊าซเรือนกระจกเข้าสู่ชั้นบรรยากาศ ในปี พ.ศ. 2564 ประเทศไทยได้แสดงเจตนารมณ์ในการมีส่วนร่วมลดการปล่อยก๊าซเรือนกระจก ต่อที่ประชุมรัฐภาคีสมาชิกของอนุสัญญาสหประชาชาติว่าด้วยการเปลี่ยนแปลงสภาพภูมิอากาศ หรือ COP26 เพื่อยกระดับการแก้ปัญหาภูมิอากาศ โดยมีเป้าหมายความเป็นกลางทางคาร์บอน (Carbon Neutrality) ภายในปี พ.ศ. 2593 การปล่อยก๊าซเรือนกระจกสุทธิเป็นศูนย์ (Net Zero Emissions) ในปี พ.ศ. 2608 และยกระดับเป้าหมายการลดก๊าซเรือนกระจก (Nationally Determined Contribution) ภายในปี พ.ศ. 2573 มีการจัดตั้งโมเดลเศรษฐกิจสู่การพัฒนาที่ยั่งยืน (Bio-Circular-Green Economy: BCG) ปี พ.ศ. 2562 เพื่อส่งเสริมเศรษฐกิจหมุนเวียน สนับสนุนการใช้พลังงานทดแทนและพลังงานทางเลือก ในปัจจุบันประเทศไทยมีการสนับสนุนการใช้พลังงานหมุนเวียน (Renewable Energy) ในการผลิตพลังงานไฟฟ้า เช่น พลังงานน้ำ พลังงานแสงอาทิตย์ พลังงานลม พลังงานความร้อนใต้พิภพ และพลังงานชีวมวล เป็นต้น (กรมพัฒนาพลังงานทดแทนและอนุรักษ์พลังงาน กระทรวงพลังงาน, 2564)

จากศักยภาพพื้นที่ของประเทศไทย ที่มีปริมาณความเข้มแสงอาทิตย์สูงเพียงพอ ทำให้การก่อสร้างและพัฒนาโรงไฟฟ้าพลังงานหมุนเวียนด้วยพลังงานแสงอาทิตย์เติบโตอย่างรวดเร็ว สอดคล้องกับแผนพัฒนาพลังงานทดแทนและพลังงานทางเลือก พ.ศ. 2561 – 2580 (AEDP2018) และแผนพัฒนากำลังผลิตไฟฟ้าของประเทศไทย พ.ศ. 2561 – 2580 ฉบับปรับปรุงครั้งที่ 1 (PDP2018 Revision 1) ที่มุ่งเน้นการพัฒนาที่ยั่งยืน สนับสนุนการผลิตและการใช้พลังงานทดแทนตามศักยภาพของแหล่งเชื้อเพลิงในพื้นที่ โดยมีเป้าหมายการผลิตพลังงานไฟฟ้าจากพลังงานแสงอาทิตย์ประเภทติดตั้งบนพื้นดิน (Solar Farm) ติดตั้งบนหลังคา (Solar Rooftop) และติดตั้งบน

ทุ่นลอยน้ำ (Floating Solar) รวมทั้งหมด 14,864 เมกะวัตต์ ภายในปี พ.ศ. 2580 (สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน, 2561)

การผลิตไฟฟ้าจากพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ หรือ โซลาร์เซลล์ลอยน้ำ (Floating Solar) เป็นแผงโซลาร์เซลล์ที่ติดตั้งบนโครงสร้างที่ลอยอยู่บนแหล่งน้ำ โดยทั่วไปจะเป็นอ่างเก็บน้ำหรือทะเลสาบ โรงไฟฟ้าประเภทนี้มีจำนวนเพิ่มมากขึ้นในประเทศจีน ญี่ปุ่น สหรัฐอเมริกา ฝรั่งเศส อินเดีย และเกาหลีใต้ เป็นต้น ถือเป็นนวัตกรรมที่หลายประเทศในยุโรปและภูมิภาคเอเชียนำมาใช้ในการผลิตไฟฟ้า (Cazzaniga & Rosa-Clot, 2021) ด้วยข้อดีในการลดการระเหยของน้ำและการที่มีโครงสร้างแบบลอยตัวบนผิวน้ำ สามารถช่วยระบบระบายความร้อนและเพิ่มประสิทธิภาพของแผงโซลาร์เซลล์ได้ดีขึ้น สำหรับประเทศไทย ได้มีการก่อสร้างและพัฒนาโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ บนพื้นที่อ่างเก็บน้ำในพื้นที่ภาคตะวันออกเฉียงเหนือ ขนาดกำลังการผลิต 45 เมกะวัตต์ สามารถผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ ช่วยเสริมความมั่นคงทางพลังงานไฟฟ้าให้กับประเทศได้อีกทางหนึ่ง แต่ข้อจำกัดที่สำคัญของโรงไฟฟ้าพลังงานแสงอาทิตย์ คือ ค่ากำลังไฟฟ้าที่ผลิตได้มีความไม่แน่นอน เนื่องจากปัจจัยด้านภูมิอากาศในช่วงเวลานั้น ๆ มีความผันผวนไม่คงที่ เช่น ความเข้มแสงอาทิตย์ อุณหภูมิอากาศ อุณหภูมิบนแผง ความชื้น ความเร็วลม เป็นต้น ล้วนเป็นปัจจัยที่ไม่สามารถควบคุมและหลีกเลี่ยงได้ ส่งผลกระทบต่อคุณภาพและเสถียรภาพของระบบไฟฟ้า จึงจำเป็นต้องปรับปรุงความแม่นยำในการพยากรณ์ล่วงหน้าแบบวันต่อวัน การมีระบบพยากรณ์ที่เชื่อถือได้จะช่วยให้การวางแผนบริหารจัดการกำลังไฟฟ้ามีประสิทธิภาพดีขึ้น

เพื่อเพิ่มเสถียรภาพให้กับการบริหารจัดการกำลังไฟฟ้าและลดข้อจำกัดด้านภูมิอากาศ จึงได้มีการนำเสนอวิธีการพยากรณ์พลังงานแสงอาทิตย์ด้วยวิธีการทางปัญญาประดิษฐ์ (Artificial Intelligence Method) ผ่านวิธีการเรียนรู้ของเครื่อง (Machine Learning) เช่น การใช้โครงข่ายประสาทเทียม (Artificial Neuron Network: ANN) งานวิจัยของ (ภาณุพงศ์ น้ำแก้ว, 2563) นำเสนอโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Network) โดยมีรูปแบบการทำงานแบบป้อนไปข้างหน้า (Feed Forward) มีการเรียนรู้แบบแพร่ย้อนกลับ (Back Propagation) มาสร้างแบบจำลองการพยากรณ์กำลังไฟฟ้าของระบบเซลล์แสงอาทิตย์ในวันถัดไป โดยใช้ข้อมูลป้อนเข้าจากอุปกรณ์วัดที่ติดตั้งบนหลังคาที่พักอาศัย ต่อมาได้มีการประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึก (Deep Learning) เช่น โครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent Neural Network: RNN) หน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) และ โครงข่ายประตูกกกลับ (Gated Recurrent Unit: GRU) สำหรับใช้จัดการข้อมูลที่มีลักษณะเป็นข้อมูลที่มีลำดับหรืออนุกรมเวลา งานวิจัยของ (Jebli et al., 2021) ได้ทำการเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 3

แบบจำลองในการพยากรณ์พลังงานแสงอาทิตย์ พบว่า แบบจำลอง RNN และ LSTM มีความเหมาะสมสำหรับการพยากรณ์ เนื่องจากความสามารถในการจัดการข้อมูลที่เป็นลำดับและปัญหาการถดถอยอนุกรมเวลา เมื่อเทียบกับแบบจำลอง Gated Recurrent Units (GRU) ที่ให้ประสิทธิภาพน้อยกว่าเล็กน้อย นอกจากนี้ยังมีการนำเสนอแบบจำลอง Convolutional Neural Network (CNN) มาใช้พยากรณ์ผสมกับแบบจำลอง LSTM เพื่อเพิ่มประสิทธิภาพในการพยากรณ์ให้กับข้อมูล เช่น งานวิจัยของ (Lim et al., 2022) ใช้แบบจำลอง CNN ในการจำแนกสภาพอากาศ ผสมกับแบบจำลอง LSTM ในการพยากรณ์กำลังไฟฟ้าตามสภาพอากาศ เพิ่มความแม่นยำในการพยากรณ์ให้สูงขึ้น

ดังนั้นในงานวิจัยนี้จึงได้นำเสนอแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) และเอ็กซ์ตรีมเกรเดียนต์บูสต์ติง (Extreme Gradient Boosting: XGBoost) ที่มีความสามารถในการจัดการกับข้อมูลที่ไม่เป็น ไม่เชิงเส้นและแยกแยะลักษณะความผันผวนของข้อมูลอนุกรมเวลาได้ดี และเพื่อให้ได้ประโยชน์สูงสุดจากแบบจำลองทั้งสองจึงได้นำเสนอวิธีการพยากรณ์แบบผสมเพื่อปรับปรุงคุณภาพและความแม่นยำในการพยากรณ์โดยรวมให้ดีขึ้น ทำให้การวางแผนการผลิตไฟฟ้าจากพลังงานแสงอาทิตย์มีประสิทธิภาพมากขึ้น และสามารถนำไปใช้ในการวางแผนและบริหารจัดการการผลิตไฟฟ้าเพื่อรองรับการเติบโตของโรงไฟฟ้าพลังงานแสงอาทิตย์ที่กำลังจะเกิดขึ้นต่อไปในอนาคต

## 2. วัตถุประสงค์การศึกษา

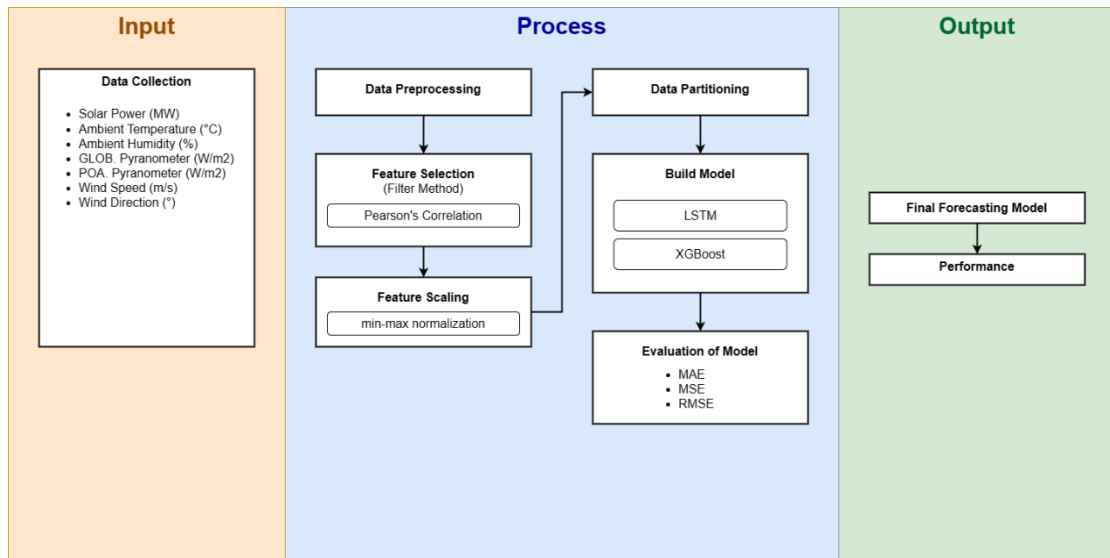
2.1. เพื่อศึกษาวิธีการและสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสต์ติง

2.2. เพื่อประเมินประสิทธิภาพของแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสต์ติง

## 3. กรอบแนวคิดการวิจัย

งานวิจัยนี้ มุ่งเน้นการสร้างแบบจำลองการเรียนรู้เชิงลึก (Deep Learning) เพื่อพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสต์ติง โดยนำชุดข้อมูลการผลิตพลังงานไฟฟ้าและชุดข้อมูลสภาพแวดล้อมจาก

อุปกรณ์เซนเซอร์ ของ โรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ ดังภาพที่ 1.1



ภาพที่ 1.1 กรอบแนวคิดการวิจัย

จากภาพที่ 1.1 กรอบแนวคิดการวิจัย โดยใช้ข้อมูลของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ ตั้งแต่เดือนกุมภาพันธ์ – เดือนตุลาคม พ.ศ. 2566 จำนวน 6,551 ระเบียบวิน และ 11 คุณลักษณะ นำมาเตรียมข้อมูล ได้แก่ การเลือกลักษณะเฉพาะ (Feature Selection) เป็นการเลือกลักษณะเฉพาะของข้อมูลที่มีความสัมพันธ์กันและเหมาะสมกับการสร้างแบบจำลอง เมื่อเลือกคุณลักษณะที่เหมาะสมแล้ว ข้อมูลจะถูกปรับขนาดให้เป็นมาตรฐาน (Normalization) ด้วยค่าต่ำสุด-สูงสุด จากนั้นชุดข้อมูลจะถูกแบ่งเป็นชุดสำหรับฝึกสอน ชุดสำหรับตรวจสอบ และชุดสำหรับทดสอบ แล้วจึงนำข้อมูลเข้าสู่แบบจำลองที่สร้าง โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) และเอ็กซ์ตรีมเกรเดียนบูสติง (Extreme Gradient Boosting: XGBoost) เพื่อให้ทราบผลลัพธ์ของแบบจำลอง และทำการประเมินประสิทธิภาพของแบบจำลองแต่ละแบบ จากนั้นใช้วิธีการพยากรณ์แบบผสม (Hybrid Forecasting Method) เพิ่มประสิทธิภาพให้กับแบบจำลองสำหรับการพยากรณ์ เพื่อให้ได้ค่าประสิทธิภาพของแบบจำลองที่ดีที่สุด สำหรับนำไปประยุกต์ใช้ต่อไปได้



#### 4. ขอบเขตของการวิจัย

การสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตสตีง กำหนดขอบเขตการศึกษาไว้ ดังต่อไปนี้

##### 4.1. ขอบเขตข้อมูล

###### 4.1.1. ข้อมูลกำลังไฟฟ้าที่ผลิตได้

###### 4.1.2. ข้อมูลสภาพอากาศจากเซนเซอร์ที่ติดตั้งบนระบบเซลล์แสงอาทิตย์

##### 4.2. ขอบเขตพื้นที่

4.2.1. พื้นที่โรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำแห่งหนึ่ง ขนาดกำลังผลิต 45 เมกะวัตต์

##### 4.3. ขอบเขตเวลา

4.3.1. ข้อมูลการผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ตั้งแต่เดือนกุมภาพันธ์ ถึง เดือนตุลาคม พ.ศ. 2566

#### 5. นิยามศัพท์เฉพาะ

5.1. พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ (Floating Solar) คือ การติดตั้งระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วยทุ่นพลาสติกบนพื้นที่ผิวน้ำเพื่อเป็นฐานให้กับแผงโซลาร์เซลล์ โดยวางไว้บนโครงสร้างที่ลอยอยู่ในพื้นที่อ่างเก็บน้ำ เพื่อให้แผงโซลาร์เซลล์อยู่เหนือผิวน้ำ พลังงานไฟฟ้าที่ผลิตจะถูกส่งผ่านสายไฟฟ้าที่อยู่น้ำส่งไปยังระบบควบคุมที่ตั้งอยู่บนฝั่ง เช่นเดียวกับระบบการผลิตพลังงานแสงอาทิตย์ทั่วไป

5.2. ข้อมูลอนุกรมเวลา (Time Series Data) คือ ข้อมูลที่เก็บรวบรวมในช่วงเวลาต่าง ๆ ถูกประยุกต์ใช้งานอย่างแพร่หลายในงานด้านต่าง ๆ เช่น งานด้านการเงิน ใช้ในการวิเคราะห์แนวโน้มในการลงทุนในตลาดหุ้น ค่าเงิน รวมถึงเป็นข้อมูลสำคัญสำหรับการสร้างแบบจำลองสภาพภูมิอากาศ การพยากรณ์อากาศ การพยากรณ์น้ำฝน เป็นต้น

5.3. การพยากรณ์พลังงานไฟฟ้าจากพลังงานหมุนเวียน (Renewable Energy Forecast) คือ ระบบพยากรณ์พลังงานไฟฟ้าที่ผลิตได้จากโรงไฟฟ้าพลังงานหมุนเวียน โดยอาศัยข้อมูลจากเซนเซอร์ที่ติดตั้งและข้อมูลการผลิตไฟฟ้า โดยทั่วไปจะเน้นศึกษา การพยากรณ์พลังงานแสงอาทิตย์ และการพยากรณ์พลังงานลม เนื่องจากเป็นพลังงานที่มีความไม่แน่นอนสูง

5.4. อัลกอริทึมเอ็กซ์ตรีมเกรเดียนบูสตีง (Extreme Gradient Boosting: XGBoost) คือเทคนิคการเรียนรู้ของเครื่องที่ใช้กันอย่างแพร่หลายรวมถึงการพยากรณ์ข้อมูลอนุกรมเวลา XGBoost ถูกพัฒนามาจากเทคนิค Gradient Boosting ด้วยการนำต้นไม้การตัดสินใจ (Decision Tree) มาทำการฝึกสอนต่อกันหลาย ๆ ต้น โดยที่ต้นไม้ตัดสินใจจะเรียนรู้จากค่าความผิดพลาดของกระบวนการก่อนหน้า จนทำให้การพยากรณ์มีความแม่นยำขึ้นเรื่อย ๆ

5.5. หน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) คือโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลลำดับ (Sequence) ถูกพัฒนาต่อขยายมาจากโครงข่ายประสาทวนซ้ำ (RNN) เพื่อแก้ปัญหาค่าการลดลงของเกรเดียนต์เนื่องจากความยาวของข้อมูลมีมากเกินไป ด้วยการเพิ่มหน่วยความจำระยะยาว คำนวณร่วมกับหน่วยความจำระยะสั้น สามารถเก็บความสัมพันธ์ของข้อมูลที่มีความยาวได้ดี

5.6. วิธีการพยากรณ์แบบผสม (Hybrid Forecasting Method) คือ การนำวิธีพยากรณ์มารวมกันเพื่อสร้างแบบจำลองการพยากรณ์ร่วมกัน โดยใช้แบบจำลองการพยากรณ์มากกว่าหนึ่งแบบจำลอง แบบจำลองแรกเรียกว่าแบบจำลองหลักถูกคำนวณเพื่อหาค่าส่วนเหลือ (Residual) แล้วนำค่าส่วนเหลือไปใช้เป็นข้อมูลนำเข้าของแบบจำลองการพยากรณ์รอง จากนั้นนำค่าพยากรณ์จากทั้ง 2 แบบจำลองมารวมกัน เพื่อให้ได้ผลลัพธ์ในการพยากรณ์

## 6. ประโยชน์ที่คาดว่าจะได้รับ

6.1. มีแบบจำลองสำหรับการพยากรณ์พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ที่สามารถนำไปประยุกต์ใช้งานต่อไปได้

6.2. เพิ่มประสิทธิภาพการพยากรณ์การผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ที่ติดตั้งบนทุ่นลอยน้ำให้มีความแม่นยำ ผ่านการเรียนรู้แบบจำลองการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกที่ให้ค่าประสิทธิภาพที่ดีที่สุด

## บทที่ 2

### วรรณกรรมที่เกี่ยวข้อง

การสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูสตีง ผู้วิจัยได้ค้นคว้าและศึกษางานวิจัยที่เกี่ยวข้อง ดังนี้

1. พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ (Floating Solar)
2. การพยากรณ์พลังงานแสงอาทิตย์ (Solar Power Forecasting)
3. หน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory)
4. เอ็กซ์ตรีมเกรเดียนต์บูสตีง (Extreme Gradient Boosting)
5. วิธีการพยากรณ์แบบผสม (Hybrid Forecasting Method)
6. การคัดเลือกคุณลักษณะ (Feature Selection)
7. การปรับขนาดข้อมูลให้สมมาตร (Scaling Data)
8. การแบ่งข้อมูลเพื่อประเมินประสิทธิภาพแบบจำลอง (Data Partition)
9. การประเมินประสิทธิภาพแบบจำลอง (Performance Evaluation)
10. งานวิจัยที่เกี่ยวข้อง

#### 1. พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ (Floating Solar)

พลังงานแสงอาทิตย์บนทุ่นลอยน้ำ (Solar Floating) หรือเซลล์แสงอาทิตย์แบบลอยตัว (FPV) บางครั้งอาจเรียกว่า floatovoltaics เป็นแผงโซลาร์เซลล์ที่ติดตั้งบนโครงสร้างที่ลอยอยู่บนแหล่งน้ำ เช่น ทะเลสาบ อ่างเก็บน้ำ บ่ออุตสาหกรรม และพื้นที่ใกล้ชายฝั่ง เป็นต้น เป็นหนึ่งในเทคโนโลยีการผลิตไฟฟ้าที่เติบโตเร็วที่สุดในปัจจุบัน เปิดโลกทัศน์ใหม่ในการขยายขนาดพลังงานแสงอาทิตย์ทั่วโลก โดยเฉพาะอย่างยิ่งในประเทศที่มีข้อจำกัดด้านที่ดิน (World Bank Group, 2018)

##### 1.1. ส่วนประกอบพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ มีส่วนประกอบที่สำคัญ ดังนี้

**1.1.1. แผงเซลล์แสงอาทิตย์ (PV Module/Panel)** คือ แผงเซลล์ที่ผลิตจากสารกึ่งตัวนำ (Semiconductor) ทำหน้าที่แปลงพลังงานแสงอาทิตย์เป็นพลังงานไฟฟ้า พลังงานไฟฟ้าที่ผลิต

ได้ จะเป็นไฟฟ้ากระแสตรง (DC) โดยที่แรงดันและกระแสไฟฟ้าที่ผลิตได้จะขึ้นอยู่กับความเข้มของรังสีจากดวงอาทิตย์ (Irradiation) และอุณหภูมิบนแผงเซลล์แสงอาทิตย์

**1.1.2. อินเวอร์เตอร์ (Inverter)** คือ อุปกรณ์ที่ทำหน้าที่ในการแปลงพลังงานไฟฟ้ากระแสตรง (DC) จากเซลล์แสงอาทิตย์ ไปเป็นพลังงานไฟฟ้ากระแสสลับ (AC) สามารถติดตั้งบนทุ่นร่วมกับแผงเซลล์แสงอาทิตย์หรือบนฝั่งตามความเหมาะสม

**1.1.3. อุปกรณ์ลอยตัว (Floating Structure)** คือ อุปกรณ์ที่ลอยบนผิวน้ำเพื่อใช้ติดตั้งแผงเซลล์แสงอาทิตย์ด้านบน

**1.1.4. ระบบทุ่นลอยน้ำ (Anchoring and Mooring System)** คือ ระบบที่ทำหน้าที่ยึดแผงเซลล์แสงอาทิตย์ทั้งหมดไม่ให้เคลื่อนที่ออกจากบริเวณพื้นที่ผิวน้ำที่กำหนด โดยใช้วิธีการยึดโยง แบ่งออกเป็น 2 ประเภท ได้แก่

1) แบบยึดติดกับชายฝั่ง (Bank Anchoring) เหมาะสำหรับพื้นที่แหล่งน้ำขนาดเล็ก ตื้น และพื้นที่แหล่งน้ำไม่สามารถทำการยึดโยงได้ การยึดโยงประเภทนี้ เป็นตัวเลือกที่คุ้มค่าในด้านของราคา ทั้งนี้ ขึ้นอยู่กับสภาพความเหมาะสมของพื้นที่ชายฝั่ง และอาจมีผลต่อทัศนียภาพชายฝั่งได้

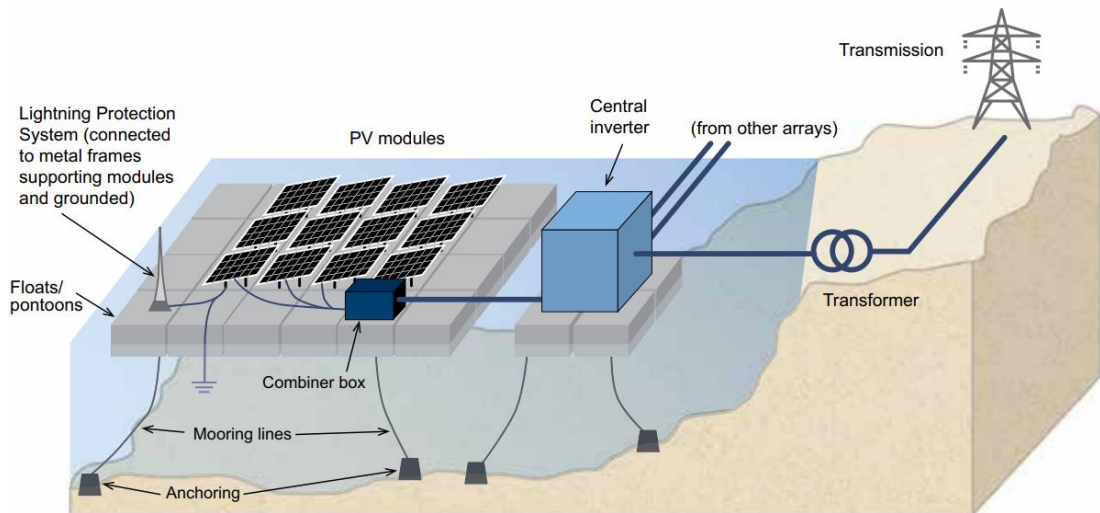
2) แบบยึดกับพื้นใต้แหล่งน้ำ (Bottom Anchoring) การยึดโยงประเภทนี้มีค่าใช้จ่ายในการติดตั้งสูง เนื่องจากต้องออกแบบและวางแผนอย่างละเอียด แต่ได้รับความนิยมมากที่สุด สามารถทำได้ 2 วิธี ได้แก่ การแทรกพุกเข้าที่พื้นใต้แหล่งน้ำโดยตรง หรือยึดผ่านบล็อกคอนกรีตที่วางไว้ที่พื้นใต้แหล่งน้ำ

**1.1.5. สายเคเบิล (Cables)** คือ สายยึดทำหน้าที่เชื่อมระหว่างระบบทุ่นลอยน้ำกับอุปกรณ์ลอยตัว 2 ระบบ ได้แก่ระบบ Supervisory Control And Data Acquisition (SCADA) หรือระบบ Monitoring โดยเป็นระบบคอมพิวเตอร์ที่คอยเก็บข้อมูลและใช้ควบคุมการปฏิบัติการทั้งหมด

**1.1.6. หม้อแปลงไฟฟ้า (Transformer)** คือ อุปกรณ์ที่ทำหน้าที่แปลงกระแสไฟฟ้าและแรงดันให้อยู่ในค่าที่เหมาะสมและนำไปใช้งานต่อได้

**1.2. หลักการทำงาน เซลล์แสงอาทิตย์ที่ใช้ในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ** จะถูกติดตั้งบนพื้นผิวน้ำโดยการวางไว้บน โครงสร้างที่ลอยอยู่ในน้ำ เพื่อให้แผงเซลล์ที่อยู่เหนือผิวน้ำ ผลิตพลังงานไฟฟ้ากระแสตรง (DC) และส่งผ่านสายไฟฟ้าไปยังชุด Inverter เพื่อแปลงกระแสไฟฟ้าเป็นกระแสสลับ (AC) จากนั้นจะถูกส่งไปยังหม้อแปลงแรงดันเพื่อแปลงแรงดันไฟฟ้าให้อยู่ในระดับที่เหมาะสม โดยกระแสและแรงดันไฟฟ้าที่ผลิตได้ จะถูกควบคุมด้วยระบบควบคุมไฟฟ้าที่ติดตั้งอยู่ที่ฝั่งก่อนที่จะนำไปใช้งานผ่านระบบส่งจ่ายกำลังไฟฟ้า

(Transmission) ตามปกติ เช่นเดียวกับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ทั่วไปที่ติดตั้งบนบก โครงสร้างการทำงานของระบบพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ แสดงดังภาพที่ 2.1



ภาพที่ 2.1 โครงสร้างการทำงานของระบบพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ

ที่มา: World Bank Group (2018)

### 1.3. ข้อดี

**1.3.1.** ลดการระเหยของน้ำในอ่างเก็บน้ำ เนื่องจากระบบแผงเซลล์แสงอาทิตย์ ถูกติดตั้งบนผิวน้ำ และมีพื้นที่ครอบคลุมบางส่วนของอ่างเก็บน้ำ จึงสามารถลดการระเหยของน้ำได้ โดยผลลัพธ์จะขึ้นอยู่กับสภาพภูมิอากาศและสัดส่วนของพื้นที่ผิวที่ปกคลุม อีกทั้งยังมีส่วนช่วยปรับปรุงคุณภาพ โดยลดการเจริญเติบโตของสาหร่ายได้แหล่งน้ำ

**1.3.2.** เพิ่มศักยภาพให้กับแผงเซลล์แสงอาทิตย์ ความเย็นของน้ำ จะช่วยลดอุณหภูมิให้กับแผงเซลล์แสงอาทิตย์ได้ดี โดยพบว่าประสิทธิภาพการระบายความร้อนของแผงเซลล์แสงอาทิตย์ที่อยู่ใกล้กับน้ำ ช่วยเพิ่มประสิทธิภาพให้กับแผงเซลล์แสงอาทิตย์ได้ร้อยละ 5 - 15

### 1.4. ข้อเสีย

**1.4.1.** ต้องเลือกวัสดุที่ทนต่อการเกิดสนิมและกัดกร่อนมากกว่าแบบปกติ การทำงานบนน้ำตลอดอายุการใช้งาน ระบบจะต้องมีความทนทานต่อการกัดกร่อนเพิ่มขึ้นอย่างมีนัยสำคัญและความสามารถในการลอยตัวในระยะยาว โดยเฉพาะอย่างยิ่งเมื่อถูกติดตั้งบนแหล่งน้ำที่มีความเค็ม

1.4.2. การซ่อมบำรุงค่อนข้างลำบาก เนื่องจากระบบถูกติดตั้งอยู่บนแหล่งน้ำ ส่งผลต่อการบำรุงรักษาแผงเซลล์แสงอาทิตย์ รวมถึงอุปกรณ์ประกอบอื่น ๆ การเดินทางไปยังจุดติดตั้งต้องอาศัยการเดินทางทางน้ำ และต้องใช้ทักษะ ความเชี่ยวชาญ ในการบำรุงรักษา จึงยากต่อการดำเนินการกับระบบที่ติดตั้งบนน้ำมากกว่าระบบที่ติดตั้งบนบก

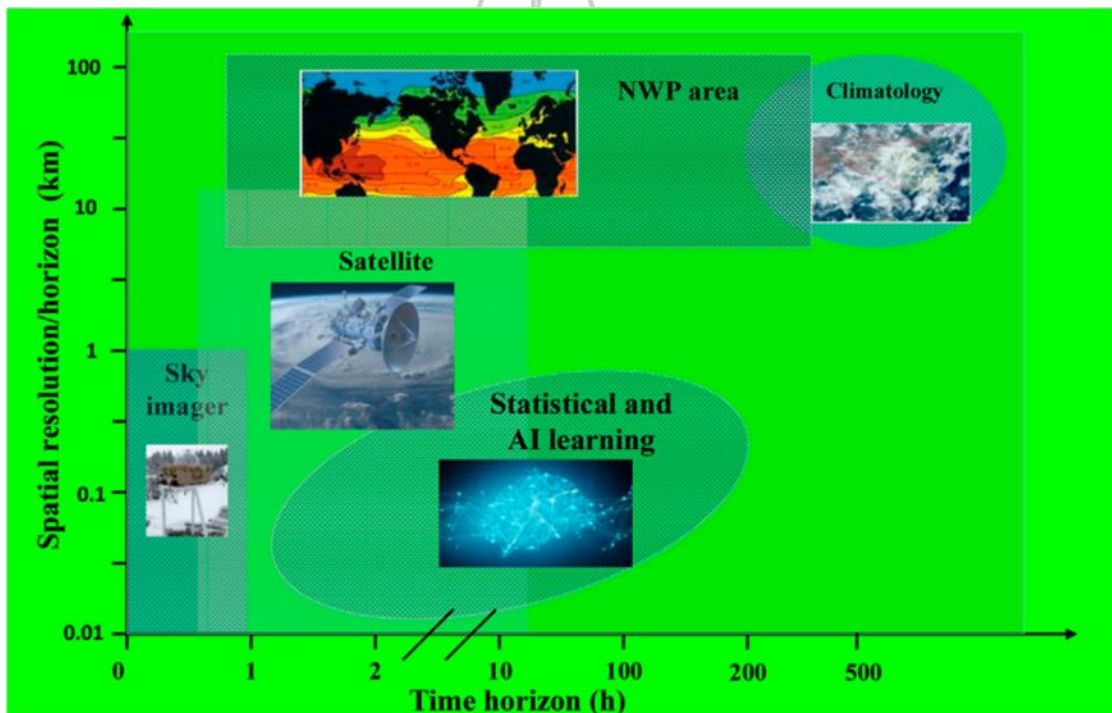
## 2. การพยากรณ์พลังงานแสงอาทิตย์ (Solar Power Forecasting)

2.1. เทคนิคการพยากรณ์ วิธีการพยากรณ์กำลังไฟฟ้าที่ผลิตได้จากพลังงานแสงอาทิตย์ แบ่งการพยากรณ์ออกเป็น 2 วิธี คือ 1) การพยากรณ์ทางตรง โดยใช้ข้อมูลกำลังไฟฟ้าที่ผลิตได้ในอดีตมาใช้ในการพยากรณ์ และ 2) การพยากรณ์ทางอ้อม โดยใช้ค่าความเข้มแสงอาทิตย์มาใช้ในการพยากรณ์ร่วมกับค่าความสัมพันธ์ของตัวแปรอื่น ๆ ในระบบการผลิต (Ye et al., 2022) เทคนิคการพยากรณ์ที่นิยมใช้ในปัจจุบัน สามารถแบ่งออกเป็น 3 ประเภท แสดงดังตารางที่ 2.1

ตารางที่ 2.1 เทคนิคการพยากรณ์และคุณลักษณะเฉพาะ

เทคนิคการพยากรณ์	คำนิยาม	คุณลักษณะเฉพาะ
การพยากรณ์ทางกายภาพ (Physical prediction method)	การพยากรณ์กำลังไฟฟ้าด้วยแบบจำลองการคำนวณกำลังไฟฟ้า โดยยึดตามมุมความสูงของแสงอาทิตย์ที่ตั้งทางภูมิศาสตร์ อุณหภูมิ และการฉายรังสีแสงอาทิตย์	แบบจำลองที่มีความซับซ้อนขึ้นอยู่กับความเหมาะสมของพารามิเตอร์
การพยากรณ์ทางสถิติ (Statistical prediction method)	การพยากรณ์กำลังไฟฟ้า โดยขึ้นอยู่กับค่าความสัมพันธ์ทางสถิติระหว่างข้อมูลน้ำเข้าและข้อมูลส่งออกของแบบจำลองการพยากรณ์	ไม่ต้องการข้อมูลทางกายภาพของโรงไฟฟ้าพลังงานแสงอาทิตย์
การพยากรณ์ด้วยปัญญาประดิษฐ์ (AI prediction method)	ฝึกสอนแบบจำลองการพยากรณ์ด้วยแบบจำลองทางปัญญาประดิษฐ์ โดยใช้ชุดข้อมูลตัวอย่าง	มีความสามารถในการเรียนรู้ด้วยตนเองและการปรับตัวที่แข็งแกร่ง แต่ต้องการข้อมูลในอดีตปริมาณมาก

2.2. รูปแบบการพยากรณ์ เทคนิคการพยากรณ์พลังงานแสงอาทิตย์ ต้องอาศัยปัจจัยข้อมูลนำเข้าเพื่อพยากรณ์กำลังไฟฟ้าให้มีความแม่นยำ เมื่อพิจารณาถึงคุณลักษณะเฉพาะของเทคนิคการพยากรณ์ที่กล่าวมาข้างต้น พบว่าการได้มาซึ่งข้อมูลทางกายภาพที่แม่นยำนั้นเป็นไปได้ยาก รวมถึงการกำหนดพารามิเตอร์ให้มีความเหมาะสมกับแบบจำลอง ขณะเดียวกันเทคนิคการพยากรณ์ทางสถิติ แม้จะไม่ต้องการข้อมูลตำแหน่งทางภูมิศาสตร์ แต่ต้องอาศัยข้อมูลในอดีตเป็นจำนวนมากเพื่อใช้ในการพยากรณ์ทางสถิติ การใช้เทคนิคทางปัญญาประดิษฐ์จึงมีความเหมาะสมและได้รับความนิยมในปัจจุบัน เนื่องจากไม่ต้องการข้อมูลทางกายภาพหรือใช้ข้อมูลในอดีตในปริมาณมาก ดังภาพที่ 2.2



ภาพที่ 2.2 แบบจำลองการพยากรณ์บนพื้นฐานของระยะทางการเชื่อมโยงข้อมูลและระยะเวลา

ที่มา: Ye et al. (2022)

รูปแบบการพยากรณ์สามารถแยกแยะตามความแตกต่างตามช่วงเวลา ได้อีก 3 รูปแบบ (สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน, 2562) ได้แก่

**2.2.1. การพยากรณ์ระยะปัจจุบัน (nowcasting)** เนื่องจากความต้องการทราบรายละเอียดของสภาพอากาศในปัจจุบัน โดยใช้การพยากรณ์สภาพอากาศล่วงหน้า 3-4 ชั่วโมง เป็นรูปแบบการพยากรณ์ระยะสั้นมาก แต่มีความสำคัญต่อผู้ควบคุมระบบโครงข่ายไฟฟ้าหลัก ในการบริหารจัดการระบบได้อย่างมีประสิทธิภาพ และเป็นการรักษาความมั่นคงของโครงข่ายไฟฟ้าเป็นหลัก สำหรับการพยากรณ์ระยะปัจจุบัน มีความสำคัญต่อผู้ควบคุมโรงไฟฟ้าพลังงานแสงอาทิตย์ ในด้านการบริหารจัดการกำลังการผลิตไฟฟ้า ประสิทธิภาพและความสามารถของระบบผลิตไฟฟ้า โดยการวิเคราะห์เพื่อเปรียบเทียบกำลังไฟฟ้าที่โรงไฟฟ้าพลังงานแสงอาทิตย์ผลิตได้จริง กับค่ากำลังการผลิตไฟฟ้าที่พยากรณ์ ผ่านการคำนวณ โดยใช้ข้อมูลความเข้มแสงอาทิตย์ที่วัดได้ ณ ขณะนั้น โดยทั่วไปแล้วการพยากรณ์ระยะปัจจุบัน จะดำเนินการทุก ๆ 10-15 นาที

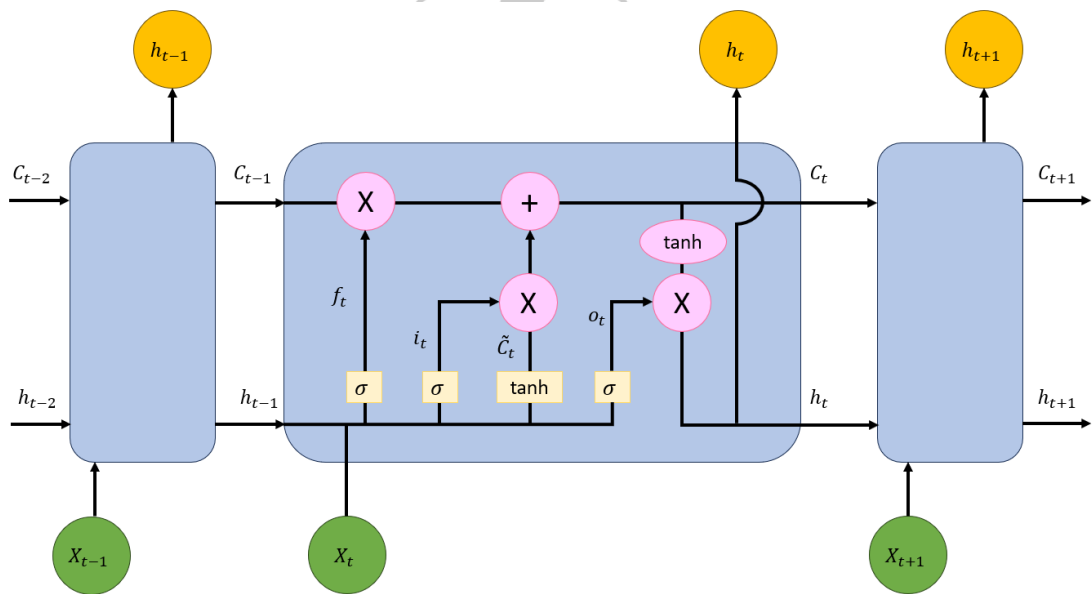
**2.2.2. การพยากรณ์ระยะสั้น (Solar PV Short-term Forecasting)** มีความสามารถในการพยากรณ์ข้อมูลล่วงหน้าได้ในระยะเวลา 7 วันข้างหน้า มีประโยชน์สำหรับผู้ควบคุมระบบโครงข่ายไฟฟ้าหลัก ช่วยสนับสนุนข้อมูลในการตัดสินใจในการวางแผนและควบคุมระบบโครงข่ายไฟฟ้าหลัก โดยข้อมูลนำเข้าจะเป็นข้อมูลทางอุตุนิยมวิทยา ผ่านการประเมินอย่างละเอียด ในเชิงพื้นที่และเชิงเวลาที่แตกต่างกัน ตัวแปรทางอุตุนิยมวิทยาและปรากฏการณ์ต่าง ๆ จะถูกมองจากมุมมองที่กว้างขึ้น ไม่ใช่ข้อมูลในระดับท้องถิ่นเหมือนการพยากรณ์ระยะปัจจุบัน วิธีการที่ใช้ในการพยากรณ์ส่วนใหญ่ จะใช้แบบจำลองการพยากรณ์อากาศเชิงตัวเลข (Numerical Weather Prediction: NWP)

**2.2.3. การพยากรณ์ระยะยาว (Solar PV Long-term Forecasting)** การพยากรณ์ประเภทนี้ จะใช้ข้อมูลทรัพยากรด้านพลังงานที่มีอยู่ในอนาคตในรายปีหรือรายเดือน มีประโยชน์ต่อผู้ผลิตพลังงานไฟฟ้าและเป็นข้อมูลสนับสนุนในการเจรจาต่อรองเพื่อทำสัญญาซื้อขายไฟฟ้า

### 3. หน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory)



หน่วยความจำระยะสั้นแบบยาว หรือ Long Short-Term Memory (LSTM) เป็นประเภทของโครงข่ายประสาทเทียมที่พัฒนามาจากโครงข่ายประสาทแบบวนซ้ำ (Recurrent neural network: RNN) เป็นที่รู้จักในปี ค.ศ. 1997 โดย (Hochreiter & Schmidhuber, 1997) เพื่อแก้ปัญหาการลดลงของเกรเดียนต์ (Vanishing Gradient) ซึ่งเป็นปัญหาที่เกิดขึ้นกับโครงข่ายประสาทแบบวนซ้ำ เมื่อความยาวของข้อมูลมีมากเกินไป โดยพบว่าขณะที่ทำการฝึกสอนแบบจำลอง การคำนวณค่าความผิดพลาดย้อนหลัง (Backpropagation) ส่งผลให้ค่าเกรเดียนต์มีขนาดเล็กลงจนมีค่าเท่ากับศูนย์ ทำให้การเรียนรู้ของแบบจำลองช้าลงหรือหยุดการทำงาน LSTM จึงถูกสร้างขึ้นมาเพื่อปัญหานี้ ด้วยการเพิ่มหน่วยความจำระยะยาว (Long term memory) คำนวณร่วมกับหน่วยความจำระยะสั้น (Short term memory) โดยที่หน่วยความจำระยะยาวใน LSTM สามารถเก็บความสัมพันธ์ของข้อมูลที่มีความยาวได้ดี มีความเหมาะสมในการแก้ปัญหาที่เกี่ยวข้องกับข้อมูลตามลำดับ (Sequential Data) โครงสร้างของหน่วยความจำระยะสั้นแบบยาว แสดงดังภาพที่ 2.3

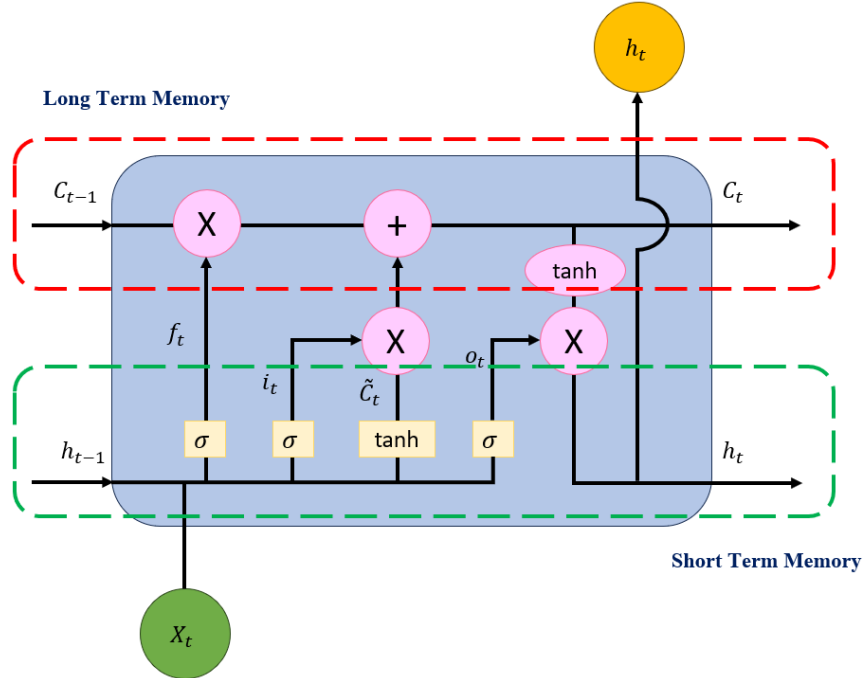


ภาพที่ 2.3 โครงสร้างของหน่วยความจำระยะสั้นแบบยาว

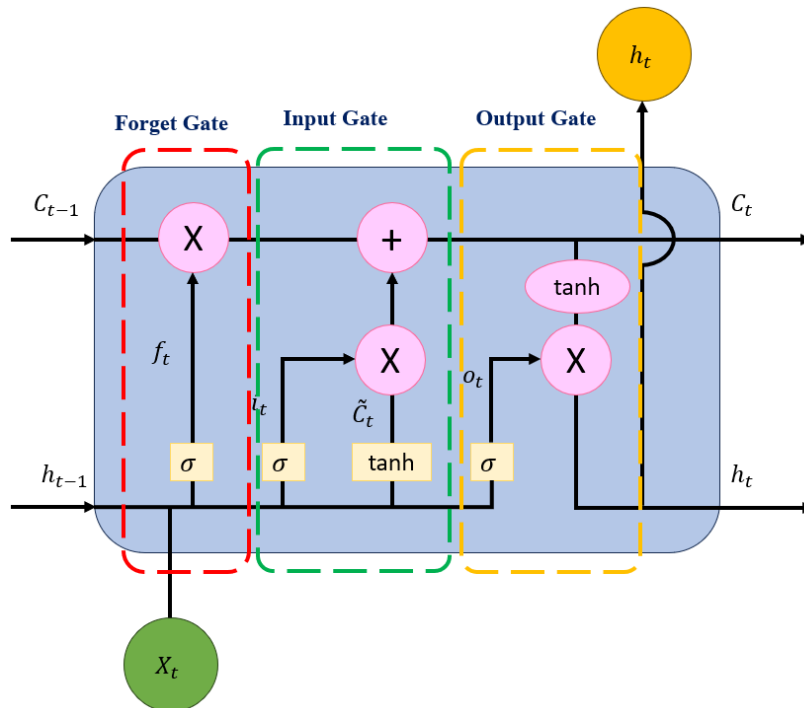
ที่มา: saxena (2021)

หน่วยความจำระยะยาว (LSTM) มีโครงสร้างหลักที่ปรับปรุงมาจากโครงข่ายประสาทแบบวนกลับ (RNN) โดยมีส่วนที่เรียกว่า หน่วยความจำระยะยาว (Long term memory) และหน่วยความจำระยะสั้น (Short term memory) แสดงดังภาพที่ 2.4 และเพิ่มประตูสัญญาณ (Gate)

สำหรับการตัดสินใจของ Cell State จำนวน 3 ประตู ได้แก่ Forget Gate Input Gate และ Output Gate แสดงดังภาพที่ 2.5

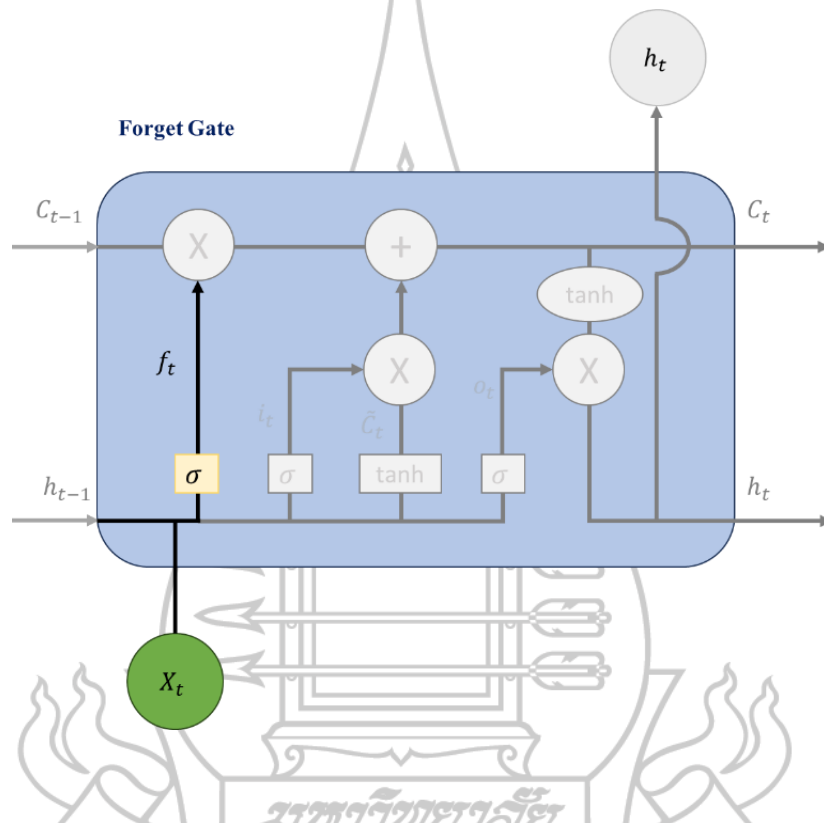


ภาพที่ 2.4 ส่วนประกอบหน่วยความจำระยะสั้นและหน่วยความจำระยะยาวของ LSTM



ภาพที่ 2.5 ประตู Forget Gate Input Gate และ Output Gate ของ LSTM

**3.1. Forget Gate** เป็นประตูที่ถูกใช้สำหรับการลืมหรือลบข้อมูลใน Cell State ซึ่งรับมาจาก Cell State ก่อนหน้า ( $C_{t-1}$ ) โดยจะพิจารณาว่าจะลบข้อมูลใดออกไปจาก Cell State ผ่านฟังก์ชันซิกมอยด์ (Sigmoid Function) ในการตัดสินใจ โดยฟังก์ชันซิกมอยด์จะให้ค่าผลลัพธ์ออกเป็นตัวเลขระหว่าง 0 หรือ 1 หากค่าเป็น 0 ให้ลบข้อมูลทิ้ง แต่ถ้าค่าเป็น 1 ให้เก็บข้อมูลนั้นไว้ ลักษณะการทำงานของ Forget Gate แสดงดังภาพที่ 2.6 และคำนวณได้ ดังสมการ

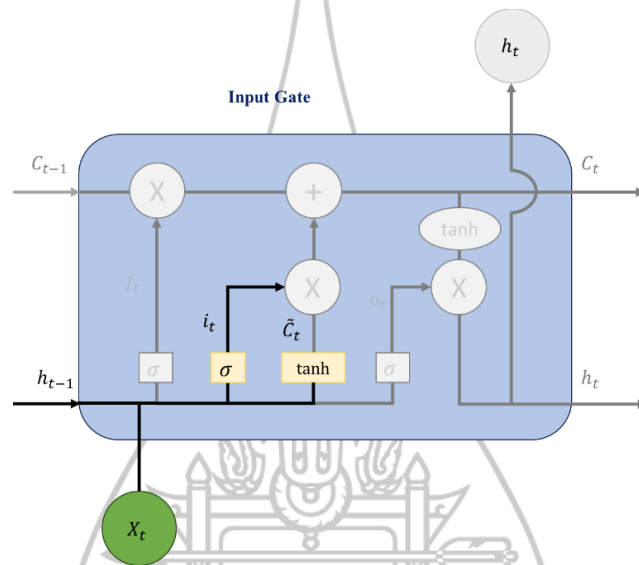


ภาพที่ 2.6 ลักษณะการทำงานของ Forget Gate

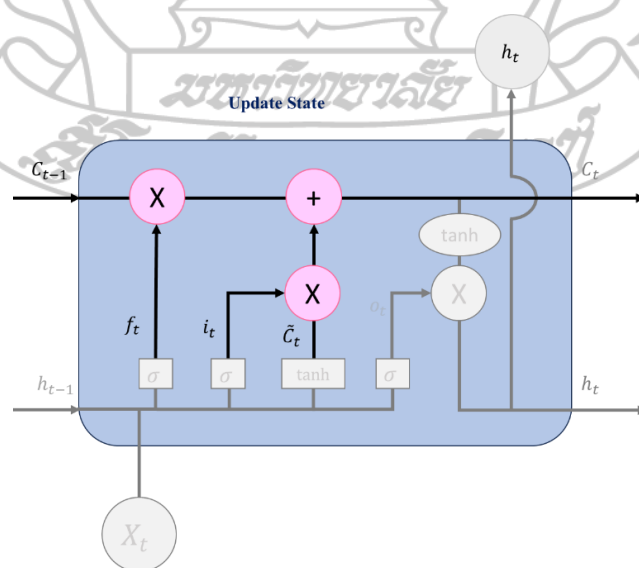
$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (2-1)$$

- โดยที่
- $f_t$  คือ Forget Gate
  - $\sigma$  คือ ฟังก์ชันซิกมอยด์
  - $W_f$  คือ ค่าน้ำหนักของ Matrices
  - $h_{t-1}$  คือ ค่าน้ำหนักของ Output ของ Cell State ก่อนหน้า ณ เวลา t-1
  - $X_t$  คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t
  - $b_f$  คือ ค่า Bias

**3.2. Input Gate** เป็นประตูที่ถูกใช้ในการรับข้อมูลใหม่ ( $X_t$ ) เข้ามา โดยเป็นการคำนวณสำหรับการอัปเดต Cell State ด้วยข้อมูลนำเข้าใหม่ ผ่านการควบคุมของฟังก์ชันซิกมอยด์ ซึ่งจะเรียกใช้ Input Gate เพื่อตัดสินใจว่าจะอัปเดต Cell State หรือไม่ หาก Input Gate ต้องการอัปเดต Cell State ฟังก์ชัน tanh จะสร้าง Candidate Values ( $\tilde{C}_t$ ) ขึ้นมา ลักษณะการทำงานของ Input Gate และการ Update Cell State แสดงดังภาพที่ 2.7 – 2.8 และคำนวณได้ ดังสมการ



ภาพที่ 2.7 ลักษณะการทำงานของ Input Gate



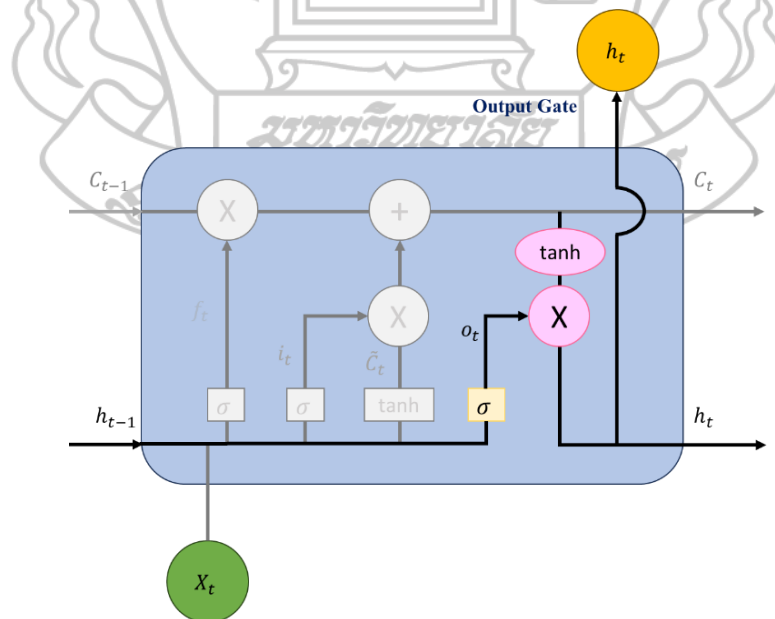
ภาพที่ 2.8 ลักษณะการทำงานของ Update Cell State

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

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2-3)$$

โดยที่	$i_t$	คือ Input Gate
	$\sigma$	คือ ฟังก์ชันซิกมอยด์
	$\tilde{C}_t$	คือ ค่า Candidate Values ของ Cell State ณ เวลา t
	$\tanh$	คือ ฟังก์ชัน tanh
	$h_{t-1}$	คือ ค่านำหน้าของ Output ของ Cell State ก่อนหน้า ณ เวลา t-1
	$x_t$	คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t
	$b_f$	คือ ค่า Bias

3.3. Output Gate เป็นประตูในการส่งออกข้อมูล ( $h_t$ ) โดยข้อมูลที่จะถูกส่งออกจะต้องได้รับการพิจารณาจาก Cell State ที่ผ่านการกรองข้อมูลด้วยการคำนวณแล้ว โดยฟังก์ชันซิกมอยด์จะเป็นตัวตัดสินใจว่าข้อมูลใดใน Cell State จะถูกส่งออกไป จากนั้นจึงนำค่าที่ได้เข้าฟังก์ชัน tanh และส่งไปคำนวณกับค่า Output ที่ได้จากฟังก์ชัน Sigmoid ลักษณะการทำงานของ Output Gate แสดงดังภาพที่ 2.9 และคำนวณได้ ดังสมการ



ภาพที่ 2.9 ลักษณะการทำงานของ Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2-4)$$

โดยที่	$o_t$	คือ Out Gate
	$\sigma$	คือ ฟังก์ชันซิกมอยด์
	$W_o$	คือ ค่าน้ำหนักของ Matrices
	$\tanh$	คือ ฟังก์ชัน tanh
	$h_{t-1}$	คือ ค่าน้ำหนักของ Output ของ Cell State ก่อนหน้า ณ เวลา t-1
	$x_t$	คือ ค่า Input ที่เข้ามาใน Cell State ณ เวลา t
	$b_o$	คือ ค่า Bias

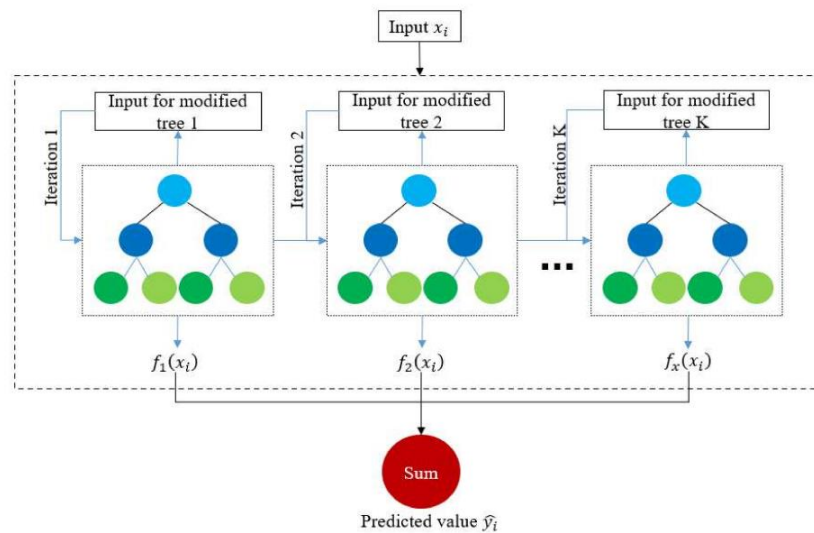
#### 4. เอ็กซ์ตรีมเกรเดียนต์บูสตีจ (Extreme Gradient Boosting)

เอ็กซ์ตรีมเกรเดียนต์บูสตีจ (Extreme Gradient Boosting: XGBoost) เป็นเทคนิคการเรียนรู้ของเครื่องที่พัฒนามาจากอัลกอริทึมต้นไม้ความเร็วแบบไล่ระดับ (Gradient Boosted Tree) จัดอยู่ในกลุ่มวิธีการเอนเซมเบิล (ensemble learning method) โดยแบบจำลองได้เลือกใช้ต้นไม้ตัดสินใจ (Decision Tree) มาฝึกสอนต่อกันจำนวนหลาย ๆ ต้น เพื่อเรียนรู้จากค่าความผิดพลาดของกระบวนการทำงานก่อนหน้า ส่งผลต่อการพยากรณ์ที่ให้ค่าความแม่นยำมากขึ้น โดยแบบจำลองจะถูกหยุดการเรียนรู้ เมื่อพบว่าการเรียนรู้ของต้นไม้ตัดสินใจต่อเนื่องจนมีความลึกมากเพียงพอ จนไม่เหลือค่าความผิดพลาดจากต้นไม้ตัดสินใจต้นก่อนหน้าให้เรียนรู้ (Chen & Guestrin, 2016) ดังสมการ

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2-5)$$

โดยที่	$\hat{y}_i$	คือ ค่าพยากรณ์ของข้อมูลลำดับที่ $i$
	$x_i$	คือ Feature Vector
	$K$	คือ จำนวนต้นไม้ในการตัดสินใจ
	$f_k(x_i)$	คือ ค่าพยากรณ์ของลำดับที่ $k$

XGBoost ถูกใช้ในการแก้ปัญหาการถดถอยและการจำแนกประเภท โครงสร้างของ XGBoost ประกอบด้วยโหนดรากหลายโหนด โหนดภายใน โหนดใบ และกิ่งก้าน จากภาพที่ 2.10 แสดงให้เห็นโครงสร้างการทำงาน โดยที่  $x_i$  ซึ่งเป็น Feature Vector จะถูกป้อนและส่งผ่านไปยังโหนดรากทั้งหมดเพื่อทำการตัดสินใจ จากนั้นโหนดภายในจะทำการตัดสินใจในภายหลัง แต่ละรากจะชี้ไปที่การตัดสินใจโดยตรงและโหนดใบแสดงถึงผลการทำนายของค่าพยากรณ์แต่ละลำดับ  $f_k(x_i)$  ผลลัพธ์ของโหนดใบทั้งหมดจะถูกรวมเข้าด้วยกันเพื่อให้ได้ผลของการทำนาย ( $\hat{y}_i$ ) ของแบบจำลอง XGBoost (Zou et al., 2022)



ภาพที่ 2.10 โครงสร้างและหลักการทำงานของ XGBoost

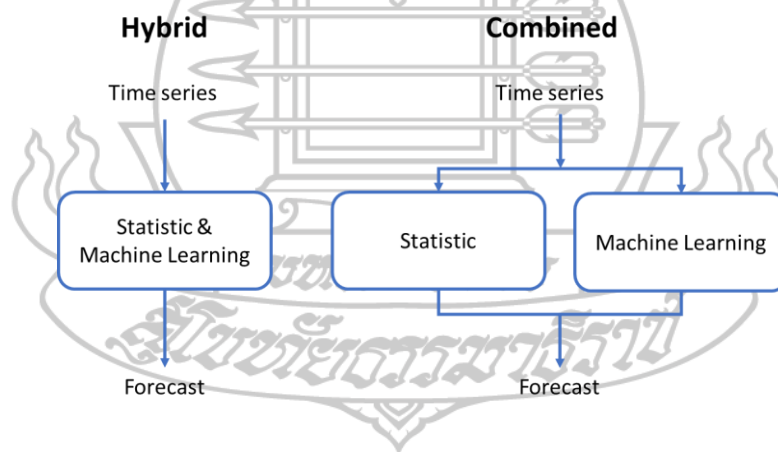
ที่มา: Zou et al. (2022)

นอกจากนี้ XGBoost ยังมีความสามารถในการตรวจจับความสัมพันธ์ของข้อมูลที่มีความเป็นลำดับ รวมถึงข้อมูลที่ไม่เป็นเชิงเส้น ทำให้แบบจำลองถูกใช้ในงานด้านการพยากรณ์อนุกรมเวลา ด้วยความสามารถที่โดดเด่น จึงได้รับความนิยมและถูกใช้ในจัดการกับข้อมูลที่มีความซับซ้อนและเปลี่ยนแปลงตลอดเวลา เนื่องจากสามารถตรวจจับแนวโน้ม ฤดูกาล และความสัมพันธ์ที่ซับซ้อนของข้อมูลได้ เช่น การพยากรณ์เชิงตัวเลข (Numerical Weather Prediction: NWP) โดยพบว่าสามารถทำงานได้ดีกับข้อมูลที่มีปริมาณมาก (Phan et al., 2021)

## 5. วิธีการพยากรณ์แบบผสม (Hybrid Forecasting Method)

วิธีการพยากรณ์แบบผสมถูกพัฒนาโดยมีแนวคิด คือ การลดข้อจำกัดของวิธีการหนึ่ง ด้วยข้อดีของอีกวิธีการหนึ่ง โดยจะผสมผสานความสามารถรวมการเรียนรู้ข้ามกัน ซึ่งเป็นความสามารถที่วิธีการพยากรณ์ทางสถิติยังขาดอยู่ วิธีการพยากรณ์แบบผสมจะช่วยแก้ปัญหาสมมติฐานเชิงเส้น เนื่องจากข้อมูลอนุกรมเวลาอาจเป็นแบบเชิงเส้นหรือไม่เป็นเชิงเส้นก็ได้ หรือมักจะมีรูปแบบทั้งสองรูปแบบผสมกัน โดยวิธีการพยากรณ์แบบผสมจะช่วยให้การพยากรณ์มีประสิทธิภาพมากขึ้นได้ในกรณีที่วิธีการพยากรณ์อนุกรมเวลาแบบดั้งเดิมมีข้อจำกัด (Berberich, 2020)

วิธีการแบบผสมจะแตกต่างจากวิธีการพยากรณ์ร่วม (Combine Forecasting Method) โดยวิธีการพยากรณ์ร่วมจะใช้ค่าเฉลี่ยแบบถ่วงน้ำหนักร่วมกับค่าผลลัพธ์จากพยากรณ์ที่ได้จากแบบจำลอง (Li et al., 2019) ซึ่งจะมีความง่ายในการคำนวณและมีประสิทธิภาพในการพยากรณ์ โดยแตกต่างจากวิธีการแบบผสมที่รวมเอาความสามารถของการเรียนรู้ของเครื่องและการพยากรณ์ทางสถิติ แสดงดังรูปที่ 2.11



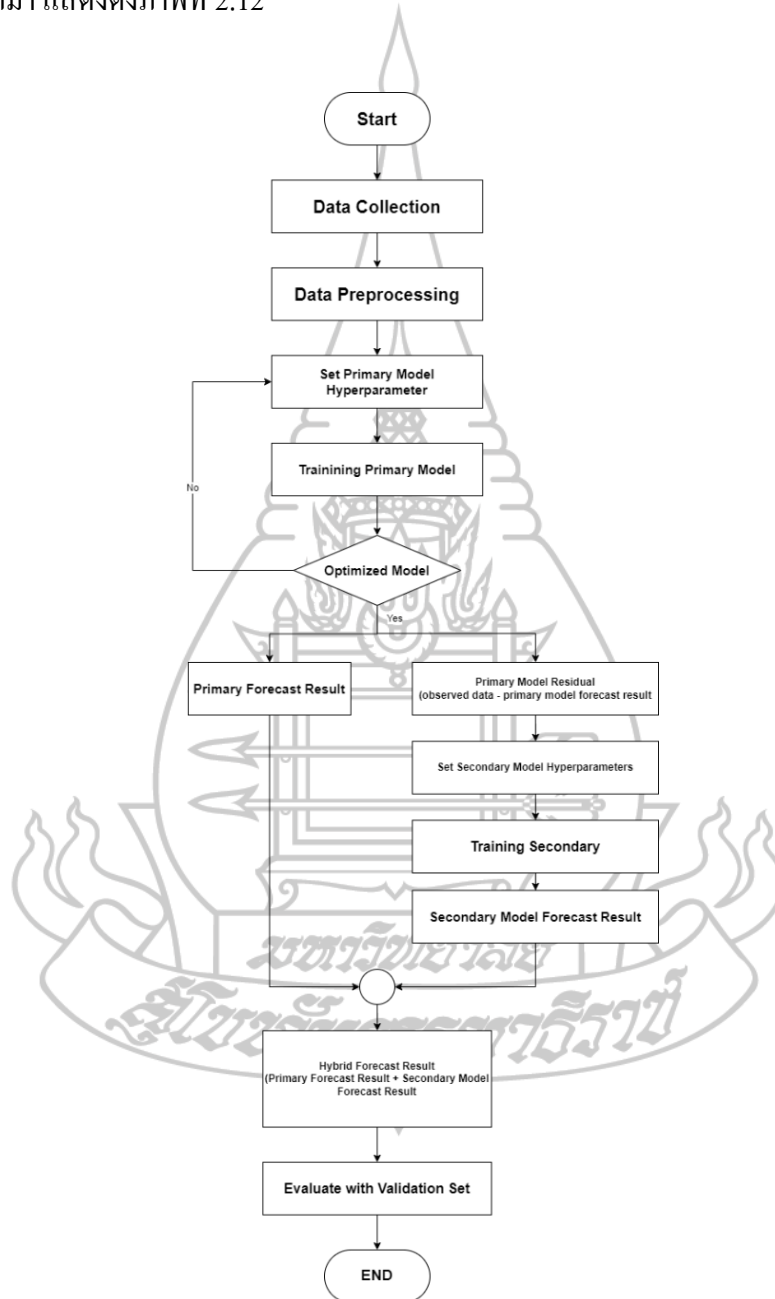
ภาพที่ 2.11 ความแตกต่างระหว่างวิธีการพยากรณ์แบบผสมและวิธีการพยากรณ์ร่วม

ที่มา: อูรชา (2564)

วิธีการแบบผสม เป็นการสร้างแบบจำลองเดี่ยวและคัดเลือกตัวแบบพยากรณ์เดี่ยวที่มีความเหมาะสมและแม่นยำมากที่สุด มาสร้างตัวแบบพยากรณ์ผสม เมื่อได้แบบจำลองพยากรณ์จาก



วิธีพยากรณ์หลัก ให้นำมาหาค่าส่วนเหลือ (Residual) จากตัวแบบพยากรณ์หลัก จากนั้นนำค่าส่วนเหลือที่ได้มาใช้เป็นข้อมูลนำเข้าให้กับแบบจำลองด้วยวิธีพยากรณ์รอง จากนั้นนำค่าพยากรณ์จากวิธีพยากรณ์หลักนำมารวมกับค่าพยากรณ์ของส่วนเหลือจากวิธีพยากรณ์รอง (อุรษา, 2564) จะได้ค่าผลลัพธ์ออกมา แสดงดังภาพที่ 2.12



ภาพที่ 2.12 Flow Chart วิธีการพยากรณ์แบบผสม

## 6. การคัดเลือกคุณลักษณะ (Feature Selection)

เป็นวิธีการที่ช่วยลดจำนวนคุณลักษณะหรือแอททริบิวต์ (Attribute) โดยหาความสัมพันธ์จากคุณลักษณะทั้งหมดที่มีเพื่อหาความเกี่ยวข้องกัน ดังนั้นจึงใช้การเลือกคุณลักษณะเพื่อกำจัดเฉพาะคุณลักษณะที่มีความเกี่ยวข้องกันเท่านั้น (ศรีรัศมิ์, 2566) ด้วยชุดข้อมูลที่มีลักษณะเฉพาะ (feature) ที่เป็นตัวเลข (numeric) และผลลัพธ์เป้าหมาย (target result) เป็นตัวเลข จึงสามารถเลือกใช้วิธีสหสัมพันธ์แบบเพียร์สัน (Pearson's Correlation)

การศึกษาเกี่ยวกับความสัมพันธ์ของตัวแปรอิสระกับตัวแปรตามทางสถิติ โดยเกณฑ์ในการคัดเลือกคุณลักษณะหากค่าสัมประสิทธิ์มี ค่าตั้งแต่ 0.5 ถึง 1 แสดงว่าตัวแปรอิสระกับตัวแปรตามมีความสัมพันธ์เป็นไปในทิศทางเดียวกัน สามารถนำไปใช้ในการสร้างแบบจำลองได้ วิธีการคำนวณ ดังสมการ

$$r_{xy} = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{n\sum x^2 - (\sum x)^2 [n\sum y^2 - (\sum y)^2]}} \quad (2-6)$$

โดยที่	$r_{xy}$	คือ สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน
	$x$	คือ ตัวแปรอิสระหรือตัวแปรต้น
	$y$	คือ ตัวแปรตาม
	$n$	คือ ขนาดข้อมูล

## 7. การปรับขนาดข้อมูลให้มาตรฐาน (Scaling Data)

เทคนิคการปรับข้อมูลให้มาตรฐาน (Scaling) เมื่อพบว่าข้อมูลส่วนใหญ่ที่เป็นตัวเลขมีความแตกต่างจากข้อมูลตัวเลขอื่น จำเป็นต้องมีการปรับขนาดข้อมูลให้มาตรฐาน (Liu et al., 2021) แบ่งออกเป็น 2 วิธี ได้แก่

7.1. การทำให้เป็นมาตรฐาน (Normalization) เป็นการแปลงค่าข้อมูลให้อยู่ในช่วงสั้นๆ ตั้งแต่ 0 – 1 หรือที่เรียกว่าการทำให้เป็นมาตรฐานต่ำสุด-สูงสุด (min-max normalization) การคำนวณ ดังสมการ

$$x^* = \frac{x - x_{min}}{x - x_{max}} \quad (2-7)$$

โดยที่  $x^*$  คือ ค่าที่จะทำให้เป็นมาตรฐานของตัวแปร  $x$   
 $x_{min}$  คือ ค่าต่ำสุดของตัวแปร  $x$   
 $x_{max}$  คือ ค่าสูงสุดของตัวแปร  $x$

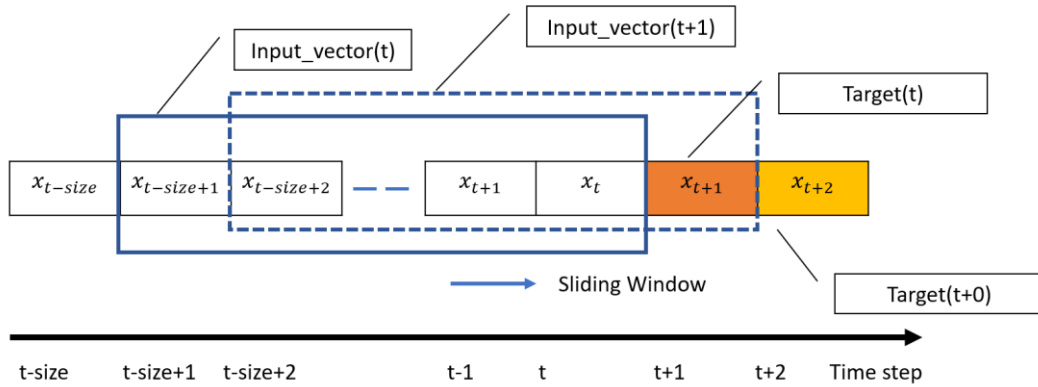
7.2. การกำหนดมาตรฐาน (Standardization) เป็นการปรับขนาดของข้อมูลตามค่าเบี่ยงเบนมาตรฐาน (standard deviation) = 1 หากค่าเบี่ยงเบนมาตรฐานของลักษณะเฉพาะมีความแตกต่างกัน จะทำให้ช่วงของข้อมูลแตกต่างกันไปด้วย การคำนวณ ดังสมการ

$$x_{std}^* = \frac{x^* - u_x}{\sigma_x} \quad (2-8)$$

โดยที่  $x^*$  คือ ค่าตัวอย่างเฉพาะของตัวแปร  $x$   
 $x_{std}^*$  คือ ค่ามาตรฐานใหม่ของ  $x^*$   
 $u_x$  คือ ค่าเฉลี่ยตัวอย่างของคอสมัน์คุณลักษณะเฉพาะ  
 $\sigma_x$  คือ ส่วนเบี่ยงเบนมาตรฐานที่สอดคล้องกัน

## 8. การแบ่งข้อมูลเพื่อประเมินประสิทธิภาพแบบจำลอง (Data Partition)

8.1. การแบ่งข้อมูลรูปแบบ Sliding Window คือ การแบ่งชุดข้อมูลฝึกสอนก่อนนำเข้าแบบจำลอง จุดเริ่มต้นของข้อมูลจะถูกขยับออกไปเรื่อย ๆ แต่จำนวนชุดข้อมูลทั้งหมดยังคงมีปริมาณคงเดิม (Judith Foster et al., 2017) การแบ่งข้อมูลประเภทนี้ เหมาะสำหรับการชุดข้อมูลอนุกรมเวลาที่มีการนำข้อมูลมาเรียงตามเวลาสำหรับการพยากรณ์ค่าถัดไป ตัวอย่างการแบ่งข้อมูลรูปแบบ Sliding Window แสดงดังภาพที่ 2.13 และตัวอย่างชุดข้อมูลที่ได้รับการแบ่ง แสดงดังภาพที่ 2.14



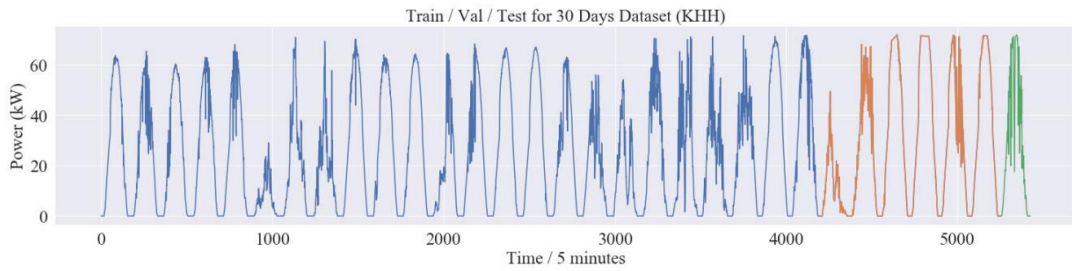
ภาพที่ 2.13 การแบ่งข้อมูลรูปแบบ Sliding Window

ที่มา: Liu et al. (2021)

ลำดับที่ 1	ลำดับที่ 2	ลำดับที่ ...	ลำดับที่ 23	ลำดับที่ 24
0.09314	<b>0.09314</b>	...	0.09314	<b>0.09314</b>
<b>0.09314</b>	<b>0.08987</b>	...	<b>0.09314</b>	<b>0.08987</b>
<b>0.08987</b>	<b>0.08007</b>	...	<b>0.08987</b>	<b>0.08007</b>
<b>0.08007</b>	<b>0.09314</b>	...	<b>0.08007</b>	<b>0.09314</b>
<b>0.09314</b>	0.08660	...	<b>0.09314</b>	0.08660

ภาพที่ 2.14 ตัวอย่างชุดข้อมูลที่ได้รับการแบ่งด้วยวิธี Sliding Window

8.2. การแบ่งข้อมูลเพื่อประเมินประสิทธิภาพแบบจำลอง ข้อมูลที่จะใช้เป็นข้อมูลนำเข้าในแบบจำลองสำหรับการพยากรณ์ โดยจะถูกแบ่งออกเป็นสองส่วน ส่วนแรกถูกใช้เป็นข้อมูลสำหรับฝึกสอนแบบจำลองและอีกส่วนใช้สำหรับเป็นข้อมูลทดสอบแบบจำลอง การแบ่งข้อมูลแบบ Split Test เป็นที่นิยม โดยจะแบ่งข้อมูลในอัตราส่วน 80 : 20 หรือ 70 : 30 สำหรับข้อมูลชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ นอกจากนี้ยังมีการแบ่งข้อมูลอีกประเภท คือ การแบ่งข้อมูลออกเป็น 3 ส่วน ได้แก่ ชุดข้อมูลฝึกสอน ชุดข้อมูลตรวจสอบ และชุดข้อมูลทดสอบ โดยอาจแบ่งในอัตราส่วน 80 : 10 : 10 หรือ 70 : 20 : 10 แสดงดังภาพที่ 2.15



ภาพที่ 2.15 ตัวอย่างการแบ่งข้อมูลประเมินประสิทธิภาพแบบจำลอง

ที่มา: Liu et al. (2021)

## 9. การประเมินประสิทธิภาพแบบจำลอง (Performance Evaluation)

เพื่อที่จะประเมินความสามารถของแบบจำลองการพยากรณ์ที่ใช้ข้อมูลอนุกรมเวลา ที่ผ่านการฝึกสอนด้วยอัลกอริทึมการเรียนรู้เชิงลึก การประเมินประสิทธิภาพของแบบจำลองถือเป็นสิ่งสำคัญ โดยประสิทธิภาพของแบบจำลองนั้นจะถูกเลือกจากความแม่นยำของผลลัพธ์ที่ได้จากแบบจำลอง หากค่าผลลัพธ์มีความคลาดเคลื่อนระหว่างค่าจริงกับค่าที่พยากรณ์ได้ ยิ่งค่าใกล้ศูนย์ จะถือว่าแบบจำลองนั้นมีประสิทธิภาพสูง (เกศินี, 2564) การวัดค่าความคลาดเคลื่อน มีดังนี้

**9.1. Mean Absolute Error (MAE)** คือ การนำค่าสัมบูรณ์ (Absolute) มาใช้หาค่าความผิดพลาดของแบบจำลอง ด้วยวิธีการหาค่าสัมบูรณ์ของผลต่างระหว่างค่าจริงและค่าที่พยากรณ์ได้ จากนั้นนำไปหาค่าเฉลี่ย ดังสมการ

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (2-9)$$

**9.2. Mean Square Error (MSE)** คือ การนำผลต่างระหว่างค่าจริงและค่าที่พยากรณ์ได้มายกกำลังสอง จากนั้นทำการหาค่าเฉลี่ยของผลลัพธ์ ดังสมการ

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (2-10)$$

**9.3. Root Mean Square Error (RMSE)** คือ การนำค่าความผิดพลาดมายกกำลังสอง จากนั้นจึงทำการถอดรากที่สอง หรือเป็นการนำค่า Mean Square Error (MSE) มาทำการถอดรากที่

สอง โดยผลลัพธ์ค่าความผิดพลาดจะเป็นบวก ซึ่งค่าที่ได้จะมีค่าน้อยกว่า Mean Square Error มีข้อดีคือทำให้ง่ายต่อการอ่านค่า ดังสมการ

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (2-11)$$

โดยที่  $y_t$  คือ ค่าจริงที่  $t$  ใด ๆ  
 $\hat{y}_t$  คือ ค่าที่ได้จากการพยากรณ์ที่  $t$  ใด ๆ  
 $n$  คือ ขนาดข้อมูล

วิธีการวัดค่าความคลาดเคลื่อนแต่ละชนิดที่กล่าวมาข้างต้น เพื่อให้ง่ายต่อความเข้าใจ จะทำการจำลองค่าในการพยากรณ์ โดยมีข้อมูลจำนวน 2 ชุด และสมมติค่าจริงและค่าที่ได้จากการพยากรณ์ เพื่อแสดงผลจากการวัดค่าความคลาดเคลื่อนทั้ง 3 วิธี แสดงดังภาพที่ 2.16

ค่าจริง	10.5	15.7	19.7
ค่าพยากรณ์ 1	11.8	19.4	22.3
ค่าพยากรณ์ 2	11.4	18.6	42.4

	MAE	MSE	RMSE
ค่าพยากรณ์ 1	2.53	7.38	2.72
ค่าพยากรณ์ 2	8.83	174.84	13.22

ภาพที่ 2.16 ตัวอย่างผลการคำนวณด้วยวิธีการวัดค่าความคลาดเคลื่อนด้วยค่าสมมติ

ทีมา สรวุฒิ และคณะ., (2566)

จากภาพที่ 2.16 ค่าพยากรณ์ 1 แสดงผลของค่าพยากรณ์ที่ใกล้เคียงกับค่าจริง และค่าพยากรณ์ 2 แสดงผลของค่าพยากรณ์ที่มีค่าแตกต่างจากค่าจริงมากในบางค่า เพื่อแสดงการเปรียบเทียบการค่าความผิดพลาดระหว่างค่าพยากรณ์ของทั้ง 2 ชุดข้อมูล ผลการพยากรณ์แสดงให้เห็นถึงค่าความคลาดเคลื่อนแต่ละชนิดเมื่อชุดข้อมูลมีลักษณะแตกต่างกัน

จากสมการ ค่า MAE เหมาะกับชุดข้อมูลที่มีค่าส่วนต่างระหว่างค่าจริงและค่าพยากรณ์ที่แตกต่างจากค่าอื่นมาก โดยเป็นค่าต่ำหรือสูงกว่าในจำนวนมาก มีความสามารถในการรับผลข้อมูลที่ที่มีค่าความแตกต่างกันมากได้ดี ในขณะที่ค่า MSE ถูกคำนวณด้วยเลขยกกำลัง ค่าส่วนต่างของชุดที่แตกต่างกันมาก ส่งผลต่อผลลัพธ์ที่คลาดเคลื่อนมากกว่าปกติ ทำให้สามารถทราบค่าความผิดพลาดได้อย่างชัดเจนกว่า RMSE ที่ใช้เลขยกกำลังในการคำนวณค่าความแตกต่างที่มากขึ้น ส่งผลต่อผลลัพธ์มากขึ้น เนื่องจากมีการถอดรากที่สองของผลลัพธ์กลับมาอยู่ในหน่วยเดียวกับข้อมูล ทำให้ง่ายต่อการวัดผล และ RMSE ยังมีความสามารถในการตรวจจับความผิดพลาดของชุดข้อมูลที่มีขนาดใหญ่ ทำให้ทราบค่าผิดพลาดได้อย่างชัดเจน หากต้องการทราบค่าความคลาดเคลื่อนแบบปกติ การใช้สมการ RMSE จะเหมาะสมกว่า ดังนั้น การเลือกใช้วิธีการวัดค่าความคลาดเคลื่อนแต่ละชนิดจะขึ้นอยู่กับลักษณะของชุดข้อมูลที่น่าใช้ฝึกสอนแบบจำลอง รวมถึงลักษณะผลลัพธ์ที่ต้องการ

## 10. งานวิจัยที่เกี่ยวข้อง

ภานุพงศ์ น้าแก้ว (2563) ได้ทำการศึกษาปัจจัยที่มีผลต่อการพยากรณ์กำลังไฟฟ้าและศึกษาทดลองฟังก์ชันถ่ายโอนของโครงข่ายประสาทเทียมแบบต่าง ๆ โดยนำข้อมูลสภาพอากาศที่ได้จากเซนเซอร์ของระบบเซลล์แสงอาทิตย์มาวิเคราะห์ด้วยโครงข่ายประสาทเทียมเพอร์เซปตรอนหลายชั้น (Multilayer Perceptron Network : MLP) โดยพบว่าข้อมูลนำเข้าที่เหมาะสมต่อการพยากรณ์กำลังไฟฟ้าของระบบเซลล์แสงอาทิตย์ ได้แก่ ค่าความเข้มของแสงอาทิตย์ อุณหภูมิบนแผงเซลล์แสงอาทิตย์ และค่ากำลังไฟฟ้าของเซลล์แสงอาทิตย์วันก่อนพยากรณ์ โดยทำการประเมินผลด้วยวิธีวัดค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) อยู่ที่ร้อยละ 25.39 จากนั้นทำการทดลองหารูปแบบโครงข่ายประสาทเทียมที่มีความแม่นยำสูงที่สุด ซึ่งพบว่ามีค่าเฉลี่ยของร้อยละความผิดพลาดสัมบูรณ์ (MAPE) เฉลี่ยตลอดทั้งปีอยู่ที่ร้อยละ 18.80

จากงานวิจัยของภานุพงศ์ น้าแก้ว พบว่าปัจจัยที่ส่งผลต่อการพยากรณ์กำลังไฟฟ้าที่ได้จากพลังงานแสงอาทิตย์ ได้แก่ ค่าความเข้มของแสงอาทิตย์ อุณหภูมิบนแผงเซลล์แสงอาทิตย์ และค่ากำลังไฟฟ้าของเซลล์แสงอาทิตย์วันก่อนพยากรณ์ จึงเป็นแนวทางในการหาความสัมพันธ์และการคัดเลือกคุณลักษณะเฉพาะของข้อมูลที่จะให้นำเข้าแบบจำลอง

เกศินี ช่อนกลิ่น (2564) ได้นำเสนอการเปรียบเทียบแบบจำลองการเรียนรู้เชิงลึกด้วยข้อมูลอนุกรมเวลา โดยใช้ชุดข้อมูลการลงทุนซื้อขายล่วงหน้า (ประเทศไทย) และใช้แบบจำลอง 1D-CNN, LSTM และ SVR เพื่อหาความเหมาะสมของแบบจำลองที่ให้ค่าประสิทธิภาพสูงสุด โดยการคัดเลือกคุณลักษณะทั้งหมดเพื่อหาค่าความสัมพันธ์กับราคาปิดด้วยวิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation) ชุดข้อมูลจะถูกแบ่งโดยวิธี Sliding Window จำนวน 20 วัน เพื่อทำนายราคาปิดของข้อมูลหุ้นล่วงหน้า กำหนดชุดข้อมูลฝึกสอนและชุดข้อมูลทดสอบ 80:20 โดยแบ่งข้อมูลแบบ 5-fold cross validation ให้กับแบบจำลอง ผลการศึกษาพบว่าแบบจำลองจากวิธีการ LSTM มีประสิทธิภาพในการพยากรณ์ที่ดีที่สุด รองลงมาคือ SVR และ 1D-CNN ตามลำดับ

จากงานวิจัยดังกล่าว พบว่าวิธีการหาค่าความสัมพันธ์ของชุดข้อมูลอนุกรมเวลาด้วยวิธีการ Pearson's Correlation และการแบ่งชุดข้อมูลด้วยวิธี Sliding Window ช่วยเพิ่มความแม่นยำให้กับการพยากรณ์ให้กับแบบจำลอง จึงเป็นสาเหตุให้เลือกใช้วิธีการนี้ในการหาคุณลักษณะที่สำคัญของข้อมูลและแบ่งข้อมูลสำหรับชุดฝึกสอน

อรุชา จันทรภา (2564) ได้ทำการศึกษาการพยากรณ์มูลค่าการส่งออกรถยนต์ อุปกรณ์ และส่วนประกอบของประเทศไทย ด้วยวิธีอนุกรมเวลา วิธีการวิเคราะห์การถดถอยเชิงเส้นพหุคูณ วิธีการเรียนรู้ของเครื่อง และวิธีพยากรณ์แบบผสม ซึ่งพบว่า แบบจำลองการพยากรณ์ที่สร้างด้วยวิธีแบบผสม ด้วยอัลกอริทึม LSTM-XGB อัลกอริทึม SARIMA-XGB และอัลกอริทึม LSTM-SARIMA มีค่าร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย MAPE เท่ากับ ร้อยละ 6.63 ร้อยละ 15.40 และร้อยละ 6.27 ตามลำดับ ทั้งนี้งานวิจัยนี้จะใช้วิธีการแบบผสมไปประยุกต์ใช้ในงานวิจัยเพื่อเพิ่มประสิทธิภาพให้กับแบบจำลองที่จะสร้างขึ้น

Foster et al., (2017) ได้นำเสนอวิธีการแบ่งข้อมูลด้วยวิธีหน้าต่างบานเลื่อน (Sliding Window) โดยใช้ชุดข้อมูลการผลิตไฟฟ้าพลังงานแสงอาทิตย์ พลังงานลม และข้อมูลที่เกี่ยวข้องกับสภาพอากาศ นำมาแบ่งข้อมูลและหาค่าข้อมูลนำเข้าที่เหมาะสมกับแบบจำลอง เพื่อหาค่าผลลัพธ์ที่เหมาะสมเป็นอย่างไร ผลการศึกษาพบว่า ค่าข้อมูลนำเข้าย้อนหลังจากการใช้วิธีหน้าต่างบานเลื่อน ทุก ๆ การเลื่อนข้อมูลออกไปทุก 7 วัน จะส่งผลต่อค่าพยากรณ์ที่ดีที่สุด รองลงมาคือ 1 ปี กับชุดข้อมูลขนาดใหญ่ และ 24 ชั่วโมง ตามลำดับ

จากงานวิจัยของ Forster และคณะ ชี้ให้เห็นว่าการเลือกใช้วิธีการหน้าต่างบานเลื่อนที่เหมาะสม จะช่วยให้ชุดข้อมูลที่นำมาใช้ในการฝึกสอนแบบจำลองมีประสิทธิภาพสูงสุด รวมถึงผลลัพธ์ของค่าความถูกต้องในการพยากรณ์ ดังนั้นในการศึกษาครั้งนี้ จึงได้นำแนวทางการใช้วิธีการหน้าต่างบานเลื่อน (Sliding Window) มาประยุกต์ใช้



Obiora et al., (2021) ได้นำข้อมูลการฉายรังสีของดวงอาทิตย์รายชั่วโมงและข้อมูลสภาพอากาศ ภายในระยะเวลา 5 ปี มาสร้างแบบจำลอง โดยใช้ Extreme Gradient Boosting (XGBoost) และเปรียบเทียบกับแบบจำลอง Support Vector Machine (SVM) ผลการศึกษาด้วยข้อมูล 1 ปี พบว่า XGBoost ให้ค่าความผิดพลาดในการพยากรณ์น้อยกว่า SVM โดยมีค่า Normalized Root Mean Squared Error (NRMSE) อยู่ที่ 6.63 และ 7.81 ตามลำดับ และเมื่อใช้แบบจำลองด้วยชุดข้อมูลระยะ 5 ปี พบว่าค่าประสิทธิภาพความคลาดเคลื่อนของแบบจำลองทั้ง 2 แบบจำลอง มีค่าสูงขึ้น เนื่องจากปัญหา Overfit และปริมาณข้อมูลนำเข้าที่มีขนาดใหญ่ขึ้น

และในปี 2022 (Obiora et al., (2022) ) ได้มีการนำเสนอแบบจำลอง CNN-LSTM เพื่อศึกษาช่วงเวลาที่ดีที่สุดสำหรับการพยากรณ์รังสีดวงอาทิตย์ล่วงหน้า โดยใช้ข้อมูลสภาพอากาศ จำนวน 2 ปี จากสถานีวัดทางอุตุนิยมวิทยา นำเข้ามาวิเคราะห์ด้วยแบบจำลองไฮบริดด้วยเทคนิคโครงข่ายประสาทคอนโวลูชันและหน่วยความจำระยะสั้นแบบยาว (CNN-LSTM) โดยเปรียบเทียบกับ Support Vector Machine (SVM) จากผลการศึกษาโดยใช้ข้อมูลที่รวบรวมในช่วงเวลา 5 นาที พบว่าการพยากรณ์ด้วยแบบจำลอง CNN-LSTM มีค่า nRMSE ที่ร้อยละ 6.2 เหนือกว่า SVM และเมื่อใช้ชุดข้อมูลที่รวบรวมได้ในช่วงเวลา 15 นาที 30 นาที และ 1 ชั่วโมง พบว่าค่าประสิทธิภาพความคลาดเคลื่อนที่วัดด้วย MAE, RMSE และ nRMSE จะสูงตามไปด้วย

จากผลการทดลองของ Obiora และคณะ งานวิจัยในปี 2021 พบว่าแบบจำลอง XGBoost และ SVM ทำงานได้ดีกับชุดข้อมูลที่มีขนาดไม่ใหญ่มาก เมื่อเทียบกับปริมาณชุดข้อมูลระยะเวลา 5 ปี ที่ส่งผลต่อการเกิด Overfit ของแบบจำลอง เนื่องจากชุดข้อมูลที่ใช้ในการศึกษาครั้งนี้ มีจำนวนน้อยกว่า 1 ปี แบบจำลอง XGBoost จึงเป็นตัวเลือกที่เหมาะสมในการนำไปสร้างแบบจำลอง และจากงานวิจัยในปี 2022 ที่มีการนำเสนอแบบจำลองแบบไฮบริดด้วย CNN-LSTM ให้ค่าวัดประสิทธิภาพสูงกว่าแบบจำลอง XGBoost จากงานวิจัยในปี 2021 อย่างไรก็ตามแบบจำลองที่สร้างจาก XGBoost และแบบไฮบริดที่ใช้ CNN-LSTM มีความน่าสนใจในการใช้งานร่วมกัน ในงานวิจัยนี้จึงให้ความสนใจแบบจำลองแบบไฮบริดที่สร้างด้วย LSTM และ XGBoost

Liu et al., (2021) ได้ศึกษาการสร้างแบบจำลองด้วย LSTM จำนวน 2 เลเยอร์ เพื่อพยากรณ์ค่าพลังงานไฟฟ้าจากพลังงานแสงอาทิตย์ล่วงหน้า 1 วัน ร่วมกับปัจจัยทางสภาพอากาศ โดยใช้ข้อมูลจำนวน 2 ชุด คือ KHH และ MFU และแบ่งข้อมูลในการทดสอบออกเป็น 18 กรณี เพื่อเปรียบเทียบความแม่นยำ จุดมุ่งหมายของการพยากรณ์ในครั้งนี้เพื่อทดสอบความสามารถของ LSTM ในการพยากรณ์กับชุดข้อมูลที่มีลักษณะอนุกรมเวลาและสภาพอากาศที่แตกต่างกัน รวมถึงความแตกต่างของพื้นที่ติดตั้งอุปกรณ์ตรวจวัดข้อมูลด้วย ผลการพยากรณ์ของแบบจำลองพบว่าได้ค่าเฉลี่ย RMSE ของแบบจำลองทุกชุดเท่ากับ 0.512

จากงานวิจัยดังกล่าว ซึ่งให้เห็นว่าความสามารถของแบบจำลองที่สร้างจาก LSTM ยังคงมีประสิทธิภาพในการพยากรณ์กับข้อมูลแบบอนุกรมเวลา และมีปริมาณและคุณภาพของข้อมูลที่แตกต่างกัน ดังนั้น LSTM จึงเป็นแบบจำลองที่จะใช้ในการศึกษาครั้งนี้

Phan et al., (2021) ได้ทำการศึกษการพยากรณ์พลังงานแสงอาทิตย์ในระยะสั้น โดยใช้ XGBoost ร่วมกับการพยากรณ์อากาศเชิงตัวเลข ซึ่งเป็นชุดข้อมูลขนาดใหญ่ ผู้วิจัยได้ทำการปรับขนาดแบบจำลองด้วยวิธี min-max normalization และใช้วิธีวิเคราะห์ห้วงค์ประกอบเพื่อปรับขนาดของข้อมูล ผลการพยากรณ์พบว่า การปรับขนาดข้อมูลให้เป็นมาตรฐาน ร่วมกับวิธีวิเคราะห์ห้วงค์ประกอบ ช่วยลดมิติของข้อมูลที่มีขนาดใหญ่ได้ดี และทำให้ค่าผลลัพธ์ในการพยากรณ์ของแบบจำลอง XGBoost มีประสิทธิภาพมากกว่าการพยากรณ์โดยไม่ถูกปรับขนาดของข้อมูล ผลการพยากรณ์งานวิจัยนี้ได้ทำการวัดค่าความคลาดเคลื่อนด้วยค่า RMSE เท่ากับ 8.621 ทั้งนี้การศึกษาในครั้งนี้จะใช้วิธีปรับขนาดของข้อมูลเพื่อให้อยู่ในลักษณะมาตรฐานและส่งผลกระทบต่อความเร็วในการพยากรณ์

Rodriguez-Leguizamon et al., (2023) ได้นำข้อมูลการผลิตพลังงานไฟฟ้าจากพลังงานแสงอาทิตย์ จำนวน 5 ปี มาสร้างแบบจำลองโดยใช้ XGBoost และ LSTM และเปรียบเทียบกับแบบจำลองการพยากรณ์ทางสถิติ SARIMA พบว่าแบบจำลอง XGBoost ให้ประสิทธิภาพค่า MAE เท่ากับ 0.8672 รองลงมาคือ LSTM ค่า MAE เท่ากับ 22.8835 และ SARIMA ค่า MEA เท่ากับ 31.9180 จากงานวิจัยนี้ชี้ให้เห็นว่าแบบจำลอง XGBoost มีประสิทธิภาพในการพยากรณ์มากกว่าวิธีการพยากรณ์ทางสถิติ และการเปรียบเทียบกับแบบจำลอง LSTM จึงมีความน่าสนใจที่จะนำสองแบบจำลองนี้มาใช้ในการวิจัยในครั้งนี้

Zhu et al., (2023) ได้นำเสนอแบบจำลองสำหรับใช้ในการพยากรณ์การเปลี่ยนแปลงของราคาหุ้น เพื่อลดความเสี่ยงให้นักลงทุนด้วยการรวบรวมข้อมูล และสร้างแบบจำลองแบบผสม โดยใช้แบบจำลองทางสถิติ ARIMA เพื่อหาค่าส่วนเหลือ จากนั้นนำข้อมูลเข้าสู่แบบจำลอง CNN-LSTM ผลลัพธ์ที่ได้จะถูกปรับแต่งโดยใช้แบบจำลอง XGBoost อย่างละเอียด ผลการศึกษาพบว่า MAE 0.38789 เท่ากับ, RMSE เท่ากับ 0.39917, MSE เท่ากับ 0.15933 และ R square เท่ากับ -4.02958

จากงานวิจัยของ Zhu และคณะ มีความน่าสนใจในการใช้แบบจำลองโครงข่ายประสาทเทียมคือ CNN-LSTM ที่มีความสามารถในการจำแนกคุณลักษณะของข้อมูลและพยากรณ์ข้อมูลลำดับเวลาได้ดีและช่วยเพิ่มประสิทธิภาพของแบบจำลองด้วย XGBoost จึงเป็นแนวทางในการประยุกต์ใช้แบบจำลอง Hybrid ในงานวิจัยครั้งนี้

ซึ่งสามารถสรุปงานวิจัยที่เกี่ยวข้อง แสดงดังตารางที่ 2.2

ตารางที่ 2.2 สรุปงานวิจัยที่เกี่ยวข้อง

กระบวนการทำงาน	งานวิจัย	หลักการ/วิธีการ
การคัดเลือกคุณลักษณะ	ภาณุพงศ์ น้ำแก้ว (2563) การพยากรณ์กำลังไฟฟ้าของระบบเซลล์แสงอาทิตย์แบบติดตั้งอุปกรณ์วัดด้วยโครงข่ายประสาทเทียม	ข้อมูลนำเข้าที่เหมาะสมต่อการพยากรณ์ ได้แก่ ค่าความเข้มของแสงอาทิตย์ อุณหภูมิบนแผง และกำลังไฟฟ้าของเซลล์แสงอาทิตย์วันก่อนพยากรณ์
	เกศินี ซ่อนกลิ่น (2564) แบบจำลองการพยากรณ์ราคาอนุพันธ์ โดยใช้เทคนิคโครงข่ายประสาทเทียมแบบสั่งวัตนาการ 1 มิติ หน่วยความจำระยะสั้นแบบยาวและซัพพอร์ตเวกตอร์รีเกรสชัน	ใช้วิธีสหสัมพันธ์แบบเพียร์สัน ในการคัดเลือกคุณลักษณะที่สำคัญเพื่อนำข้อมูลเข้าสู่แบบจำลอง
การปรับขนาดข้อมูล	Phan et al., (2021) Short-term Solar Power Forecasting Using XGBoost with Numerical Weather Prediction	ใช้วิธี min-max normalization และใช้วิธีวิเคราะห์องค์ประกอบเพื่อปรับขนาดของข้อมูล ที่มีค่าความแปรปรวนและมีขนาดใหญ่
การแบ่งข้อมูล	Foster et al., (2017) Adaptive Sliding Window Load Forecasting	การแบ่งข้อมูลด้วยวิธีหน้าต่างบานเลื่อนที่เหมาะสมกับชุดข้อมูลมากที่สุด
การสร้างแบบจำลอง	Obiora et al., (2021) Implementing Extreme Gradient Boosting (XGBoost) Algorithm in Predicting Solar Irradiance	ใช้อัลกอริทึม XGBoost และได้ผลการประเมินประสิทธิภาพที่สูง และเป็นกรวิจัยเกี่ยวกับด้านการพยากรณ์พลังงานแสงอาทิตย์
	Liu et al., (2021) A Simplified LSTM Neural Networks for One	ใช้อัลกอริทึม LSTM และได้ผลการประเมินประสิทธิภาพที่สูง

กระบวนการทำงาน	งานวิจัย	หลักการ/วิธีการ
	Day-Ahead Solar Power Forecasting	และเป็นการวิจัยเกี่ยวกับด้านการพยากรณ์พลังงานแสงอาทิตย์
	Obiora et al., (2022) Efficient Method of Finding the Best Time Interval for Predicting Short Term Solar Radiation Using CNN-LSTM Model	ใช้อัลกอริทึม CNN-LSTM และได้ผลการประเมินประสิทธิภาพที่สูง และเป็นการวิจัยเกี่ยวกับด้านการพยากรณ์พลังงานแสงอาทิตย์
	Rodriguez-Leguizamon et al., (2023) PV Power Generation Forecasting Based on XGBoost and LSTM Models	ใช้อัลกอริทึม XGBoost และได้ผลการประเมินประสิทธิภาพที่สูง เมื่อเทียบกับ LSTM, SARIMA และเป็นการวิจัยเกี่ยวกับด้านการพยากรณ์พลังงานแสงอาทิตย์
วิธีการพยากรณ์แบบผสม	อุรษา จันทรภา (2564) การพยากรณ์มูลค่าการส่งออกรถยนต์ อุปกรณ์และส่วนประกอบของประเทศไทย ด้วยวิธีอนุกรมเวลา	ใช้วิธีการสร้างแบบจำลองด้วยวิธีการแบบผสม โดยใช้แบบจำลองหลักในการหาค่าส่วนเหลือเพื่อเป็นข้อมูลนำเข้าของแบบพยากรณ์รอง
	Zhu et al., (2023) XGBoost and CNN-LSTM hybrid model with Attention-based stock prediction	ใช้วิธีการแบบผสมกับแบบจำลองทางสถิติ แบบจำลองการเรียนรู้เชิงลึก และแบบจำลองการเรียนรู้ของเครื่อง

## บทที่ 3

### วิธีดำเนินการศึกษา

วิธีการดำเนินการศึกษาในการสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์ บูลสต์ ดังนี้

1. ประชากรและกลุ่มตัวอย่าง
2. เครื่องมือที่ใช้ในการศึกษา
3. การศึกษาและวิเคราะห์ปัญหา
4. การเก็บรวบรวมข้อมูลและสร้างแบบจำลอง
5. การทดสอบประสิทธิภาพของแบบจำลอง

#### 1. ประชากรและกลุ่มตัวอย่าง

ชุดข้อมูลการผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำแห่งหนึ่ง ขนาดกำลังการผลิต 45 เมกะวัตต์ ตั้งแต่เดือน กุมภาพันธ์ ถึง เดือนตุลาคม พ.ศ. 2566 จำนวน 6,551 ระเบียบ (Records) และ 11 คุณลักษณะ (Features)

#### 2. เครื่องมือที่ใช้ในการวิจัย

2.1. โปรแกรมตารางคำนวณ (Microsoft Office Excel)

2.2. เครื่องคอมพิวเตอร์พกพา

2.2.1. ระบบปฏิบัติการ Windows 11 Home 64 bit

2.2.2. หน่วยประมวลผล CPU Core i5

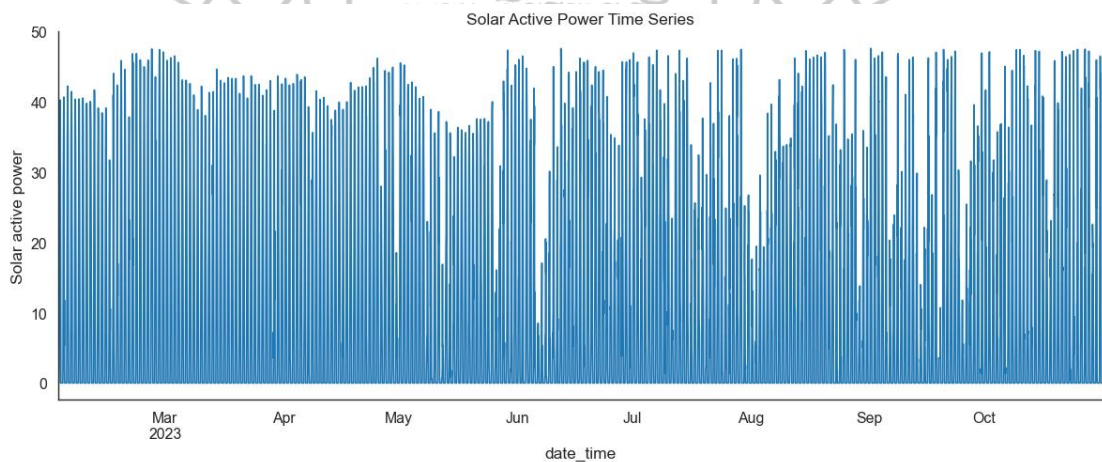
2.2.3. หน่วยความจำหลัก (RAM) 16 GB

2.3. เครื่องมือ **Anaconda 3** และ Jupyter Notebook เป็น IDE ที่รองรับภาษาการเขียน

โปรแกรม Python จำเป็นต้องดาวน์โหลดและติดตั้งโปรแกรมในเครื่องคอมพิวเตอร์ รวมถึงการติดตั้งสภาพแวดล้อมของเครื่อง (Environment) และ Library ที่จำเป็นก่อนเริ่มการใช้งาน

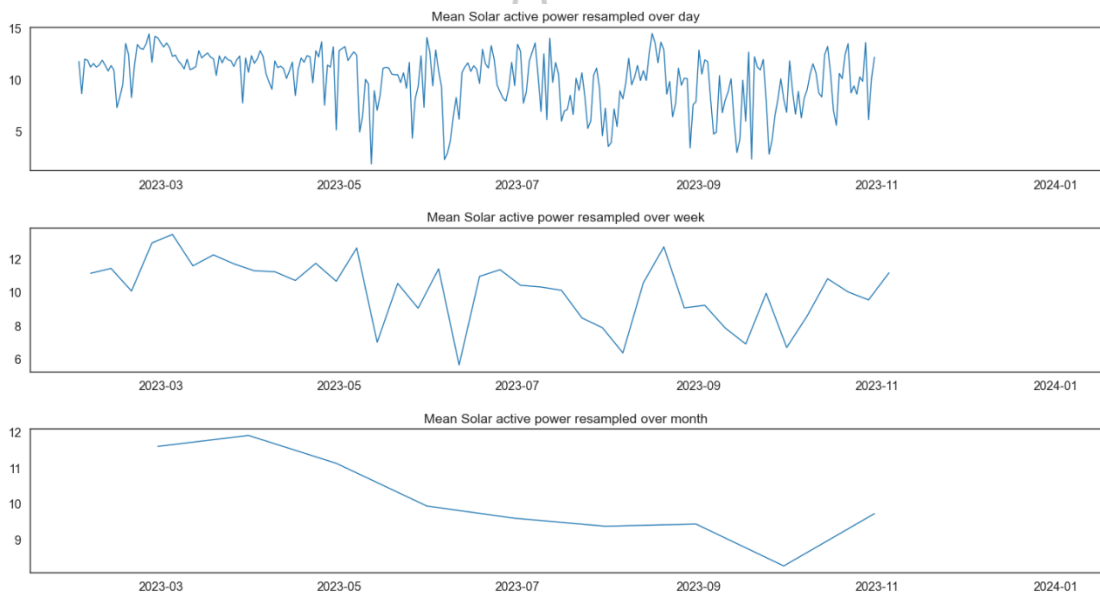
### 3. การศึกษาและวิเคราะห์ปัญหา

จากศักยภาพพื้นที่ของประเทศไทย ซึ่งเป็นพื้นที่ที่มีปริมาณความเข้มของพลังงานแสงอาทิตย์สูงเพียงพอ ทำให้การก่อสร้างและพัฒนาโรงไฟฟ้าพลังงานหมุนเวียนด้วยพลังงานแสงอาทิตย์เติบโตอย่างรวดเร็ว มีการได้ดำเนินการก่อสร้างและพัฒนาโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ บนพื้นที่อ่างเก็บน้ำในพื้นที่ภาคตะวันออกเฉียงเหนือ ขนาดกำลังการผลิต 45 เมกะวัตต์ ถือว่าเป็นโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำแห่งใหม่ ที่มีความสามารถในการผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ ช่วยเสริมความมั่นคงทางพลังงานไฟฟ้าให้กับระบบส่งจ่ายกำลังไฟฟ้าที่เชื่อมโยงกับระบบไฟฟ้าภายในประเทศ แต่ข้อจำกัดที่สำคัญของโรงไฟฟ้าพลังงานแสงอาทิตย์ คือ ค่ากำลังไฟฟ้าที่ผลิตได้มีความผันผวน เนื่องจากปัจจัยด้านสภาพอากาศที่มีความไม่คงที่ เช่น ความเข้มแสงอาทิตย์ อุณหภูมิสภาพอากาศ อุณหภูมิบนแผงเซลล์แสงอาทิตย์ ค่าความชื้น ความเร็วลม ทิศทางลม ซึ่งเป็นปัจจัยที่ไม่สามารถควบคุมและหลีกเลี่ยงได้ ส่งผลกระทบต่อคุณภาพและเสถียรภาพของระบบการผลิตพลังงานไฟฟ้า หากมีระบบการพยากรณ์พลังงานแสงอาทิตย์ที่มีความน่าเชื่อถือ จะช่วยให้การวางแผนบริหารจัดการกำลังไฟฟ้าในระบบมีประสิทธิภาพและเสถียรภาพดีขึ้น จากการเก็บรวบรวมข้อมูลของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำแห่งนี้ พบว่าปริมาณข้อมูลที่ถูกบันทึกบนฐานข้อมูลที่มีความสมบูรณ์พร้อมสำหรับการนำมาใช้ในการสร้างแบบจำลองในการพยากรณ์ สามารถรวบรวมได้จำนวน 9 เดือน หรือคิดเป็นจำนวน 6,551 ระเบียบ (Record) และ 11 คุณลักษณะ (Feature)



ภาพที่ 3.1 ปริมาณค่ากำลังไฟฟ้าที่ผลิตได้จากพลังงานแสงอาทิตย์

จากภาพที่ 3.1 กราฟแสดงค่ากำลังไฟฟ้าที่ผลิตได้จากพลังงานแสงอาทิตย์ ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ จำนวน 9 เดือน ในปี พ.ศ. 2566 จะสังเกตว่า ในช่วงเดือน พฤษภาคม ถึงเดือน ตุลาคม ค่ากำลังไฟฟ้ามีความผันผวนและปริมาณลดลงต่ำสุดในบางช่วง เนื่องจากประเทศไทยเข้าสู่ช่วงฤดูฝน เมื่อเทียบระหว่าง เดือน กุมภาพันธ์ ถึงเดือน เมษายน ที่ปริมาณการผลิตผันผวนน้อยกว่า ทำให้ค่ากำลังไฟฟ้ามีปริมาณมากกว่า แสดงดังภาพที่ 3.2



ภาพที่ 3.2 ค่ากำลังไฟฟ้าพลังงานแสงอาทิตย์เฉลี่ยแบ่งกลุ่มตามวัน สัปดาห์ และเดือน

จากภาพที่ 3.2 กราฟแสดงค่าเฉลี่ยกำลังไฟฟ้าพลังงานแสงอาทิตย์ เพื่อดูแนวโน้มของค่ากำลังไฟฟ้าที่ผลิต กราฟบนแสดงแนวโน้มของค่าเฉลี่ยพลังงานแสงอาทิตย์รายวัน จากลักษณะของกราฟพบว่าในแต่ละวันค่ากำลังไฟฟ้ามีการขึ้นลงตามรอบเวลาจากแสงอาทิตย์ ความเสถียรของการผลิตจะอยู่ในช่วงเดือนกุมภาพันธ์ ถึงเดือน เมษายน และมีความผันผวนตั้งแต่เดือนพฤษภาคม ถึงเดือนตุลาคม

กราฟกลางแสดงแนวโน้มของค่าเฉลี่ยพลังงานแสงอาทิตย์รายสัปดาห์ พบว่าค่ากำลังไฟฟ้ามีความผันผวนและผลิตได้ต่ำสุดในช่วงปลายสัปดาห์ของเดือนมิถุนายน เดือนสิงหาคม และเดือนตุลาคม

กราฟล่างแสดงแนวโน้มของค่าเฉลี่ยพลังงานแสงอาทิตย์รายเดือน พบว่าแนวโน้มค่ากำลังไฟฟ้าผลิตได้สูงสุดจะอยู่ในช่วงเดือนกุมภาพันธ์ และมีแนวโน้มลดลงจนถึงเดือนตุลาคมและเพิ่มขึ้นในเดือนพฤศจิกายน

#### 4. การเก็บรวบรวมข้อมูลและสร้างแบบจำลอง

แบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง แบ่งออกเป็น 5 ขั้นตอน แสดงดังภาพที่ 3.3 ประกอบด้วย

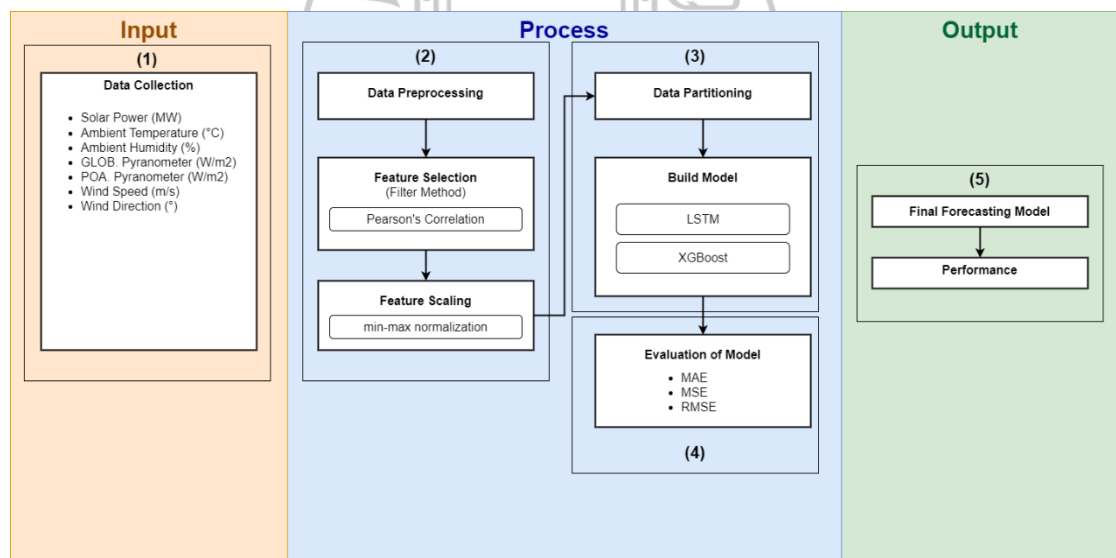
ขั้นตอนที่ 1 การเก็บข้อมูล โดยใช้ชุดข้อมูลการผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำบนฐานข้อมูลของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์

ขั้นตอนที่ 2 การเตรียมข้อมูล ได้แก่ การแปลงข้อมูล การคัดเลือกคุณลักษณะด้วยการหาค่าสหสัมพันธ์เพียร์สัน (Pearson's Correlation) การปรับค่าให้เป็นมาตรฐานด้วยวิธี min-max normalization

ขั้นตอนที่ 3 แบ่งข้อมูลและสร้างแบบจำลอง แบ่งข้อมูลสำหรับชุดฝึกสอน ชุดทดสอบ และชุดทดสอบ จากนั้นนำข้อมูลเข้าสู่แบบจำลอง LSTM แบบจำลอง XGBoost

ขั้นตอนที่ 4 ประเมินประสิทธิภาพของแบบจำลอง ประเมินประสิทธิภาพของแบบจำลองด้วยวิธีการวัดค่าความคลื่อน MSE และ RMSE

ขั้นตอนที่ 5 แบบจำลองขั้นสุดท้ายและการประเมินประสิทธิภาพ



ภาพที่ 3.3 ขั้นตอนการวิจัย



**4.1. การเก็บข้อมูล** เป็นการเก็บรวบรวมชุดข้อมูลเพื่อนำเข้าสู่กระบวนการสร้างแบบจำลองพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยใช้ข้อมูลค่ากำลังไฟฟ้าที่ผลิตได้และข้อมูลจากอุปกรณ์เซนเซอร์ที่อยู่บนฐานข้อมูลของโรงไฟฟ้า ตั้งแต่เดือนกุมภาพันธ์ ถึงเดือนตุลาคม พ.ศ. 2566 จำนวน 6,511 ระเบียบ (Record) และ 11 คุณลักษณะ (Feature) จัดเก็บในรูปแบบไฟล์ CSV แสดงตัวอย่างชุดข้อมูล แสดงดังตารางที่ 3.4

Date	Time	Solar_active_power	Obliquity_radiation	Environment_temperature	Environment_humidity	Wind_speed	Wind_direction	Panel_temperature_1	Panel_temperature_2	Panel_temperature_3
0	1/2/2023 1:00:00	0.000000	0	21.1	90	0	44	18.6	18.3	18.0
1	1/2/2023 2:00:00	0.000000	0	21.1	92	0	13	18.6	18.3	17.9
2	1/2/2023 3:00:00	0.000000	0	21.0	92	0	335	18.4	18.1	17.6
3	1/2/2023 4:00:00	0.000000	0	20.1	90	0	343	17.8	17.5	17.2
4	1/2/2023 5:00:00	0.000000	0	20.5	84	4	206	18.6	18.3	18.2
5	1/2/2023 6:00:00	0.000000	0	20.1	85	15	172	18.2	17.8	17.6
6	1/2/2023 7:00:00	2.441149	62	20.9	84	4	206	20.0	19.2	18.9
7	1/2/2023 8:00:00	13.145468	293	21.6	80	3	234	30.1	26.9	25.7
8	1/2/2023 9:00:00	25.093370	536	22.8	78	4	230	41.5	38.8	39.2
9	1/2/2023 10:00:00	32.221780	707	23.2	82	8	297	49.8	46.0	47.2
10	1/2/2023 11:00:00	32.703540	723	26.5	61	7	234	57.8	52.5	54.4
11	1/2/2023 12:00:00	40.265860	917	27.1	57	3	281	60.6	56.1	59.0
12	1/2/2023 13:00:00	39.213320	896	30.0	59	10	195	64.2	58.7	60.5
13	1/2/2023 14:00:00	34.447290	780	29.7	50	12	247	59.6	54.0	54.0
14	1/2/2023 15:00:00	28.231760	640	30.4	49	13	191	55.0	49.7	49.4
15	1/2/2023 16:00:00	17.459290	399	30.2	53	13	232	45.6	40.8	39.9
16	1/2/2023 17:00:00	5.899177	149	29.3	59	9	215	36.5	32.4	31.6

ภาพที่ 3.4 ตัวอย่างชุดข้อมูลในรูปแบบไฟล์ CSV

โดยรายละเอียดของคุณลักษณะ ได้แก่ วันที่ เวลา ค่ากำลังไฟฟ้า ค่าความเข้มแสงอาทิตย์ อุณหภูมิสภาพแวดล้อม ค่าความชื้น ความเร็วลม ทิศทางลม อุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 1 อุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 2 และอุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 3 แสดงดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดของคุณลักษณะ

คุณลักษณะ	รายละเอียด	ช่วงข้อมูล	หน่วยวัด
Date	วันที่	1/2/2023 - 31/10/2023	-
Time	เวลา	00:00 - 23:00	-
Solar_active_power	ค่ากำลังไฟฟ้า	0.00 - 47.61961	เมกะวัตต์ (MW)
Obliquity_radiation	ค่าความเข้มแสงอาทิตย์	0.00 - 1321.00	วัตต์ต่อตารางเมตร (W/m <sup>2</sup> )

คุณลักษณะ	รายละเอียด	ช่วงข้อมูล	หน่วยวัด
Environment_temperature	อุณหภูมิ สภาพแวดล้อม	16.5 - 37.1	องศาเซลเซียส (°C)
Environment_humidity	ค่าความชื้น	82.0 - 99.0	ร้อยละ (%)
Win_speed	ความเร็วลม	0.00 - 42.00	เมตรต่อวินาที (m/s)
Wind_direction	ทิศทางลม	0.00 - 360	องศา (Degree)
Panel_temperature_1	อุณหภูมิต่อแผง เซลล์แสงอาทิตย์ ชุดที่ 1	12.9 - 74.1	องศาเซลเซียส (°C)
Panel_temperature_2	อุณหภูมิต่อแผง เซลล์แสงอาทิตย์ ชุดที่ 2	10.4 - 69.3	องศาเซลเซียส (°C)
Panel_temperature_3	อุณหภูมิต่อแผง เซลล์แสงอาทิตย์ ชุดที่ 3	10.0 - 100	องศาเซลเซียส (°C)

ดำเนินการนำเข้าสู่ข้อมูลในโปรแกรม Jupyter Notebook และใช้ไลบรารี Scikit-learn เพื่ออ่านไฟล์ชุดข้อมูล CSV แสดงรหัสโปรแกรมดังตารางที่ 3.2 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.3 และแสดงผลการประมวลผลรหัสโปรแกรมดังภาพที่ 3.5

ตารางที่ 3.2 รหัสโปรแกรมนำเข้าสู่ชุดข้อมูล CSV

บรรทัดที่	รหัสโปรแกรม
1	import numpy as np
2	import pandas as pd
3	df = pd.read_csv('data/PV02-10-2023.csv')
4	print('Number of rows and columns:', df.shape)
5	df.head(10)

### ตารางที่ 3.3 การอธิบายรหัสโปรแกรมนำเข้าชุดข้อมูล CSV

บรรทัดที่	คำอธิบาย
1	นำเข้าไลบรารี numpy สำหรับจัดการข้อมูลประเภทตัวเลข การคำนวณทางคณิตศาสตร์ โดยเรียกอ้างอิงแทนในชื่อ np
2	นำเข้าไลบรารี pandas สำหรับกำหนดและวิเคราะห์ข้อมูลในรูปแบบตาราง โดยเรียกอ้างอิงแทนในชื่อ pd
3	อ่านข้อมูลจากไฟล์ PV02-10-2023.csv โดยเก็บข้อมูลไว้ในตัวแปร df
4	แสดงผลข้อมูลแสดงแถวและคอลัมน์
5	แสดงข้อมูล 10 รายการแรก

Number of rows and columns: (6551, 11)

	Date	Time	Solar_active_power	Obliquity_radiation	Environment_temperature	Environment_humidity	Wind_speed	Wind_direction	Panel_temperature_1	Panel
0	1/2/2023	1:00:00	0.000000	0	21.1	90	0	44	18.6	
1	1/2/2023	2:00:00	0.000000	0	21.1	92	0	13	18.6	
2	1/2/2023	3:00:00	0.000000	0	21.0	92	0	335	18.4	
3	1/2/2023	4:00:00	0.000000	0	20.1	90	0	343	17.8	
4	1/2/2023	5:00:00	0.000000	0	20.5	84	4	206	18.6	
5	1/2/2023	6:00:00	0.000000	0	20.1	85	15	172	18.2	
6	1/2/2023	7:00:00	2.441149	62	20.9	84	4	206	20.0	
7	1/2/2023	8:00:00	13.145468	293	21.6	80	3	234	30.1	
8	1/2/2023	9:00:00	25.093370	536	22.8	78	4	230	41.5	
9	1/2/2023	10:00:00	32.221780	707	23.2	82	8	297	49.8	

### ภาพที่ 3.5 ตัวอย่างการอ่านชุดข้อมูล

จากภาพที่ 3.5 แสดงตัวอย่างข้อมูลการผลิตพลังงานแสงอาทิตย์ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ จากการอ่านชุดข้อมูล PV02-10-2023.csv โดยแสดงรายการข้อมูล 10 Record และ 11 Feature

## 4.2. การเตรียมข้อมูล

**4.2.1. การแปลงข้อมูล** โดยทำการทำความสะอาดข้อมูลเพื่อตัดข้อมูลที่ไม่ว่างคั่นกันออก ข้อมูลที่ขาดหายไป (Missing Values) จัดกลุ่มของคุณลักษณะ โดยใช้โปรแกรม Microsoft Office Excel และทำการแปลงข้อมูลซึ่งเป็นการทำให้ข้อมูลอยู่ในรูปแบบที่พร้อมนำเอา

ไปใช้ในการวิเคราะห์ข้อมูลตามอัลกอริทึมที่เลือกไว้ แสดงรหัสโปรแกรมดังตารางที่ 3.4 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.5 และแสดงผลการประมวลผลรหัสโปรแกรมดังภาพที่ 3.6

ตารางที่ 3.4 รหัสโปรแกรมการแปลงข้อมูล

บรรทัดที่	รหัสโปรแกรม
1	<code>df['Datetime'] = pd.to_datetime(df['Date'] + ' ' + df['Time'], format='%d/%m/%Y %H:%M:%S')</code>
2	<code>df= df.set_index('Datetime')</code>
3	<code>df = df.drop(['Date','Time'], axis=1)</code>
4	<code>df.head()</code>

ตารางที่ 3.5 การอธิบายรหัสโปรแกรมการแปลงข้อมูล

บรรทัดที่	คำอธิบาย
1	รวมคอลัมน์ Date และคอลัมน์ Time โดยจัดรูปแบบวันที่ให้อยู่ในรูปแบบที่ถูกต้อง โดยสร้างคอลัมน์ใหม่ที่ชื่อ df['Datetime']
2	ตั้งค่าคอลัมน์ Datetime ให้เป็น index
3	ลบคอลัมน์ Date และ Time เดิมทิ้ง
4	แสดงรายการข้อมูล

Datetime	Solar_active_power	Obliquity_radiation	Environment_temperature	Environment_humidity	Wind_speed	Wind_direction	Panel_temperature_1	Panel_temperatur
2023-02-01 01:00:00	0.0	0	21.1	90	0	44	18.6	1
2023-02-01 02:00:00	0.0	0	21.1	92	0	13	18.6	1
2023-02-01 03:00:00	0.0	0	21.0	92	0	335	18.4	1
2023-02-01 04:00:00	0.0	0	20.1	90	0	343	17.8	1

ภาพที่ 3.6 ตัวอย่างข้อมูลที่ทำให้การแปลงข้อมูล

จากภาพที่ 3.6 แสดงตัวอย่างข้อมูลที่ทำให้การแปลงข้อมูลของคุณลักษณะ Feature วันที่ และเวลาให้อยู่ในรูปแบบที่ถูกต้องและรวมเป็นคอลัมน์เดียว จากนั้นตั้งค่าคอลัมน์ Datetime ให้เป็น index ของตารางข้อมูล

**4.2.2. การคัดเลือกคุณลักษณะ** เนื่องจากชุดข้อมูลมีลักษณะเป็นตัวเลข (numerical) และผลลัพธ์เป้าหมาย (Target result) เป็นตัวเลข (numerical) ผู้วิจัยจึงเลือกใช้การคัดเลือกคุณลักษณะด้วยวิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation) เพื่อหาคุณลักษณะที่สัมพันธ์กัน โดยผู้วิจัยได้ใช้เกณฑ์การคัดเลือกข้อมูลที่มีค่าตั้งแต่ 0.5 – 1 จึงจะถือว่าคุณลักษณะนั้นมีความสัมพันธ์กัน แสดงรหัสโปรแกรมดังตารางที่ 3.6 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.7 และแสดงผลการประมวลผลรหัสโปรแกรมดังภาพที่ 3.7

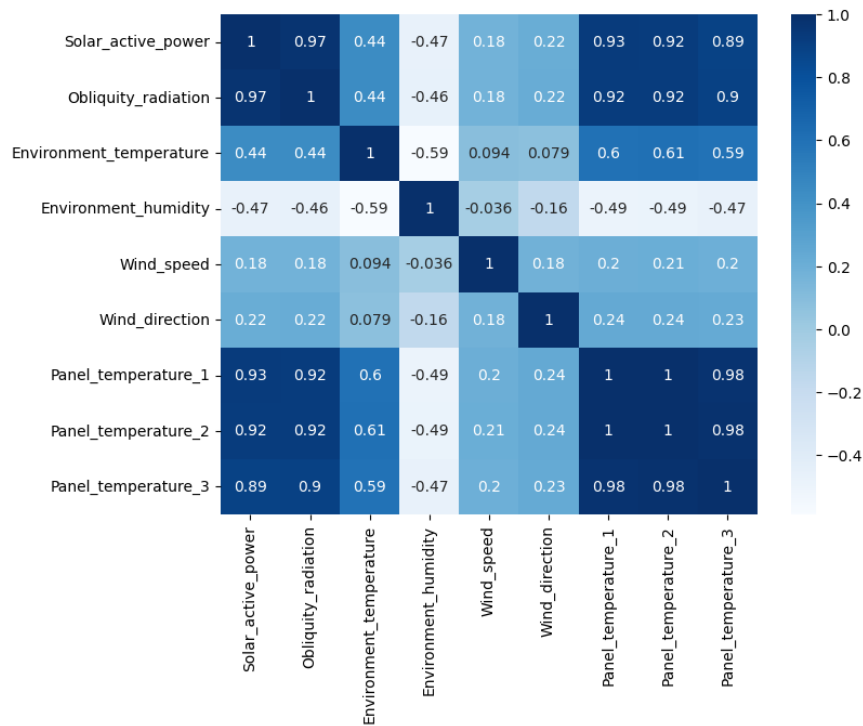
ตารางที่ 3.6 รหัสโปรแกรมการเลือกคุณลักษณะ

บรรทัดที่	รหัสโปรแกรม
1	import seaborn as sns
2	import matplotlib.pyplot as plt
3	import plotly.graph_objects as go
4	corr = df.corr()
5	plt.figure(figsize=(8,6))
6	sns.heatmap(corr, cmap="Blues",annot=True)
7	plt.show()

ตารางที่ 3.7 การอธิบายรหัสโปรแกรมการเลือกคุณลักษณะ

บรรทัดที่	คำอธิบาย
1	นำเข้าไลบรารี seaborn สำหรับการแสดงผลข้อมูล โดยเรียกอ้างอิงแทนในชื่อ sns
2	นำเข้าไลบรารี matplotlib.pyplot สำหรับการแสดงผลข้อมูลด้วยกราฟ โดยเรียกอ้างอิงแทนในชื่อ plt

บรรทัดที่	คำอธิบาย
3	นำเข้าไลบรารี plotly.graph_objects สำหรับการแสดงผลข้อมูลด้วยกราฟ โดยเรียกอ้างอิงแทนในชื่อ go
4	สร้างตัวแปร corr เพื่อเรียกใช้วิธีสหสัมพันธ์แบบเพียร์สัน
5	ตั้งค่าขนาดกราฟ 8×6 พิกเซล
6	แสดงผลข้อมูลด้วย heatmap โดยกำหนดค่าเป็น Blue และกำหนดการแสดงค่าข้อมูลเป็น true
7	แสดงผล heatmap



ภาพที่ 3.7 ผลการหาค่าความสัมพันธ์ของข้อมูล

จากภาพที่ 3.7 แสดงค่าความสัมพันธ์ของข้อมูล ด้วยวิธีสหสัมพันธ์เพียร์สัน พบว่าคุณลักษณะที่สัมพันธ์กับค่ากำลังไฟฟ้า โดยมีค่ามากกว่า 0.5 ได้แก่ ค่าความเข้มแสงอาทิตย์ อุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 1 - 3 โดยจะถูกเลือกสำหรับการสร้างแบบจำลองต่อไป แสดงรหัสโปรแกรมดังตารางที่ 3.8 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.9 และแสดงผลการประมวลผลรหัสโปรแกรมดังภาพที่ 3.8

ตารางที่ 3.8 รหัสโปรแกรมการคัดเลือกคุณลักษณะที่ต้องการนำไปใช้งาน

บรรทัดที่	รหัสโปรแกรม
1	<code>features_list=['Solar_active_power', 'Obliquity_radiation','Panel_temperature_1','Panel_temperature_2','Panel_temperature_3']</code>
2	<code>df_used = df[features_list]</code>
3	<code>df_used.head()</code>

ตารางที่ 3.9 การอธิบายรหัสโปรแกรมคัดเลือกคุณลักษณะที่ต้องการนำไปใช้งาน

บรรทัดที่	คำอธิบาย
1	เลือกคุณลักษณะที่ต้องการใช้งานเก็บไว้ในตัวแปร <code>features_list</code>
2	นำตัวแปรคุณลักษณะเก็บไว้ในตัวแปร <code>df_used</code>
3	แสดงผลข้อมูล

	Solar_active_power	Obliquity_radiation	Panel_temperature_1	Panel_temperature_2	Panel_temperature_3
Datetime					
2023-02-01 01:00:00	0.0	0	18.6	18.3	18.0
2023-02-01 02:00:00	0.0	0	18.6	18.3	17.9
2023-02-01 03:00:00	0.0	0	18.4	18.1	17.6
2023-02-01 04:00:00	0.0	0	17.8	17.5	17.2
2023-02-01 05:00:00	0.0	0	18.6	18.3	18.2

ภาพที่ 3.8 ผลการหาเลือกคุณลักษณะที่ต้องการนำไปใช้งาน

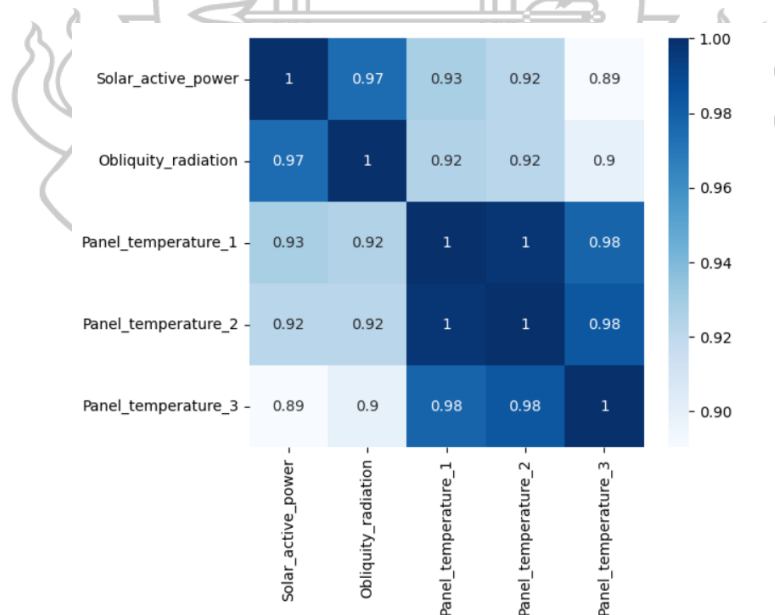
จากภาพที่ 3.7 แสดงผลการหาเลือกคุณลักษณะที่ต้องการนำไปใช้งาน เพื่อยืนยันความถูกต้องแสดงค่าความสัมพันธ์ด้วย Heatmap อีกครั้ง แสดงรหัสโปรแกรมดังตารางที่ 3.10 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.11 และแสดงผลการประมวลผลรหัสโปรแกรม ดังภาพที่ 3.9

ตารางที่ 3.10 รหัสโปรแกรมการแสดงผล Heatmap

บรรทัดที่	รหัสโปรแกรม
1	<code>corr = df.corr()</code>
2	<code>plt.figure(figsize=(8,6))</code>
3	<code>sns.heatmap(corr, cmap="Blues",annot=True)</code>
4	<code>plt.show()</code>

ตารางที่ 3.11 การอธิบายรหัสโปรแกรมการแสดงผล Heatmap

บรรทัดที่	คำอธิบาย
1	สร้างตัวแปร corr เพื่อเรียกใช้วิธีหสัมพันธ์แบบเพียร์สัน
2	ตั้งค่าขนาดกราฟ 8×6 พิกเซล
3	แสดงผลข้อมูลด้วย heatmap โดยกำหนดค่าเป็น Blue และกำหนดการแสดงค่าข้อมูลเป็น true
4	แสดงผล heatmap



ภาพที่ 3.9 การแสดงผลภาพ Heatmap ของข้อมูลที่เลือกใช้งาน



**4.2.3. การปรับค่าให้เป็นมาตรฐาน (Scaling Data)** เนื่องจากแต่ละ feature มีช่วงข้อมูลที่แตกต่างกันอย่างมาก สังเกตได้จากค่าอหุณุมิบนแวง เทียบกับค่าความเข้มแสงอาทิตย์ที่มีช่วงข้อมูลกว้างกว่า เมื่อพบว่าข้อมูลส่วนใหญ่ที่เป็นตัวเลขมีความแตกต่างจากข้อมูลตัวเลขอื่น จำเป็นต้องมีการปรับขนาดข้อมูลให้สมมาตร ผู้วิจัยใช้วิธีการทำให้เป็นมาตรฐาน (Normalization) เป็นการแปลงค่าข้อมูลให้อยู่ในช่วงสั้น ๆ ตั้งแต่ 0 – 1 หรือที่เรียกว่าการทำให้เป็นมาตรฐานต่ำสุด-สูงสุด (min-max normalization) โดยแสดงรหัสโปรแกรมดังตารางที่ 3.12 อธิบายรหัสโปรแกรมดังตารางที่ 3.13

ตารางที่ 3.12 รหัสโปรแกรมการปรับค่าให้เป็นมาตรฐาน

บรรทัดที่	รหัสโปรแกรม
1	<code>from sklearn.preprocessing import MinMaxScaler</code>
2	<code>X = df_used.drop(['Solar_active_power'], axis = 1)</code>
3	<code>y= df_used[['Solar_active_power']]</code>
4	<code>scaler_x = MinMaxScaler()</code>
5	<code>scaler_x = scaler.fit(X)</code>
6	<code>X_norm = scaler_x.transform(X)</code>
7	<code>scaler_y = MinMaxScaler()</code>
8	<code>scaler_y.fit(y)</code>
9	<code>y_norm = scaler_y.transform(y)</code>

ตารางที่ 3.13 การอธิบายรหัสโปรแกรมการปรับค่าให้เป็นมาตรฐาน

บรรทัดที่	คำอธิบาย
1	เรียกชุดไลบรารี sklearn.preprocessing นำเข้าไลบรารี MinMaxScaler
2	เก็บข้อมูลที่ยกเว้นคุณลักษณะผลลัพธ์ (result) ไว้ในตัวแปร X
3	เก็บข้อมูลเฉพาะคุณลักษณะผลลัพธ์ (result) ไว้ในตัวแปร y
4	สร้างตัวแปร scaler_x เพื่อเก็บไลบรารี MinMaxScaler()
5	คำนวณค่าเฉลี่ยและมาตรฐานเพื่อใช้สำหรับการปรับขนาดด้วย scaler.fit(X) เก็บในตัวแปร scaler_x

บรรทัดที่	คำอธิบาย
6	สร้างตัวแปร X_norm เพื่อเก็บข้อมูลการแปลงข้อมูลตัวแปร X ให้สมมาตรด้วย scaler_x.transform(X)
7	สร้างตัวแปร scaler_y เพื่อเก็บไลบรารี MinMaxScaler()
8	คำนวณค่าเฉลี่ยและมาตรฐานเพื่อใช้สำหรับการปรับขนาดด้วย scaler.fit(y) เก็บในตัวแปร scaler_y
9	สร้างตัวแปร y_norm เพื่อเก็บข้อมูลการแปลงข้อมูลตัวแปร y ให้สมมาตรด้วย scaler_y.transform(y)

### 4.3. แบ่งข้อมูลและสร้างแบบจำลอง

**4.3.1. แบ่งข้อมูลสำหรับการสร้างแบบจำลอง** ผู้วิจัยใช้วิธีการแบ่งชุดข้อมูลออกเป็น 2 วิธี คือ แบ่งข้อเป็นชุดฝึกสอน ชุดตรวจสอบ และชุดทดสอบ 80:10:10 และแบ่งข้อมูลด้วยวิธีหน้าต่างบานเลื่อน (Sliding Window) ทุก ๆ 24 ค่า เพื่อพยากรณ์ค่าถัดไป (t+1) แสดงรหัสโปรแกรมดังตารางที่ 3.14 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.15 และแสดงผลการประมวลผลรหัสโปรแกรม ดังภาพที่ 3.10

ตารางที่ 3.14 รหัสโปรแกรมการแบ่งข้อมูลสำหรับการสร้างแบบจำลอง

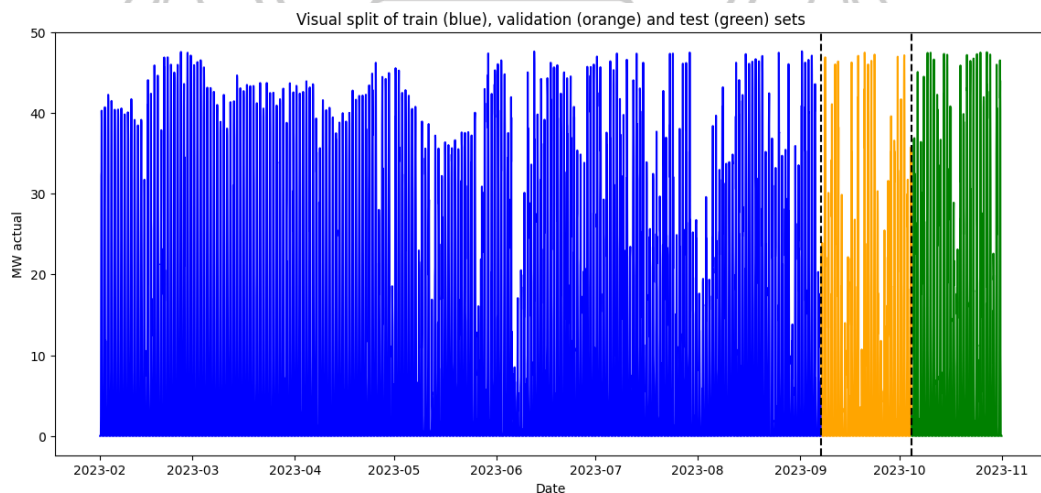
บรรทัดที่	รหัสโปรแกรม
1	train_cutoff = int(0.8*X_norm.shape[0])
2	val_cutoff = int(0.9*X_norm.shape[0])
3	def windowing(X_input,y_input, history_size): data = [] labels = [] for i in range(history_size, len(y_input)): data.append(X_input[i - history_size : i, :]) labels.append(y_input[i]) return np.array(data), np.array(labels).reshape(-1,1)
4	hist_size= 24

บรรทัดที่	รหัสโปรแกรม
5	<code>data_final = np.concatenate((X_norm,y_norm), axis = 1)</code>
6	<code>X_train, y_train = windowing(data_final[:train_cutoff,:],data_final[:train_cutoff,-1], hist_size)</code>
7	<code>X_val, y_val = windowing(data_final[train_cutoff :val_cutoff,:],data_final[train_cutoff:val_cutoff,-1], hist_size)</code>
8	<code>X_test, y_test = windowing(data_final[val_cutoff :,:],data_final[val_cutoff:-1], hist_size)</code>
9	<code>fig, axes = plt.subplots(figsize = (14,6))</code>
10	<code>axes.plot(df_used['Solar_active_power'].iloc[:train_cutoff], color = 'blue')</code>
11	<code>axes.plot(df_used['Solar_active_power'].iloc[train_cutoff + 1 : val_cutoff], color = 'orange')</code>
12	<code>axes.plot(df_used['Solar_active_power'].iloc[val_cutoff + 1 :], color = 'green')</code>
13	<code>axes.axvline(x=df_used.index[train_cutoff], color='black', linestyle='--')</code>
14	<code>axes.axvline(x=df_used.index[val_cutoff], color='black', linestyle='--')</code>
15	<code>axes.set_title('Visual split of train (blue), validation (orange) and test (green) sets')</code>
16	<code>axes.set_xlabel('Date')</code>
17	<code>axes.set_ylabel('MW actual')</code>
18	<code>plt.show()</code>

ตารางที่ 3.15 การอธิบายรหัส โปรแกรมการแบ่งข้อมูลสำหรับการสร้างแบบจำลอง

บรรทัดที่	คำอธิบาย
1	แบ่งข้อมูลชุดฝึกสอน 80% เก็บไว้ในตัวแปร <code>train_cutoff</code>
2	แบ่งข้อมูลชุดตรวจสอบ 10% เก็บไว้ในตัวแปร <code>val_cutoff</code>
3	ฟังก์ชัน <code>windowing</code> การวนลูปเพื่อสร้างชุดข้อมูลสำหรับการทำ Sliding Window
4	ค่าของข้อมูลที่ต้องการใช้พยากรณ์ย้อนหลัง 24 ชั่วโมง เก็บไว้ในตัวแปร <code>hist_size</code>
5	รวมข้อมูลชุด <code>X_norm</code> และ <code>y_norm</code> เก็บไว้ในตัวแปร <code>data_final</code>

บรรทัดที่	คำอธิบาย
6	แบ่งชุดข้อมูลสำหรับฝึกสอนด้วยฟังก์ชันหน้าต่างบานเลื่อน windowing เก็บไว้ในตัวแปร X_train และตัวแปร y_train
7	แบ่งชุดข้อมูลสำหรับตรวจสอบด้วยฟังก์ชันหน้าต่างบานเลื่อน windowing เก็บไว้ในตัวแปร X_val และตัวแปร y_val
8	แบ่งชุดข้อมูลสำหรับทดสอบด้วยฟังก์ชันหน้าต่างบานเลื่อน windowing เก็บไว้ในตัวแปร X_test และตัวแปร y_test
9	กำหนดขนาดภาพกราฟที่จะแสดงผลขนาด 14×6
10	กำหนดค่าการแสดงผลของกราฟสำหรับชุดฝึกสอน
11	กำหนดค่าการแสดงผลของกราฟสำหรับชุดทดสอบ
12	กำหนดค่าการแสดงผลของกราฟสำหรับชุดตรวจสอบ
13	กำหนดลักษณะเส้นแบ่งข้อมูล
14	กำหนดลักษณะเส้นแบ่งข้อมูล
15	กำหนดข้อความที่ใช้แสดงผล
16	กำหนดข้อความแสดงผลแกน X
17	กำหนดข้อความแสดงผลแกน Y
18	แสดงผลกราฟข้อมูลที่ถูกแบ่ง



ภาพที่ 3.10 ผลการแบ่งข้อมูลสำหรับสร้างแบบจำลอง

จากภาพที่ 3.10 แสดงกราฟการแบ่งข้อมูลออกเป็นส่วน ๆ สำหรับฝึกสอนแบบจำลอง และให้แบบจำลองจดจำรูปแบบข้อมูล โดยแบ่งเป็น กราฟสีน้ำเงิน คือ ชุดข้อมูลการฝึกสอน (training data) คิดเป็น 80% สำหรับฝึกสอนแบบจำลองให้เรียนรู้จำรูปแบบข้อมูลจากข้อมูลชุดฝึกสอน กราฟสีส้ม คือ ชุดข้อมูลการตรวจสอบ (validation data) คิดเป็น 10% สำหรับการทดสอบแบบจำลองกับข้อมูลที่ไม่เคยเห็นมาก่อน และกราฟสีเขียว คือ ชุดข้อมูลการทดสอบ (testing data) คิดเป็น 10% สำหรับการทดสอบการทำนายผลการพยากรณ์ของแบบจำลองที่ผ่านการฝึกสอนแล้ว

**4.3.2. สร้างแบบจำลอง โดยใช้อัลกอริทึม Long Short-Term Memory (LSTM)**  
โดยใช้ไลบรารี tensorflow keras และอัลกอริทึม eXtream Gradient Boosting (XGBoost) โดยใช้ไลบรารี XGBRegressor จากนั้นนำเข้าสู่ชุดข้อมูลที่แบ่งจากขั้นตอนที่ 4.3.1 เพื่อนำชุดข้อมูลไปทำการเรียนรู้ แสดงรหัสโปรแกรมดังตารางที่ 3.16 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.17

ตารางที่ 3.16 รหัสโปรแกรมการสร้างแบบจำลอง

บรรทัดที่	รหัสโปรแกรม
1	import math
2	import tensorflow
3	from tensorflow.keras.models import Sequential
4	from tensorflow.keras.layers import LSTM, Dense, Dropout, Flatten
5	from tensorflow.keras.optimizers import Adam
6	from tensorflow.keras.callbacks import EarlyStopping
7	epoch = 100
8	batch_size = 64
9	callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=50)
10	optimizer = 'Adam'
11	def base_model_lstm(): model = Sequential() model.add(LSTM(units = 32, return_sequences = True, activation="relu", input_shape = X_train.shape[-2:])) model.add(Flatten())

บรรทัดที่	รหัสโปรแกรม
	<pre> model.add(Dense(units=128, activation='relu')) model.add(Dropout(0.1)) model.add(Dense(1)) return model lstm_model = base_model_lstm() lstm_model.compile(optimizer = optimizer , loss = 'mean_absolute_error') lstm_model.summary() </pre>
12	<pre> history = lstm_model.fit(X_train, y_train, validation_data = (X_val, y_val), epochs = epoch, batch_size = batch_size, callbacks=[callback]) y_pred = lstm_model.predict(X_test) </pre>
13	<pre> lstm_pred = y_pred </pre>
14	<pre> def reshaping(X):     reshaped_x = X.reshape(-1 , X.shape[1] * X.shape[2])     return reshaped_x X_train_xgb = reshaping(X_train) X_val_xgb = reshaping(X_val) X_test_xgb = reshaping(X_test) </pre>
15	<pre> import xgboost as xgb </pre>
16	<pre> xgb_model = xgb.XGBRegressor(eval_metric='mae', early_stopping_rounds=20, colsample_bytree= 0.7, learning_rate=0.01, max_depth=3, n_estimators=1000) </pre>
17	<pre> xgb_model.fit(X_train_xgb, y_train, eval_set=[(X_train_xgb, y_train), X_val_xgb, y_val]), verbose=False) y_pred = xgb_model.predict(X_test_xgb) </pre>
18	<pre> xgb_pred = y_pred </pre>
19	<pre> y_test_inv = scaler_y.inverse_transform(y_test) </pre>

ตารางที่ 3.17 การอธิบายรหัสโปรแกรมการสร้างแบบจำลอง

บรรทัดที่	คำอธิบาย
1	นำเข้าไลบรารี math
2	นำเข้าไลบรารี tensorflow
3	นำเข้าไลบรารี Sequential
4	นำเข้าไลบรารี LSTM, Dense, Dropout และ Flatten
5	นำเข้าไลบรารี optimizers Adam
6	นำเข้าไลบรารี EarlyStopping
7	กำหนดรอบ (epoch) การฝึกสอนแบบจำลองจำนวน 100 รอบ
8	กำหนดขนาด batch size เท่ากับ 64
9	สร้างตัวแปร callback สำหรับเก็บข้อมูลการสั่งให้แบบจำลองหยุดการเรียนรู้
10	ใช้เครื่องมือเพิ่มประสิทธิภาพ Adam เก็บไว้ในตัวแปร optimizer
11	สร้างฟังก์ชัน base_model_lstm เพื่อกำหนดโครงสร้างของแบบจำลอง LSTM
12	สร้างตัวแปร history เก็บข้อมูลการเรียนรู้แบบจำลองและพยากรณ์ผลแบบจำลอง
13	สร้างตัวแปร lstm_pred เก็บผลการพยากรณ์
14	สร้างฟังก์ชันการปรับสเกลข้อมูลเพื่อเรียนรู้ X_train_xgb ข้อมูลตรวจสอบ X_val_xgb และข้อมูลทดสอบ X_test_xgb
15	นำเข้าไลบรารี xgboost โดยเรียกอ้างอิงแทนในชื่อ xgb
16	สร้างตัวแปร xgb_model เพื่อเรียกใช้ไลบรารี XGBRegressor กำหนด parameter ฟังก์ชันการสูญเสีย และรอบหยุดการเรียนรู้ขั้นต่ำ
17	เรียนรู้แบบจำลอง และทำนายผลแบบจำลอง
18	สร้างตัวแปร xgb_pred เก็บผลการพยากรณ์
19	ปรับสเกลค่าของ y_test เก็บไว้ที่ตัวแปร y_test_inv

4.4. ประเมินประสิทธิภาพของแบบจำลอง โดยเปรียบเทียบประสิทธิภาพของแบบจำลอง จากการนำชุดข้อมูลการผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยใช้ไลบรารี Scikit-learn แสดงรหัสโปรแกรมหาดังตารางที่ 3.18 โดยอธิบายรหัสโปรแกรมหาดังตารางที่ 3.19 และแสดงผลการประมวลผลรหัสโปรแกรมหาดังภาพที่ 3.11 - 3.12

ตารางที่ 3.18 รหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง

บรรทัดที่	รหัสโปรแกรม
1	<code>from sklearn.metrics import mean_absolute_error, mean_squared_error</code>
2	<code>mae_lstm = mean_absolute_error(lstm_pred,y_test)</code> <code>mse_lstm = mean_squared_error(lstm_pred,y_test)</code> <code>rmse_lstm = np.sqrt(mse_lstm)</code>
3	<code>print(f'LSTM MAE for test set : {round(mae_lstm,4)}')</code> <code>print(f'LSTM MSE for test set : {round(mse_lstm,4)}')</code> <code>print(f'LSTM RMSE for test set : {round(rmse_lstm,4)}')</code> <code>y_pred_actual = scaler_y.inverse_transform(y_pred)</code>
4	<code>y_pred_actual = scaler_y.inverse_transform(xgb_pred.reshape(-1,1))</code>
5	<code>mae_xgb = mean_absolute_error(xgb_pred,y_test)</code> <code>mse_xgb = mean_squared_error(xgb_pred,y_test)</code> <code>rmse_xgb = np.sqrt(mse_xgb)</code>
6	<code>print(f'XGBoost MAE for test set : {round(mae_xgb,4)}')</code> <code>print(f'XGBoost MSE for test set : {round(mse_xgb,4)}')</code> <code>print(f'XGBoost RMSE for test set : {round(rmse_xgb,4)}')</code>

ตารางที่ 3.19 การอธิบายรหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลอง

บรรทัดที่	คำอธิบาย
1	นำเข้าไลบรารีสำหรับการประเมินประสิทธิภาพแบบจำลอง
2	สร้างตัวแปร <code>mae_lstm</code> เพื่อเก็บผลการคำนวณ MAE สร้างตัวแปร <code>mse_lstm</code> เพื่อเก็บผลการคำนวณ MSE สร้างตัวแปร <code>rmse_lstm</code> เพื่อเก็บผลการคำนวณ RMSE
3	แสดงผลการประเมินแบบจำลอง LSTM
4	แสดงผลการพยากรณ์แบบจำลอง LSTM ด้วยกราฟ
5	สร้างตัวแปร <code>mae_xgb</code> เพื่อเก็บผลการคำนวณ MAE



บรรทัดที่	คำอธิบาย
	สร้างตัวแปร mse_xgb เพื่อเก็บผลการคำนวณ MSE
	สร้างตัวแปร rmse_xgb เพื่อเก็บผลการคำนวณ RMSE
6	แสดงผลการประเมินแบบจำลอง XGBoost

```
LSTM MAE for test set : 0.0585
LSTM MSE for test set : 0.0148
LSTM RMSE for test set : 0.1215
```

### ภาพที่ 3.11 ผลการประเมินประสิทธิภาพแบบจำลอง LSTM ด้วยชุดข้อมูลทดสอบ

จากภาพที่ 3.10 แสดงผลการประเมินประสิทธิภาพแบบจำลอง LSTM ด้วยชุดข้อมูลทดสอบ โดยใช้วิธีวัดค่าความเคลื่อนด้วย Mean Absolute Error (MAE), Mean Squared Error (MSE) และ Root Mean Squared Error (RMSE)

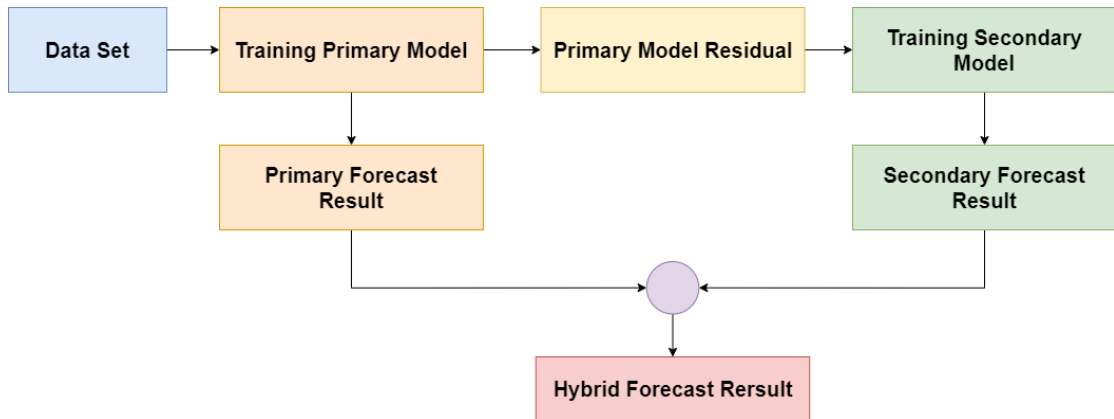
```
XGBoost MAE for test set : 0.063
XGBoost MSE for test set : 0.0147
XGBoost RMSE for test set : 0.1211
```

### ภาพที่ 3.12 ผลการประเมินประสิทธิภาพแบบจำลอง XGBoost ด้วยชุดข้อมูลทดสอบ

จากภาพที่ 3.11 แสดงผลการประเมินประสิทธิภาพแบบจำลอง XGBoost ด้วยชุดข้อมูลทดสอบ โดยใช้วิธีวัดค่าความเคลื่อนด้วย Mean Absolute Error (MAE), Mean Squared Error (MSE) และ Root Mean Squared Error (RMSE)

## 4.5. แบบจำลองขั้นสุดท้ายและการประเมินประสิทธิภาพ

**4.5.1.** แบบจำลองการพยากรณ์ด้วยวิธีพยากรณ์แบบผสม เพื่อเพิ่มประสิทธิภาพให้กับแบบจำลองขั้นสุดท้าย วิธีการพยากรณ์แบบผสมจึงถูกนำมาใช้ในขั้นตอนนี้ แสดงขั้นตอนการทำงานของแบบจำลองขั้นสุดท้ายด้วยวิธีการพยากรณ์แบบผสม แสดงดังภาพที่ 3.13 แสดงรหัสโปรแกรมดังตารางที่ 3.20 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.21



ภาพที่ 3.13 ขั้นตอนการทำงานของแบบจำลองขั้นสุดท้ายด้วยวิธีการพยากรณ์แบบผสม

ตารางที่ 3.20 รหัสโปรแกรมแบบจำลองวิธีการพยากรณ์แบบผสม

บรรทัดที่	รหัสโปรแกรม
1	<pre> def base_model_lstm():     model = Sequential()     model.add(LSTM(units = 32, return_sequences = True, activation="relu", input_shape = X_train.shape[-2:]))     model.add(Flatten())     model.add(Dense(units=128, activation='relu'))     model.add(Dropout(0.1))     model.add(Dense(1))      return model  lstm_hybrid_model = base_model_lstm()  lstm_hybrid_model.compile(optimizer = 'Adam' , loss = 'mean_absolute_error') </pre>
2	<pre> lstm_hybrid_model.fit(X_train, y_train, validation_data = (X_val, y_val), epochs = epoch, batch_size = batch_size, callbacks=[callback]) </pre>
3	<pre> residuals_train = y_train - lstm_hybrid_model.predict(X_train) </pre>
4	<pre> residuals_val = y_val - lstm_hybrid_model.predict(X_val) </pre>
5	<pre> xgb_hybrid = xgb.XGBRegressor(eval_metric='mae', </pre>

บรรทัดที่	รหัสโปรแกรม
	<pre>early_stopping_rounds=20, colsample_bytree= 0.7, learning_rate=0.01, max_depth=3, n_estimators=1000)</pre>
6	<code>xgb_hybrid.fit(X_train_xgb, residuals_train, eval_set=[(X_train_xgb, residuals_train), (X_val_xgb, residuals_val)], verbose=False)</code>
7	<code>y_train_pred = lstm_hybrid_model.predict(X_train) + xgb_hybrid.predict(X_train_xgb).reshape(-1,1)</code>
8	<code>y_val_pred = lstm_hybrid_model.predict(X_val) + xgb_hybrid.predict(X_val_xgb).reshape(-1,1)</code>

ตารางที่ 3.21 การอธิบายรหัสโปรแกรมแบบจำลองวิธีการพยากรณ์แบบผสม

บรรทัดที่	คำอธิบาย
1	สร้างฟังก์ชัน <code>base_model_lstm</code> เพื่อกำหนดโครงสร้างของแบบจำลอง LSTM
2	เรียนรู้แบบจำลองและพยากรณ์ผลแบบจำลอง
3	สร้างตัวแปร <code>residuals_train</code> เพื่อเก็บค่าส่วนเหลือจากชุดข้อมูลการฝึกสอนแบบจำลอง
4	สร้างตัวแปร <code>residuals_val</code> เพื่อเก็บค่าส่วนเหลือจากชุดข้อมูลตรวจสอบแบบจำลอง
5	สร้างตัวแปร <code>xgb_hybrid</code> เพื่อเรียกใช้ไลบรารี <code>XGBRegressor</code> กำหนด parameter ฟังก์ชันการสูญเสีย และรอบหยุดการเรียนรู้ขั้นต่ำ
6	เรียนรู้แบบจำลอง และทำนายผลแบบจำลอง โดยข้อมูลนำเข้าเป็นค่าส่วนเหลือจากแบบจำลอง LSTM
7	สร้างตัวแปร <code>y_train_pred</code> เก็บค่าพยากรณ์จากชุดข้อมูลฝึกสอน
8	สร้างตัวแปร <code>y_val_pred</code> เก็บค่าพยากรณ์จากชุดข้อมูลตรวจสอบ

4.5.2. การประเมินประสิทธิภาพของแบบจำลองวิธีการพยากรณ์แบบผสม โดยใช้ไลบรารี Scikit-learn แสดงรหัสโปรแกรมดังตารางที่ 3.22 โดยอธิบายรหัสโปรแกรมดังตารางที่ 3.23 และแสดงผลการประมวลผลรหัสโปรแกรมดังภาพที่ 3.14

ตารางที่ 3.22 รหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลองวิธีการพยากรณ์แบบผสม

บรรทัดที่	รหัสโปรแกรม
1	<code>y_test_pred = lstm_hybrid_model.predict(X_test) + xgb_hybrid.predict(X_test_xgb).reshape(-1,1)</code>
2	<code>hybrid_pred = y_test_pred</code>
3	<code>mae_hybrid = mean_absolute_error(y_test_pred,y_test) mse_hybrid = mean_squared_error(y_test_pred,y_test) rmse_hybrid = np.sqrt(mse_hybrid)</code>
4	<code>mae_hybrid_val = mean_absolute_error(y_val_pred,y_val) mse_hybrid_val = mean_squared_error(y_val_pred,y_val) rmse_hybrid_val = np.sqrt(mse_hybrid)</code>
5	<code>print(f'LSTM-XGBoost MAE for test set : {round(mae_hybrid,4)}') print(f'LSTM-XGBoost MSE for test set : {round(mse_hybrid,4)}') print(f'LSTM-XGBoost RMSE for test set : {round(rmse_hybrid,4)}')</code>

ตารางที่ 3.23 การอธิบายรหัสโปรแกรมการประเมินประสิทธิภาพของแบบจำลองวิธีการพยากรณ์แบบผสม

บรรทัดที่	คำอธิบาย
1	สร้างตัวแปร y_test_pred เก็บผลการพยากรณ์
2	สร้างตัวแปร hybrid_pred เพื่อเก็บผลการพยากรณ์
3	สร้างตัวแปร mae_hybrid เพื่อเก็บผลการคำนวณ MAE สร้างตัวแปร mse_hybrid เพื่อเก็บผลการคำนวณ MSE สร้างตัวแปร rmse_hybrid เพื่อเก็บผลการคำนวณ RMSE

บรรทัดที่	คำอธิบาย
4	สร้างตัวแปร mae_hybrid_val เพื่อเก็บผลการคำนวณ MAE สร้างตัวแปร mse_hybrid_val เพื่อเก็บผลการคำนวณ MSE สร้างตัวแปร rmse_hybrid_val เพื่อเก็บผลการคำนวณ RMSE
5	แสดงผลการประเมินแบบจำลองวิธีการพยากรณ์แบบผสม

LSTM-XGBoost MAE for test set : 0.0577  
LSTM-XGBoost MSE for test set : 0.0143  
LSTM-XGBoost RMSE for test set : 0.1196

ภาพที่ 3.14 ผลการประเมินประสิทธิภาพของแบบจำลองวิธีการพยากรณ์แบบผสม

## 5. การทดสอบประสิทธิภาพของแบบจำลอง

จากการนำชุดข้อมูลการผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำเข้าสู่กระบวนการสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตตึง และทดสอบประสิทธิภาพของแบบจำลองด้วยวิธีการวัดประสิทธิภาพของแบบจำลองการถดถอย ได้แก่ การวัดค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) การหาค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และการหาค่ารากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) ซึ่งเป็นเครื่องมือที่ได้รับความนิยมในการนำมาประเมินประสิทธิภาพของแบบจำลองแบบ Regression ซึ่งการประเมินประสิทธิภาพ คำนวณจากสมการ ดังนี้

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (3-1)$$

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (3-2)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (3-3)$$

โดยที่  $y_t$  คือ ค่าจริงที่  $t$  ใด ๆ  
 $\hat{y}_t$  คือ ค่าที่ได้จากการพยากรณ์ที่  $t$  ใด  
 $n$  คือ ขนาดข้อมูล

จากสมการ MAE, MSE และ RMSE เป็นการประเมินความสามารถของแบบจำลองที่ได้ผ่านการฝึกสอนแล้ว โดยการทดสอบประสิทธิภาพของแบบจำลองจะถูกตัดสินด้วยความแม่นยำของผลลัพธ์ที่ได้จากแบบจำลอง หากค่าความผิดพลาดจากค่าจริงน้อย คือ มีค่าเข้าใกล้ 0 แสดงว่าแบบจำลองสามารถทำนายได้ถูกต้องและมีประสิทธิภาพที่สูง



## บทที่ 4

### ผลการวิเคราะห์ข้อมูล

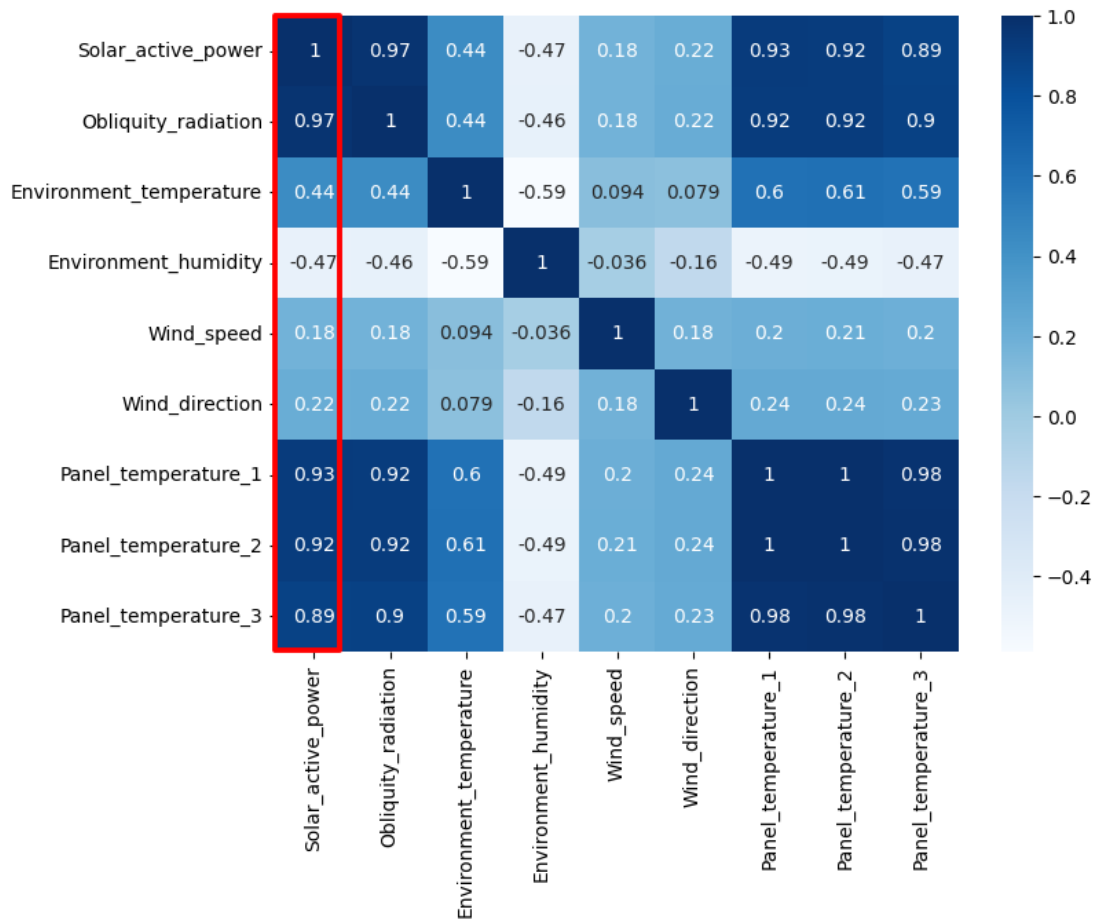
ในการศึกษาและสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง ซึ่งใช้ข้อมูลการผลิตพลังงานไฟฟ้าและข้อมูลจากเซนเซอร์ที่บันทึกอยู่บนฐานข้อมูลของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ ข้อมูลถูกเก็บรวบรวม ตั้งแต่เดือนกุมภาพันธ์ ถึงเดือนตุลาคม พ.ศ. 2566 ผลการวิเคราะห์โดยการประเมินประสิทธิภาพของแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ด้วยค่าเฉลี่ยของผลต่างสัมบูรณ์ (Mean Absolute Error: MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (Mean Squared Error: MSE) และรากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (Root Mean Square Error: RMSE)

ผู้วิจัยได้ดำเนินการวิจัยโดยการศึกษาตามขบวนการและขั้นตอนต่าง ๆ ตลอดจนการวัดประสิทธิภาพ เพื่อให้บรรลุจุดประสงค์ของการวิจัยที่ได้กำหนดไว้ โดยมีรายละเอียดดังนี้

1. ผลลัพธ์ของแบบจำลอง โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว
2. ผลลัพธ์ของแบบจำลอง โดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง
3. ผลลัพธ์ของแบบจำลอง โดยอาศัยวิธีการพยากรณ์แบบผสม
4. ประเมินประสิทธิภาพของแบบจำลองทั้ง 3 วิธี

#### 1. ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว

จากการสร้างแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) โดยใช้ข้อมูลการกำลังไฟฟ้าที่ผลิตได้และข้อมูลจากเซนเซอร์ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ จำนวน 6,551 ระเบียบ 11 คุณลักษณะ โดยใช้วิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation) เพื่อคัดเลือกคุณลักษณะที่สัมพันธ์กัน โดยใช้เกณฑ์การคัดเลือกข้อมูลที่มีค่าตั้งแต่ 0.5 – 1 จึงจะถือว่าคุณลักษณะนั้นมีความสัมพันธ์กัน แสดงดังภาพที่ 4.1



ภาพที่ 4.1 แสดงผลการคัดเลือกคุณลักษณะด้วยวิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation)

จากภาพที่ 4.1 แสดงผลการคัดเลือกคุณลักษณะด้วยวิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation) พบว่าคุณลักษณะที่ประกอบด้วยค่ากำลังไฟฟ้า (Solar\_active\_power) ค่าความเข้มแสงอาทิตย์ (Obliquity\_radiation) อุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 1 (Panel\_temperature\_1) อุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 2 (Panel\_temperature\_2) และอุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 3 (Panel\_temperature\_3) เป็นคุณลักษณะที่มีความสัมพันธ์กับค่ากำลังไฟฟ้า จึงถูกใช้เป็นข้อมูลนำเข้าในการสร้างแบบจำลองทั้ง 3 วิธี

ข้อมูลได้ถูกแบ่งสำหรับการฝึกสอนแบบจำลอง ออกเป็น 3 ชุด ได้แก่ ชุดข้อมูลฝึกสอน (Training data) เท่ากับ 80% ชุดข้อมูลตรวจสอบ (Validation data) เท่ากับ 10% และชุดข้อมูลทดสอบ (Testing data) เท่ากับ 10% และแบ่งข้อมูลด้วยวิธีหน้าต่างบานเลื่อน (Sliding Window) เท่ากับ 24 ชั่วโมง จากนั้นสร้างแบบจำลอง โดยมีไฮเปอร์พารามิเตอร์ที่ถูกใช้ในแบบจำลอง LSTM แสดงดังตารางที่ 4.1



ตารางที่ 4.1 ไฮเปอร์พารามิเตอร์ที่ใช้กับแบบจำลองหน่วยความจำระยะสั้นแบบยาว

Hyperparameter	LSTM
Model	Sequential
Activation	Relu
Input	Input Array
Hidden Node	32 neurons
Hidden Layer	(Dropout)
Output	Output Array
Loss Calculation	MAE
Optimizer	Adam

จากตารางที่ 4.1 แบบจำลองได้กำหนดจำนวน Hidden Node เท่ากับ 32 จำนวน Hidden Layer คือ Dropout ฟังก์ชันกระตุ้น (Activation Function) คือ Relu ใช้ Optimizer คือ Adam กำหนด Loss Function คือ Mean Absolute Error (MAE) โดยกำหนดให้ Epochs เท่ากับ 100 ฟังก์ชัน Callback Early Stopping เท่ากับ 50 และ Batch size เท่ากับ 32

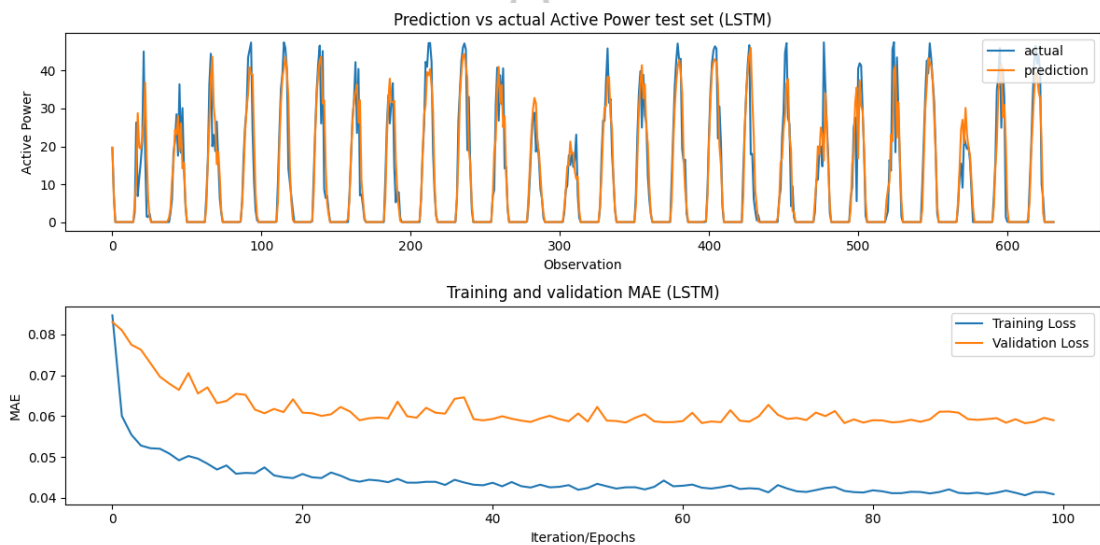
ผลลัพธ์การพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากการสร้างแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว (LSTM) แสดงดังตารางที่ 4.2 ผลลัพธ์จากการวัดประสิทธิภาพของแบบจำลองด้วยค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และรากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) แสดงดังตารางที่ 4.3 และผลการพยากรณ์ แสดงดังภาพที่ 4.2

ตารางที่ 4.2 ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง LSTM

Period	Actual Value	Forecast Value	Error
t+1	0.00	0.01	-0.01
t+2	0.00	0.01	-0.01
t+3	0.00	0.01	-0.01
t+4	0.00	0.01	-0.01
t+5	0.00	0.01	-0.01
t+6	0.00	0.01	-0.01
t+7	0.00	0.01	-0.01
t+8	0.00	0.03	-0.03
t+9	2.57	4.85	-2.28
t+10	26.32	10.66	15.66
t+11	6.85	28.77	-21.93
t+12	11.63	19.70	-8.08
t+13	15.74	19.39	-3.65
t+14	21.07	22.03	-0.96
t+15	45.06	26.58	18.48
t+16	17.50	36.85	-19.35
t+17	1.44	17.53	-16.08
t+18	1.34	6.10	-4.76
t+19	2.18	1.23	0.95
t+20	0.00	0.01	-0.01
t+21	0.00	0.01	-0.01
t+22	0.00	0.01	-0.01
t+23	0.00	0.01	-0.01
t+24	0.00	0.01	-0.01

ตารางที่ 4.3 ผลการประเมินประสิทธิภาพแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว

Methods	MAE	MSE	RMSE
LSTM	0.0585	0.0148	0.1215



ภาพที่ 4.2 ผลการพยากรณ์ของแบบจำลอง LSTM ด้วยชุดข้อมูลทดสอบ

จากภาพที่ 4.2 กราฟบน กำหนดเส้นสีน้ำเงิน คือ ค่ากำลังไฟฟ้าจริง และเส้นสีส้ม คือ ค่ากำลังไฟฟ้าจากการพยากรณ์ จากชุดข้อมูลทดสอบ 632 ชั่วโมง พบว่าแบบจำลองที่สร้างโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว พบว่า ค่าพยากรณ์ที่ได้มีความใกล้เคียงกับค่ากำลังไฟฟ้าจริงตั้งแต่ชั่วโมงที่ 0 - ชั่วโมงที่ 632

กราฟล่าง กำหนดเส้นสีน้ำเงิน คือ ค่าสูญเสียจากข้อมูลชุดฝึกสอน (Training Loss) และเส้นสีส้ม คือ ค่าสูญเสียจากชุดข้อมูลการตรวจสอบ (Validation Loss) จากการเรียนรู้แต่ละรอบด้วย Loss Function พบว่า เมื่อมีการฝึกสอนแบบจำลอง ค่า Training Loss มีแนวโน้มลดลง เช่นเดียวกับ ค่า Validation Loss แต่จะพบช่องระหว่างค่าทั้ง 2

## 2. ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสต์

จากการสร้างแบบจำลองโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสต์ (eXtream Gradient Boosting: XGBoost) โดยใช้ข้อมูลการกำลังไฟฟ้าที่ผลิตได้และข้อมูลจากเซนเซอร์ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ ขนาดกำลังการผลิต 45 เมกะวัตต์ จำนวน 6,551 ระเบียบ 11 คุณลักษณะ ใช้วิธีสหสัมพันธ์เพียร์สัน (Pearson's Correlation) คัดเลือกคุณลักษณะที่สัมพันธ์กัน ผลการคัดเลือกนำเข้าไปประกอบด้วยค่ากำลังไฟฟ้า ค่าความเข้มแสงอาทิตย์ และอุณหภูมิบนแผงเซลล์แสงอาทิตย์ชุดที่ 1 – 3 โดยแบ่งข้อมูลสำหรับฝึกสอนแบบจำลอง จำนวน 3 ชุด ได้แก่ ชุดข้อมูลฝึกสอน (Train) เท่ากับ 80% ชุดข้อมูลตรวจสอบ (Validation) เท่ากับ 10% และชุดข้อมูลทดสอบ (Test) เท่ากับ 10% และแบ่งข้อมูลด้วยวิธีหน้าต่างบานเลื่อน (Sliding Window) เท่ากับ 24 ชั่วโมง จากนั้นสร้างแบบจำลอง XGBoost โดยกำหนดไฮเปอร์พารามิเตอร์จากงานวิจัยของ (Li et al., 2019) และ (Obiora et al., 2021) แสดงดังตารางที่ 4.4

ตารางที่ 4.4 ไฮเปอร์พารามิเตอร์ที่ใช้กับแบบจำลองเอ็กซ์ตรีมเกรเดียนต์บูสต์

Class	Parameter	Value
General Parameters	Default	-
Booster Parameters	colsample_bytree	0.7
Booster Parameters	learning_rate	0.01
Booster Parameters	max_depth	3
Booster Parameters	n_estimators	1000
Early Stopping	early_stopping_rounds	20
Learning Task Parameters	eval_metric	MAE

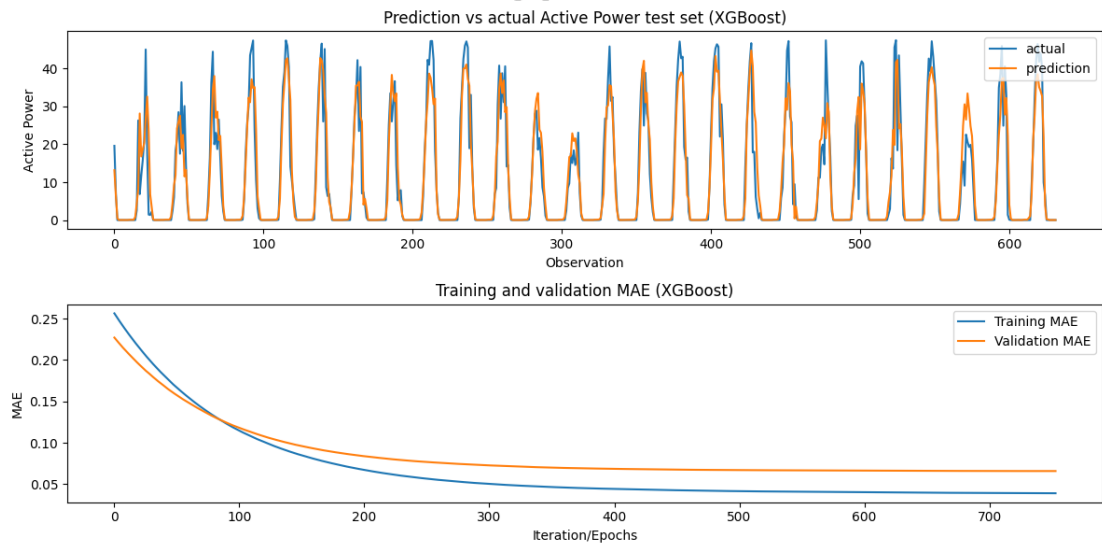
ผลลัพธ์การพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากการสร้างแบบจำลองโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสต์ (XGBoost) แสดงดังตารางที่ 4.5 ผลลัพธ์จากการทดสอบประสิทธิภาพของแบบจำลองด้วยค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และรากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) แสดงดังตารางที่ 4.6 และผลการพยากรณ์ แสดงดังภาพที่ 4.3

ตารางที่ 4.5 ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง XGBoost

Period	Actual Value	Forecast Value	Error
t+1	0.00	0.01	-0.01
t+2	0.00	0.01	-0.01
t+3	0.00	0.01	-0.01
t+4	0.00	0.06	-0.06
t+5	0.00	0.06	-0.06
t+6	0.00	0.07	-0.07
t+7	0.00	0.08	-0.08
t+8	0.00	0.84	-0.84
t+9	2.57	4.38	-1.81
t+10	26.32	8.28	18.04
t+11	6.85	28.15	-21.30
t+12	11.63	16.79	-5.16
t+13	15.74	18.54	-2.79
t+14	21.07	20.34	0.73
t+15	45.06	26.82	18.24
t+16	17.50	32.62	-15.12
t+17	1.44	23.52	-22.07
t+18	1.34	6.79	-5.45
t+19	2.18	3.52	-1.34
t+20	0.00	0.21	-0.21
t+21	0.00	0.01	-0.01
t+22	0.00	0.01	-0.01
t+23	0.00	0.01	-0.01
t+24	0.00	0.01	-0.01

ตารางที่ 4.6 ผลการประเมินประสิทธิภาพแบบจำลองโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสตีง

Methods	MAE	MSE	RMSE
XGBoost	0.0630	0.0147	0.1211



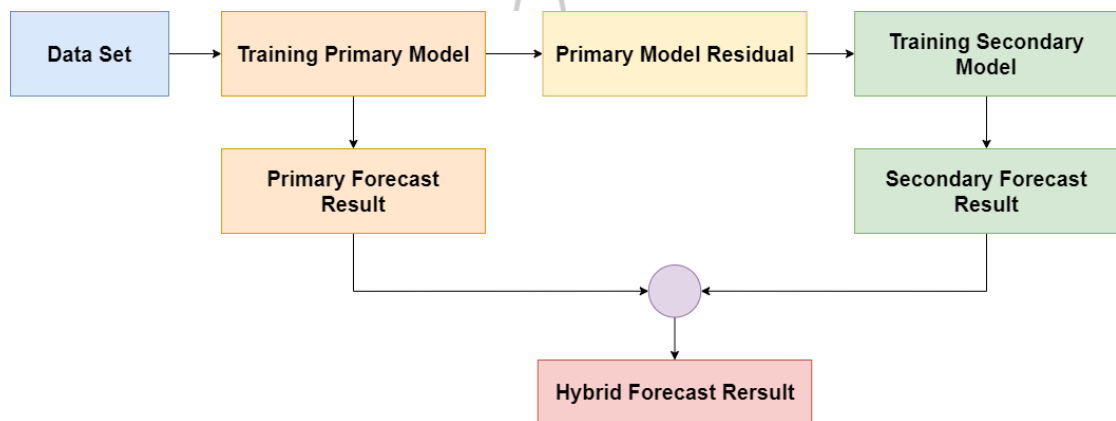
ภาพที่ 4.3 ผลการพยากรณ์ของแบบจำลอง XGBoost ด้วยชุดข้อมูลทดสอบ

จากภาพที่ 4.3 กราฟบน กำหนดเส้นสีน้ำเงิน คือ ค่ากำลังไฟฟ้าจริง และเส้นสีส้ม คือ ค่ากำลังไฟฟ้าจากการพยากรณ์ จากชุดข้อมูลทดสอบ 632 ชั่วโมง พบว่าแบบจำลองที่สร้างโดยอาศัยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสตีง พบว่า ค่าพยากรณ์ที่ได้มีความใกล้เคียงกับค่ากำลังไฟฟ้าจริง ตั้งแต่ชั่วโมงที่ 100 - ชั่วโมงที่ 200

กราฟล่าง กำหนดเส้นสีน้ำเงิน คือ ค่า MAE จากข้อมูลชุดฝึกสอน (Training MAE) และเส้นสีส้ม คือ ค่า MAE จากชุดข้อมูลการตรวจสอบ (Validation MAE) จากการเรียนรู้แต่ละรอบของแบบจำลอง พบว่า เมื่อมีการฝึกสอนแบบจำลอง ค่า Training MAE มีแนวโน้มลดลง เช่นเดียวกับ ค่า Validation MAE แต่จะพบช่องว่างระหว่างค่าทั้ง 2

### 3. ผลลัพธ์ของแบบจำลองโดยอาศัยวิธีการพยากรณ์แบบผสม

จากการสร้างแบบจำลองโดยอาศัยวิธีการพยากรณ์แบบผสม ด้วยหน่วยจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง (LSTM-XGBoost) เพื่อเพิ่มประสิทธิภาพการพยากรณ์ด้วยการฝึกสอนแบบจำลอง LSTM เพื่อพยากรณ์ค่าส่วนเหลือ (Residual) สำหรับเป็นข้อมูลฝึกสอนแบบจำลอง XGBoost ขั้นตอนการทำงานของวิธีการพยากรณ์แบบผสมแสดงดังภาพที่ 4.4



ภาพที่ 4.5 ขั้นตอนการทำงานของวิธีการพยากรณ์แบบผสม

จากภาพที่ 4.4 แบบจำลอง LSTM ถูกเลือกให้เป็นแบบจำลองหลัก (Primary Model) เพื่อหาค่าส่วนเหลือ (Residual) สำหรับเป็นข้อมูลนำเข้าแบบจำลอง XGBoost ที่ถูกเลือกให้เป็นแบบจำลองรอง (Secondary Model) โดยผลลัพธ์ของการพยากรณ์ทั้ง 2 แบบจำลองจะถูกรวมเข้าด้วยกัน

ผลลัพธ์การพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากการสร้างแบบจำลองโดยอาศัยวิธีการพยากรณ์แบบผสม แสดงดังตารางที่ 4.7 ผลการประเมินประสิทธิภาพแบบจำลองแสดงดังตารางที่ 4.8 และผลการพยากรณ์ แสดงดังภาพที่ 4.5

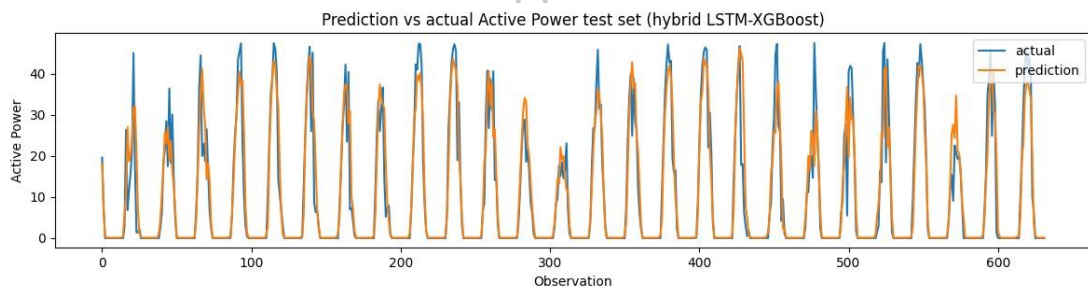
ตารางที่ 4.7 ผลการพยากรณ์เปรียบเทียบกับค่าจริงในช่วง 24 ชั่วโมง จากแบบจำลอง Hybrid

Period	Actual Value	Forecast Value	Error
t+1	0.00	0.00	0.00
t+2	0.00	0.00	0.00
t+3	0.00	0.00	0.00
t+4	0.00	0.00	0.00
t+5	0.00	0.00	0.00
t+6	0.00	0.00	0.00
t+7	0.00	0.00	0.00
t+8	0.00	0.00	0.00
t+9	2.57	5.30	-2.73
t+10	26.32	11.35	14.97
t+11	6.85	29.46	-22.61
t+12	11.63	21.25	-9.62
t+13	15.74	20.94	-5.20
t+14	21.07	23.01	-1.94
t+15	45.06	28.04	17.02
t+16	17.50	35.66	-18.16
t+17	1.44	22.27	-20.83
t+18	1.34	6.13	-4.79
t+19	2.18	0.80	1.38
t+20	0.00	0.02	-0.02
t+21	0.00	0.00	0.00
t+22	0.00	0.00	0.00
t+23	0.00	0.00	0.00
t+24	0.00	0.00	0.00



ตารางที่ 4.8 ผลการประเมินประสิทธิภาพแบบจำลองโดยอาศัยวิธีการพยากรณ์แบบผสม

Methods	MAE	MSE	RMSE
Hybrid	0.0577	0.0143	0.1196



ภาพที่ 4.5 ผลการพยากรณ์ของแบบจำลอง Hybrid ด้วยชุดข้อมูลทดสอบ

จากภาพที่ 4.5 กำหนดเส้นสีน้ำเงิน คือ ค่ากำลังไฟฟ้าจริง และเส้นสีส้ม คือ ค่ากำลังไฟฟ้าจากการพยากรณ์ จากชุดข้อมูลทดสอบ 632 ชั่วโมง พบว่าแบบจำลองที่สร้างโดยอาศัยวิธีการแบบผสม ด้วยหน่วยจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง (LSTM-XGBoost) พบว่า ค่าพยากรณ์ที่ได้มีความใกล้เคียงกับค่ากำลังไฟฟ้าจริงตั้งแต่ชั่วโมงที่ 48 - ชั่วโมงที่ 480

#### 4. ประเมินประสิทธิภาพของแบบจำลองทั้ง 3 วิธี

ผลการเปรียบเทียบประสิทธิภาพของแบบจำลองโดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาว วิธีการเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง และวิธีการพยากรณ์แบบผสม โดยคำนวณ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และรากที่สองของความผิดพลาดกำลังสอง (RMSE) แสดงดังตารางที่ 4.9 และภาพที่ 4.6

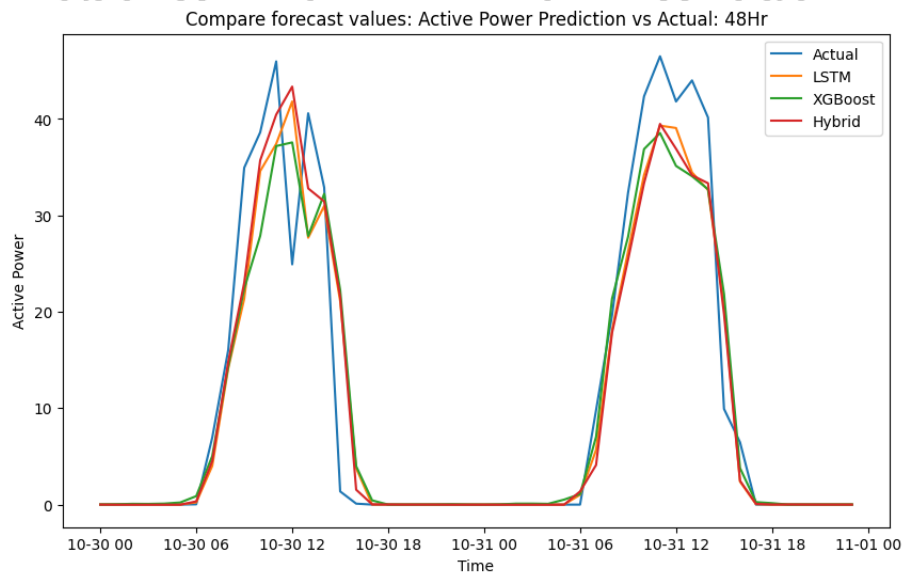
ตารางที่ 4.9 ตารางเปรียบเทียบประสิทธิภาพของแบบจำลองทั้ง 3 วิธี

Methods	MAE	MSE	RMSE
LSTM	0.0585	0.0148	0.1215
XGBoost	0.0630	0.0147	0.1211
Hybrid	0.0577	0.0143	0.1196

จากตารางที่ 4.4 การวิเคราะห์เปรียบเทียบประสิทธิภาพของแบบจำลอง โดยเปรียบเทียบประสิทธิภาพค่าความคลาดเคลื่อนด้วย MAE ค่าที่ดีที่สุด คือ แบบจำลองวิธีการพยากรณ์แบบผสม เท่ากับ 0.0577 แบบจำลอง LSTM เท่ากับ 0.0585 และแบบจำลอง XGBoost เท่ากับ 0.0630 ตามลำดับ

เมื่อเปรียบเทียบประสิทธิภาพค่าความคลาดเคลื่อนด้วย MSE ค่าที่ดีที่สุด คือ แบบจำลองวิธีการพยากรณ์แบบผสม เท่ากับ 0.01430 แบบจำลอง XGBoost เท่ากับ 0.0147 และแบบจำลอง LSTM เท่ากับ 0.0148 ตามลำดับ

เมื่อเปรียบเทียบประสิทธิภาพค่าความคลาดเคลื่อนด้วย RMSE ค่าที่ดีที่สุด คือ แบบจำลองวิธีการพยากรณ์แบบผสม เท่ากับ 0.1196 แบบจำลอง XGBoost เท่ากับ 0.1211 และแบบจำลอง LSTM เท่ากับ 0.1215 ตามลำดับ



ภาพที่ 4.6 เปรียบเทียบผลการพยากรณ์ของแบบจำลองทั้ง 3 วิธี

จากภาพที่ 4.6 แสดงการเปรียบเทียบผลการพยากรณ์ของแบบจำลองทั้ง 3 วิธี โดยเป็นการเปรียบเทียบระหว่างค่าจริงและค่าที่แบบจำลองพยากรณ์ได้ 48 ชั่วโมง จากกราฟพบว่าแบบจำลองที่สร้างด้วยวิธีการพยากรณ์แบบผสม ให้ผลลัพธ์ความถูกต้องและประสิทธิภาพดีกว่าแบบจำลองอื่น ๆ



## บทที่ 5

### สรุปการศึกษา อภิปรายผล และข้อเสนอแนะ

การศึกษาค้นคว้าครั้งนี้มีวัตถุประสงค์ 1) เพื่อศึกษาวิธีการและสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตตึง 2) เพื่อประเมินประสิทธิภาพของแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตตึง โดยข้อมูลที่นำมาใช้ในการศึกษาเป็นข้อมูลกำลังไฟฟ้าและข้อมูลจากอุปกรณ์เซนเซอร์ของโรงไฟฟ้าพลังงานแสงอาทิตย์บนทุ่นลอยน้ำขนาดกำลังการผลิต 45 เมกะวัตต์ โดยเก็บรวบรวมตั้งแต่เดือนกุมภาพันธ์ ถึงเดือนตุลาคม พ.ศ. 2566 จำนวน 6,551 ระเบียบ และ 11 คุณลักษณะ งานวิจัยนี้นำเสนอวิธีการสร้างแบบจำลองด้วย 1) หน่วยความจำระยะสั้นแบบยาว 2) เอ็กซ์ตรีมเกรเดียนต์บูตตึง และ 3) วิธีการพยากรณ์แบบผสม ใช้การวิเคราะห์ด้วยวิธีสหสัมพันธ์เพียร์สันเพื่อคัดเลือกคุณลักษณะที่ส่งผลต่อการพยากรณ์เข้าสู่แบบจำลอง และประเมินประสิทธิภาพของแบบจำลอง สามารถสรุปผลการศึกษาดังต่อไปนี้

1. สรุปการศึกษา
2. อภิปรายผล
3. ข้อเสนอแนะ

#### 1. สรุปการศึกษา

ในการสรุปผลของการศึกษา ผู้วิจัยได้ทำการสรุปผลตามวัตถุประสงค์ในแต่ละข้อที่ได้กล่าวไว้ข้างต้น ดังต่อไปนี้

**1.1. แบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตตึง** ซึ่งเป็นการประยุกต์ใช้วิธีการแบบผสมด้วยอัลกอริทึมการเรียนรู้เชิงลึก 2 อัลกอริทึม ได้แก่ 1) หน่วยความจำระยะสั้นแบบยาว (LSTM) ที่มีความสามารถในการจดจำข้อมูลระยะยาวภายใต้ข้อมูลอนุกรมเวลา และ 2) เอ็กซ์ตรีมเกรเดียนต์บูตตึง (XGBoost) ที่มีความสามารถการตรวจจับข้อมูลที่มีความผันผวน งานวิจัยนี้ได้นำเสนอวิธีการพยากรณ์โดยใช้แบบจำลองหลัก (Primary Model) ที่สร้างจากอัลกอริทึมหน่วยความจำระยะสั้นแบบยาว เพื่อหาค่าพยากรณ์ส่วนเหลือ (Residual) สำหรับเป็นข้อมูลนำเข้าสู่สำหรับ

แบบจำลองรอง โดยแบบจำลองรอง (Secondary Model) ที่สร้างจากอัลกอริทึมเอ็กซ์ตรีมเกรเดียนต์บูตส์ตริง (XGBoost) จากนั้นผลการพยากรณ์ทั้งสองแบบจำลองจะถูกรวมเข้าด้วยกัน จากผลลัพธ์พบว่า การพยากรณ์แบบผสมมีประสิทธิภาพสูงกว่าการพยากรณ์ที่อาศัยวิธีการพยากรณ์แบบเดี่ยว แสดงดังตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบผลการพยากรณ์และค่าพลังงานแสงอาทิตย์ในแบบจำลองแต่ละแบบ

Actual Value	Forecast Value		
	LSTM	XGBoost	Hybrid
19.65	18.08	21.37	18.82
32.34	26.25	27.77	26.41
42.35	34.18	36.87	37.34
46.50	39.31	38.54	39.48
41.81	39.06	35.13	39.91
44.00	34.46	34.06	34.54
40.15	32.63	32.70	33.33
9.92	20.55	21.85	19.76
6.47	2.38	3.79	5.48

1.2. ผลการประเมินประสิทธิภาพของแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตส์ตริง การประเมินประสิทธิภาพของแบบจำลองด้วยค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ค่าเฉลี่ยความผิดพลาดกำลังสอง (MSE) และรากที่สองของค่าเฉลี่ยความผิดพลาดกำลังสอง (RMSE) กับข้อมูลชุดทดสอบ (Testing data) พบว่าแบบจำลองที่ใช้วิธีการพยากรณ์แบบผสม (Hybrid) ให้ค่าประสิทธิภาพที่ดีที่สุด คือ มีค่าใกล้เคียง 0 มากที่สุด โดยมีค่า MAE เท่ากับ 0.0577 ค่า MSE เท่ากับ 0.0143 รองลงมาคือแบบจำลอง LSTM และแบบจำลอง XGBoost ตามลำดับ ประสิทธิภาพของแบบจำลองทั้งหมด แสดงดังตารางที่ 5.2

ตารางที่ 5.2 ตารางแสดงประสิทธิภาพของแบบจำลองทั้งหมด

Methods	MAE	MSE	RMSE
LSTM	0.0585	0.0148	0.1215
XGBoost	0.0630	0.0147	0.1211
Hybrid	<b>0.0577</b>	<b>0.0143</b>	<b>0.1196</b>

## 2. อภิปรายผล

ผลการศึกษการสร้างแบบจำลองการพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ โดยอาศัยวิธีการหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง สามารถอภิปรายผลได้ ดังนี้

2.1. จากผลการศึกษาพบว่า การพยากรณ์การผลิตพลังงานแสงอาทิตย์บนทุ่นลอยน้ำ และสร้างแบบจำลองด้วยหน่วยความจำระยะสั้นแบบยาวและเอ็กซ์ตรีมเกรเดียนต์บูตติ้ง พบว่าแบบจำลอง LSTM และแบบจำลอง XGBoost สามารถพยากรณ์ค่ากำลังไฟฟ้าล่วงหน้า 24 ชั่วโมงได้ใกล้เคียงกับค่ากำลังไฟฟ้าจริง หากพิจารณาจากกราฟการพยากรณ์ จะพบว่าแบบจำลองที่สร้างด้วย LSTM ให้ค่าพยากรณ์ที่ใกล้เคียงกับค่ากำลังไฟฟ้าจริงตั้งแต่ชั่วโมงที่ 0 เป็นต้นไป จึงมีความเหมาะสมในการพยากรณ์แบบรายชั่วโมง ซึ่งได้ผลคล้ายกับงานวิจัยของ Liu และคณะ ในขณะที่กราฟการพยากรณ์ที่สร้างจากแบบจำลองที่สร้างด้วย XGBoost มีค่าพยากรณ์ใกล้เคียงกับค่ากำลังไฟฟ้าจริงตั้งแต่ชั่วโมงที่ 100 – ชั่วโมงที่ 200 และพบว่าเกิดค่าความคลาดเคลื่อนสูงในชั่วโมงที่ 480 เป็นต้นไป เมื่อพิจารณาจากกราฟการสูญเสียระหว่างการฝึกสอนแบบจำลองพบว่า ค่า Training Loss และ Validation Loss ของแบบจำลองทั้ง 2 แบบจำลอง มีแนวโน้มลดลง แต่จะพบช่องว่าง (Gap) ระหว่างค่า Training Loss กับค่า Validation Loss ซึ่งแปลผลได้ว่า ชุดข้อมูลสำหรับการฝึกสอนแบบจำลองอาจมีน้อยเกินไป

2.2. การเพิ่มประสิทธิภาพของการพยากรณ์ด้วยวิธีการพยากรณ์แบบผสม ด้วยการสร้างแบบจำลอง LSTM เป็นแบบจำลองหลัก และแบบจำลอง XGBoost เป็นแบบจำลองรอง พบว่าการพยากรณ์ค่ากำลังไฟฟ้าล่วงหน้า 24 ชั่วโมงได้ใกล้เคียงกับค่ากำลังไฟฟ้าจริง หากพิจารณาจากกราฟการพยากรณ์ จะพบว่าแบบจำลองที่สร้างด้วย LSTM-XGBoost ให้ค่าพยากรณ์ที่ใกล้เคียงกับค่ากำลังไฟฟ้าจริงตั้งแต่ชั่วโมงที่ 48 เป็นต้นไป ซึ่งมีความเหมาะสมกับการพยากรณ์ในระยะสั้น (Short-term Forecasting) และเมื่อพิจารณาจากค่าความคลาดเคลื่อนเปรียบเทียบกับแบบจำลองแบบเดี่ยวที่

สร้างขึ้น พบว่าแบบจำลองแบบผสมที่สร้างด้วยวิธีการพยากรณ์แบบผสม มีประสิทธิภาพการพยากรณ์ที่ดีที่สุด สอดคล้องกับงานวิจัยของ อูรชา จันทรภา และ Zhu และคณะ ในการสร้างแบบจำลองแบบผสม เพื่อเพิ่มประสิทธิภาพในการพยากรณ์ข้อมูล เนื่องจากการพยากรณ์ด้วยวิธีการนี้เป็นการหาค่าส่วนเหลือ (Residual) ที่แบบจำลองหลักพยากรณ์ผิดพลาด (Error) เพื่อเป็นชุดข้อมูลสำหรับฝึกสอนให้กับแบบจำลองรอง

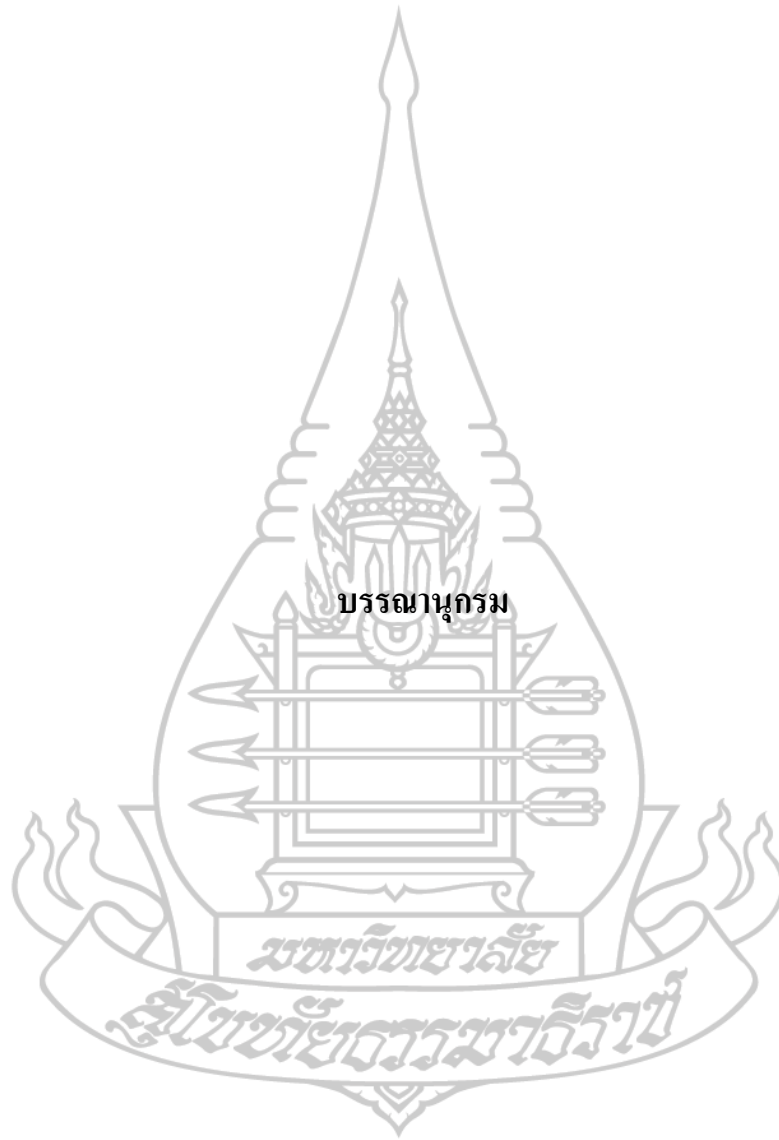
### 3. ข้อเสนอแนะ

3.1. พิจารณาข้อมูลคุณลักษณะ (Feature) อื่น ๆ ที่เป็นปัจจัยหรือส่งผลกระทบต่อการพยากรณ์ รวมถึงการเก็บรวบรวมข้อมูลในอดีตในปริมาณมาก จะช่วยให้แบบจำลองสามารถเรียนรู้ได้อย่างแม่นยำและประสิทธิภาพดียิ่งขึ้น

3.2. ควรมีการปรับปรุงไฮเปอร์พารามิเตอร์ในแบบจำลองการเรียนรู้ XGBoost โดยใช้วิธีการค้นหาค่าไฮเปอร์พารามิเตอร์ที่ดีที่สุด สำหรับแบบจำลอง LSTM ควรเพิ่มการใช้ฟังก์ชันกระตุ้นชนิดอื่น ๆ ปรับปรุงชั้นซ่อน ฟังก์ชันการสูญเสียชนิดอื่น ๆ รวมถึงรอบในการฝึกสอนแบบจำลอง เพื่อเปรียบเทียบค่าผลลัพธ์ จะทำให้แบบจำลองมีประสิทธิภาพดีขึ้น

3.3. ประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึกอื่น ๆ ที่มีความสามารถในการพยากรณ์ข้อมูลแบบลำดับหรืออนุกรมเวลา เช่น GRU, LSTM-Attention เป็นต้น

3.4. แบบจำลองที่ได้จากการวิจัยนี้ สามารถนำไปพัฒนาเขียนเป็นโปรแกรมสำหรับระบบการพยากรณ์พลังงานแสงอาทิตย์บนหุ่นลอยน้ำแบบรายชั่วโมง หรือรายวัน เพื่อช่วยให้ผู้ควบคุมระบบ สามารถบริหารจัดการการผลิตพลังงานไฟฟ้าจากพลังงานแสงอาทิตย์ได้อย่างมีประสิทธิภาพ



บรรณานุกรม

มหาวิทยาลัยราชภัฏวชิรเวศน์

ศูนย์วิทยบริการวชิรเวศน์



## บรรณานุกรม

- กรมพัฒนาพลังงานทดแทนและอนุรักษ์พลังงาน กระทรวงพลังงาน. (ม.ป.ป.). การจัดการก๊าซเรือนกระจก .สืบค้นจาก <https://www.dede.go.th/articles?id=191>
- เกศินี ช่อนกลิ่น. (2566). แบบจำลองการพยากรณ์ราคาอนุพันธ์ โดยใช้เทคนิคโครงข่ายประสาทเทียมแบบสังวัตนาการ 1 มิติ หน่วยความจำระยะสั้นแบบยาวและซัพพอร์ตเวกเตอร์รีเกรสชัน [วิทยานิพนธ์ ปริญญาโทบริหารธุรกิจ, สถาบันเทคโนโลยีไทย-ญี่ปุ่น].  
[https://tdc.thailis.or.th/tdc/search\\_result.php](https://tdc.thailis.or.th/tdc/search_result.php)
- ภาณุพงศ์ น้ำแก้ว. (2563). การพยากรณ์กำลังไฟฟ้าของระบบเซลล์แสงอาทิตย์แบบติดตั้งอุปกรณ์วัดด้วยโครงข่ายประสาทเทียม. *Chulalongkorn University Theses and Dissertations (Chula ETD)*. <https://doi.org/10.58837/CHULA.THE.2020.1112>
- ศรัริรักษ์ ศรีทองชัย. (2565). แบบจำลองการพยากรณ์ค่า PM2.5 โดยใช้โครงข่ายประสาทเทียมแบบ LSTM ในพื้นที่กรุงเทพมหานคร. *Journal of Engineering and Digital Technology (JEDT)*, 10(1), 1-9.
- สรุวุฒิ จิตตานนท์ และจักรกฤษ เต็มฤทธิกุล. (2566). การพยากรณ์การใช้พลังงานภายในอาคาร SGtech อย่างชาญฉลาดด้วยการเรียนรู้เชิงลึก [วิทยานิพนธ์ ปริญญาโทบริหารธุรกิจ, มหาวิทยาลัยนเรศวร]. <http://nuir.lib.nu.ac.th/dspace/handle/123456789/5913>
- สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน. (2561). แผนพัฒนากำลังผลิตไฟฟ้าของประเทศไทย (Power Development Plan : PDP) .สืบค้นจาก <https://www.eppo.go.th/index.php/th/electricity/pdp>
- อุรษา จันทรภา. (2564). การพยากรณ์มูลค่าการส่งออกรถยนต์ อุปกรณ์ และส่วนประกอบของประเทศไทย. *Chulalongkorn University Theses and Dissertations (Chula ETD)*.  
<https://doi.org/10.58837/CHULA.THE.2021.1012>
- Berberich, D. (2020, October 19). *Hybrid Methods for Time Series Forecasting*. inovex GmbH.  
<https://www.inovex.de/de/blog/hybrid-time-series-forecasting/>
- Cazzaniga, R., & Rosa-Clot, M. (2021). The booming of floating PV. *Solar Energy*, 219, 3–10.  
<https://doi.org/10.1016/j.solener.2020.09.057>

- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Foster, J., Liu, X., & McLoone, S. (2017). Adaptive sliding window load forecasting. *2017 28th Irish Signals and Systems Conference (ISSC)*, 1–6. <https://doi.org/10.1109/ISSC.2017.7983638>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Jebli, I., Belouadha, F.-Z., Kabbaj, M. I., & Tilioua, A. (2021). Deep Learning based Models for Solar Energy Prediction. *Advances in Science, Technology and Engineering Systems Journal*, 6(1), 349–355. <https://doi.org/10.25046/aj060140>
- Li, C., Chen, Z., Liu, J., Li, D., Gao, X., Di, F., Li, L., & Ji, X. (2019). Power Load Forecasting Based on the Combined Model of LSTM and XGBoost. *Proceedings of the 2019 the International Conference on Pattern Recognition and Artificial Intelligence*, 46–51. <https://doi.org/10.1145/3357777.3357792>
- Lim, S.-C., Huh, J.-H., Hong, S.-H., Park, C.-Y., & Kim, J.-C. (2022). Solar Power Forecasting Using CNN-LSTM Hybrid Model. *Energies*, 15(21), 8233. <https://doi.org/10.3390/en15218233>
- Liu, C.-H., Gu, J.-C., & Yang, M.-T. (2021). A Simplified LSTM Neural Networks for One Day-Ahead Solar Power Forecasting. *IEEE Access*, 9, 17174–17195. <https://doi.org/10.1109/ACCESS.2021.3053638>
- Obiora, C. N., Ali, A., & Hasan, A. N. (2021). Implementing Extreme Gradient Boosting (XGBoost) Algorithm in Predicting Solar Irradiance. *2021 IEEE PES/IAS PowerAfrica*, 1–5. <https://doi.org/10.1109/PowerAfrica52236.2021.9543159>
- Obiora, C. N., Ali, A., & Hasan, A. N. (2022). Efficient Method of Finding the Best Time Interval for Predicting Short Term Solar Radiation Using CNN-LSTM Model. *2022 11th International Conference on Power Science and Engineering (ICPSE)*, 159–163. <https://doi.org/10.1109/ICPSE56329.2022.9935441>
- Phan, Q.-T., Wu, Y.-K., & Phan, Q.-D. (2021). Short-term Solar Power Forecasting Using XGBoost with Numerical Weather Prediction. *2021 IEEE International Future Energy*

*Electronics Conference (IFEEC)*, 1–6.

<https://doi.org/10.1109/IFEEC53238.2021.9661874>

Saxena, S. (2021, March 16). What is LSTM? Introduction to Long Short-Term Memory.

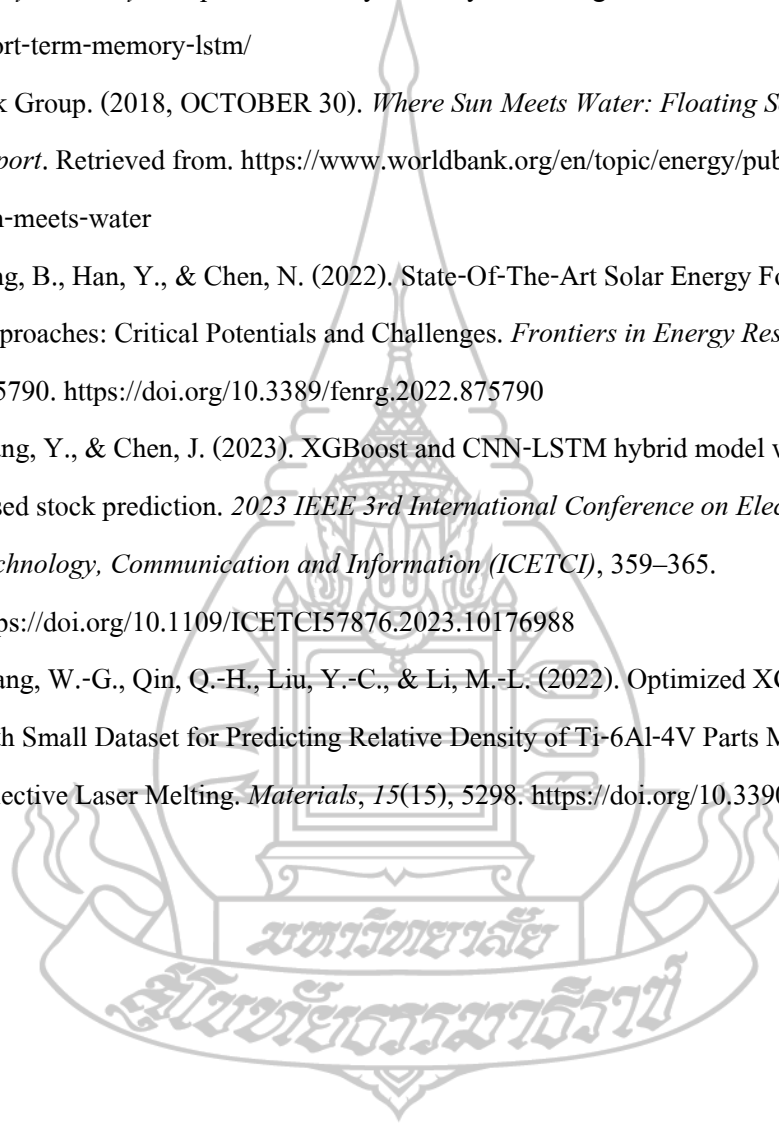
*Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>

World Bank Group. (2018, OCTOBER 30). *Where Sun Meets Water: Floating Solar Market Report*. Retrieved from. <https://www.worldbank.org/en/topic/energy/publication/where-sun-meets-water>

Ye, H., Yang, B., Han, Y., & Chen, N. (2022). State-Of-The-Art Solar Energy Forecasting Approaches: Critical Potentials and Challenges. *Frontiers in Energy Research*, 10, 875790. <https://doi.org/10.3389/fenrg.2022.875790>

Zhu, R., Yang, Y., & Chen, J. (2023). XGBoost and CNN-LSTM hybrid model with Attention-based stock prediction. *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 359–365. <https://doi.org/10.1109/ICETCI57876.2023.10176988>

Zou, M., Jiang, W.-G., Qin, Q.-H., Liu, Y.-C., & Li, M.-L. (2022). Optimized XGBoost Model with Small Dataset for Predicting Relative Density of Ti-6Al-4V Parts Manufactured by Selective Laser Melting. *Materials*, 15(15), 5298. <https://doi.org/10.3390/ma15155298>





ภาคผนวก

มหาวิทยาลัย

สุโขทัยธรรมมาธิราช

## 1. การสร้างแบบจำลองด้วยวิธีการหน่วยความจำระยะสั้นแบบยาว

### รหัสโปรแกรม

```
import numpy as np
```

```
import pandas as pd
```

```
df = pd.read_csv('/data/PV02-10-2023.csv')
```

```
print('Number of rows and columns:', df.shape)
```

```
df.head(10)
```

```
df.describe().T
```

```
corr = df.corr()
```

```
plt.figure(figsize=(8,6))
```

```
sns.heatmap(corr, cmap="Blues",annot=True)
```

```
plt.show()
```

```
features_list=['Solar_active_power',
```

```
'Obliquity_radiation','Panel_temperature_1','Panel_temperature_2','Panel_temperature_3']
```

```
df_used = df[features_list]
```

```
df_used.head()
```

```
corr = df_used.corr()
```

```
plt.figure(figsize=(6,5))
```

```
sns.heatmap(corr, cmap="Blues",annot=True)
```

```
plt.show()
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
```

```
scaled_data = scaler.fit_transform(df_used)
```

```
X = df_used.drop(['Solar_active_power'], axis = 1)
```

```
y= df_used[['Solar_active_power']]
```

```
scaler_x = MinMaxScaler()
```

```
scaler_x = scaler_x.fit(X)
```

```
X_norm = scaler_x.transform(X)
```

```
scaler_y = MinMaxScaler()
```

```

scaler_y.fit(y)
y_norm = scaler_y.transform(y)

train_cutoff = int(0.8*X_norm.shape[0])
val_cutoff = int(0.9*X_norm.shape[0])

def windowing(X_input,y_input, history_size):
    data = []
    labels = []
    for i in range(history_size, len(y_input)):
        data.append(X_input[i - history_size : i, :])
        labels.append(y_input[i])
    return np.array(data), np.array(labels).reshape(-1,1)

hist_size= 24
data_final = np.concatenate((X_norm,y_norm), axis = 1)

X_train, y_train = windowing(data_final[:train_cutoff,:],data_final[:train_cutoff,-1], hist_size)
X_val, y_val = windowing(data_final[train_cutoff
:val_cutoff,:],data_final[train_cutoff:val_cutoff,-1], hist_size)
X_test, y_test = windowing(data_final[val_cutoff :,:],data_final[val_cutoff:-1], hist_size)

import math
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go

fig, axes = plt.subplots(figsize = (14,6))
axes.plot(df_used['Solar_active_power'].iloc[:train_cutoff], color = 'blue')
axes.plot(df_used['Solar_active_power'].iloc[train_cutoff + 1 : val_cutoff], color = 'orange')
axes.plot(df_used['Solar_active_power'].iloc[val_cutoff + 1 :], color = 'green')
axes.axvline(x=df_used.index[train_cutoff], color='black', linestyle='--')
axes.axvline(x=df_used.index[val_cutoff], color='black', linestyle='--')
axes.set_title('Visual split of train (black), validation (green) and test (blue) sets')
axes.set_xlabel('Date')
axes.set_ylabel('MW actual')

```

```
plt.show()
```

```
def plot_results(y_pred_actual, y_test_inv, model, model_name):
```

```
    fig, ax = plt.subplots(2, 1, figsize=(12, 6))
```

```
    ax[0].plot(y_test_inv[:1000])
```

```
    ax[0].plot(y_pred_actual[:1000])
```

```
    ax[0].legend(['actual', 'prediction'], loc='upper right')
```

```
    ax[0].set_title(f'Prediction vs actual MW observation in test set ({model_name})')
```

```
    ax[0].set_xlabel('Observation')
```

```
    ax[0].set_ylabel('Active Power')
```

```
    if model != 'hybrid':
```

```
        if model_name == 'XGBoost':
```

```
            train_mae = model.evals_result()['validation_0']['mae']
```

```
            val_mae = model.evals_result()['validation_1']['mae']
```

```
            ax[1].plot(train_mae, label='Training MAE')
```

```
            ax[1].plot(val_mae, label='Validation MAE')
```

```
        else:
```

```
            ax[1].plot(history.history['loss'], label='Training Loss')
```

```
            ax[1].plot(history.history['val_loss'], label='Validation Loss')
```

```
            ax[1].legend()
```

```
            ax[1].set_title(f'Training and validation MAE ({model_name})')
```

```
            ax[1].set_xlabel('Iteration/Epochs')
```

```
            ax[1].set_ylabel('MAE')
```

```
    fig.tight_layout()
```

```
    plt.show()
```

```
import tensorflow as tf
```

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import LSTM, Dense, Dropout, Flatten
```

```
from tensorflow.keras.optimizers import Adam, SGD
```

```
from tensorflow.keras.callbacks import EarlyStopping
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```

# inputs for DL models
epoch = 100
batch_size = 64
callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=50)
optimizer = 'Adam'

def base_model_lstm():
    model = Sequential()
    model.add(LSTM(units = 32, return_sequences = True, activation="relu", input_shape =
X_train.shape[-2:]))
    model.add(Flatten())
    model.add(Dense(units=128, activation='relu'))
    model.add(Dropout(0.1))
    model.add(Dense(1))
    return model

lstm_model = base_model_lstm()
lstm_model.compile(optimizer = optimizer , loss = 'mean_absolute_error')
lstm_model.summary()

history = lstm_model.fit(X_train, y_train, validation_data = (X_val, y_val), epochs =epoch,
batch_size = batch_size, callbacks=[callback])

y_pred = lstm_model.predict(X_test)
y_test_inv = scaler_y.inverse_transform(y_test)
lstm_pred = y_pred

# Calculate evaluation metrics
mae_lstm = mean_absolute_error(lstm_pred,y_test)
mse_lstm = mean_squared_error(lstm_pred,y_test)
rmse_lstm = np.sqrt(mse_lstm)

print("")
print("")
print('-----')
print(f'LSTM MAE for test set : {round(mae_lstm,4)}')

```



```

print(f'LSTM MSE for test set : {round(mse_lstm,4)}')
print(f'LSTM RMSE for test set : {round(rmse_lstm,4)}')
print('-----')
y_pred_actual = scaler_y.inverse_transform(lstm_pred)
lstm_pred_actual = y_pred_actual
print("")
plot_results(y_pred_actual, y_test_inv, history,'LSTM')

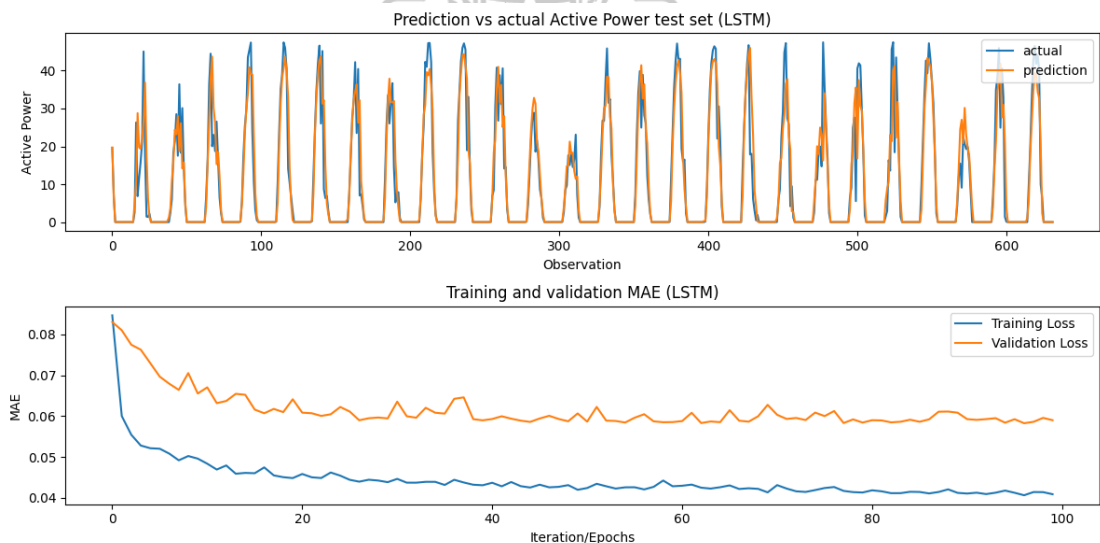
```

### ผลการพยากรณ์ด้วยวิธีหน่วยความจำระยะสั้นแบบยาว

```

LSTM MAE for test set : 0.0585
LSTM MSE for test set : 0.0148
LSTM RMSE for test set : 0.1215

```



## 2. การสร้างแบบจำลองด้วยวิธีการเอ็กซ์ตรีมเกรเดียนต์บูสตีง

### รหัสโปรแกรม

```
import xgboost as xgb
```

```
def reshaping(X):
```

```

reshaped_x = X.reshape(-1 , X.shape[1] * X.shape[2])

return reshaped_x

X_train_xgb = reshaping(X_train)
X_val_xgb = reshaping(X_val)
X_test_xgb = reshaping(X_test)

xgb_model = xgb.XGBRegressor(eval_metric='mae',
                             early_stopping_rounds=20,
                             colsample_bytree= 0.7,
                             learning_rate=0.01,
                             max_depth=3,
                             n_estimators=1000)

# Fit the model
xgb_model.fit(X_train_xgb, y_train,
              eval_set=[(X_train_xgb, y_train), (X_val_xgb, y_val)],
              verbose=False)

y_pred = xgb_model.predict(X_test_xgb)
xgb_pred = y_pred

# Calculate evaluation metrics
mae_xgb = mean_absolute_error(xgb_pred,y_test)
mse_xgb = mean_squared_error(xgb_pred,y_test)
rmse_xgb = np.sqrt(mse_xgb)

print("")
print("")
print('-----')
print(f'XGBoost MAE for test set : {round(mae_xgb,4)}')
print(f'XGBoost MSE for test set : {round(mse_xgb,4)}')
print(f'XGBoost RMSE for test set : {round(rmse_xgb,4)}')
print('-----')

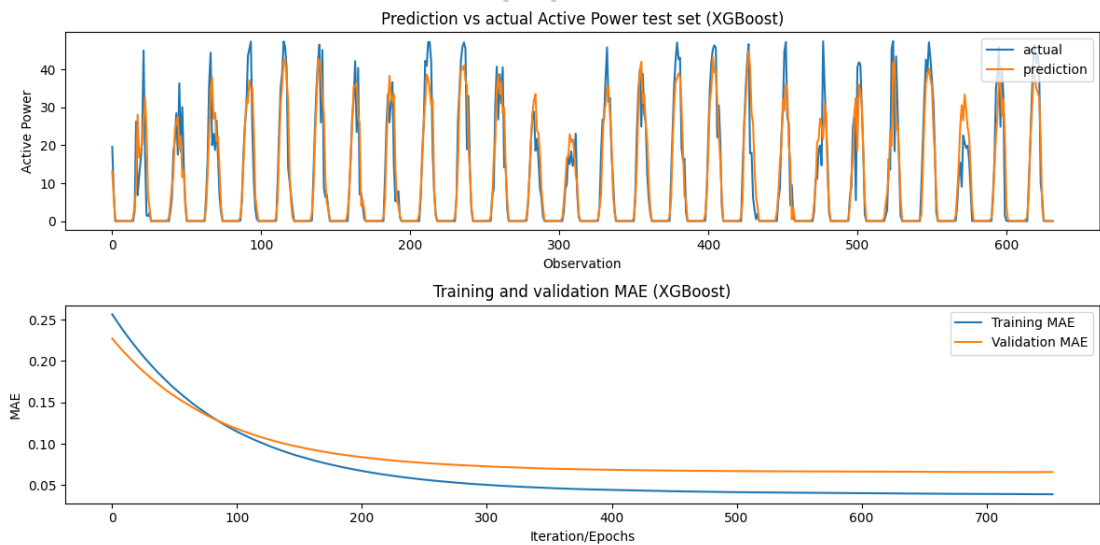
y_pred_actual = scaler_y.inverse_transform(xgb_pred.reshape(-1,1))
xgb_pred_actual = y_pred_actual

```

```
print("")
plot_results(y_pred_actual, y_test_inv, xgb_model,'XGBoost')
```

ผลการพยากรณ์ด้วยวิธีเอ็กซ์ตรีมเกรเดียนต์บูสติง

XGBoost MAE for test set : 0.063  
 XGBoost MSE for test set : 0.0147  
 XGBoost RMSE for test set : 0.1211



### 3. การสร้างแบบจำลองด้วยวิธีการพยากรณ์แบบผสม

รหัสโปรแกรม

```
def base_model_lstm():
    model = Sequential()
    model.add(LSTM(units = 32, return_sequences = True, activation="relu", input_shape =
X_train.shape[-2:]))
    model.add(Flatten())
    model.add(Dense(units=128, activation='relu'))
    model.add(Dropout(0.1))
    model.add(Dense(1))
    return model
```

```

lstm_hybrid_model = base_model_lstm()
lstm_hybrid_model.compile(optimizer = 'Adam' , loss = 'mean_absolute_error')
print("")
print('LSTM is fitting a model on train-validation sets...')
print("")
lstm_hybrid_model.fit(X_train, y_train, validation_data = (X_val, y_val), epochs = epoch,
                      batch_size = batch_size, callbacks=[callback])
print("")
print('-----')
print('LSTM is predicting...')
print("")
residuals_train = y_train - lstm_hybrid_model.predict(X_train)
residuals_val = y_val - lstm_hybrid_model.predict(X_val)
print("")
print('-----')
print('XGBoost is fitting a model on residuals...')
print("")
xgb_hybrid = xgb.XGBRegressor(eval_metric='mae',
                              early_stopping_rounds=20,
                              colsample_bytree= 0.7,
                              learning_rate=0.01,
                              max_depth=3,
                              n_estimators=1000)
xgb_hybrid.fit(X_train_xgb, residuals_train,
              eval_set=[(X_train_xgb, residuals_train), (X_val_xgb, residuals_val)],
              verbose=False)
y_train_pred = lstm_hybrid_model.predict(X_train) +
xgb_hybrid.predict(X_train_xgb).reshape(-1,1)
y_val_pred = lstm_hybrid_model.predict(X_val) +
xgb_hybrid.predict(X_val_xgb).reshape(-1,1)

```

```

y_test_pred = lstm_hybrid_model.predict(X_test) + xgb_hybrid.predict(X_test_xgb).reshape(-
1,1)
print("")
print("")
hybrid_pred = y_test_pred
# Calculate evaluation metrics
mae_hybrid = mean_absolute_error(hybrid_pred,y_test)
mse_hybrid = mean_squared_error(hybrid_pred,y_test)
rmse_hybrid = np.sqrt(mse_hybrid)
print('-----')
print(f'LSTM-XGBoost MAE for test set : {round(mae_hybrid,4)}')
print(f'LSTM-XGBoost MSE for test set : {round(mse_hybrid,4)}')
print(f'LSTM-XGBoost RMSE for test set : {round(rmse_hybrid,4)}')
print('-----')
y_pred_actual = scaler_y.inverse_transform(hybrid_pred)
lstm_xgb_pred_actual = y_pred_actual
print("")
plot_results(y_pred_actual, y_test_inv, 'hybrid', 'hybrid LSTM-XGBoost')

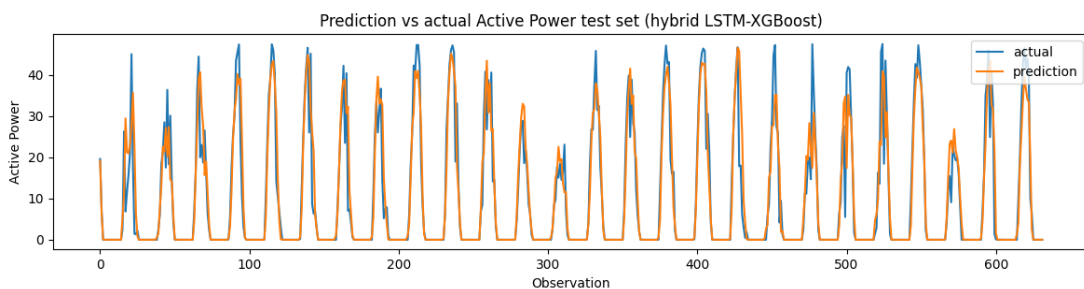
```

### ผลการพยากรณ์ด้วยวิธีการพยากรณ์แบบผสม

LSTM-XGBoost MAE for test set : 0.0577

LSTM-XGBoost MSE for test set : 0.0143

LSTM-XGBoost RMSE for test set : 0.1196



## ประวัติผู้ศึกษา

ชื่อสกุล	นายอดิเรก ปั่นบุญ
วัน เดือน ปี เกิด	8 มีนาคม 2533
สถานที่เกิด	จังหวัดกำแพงเพชร
ที่อยู่ปัจจุบัน	กฟผ. เขื่อนสิริกิติ์ 40 ม. 10 ต.ผาเลือด อ.ท่าปลา จ.อุตรดิตถ์
ประวัติการศึกษา	ปริญญาวิทยาศาสตรบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร) มหาวิทยาลัยสุโขทัยธรรมมาธิราช พ.ศ. 2564
ประวัติการทำงาน	พนักงานรัฐวิสาหกิจ การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

