

การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้าโดยใช้การ  
เรียนรู้ของเครื่องและการเรียนรู้เชิงลึก

**Energy Consumption Prediction for Boiler Feed Pumps in Power Plants  
Using Machine Learning and Deep Learning**

พลวุฒิ คำฝอย

POLAWUT KHAMFOY

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์  
คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2567

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**Energy Consumption Prediction for Boiler Feed Pumps in Power Plants  
Using Machine Learning and Deep Learning**



**AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS  
FOR THE DEGREE OF MASTER OF SCIENCE PROGRAM IN DATA SCIENCE AND  
ANALYTICS**

**SCHOOL OF SCIENCE**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2024**

**COPYRIGHT OF SCHOOL OF SCIENCE**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

ชื่อเรื่อง	การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำใน โรงไฟฟ้าโดยใช้การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก
นักศึกษา	พลวุฒิ คำฝอย
รหัสประจำตัว	66056057
ปริญญา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการข้อมูลและการ วิเคราะห์
ปีการศึกษา	2567
อาจารย์ที่ปรึกษาหลัก	ผู้ช่วยศาสตราจารย์ ดร.ยุวดี กล่อมวิเศษ

### บทคัดย่อ

การเพิ่มประสิทธิภาพการใช้พลังงานและความน่าเชื่อถือของการดำเนินงานถือเป็นปัจจัยสำคัญในการบริหารจัดการโรงไฟฟ้า โดยเฉพาะอุปกรณ์ที่มีการใช้พลังงานสูงและมีบทบาทสำคัญ เช่น เครื่องสูบน้ำป้อนหม้อไอน้ำ การศึกษานี้นำเสนอการพัฒนาตัวแบบการทำนายการใช้พลังงานโดยใช้เทคนิคการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก เพื่อลดความสูญเสียและตรวจจับความผิดปกติด้านการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ ข้อมูลที่ใช้ในการศึกษาเป็นบันทึกการทำงานรายชั่วโมงจำนวน 43,082 รายการ ครอบคลุมระยะเวลา 5 ปี และมี 18 คุณลักษณะที่สำคัญ กระบวนการศึกษาครอบคลุมการจัดเตรียมและการทำความสะอาดข้อมูล พร้อมด้วยเทคนิคการคัดเลือกคุณลักษณะ ก่อนทำการพัฒนาตัวแบบและประเมินผล โดยตัวแบบที่นำมาเปรียบเทียบประสิทธิภาพ ได้แก่ กลุ่มตัวแบบการถดถอยเชิงเส้น, การถดถอยเชิงเส้นแบบเร็กกิวลารีเซชัน, การถดถอยแบบเคอร์เนลเชิงเส้น, การถดถอยแบบต้นไม้ตัดสินใจ, การถดถอยแบบอองเซมเบิล และโครงข่ายประสาทเทียม ผลการศึกษาพบว่า ตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน มีประสิทธิภาพสูงสุด โดยมีค่า MSE เท่ากับ 13.5573 และค่า  $R^2$  เท่ากับ 0.9838 รองลงมาคือ ตัวแบบการถดถอยแบบไลต์จีบีเอ็ม นอกจากนี้ มีการประยุกต์ใช้ตัวแบบที่มีประสิทธิภาพสูง ในการตรวจจับค่าผิดปกติด้วยวิธี IQR เพื่อระบุความผิดปกติของเครื่องจักรพร้อมระบบการแจ้งเตือน ผู้ปฏิบัติงานก่อนเกิดเหตุฉุกเฉิน ผลลัพธ์ของการศึกษานี้แสดงให้เห็นถึงศักยภาพของการนำการวิเคราะห์เชิงทำนายและการตรวจจับความผิดปกติมาประยุกต์ใช้ในการดำเนินงานของโรงไฟฟ้า ซึ่งสามารถนำไปต่อยอดพัฒนาและปรับปรุงระบบบริหารจัดการพลังงานในอนาคต

**คำสำคัญ:** การทำนายการใช้พลังงาน, เครื่องสูบน้ำป้อนหม้อไอน้ำ, การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก, การตรวจจับค่าผิดปกติ

<b>Title</b>	<b>Energy Consumption Prediction for Boiler Feed Pumps in Power Plants Using Machine Learning and Deep Learning</b>
<b>Student</b>	<b>POLAWUT KHAMFOY</b>
<b>Student ID</b>	<b>66056057</b>
<b>Degree</b>	<b>Master of Science Program in Data Science and Analytics</b>
<b>Academic Year</b>	<b>2024</b>
<b>Advisor</b>	<b>Asst. Prof. Yuwadee Klomwises</b>

## ABSTRACT

Enhancing energy efficiency and operational reliability is a critical factor in power plant management, particularly for high-energy-consuming and essential equipment such as boiler feed water pumps. This study presents the development of energy consumption prediction models using machine learning and deep learning techniques to reduce energy losses and detect anomalies in the power consumption of boiler feed water pumps. The dataset used in this study consists of 43,082 hourly operational records collected over five years, incorporating 18 key operational features. The research process includes data preparation, cleaning, and feature selection techniques, followed by model development and evaluation. The models compared in this study include linear regression models, regularized regression models, kernel-based regression models, decision tree-based regression models, ensemble regression models, and deep neural networks. The study found that Support Vector Regression (SVR) demonstrated the highest prediction accuracy, achieving a MSE of 13.5573 and a  $R^2$  of 0.9838, followed by LightGBM regression. Additionally, the most effective models were applied in anomaly detection using IQR method, enabling the identification of equipment abnormalities and providing an early warning system for operators to take preventive actions before critical failures occur. The findings highlight the potential of predictive analytics and anomaly detection in power plant operations, offering a strong foundation for further development and enhancement of energy management systems in the future.

**Keywords:** Energy Consumption Prediction, Boiler Feed Water Pump, Machine Learning and Deep Learning, Anomaly Detection

## กิตติกรรมประกาศ

การค้นคว้าอิสระเรื่อง "การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้าโดยใช้การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก" ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายฝ่าย ข้าพเจ้าขอแสดงความขอบคุณอย่างจริงใจต่อทุกท่านที่มีส่วนเกี่ยวข้องในการดำเนินงานวิจัยนี้

ข้าพเจ้าขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ยุวดี กล่อมวิเศษ อาจารย์ที่ปรึกษา ซึ่งได้ให้คำแนะนำที่มีคุณค่า ความรู้ และแนวทางในการทำวิจัย รวมถึงให้ข้อเสนอแนะที่เป็นประโยชน์ตลอดระยะเวลาการดำเนินงานวิจัย ทำให้ข้าพเจ้าสามารถพัฒนาผลงานชิ้นนี้ให้สำเร็จลุล่วงไปด้วยดี

ข้าพเจ้าขอกราบขอบพระคุณ คณะกรรมการสอบค้นคว้าอิสระ ผู้ช่วยศาสตราจารย์ ดร.ปัทมา เจริญพร และรองศาสตราจารย์ ดร.ณชญาตา กมลมิธิชม ที่ได้กรุณาให้ข้อเสนอแนะอันเป็นประโยชน์ยิ่งต่อการปรับปรุงและพัฒนาคุณภาพของงานวิจัยให้มีความสมบูรณ์ ข้อคิดเห็นจากคณะกรรมการถือเป็นแนวทางอันมีค่าสำหรับการต่อยอดองค์ความรู้ในอนาคต

ข้าพเจ้าขอขอบพระคุณ บริษัท โกลบอล เพาเวอร์ ซินเนอร์ยี จำกัด ที่ให้การสนับสนุนข้อมูลที่สำคัญต่อการดำเนินการวิจัย โดยเฉพาะอย่างยิ่ง ทีมงานแผนกเทคโนโลยีและดิจิทัลไลเซชัน ที่ได้ให้คำแนะนำและข้อมูลที่เกี่ยวข้องกับกระบวนการการทำงานของโรงไฟฟ้าและเครื่องสูบน้ำป้อนหม้อไอน้ำ ซึ่งมีบทบาทสำคัญอย่างยิ่งต่อความสำเร็จของงานวิจัยนี้

ข้าพเจ้าขอขอบคุณเพื่อนร่วมงานและครอบครัวที่ให้กำลังใจและสนับสนุนข้าพเจ้าในทุก ๆ ด้าน ไม่ว่าจะเป็นด้านวิชาการหรือด้านกำลังใจ ทำให้ข้าพเจ้ามีความมุ่งมั่นและสามารถดำเนินงานวิจัยจนแล้วเสร็จได้

สุดท้ายนี้ ข้าพเจ้าหวังเป็นอย่างยิ่งว่า งานวิจัยฉบับนี้จะเป็นประโยชน์ต่อวงการวิชาการและผู้สนใจนำไปต่อยอดพัฒนาเพื่อการเพิ่มประสิทธิภาพการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้าต่อไป

พลวุฒิ คำฝอย

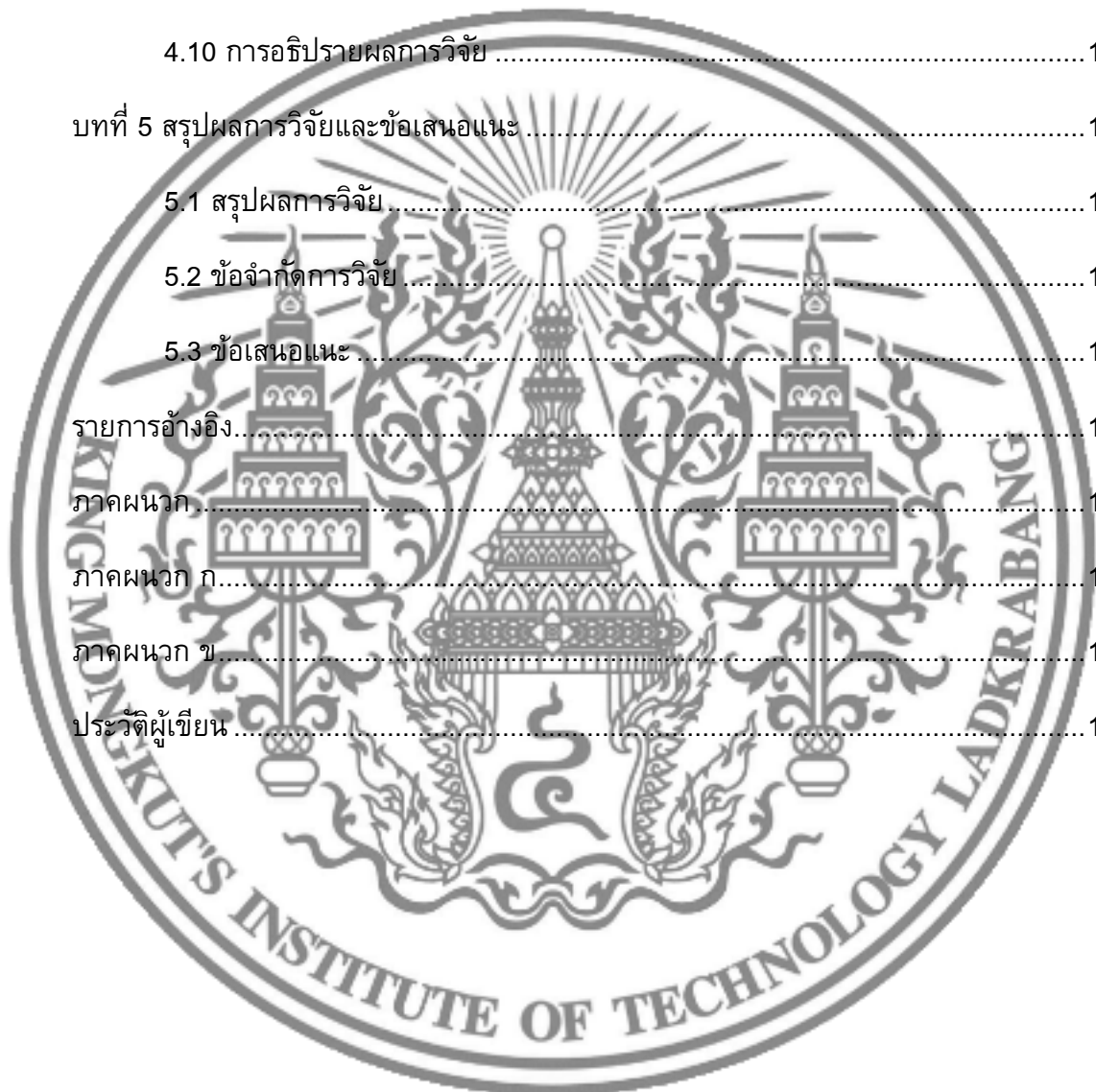
# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	ค
บทคัดย่อภาษาอังกฤษ .....	ง
กิตติกรรมประกาศ .....	จ
สารบัญ .....	ฉ
สารบัญตาราง .....	ฅ
สารบัญรูปภาพ .....	ฎ
บทที่ 1 บทนำ .....	14
1.1 ที่มาและความสำคัญของปัญหา .....	14
1.2 วัตถุประสงค์ของงานค้นคว้าอิสระ .....	15
1.3 ขอบเขตของการจัดสร้างโครงการ .....	15
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	15
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	16
2.1 โรงไฟฟ้าและการผลิตพลังงานไฟฟ้า .....	16
2.2 เครื่องสูบน้ำป้อนหม้อไอน้ำ .....	19
2.3 การเรียนรู้ของเครื่อง .....	22
2.4 การถอดถอยเชิงเส้นอย่างง่าย .....	24
2.5 การถอดถอยเชิงเส้นพหุคูณ .....	26
2.6 การถอดถอยเชิงเส้นด้วยเร็กคิวลาไรซ์เซชัน .....	27
2.7 การถอดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน .....	30
2.8 การถอดถอยแบบต้นไม้ตัดสินใจ .....	32

2.9 การถอดถอยแบบป่าสุ่ม .....	33
2.10 การถอดถอยแบบตเอ็กจิบูส.....	35
2.11 การถอดถอยแบบแคทบูล.....	37
2.12 การถอดถอยแบบไลท์จีบีเอ็ม.....	38
2.13 โครงข่ายประสาทเทียม.....	41
2.14 การเปรียบเทียบประสิทธิภาพการทำนายของตัวแบบ.....	47
2.15 การตรวจสอบค่าความผิดพลาด.....	50
2.16 งานวิจัยที่เกี่ยวข้อง.....	51
บทที่ 3 วิธีการดำเนินงานวิจัย.....	58
3.1 ขั้นตอนการดำเนินงานวิจัย.....	58
3.2 เครื่องมือที่ใช้ในการวิจัย.....	58
3.3 การรวบรวมข้อมูล.....	64
3.4 การจัดเตรียมข้อมูล.....	67
3.5 การพัฒนาตัวแบบการเรียนรู้.....	70
3.6 การวัดประสิทธิภาพการทำนาย.....	73
บทที่ 4 ผลการวิจัยและอภิปรายผล.....	74
4.1 การสร้างตัวแบบถอดถอยเชิงเส้นพหุคูณ.....	74
4.2 การสร้างตัวแบบถอดถอยเชิงเส้นด้วยเร็กคิวลาไรซ์เซชัน.....	79
4.3 การสร้างตัวแบบถอดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน.....	82
4.4 การสร้างตัวแบบถอดถอยแบบต้นไม้ตัดสินใจ.....	85
4.5 การสร้างตัวแบบถอดถอยแบบป่าสุ่ม.....	89
4.6 การสร้างตัวแบบถอดถอยแบบตเอ็กจิบูส.....	93

4.7 การสร้างตัวแบบถดถอยแบบแคทบูล	97
4.8 การสร้างตัวแบบถดถอยแบบโลทิจีบีเอ็ม	100
4.9 การสร้างตัวแบบโครงข่ายประสาทเทียม	104
4.10 การอธิบายผลการวิจัย	107
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	115
5.1 สรุปผลการวิจัย	115
5.2 ข้อจำกัดการวิจัย	116
5.3 ข้อเสนอแนะ	116
รายการอ้างอิง	117
ภาคผนวก	120
ภาคผนวก ก	121
ภาคผนวก ข	125
ประวัติผู้เขียน	149



# สารบัญตาราง

	หน้า
2.1 เปรียบเทียบตัวแบบการถดถอยแบบการเรียนรู้แบบรวมตัว .....	40
2.2 เปรียบเทียบและสรุปสาระสำคัญของงานวิจัยที่เกี่ยวข้อง.....	54
3.1 ไลบรารีพื้นฐานสำหรับการจัดการข้อมูล .....	60
3.2 ไลบรารีสำหรับการแสดงผลข้อมูล.....	60
3.3 ไลบรารีการจัดการข้อมูลและการเตรียมข้อมูล.....	60
3.4 ไลบรารีการสร้างตัวแบบการเรียนรู้.....	61
3.5 ไลบรารีสำหรับการเลือกโมเดลและการปรับค่าพารามิเตอร์.....	62
3.6 ไลบรารีสำหรับการแจ้งเตือนผ่านอีเมล.....	63
3.7 คุณลักษณะของข้อมูลเครื่องสูบน้ำป้อนหม้อไอน้ำ .....	66
3.8 การปรับแต่งพารามิเตอร์ตัวแบบการเรียนรู้ .....	72
4.1 การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่า P-Value $\leq 0.05$ .....	75
4.2 การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่าดัชนี VIF $\leq 10$ (รอบที่ 1) .....	76
4.3 การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่าดัชนี VIF $\leq 10$ (รอบที่ 2) .....	76
4.4 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยเชิงเส้นพหุคูณ .....	77
4.5 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชัน ..	80
4.6 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชัน.....	80
4.7 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีชัพพอร์ตเวกเตอร์แมชชีน .....	82
4.8 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบชัพพอร์ตเวกเตอร์แมชชีน...	83

4.9 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยเชิงเส้นแบบซัพพอร์ตเวกเตอร์แมชชีน.....	84
4.10 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีต้นไม้ตัดสินใจ.....	86
4.11 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบต้นไม้ตัดสินใจ .....	87
4.12 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจ .....	88
4.13 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีป่าสุ่ม .....	90
4.14 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบป่าสุ่ม.....	91
4.15 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่ม .....	91
4.16 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีเอ็กจิบูส.....	94
4.17 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบเอ็กจิบูส .....	94
4.18 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจิบูส.....	95
4.19 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีแคทบูล.....	97
4.20 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบแคทบูล.....	98
4.21 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบแคทบูล.....	99
4.22 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีไลท์จีบีเอ็ม.....	101
4.23 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบไลท์จีบีเอ็ม .....	102
4.24 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบไลท์จีบีเอ็ม.....	102
4.25 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึก .....	106
4.26 เปรียบเทียบประสิทธิภาพการสร้างตัวแบบการเรียนรู้.....	111

# สารบัญรูปภาพ

หน้า

2.1	แผนที่การใช้พลังงานไฟฟ้าต่อประชากรในแต่ละประเทศ .....	17
2.2	โรงไฟฟ้าพลังความร้อนร่วมกำลังการผลิต 920 เมกะวัตต์ประเทศสหรัฐอเมริกา .....	18
2.3	บทบาทของเครื่องสูบน้ำป้อนหม้อไอน้ำในกระบวนการผลิตไฟฟ้าโรงไฟฟ้ากังหันก๊าซ ...	19
2.4	เครื่องสูบน้ำป้อนหม้อไอน้ำในกระบวนการผลิตไฟฟ้า .....	20
2.5	หม้อไอน้ำชนิดกึ่งความร้อนในกระบวนการผลิตไฟฟ้า .....	21
2.6	การแบ่งประเภทของการเรียนรู้ของเครื่อง .....	23
2.7	หลักการการทำงานของตัวแบบการถดถอยเชิงเส้นอย่างง่าย .....	24
2.8	หลักการการทำงานของตัวแบบการถดถอยเชิงเส้นพหุคูณ .....	26
2.9	เทคนิคเร็กคิวลาไรซ์เซชันสำหรับตัวแบบการถดถอยเชิงเส้น .....	27
2.10	เปรียบเทียบการปรับค่าพารามิเตอร์ตัวแบบถดถอยเชิงเส้นด้วยเร็กคิวลาไรซ์เซชัน .....	30
2.11	หลักการการทำงานของตัวแบบซัพพอร์ตเวกเตอร์แมชชีน .....	30
2.12	แปลงข้อมูลแบบไม่เชิงเส้นไปเป็นเชิงเส้นในตัวแบบถดถอยซัพพอร์ตเวกเตอร์แมชชีน ..	32
2.13	หลักการการทำงานของตัวแบบการถดถอยแบบต้นไม้ตัดสินใจ .....	33
2.14	กระบวนการ Bagging สำหรับการฝึกฝนตัวแบบด้วยแนวคิดการเรียนรู้แบบรวม .....	34
2.15	หลักการการทำงานของตัวแบบการถดถอยแบบป่าสุ่ม .....	34
2.16	กระบวนการ Boosting สำหรับการฝึกฝนตัวแบบด้วยแนวคิดการเรียนรู้แบบรวม .....	35
2.17	การสร้างต้นไม้ตัดสินใจในการถดถอยแบบเอ็กจีบูส .....	36
2.18	หลักการการทำงานของตัวแบบการถดถอยแบบเอ็กจีบูส .....	36
2.19	การสร้างต้นไม้ตัดสินใจในการถดถอยแบบแคทบูส .....	37
2.20	กระบวนการ Histogram-based Decision Tree และ Leaf-wise Growth Strategy .....	38

2.21 เซลล์ประสาทชีวภาพสู่เซลล์ประสาทเทียม .....	41
2.22 โครงสร้างชั้นที่สำคัญของโครงข่ายประสาทเทียม .....	42
2.23 กระบวนการเรียนรู้ในโครงข่ายประสาทเทียม.....	43
2.24 กระบวนการลดข้อผิดพลาดตัวแบบโดยใช้ขั้นตอนวิธีหาค่าเหมาะสมที่สุด.....	43
2.25 ความแตกต่างของโครงข่ายประสาทเทียมและโครงข่ายประสาทเทียมเชิงลึก .....	44
2.26 โครงสร้างของโครงข่ายประสาทเทียมเชิงสังวัตนาการ .....	45
2.27 โครงข่ายประสาทเทียมวงกลับ .....	46
2.28 เปรียบเทียบโครงสร้างโครงข่ายประสาทเทียมวงกลับทั้ง 3 รูปแบบ.....	46
2.29 การแบ่งข้อมูลเพื่อทดสอบและวัดประสิทธิภาพตัวแบบ .....	49
2.30 การค้นหาพารามิเตอร์ที่เหมาะสม .....	49
2.31 การแสดงข้อมูลเชิงสถิติโดยใช้แผนภาพกล่องและการตรวจจับค่าพลาดด้วย IQR .....	50
3.1 ขั้นตอนการดำเนินงาน.....	59
3.2 โครงสร้างระบบสารสนเทศโรงไฟฟ้า.....	64
3.3 ตัวอย่างชุดข้อมูลที่บันทึกค่าจากเซ็นเซอร์ในรูปแบบรายชั่วโมง .....	65
3.4 กรอบแนวคิดการจัดเตรียมข้อมูล.....	67
3.5 แผนภาพความสัมพันธ์เชิงเส้นของตัวแปร .....	69
3.6 กรอบแนวคิดการพัฒนาตัวแบบการเรียนรู้.....	70
4.1 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นเชิงเส้นพหุคูณด้วยค่า MSE.....	78
4.2 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยเชิงเส้นพหุคูณด้วยค่า R2 .....	79
4.3 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นแบบเร็กกิวลาไรซ์เซชันด้วยค่า MSE .....	81
4.4 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นแบบเร็กกิวลาไรซ์เซชันด้วยค่า R2 .....	81

4.5	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีนด้วยค่า MSE ....	84
4.6	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีนด้วยค่า MSE ....	85
4.7	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจด้วยค่า MSE.....	88
4.8	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจด้วยค่า R2.....	89
4.9	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่มด้วยค่า MSE .....	92
4.10	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่มด้วยค่า R2 .....	93
4.11	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจีบีเอสด้วยค่า MSE.....	96
4.12	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจีบีเอสด้วยค่า R2.....	97
4.13	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบแคทบูสด้วยค่า MSE .....	99
4.14	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบแคทบูสด้วยค่า R2.....	100
4.15	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบโลจิสติกด้วยค่า MSE.....	103
4.16	เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบโลจิสติกด้วยค่า R2.....	104
4.17	เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึกด้วยค่า MSE .....	106
4.18	เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึกด้วยค่า R2 .....	107
4.19	ประยุกต์ตัวแบบตรวจจับความผิดปกติของการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ.....	109
4.20	กระบวนการยืนยันการตรวจจับความผิดปกติด้วยตัวแบบการทำนายทั้งสามตัวแบบ ...	109
4.21	การแจ้งเตือนเหตุการณ์ผิดปกติของการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ.....	109
4.22	กราฟสรุปการเปรียบเทียบประสิทธิภาพตัวแบบการเรียนรู้ .....	114
ก.1	แบบแสดงแผนการติดตั้งเครื่องสูบน้ำป้อนหม้อไอน้ำ.....	122
ก.2	แบบสัญลักษณ์เพื่อแทนการแสดงรายละเอียดอุปกรณ์ต่างๆที่อยู่ในระบบท่อ และ เครื่องมือวัดทางอุตสาหกรรมของเครื่องสูบน้ำป้อนหม้อไอน้ำ .....	122
ก.3	แบบภาพตัดเครื่องสูบน้ำป้อนหม้อไอน้ำ.....	123

# บทที่ 1

## บทนำ

### 1.1. ที่มาและความสำคัญของปัญหา

การดำเนินงานของโรงไฟฟ้าและการซ่อมบำรุงรักษาเครื่องจักรเป็นปัจจัยสำคัญที่ส่งผลโดยตรงต่อความต่อเนื่องและความมั่นคงในการผลิตไฟฟ้า หนึ่งในอุปกรณ์ที่มีบทบาทสำคัญในกระบวนการผลิตพลังงานไฟฟ้าคือ เครื่องสูบน้ำป้อนหม้อไอน้ำ ซึ่งทำหน้าที่ ป้อนน้ำเข้าสู่หม้อไอน้ำ เพื่อผลิตไอน้ำที่ใช้ในกระบวนการสร้างพลังงานไฟฟ้า เครื่องสูบน้ำป้อนหม้อไอน้ำ ต้องทำงานในสถานะที่มีแรงดันและอุณหภูมิสูง ซึ่งต้องการพลังงานจำนวนมาก การทำงานที่ไม่มีประสิทธิภาพหรือการใช้พลังงานเกินจำเป็นของเครื่องสูบน้ำชนิดนี้ ไม่เพียงเพิ่มต้นทุนการดำเนินงานของโรงไฟฟ้า แต่ยังเป็นสัญญาณของปัญหาหรือความผิดปกติในระบบ เช่น การสึกหรอของชิ้นส่วนเครื่องจักร หรือการตั้งค่าการทำงานที่ไม่เหมาะสม

โรงไฟฟ้ามีหลายประเภท เช่น โรงไฟฟ้าพลังความร้อน โรงไฟฟ้าพลังงานก๊าซ และโรงไฟฟ้าพลังน้ำ ซึ่งแต่ละประเภทต้องใช้เครื่องสูบน้ำป้อนหม้อไอน้ำ ในการทำงานเพื่อส่งน้ำเข้าสู่ระบบหม้อไอน้ำ ความสำคัญของเครื่องสูบน้ำป้อนหม้อไอน้ำ นี้จึงไม่อาจมองข้ามได้ เนื่องจากการทำงานที่มีประสิทธิภาพของเครื่องสูบน้ำป้อนหม้อไอน้ำ ส่งผลโดยตรงต่อความสามารถในการผลิตพลังงานของโรงไฟฟ้า

การทำนายการใช้พลังงานของ เครื่องสูบน้ำป้อนหม้อไอน้ำ มีบทบาทสำคัญในการตรวจสอบวางแผนการดำเนินงาน และการซ่อมบำรุง เพื่อเพิ่มประสิทธิภาพและลดต้นทุนการดำเนินงาน การพัฒนาตัวแบบที่ใช้ การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก โดยใช้ข้อมูลจากเซ็นเซอร์ ได้แก่ อุณหภูมิ การสั่นสะเทือน กระแสไฟฟ้า และความแตกต่างของความดัน เพื่อทำนายค่าพลังงานที่ใช้ของเครื่องสูบน้ำป้อนหม้อไอน้ำ และเปรียบเทียบกับค่าพลังงานที่ใช้จริง กระบวนการนี้ช่วยให้สามารถตรวจสอบความผิดปกติที่อาจเกิดขึ้น เช่น ปัญหาการใช้พลังงานเกินความจำเป็น ปัญหาเชิงกลไก หรือการตั้งค่าการทำงานที่ไม่เหมาะสม ซึ่งการแก้ไขปัญหาเหล่านี้ มีส่วนช่วยให้การดำเนินงานของโรงไฟฟ้ามีความยั่งยืนในระยะยาว ผลลัพธ์จากการพยากรณ์พลังงานนี้ยังช่วยในการวางแผนการบำรุงรักษาและการซ่อมบำรุงเป็นไปอย่างมีประสิทธิภาพมากขึ้น ลดความเสี่ยงจากการหยุดทำงานโดยไม่คาดคิด รวมถึงเพิ่มความปลอดภัยในการดำเนินงาน ส่งผลให้โรงไฟฟ้ามีประสิทธิภาพและความมั่นคงในระบบการผลิตพลังงานไฟฟ้าอย่างต่อเนื่อง

## 1.2. วัตถุประสงค์ของงานวิจัย

- 1) พัฒนาตัวแบบทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้าด้วยเทคนิคการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก
- 2) ประเมิน เปรียบเทียบ และคัดเลือกตัวแบบที่มีประสิทธิภาพและความแม่นยำสูงสุดในการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ
- 3) เพื่อวิเคราะห์ตัวแปรอิสระที่มีผลต่อการพัฒนาตัวแบบ และประยุกต์ใช้ตัวแบบที่ดีที่สุดในการตรวจจับพฤติกรรมความผิดปกติด้านการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า เพื่อแจ้งเตือนความผิดปกติล่วงหน้าแก่ทีมปฏิบัติการ

## 1.3 ขอบเขตของการวิจัย

การศึกษานี้มุ่งเน้นไปที่การพัฒนา ประเมิน เปรียบเทียบ และคัดเลือกตัวแบบการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า โดยใช้ตัวแบบการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก ซึ่งใช้ข้อมูลการทำงานของเครื่องจักรที่รวบรวมได้จากเซ็นเซอร์ของเครื่องสูบน้ำป้อนหม้อไอน้ำ เป็นระยะเวลา 5 ปี จากระบบ PI (Plant information) ของโรงไฟฟ้าในช่วงปี พ.ศ. 2563-2567 ซึ่งได้แก่ ตัวแปรอิสระที่เกี่ยวข้องกับอุณหภูมิที่จุดต่างๆ ค่าการสันสถานะ ค่าแรงดันน้ำออก ค่ากำลังไฟฟ้า และได้ทำการแบ่งกลุ่มของข้อมูลเพื่อวิเคราะห์และสร้างตัวแบบการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ ที่สามารถตรวจจับพฤติกรรมความผิดปกติและระบุแนวโน้มที่อาจนำไปสู่การใช้พลังงานที่เกินความจำเป็น โดยภาษาการเขียนโปรแกรมไพทอน ถูกใช้เป็นภาษาหลักในการพัฒนาโปรแกรมและการวิเคราะห์ จัดการข้อมูล

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้ตัวแบบที่สามารถทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำได้อย่างมีประสิทธิภาพและแม่นยำ
- 2) สามารถตรวจจับความผิดปกติ ด้านการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในระยะแรกเริ่ม ลดค่าใช้จ่ายในการซ่อมแซมและลดการใช้พลังงานที่เกินความจำเป็น
- 3) ได้ข้อมูลเชิงลึกที่ช่วยสนับสนุนการออกแบบกลยุทธ์การประหยัดพลังงานและการดำเนินงานอย่างยั่งยืน
- 4) ส่งเสริมการใช้แนวคิด Digital Transformation ในอุตสาหกรรมพลังงาน เพื่อยกระดับประสิทธิภาพ ความเสถียร ความน่าเชื่อถือ และความปลอดภัย ในการดำเนินงานของโรงไฟฟ้า

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยเรื่อง “การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำโดยใช้การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก” ผู้วิจัยได้รวบรวมแนวคิดทฤษฎี และหลักการจากเอกสารหนังสือและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดของเนื้อหา ดังนี้

1. โรงไฟฟ้าและการผลิตพลังงานไฟฟ้า (Power Plant & Power Generation)
2. เครื่องสูบน้ำป้อนหม้อไอน้ำ (Boiler Feed Pump)
3. การเรียนรู้ของเครื่อง (Machine Learning)
4. การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression)
5. การถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression)
6. การถดถอยเชิงเส้นด้วยเรกิวลารีเซชัน (Linear Regression with Regularization)
7. การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Regression - SVR)
8. การถดถอยแบบต้นไม้ตัดสินใจ (Decision Tree Regressor)
9. การถดถอยแบบป่าสุ่ม (Random Forest Regression)
10. การถดถอยแบบเอ็กซ์ทราโบสต์ (XGBoost Regression)
11. การถดถอยแบบแคตโบสต์ (CatBoost Regression)
12. การถดถอยแบบไลท์จีบีเอ็ม (LightGBM Regression)
13. โครงข่ายประสาทเทียม (Artificial Neural Networks - ANN)
14. การเปรียบเทียบประสิทธิภาพการทำนายของตัวแบบ (Model Evaluation)
15. การตรวจสอบค่าความผิดปกติ (Outlier Detection)
16. งานวิจัยที่เกี่ยวข้อง (Related Research)

#### 2.1 โรงไฟฟ้าและการผลิตพลังงานไฟฟ้า (Power Plant & Power Generation)

การผลิตพลังงานไฟฟ้าเป็นกระบวนการสำคัญที่มีบทบาทในการพัฒนาเศรษฐกิจและสังคมในยุคปัจจุบัน เนื่องจากไฟฟ้าเป็นปัจจัยพื้นฐานที่สนับสนุนกิจกรรมต่าง ๆ ทั้งในชีวิตประจำวันและการพัฒนาอุตสาหกรรม การผลิตไฟฟ้าดำเนินการโดยการเปลี่ยนพลังงานในรูปแบบต่าง ๆ เช่น พลังงานกล พลังงานศักย์ พลังงานเคมี และพลังงานนิวเคลียร์ ให้กลายเป็นพลังงานไฟฟ้าผ่านการทำงานของโรงไฟฟ้า (Electric Power Plant) หรือที่เรียกว่าโรงจักรต้นกำลัง (Power Plant) โรงไฟฟ้าถือเป็น

หัวใจสำคัญในโครงสร้างระบบไฟฟ้าของประเทศ โดยมีผลต่อการเติบโตทางเศรษฐกิจ เพราะการใช้ไฟฟ้าสามารถบ่งบอกถึงความก้าวหน้าทางเศรษฐกิจและคุณภาพชีวิตของประชากรได้.



รูปที่ 2.1 แผนที่การใช้พลังงานไฟฟ้าต่อประชากรในแต่ละประเทศ

(ที่มา: <https://www.indexmundi.com/map/?v=81000>)

โรงไฟฟ้ามักมีการแบ่งประเภทตามลักษณะเชื้อเพลิงและกระบวนการผลิตไฟฟ้าที่ใช้ ซึ่งแต่ละประเภทมีข้อดีและข้อเสียที่แตกต่างกัน โดยสามารถสรุปได้ดังนี้

- 1) โรงไฟฟ้าพลังความร้อน (Thermal Power Plant) ใช้เชื้อเพลิงฟอสซิล เช่น ถ่านหิน น้ำมัน หรือชีวมวล เพื่อผลิตไอน้ำที่มีแรงดันสูงสำหรับหมุนกังหันไอน้ำ ข้อดีคือการผลิตไฟฟ้าในปริมาณมากและต้นทุนเชื้อเพลิงต่ำ แต่มีข้อเสียคือกระบวนการผลิตปล่อยก๊าซเรือนกระจกจำนวนมาก
- 2) โรงไฟฟ้ากังหันก๊าซ (Gas Turbine Power Plant) ใช้ก๊าซธรรมชาติในการเผาไหม้ผลิตก๊าซไอเสียความดันสูง อุณหภูมิสูง เพื่อนำไปหมุนกังหันก๊าซ โรงไฟฟ้านี้มีต้นทุนการก่อสร้างต่ำและใช้เวลาก่อสร้างสั้น แต่ต้นทุนเชื้อเพลิงสูง
- 3) โรงไฟฟ้าพลังความร้อนร่วม (Combined-Cycle Power Plant) รวมกระบวนการผลิตจากกังหันก๊าซและกังหันไอน้ำเข้าด้วยกันเพื่อเพิ่มประสิทธิภาพการผลิตไฟฟ้า ข้อดีคือใช้เชื้อเพลิงอย่างคุ้มค่าและลดการปล่อยก๊าซเรือนกระจก
- 4) โรงไฟฟ้าพลังงานนิวเคลียร์ (Nuclear Power Plant) ใช้ปฏิกิริยานิวเคลียร์ฟิชชันในการผลิตไอน้ำแรงดันสูงสำหรับหมุนกังหัน โรงไฟฟ้านี้ผลิตไฟฟ้าได้จำนวนมากโดยไม่ปล่อยก๊าซเรือนกระจก แต่ต้องการการจัดการด้านความปลอดภัยและการกำจัดกากกัมมันตรังสีที่ซับซ้อน

5) โรงไฟฟ้าพลังงานน้ำ (Hydroelectric Power Plant) ใช้พลังงานศักย์ของน้ำจากเขื่อนในการหมุนกังหัน โรงไฟฟ้าประเภทนี้ไม่มีค่าเชื้อเพลิงและไม่ปล่อยก๊าซเรือนกระจก แต่กระบวนการก่อสร้างอาจส่งผลกระทบต่อสิ่งแวดล้อมอย่างมีนัยสำคัญ

6) โรงไฟฟ้าเครื่องยนต์ดีเซล (Diesel Engine Power Plant) ใช้น้ำมันดีเซลในการเดินเครื่องยนต์ดีเซลขนาดใหญ่ เหมาะสำหรับพื้นที่ห่างไกลหรือใช้เป็นแหล่งพลังงานสำรอง ข้อเสียคือค่าเชื้อเพลิงสูง

7) โรงไฟฟ้าพลังงานหมุนเวียน (Renewable Energy Power Plant) ผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ ลม ความร้อนใต้พิภพ และชีวมวล แม้จะเป็นพลังงานสะอาดและไม่มีค่าใช้จ่ายเชื้อเพลิง แต่ต้นทุนการก่อสร้างยังสูง และประสิทธิภาพการผลิตยังต่ำเมื่อเทียบกับโรงไฟฟ้าประเภทอื่น

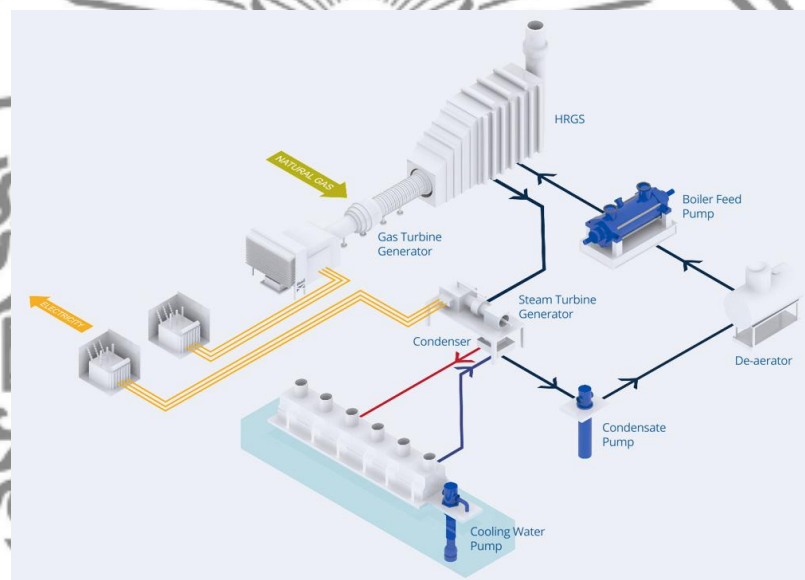
โรงไฟฟ้าส่วนใหญ่ในปัจจุบันใช้กระบวนการผลิตไฟฟ้าโดยการเปลี่ยนพลังงานความร้อนเป็นพลังงานกล ผ่านกระบวนการผลิตไอน้ำ (Steam) เพื่อนำไปขับเคลื่อนกังหันไอน้ำหรือใช้ไอน้ำในกระบวนการอุตสาหกรรมอื่น ๆ ซึ่งโรงไฟฟ้าประเภทนี้ใช้เชื้อเพลิงฟอสซิล เช่น น้ำมัน ถ่านหิน และก๊าซธรรมชาติ เป็นแหล่งพลังงานหลัก การบริหารจัดการพลังงานในโรงไฟฟ้ามีความสำคัญอย่างยิ่งต่อประสิทธิภาพการผลิต การลดต้นทุน และการลดผลกระทบต่อสิ่งแวดล้อม ซึ่งหนึ่งในองค์ประกอบสำคัญของกระบวนการผลิตพลังงานไฟฟ้าและไอน้ำ คือ เครื่องสูบน้ำป้อนหม้อไอน้ำ ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่ส่งน้ำป้อนเข้าสู่หม้อไอน้ำเพื่อใช้ในการผลิตไอน้ำ



รูปที่ 2.2 โรงไฟฟ้าพลังความร้อนร่วมกำลังการผลิต 920 เมกะวัตต์ประเทศสหรัฐอเมริกา  
(ที่มา: <https://www.power-eng.com/gas/combined-cycle/a-report-on-combined-cycle-projects-in-north-america-2/>)

## 2.2 เครื่องสูบน้ำป้อนหม้อไอน้ำ (Boiler Feed Pumps)

เครื่องสูบน้ำป้อนหม้อไอน้ำเป็นองค์ประกอบสำคัญของโรงไฟฟ้าพลังงานความร้อนและพลังงานความร้อนร่วม โดยทำหน้าที่สูบน้ำเข้าสู่หม้อไอน้ำเพื่อเปลี่ยนเป็นไอน้ำแรงดันสูงสำหรับใช้ในกังหันไอน้ำ (Steam Turbine) หรือกระบวนการผลิตไอน้ำในอุตสาหกรรม ความน่าเชื่อถือของเครื่องสูบน้ำป้อนหม้อไอน้ำเป็นสิ่งสำคัญ เนื่องจากต้องทำงานอย่างต่อเนื่องภายใต้แรงดันสูง และสามารถตอบสนองต่อความต้องการโหลดของโรงไฟฟ้าที่เปลี่ยนแปลงได้ (Ku, 2015).



รูปที่ 2.3 บทบาทของเครื่องสูบน้ำป้อนหม้อไอน้ำในกระบวนการผลิตไฟฟ้าของโรงไฟฟ้า  
(ที่มา: <https://www.ruhrpumpen.com/en/markets/power-generation/combined-cycle-power-plants.html>)

### 2.2.1 คุณสมบัติและประเภทของเครื่องสูบน้ำป้อนหม้อไอน้ำ

เครื่องสูบน้ำป้อนหม้อไอน้ำถูกออกแบบให้รองรับสภาวะแรงดันสูงและการเปลี่ยนแปลงโหลดได้ดี ประเภทที่ใช้กันมากที่สุดคือ เครื่องสูบน้ำแรงเหวี่ยงแบบหลายชั้นตอน (Multistage Centrifugal Pumps) ซึ่งสามารถสร้างแรงดันสูงที่เหมาะสมสำหรับจ่ายน้ำเข้าสู่หม้อไอน้ำ โครงสร้างของเครื่องสูบน้ำชนิดนี้อาจเป็นแบบ Diffuser หรือ Volute โดยแบบ Diffuser มักถูกเลือกใช้ในระบบที่ต้องการประสิทธิภาพสูงและลดการสูญเสียพลังงาน เครื่องสูบน้ำเหล่านี้สามารถออกแบบให้เหมาะสมกับความต้องการใช้งานที่แตกต่างกัน ทั้งด้านปริมาณการไหล (Flow Rate) และแรงดัน (Pressure) เช่น การ

ใช้งานในโรงไฟฟ้าขนาดใหญ่หรือโรงงานอุตสาหกรรมที่ต้องการไอน้ำในปริมาณที่เหมาะสม อีกทั้งยังต้องสามารถรองรับการเริ่มต้นและหยุดทำงานบ่อยครั้งเมื่อโหลดมีการเปลี่ยนแปลง

## 2.2.2 ความสำคัญของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า

ในโรงไฟฟ้าเครื่องสูบน้ำป้อนหม้อไอน้ำมีบทบาทสำคัญในการรักษาความต่อเนื่องของการผลิตพลังงานไฟฟ้าและไอน้ำ หากเครื่องสูบน้ำไม่สามารถทำงานได้อย่างมีประสิทธิภาพ อาจทำให้กระบวนการผลิตหยุดชะงัก เกิดความเสียหายต่อผู้ใช้งาน ลูกค้าหรือต่ออุปกรณ์สำคัญ เช่น หม้อไอน้ำ และกังหันไอน้ำ ดังนั้น การออกแบบ การวางแผนและแนวทางการบำรุงรักษาเครื่องสูบน้ำป้อนหม้อไอน้ำจึงเป็นปัจจัยสำคัญ โดยเฉพาะในโรงไฟฟ้าที่ต้องการเพิ่มประสิทธิภาพพลังงานและลดต้นทุนการดำเนินงาน (Li et al., 2018).



รูปที่ 2.4 เครื่องสูบน้ำป้อนหม้อไอน้ำในกระบวนการผลิตไฟฟ้า

## 2.2.3 การใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า

เครื่องสูบน้ำป้อนหม้อไอน้ำเป็นหนึ่งในอุปกรณ์ที่ใช้พลังงานมากที่สุดในระบบหม้อไอน้ำของโรงไฟฟ้า เนื่องจากต้องทำงานภายใต้แรงดันสูงและส่งน้ำเข้าสู่หม้อไอน้ำเพื่อผลิตไอน้ำ ความต้องการพลังงานของเครื่องสูบน้ำขึ้นอยู่กับหลายปัจจัยดังต่อไปนี้

- 1) ปริมาณการไหลและแรงดัน เครื่องสูบน้ำต้องสร้างแรงดันสูงเพื่อเอาชนะแรงดันภายใน

หม้อไอน้ำและการสูญเสียในท่อ (Friction Loss) ยิ่งโรงไฟฟ้ามีขนาดใหญ่หรือหม้อไอน้ำทำงานที่แรงดันสูง เช่น หม้อไอน้ำประเภทซูเปอร์คริติคัล (Supercritical Boilers) หรืออัลตราซูเปอร์คริติคัล (Ultra-Supercritical Boilers) เครื่องสูบน้ำจะต้องใช้พลังงานมากขึ้น

2) ประสิทธิภาพของเครื่องสูบน้ำป้อนหม้อไอน้ำ เครื่องสูบน้ำแรงเหวี่ยงแบบหลายขั้นตอนเป็นชนิดที่ใช้กันอย่างแพร่หลายเนื่องจากมีประสิทธิภาพสูง อย่างไรก็ตาม หากอุปกรณ์เสื่อมสภาพหรือไม่ได้รับการบำรุงรักษาอย่างเหมาะสม ประสิทธิภาพของเครื่องสูบน้ำจะลดลง ทำให้ต้องใช้พลังงานมากขึ้นในการส่งน้ำเข้าสู่หม้อไอน้ำ

3) การควบคุมการทำงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ การใช้งานเครื่องสูบน้ำในช่วงโหลดต่ำ (Partial Load) โดยไม่มีการควบคุมที่เหมาะสม เช่น การใช้วาล์วควบคุมแรงดันแทนการปรับความเร็วรอบ (Variable Frequency Drives, VFD) อาจทำให้เกิดการสูญเสียพลังงาน การปรับความเร็วรอบของเครื่องสูบน้ำป้อนหม้อไอน้ำเป็นวิธีที่ช่วยลดการใช้พลังงานได้อย่างมีประสิทธิภาพ



รูปที่ 2.5 หม้อไอน้ำชนิดกู้คืนความร้อนในกระบวนการผลิตไฟฟ้า  
(Heat Recovery Steam Generator)

(ที่มา: [https://www.bhi.co.kr/en\\_page/img/hrsg\\_img05\\_1.jpg](https://www.bhi.co.kr/en_page/img/hrsg_img05_1.jpg))

## 2.3 การเรียนรู้ของเครื่อง (Machine Learning)

### 2.3.1 นิยามการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง เป็นสาขาย่อยของปัญญาประดิษฐ์ (Artificial Intelligence) ที่เกี่ยวข้องกับการพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถปรับปรุงการทำงานของตนเองได้โดยอาศัยข้อมูลและประสบการณ์ที่ได้รับ (Mitchell, 1997) ความสำคัญของการเรียนรู้ของเครื่องอยู่ที่การออกแบบอัลกอริทึมที่สามารถเรียนรู้รูปแบบที่ซับซ้อนจากข้อมูลปริมาณมากและปรับปรุงความแม่นยำเมื่อได้รับข้อมูลเพิ่มเติม กระบวนการเรียนรู้เริ่มจากการใช้ชุดข้อมูลฝึกสอน (Training Set) พัฒนาแบบจำลองที่สามารถอนุมานถึงเป้าหมาย โดยแบบจำลองดังกล่าวต้องผ่านกระบวนการวัดประสิทธิภาพ เพื่อประเมินประสิทธิภาพในการเรียนรู้ของตัวแบบ (Bishop, 2006)

### 2.3.2 การแก้ไขปัญหาโดยการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง สามารถจัดประเภทปัญหาที่แก้ไขได้เป็นสองรูปแบบหลัก คือ ปัญหาการจำแนกคลาส (Classification Problems) และ ปัญหาการถดถอย (Regression Problems) ซึ่งมีจุดประสงค์และการประยุกต์ใช้ที่แตกต่างกันขึ้นอยู่กับลักษณะของข้อมูลและผลลัพธ์ที่ต้องการ (Hastie, Tibshirani, & Friedman, 2009) สำหรับปัญหาการจำแนกคลาส จุดประสงค์ของการเรียนรู้คือการทำนายค่าข้อมูลเชิงคุณภาพ (Qualitative Data) ซึ่งมักแสดงในรูปแบบของกลุ่มหรือประเภท เช่น การจำแนกอีเมลว่าเป็นสแปมหรือไม่ การระบุว่าลูกค้าจะเลิกใช้บริการหรือไม่ และการวิเคราะห์ความคิดเห็นจากข่าวสารเพื่อระบุว่าเป็นเชิงบวกหรือเชิงลบ ตัวอย่างที่สำคัญอีกตัวหนึ่งคือการจำแนกภาพทางการแพทย์ เช่น การระบุว่าภาพถ่ายรังสีแสดงถึงก้อนเนื้อร้ายหรือก้อนเนื้อปกติ (Nguyen et al., 2019) ในขณะเดียวกัน ปัญหาการถดถอยมุ่งเน้นการทำนายค่าข้อมูลเชิงปริมาณ (Quantitative Data) เช่น การทำนายระดับมลพิษในกรุงเทพมหานคร หรือการประเมินราคาของอสังหาริมทรัพย์

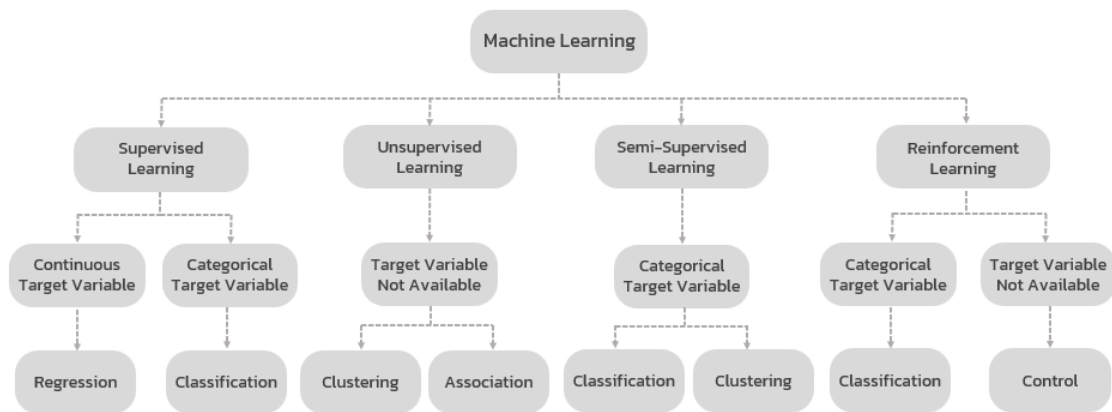
### 2.3.3 ชุดข้อมูล

ชุดข้อมูล เป็นองค์ประกอบที่สำคัญสำหรับกำหนดประสิทธิภาพของตัวแบบ โดยชุดข้อมูลจะประกอบด้วย ตัวอย่างข้อมูลจำนวนหนึ่ง ซึ่งแต่ละตัวอย่างมีรายละเอียดที่เรียกว่าฟีเจอร์ (Feature) หรือในเชิงสถิติเรียกว่า ตัวแปรอิสระ (Independent Variable) และผลลัพธ์เป้าหมายที่ต้องการเรียนรู้เรียกว่า ตัวแปรตาม (Dependent Variable) หรือคลาส (Class) ทั้งนี้ ประสิทธิภาพของกระบวนการเรียนรู้ขึ้นอยู่กับคุณลักษณะสำคัญของชุดข้อมูลในสามประเด็นหลัก ดังนี้ (Hastie, Tibshirani, & Friedman, 2009)

- 1) จำนวนตัวอย่างของชุดข้อมูล ชุดข้อมูลที่มีตัวอย่างไม่เพียงพออาจส่งผลให้ตัวแบบไม่สามารถเรียนรู้หรือตัดสินใจได้อย่างแม่นยำ โดยเฉพาะเมื่อชุดข้อมูลมีฟีเจอร์จำนวนมากหรือฟีเจอร์ที่มีค่าความเป็นไปได้หลายค่า เช่น ปัญหาที่ต้องวิเคราะห์ข้อมูลเชิงซับซ้อน การมีจำนวนตัวอย่างมากเพียงพอ จะช่วยเพิ่มความสามารถในการเรียนรู้ของตัวแบบ (Goodfellow et al., 2016)
- 2) ความถูกต้องของข้อมูล ส่งผลโดยตรงต่อประสิทธิภาพของการเรียนรู้ ข้อมูลที่มีค่าผิดพลาด เช่น ค่าผิดพลาดจากการนำเข้าสู่ข้อมูลของผู้ใช้งาน หรือ ค่าที่ผิดพลาดจากความคลาดเคลื่อนของเครื่องมือวัด ลดทอนประสิทธิภาพของตัวแบบ โดยทั่วไป จำเป็นต้องมีการตรวจสอบและจัดการข้อมูลก่อนนำข้อมูลไปสร้างตัวแบบ (Zhang et al., 2016).
- 3) ความสมดุลของตัวแปรเป้าหมาย ปัญหาการจำแนกประเภท ความไม่สมดุลของจำนวนตัวอย่างในแต่ละคลาสส่งผลต่อประสิทธิภาพของตัวแบบ จึงต้องมีการใช้เทคนิคการสุ่มเพิ่มหรือลดตัวอย่าง (He & Garcia, 2009)

### 2.3.4 ประเภทของการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่องแบ่งออกเป็น 4 ประเภทหลัก (Goodfellow et al., 2016)



รูปที่ 2.6 การแบ่งประเภทของการเรียนรู้ของเครื่อง

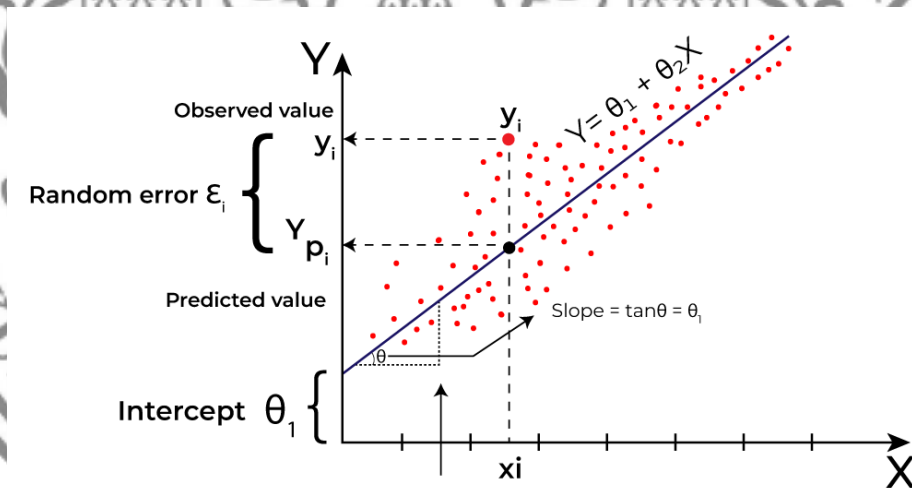
- 1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) ใช้ชุดข้อมูลฝึกสอนที่มีป้ายกำกับ (Label) หรือผลเฉลยในแต่ละตัวอย่าง เพื่อให้ตัวแบบเรียนรู้ความสัมพันธ์ระหว่างฟีเจอร์และตัวแปรเป้าหมาย โดยตัวแบบที่นิยมใช้ได้แก่ การถดถอยเชิงเส้น, นาอ์ฟเบย์, ต้นไม้ตัดสินใจ และ ขั้นตอนวิธีการค้นหาเพื่อนบ้านใกล้สุด  $k$  ตัว (Hastie et al., 2009)

2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ใช้ชุดข้อมูลฝึกสอนที่ไม่มีป้ายกำกับ หรือผลเฉลยในแต่ละตัวอย่าง แต่เน้นการวิเคราะห์โครงสร้างหรือรูปแบบในข้อมูล เช่น การจัดกลุ่ม (Clustering) หรือการลดมิติ (Dimensionality Reduction) โดยตัวแบบที่นิยมใช้ ได้แก่ การจัดกลุ่มแบบเคมีน, และการเรียนรู้กฎความเกี่ยวพัน (Aggarwal & Reddy, 2013)

3) การเรียนรู้แบบกึ่งสอน (Semi-Supervised Learning) ใช้ข้อมูลฝึกสอนที่มีป้ายกำกับเพียงบางส่วนและข้อมูลที่ไม่มีป้ายกำกับช่วยในการเรียนรู้ เหมาะกับกรณีที่มีข้อมูลจำนวนมาก เช่น การจำแนกประเภทเว็บไซต์ โดยตัวแบบที่นิยมใช้ ได้แก่ Co-Training (Zhu, 2005)

4) การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) วิธีการนี้ ไม่มีต้องป้ายกำกับหรือผลเฉลยในแต่ละตัวอย่างในการเรียนรู้ แต่ใช้กลไกการให้รางวัลหรือการลงโทษเพื่อเรียนรู้พฤติกรรม เช่น การขับรถอัตโนมัติ โดยตัวแบบที่นิยมใช้ ได้แก่ Q-Learning และ Deep Q-Networks (DQN) (Sutton & Barto, 2018)

#### 2.4 การถดถอยเชิงเส้นอย่างง่าย (Simple Linear Regression)



รูปที่ 2.7 หลักการทำงานของตัวแบบการถดถอยเชิงเส้นอย่างง่าย  
(ที่มา: <https://www.seeksforgeeks.org/ml-linear-regression>)

การถดถอยเชิงเส้นอย่างง่ายเป็นวิธีพื้นฐานในการสร้างตัวแบบความสัมพันธ์เชิงเส้นระหว่างตัวแปรตามและตัวแปรอิสระ เพียงตัวเดียวโดยสมมติว่าความสัมพันธ์ระหว่างตัวแปรทั้งสองเป็นเชิงเส้นอย่างชัดเจน วิธีนี้มีรากฐานมาจากสถิติและได้รับการนำไปใช้ในงานด้านการเรียนรู้ของเครื่องสำหรับการทำนายและวิเคราะห์ข้อมูลเชิงปริมาณ (Hastie, Tibshirani, & Friedman, 2009) พื้นฐานของการถดถอยเชิงเส้นคือการหาเส้นตรงที่ดีที่สุดเพื่ออธิบายความสัมพันธ์ระหว่างตัวแปร โดยเส้นตรงนี้แทนด้วยสมการ

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (2.1)$$

โดยที่

$y$  คือ ค่าของตัวแปรตาม

$x$  คือ ค่าของตัวแปรอิสระ

$\beta_0$  คือ ส่วนตัดแกน  $y$  หรือค่าของ  $y$  เมื่อ  $x$  เป็น 0

$\beta_1$  คือ ค่าความชันของเส้นตรงแสดงถึงการเปลี่ยนแปลงของ  $y$  เมื่อ  $x$  เปลี่ยนไปหนึ่งหน่วย

$\varepsilon$  คือ ค่าความผิดพลาดหรือความคลาดเคลื่อนแบบสุ่ม (Random Error)

การหาเส้นตรงที่ดีที่สุดทำได้โดยวิธีกำลังสองน้อยที่สุด (Least Squares Method) ซึ่งมีเป้าหมายเพื่อลดผลรวมของกำลังสองของความคลาดเคลื่อน (Sum of Squared Errors, SSE) ให้มีค่าต่ำสุดดังสมการ

$$SSE = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2 \quad (2.2)$$

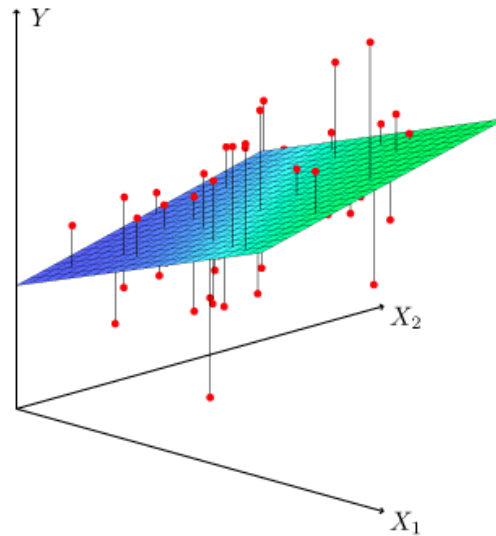
โดยค่าพารามิเตอร์  $\beta_0$  และ  $\beta_1$  คำนวณได้จากสมการ

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (2.3)$$

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.4)$$

โดยที่  $\bar{x}$  และ  $\bar{y}$  แทนค่าเฉลี่ยของตัวแปรอิสระและตัวแปรตามตามลำดับ ซึ่งสมมติฐานที่สำคัญของการถดถอยเชิงเส้นประกอบด้วยความสัมพันธ์เชิงเส้นของตัวแปรตามและตัวแปรอิสระต้องเป็นเชิงเส้น (Linear Relationship), ค่าความคลาดเคลื่อน  $\varepsilon$  ต้องเป็นอิสระจากกัน (Independence), มีความแปรปรวน ( $\sigma^2$ ) คงที่ (Homoscedasticity), มีการแจกแจงแบบปกติ (Normality)

## 2.5 การถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression)



รูปที่ 2.8 หลักการทำงานของตัวแบบการถดถอยเชิงเส้นแบบพหุคูณ  
(ที่มา: <https://statsandr.com/blog/multiple-linear-regression-made-simple/>)

การถดถอยเชิงเส้นพหุคูณเป็นการขยายแนวคิดของการถดถอยเชิงเส้นอย่างง่าย โดยเพิ่มจำนวนตัวแปรอิสระมากกว่าหนึ่งตัว เพื่อสร้างตัวแบบที่สามารถอธิบายความสัมพันธ์ระหว่างตัวแปรตามกับตัวแปรอิสระหลายตัวได้ สมการของแบบจำลองมีรูปแบบดังนี้

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \epsilon \quad (2.5)$$

โดยที่

- $y$  คือ ค่าของตัวแปรตาม
- $x_1, x_2, \dots, x_k$  คือ ค่าของตัวแปรอิสระ
- $\beta_0$  คือ ส่วนตัดแกน  $y$  หรือค่าของ  $y$  เมื่อ  $x_1, x_2, \dots, x_k = 0$
- $\beta_1, \beta_2, \dots, \beta_k$  คือ สัมประสิทธิ์ถดถอยที่แสดงถึงการเปลี่ยนแปลงของ  $y$  เมื่อ  $x_i$  เปลี่ยนไปหนึ่งหน่วย โดยที่ตัวแปรอิสระอื่นคงที่

ค่าพารามิเตอร์  $\beta_0, \beta_1, \dots, \beta_k$  คำนวณโดยวิธีกำลังสองน้อยที่สุดเพื่อหาผลรวมของกำลังสองของค่าความคลาดเคลื่อนให้มีค่าน้อยที่สุด

$$SSE = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}))^2 \quad (2.6)$$

การคำนวณ  $\beta$  สามารถทำได้โดยการใช้เมทริกซ์เชิงเส้น เพื่อแก้ปัญหาในลักษณะทั่วไป

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (2.7)$$

โดยที่

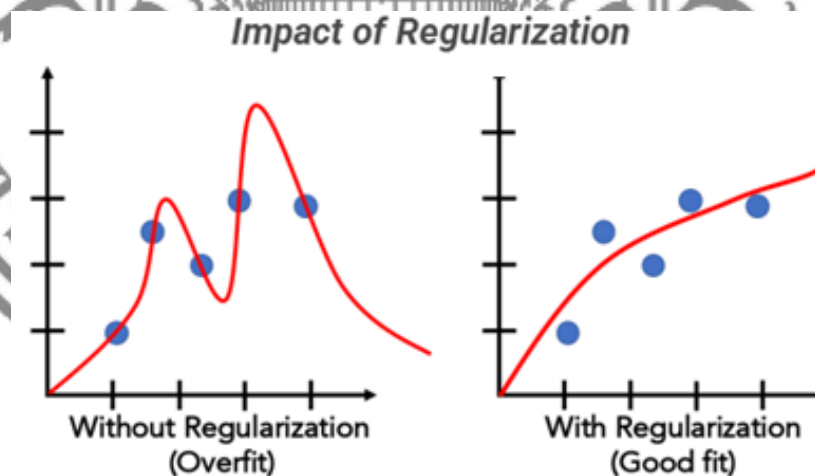
$\mathbf{X}$  คือ เมทริกซ์ของตัวแปรอิสระ

$\mathbf{y}$  คือ เวกเตอร์ของตัวแปรตาม

$\beta$  คือ เวกเตอร์ของสัมประสิทธิ์ถดถอย

สมมุติฐานสำคัญของการถดถอยเชิงเส้นพหุคูณคล้ายกับการถดถอยเชิงเส้นอย่างง่าย ได้แก่ ความสัมพันธ์เชิงเส้นของตัวแปรตามและตัวแปรอิสระต้องเป็นเชิงเส้น, ค่าความคลาดเคลื่อน  $\epsilon$  ต้องเป็นอิสระจากกัน, ค่าความคลาดเคลื่อน  $\epsilon$  มีความแปรปรวน  $\sigma^2$  คงที่, มีการแจกแจงแบบปกติ และตัวแปรอิสระไม่ควรมีความสัมพันธ์เชิงเส้นสูงระหว่างกัน โดยสามารถตรวจสอบด้วยวิธีปัจจัยการขยายตัวของความแปรปรวน (Variance Inflation Factor) (มัลลิกา บุนนาค, 2551)

## 2.6 การถดถอยเชิงเส้นด้วยเร็กกิวลารีเซชัน (Linear Regression with Regularization)



รูปที่ 2.9 เทคนิคเร็กกิวลารีเซชันสำหรับตัวแบบการถดถอยเชิงเส้น

(ที่มา: <https://www.analyticsvidhya.com/blog/2021/11/study-of-regularization-techniques-of-linear-model-and-its-roles/>)

การถดถอยเชิงเส้นด้วยเร็กกิวลาไรซ์เซชันเป็นเทคนิคที่เพิ่มข้อจำกัดในกระบวนการประมาณค่าพารามิเตอร์ของตัวแบบถดถอยเชิงเส้น เพื่อลดปัญหาความซับซ้อนเกินไปของตัวแบบ (Overfitting) โดยเฉพาะในกรณีที่มีตัวแปรอิสระจำนวนมากหรือมีความสัมพันธ์กันสูง ฟังก์ชันเป้าหมายของเร็กกิวลาไรซ์เซชันจะมีฟังก์ชันลงโทษ (Penalty Function) เพิ่มเข้ามา ดังนี้

$$\text{Objective Function} = \text{SSE} + \lambda \cdot \text{Penalty Term} \quad (2.8)$$

โดยที่  $\lambda$  คือ พารามิเตอร์ที่ควบคุมความเข้มงวดของบทลงโทษ ( $\lambda > 0$ ) และ Penalty Term มีลักษณะแตกต่างกันขึ้นอยู่กับประเภทของเร็กกิวลาไรซ์เซชัน

### 2.6.1 การถดถอยเชิงเส้นแบบลาซโซ (Lasso Regression)

การถดถอยเชิงเส้นแบบลาซโซ เป็นตัวแบบที่ถูกพัฒนาขึ้นเพื่อจัดการกับปัญหาความซับซ้อนของตัวแบบที่เกิดจากการมีตัวแปรอิสระจำนวนมาก ตัวแบบถดถอยเชิงเส้น ค่าสัมประสิทธิ์ ( $\beta$ ) ของตัวแปรอิสระแต่ละตัวอาจมีค่ามากหรือน้อยแตกต่างกัน ซึ่งในบางกรณีพบว่าตัวแปรอิสระบางตัวมีค่าสัมประสิทธิ์ที่น้อยมาก หรือใกล้ศูนย์ ซึ่งบ่งชี้ว่าตัวแปรเหล่านั้นอาจไม่มีผลสำคัญต่อตัวแปรตาม เพื่อจัดการกับปัญหาดังกล่าว การถดถอยเชิงเส้นแบบลาซโซ จึงใช้ฟังก์ชันลงโทษที่แบบ L1-Regularization ซึ่งคำนวณจากผลรวมของค่าสัมบูรณ์ของสัมประสิทธิ์  $\beta$  ทำให้บางค่าของ  $\beta$  กลายเป็นศูนย์ ส่งผลให้ตัวแบบสามารถคัดเลือกตัวแปรอัตโนมัติได้ (Tibshirani, 1996)

$$\text{Objective Function} = \text{SSE} + \lambda \cdot \sum_{j=1}^k |\beta_j| \quad (2.9)$$

### 2.6.2 การถดถอยเชิงเส้นแบบริดจ์ (Ridge Regression)

การถดถอยเชิงเส้นแบบริดจ์ เป็นตัวแบบที่ถูกพัฒนาขึ้นเพื่อจัดการกับปัญหาความสัมพันธ์เชิงเส้นสูง ระหว่างตัวแปรอิสระซึ่งอาจส่งผลให้ค่าประมาณของสัมประสิทธิ์ ( $\beta$ ) ในการถดถอยเชิงเส้นไม่มีเสถียรภาพ และยากต่อการตีความ การถดถอยแบบริดจ์แก้ไขปัญหาดังกล่าวโดยการใช้ฟังก์ชันลงโทษแบบ L2-Regularization ซึ่งคำนวณจากผลรวมของกำลังสองของสัมประสิทธิ์  $\beta$  ทำให้ตัวแบบมีเสถียรภาพมากขึ้น แม้จะเพิ่มค่าอคติ (Bias) เล็กน้อย (Hoerl & Kennard, 1970)

$$\text{Objective Function} = \text{SSE} + \lambda \cdot \sum_{j=1}^k \beta_j^2 \quad (2.10)$$

### 2.6.3 การถดถอยแบบอิลาสติกเน็ต (Elastic Net Regression)

การถดถอยแบบอิลาสติกเน็ต เป็นตัวแบบที่พัฒนาขึ้นมาเพื่อตอบโจทย์ข้อจำกัดของ การถดถอยเชิงเส้นแบบลาซโซ และการถดถอยเชิงเส้นแบบริดจ์ โดยเฉพาะในกรณีที่ข้อมูลมีตัวแปรอิสระจำนวนมากและมีความสัมพันธ์เชิงเส้นสูง การถดถอยแบบอิลาสติกเน็ต ผสมผสานระหว่างการใช้

ฟังก์ชันลงโทษ เพื่อรวมข้อดีของทั้งสองวิธีในการจัดการกับปัญหาความซับซ้อนของตัวแบบและการคัดเลือกตัวแปร ซึ่งมีรูปแบบดังนี้

$$\text{Objective Function} = \text{SSE} + \lambda (\alpha \sum_{j=1}^k |\beta_j| + (1 - \alpha) \sum_{j=1}^k \beta_j^2) \quad (2.11)$$

โดยที่  $\alpha$  เป็นพารามิเตอร์ที่ควบคุมสัดส่วนระหว่าง L1 และ L2-Regularization (Zou & Hastie, 2005)

#### 2.6.4 การเลือกค่าพารามิเตอร์ในเร็กกิวลาไรซ์เซชัน

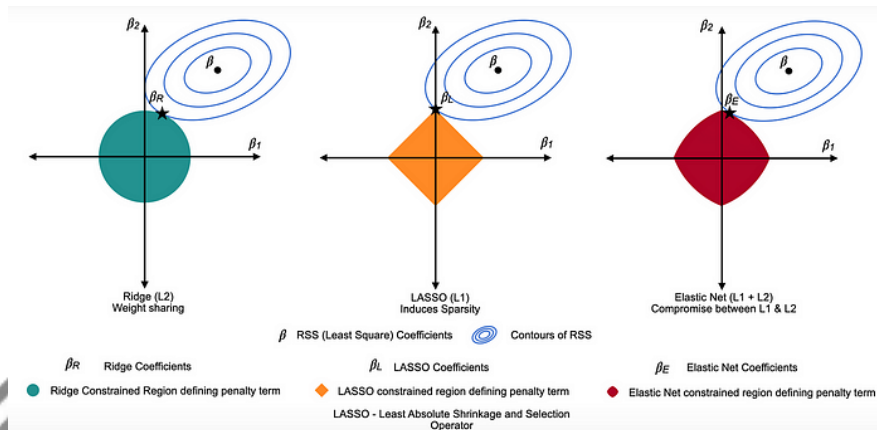
การเลือกค่าพารามิเตอร์สำหรับตัวแบบการถดถอยเชิงเส้นด้วยเร็กกิวลาไรซ์เซชัน เป็นขั้นตอนสำคัญที่ส่งผลต่อประสิทธิภาพของตัวแบบ โดยเฉพาะใน การถดถอยแบบอิลาสติกเน็ต ซึ่งทั้งสองค่าควบคุม ระดับความเข้มงวด และ ลักษณะของบทลงโทษที่ใช้ในตัวแบบ เพื่อให้ได้ค่าที่เหมาะสมที่สุด

##### 1) การเลือกค่าพารามิเตอร์ควบคุมความเข้มงวดของบทลงโทษ ( $\lambda$ )

$\lambda$  เป็นตัวกำหนดระดับความเข้มงวดของบทลงโทษ หาก  $\lambda = 0$  ตัวแบบจะไม่มีบทลงโทษ ทำให้ผลลัพธ์เหมือนกับการถดถอยเชิงเส้นปกติ (Ordinary Least Squares, OLS) แต่หาก  $\lambda$  มีค่าสูงมาก ตัวแบบจะลดค่าสัมประสิทธิ์ ( $\beta$ ) ของตัวแปรลงจนใกล้ศูนย์หรือเป็นศูนย์ ซึ่งการเลือกค่า  $\lambda$  ที่เหมาะสมมักทำผ่านการตรวจสอบความสมเหตุสมผลแบบไขว้จำนวนเคพับ (K-fold Cross-Validation) โดยแบ่งข้อมูลออกเป็น  $k$  ชุดย่อย ใช้  $k - 1$  ชุดในการฝึกตัวแบบ และใช้ชุดที่เหลือในการทดสอบ ซ้ำกระบวนการนี้หลายครั้งสำหรับค่าของ  $\lambda$  ที่ต่างกัน และเลือกค่าที่ให้ตัวแบบมีประสิทธิภาพมากที่สุด (Friedman et al., 2010)

##### 2) การเลือกค่าพารามิเตอร์กำหนดสัดส่วนระหว่าง L1 และ L2-Regularization

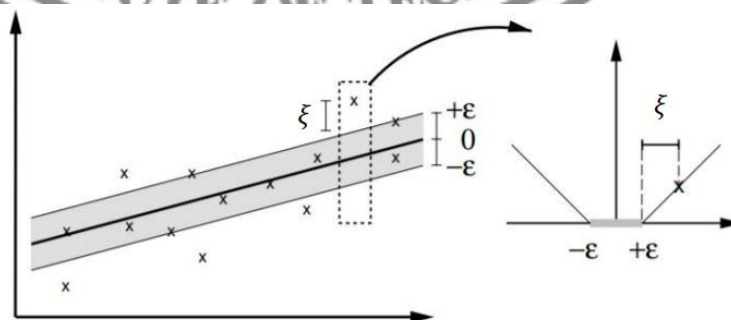
$\alpha$  เป็นตัวกำหนดว่าส่วนประกอบของบทลงโทษจะเป็นแบบ L1 และ L2-Regularization ในสัดส่วนเท่าใด โดย  $\alpha$  ใกล้ 1 ตัวแบบมีความสามารถในการคัดเลือกตัวแปรได้ดี แต่เสี่ยงต่อการละเว้นตัวแปรสำคัญที่มีความสัมพันธ์กัน ในกรณีที่  $\alpha$  ใกล้ 0 ตัวแบบจัดการกับ ปัญหาตัวแปรอิสระมีความสัมพันธ์เชิงเส้นสูงระหว่างกันได้ดี แต่ไม่สามารถลดตัวแปรที่ไม่สำคัญออกได้ ซึ่งการเลือกค่า  $\alpha$  ที่เหมาะสมมักทำผ่านการตรวจสอบความสมเหตุสมผลแบบไขว้จำนวนเคพับ จะช่วยให้ทราบค่าสัดส่วนที่เหมาะสมที่สุด (Zou & Hastie, 2005)



รูปที่ 2.10 เปรียบเทียบวิธีการปรับค่าพารามิเตอร์ในตัวแบบการถดถอยเชิงเส้นด้วยเร็กคิวลารีเซชัน (ที่มา: <https://medium.com/@tavishi.1402/regularization-techniques-lasso-and-ridge-90d3cc73ca4c>)

## 2.7 การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Regression)

การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Regression, SVR) เป็นตัวแบบที่พัฒนามาจากตัวแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machines) ซึ่งเดิมถูกออกแบบมาเพื่อการจำแนกประเภท แต่ถูกปรับให้สามารถใช้ในการพยากรณ์ค่าตัวแปรตามได้ โดยสามารถทำงานได้ทั้งในกรณีของข้อมูลเชิงเส้นและไม่เชิงเส้น ตัวแบบ SVR มีความสามารถในการจัดการกับข้อมูลที่มีมิติสูงและซับซ้อน โดยใช้หลักการของระนาบเชิงเส้นที่เหมาะสมที่สุดในการพยากรณ์ค่าของตัวแปรตาม และกำหนดขอบเขตข้อผิดพลาด ( $\epsilon$ ) เพื่อควบคุมความแม่นยำและความเรียบง่ายของตัวแบบ (Drucker et al., 1997; Smola & Schölkopf, 2004)



รูปที่ 2.11 หลักการทำงานของตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ที่มา: <https://qph.cf2.quoracdn.net/main-qimg-1e97a0c1fc7c5bac2e67bcbe2ec08d82>)

พื้นฐานของตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน ใช้ฟังก์ชันเป้าหมายสร้าง  
 ระนาบ ที่เหมาะสมที่สุดในการทำนายค่าของตัวแปรตาม โดยระนาบนี้ถูกกำหนดโดยสมการ

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (2.12)$$

โดยที่

$\mathbf{w}$	คือ เวกเตอร์น้ำหนัก (Weight Vector) ที่กำหนดทิศทางของระนาบ
$\mathbf{x}$	คือ ค่าของตัวแปรอิสระ
$\langle \mathbf{w}, \mathbf{x} \rangle$	คือ ผลคูณเชิงจุด (Dot Product) ระหว่าง $\mathbf{w}$ และ $\mathbf{x}$
$b$	คือ ค่าอคติ (Bias) ที่ช่วยปรับตำแหน่งของระนาบ

ระนาบ  $f(\mathbf{x})$  ถูกสร้างขึ้นเพื่อทำนายค่าของตัวแปรตาม  $y$  ได้ โดยพยายามลดข้อผิดพลาดในการทำ  
 โดยอนุญาตให้ค่าทำนายอยู่ภายในขอบเขตข้อผิดพลาดที่กำหนด ( $\epsilon$ ) และลดบทลงโทษสำหรับค่าที่อยู่  
 นอกขอบเขตนี้ โดยฟังก์ชันเป้าหมายมีรูปแบบดังนี้

$$\text{Minimize: } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (2.13)$$

โดยที่

$\ \mathbf{w}\ ^2$	ค่า แทนความซับซ้อนของตัวแบบ
$C$	คือ ค่าควบคุมความสมดุลระหว่างความซับซ้อนของตัวแบบและข้อผิดพลาด หาก มี ค่าสูงตัวแบบจะมีความยืดหยุ่นต่ำและพยายามลดข้อผิดพลาดให้ต่ำที่สุด
$\xi_i, \xi_i^*$	คือ ค่าส่วนขาด (Slack Variable) ใช้จัดการกับจุดข้อมูลที่อยู่นอกเขตข้อผิดพลาด ที่กำหนด
$\epsilon$	คือ ควบคุมขอบเขตที่อนุญาตให้ค่าพยากรณ์มีความคลาดเคลื่อน ตัวแบบจะ พยายามให้ค่าทำนายอยู่ภายในขอบเขตที่กำหนดนี้

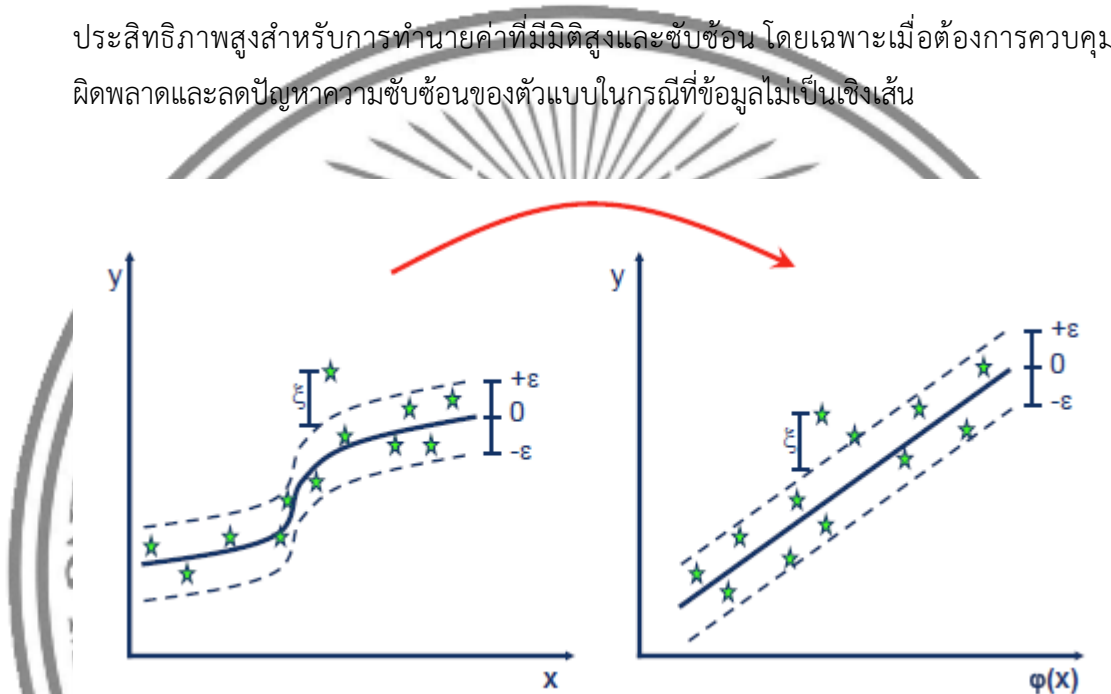
โดยระนาบ  $f(\mathbf{x})$  ต้องรองรับข้อจำกัด (Constrain) ดังนี้

$$y_i - \langle \mathbf{w}, \mathbf{x} \rangle - b \leq \epsilon + \xi_i \quad (2.14)$$

$$\langle \mathbf{w}, \mathbf{x} \rangle + b - y_i \leq \epsilon + \xi_i^* \quad (2.15)$$

$$\xi_i, \xi_i^* \geq 0 \quad (2.16)$$

เพื่อเพิ่มความยืดหยุ่นของตัวแบบ สามารถเลือกใช้ฟังก์ชันเคอร์เนล (Kernel Function) เพื่อช่วยแปลงข้อมูลไปยังพื้นที่ที่สามารถจำแนกหรือทำนายได้ดีขึ้น ซึ่งเหมาะสำหรับข้อมูลที่มีความซับซ้อนสูง เช่น เคอร์เนลแบบพหุนาม (Polynomial Kernel) และเคอร์เนลแบบเรเดียลเบส (Radial Basis Function, RBF) การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน จึงเป็นตัวแบบที่มีประสิทธิภาพสูงสำหรับการทำนายค่าที่มีมิติสูงและซับซ้อน โดยเฉพาะเมื่อต้องการควบคุมความผิดพลาดและลดปัญหาความซับซ้อนของตัวแบบในกรณีที่ข้อมูลไม่เป็นเชิงเส้น



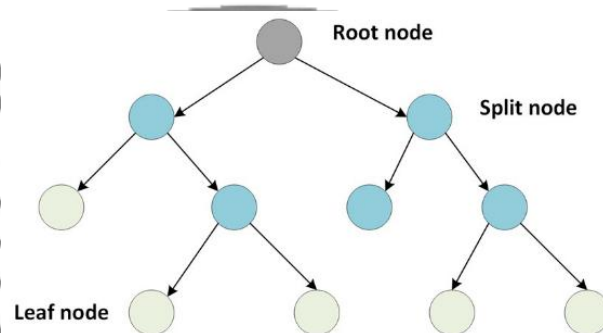
รูปที่ 2.12 แปลงข้อมูลแบบไม่เชิงเส้นไปเป็นเชิงเส้นในตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ที่มา: [https://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](https://www.saedsayad.com/support_vector_machine_reg.htm))

## 2.8 การถดถอยแบบต้นไม้ตัดสินใจ (Decision Tree Regressor)

การถดถอยแบบต้นไม้ตัดสินใจ เป็นตัวแบบการเรียนรู้ของเครื่องที่ใช้โครงสร้างแบบต้นไม้ (Tree Structure) ในการทำนายค่าตัวแปรตาม จากค่าของตัวแปรอิสระโดยการแบ่งข้อมูลออกเป็นส่วนย่อย (Partitions) ตามคุณลักษณะที่สำคัญที่สุดในแต่ละระดับของโครงสร้างต้นไม้ การแบ่งนี้ช่วยลดความแปรปรวนและเพิ่มความแม่นยำของการทำนาย (Breiman et al., 1984)

การสร้างต้นไม้ตัดสินใจเริ่มจากโหนดราก (Root Node) ซึ่งประกอบด้วยข้อมูลทั้งหมดในชุดข้อมูล ทำการแบ่งข้อมูลโดยการเลือกคุณลักษณะที่มีความสำคัญสูงสุด หรือมีผลต่อค่าตัวแปรตามมากที่สุด การแบ่งนี้จะดำเนินการซ้ำในแต่ละโหนดย่อย (Child Nodes) จนกว่าจะถึงเงื่อนไขหยุด เช่น ไม่มีข้อมูลเหลือสำหรับการแบ่งเพิ่ม หรือถึงความลึกสูงสุดที่กำหนด เกณฑ์การแบ่งข้อมูล คำนวณค่าผิดพลาดกำลังสอง (MSE) วัดความแปรปรวนของค่าตัวแปรตามในแต่ละโหนด และเลือกการแบ่งที่

ลดค่า MSE ได้มากที่สุด เพื่อให้โหนดสุดท้ายมีค่าที่ใกล้เคียงกับค่าจริงมากที่สุด เมื่อสร้างต้นไม้เสร็จ การทำนายค่าจะดำเนินการโดยนำค่าตัวแปรอิสระเข้าสู่ต้นไม้ แล้วไล่ลงมาตามโหนดที่ตรงกับเงื่อนไขจนถึงใบไม้ (Leaf Node) ซึ่งให้ค่าทำนายเป็นค่าเฉลี่ยของตัวแปรตามภายในโหนดนั้น (Loh, 2011)



รูปที่ 2.13 หลักการทำงานของตัวแบบการถดถอยแบบต้นไม้ตัดสินใจ

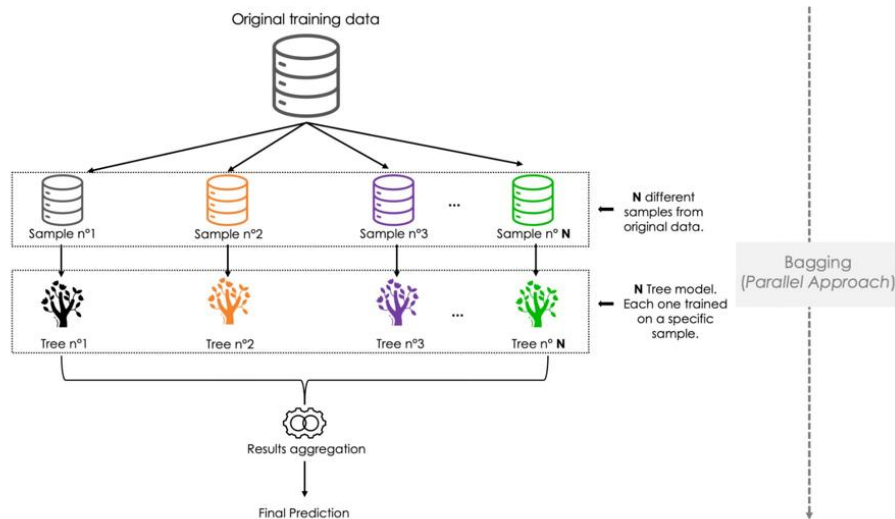
(ที่มา: [https://www.researchgate.net/figure/Example-of-a-decision-tree-regressor-We-denote-the-training-dataset-of-the-RF-algorithm\\_fig2\\_370309954](https://www.researchgate.net/figure/Example-of-a-decision-tree-regressor-We-denote-the-training-dataset-of-the-RF-algorithm_fig2_370309954))

พารามิเตอร์สำคัญที่ควบคุมพฤติกรรมของต้นไม้ตัดสินใจ เพื่อป้องกันปัญหา Overfitting หรือ Underfitting ได้แก่

- 1) ความลึกสูงสุดของต้นไม้ กำหนดเพื่อลดการขยายตัวที่มากเกินไปซึ่งนำไปสู่ความซับซ้อน
- 2) จำนวนตัวอย่างขั้นต่ำในการแบ่งโหนด กำหนดจำนวนตัวอย่าง ขั้นต่ำที่ต้องมีในโหนดก่อนที่จะสามารถแบ่งออกเป็นโหนดย่อยได้
- 3) จำนวนตัวอย่างขั้นต่ำในโหนดใบไม้ กำหนดจำนวนตัวอย่างขั้นต่ำ ที่ต้องมีในโหนดใบไม้ เพื่อให้แน่ใจว่าต้นไม้ไม่แบ่งข้อมูลมากเกินไปจนเกิดโหนดใบไม้ที่มีข้อมูลจำนวนน้อยเกินไป

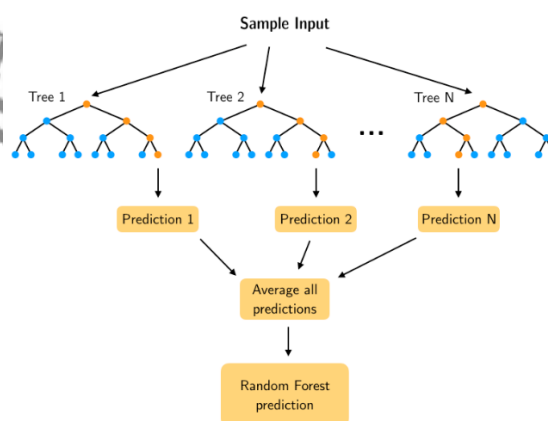
## 2.9 การถดถอยแบบป่าสุ่ม (Random Forrest Regression)

การถดถอยแบบป่าสุ่ม เป็นตัวแบบที่พัฒนาต่อยอดจากต้นไม้ตัดสินใจ โดยใช้แนวคิดการเรียนรู้แบบรวม (Ensemble Learning) เพื่อเพิ่มความแม่นยำและลดข้อผิดพลาดของการทำนายค่าตัวแปรตาม โดยเฉพาะเมื่อข้อมูลมีความซับซ้อนหรือมีมิติสูง ตัวแบบนี้ทำงานโดยการรวมผลลัพธ์จากหลายต้นไม้ตัดสินใจที่สร้างขึ้นอย่างสุ่ม เพื่อลดความเอนเอียงและความแปรปรวนของตัวแบบ (Breiman, 2001)



รูปที่ 2.14 กระบวนการ Bagging สำหรับการฝึกฝนตัวแบบด้วยแนวคิดการเรียนรู้แบบรวม  
(ที่มา: <https://www.datacamp.com/tutorial/ensemble-learning-python>)

การถดถอยแบบป่าสุ่ม ใช้เทคนิคแบ็กกิง (Bagging) ซึ่งเป็นการสร้างต้นไม้ตัดสินใจย่อยหลายต้น โดยสุ่มตัวอย่างข้อมูลจากชุดข้อมูลต้นฉบับแบบมีการคืนตัวอย่าง (Bootstrap Sampling) ทำให้แต่ละต้นไม่มีชุดข้อมูลที่แตกต่างกันเล็กน้อย เทคนิคนี้ช่วยลดความแปรปรวนของโมเดลและเพิ่มความแม่นยำในการทำนาย อีกทั้ง ในแต่ละต้นไม้ตัดสินใจย่อย ระบบจะทำการเลือกคุณลักษณะแบบสุ่มแทนที่จะพิจารณาคคุณลักษณะทั้งหมด วิธีนี้ช่วยลดความเอนเอียงของตัวแบบและทำให้ต้นไม้ตัดสินใจย่อยมีความหลากหลายมากขึ้น ส่งผลให้ตัวแบบรวมสามารถทำนายข้อมูลใหม่ได้อย่างแม่นยำยิ่งขึ้น (Breiman, 2001; Ho, 1995; นวลวรรณ สุนทรพิชัย, 2563).



รูปที่ 2.15 หลักการทำงานของตัวแบบการถดถอยแบบป่าสุ่ม

(ที่มา: [https://lwerra.github.io/dslectures/lesson06\\_random-forest-deep-dive.html](https://lwerra.github.io/dslectures/lesson06_random-forest-deep-dive.html))

การถดถอยแบบป่าสุ่มรวมค่าทำนายจากต้นไม้ตัดสินใจย่อยโดยการหาค่าเฉลี่ยของผลลัพธ์ที่ได้จากต้นไม้ทั้งหมด ซึ่งคำนวณได้จาก

$$f(x) = \frac{1}{T} \sum_{t=1}^T f_t(x) \quad (2.17)$$

โดยที่

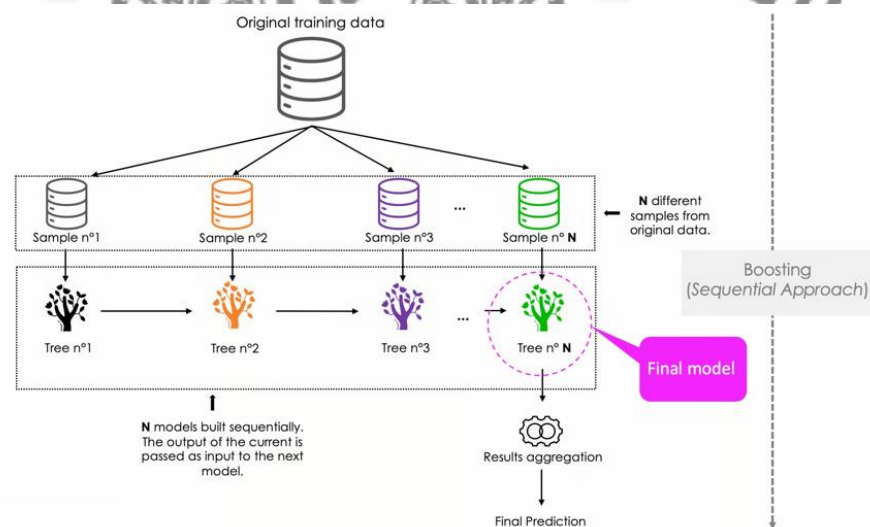
$T$  คือ จำนวนต้นไม้ตัดสินใจย่อยในตัวแบบการถดถอยแบบป่าสุ่ม

$f_t(x)$  คือ ค่าทำนายของต้นไม้ตัดสินใจที่  $t$

การถดถอยแบบป่าสุ่มเป็นตัวอย่างที่มีประสิทธิภาพสำหรับการทำนายค่าเชิงปริมาณ โดยเฉพาะในงานที่เกี่ยวข้องกับข้อมูลที่มีความไม่เป็นเชิงเส้นสูงและมีตัวแปรจำนวนมาก

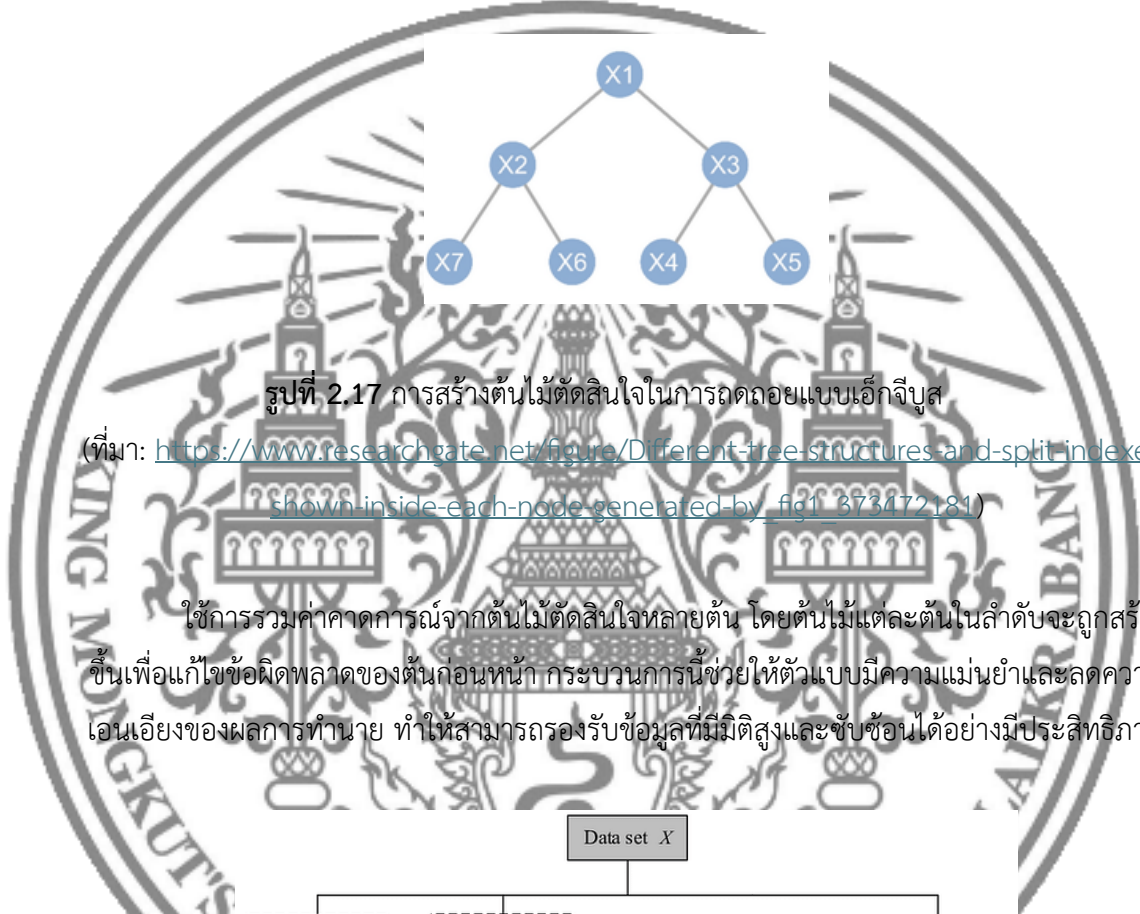
## 2.10 การถดถอยแบบเอ็กซ์ทรีม (XGBoost Regression)

การถดถอยแบบเอ็กซ์ทรีม เป็นตัวอย่างที่ได้รับการพัฒนาจากหลักการของ Gradient Boosting Decision Trees (GBDT) ซึ่งเป็นหนึ่งในเทคนิคการเรียนรู้แบบรวม โดยมุ่งเน้นการแก้ไขข้อผิดพลาดที่เกิดขึ้นจากตัวแบบก่อนหน้าในลำดับการเรียนรู้ (Chen & Guestrin, 2016) ซึ่งหลักการสำคัญของตัวแบบการถดถอยแบบเอ็กซ์ทรีม คือการเพิ่มน้ำหนัก (Weight) ให้กับข้อมูลตัวอย่างที่มีข้อผิดพลาดสูงในแต่ละรอบของการสร้างตัวแบบ และใช้การคำนวณค่าการสูญเสีย (Loss Function) ที่เหมาะสมที่สุดเพื่อปรับปรุงตัวแบบอย่างต่อเนื่อง



รูปที่ 2.16 กระบวนการ Boosting สำหรับการฝึกฝนตัวแบบด้วยด้วยแนวคิดการเรียนรู้แบบรวม (ที่มา: [https://lvwerra.github.io/dslectures/lesson06\\_random-forest-deep-dive.html](https://lvwerra.github.io/dslectures/lesson06_random-forest-deep-dive.html))

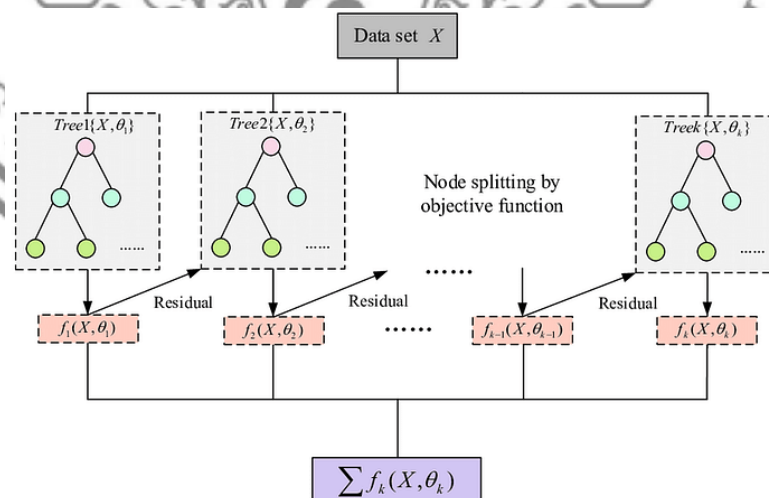
การสร้างต้นไม้ตัดสินใจใน XGBoost ใช้วิธีการแยกข้อมูล (Splitting) โดยเลือกเกณฑ์ที่เพิ่มคะแนนวัตถุประสงค์ (Objective Score) สูงที่สุด ตัวแบบพิจารณาค่าการแยก (Gain) ที่เกิดจากการแบ่งข้อมูลเพื่อหาจุดแบ่งที่เหมาะสมที่สุดในแต่ละโหนด และใช้ข้อจำกัด เช่น ความลึกสูงสุดของต้นไม้ และจำนวนโหนดขั้นต่ำที่ต้องมีข้อมูล เพื่อป้องกันการเกิดปัญหาซับซ้อนเกินไปของตัวแบบ



รูปที่ 2.17 การสร้างต้นไม้ตัดสินใจในการถดถอยแบบเอ็กจิบูส

(ที่มา: [https://www.researchgate.net/figure/Different-tree-structures-and-split-indices-shown-inside-each-node-generated-by\\_fig1\\_373472181](https://www.researchgate.net/figure/Different-tree-structures-and-split-indices-shown-inside-each-node-generated-by_fig1_373472181))

ใช้การรวมค่าคาดการณ์จากต้นไม้ตัดสินใจหลายต้น โดยต้นไม้แต่ละต้นในลำดับจะถูกสร้างขึ้นเพื่อแก้ไขข้อผิดพลาดของต้นก่อนหน้า กระบวนการนี้ช่วยให้ตัวแบบมีความแม่นยำและลดความเอนเอียงของผลการทำนาย ทำให้สามารถรองรับข้อมูลที่มีมิติสูงและซับซ้อนได้อย่างมีประสิทธิภาพ



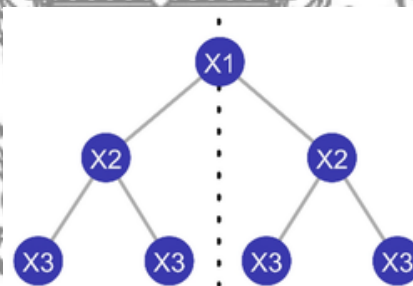
รูปที่ 2.18 หลักการทำงานของตัวแบบการถดถอยแบบเอ็กจิบูส

(ที่มา: <https://medium.com/@fraidoonomarzai99/xgboost-regression-in-depth-cb2b3f623281>)

## 2.11 การถดถอยแบบแคทบูสต์ (CatBoost Regression)

การถดถอยแบบแคทบูสต์ (CatBoost Regression) เป็นตัวแบบการเรียนรู้แบบรวมตัว (Ensemble Learning) ที่พัฒนาขึ้นจาก Gradient Boosting Decision Trees (GBDT) โดยมีการออกแบบให้สามารถจัดการกับข้อมูลประเภทเชิงหมวดหมู่ (Categorical Data) ได้อย่างมีประสิทธิภาพโดยไม่ต้องแปลงข้อมูลเป็นค่าตัวเลขด้วยเทคนิคเช่น One-Hot Encoding ซึ่งเป็นจุดเด่นที่สำคัญของ CatBoost ช่วยลดความซับซ้อนของกระบวนการเตรียมข้อมูลและเพิ่มความแม่นยำในการทำนาย (Dorogush et al., 2018) การถดถอยแบบแคทบูสต์ ใช้แนวทางของ Gradient Boosting ในการสร้างต้นไม้ตัดสินใจเพื่อลดข้อผิดพลาดของการทำนาย ซึ่งแตกต่างจากตัวแบบอื่นด้วยการจัดการข้อมูลเชิงหมวดหมู่ผ่านกระบวนการที่เรียกว่า Ordered Boosting และ Target-Based Encoding ซึ่งช่วยลด Bias ในการแปลงค่าข้อมูลเชิงหมวดหมู่และเพิ่มความเสถียรของตัวแบบ

กระบวนการสร้างต้นไม้ตัดสินใจ ใช้การแยกข้อมูล (Splitting) แบบสมมาตร (Symmetric Tree) ซึ่งกำหนดจุดแบ่ง (Split Points) ของคุณลักษณะล่วงหน้า และต้นไม้ทั้งหมดจะใช้จุดแบ่งเดียวกันในทุกโหนด ความสมมาตรนี้ช่วยเพิ่มความเร็วในการอนุมานผลลัพธ์ (Inference) และลดความซับซ้อนในการคำนวณ



รูปที่ 2.19 การสร้างต้นไม้ตัดสินใจในการถดถอยแบบแคทบูสต์

(ที่มา: [https://www.researchgate.net/figure/Different-tree-structures-and-split-indexes-shown-inside-each-node-generated-by\\_fig1\\_373472181](https://www.researchgate.net/figure/Different-tree-structures-and-split-indexes-shown-inside-each-node-generated-by_fig1_373472181))

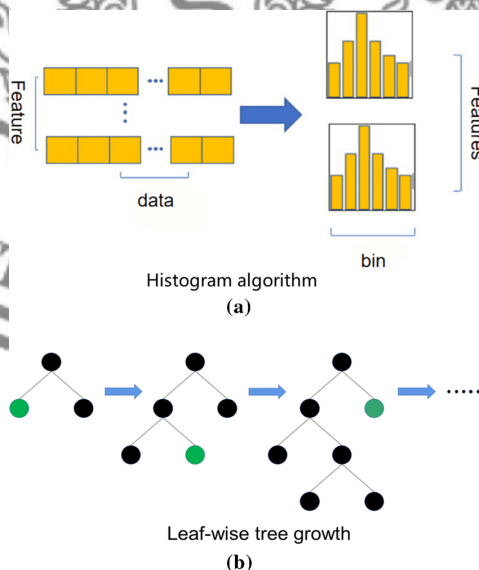
กระบวนการทำนายผลใช้ผลรวมของค่าคาดการณ์จากต้นไม้ตัดสินใจหลายต้นที่สร้างขึ้นในลำดับของ Gradient Boosting แต่ละต้นไม้จะถูกออกแบบให้ช่วยลดข้อผิดพลาดที่เหลือจากต้นไม้ก่อนหน้า ส่งผลให้ตัวแบบมีความแม่นยำสูงและสามารถจัดการกับข้อมูลที่ซับซ้อนได้

## 2.12 การถดถอยแบบไลท์จีบีเอ็ม (LightGBM Regression)

การถดถอยแบบไลท์จีบีเอ็ม (LightGBM Regression) เป็นตัวแบบการเรียนรู้แบบรวมตัว (Ensemble Learning) ที่พัฒนาขึ้นจากหลักการของ Gradient Boosting Decision Trees (GBDT) โดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพในด้านความเร็วและความสามารถในการจัดการข้อมูลขนาดใหญ่ รวมถึงข้อมูลที่มีมิติสูง (Ke et al., 2017) การถดถอยแบบไลท์จีบีเอ็ม ออกแบบมาเพื่อลดข้อจำกัดของตัวแบบ GBDT แบบดั้งเดิม ทั้งในด้านเวลาและทรัพยากรการประมวลผล

การถดถอยแบบไลท์จีบีเอ็ม ใช้แนวคิดของ Gradient Boosting ในการปรับค่าการสูญเสีย (Loss Function) อย่างต่อเนื่องในลำดับการเรียนรู้โดยใช้เทคนิคที่เพิ่มประสิทธิภาพ ได้แก่

- 1) Histogram-based Decision Tree ลดจำนวนการเปรียบเทียบในการสร้างต้นไม้ตัดสินใจ โดยแปลงค่าคุณลักษณะเป็นช่วงข้อมูลแบบฮิสโตแกรม ทำให้การคำนวณเร็วขึ้นและลดการใช้หน่วยความจำ
- 2) Leaf-wise Growth Strategy แทนที่จะเพิ่มความลึกของต้นไม้แบบระดับ (Level-wise) เหมือน GBDT แบบดั้งเดิม ตัวแบบการถดถอยแบบไลท์จีบีเอ็มจะเพิ่มโหนดใหม่ในสาขาที่มีค่าการสูญเสียสูงที่สุดก่อน วิธีนี้ช่วยให้ตัวแบบมีประสิทธิภาพสูงขึ้นและลดข้อผิดพลาดได้ดีขึ้น
- 3) Exclusive Feature Bundling (EFB) ลดขนาดของชุดข้อมูลโดยรวมคุณลักษณะที่ไม่สัมพันธ์กันเข้าด้วยกัน เพื่อลดเวลาในการประมวลผลโดยไม่สูญเสียข้อมูลสำคัญ



รูปที่ 2.20 กระบวนการ Histogram-based Decision Tree และ Leaf-wise Growth Strategy

(ที่มา: [https://www.researchgate.net/figure/Schematic-illustration-of-the-histogram-based-algorithm-and-the-model-training-strategy\\_fig1\\_361130234](https://www.researchgate.net/figure/Schematic-illustration-of-the-histogram-based-algorithm-and-the-model-training-strategy_fig1_361130234))

การทำนายผลโดยรวมค่าคาดการณ์จากต้นไม้ตัดสินใจหลายต้นที่ถูกสร้างขึ้นในลำดับของ Gradient Boosting แต่ละต้นไม้จะถูกออกแบบมาเพื่อลดข้อผิดพลาดของตัวแบบก่อนหน้า ส่งผลให้ตัวแบบสามารถเรียนรู้ข้อมูลได้อย่างรวดเร็วและแม่นยำ โดยเฉพาะในกรณีที่มีข้อมูลจำนวนมากหรือมีมิติสูง

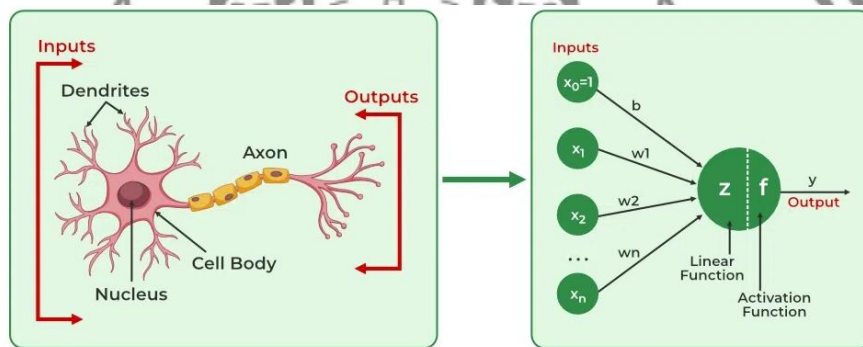


ตารางที่ 2.1 เปรียบเทียบรูปแบบการเรียนรู้แบบรวมตัว

คุณสมบัติ	การถดถอยแบบป่าสุ่ม	การถดถอยแบบเอ็กจิบูต	การถดถอยแบบแคพูสต์	การถดถอยแบบโลทีลึปีเอ็ม
ประเภทของการรวมตัว	Bagging	Boosting	Boosting	Boosting
วิธีการสร้างตัวแบบ	สุ่มตัวอย่างข้อมูลซ้ำ จากชุดข้อมูลเดิมมาใช้สร้างตัวแบบย่อย โดยแต่ละตัวแบบได้รับชุดข้อมูลที่แตกต่างกันเล็กน้อย ข้อมูลที่แตกต่างกันเล็กน้อย	สร้างตัวแบบย่อยต่อเนื่องกัน โดยโมเดลใหม่จะถูกฝึกเพื่อแก้ไขข้อผิดพลาดที่เกิดจากโมเดลก่อนหน้า	สร้างตัวแบบย่อยต่อเนื่องกัน โดยโมเดลใหม่จะถูกฝึกเพื่อแก้ไขข้อผิดพลาดที่เกิดจากโมเดลก่อนหน้า และใช้ Ordered Boosting เพื่อลด Target Leakage	สร้างตัวแบบย่อยต่อเนื่องกัน โดยโมเดลใหม่จะถูกฝึกเพื่อแก้ไขข้อผิดพลาดที่เกิดจากโมเดลก่อนหน้า และใช้ เทคนิค Leaf-wise เพื่อเพิ่มประสิทธิภาพและความเร็ว
วิธีรวมผลลัพธ์	ค่าเฉลี่ยของผลลัพธ์จากทุกตัวแบบ	ผลรวมของผลลัพธ์จากตัวแบบย่อยโดยมีน้ำหนักการเรียนรู้	ผลรวมของผลลัพธ์โดยใช้ Ordered Boosting	ผลรวมของผลลัพธ์จากตัวแบบที่เติบโตแบบ Leaf-wise
ความสามารถในการป้องกัน Overfitting	ค่อนข้างดี เนื่องจากใช้การเฉลี่ยผลลัพธ์	ใช้ Regularization (L1, L2) และ Learning Rate	ใช้ Ordered Boosting และ L2 Regularization	ใช้ L1/L2 Regularization และ GOSS
การรองรับข้อมูลเชิงหมวดหมู่	ต้องใช้ One-Hot Encoding	ต้องใช้ One-Hot Encoding หรือ Label Encoding	รองรับข้อมูลเชิงหมวดหมู่โดยตรง	ต้องใช้ One-Hot Encoding หรือ Label Encoding
โครงสร้างของต้นไม้	Level-wise Growth (เติบโตทุกโหนดพร้อมกัน)	Level-wise Growth (เติบโตทุกโหนดพร้อมกัน)	Symmetric Tree (ใช้โครงสร้างเหมือนกันในทุกโหนด)	Leaf-wise Growth (เลือกเพิ่มโหนดที่ลด Loss ได้ดีที่สุด)

## 2.13 โครงข่ายประสาทเทียม (Artificial Neural Networks - ANN)

โครงข่ายประสาทเทียม เป็นตัวแบบที่ได้รับแรงบันดาลใจจากโครงสร้างและการทำงานของ เซลล์ประสาทในสมองมนุษย์ (Biological Neural Networks) โดยมีการจัดการข้อมูลผ่านชั้นของ โหนดที่เชื่อมต่อกันในลักษณะของกราฟ โครงข่ายประสาทเทียมถูกออกแบบมาเพื่อแก้ปัญหาที่ ซับซ้อน เช่น การจำแนกรูปแบบ การทำนายข้อมูล และการวิเคราะห์ภาพเสียง โดยโครงข่ายนี้ สามารถเรียนรู้จากข้อมูลตัวอย่างและปรับปรุงน้ำหนัก (Weights) ของการเชื่อมต่อเพื่อเพิ่มความ แม่นยำในการทำนาย (Goodfellow et al., 2016).



รูปที่ 2.21 เซลล์ประสาทชีวภาพสู่เซลล์ประสาทเทียม

(ที่มา: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>)

### โครงสร้างพื้นฐานของโครงข่ายประสาทเทียม

- 1) โหนด (Node) หรือ นิวรอนเทียม (Artificial Neuron)

ทำหน้าที่เป็นหน่วยประมวลผลพื้นฐานที่รับข้อมูลเข้า ผ่านการเชื่อมต่อที่มีน้ำหนักกำกับ

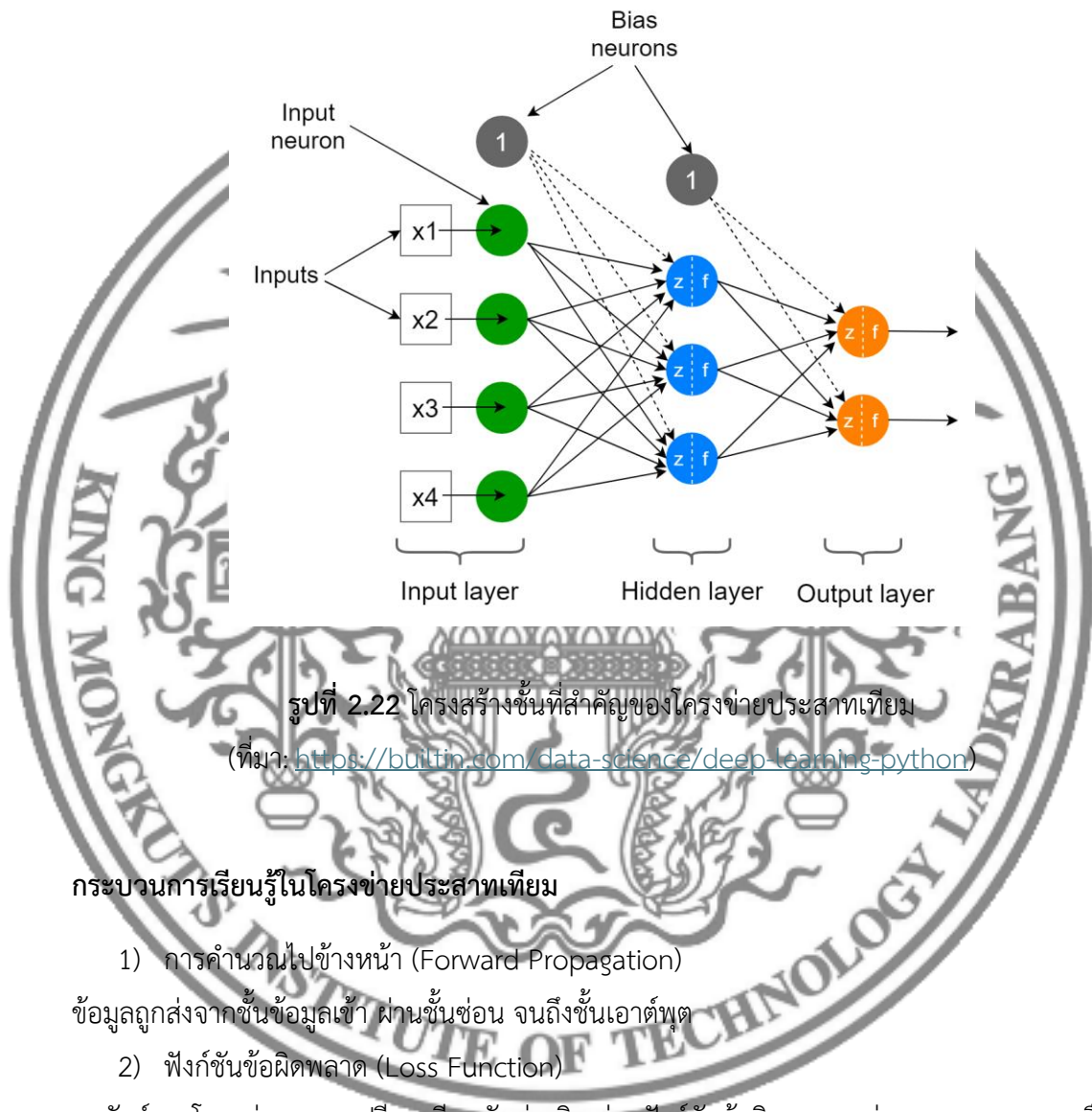
- 2) ฟังก์ชันกระตุ้น (Activation Function)

หลังจากคำนวณผลรวมแล้วผลลัพธ์จะถูกส่งผ่านฟังก์ชันกระตุ้น เพื่อกำหนดลักษณะของโหนดให้นำ ข้อมูลออกในรูปแบบที่เหมาะสม ตัวอย่างฟังก์ชันกระตุ้นที่นิยมใช้ ได้แก่ Sigmoid (0 ถึง 1), ReLU (ค่าบวกหรือศูนย์), และ Tanh (-1 ถึง 1)

- 3) โครงสร้างชั้น (Layers)

- ชั้นข้อมูลเข้า (Input Layer) รับข้อมูลจากชุดข้อมูลต้นฉบับ
- ชั้นซ่อน (Hidden Layers) เป็นชั้นที่อยู่ระหว่างชั้นข้อมูลเข้าและชั้นเอาต์พุตทำหน้าที่ประมวลผลข้อมูลด้วยฟังก์ชันกระตุ้น

- ชั้นเอาต์พุต (Output Layer) ให้ผลลัพธ์สุดท้ายของโครงข่าย เช่น การทำนายค่าตัวแปรหรือการจำแนกประเภท



รูปที่ 2.22 โครงสร้างชั้นที่สำคัญของโครงข่ายประสาทเทียม

(ที่มา: <https://builtin.com/data-science/deep-learning-python>)

### กระบวนการเรียนรู้ในโครงข่ายประสาทเทียม

- 1) การคำนวณไปข้างหน้า (Forward Propagation)

ข้อมูลถูกส่งจากชั้นข้อมูลเข้า ผ่านชั้นซ่อน จนถึงชั้นเอาต์พุต

- 2) ฟังก์ชันข้อผิดพลาด (Loss Function)

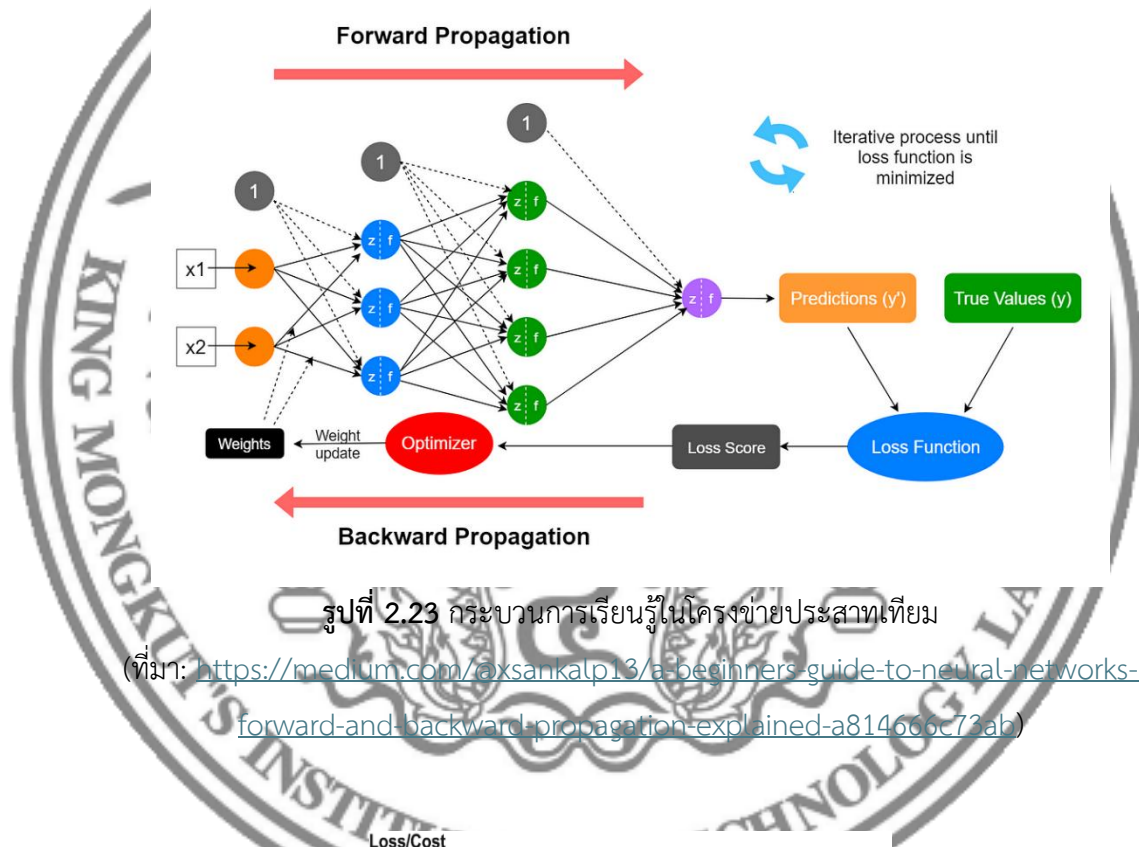
ผลลัพธ์ของโครงข่ายจะถูกเปรียบเทียบกับค่าจริง ผ่านฟังก์ชันข้อผิดพลาด เช่น ความคลาดเคลื่อนเฉลี่ยกำลังสอง สำหรับการทำนาย หรือ วิเอนโทรปีไขว้ สำหรับงานจำแนกประเภท

- 3) การย้อนกลับเพื่อลดข้อผิดพลาด (Backpropagation)

กระบวนการย้อนกลับจะคำนวณเกรเดียนต์ของฟังก์ชันข้อผิดพลาดเพื่อนำไปปรับน้ำหนักและค่าอคติในโครงข่าย โดยใช้ขั้นตอนวิธีหาค่าเหมาะสมที่สุด เช่น

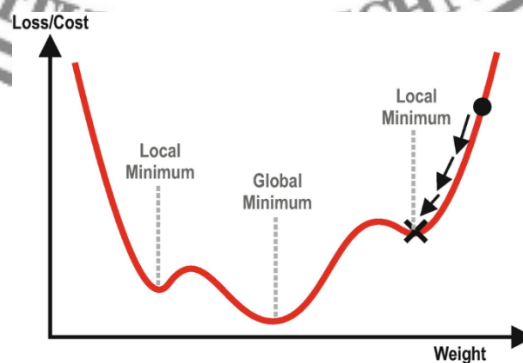
- Stochastic Gradient Descent (SGD) เป็นอัลกอริทึมพื้นฐานที่ปรับปรุงพารามิเตอร์ของโมเดลโดยใช้เกรเดียนต์ที่คำนวณจากตัวอย่างข้อมูลหนึ่งตัวอย่างหรือชุดข้อมูลย่อย (Mini-Batch) แทนการคำนวณจากข้อมูลทั้งหมด

- Momentum เพิ่มฟังก์ชันโมเมนตัมใน SGD เพื่อช่วยให้การปรับค่าพารามิเตอร์มีความราบรื่นมากขึ้น โดยลดการสั่นของเกรเดียนต์ในทิศทางที่ไม่จำเป็น
- AdaGrad ปรับอัตราการเรียนรู้ (Learning Rate) โดยอัตโนมัติสำหรับแต่ละพารามิเตอร์
- RMSprop พัฒนาต่อจาก AdaGrad โดยใช้ค่าเฉลี่ยถ่วงน้ำหนักของเกรเดียนต์กำลังสอง เพื่อให้การปรับค่าเหมาะสมกับข้อมูลขนาดใหญ่
- Adam ผสมผสาน Momentum และ RMSprop ให้มีประสิทธิภาพและความเร็วสูง



รูปที่ 2.23 กระบวนการเรียนรู้ในโครงข่ายประสาทเทียม

(ที่มา: <https://medium.com/@xsankalp13/a-beginners-guide-to-neural-networks-forward-and-backward-propagation-explained-a81466c73ab>)

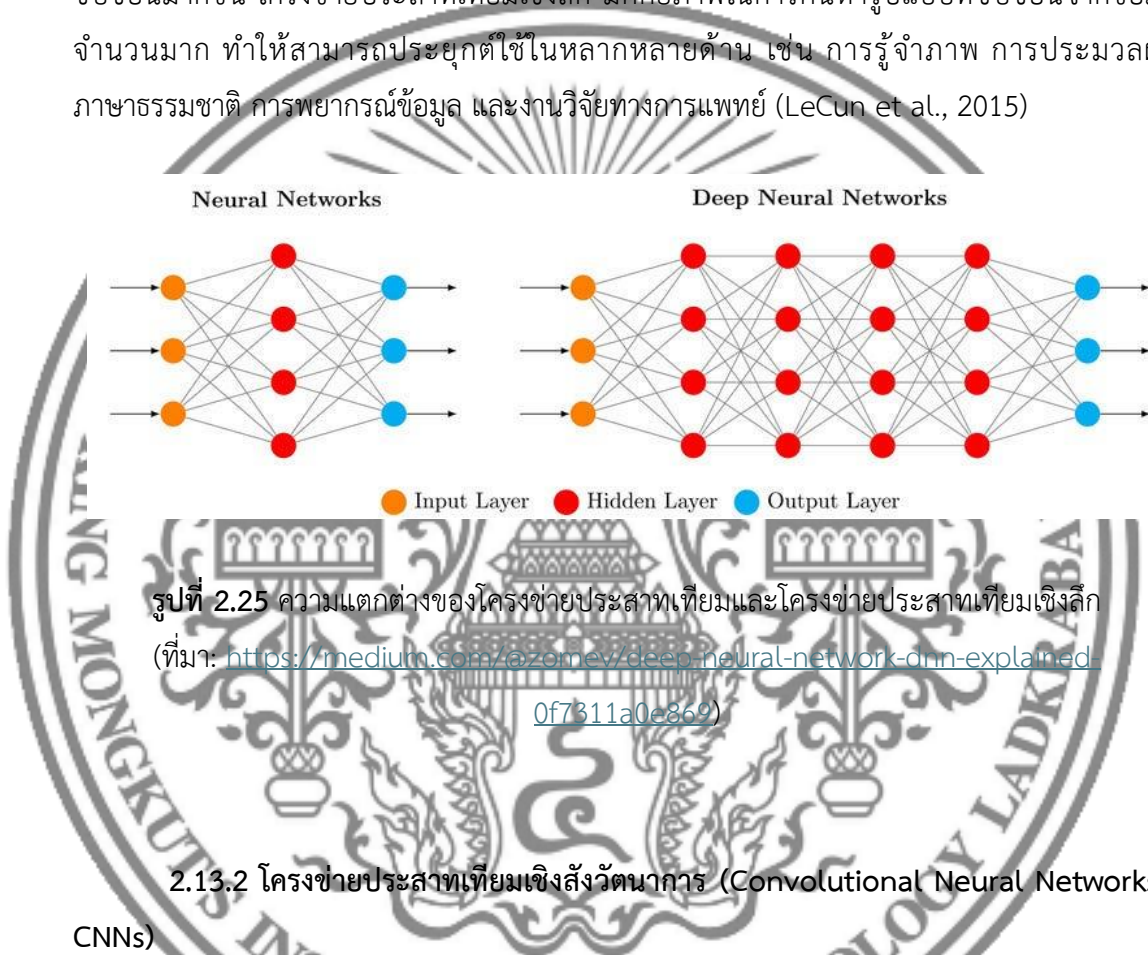


รูปที่ 2.24 กระบวนการลดข้อผิดพลาดตัวแบบโดยใช้ขั้นตอนวิธีหาค่าเหมาะสมที่สุด

(ที่มา: [https://link.springer.com/chapter/10.1007/978-1-4842-6513-0\\_3](https://link.springer.com/chapter/10.1007/978-1-4842-6513-0_3))

### 2.13.1 โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Networks - DNNs)

โครงข่ายประสาทเทียมเชิงลึก เป็นตัวแบบที่มีพื้นฐานมาจากแนวคิดของโครงข่ายประสาทเทียม โดยเพิ่มความลึกของโครงข่าย เพื่อเพิ่มความสามารถในการเรียนรู้และประมวลผลข้อมูลที่ซับซ้อนมากขึ้น โครงข่ายประสาทเทียมเชิงลึก มีศักยภาพในการค้นหารูปแบบที่ซับซ้อนจากข้อมูลจำนวนมาก ทำให้สามารถประยุกต์ใช้ในหลากหลายด้าน เช่น การรู้จำภาพ การประมวลผลภาษาธรรมชาติ การพยากรณ์ข้อมูล และงานวิจัยทางการแพทย์ (LeCun et al., 2015)

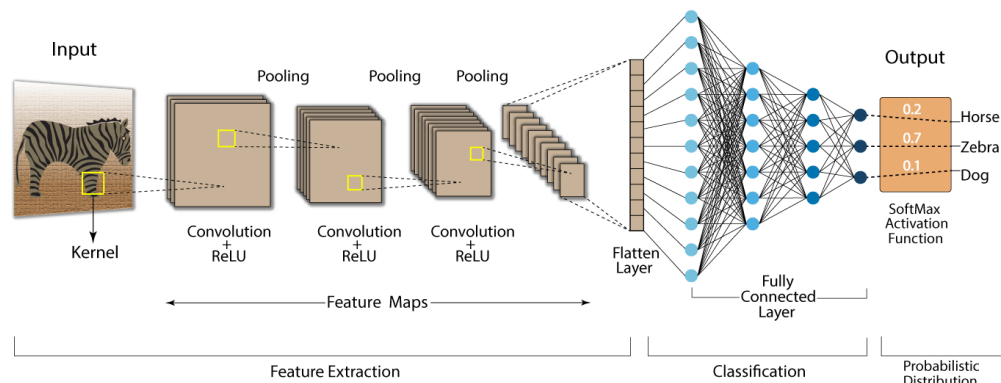


### 2.13.2 โครงข่ายประสาทเทียมเชิงสังวัตนาการ (Convolutional Neural Networks - CNNs)

โครงข่ายประสาทเทียมเชิงสังวัตนาการ เป็นโครงข่ายที่ได้รับการออกแบบมาเพื่อประมวลผลข้อมูลที่มีโครงสร้างเชิงพื้นที่ เช่น ภาพหรือวิดีโอ โดยได้รับแรงบันดาลใจจากกระบวนการมองเห็นของสมองมนุษย์ โครงสร้างของโครงข่ายประสาทเทียมเชิงสังวัตนาการใช้ชั้นคอนโวลูชัน (Convolutional Layers) เพื่อสกัดลักษณะเด่น (Feature Extraction) จากข้อมูลภาพหรือวิดีโอ โดยใช้ฟิลเตอร์ (Filters) หรือเคอร์เนล (Kernels) ในการคำนวณค่าความสัมพันธ์ของพิกเซลและลดความซับซ้อนของข้อมูล ทำให้สามารถเรียนรู้ลักษณะสำคัญได้อย่างมีประสิทธิภาพ (Krizhevsky et al., 2012) โครงสร้างของโครงข่ายประสาทเทียมเชิงสังวัตนาการประกอบด้วยองค์ประกอบหลักดังต่อไปนี้

1) ชั้นคอนโวลูชัน (Convolutional Layer) ทำหน้าที่ดึงคุณลักษณะสำคัญโดยใช้ฟิลเตอร์เลื่อนผ่านข้อมูลอินพุตเพื่อลดมิติของภาพและเน้นลักษณะที่สำคัญ เช่น ขอบและพื้นผิว

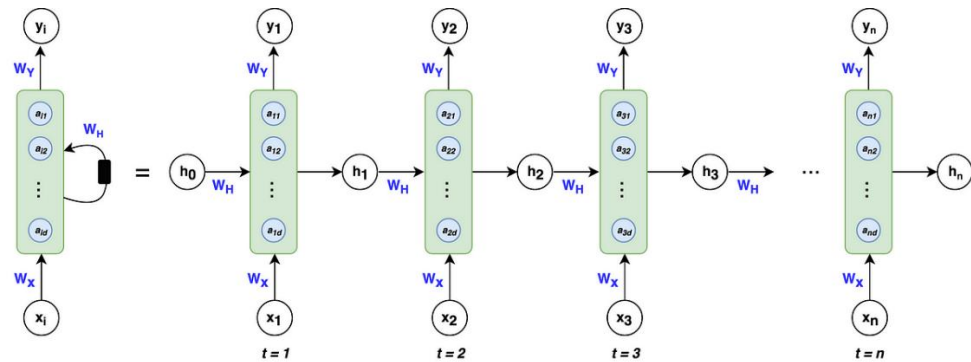
- 2) ชั้นบูรณาการสูงสุด (Pooling Layer) ลดขนาดของพีเจอร์แมพ (Feature Map) โดยเลือกค่าที่สำคัญที่สุด เช่น Max Pooling หรือ Average Pooling เพื่อลดการใช้หน่วยความจำและเพิ่มความทนทานของโมเดล
- 3) ชั้นปรับให้เป็นเวกเตอร์ (Flattening Layer) แปลงพีเจอร์แมพให้เป็นเวกเตอร์หนึ่งมิติเพื่อเตรียมข้อมูลสำหรับการจำแนกผลลัพธ์
- 4) ชั้นเชื่อมต่อทั้งหมด (Fully Connected Layer - FC Layer): นำเวกเตอร์ที่ได้จากชั้นก่อนหน้ามาผ่านโครงข่ายประสาทเทียมแบบดั้งเดิมเพื่อทำการจำแนกประเภทหรือพยากรณ์ค่าผลลัพธ์
- 5) ฟังก์ชันกระตุ้น (Activation Function): ใช้ฟังก์ชัน เช่น ReLU เพื่อเพิ่มความไม่เป็นเชิงเส้นและช่วยให้โมเดลสามารถเรียนรู้รูปแบบที่ซับซ้อนได้



รูปที่ 2.26 โครงสร้างของโครงข่ายประสาทเทียมเชิงสังวัตนาการ  
(ที่มา: <https://developersbread.com/convolution-neural-network-deep-learning/>)

### 2.13.3 โครงข่ายประสาทเทียมวนกลับ (Recurrent Neural Networks - RNNs)

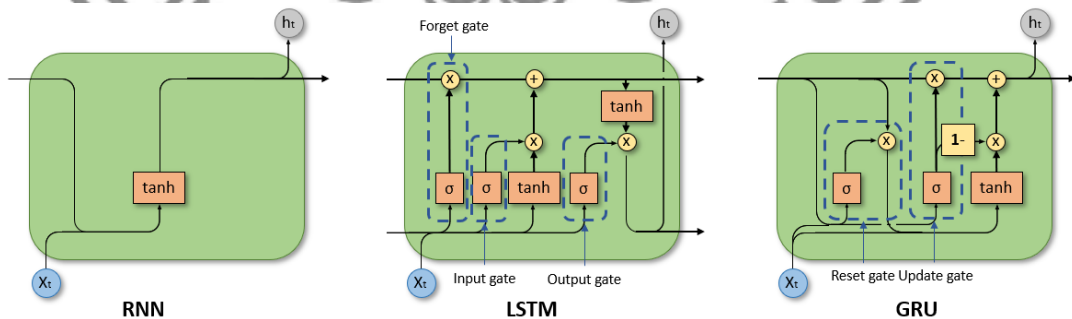
โครงข่ายประสาทเทียมวนกลับ เป็นรูปแบบของโครงข่ายประสาทเทียมที่ได้รับการออกแบบมาเพื่อจัดการกับข้อมูลตามลำดับ (Sequential Data) เช่น การประมวลผลภาษาธรรมชาติ (Natural Language Processing - NLP) และการพยากรณ์ข้อมูลเชิงเวลา (Time Series Forecasting) โดยมีความสามารถในการเก็บบริบทของข้อมูลในลำดับเวลาต่าง ๆ ผ่านโครงสร้างที่สามารถส่งค่าผลลัพธ์กลับไปเป็นอินพุตของตัวเอง โมเดลนี้ใช้ Hidden State เพื่อรักษาข้อมูลที่ได้รับมาก่อนหน้า ทำให้สามารถเรียนรู้ความสัมพันธ์ในข้อมูลที่มีความต่อเนื่องกันได้ดีกว่าโครงข่ายประสาทเทียมแบบดั้งเดิม



รูปที่ 2.27 โครงข่ายประสาทเทียมวนกลับ

(ที่มา: <https://pub.towardsai.net/whirlwind-tour-of-rnns-a11effb7808f>)

อย่างไรก็ตามโครงข่ายประสาทเทียมวนกลับ มีข้อจำกัดที่สำคัญ ได้แก่ ปัญหา Exploding Gradient และ Gradient Vanishing ซึ่งเกิดขึ้นเมื่อข้อมูลต้องส่งต่อเป็นลำดับยาว ๆ ทำให้ค่าของเกรเดียนต์อาจเพิ่มขึ้นอย่างรวดเร็วหรือหายไป ส่งผลให้โมเดลไม่สามารถเรียนรู้ข้อมูลในช่วงเวลาที่ยาวขึ้นได้ ปัญหานี้ถูกแก้ไขด้วยโครงสร้างที่ได้รับการพัฒนาเพิ่มเติม ได้แก่ Long Short-Term Memory (LSTM) และ Gated Recurrent Units (GRUs) โดย LSTM มีการเพิ่มหน่วยความจำ (Memory Cell) และกลไกควบคุมข้อมูลผ่าน Forget Gate, Input Gate และ Output Gate ซึ่งช่วยให้สามารถเรียนรู้ข้อมูลในระยะยาวได้ดีขึ้น ในขณะที่ GRUs เป็นโครงสร้างที่คล้ายคลึงกันแต่มีจำนวนน้ำหนักที่น้อยกว่า ทำให้มีประสิทธิภาพในการคำนวณที่รวดเร็วขึ้น โดยใช้เพียง Update Gate และ Reset Gate ในการควบคุมการไหลของข้อมูล (วโรดม ตูจันดา, 2023)



รูปที่ 2.28 เปรียบเทียบโครงสร้างโครงข่ายประสาทเทียมวนกลับทั้ง 3 รูปแบบ

(ที่มา: <https://aiml.com/compare-the-different-sequence-models-rnn-lstm-gru-and-transformers/>)

## 2.14 การเปรียบเทียบประสิทธิภาพการทำนายของตัวแบบ

ในงานวิจัยที่มุ่งเน้นการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ กระบวนการเปรียบเทียบประสิทธิภาพของตัวแบบการถดถอย ที่ใช้ทั้งเทคนิคการเรียนรู้ของเครื่อง และโครงข่ายประสาทเทียมเชิงลึก มีบทบาทสำคัญในการเลือกตัวแบบที่เหมาะสมที่สุดสำหรับปัญหาที่ซับซ้อนและต้องการความแม่นยำสูง การประเมินประสิทธิภาพของตัวแบบต้องอาศัยเกณฑ์ที่ครอบคลุมและมีความแม่นยำ เพื่อให้สามารถเลือกโมเดลที่มีศักยภาพสูงสุดในการทำนายข้อมูลที่ซับซ้อน (กิต์สุชาติ พสุภา, 2021)

### 2.14.1 เมทริกซ์การประเมินผล (Evaluation Matrix)

การวัดประสิทธิภาพของตัวแบบการถดถอยสามารถทำได้โดยใช้ตัวชี้วัดหลักต่อไปนี้

#### 1) ค่าสัมประสิทธิ์แสดงการตัดสินใจ (Coefficient of Determination – $R^2$ )

ค่าสัมประสิทธิ์แสดงการตัดสินใจ เป็นการวัดประสิทธิภาพของตัวแบบการถดถอย โดยจะวัดความแปรปรวนของข้อมูลระหว่างค่าจริง และค่าทำนาย ซึ่งสามารถคำนวณค่าสัมประสิทธิ์แสดงการตัดสินใจ ได้จากสมการ

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2} \quad (2.18)$$

โดยที่

- $y_i$  คือ ค่าจริง
- $\hat{y}_i$  คือ ค่าทำนายจากตัวแบบ
- $\bar{y}_i$  คือค่าเฉลี่ยของ  $y_i$
- $N$  คือจำนวนข้อมูลทั้งหมด

ซึ่งหาก  $R^2$  เข้าใกล้ 1 แสดงว่าตัวแบบสามารถอธิบายความแปรปรวนของข้อมูลได้ดี ในขณะที่ค่าที่ใกล้ 0 หมายถึงตัวแบบไม่มีความสามารถในการอธิบายข้อมูล

#### 2) ความผิดพลาดสัมบูรณ์เฉลี่ย (Mean Absolute Error - MAE)

ความผิดพลาดสัมบูรณ์เฉลี่ย เป็นการวัดความผิดพลาดโดยคำนวณจากค่าเฉลี่ยของผลรวมของค่าสัมบูรณ์ของผลต่างระหว่างค่าจริง และค่าทำนาย โดยที่ไม่สนใจทิศทางของความผิดพลาดเหล่านั้น ซึ่งสามารถคำนวณ ความผิดพลาดสัมบูรณ์เฉลี่ย ได้จากสมการ

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.19)$$

ซึ่งค่าของความผิดพลาดสัมบูรณ์เฉลี่ยจะอยู่ในช่วง  $[0, \infty)$  ค่าที่ต่ำแสดงว่าตัวแบบมีความแม่นยำสูง

3) ค่าผิดพลาดกำลังสองเฉลี่ย (Mean Square Error - MSE)

ค่าผิดพลาดกำลังสองเฉลี่ย เป็นการวัดความผิดพลาดกำลังสองเฉลี่ยที่นิยมมากที่สุด โดยคำนวณจากค่าเฉลี่ยของผลรวมของผลต่างระหว่างค่าจริง และค่าทำนายกำลังสอง ซึ่งสามารถคำนวณค่าผิดพลาดกำลังสองเฉลี่ยได้จากสมการ

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.20)$$

ซึ่งค่าของความผิดพลาดสัมบูรณ์เฉลี่ยจะอยู่ในช่วง  $[0, \infty)$  โดยมีข้อสังเกต คือ หน่วยของค่าผิดพลาดกำลังสองเฉลี่ยเป็นหน่วยยกกำลังสองของข้อมูล

4) รากที่สองของความผิดพลาดกำลังสองเฉลี่ย (Root Mean Square Error - RMSE)

รากที่สองของความผิดพลาดกำลังสองเฉลี่ย เป็นการปรับหน่วยของค่าผิดพลาดกำลังสองเฉลี่ยให้มาอยู่ในมาตราส่วนเดียวกันโดยการใส่รากที่สองให้กับค่าผิดพลาดกำลังสองเฉลี่ย ซึ่งสามารถคำนวณ รากที่สองของความผิดพลาดกำลังสองเฉลี่ยได้จากสมการ

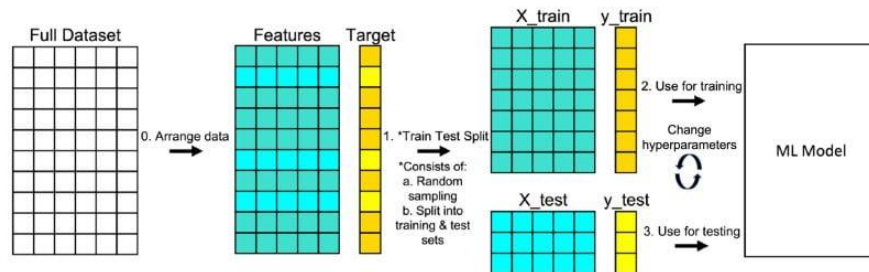
$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (2.21)$$

#### 2.14.2 การตรวจสอบความสามารถในการปรับตัวกับข้อมูลใหม่ (Generalization)

การตรวจสอบความสามารถในการปรับตัวกับข้อมูลใหม่ เป็นความสามารถของตัวแบบในการเรียนรู้จากข้อมูลฝึก (Training Data) และนำความรู้ที่ได้ไปใช้งานกับข้อมูลใหม่ (Unseen Data) อย่างมีประสิทธิภาพ ตัวแบบที่สามารถทำงานได้ดีทั้งในชุดข้อมูลฝึกและข้อมูลที่ไม่เคยเห็นมาก่อนถือว่าเป็นตัวแบบที่มี Generalization ที่ดีซึ่งการตรวจสอบ Generalization เป็นหัวใจสำคัญของการเรียนรู้ของเครื่อง และโครงข่ายประสาทเทียมเชิงลึก เนื่องจากตัวแบบที่มีปัญหาความซับซ้อนที่มากหรือน้อยเกินไป จะส่งผลต่อประสิทธิภาพการทำนายข้อมูลใหม่ (กิตส์ุชาติ พสุภา, 2021) (วโรตม ตูจินดา, 2023)

1) การแบ่งข้อมูลโดยกำหนดสัดส่วน (Train-Test Split)

เป็นวิธีพื้นฐานที่แบ่งชุดข้อมูลออกเป็น ชุดข้อมูลฝึก (Training Set) และ ชุดข้อมูลทดสอบ (Test Set) เพื่อวัดประสิทธิภาพของตัวแบบโดยใช้ข้อมูลที่ไม่เคยเห็นมาก่อน

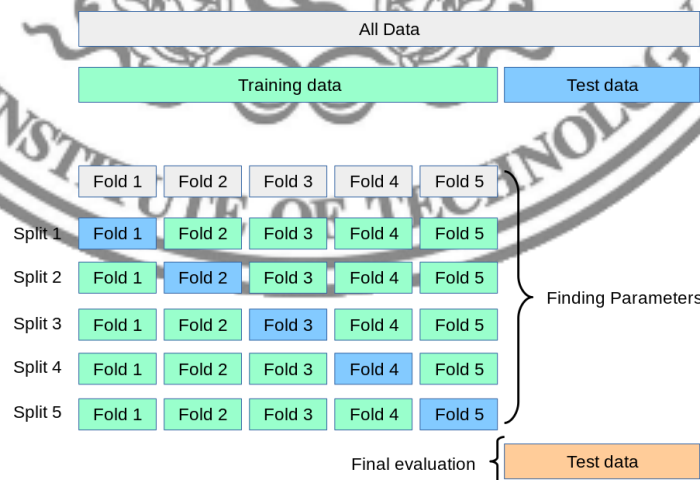


รูปที่ 2.29 การแบ่งข้อมูลเพื่อทดสอบและวัดประสิทธิภาพตัวแบบ

(ที่มา: [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html))

2) การตรวจสอบไขว้เคพับ (K-fold Cross-Validation)

เป็นเทคนิคที่แบ่งข้อมูลออกเป็น k ส่วน โดยแต่ละรอบจะใช้ k-1 ส่วนในการฝึกตัวแบบ และใช้ส่วนที่เหลือในการทดสอบ กระบวนการนี้ทำซ้ำ k ครั้งเพื่อให้ครอบคลุมทุกส่วน ซึ่งวิธีนี้จะช่วยลดผลกระทบจากการสุ่มเลือกชุดข้อมูลและให้ผลลัพธ์ที่แม่นยำมากขึ้น



รูปที่ 2.30 การค้นหาพารามิเตอร์ที่เหมาะสม

(ที่มา: [https://vitalflux.com/k-fold-cross-validation-python-example/#google\\_vignette](https://vitalflux.com/k-fold-cross-validation-python-example/#google_vignette))

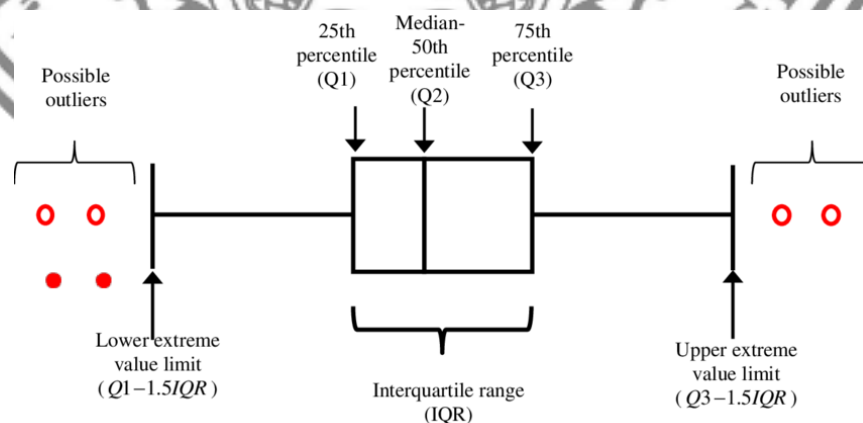
การค้นหาค่าพารามิเตอร์ที่เหมาะสมโดยใช้ RandomSearch ร่วมกับ K-fold Cross-Validation เป็นเทคนิคที่ใช้สำหรับค้นหาค่าพารามิเตอร์ที่ดีที่สุดของตัวแบบ โดยการทดลองค่าพารามิเตอร์หลายค่าและเลือกค่าที่ให้ผลลัพธ์ที่ดีที่สุด RandomSearch มักถูกใช้ร่วมกับ K-fold Cross-Validation เพื่อให้ได้พารามิเตอร์ที่มีประสิทธิภาพสูงสุดโดยหลีกเลี่ยงปัญหา Overfitting โดยมีกระบวนการทำงานดังนี้

- กำหนดค่าพารามิเตอร์ที่ต้องการทดสอบ
- ใช้ K-fold Cross-Validation แบ่งข้อมูลออกเป็น K ชุด
- ฝึกตัวแบบและทดสอบแต่ละค่าพารามิเตอร์กับทุกชุดของข้อมูล
- เลือกค่าพารามิเตอร์ที่ให้ค่าความแม่นยำสูงสุด

กระบวนการนี้ช่วยให้มั่นใจได้ว่าพารามิเตอร์ที่เลือกมีประสิทธิภาพสูงสุดในการทำนายและสามารถนำไปใช้ได้ดีในสถานการณ์จริง

## 2.15 การตรวจสอบค่าความผิดปกติ (Outlier Detection)

การตรวจสอบค่าความผิดปกติ เป็นขั้นตอนสำคัญในการประเมินความแม่นยำของตัวแบบทำนายพลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ การตรวจสอบค่าความผิดปกติจะใช้ค่าความคลาดเคลื่อน (Residual) ซึ่งนิยามเป็นผลต่างระหว่างค่าทำนายและค่าจริง ซึ่งสามารถใช้วิธี Interquartile Range (IQR) เพื่อตรวจจับค่าผิดปกติในด้านของการใช้พลังงาน ที่บ่งชี้ถึงปัญหาการทำงานของเครื่องจักรหรือความคลาดเคลื่อนที่สูงหรือต่ำกว่าปกติ



รูปที่ 2.31 การแสดงข้อมูลเชิงสถิติโดยใช้ Box Plot และการตรวจจับค่าผิดปกติด้วย IQR

(ที่มา: [https://www.researchgate.net/figure/Box-plot-with-outliers-detection-range\\_fig3\\_322129094](https://www.researchgate.net/figure/Box-plot-with-outliers-detection-range_fig3_322129094))

แนวคิดของ IQR ในการตรวจสอบค่าความคลาดเคลื่อนโดย IQR เป็นค่าที่ใช้วัดช่วงกระจายของข้อมูลที่ไม่ถูกกระทบโดยค่าผิดปกติ โดยใช้ค่าควอร์ไทล์ที่หนึ่ง (Q1 - 25%) และควอร์ไทล์ที่สาม (Q3 - 75%) ในการกำหนดช่วงของค่าปกติ และใช้เป็นเกณฑ์ในการตรวจจับค่าผิดปกติในค่าความคลาดเคลื่อนของตัวแบบตั้งสมการ

$$IQR_{res} = Q3_{res} - Q1_{res} \quad (2.22)$$

ขอบเขตค่าผิดปกติถูกกำหนดโดย

$$Lower\ Bound_{res} = Q1_{res} - (Multiplier)(IQR_{res}) \quad (2.23)$$

$$Upper\ Bound_{res} = Q3_{res} + (Multiplier)(IQR_{res}) \quad (2.24)$$

โดย Multiplier เป็น ค่าคงที่ ที่ใช้ในการกำหนดขอบเขตของค่าผิดปกติ (Outlier Thresholds) ซึ่งใช้แทนค่า 1.5 หรือ 3 ที่เป็นมาตรฐานดั้งเดิมที่กำหนดโดย John Tukey (Brett Kennedy, 2025)

## 2.16 งานวิจัยที่เกี่ยวข้อง (Related Research)

Yin, et al., (2023) ศึกษาและพัฒนาตัวแบบการทำนายพลังงานไฟฟ้าของเครื่องสูบน้ำ โดยใช้ข้อมูลจากระบบเครื่องสูบน้ำหมุนเวียนในสถานียขนาดใหญ่ที่ควบคุมด้วยระบบพีแอลซี (Programmable Logic Controller: PLC) ข้อมูลที่ใช้ในการศึกษาเป็นข้อมูลย้อนหลังตั้งแต่เดือนกันยายน 2020 ถึงเดือนมกราคม 2022 จากเครื่องสูบน้ำทั้งหมด 10 เครื่อง งานวิจัยเริ่มต้นด้วยการวิเคราะห์ข้อมูลเพื่อระบุคุณลักษณะที่มีผลต่อการใช้พลังงานไฟฟ้าของเครื่องสูบน้ำ โดยใช้วิธี Pearson Correlation Coefficient และ Single-Factor Variance Analysis เพื่อคัดกรองคุณลักษณะจากตัวแปรทั้งหมด 18 ตัวที่เกี่ยวข้องกับการใช้พลังงาน และเลือก 6 ตัวแปรที่มีความสำคัญที่สุดมาใช้เป็นอินพุตสำหรับพัฒนาตัวแบบ ในการพัฒนาตัวแบบการทำนาย ได้ใช้ตัวแบบโลทิจีบีเอ็ม ซึ่งให้ผลลัพธ์ที่มีความแม่นยำสูงที่สุด โดยได้ค่า  $R^2$  เท่ากับ 0.936 เมื่อเปรียบเทียบกับตัวแบบอื่นๆ ได้แก่ ตัวแบบเอ็กจีบูส มีค่า  $R^2$  เท่ากับ 0.897, เอ็กต้าทรี ค่า  $R^2$  เท่ากับ 0.894, ตัวแบบซัพพอร์ตเวกเตอร์แมชชีน มีค่า  $R^2$  เท่ากับ 0.873 และ โครงข่ายประสาทเทียมเชิงลึก มีค่า  $R^2$  เท่ากับ 0.69 ทั้งหมดนี้ ดำเนินการบนชุดข้อมูลที่ผ่านกระบวนการจัดการข้อมูล (Feature Engineering) อย่างเหมาะสม ผลการศึกษาของงานวิจัยแสดงถึงศักยภาพของการใช้ตัวแบบโลทิจีบีเอ็ม ในการทำนายการใช้พลังงานไฟฟ้าของเครื่องสูบน้ำ โดยเฉพาะในบริบทของระบบขนาดใหญ่ที่ต้องการความแม่นยำสูง ทั้งนี้ งานวิจัยยังสนับสนุนแนวทางการคัดกรองคุณลักษณะที่สำคัญเพื่อปรับปรุงประสิทธิภาพของตัวแบบการพยากรณ์ในงานอุตสาหกรรมที่เกี่ยวข้องกับการจัดการพลังงาน

Parraga-Alava, et al., (2024) ศึกษาการพัฒนาตัวแบบการทำนายการใช้พลังงานไฟฟ้าโดยใช้เทคนิคการเรียนรู้ของเครื่องร่วมกับตัวแบบพันธุกรรม (Genetic Algorithms: GA) โดยใช้ข้อมูลจาก Centrales Eléctricas de Nariño (CEDENAR) ในประเทศโคลอมเบีย ข้อมูลที่ใช้ในการวิจัยประกอบด้วยข้อมูลเชิงประชากรสังคมและข้อมูลการใช้พลังงานไฟฟ้าที่วัดได้ งานวิจัยนี้ได้ประยุกต์ใช้ตัวแบบการทำนาย 3 รูปแบบ ได้แก่ ตัวแบบป่าสุ่ม, เกรเดียนต์แมทซิ่ง และ ซัพพอร์ตเวกเตอร์แมชชีน เพื่อทำนายการใช้พลังงานไฟฟ้า นอกจากนี้ งานวิจัยยังได้ใช้ ตัวแบบพันธุกรรม เพื่อเพิ่มประสิทธิภาพในการปรับพารามิเตอร์ของแต่ละตัวแบบ ผลการทดลองชี้ให้เห็นว่า ตัวแบบทั้งสามแบบสามารถเพิ่มประสิทธิภาพการทำนายได้อย่างมีนัยสำคัญ ทั้งในกรณีที่ใช้ตัวแบบดั้งเดิมและเมื่อมีการประยุกต์ใช้ตัวแบบพันธุกรรม โดยเฉพาะตัวแบบป่าสุ่มที่รวมกับ ตัวแบบพันธุกรรม แสดงถึงความสามารถที่เหนือกว่าในเชิงสถิติ โดยตัวชี้วัด MAE และ MAPE มีค่าเท่ากับ 0.22 และ 60.62 ตามลำดับ และมีค่า  $R^2$  เท่ากับ 0.76

Chaerun Nisa, et al., (2021) ศึกษาและประเมินเปรียบเทียบวิธีการทำนายและการใช้พลังงานไฟฟ้าของระบบทำความเย็นแบบใช้น้ำ (Water-Cooled Chiller) โดยอาศัยตัวแบบการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก เพื่อรองรับการจัดการความต้องการพลังงานที่เพิ่มขึ้นและใช้พลังงานอย่างมีประสิทธิภาพ ข้อมูลที่ใช้ในการศึกษาได้รับการเก็บทุกๆ หนึ่งนาที่จากอาคารมหาวิทยาลัยแห่งหนึ่งในไต้หวัน โดยตัวแบบที่ใช้ในการทำนายพลังงานไฟฟ้าประกอบด้วย Thermodynamic Model และ Multi-Layer Perceptron (MLP) ในขณะที่ตัวแบบสำหรับการทำนายล่วงหน้าประกอบด้วย MLP, 1D-Convolutional Neural Network (1D-CNN) และ Long Short-Term Memory (LSTM) ซึ่งถูกนำมาเปรียบเทียบเพื่อเลือกตัวแบบที่ดีที่สุดสำหรับการนำไปใช้จริง ผลการศึกษาแสดงให้เห็นว่า MLP เป็นแบบจำลองที่ดีที่สุดในการทำนายพลังงานไฟฟ้า โดยได้ค่า  $R^2$  เท่ากับ 0.971, MAE เท่ากับ 0.743 kW และ RMSE เท่ากับ 1.157 kW ขณะที่แบบจำลองสำหรับการทำนายพลังงานไฟฟ้าล่วงหน้าแบบหนึ่งนาที่พบว่า LSTM มีประสิทธิภาพสูงสุด โดยให้ค่า  $R^2$  เท่ากับ 0.994, MAE เท่ากับ 0.233 kW และ RMSE เท่ากับ 1.415 kW การศึกษายังพบว่าแบบจำลอง LSTM สามารถลดค่าความผิดพลาดลงได้อย่างต่อเนื่องเมื่อใช้ข้อมูลฝึกอบรมเพิ่มขึ้นในแต่ละวัน อีกทั้งยังมีความแม่นยำสูงกว่าทั้ง MLP และ 1D-CNN ในการพยากรณ์พลังงานไฟฟ้าในระบบทำความเย็น นอกจากนี้ งานวิจัยยังได้เสนอแนวทางการประยุกต์ใช้โมเดลที่ดีที่สุด ได้แก่ MLP ถูกนำมาใช้เพื่อวิเคราะห์สมรรถนะของระบบและวางแผนการบำรุงรักษาเชิงป้องกัน ในขณะที่ LSTM ถูกนำมาใช้เพื่อให้การแจ้งเตือนล่วงหน้าเกี่ยวกับพฤติกรรมพลังงานที่ผิดปกติ รวมถึงการตัดสินใจเลือกแหล่งพลังงานที่เหมาะสม เช่น สลับการใช้พลังงานจากโครงข่ายไฟฟ้าไปยังแบตเตอรี่สำรองในช่วงที่พลังงานถูกคาดการณ์ว่าจะสูงสุด เพื่อลดต้นทุนพลังงานโดยรวม ผลการศึกษานี้เน้นย้ำถึง

ประสิทธิภาพของโครงข่ายประสาทเทียมเชิงลึกในการจัดการพลังงานที่ซับซ้อนและช่วยให้ระบบทำความเย็นมีการทำงานที่มีประสิทธิภาพมากขึ้น

Allal, et al., (2024). ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานในโรงงาน DAEWOO Steel Co., Ltd. ประเทศเกาหลีใต้ โดยใช้ข้อมูลการดำเนินงานรายวันที่ถูกสุ่มตัวอย่างทุกๆ 15 นาที ข้อมูลดังกล่าวได้รับการปรับปรุงคุณภาพ (Preprocessing) เพื่อให้เหมาะสมสำหรับการใช้งานกับตัวแบบการเรียนรู้ของเครื่อง งานวิจัยนี้ได้นำเทคนิคแบบจำลองการเรียนรู้ของเครื่องที่ใช้โครงสร้างต้นไม้ (Tree-Based Algorithms) จำนวน 6 รูปแบบ ได้แก่ ตัวแบบต้นไม้ตัดสินใจ, ตัวแบบป่าสุ่ม, ตัวแบบเอ็กจิบูส, ตัวแบบโลทจีบีเอ็ม, ตัวแบบแคทบูส และ เอ็กต้าทรี รวมถึงแบบจำลอง Ensemble Learners อีก 3 รูปแบบ ได้แก่ Stacking, Voting และ Bagging เพื่อนำมาพยากรณ์การใช้พลังงานใน 3 ช่วงเวลา ได้แก่ 15 นาที, 1 ชั่วโมง และ 1 วันล่วงหน้า ผลการทดลองชี้ให้เห็นว่าประสิทธิภาพของแบบจำลองแตกต่างกันไปตามช่วงเวลาที่ยพยากรณ์ โดยในช่วงเวลา 15 นาที แบบจำลอง Stacking ให้ผลลัพธ์ที่ดีที่สุดด้วยค่า  $R^2$  เท่ากับ 0.999 ในขณะที่ในช่วงเวลา 1 ชั่วโมง ตัวแบบเอ็กจิบูส แสดงผลลัพธ์ที่แม่นยำที่สุดด้วยค่า 0.997 สำหรับการทำนายในช่วงเวลา 1 วันล่วงหน้า แบบจำลอง เอ็กต้าทรี ให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า 0.993 งานวิจัยนี้เน้นย้ำถึงศักยภาพของเทคนิค Ensemble Learning และการเลือกใช้แบบจำลองที่เหมาะสมตามระยะเวลาการทำนาย เพื่อเพิ่มความแม่นยำและประสิทธิภาพในการทำนายการใช้พลังงานในอุตสาหกรรมหนักอย่างมีนัยสำคัญ

Satan, et al., (2022) ศึกษาและพัฒนาตัวแบบการทำนายพลังงานไฟฟ้าโดยประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) และการเรียนรู้เชิงลึก (Deep Learning) ในบริบทของข้อมูลแบบ Time-Series โดยแบบจำลองที่ใช้ประกอบด้วย ตัวแบบซัพพอร์ตเวกเตอร์แมชชีน และ โครงข่ายประสาทเทียมหน่วยความจำระยะสั้นและระยะยาว ซึ่งเป็นส่วนหนึ่งของโครงข่ายประสาทเทียมแบบ โครงข่ายประสาทเทียมวงกลับ การศึกษานี้เน้นการทำนายโหลดไฟฟ้าล่วงหน้าสำหรับระยะเวลาสองเดือนในเขต Aqmola ประเทศคาซัคสถาน แบบจำลองทั้งสองถูกพัฒนาและประเมินผลโดยการเปรียบเทียบค่าพยากรณ์กับค่าจริง ผลการวิเคราะห์พบว่า โครงข่ายประสาทเทียมหน่วยความจำระยะสั้นและระยะยาว ให้ผลลัพธ์ที่มีความแม่นยำสูงกว่า ตัวแบบซัพพอร์ตเวกเตอร์แมชชีน แม้ว่าความแตกต่างด้านประสิทธิภาพระหว่างสองแบบจำลองจะมีเพียงเล็กน้อย งานวิจัยสรุปว่า โครงข่ายประสาทเทียมหน่วยความจำระยะสั้นและระยะยาว และ ตัวแบบซัพพอร์ตเวกเตอร์แมชชีน มีศักยภาพในการนำไปประยุกต์ใช้สำหรับการวางแผนการดำเนินงานของโรงไฟฟ้า รวมถึงการสนับสนุนการพัฒนาาระบบพลังงานในบริบทของเมืองอัจฉริยะ (Smart City) โดยเฉพาะอย่างยิ่งในระบบที่ต้องการความแม่นยำสูง

Schummer, et al., (2024). ศึกษาและออกแบบ พัฒนา และประเมินผลของระบบตรวจจับความผิดปกติในเครือข่ายโดยใช้การเรียนรู้ของเครื่องเพื่อตรวจจับจุดผิดปกติ (point anomalies) ภายในข้อมูลการรับส่งข้อมูลเครือข่าย งานวิจัยนี้ใช้ เทคนิคการเรียนรู้แบบมีผู้สอน และการเรียนรู้แบบไม่มีผู้สอน ในการระบุความผิดปกติของเครือข่าย โดยเฉพาะการใช้ Change Point Detection, Clustering และ Classification Models เพื่อตรวจจับความผิดปกติ พร้อมทั้งใช้ SHAP Values เพื่อเพิ่มความสามารถในการตีความของโมเดล ในส่วนของ แบบจำลองการเรียนรู้แบบไม่มีผู้สอน พบว่า Change Point Detection และ K-means Clustering สามารถตรวจจับรูปแบบของข้อมูลได้ดี โดย Change Point Detection สามารถระบุช่วงเวลาที่มีการเปลี่ยนแปลงผิดปกติในปริมาณข้อมูลที่รับส่ง (Throughput) และอัตราการสูญเสียแพ็กเก็ต (Packet Loss) ในเครือข่ายได้อย่างแม่นยำ ขณะที่ K-means Clustering สามารถระบุช่วงเวลาที่มึภาวะ Congestion สูงเกิน 60% ได้อย่างมีประสิทธิภาพ ทำให้สามารถตรวจจับปัญหาความแออัดของเครือข่ายได้ล่วงหน้าสำหรับ แบบจำลองการเรียนรู้แบบมีผู้สอน พบว่าโมเดล Random Forest มีความแม่นยำสูงสุดที่ 94.3% ตามมาด้วย Support Vector Machine (SVM) ที่ให้ความแม่นยำ 96.5% แต่มีอัตราการเกิด False Positives สูงกว่าเล็กน้อย นอกจากนี้ เทคนิค Logistic Regression ยังสามารถทำนายค่าความผิดปกติได้ดีแต่มีความแม่นยำน้อยกว่าทั้งสองโมเดล งานวิจัยยังชี้ให้เห็นถึงความสำคัญของ Congestion และ Packet Loss ในการพิจารณาว่าข้อมูลใดมีแนวโน้มจะเป็น Anomaly โดยผลจากการวิเคราะห์ SHAP Values พบว่า ค่าความแออัดของเครือข่าย (Congestion) มีอิทธิพลมากที่สุดต่อการทำนายค่าผิดปกติ รองลงมาคือค่า Latency และ Packet Loss งานวิจัยนี้สามารถใช้เป็นแนวทางในการพัฒนาระบบตรวจจับความผิดปกติในเครือข่ายที่สามารถนำไปใช้งานจริง และสามารถช่วย คาดการณ์ความผิดปกติในระบบเครือข่ายก่อนที่จะเกิดเหตุการณ์เสียหายได้

ตารางที่ 2.2 เปรียบเทียบและสรุปสาระสำคัญของงานวิจัยที่เกี่ยวข้อง

ผู้วิจัย	หัวข้อการศึกษา	เทคนิคที่ใช้ในงานวิจัย	ผลลัพธ์ที่สำคัญ
Yin, et al. (2023)	ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานไฟฟ้าของเครื่องสูบน้ำขนาดใหญ่ ในอุตสาหกรรมการบำบัดน้ำเสีย	ตัวแบบการเรียนรู้ LightGBM, XGBoost, ExtraTree, SVM, DNN <u>คัดเลือกตัวแปรอิสระ</u> Pearson Correlation	ตัวแบบ LightGBM มีประสิทธิภาพสูงสุด ( $R^2 = 0.936$ ) แสดงศักยภาพของ Feature Engineering ในการทำนายการใช้พลังงานอุตสาหกรรม

ผู้วิจัย	หัวข้อการศึกษา	เทคนิคที่ใช้ในงานวิจัย	ผลลัพธ์ที่สำคัญ
Parraga-Alava, et al. (2024)	ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานไฟฟ้าของประชากรในประเทศโคลอมเบีย	ตัวแบบการเรียนรู้ Random Forest, Gradient Boosting, SVM การปรับพารามิเตอร์ Genetic Algorithm (GA)	ตัวแบบ Random Forest ร่วมกับการใช้ GA มีประสิทธิภาพสูงสุด (MAE = 0.22, MAPE = 60.62, $R^2 = 0.76$ )  การปรับแต่งตัวแบบด้วย GA ส่งผลให้ประสิทธิภาพตัวแบบเพิ่มขึ้น
Chaerun Nisa, et al. (2021)	ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานของเครื่องทำความเย็นในโรงงานอุตสาหกรรม	ตัวแบบการเรียนรู้ Thermodynamic, MLP, CNN, LSTM	ตัวแบบ MLP มีประสิทธิภาพสูงสุดในการทำนายแบบเรียลไทม์ ( $R^2 = 0.971$ )  ตัวแบบ LSTM มีประสิทธิภาพสูงสุดในการทำนายแบบล่วงหน้า 1 นาที ( $R^2 = 0.994$ )
Allal, et al. (2024)	ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานในโรงงานเหล็กประเทศเกาหลี	ตัวแบบการเรียนรู้ Ensemble Method	ตัวแบบ Stacking ( $R^2 = 0.999$ ) มีประสิทธิภาพสูงสุดในการทำนายแบบล่วงหน้า 15 นาที  ตัวแบบ XGBoost ( $R^2 = 0.997$ )

ผู้วิจัย	หัวข้อการศึกษา	เทคนิคที่ใช้ในงานวิจัย	ผลลัพธ์ที่สำคัญ
			<p>มีประสิทธิภาพสูงสุดในการทำนายแบบล่วงหน้า 1 ชั่วโมง</p> <p>ตัวแบบ ExtraTree (<math>R^2 = 0.993</math>)</p> <p>มีประสิทธิภาพสูงสุดในการทำนายแบบล่วงหน้า 1 วัน</p>
Satan, et al. (2022)	ศึกษาและพัฒนาตัวแบบการทำนายการใช้พลังงานของเมืองอัครโมลา ประเทศคาซัคสถาน	ตัวแบบการเรียนรู้ SVM, LSTM	ตัวแบบ LSTM (RMSE = 20.7%) มีประสิทธิภาพสูงกว่าตัวแบบ SVM (RMSE = 26.3%) ในการทำนายแบบล่วงหน้า 2 เดือน
Schummer, et al. (2024)	ศึกษาและพัฒนาตัวแบบการตรวจจับความผิดปกติของเครือข่ายคอมพิวเตอร์	<p>ตัวแบบการเรียนรู้แบบไม่มีผู้สอน Change Point Detection, K-means Clustering</p> <p>ตัวแบบการเรียนรู้แบบมีผู้สอน Random Forest, SVM, Logistic Regression</p>	<p>ตัวแบบการเรียนรู้แบบไม่มีผู้สอนสามารถตรวจจับความผิดปกติได้บางส่วน แต่ตีความและหาสาเหตุที่แท้จริงได้ยาก</p> <p>ตัวแบบการเรียนรู้แบบมีผู้สอน Random Forest และ SVM มีประสิทธิภาพสูงในการทำนายความผิดปกติใกล้เคียงกับค่าจริงถึง 94.3% และ 96.5% ตามลำดับ</p>

จากงานวิจัยที่เกี่ยวข้องทั้งหมดที่ได้พบทวนในบทนี้ พบว่าแนวคิด เทคนิค และกระบวนการที่ ถูกนำเสนอมีอิทธิพลสำคัญต่อการพัฒนาและออกแบบการศึกษาครั้งนี้ ทั้งในแง่ของการเลือกเทคนิคการ เรียนรู้ของเครื่องและการเรียนรู้เชิงลึก การจัดการข้อมูล การคัดเลือกคุณลักษณะ ไปจนถึงแนวทางการ ประเมินประสิทธิภาพของตัวแบบ งานวิจัยก่อนหน้าได้แสดงให้เห็นถึงความสามารถของตัวแบบต่าง ๆ ใน การจัดการกับข้อมูล และการทำนายการใช้พลังงาน นอกจากนี้ แนวคิดเกี่ยวกับการประยุกต์ใช้เทคนิค การตรวจจับค่าความผิดปกติจากข้อมูลพฤติกรรมของเครื่องจักรในเชิงอุตสาหกรรม ยังได้ให้มุมมองใหม่ที่ นำไปสู่การเสริมฟังก์ชันการตรวจจับพฤติกรรมที่ผิดปกติของเครื่องสูบน้ำป้อนหม้อไอน้ำผ่านการใช้วิธีการ ทางสถิติเชิงปริมาณ เทคนิคการคัดเลือกตัวแปรที่มีอิทธิพลจากงานวิจัยต่าง ๆ ทำให้สามารถนำมาปรับใช้ ในการวิเคราะห์ความสัมพันธ์ระหว่างตัวแปรที่มีอยู่ในชุดข้อมูลจริงของโรงไฟฟ้า ซึ่งมีหลายมิติ เช่น แรงดันน้ำออก อุณหภูมิที่จุดต่างๆ ของเครื่องจักร กำลังไฟฟ้า เป็นต้น ทำให้ได้ตัวแบบที่มีความแม่นยำ และสามารถใช้งานได้จริง โดดงจากการสังเคราะห์องค์ความรู้จากงานวิจัยที่เกี่ยวข้อง จึงสามารถออกแบบ การทดลอง พัฒนาตัวแบบ และประเมินผลได้อย่างเป็นระบบ อีกทั้งยังสามารถเชื่อมโยงไปสู่การ ประยุกต์ใช้งานจริงในสภาพแวดล้อมของโรงไฟฟ้า ซึ่งถือเป็นประโยชน์เชิงวิชาการและเชิงปฏิบัติอย่างยิ่ง ในการพัฒนาระบบบริหารจัดการพลังงานในยุคของการเปลี่ยนผ่านสู่โรงไฟฟ้าดิจิทัลอย่างยั่งยืน

## บทที่ 3

### วิธีการดำเนินงานวิจัย

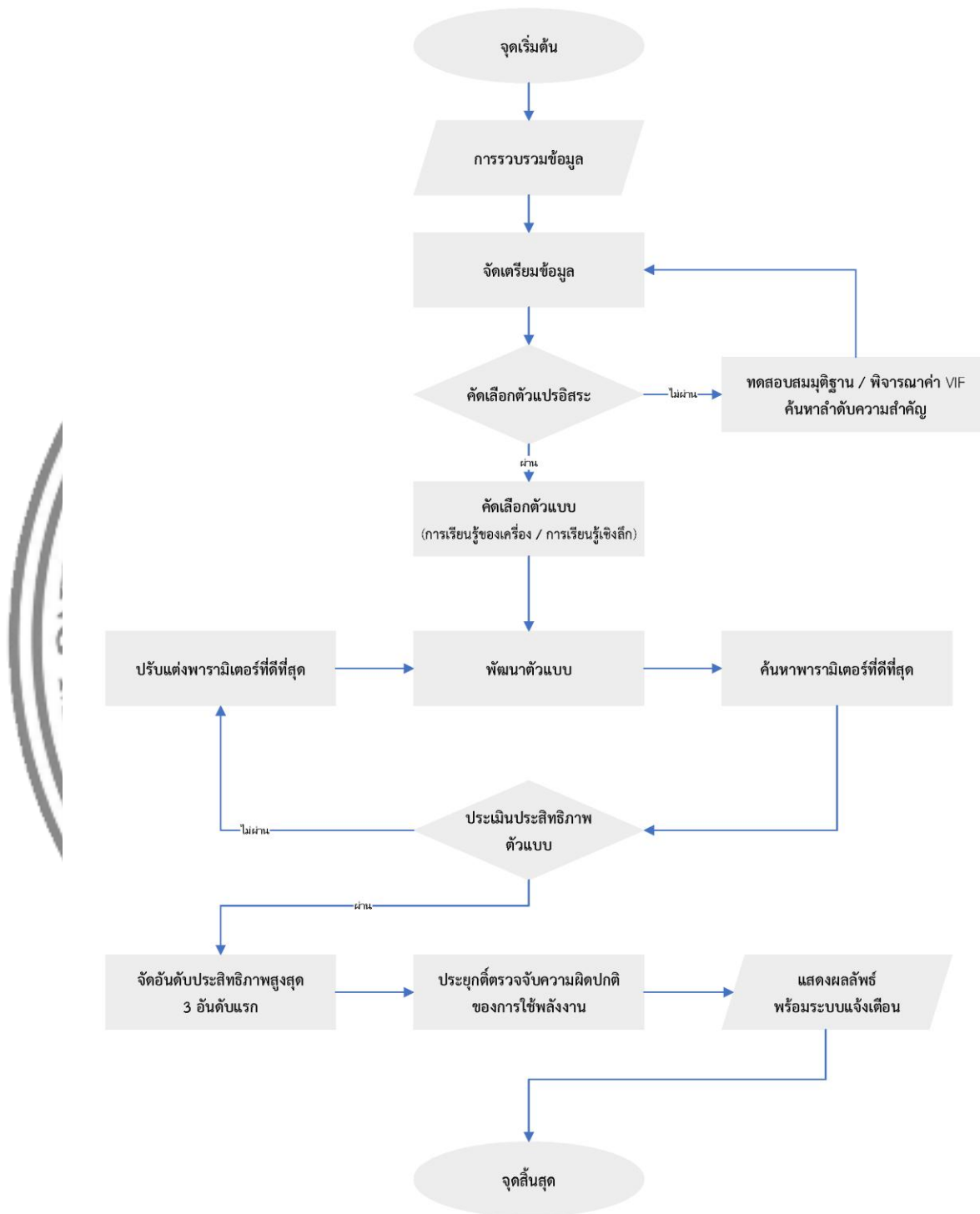
งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนา และเปรียบเทียบประสิทธิภาพของตัวแบบการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า ด้วยเทคนิคการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก พร้อมการวิเคราะห์ความผิดปกติด้านการใช้พลังงาน โดยตรวจสอบความแตกต่างระหว่างค่าพลังงานที่คำนวณได้กับค่าจริง เพื่อลดการสูญเสียพลังงานและเพิ่มประสิทธิภาพการทำงานของระบบเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า โดยผู้วิจัยได้นำทฤษฎี แนวคิด และงานวิจัยที่เกี่ยวข้องมากำหนดขั้นตอนในการศึกษาดังนี้

#### 3.1 ขั้นตอนการดำเนินงานวิจัย

การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า ด้วยเทคนิคการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก เพื่อให้เข้าใจกระบวนการวิจัย เกิดความเข้าใจคุณลักษณะ และกระบวนการจัดเตรียมข้อมูลเพื่อการสร้างตัวแบบและการวิเคราะห์ สามารถแบ่งเป็นขั้นตอนที่สำคัญได้ตามแผนภาพรูปที่ 3.1 ซึ่งแสดงขั้นตอนการดำเนินงานตั้งแต่เริ่มต้นจนถึงสิ้นสุดกระบวนการ

#### 3.2 เครื่องมือที่ใช้ในการวิจัย

งานวิจัยนี้ใช้โปรแกรมภาษาไพธอน (Python 3) เป็นเครื่องมือหลักในการพัฒนางานวิจัย โดยดำเนินการผ่านแพลตฟอร์ม Google Colab ซึ่งเป็นเครื่องมือพัฒนาแบบออนไลน์สำหรับการเขียนและรันโค้ด โปรแกรมดังกล่าวถูกนำมาใช้ในทุกขั้นตอนของการวิจัย ตั้งแต่การนำเข้าและจัดการข้อมูล การเตรียมข้อมูล การพัฒนาตัวแบบการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกสำหรับการทำนายการใช้พลังงาน การประเมินและเปรียบเทียบประสิทธิภาพของตัวแบบ ตลอดจนการประยุกต์ใช้ในการตรวจสอบความผิดปกติของเครื่องจักร ทั้งนี้ การดำเนินงานได้อาศัยไลบรารี (Library) ในแต่ละขั้นตอนที่สำคัญ ซึ่งสรุปไว้ในตารางที่ 3.1 ถึง 3.6 เพื่อสนับสนุนกระบวนการวิจัยให้มีประสิทธิภาพและครบถ้วนสมบูรณ์



รูปที่ 3.1 ขั้นตอนการดำเนินงาน

ตารางที่ 3.1 ไลบรารีพื้นฐานสำหรับการจัดการข้อมูล

ไลบรารี	คำอธิบาย
numpy	ไลบรารีสำหรับการคำนวณเชิงตัวเลข รองรับการประมวลผลอาร์เรย์หลายมิติและฟังก์ชันคำนวณเชิงตัวเลขขั้นสูง เช่น การดำเนินการเมทริกซ์และการคำนวณสถิติพื้นฐาน
pandas	ไลบรารีสำหรับการจัดการและวิเคราะห์ข้อมูลในรูปแบบของ DataFrame รองรับการดำเนินการ เช่น การจัดการค่าที่สูญหาย การแปลงข้อมูล และการสรุปค่าทางสถิติ

ตารางที่ 3.2 ไลบรารีสำหรับการแสดงผลข้อมูล

ไลบรารี	คำอธิบาย
matplotlib.pyplot	ไลบรารีสำหรับการสร้างกราฟพื้นฐาน เช่น กราฟเส้น, ฮิสโตแกรม และแผนภูมิการกระจาย ซึ่งใช้สำหรับวิเคราะห์แนวโน้มของข้อมูล
seaborn	ไลบรารีที่พัฒนาต่อยอดจาก matplotlib ช่วยให้การสร้างกราฟมีความสวยงามและแสดงผลข้อมูลทางสถิติได้ง่ายขึ้น
plotly	ไลบรารีสำหรับการสร้างกราฟที่สามารถโต้ตอบได้ รองรับการแสดงผลข้อมูลแบบไดนามิกและการสร้างกราฟที่สามารถเลื่อน ซูม และคลิกดูรายละเอียดได้

ตารางที่ 3.3 ไลบรารีการจัดการข้อมูลและการเตรียมข้อมูล

ไลบรารี	คำอธิบาย
sklearn.preprocessing.StandardScaler	ไลบรารีสำหรับการปรับค่าข้อมูลให้อยู่ในช่วงมาตรฐาน โดยทำให้ค่ามีค่าเฉลี่ยเป็น 0 และค่าเบี่ยงเบนมาตรฐานเป็น 1 เพื่อให้โมเดลสามารถเรียนรู้ได้อย่างมีประสิทธิภาพมากขึ้น

ไลบรารี	คำอธิบาย
<code>sklearn.model_selection.train_test_split</code>	ฟังก์ชันที่ใช้แบ่งชุดข้อมูลออกเป็นชุดฝึก (Training Set) และชุดทดสอบ (Testing Set) เพื่อใช้ในการประเมินผลของโมเดล
<code>sklearn.model_selection.KFold</code> , <code>cross_val_score</code>	เครื่องมือสำหรับการแบ่งข้อมูลแบบ Cross-Validation ซึ่งช่วยประเมินความสามารถของโมเดลโดยใช้ชุดข้อมูลฝึกและทดสอบในหลายรอบ เพื่อให้ได้ผลลัพธ์ที่แม่นยำยิ่งขึ้น
<code>statsmodels.stats.outliers_influence.variance_inflation_factor</code>	ฟังก์ชันที่ใช้วิเคราะห์ความสัมพันธ์ระหว่างตัวแปรอิสระ โดยคำนวณค่า VIF เพื่อลดปัญหาการมีตัวแปรที่สัมพันธ์กันมากเกินไป

#### ตารางที่ 3.4 ไลบรารีการสร้างตัวแบบการเรียนรู้

ไลบรารี	คำอธิบาย
<code>statsmodels.api.OLS</code>	ไลบรารีสำหรับพัฒนาตัวแบบการถดถอยเชิงเส้น ซึ่งสามารถวิเคราะห์ค่าความสำคัญของตัวแปร และเลือกเฉพาะตัวแปรที่มีนัยสำคัญทางสถิติ
<code>sklearn.linear_model.LinearRegression</code> , <code>Ridge</code> , <code>Lasso</code> , <code>ElasticNet</code>	ไลบรารีสำหรับพัฒนาตัวแบบการถดถอยเชิงเส้น และ ใช้สำหรับลด Overfitting ผ่านกระบวนการ Regularization
<code>sklearn.tree.DecisionTreeRegressor</code>	ไลบรารีสำหรับพัฒนาตัวแบบการถดถอยโดยใช้โครงสร้างต้นไม้ตัดสินใจ ซึ่งสามารถจัดการกับข้อมูลที่มีความซับซ้อนได้ดี
<code>sklearn.svm.SVR</code>	ไลบรารีที่ใช้พัฒนาตัวแบบการถดถอยโดยอาศัยหลักการของซัพพอร์ตเวกเตอร์แมชชีน
<code>sklearn.ensemble.RandomForestRegressor</code>	ไลบรารีที่ใช้พัฒนาตัวแบบถดถอยแบบป่าสุ่ม ที่ใช้หลายต้นไม้ตัดสินใจมาทำงานร่วมกันเพื่อเพิ่มความแม่นยำของการทำนาย
<code>xgboost.XGBRegressor</code>	ไลบรารีสำหรับพัฒนาตัวแบบการถดถอยโดยใช้วิธีเอ็กซ์จีบูส ซึ่งเป็นตัวแบบที่มีประสิทธิภาพสูง

ไลบรารี	คำอธิบาย
	และสามารถปรับแต่งพารามิเตอร์ได้อย่างยืดหยุ่น
catboost.CatBoostRegressor	ไลบรารีที่ใช้พัฒนาตัวแบบการถดถอยโดยใช้แคตบูสซึ่งเหมาะสำหรับข้อมูลที่มีค่าหมวดหมู่
lightgbm.LGBMRegressor	ไลบรารีสำหรับพัฒนาตัวแบบการถดถอยแบบ LightGBM ซึ่งเป็นตัวแบบที่สามารถเรียนรู้ได้รวดเร็วและมีประสิทธิภาพสูง
tensorflow.keras.models.Sequential	ไลบรารีสำหรับพัฒนาตัวแบบโครงข่ายประสาทเทียม แบบลำดับ
tensorflow.keras.layers.Dense	ไลบรารีสำหรับสร้างและกำหนดโครงสร้างของเลเยอร์ในโมเดลโครงข่ายประสาทเทียม
tensorflow.keras.optimizers.Adam	ไลบรารีสำหรับกำหนดตัวปรับค่าถ่วงน้ำหนัก (Optimizer) ของตัวแบบโครงข่ายประสาทเทียมโดยใช้ Adam Optimizer
tensorflow.keras.callbacks.EarlyStopping	เครื่องมือที่ใช้หยุดการฝึกตัวแบบโดยอัตโนมัติหากค่าการสูญเสีย (Loss) ไม่ลดลงต่อเนื่อง เพื่อป้องกัน Overfitting
tensorflow.keras.layers.SimpleRNN, GRU, LSTM	ไลบรารีที่ใช้สร้างตัวแบบการเรียนรู้แบบลำดับ ซึ่งสามารถนำไปใช้ในการพยากรณ์ข้อมูลที่มีลำดับเวลา

ตารางที่ 3.5 ไลบรารีสำหรับการเลือกโมเดลและการปรับค่าพารามิเตอร์

ไลบรารี	คำอธิบาย
sklearn.model_selection.RandomizedSearchCV	ไลบรารีที่ใช้สำหรับการค้นหาค่าพารามิเตอร์ที่เหมาะสมของตัวแบบโดยการสุ่มทดลองค่าต่างๆ และเลือกค่าที่ให้ผลลัพธ์ที่ดีที่สุด
sklearn.metrics.mean_squared_error, r2_score, make_scorer	ไลบรารีที่ใช้สำหรับการประเมินประสิทธิภาพของตัวแบบผ่านค่าความผิดพลาด (MSE) และค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ Score)

ไลบรารี	คำอธิบาย
sklearn.inspection.permutation_importance, model.feature_importances_, model.coef_	เครื่องมือสำหรับวิเคราะห์ความสำคัญของตัวแปรอิสระ เพื่อให้สามารถเลือกตัวแปรที่ส่งผลต่อโมเดลมากที่สุด

ตารางที่ 3.6 ไลบรารีสำหรับการแจ้งเตือนผ่านอีเมล

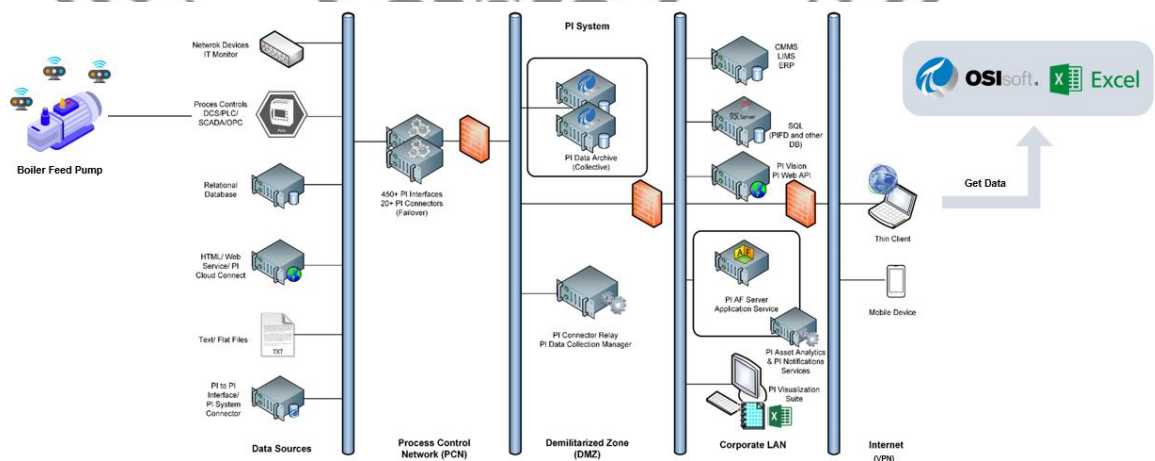
ไลบรารี	คำอธิบาย
yagmail	ไลบรารีที่ใช้สำหรับส่งอีเมลแจ้งเตือนอัตโนมัติเมื่อพบความผิดปกติในข้อมูล
logging	ไลบรารีที่ใช้สำหรับบันทึกข้อมูล Log ของระบบเพื่อใช้ในการติดตามการทำงานของโปรแกรม
datetime	ไลบรารีที่ใช้สำหรับจัดการข้อมูลเวลาและวันที่เพื่อกำหนด Timestamp ให้กับข้อมูลที่ถูกบันทึก

### 3.3 การรวบรวมข้อมูล

ข้อมูลที่ใช้ในงานวิจัย รวบรวมจากเซ็นเซอร์ที่ติดตั้งในระบบเครื่องสูบน้ำป้อนหม้อไอน้ำ โดยทำการรวบรวมข้อมูลระหว่างวันที่ 1 มกราคม 2563 ถึง 30 พฤศจิกายน 2567 ซึ่งเป็นช่วงเวลาที่ครอบคลุมการเปลี่ยนแปลงและการทำงานในสภาวะต่าง ๆ ของระบบ การดึงข้อมูลดังกล่าวดำเนินการผ่าน ระบบสารสนเทศโรงไฟฟ้า (Plant Information System) ดังรูปที่ 3.2 โดยทำการบันทึกค่าจากเซ็นเซอร์ในรูปแบบรายชั่วโมงลงบนโปรแกรมซอฟต์แวร์สเปรดชีต (Excel) เพื่อใช้ในการวิเคราะห์ ชุดข้อมูลที่รวบรวมประกอบด้วยตัวแปรที่เกี่ยวข้องกับการทำงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ ซึ่งสามารถจำแนกออกเป็น กลุ่มข้อมูลหลัก ได้แก่

1. ข้อมูลเชิงเวลาการทำงานของระบบ วัน/เวลา ที่ทำการบันทึกข้อมูล
2. ตัวแปรสภาพการทำงาน of เครื่องสูบน้ำ
  - อุณหภูมิตลับลูกปืน (Bearing Temperature)
  - ค่าสั่นสะเทือนของตัวเรือนตลับลูกปืน (Bearing Housing Vibration)
  - อุณหภูมิขดลวดมอเตอร์ (Motor Winding Temperature)
  - แรงดันน้ำออกจากเครื่องสูบน้ำ (Pump Discharge Pressure)
3. ตัวแปรทางไฟฟ้า
  - กำลังไฟฟ้าที่ผลิตจากกังหันก๊าซ (Gas Turbine Generated Power)
  - กำลังไฟฟ้าที่ใช้กับเครื่องสูบน้ำ (Pump Power Consumption)

ข้อมูลทั้งหมดมีจำนวน 43,082 ตัวอย่าง และประกอบด้วย 18 คุณลักษณะ (Features) ตามที่แสดงในตารางที่ 3.7



รูปที่ 3.2 โครงสร้างระบบสารสนเทศโรงไฟฟ้า

	A	B	C	D	E	F	G
	BFW#1 PUMP BEARING TEMP#A	BFW#1 PUMP BEARING TEMP#B	BFW#1 PUMP BEARING TEMP#C	BFW#1 PUMP BEARING TEMP#D	BFW#1 MOTOR BEARING TEMP#A	BFW#1 MOTOR BEARING TEMP#B	BFW#1 BEARING HOUSING WBR A
	138T1521A	138T1521B	138T1521C	138T1521C	138T1523A	138T1523B	138VE80TA
1							
2							
3	1/1/2020 0:00	50.13201904	54.55145264	38.16015244	52.03660965	44.57913208	2.683897495
4	1/1/2020 1:00	51.17950439	53.86006165	37.48407757	53.18946457	45.87673187	2.675510168
5	1/1/2020 2:00	49.59421921	54.50472022	39.50264359	51.69367981	44.0866127	2.582624912
6	1/1/2020 3:00	49.22756195	54.15450668	38.58868408	51.84739685	43.39414978	2.565063
7	1/1/2020 4:00	48.79058827	53.62129593	38.09224701	50.2047638	43.27204514	2.645661831
8	1/1/2020 5:00	48.12787247	53.94266129	38.21603012	50.08148193	43.16604233	2.598665118
9	1/1/2020 6:00	48.4501915	53.44613266	36.82475281	50.18661499	41.73890305	2.615232229
10	1/1/2020 7:00	48.20558929	52.8948288	36.48192215	50.50465012	42.3819313	2.53438746
11	1/1/2020 8:00	48.65281296	53.48517227	37.15797806	49.63615799	42.9544487	2.575379848
12	1/1/2020 9:00	50.36894989	52.00824356	35.73657227	51.2504986	45.32843018	2.806415081
13	1/1/2020 10:00	50.8406868	52.64807129	35.86305618	54.08932877	45.92163849	2.665086269
14	1/1/2020 11:00	52.48660278	52.7441597	37.45138768	55.9253997	46.7820015	2.732183456
15	1/1/2020 12:00	54.37211227	54.6747798	38.73297501	57.19260406	48.87995911	2.636805534
16	1/1/2020 13:00	53.61408997	56.15420151	40.4769783	57.30882263	50.25481415	2.725678921
17	1/1/2020 14:00	55.04911804	55.1149979	39.86347961	57.58953094	49.5801239	2.716920376
18	1/1/2020 15:00	53.45655823	53.60116577	38.74598312	55.97935867	49.85671234	2.881093502
19	1/1/2020 16:00	53.33686066	53.20992279	37.44024658	55.3524152	49.87956238	2.748620987
20	1/1/2020 17:00	51.82390213	54.1875	39.6562084	55.99390793	48.15373611	2.935795069
21	1/1/2020 18:00	50.99852371	55.12508392	40.77813339	53.4844017	46.3107338	2.77994895
22	1/1/2020 19:00	49.96381378	55.18648148	41.16599274	52.18086243	44.94096756	2.88267827
23	1/1/2020 20:00	49.8683167	55.02170563	41.13214874	52.06353378	44.82642365	2.817611694
24	1/1/2020 21:00	49.09202576	54.50663757	39.95148087	52.35685349	44.20889664	2.852214575
25	1/1/2020 22:00	49.34610367	54.57640457	39.14915848	51.80663681	43.87517166	2.854225636
26	1/1/2020 23:00	48.9832047	54.18865885	39.35630798	50.68884995	43.34793854	2.825780153
27	1/2/2020 0:00	49.04143906	54.04322433	37.9829102	51.83561325	44.54764175	2.796915054
28	1/2/2020 1:00	49.25139236	53.77254486	37.22902679	52.07160187	44.40714645	2.831744194
29	1/2/2020 2:00	50.13380814	53.19144821	35.94346237	52.78985214	44.59159088	2.768942356
30	1/2/2020 3:00	48.3522644	53.65563965	37.69888687	50.57894897	43.02723694	2.794644117
31	1/2/2020 4:00	48.54751968	53.37410736	37.85104752	42.74759293	42.27008057	2.718019247
32	1/2/2020 5:00	47.20767593	52.90181732	36.79303741	49.11919785	42.27008057	2.837403774
33	1/2/2020 6:00	47.88986639	52.92481854	37.92106247	48.76794434	42.03514862	2.743489265
34	1/2/2020 7:00	47.25772476	52.78769684	37.35199738	48.1656456	40.62043381	2.749300241
35	1/2/2020 8:00	47.52135058	52.71245575	37.01592636	48.50503922	41.94249859	2.842280865
36	1/2/2020 9:00	48.22739029	54.21641159	39.0350232	49.96200943	42.50370407	2.912952423
37	1/2/2020 10:00	50.35029221	53.25056458	36.03531647	52.19358444	45.37346268	2.842280865
38	1/2/2020 11:00	52.8153268	53.84997177	38.68569665	55.0472641	47.5982666	2.945138454
39	1/2/2020 12:00	54.81231308	54.7862777	39.69541931	54.94500351	48.3066864	2.952754259
40	1/2/2020 13:00	53.21168137	55.61008072	39.5638504	56.74023438	49.3313397	2.871441126
41	1/2/2020 14:00	53.52804184	53.1513443	37.39774223	55.30059433	47.94971085	2.956478834
42	1/2/2020 15:00	52.77378845	53.20604706	36.87849523	55.90745926	48.82725143	2.995647108
43	1/2/2020 16:00	52.68491364	53.84983444	37.92271423	55.86623047	48.33762741	2.902241945
44	1/2/2020 17:00	51.99967194	54.09365463	38.0817337	55.61276627	47.69701004	2.931128979

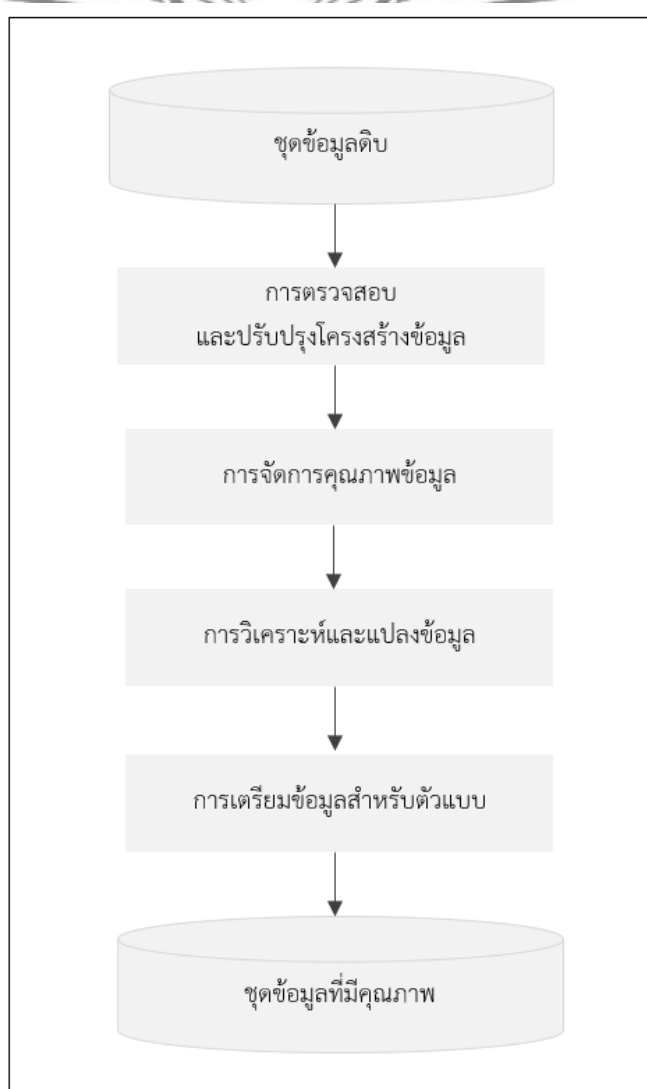
รูปที่ 3.3 ตัวอย่างชุดข้อมูลที่เป็นที่ค้ำจายเซ็นเซอร์ในรูปแบบรายชั่วโมง

ตารางที่ 3.7 คุณลักษณะของข้อมูลเครื่องสูบน้ำบ่อนหม้อไอน้ำ

ลำดับที่	ชื่อคุณลักษณะ	คำอธิบาย	ประเภทข้อมูลก่อนแปลง	ประเภทข้อมูลหลังแปลง
1	TIMESTAMP	เวลาที่บันทึกข้อมูลในรูปแบบวันและเวลา	DateTime	DateTime
2	BFW#1_PUMP_BEARING_TEMP#A	อุณหภูมิของตลับลูกปืนเครื่องสูบน้ำบ่อนหม้อไอน้ำตำแหน่ง A	Object (°C)	Float (°C)
3	BFW#1_PUMP_BEARING_TEMP#B	อุณหภูมิของตลับลูกปืนเครื่องสูบน้ำบ่อนหม้อไอน้ำตำแหน่ง B	Object (°C)	Float (°C)
4	BFW#1_PUMP_BEARING_TEMP#C	อุณหภูมิของตลับลูกปืนเครื่องสูบน้ำบ่อนหม้อไอน้ำตำแหน่ง C	Object (°C)	Float (°C)
5	BFW#1_MOTOR_BEARING_TEMP#A	อุณหภูมิของตลับลูกปืนมอเตอร์ตำแหน่ง A	object(°C)	Float (°C)
6	BFW#1_MOTOR_BEARING_TEMP#B	อุณหภูมิของตลับลูกปืนมอเตอร์ตำแหน่ง B	Object (°C)	Float (°C)
7	BFW#1_BEARING_HOUSING_VIBR_A	ค่าสั่นสะเทือนของตัวเรือนตลับลูกปืนตำแหน่ง A	Object (mm/s <sup>2</sup> )	Float (mm/s <sup>2</sup> )
8	BFW#1_BEARING_HOUSING_VIBR_B	ค่าสั่นสะเทือนของตัวเรือนตลับลูกปืนตำแหน่ง B	Object (mm/s <sup>2</sup> )	Float (mm/s <sup>2</sup> )
9	BFW#1_MOTOR_WINDING_TEMP#A	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง A	Object (°C)	Float (°C)
10	BFW#1_MOTOR_WINDING_TEMP#B	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง B	Object (°C)	Float (°C)
11	BFW#1_MOTOR_WINDING_TEMP#C	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง C	Object (°C)	Float (°C)
12	BFW#1_MOTOR_WINDING_TEMP#D	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง D	Object (°C)	Float (°C)
13	BFW#1_MOTOR_WINDING_TEMP#E	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง E	Object (°C)	Float (°C)
14	BFW#1_MOTOR_WINDING_TEMP#F	อุณหภูมิขดลวดมอเตอร์ตำแหน่ง F	Object (°C)	Float (°C)
15	BFW_PUMP_DISCHARGE_PRESSURE	แรงดันน้ำออกจากเครื่องสูบน้ำ	Object (kg/cm <sup>2</sup> )	Float (kg/cm <sup>2</sup> )
16	GTG11_POWER	กำลังไฟฟ้าที่ผลิตได้จากกังหันก๊าซเครื่องที่ 11	Object (kW)	Float (MW)
17	GTG12_POWER	กำลังไฟฟ้าที่ผลิตได้จากกังหันก๊าซเครื่องที่ 12	Object (kW)	Float (MW)
18	BFW#1_MOTOR_ACTIVE_POWER	กำลังไฟฟ้าใช้งานเครื่องสูบน้ำบ่อนหม้อไอน้ำ	Object (kW)	Float (kW)

### 3.4 การจัดเตรียมข้อมูล

การจัดเตรียมข้อมูลในงานวิจัยนี้ถูกดำเนินการด้วยกระบวนการและวิธีการที่หลากหลาย เพื่อช่วยลดข้อผิดพลาดที่อาจเกิดขึ้นจากความไม่สมบูรณ์ของข้อมูล และเพิ่มประสิทธิภาพของตัวแบบให้สามารถเรียนรู้และทำนายการผลลัพธ์ได้อย่างแม่นยำในบริบทของระบบโรงไฟฟ้าที่ต้องการความแม่นยำในการทำนายการใช้พลังงานโดยกระบวนการนี้สามารถแบ่งออกเป็น 4 ขั้นตอนหลักตามกรอบแนวคิดการจัดเตรียมข้อมูล



รูปที่ 3.4 กรอบแนวคิดการจัดเตรียมข้อมูล

### 3.4.1 การตรวจสอบและปรับปรุงโครงสร้างข้อมูล

#### 1) การโหลดและตรวจสอบข้อมูล

ข้อมูลที่ใช้ในงานวิจัยถูกนำเข้าในรูปแบบไฟล์ Excel (BFW-A.xlsx) โดยระบุ sheet\_name='Sheet1' และโหลดเข้าสู่โปรแกรมด้วยไลบรารี pandas พร้อมแสดงข้อมูล 10 แถวแรกและท้ายสุดเพื่อดูภาพรวม มีการตรวจสอบโครงสร้างของข้อมูล (df.info()) และจำนวนตัวอย่างทั้งหมด (df.shape) เพื่อให้ทราบจำนวนตัวแปรและลักษณะข้อมูลเบื้องต้น

#### 2) การเปลี่ยนชื่อและการจัดรูปแบบคุณลักษณะ

ชื่อคุณลักษณะถูกปรับให้อยู่ในรูปแบบที่ง่ายต่อการเรียกใช้ โดยแทนที่ช่องว่างด้วย \_ และเปลี่ยนเป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด คุณลักษณะ Unnamed: 0 ถูกเปลี่ยนชื่อเป็น TIMESTAMP และข้อมูลในคุณลักษณะนี้ถูกแปลงให้อยู่ในรูปแบบวันที่ (datetime) เพื่อเตรียมใช้งานในลักษณะข้อมูลอนุกรมเวลา (time series)

#### 3) การแปลงประเภทข้อมูล

ข้อมูลทั้งหมดถูกแปลงให้อยู่ในรูปแบบตัวเลข (numeric) ยกเว้นคุณลักษณะ TIMESTAMP เพื่อรองรับการคำนวณและการวิเคราะห์เชิงสถิติ

### 3.3.2 การจัดการคุณภาพข้อมูล

#### 1. การตรวจสอบค่าที่ขาดหาย (missing values)

มีตรวจสอบค่าที่ขาดไปและสรุปรายงานการขาดหายของข้อมูลในแต่ละคุณลักษณะพบว่า คุณลักษณะ BFW#1 MOTOR\_ACTIVE\_POWER ซึ่งมีค่าขาดหายถึง 7,618 ตัวอย่าง คิดเป็น 17.68% ของจำนวนตัวอย่างทั้งหมด รองลงมาเป็น GTG12\_POWER ซึ่งซึ่งมีค่าขาดหาย 254 ตัวอย่าง จากนั้นทำการลบข้อมูลที่มีค่าสูญหาย (dropna) ทั้งแถวเพื่อความสมบูรณ์ในการสร้างตัวแบบ

#### 2. การจัดการข้อมูลซ้ำซ้อน

มีการตรวจสอบข้อมูลซ้ำ (duplicate data) และลบข้อมูลที่ซ้ำซ้อน

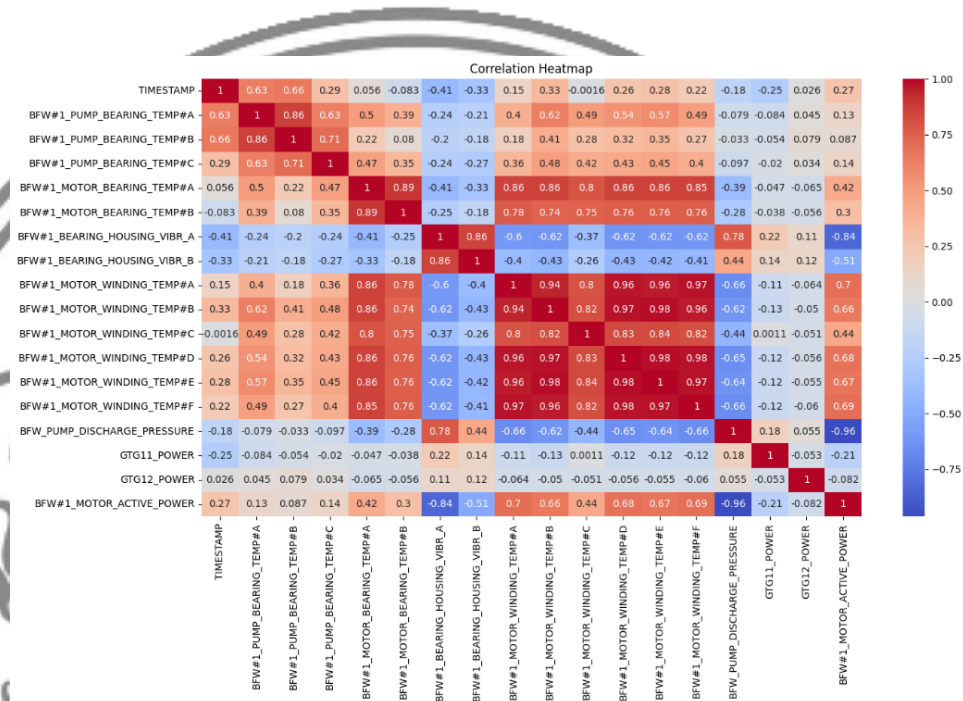
#### 3. การจัดการข้อมูลผิดปกติ (Outliers)

ใช้การตรวจสอบข้อมูลผิดปกติด้วยแผนภาพกล่อง (Box Plot) เพื่อแสดงค่าที่อยู่ห่างจากค่าปกติอย่าง โดยใช้เทคนิค IQR (Interquartile Range) เพื่อจัดการค่าผิดปกติ โดยทำการลบข้อมูลที่อยู่นอกช่วงขอบบน (Maximum) และขอบล่าง (Minimum) ออกเพื่อจัดการข้อมูลผิดปกติ

### 3.3.3 การวิเคราะห์และแปลงข้อมูล

#### 1) การวิเคราะห์เชิงสำรวจข้อมูล (Exploratory Data Analysis - EDA)

ใช้สถิติเชิงพรรณนา (df.describe()) เพื่อสรุปค่าทางสถิติ เช่น ค่าเฉลี่ย ค่าส่วนเบี่ยงเบนมาตรฐาน และค่ามาก-น้อยสุด แสดงความสัมพันธ์ระหว่างตัวแปรด้วยแผนภาพกระจาย (pairplot) และแผนภาพความสัมพันธ์เชิงเส้น (correlation heatmap) เพื่อระบุว่าตัวแปรใดมีผลต่อเป้าหมาย



รูปที่ 3.5 แผนภาพความสัมพันธ์เชิงเส้นของตัวแปร

## 2) การตรวจสอบความสัมพันธ์ระหว่างตัวแปร (Multicollinearity)

ตรวจสอบปัญหาความสัมพันธ์สูง ด้วยค่าดัชนี VIF (Variance Inflation Factor) เพื่อระบุว่าตัวแปรใดมีความสัมพันธ์ซ้ำซ้อนที่อาจส่งผลกระทบต่อวิเคราะห์ ตัวแปรที่มีค่าดัชนี VIF สูงถูกพิจารณาถึงความสำคัญหรือตัดออก

### 3.3.4 การเตรียมข้อมูลสำหรับแบบจำลอง

#### 1) การกำหนดตัวแปรเป้าหมายและตัวแปรอิสระ

กำหนด Target Variable (y) และตัวแปรอิสระ (X) สำหรับสร้างแบบจำลอง

#### 2) การปรับขนาดข้อมูล (Feature Scaling)

ใช้เทคนิค StandardScaler ในการปรับค่าข้อมูล (standardization) ให้อยู่ในช่วงเดียวกัน เพื่อให้ตัวแบบการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกสามารถทำงานได้อย่างมีประสิทธิภาพ

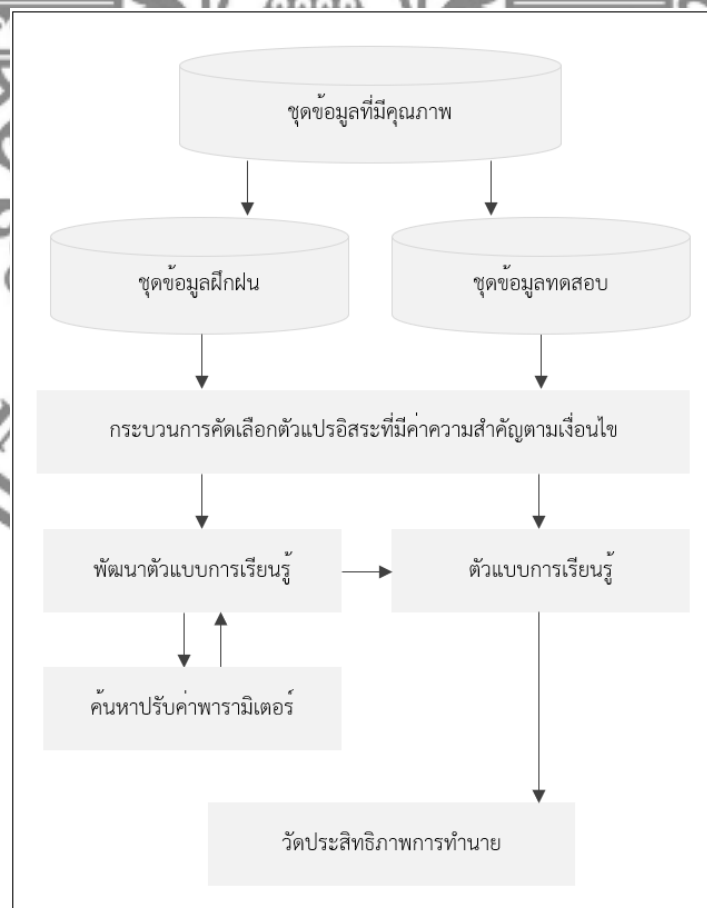
#### 3) การแบ่งชุดข้อมูล (Train-Test Split)

แบ่งข้อมูลออกเป็น ชุดฝึก (Training Set) 80% ซึ่งมีจำนวน 18,638 ตัวอย่าง และชุดทดสอบ (Test Set) 20% จำนวน 4,660 ตัวอย่าง เพื่อประเมินประสิทธิภาพของตัวแบบ

หลังจากผ่านกระบวนการจัดเตรียมข้อมูลทั้ง 4 ขั้นตอน ได้ชุดข้อมูลที่มีคุณภาพสูงขึ้น พร้อมสำหรับการพัฒนาตัวแบบ โดยข้อมูลที่เหลือมีจำนวน 23,298 ตัวอย่าง และมี 18 คุณลักษณะ (Features) ที่จะใช้ในการวิเคราะห์ต่อไป

### 3.5 การพัฒนาตัวแบบการเรียนรู้

ในการพัฒนาตัวแบบทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ งานวิจัยนี้ได้ดำเนินการพัฒนาตัวแบบโดยใช้ทั้ง การเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก เพื่อให้ได้ตัวแบบที่เหมาะสมสำหรับการทำนายค่าการใช้พลังงาน โดยมีกระบวนการพัฒนาเป็นไปตามกรอบแนวคิดที่แสดงในรูปที่ 3.5 ซึ่งสามารถแบ่งเป็น 3 ขั้นตอนหลัก ได้แก่ 1. การคัดเลือกตัวแปรอิสระ, 2. การพัฒนาตัวแบบการเรียนรู้, 3. การปรับแต่งพารามิเตอร์



รูปที่ 3.6 กรอบแนวคิดการพัฒนาตัวแบบการเรียนรู้

### 3.5.1 การคัดเลือกตัวแปรอิสระสำหรับการพัฒนาตัวแบบ

ชุดข้อมูลที่ผ่านมากระบวนการทำความสะอาดและจัดเตรียมแล้วจะถูกแบ่งออกเป็น 2 ส่วนหลัก ได้แก่ ชุดข้อมูลฝึกฝน และชุดข้อมูลทดสอบ จากนั้นชุดข้อมูลทั้ง 2 ส่วนจะถูกนำเข้าสู่กระบวนการค้นหาและคัดการคัดเลือกตัวแปรอิสระที่มีความสำคัญแต่ละเงื่อนไข ในงานวิจัยนี้ ได้ใช้วิธีการวิเคราะห์ความสำคัญของตัวแปรหลายรูปแบบ ได้แก่ Coefficient Analysis, Feature Importance จาก Tree-based Models, Permutation Importance และการวิเคราะห์ความสัมพันธ์เชิงสถิติ เพื่อให้มั่นใจว่าตัวแปรที่เลือกมีผลกระทบอย่างมีนัยสำคัญต่อค่าการทำนายพลังงาน ซึ่งเพื่อประเมินผลกระทบของตัวแปรต่อการทำนาย งานวิจัยนี้กำหนดเงื่อนไขการคัดเลือกตัวแปรอิสระสำหรับการพัฒนาตัวแบบ หลายรูปแบบ ได้แก่

- 1) ใช้ตัวแปรทั้งหมด (All Features)  
เพื่อประเมินว่าตัวแบบสามารถทำงานได้ดีเพียงใดเมื่อใช้ข้อมูลทั้งหมด
- 2) ใช้ตัวแปรที่มีค่า p-value  $\leq 0.05$  (สำหรับตัวแบบการถดถอยเชิงเส้น)  
เพื่อเป็นการเลือกตัวแปรที่มีนัยสำคัญทางสถิติในการอธิบายค่าทำนาย
- 3) ใช้ตัวแปรที่มี Variance Inflation Factor (VIF)  $\leq 10$  (สำหรับตัวแบบการถดถอยเชิงเส้น)  
เพื่อลดปัญหาความสัมพันธ์เชิงเส้นสูงระหว่างตัวแปรอิสระ (Multicollinearity)
- 4) ใช้ตัวแปรที่ได้จากการคัดเลือกด้วย Feature Importance  
โดยพิจารณาจากตัวแปรที่มีค่าความสำคัญสูงสุด 5 และ 10 (สำหรับตัวแบบกลุ่ม Tree-Based)

### 3.5.2 การพัฒนาตัวแบบการเรียนรู้ (Model Development)

ตัวแบบที่ใช้ในงานวิจัยนี้สามารถแบ่งออกเป็น 2 กลุ่มหลัก ได้แก่

- 1) ตัวแบบการเรียนรู้ของเครื่อง (Machine Learning Models) ตัวแบบที่ใช้ในการทำนายประกอบด้วย
  - การถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression - MLR)
  - การถดถอยเชิงแบบเร็กิวลาร์ไรซ์เซชัน (Regularized Regression)
  - การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Regression)
  - การถดถอยแบบต้นไม้ตัดสินใจ (Decision Tree Regression)
  - การถดถอยแบบป่าสุ่ม (Random Forest Regression)
  - การถดถอยแบบเอ็กซ์ทรีมเกรดिएนต (XGBoost Regression)
  - การถดถอยแบบแคตบูส (CatBoost Regression)
  - การถดถอยแบบไลท์จีบีเอ็ม (LightGBM Regression)

2) ตัวแบบการเรียนรู้เชิงลึก (Deep Learning Models) ตัวแบบการเรียนรู้เชิงลึกที่ใช้ในงานวิจัยนี้ได้แก่

- โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Networks - DNNs)
- โครงข่ายประสาทเทียมวงกกลับ (Recurrent Neural Networks - RNNs)
- โครงข่ายประสาทเทียมแบบ GRU (Gated Recurrent Units - GRUs)
- โครงข่ายประสาทเทียมแบบ LSTM (Long Short-Term Memory - LSTMs)

### 3.5.3 การปรับแต่งพารามิเตอร์ (Hyperparameter Tuning)

เพื่อให้ตัวแบบมีประสิทธิภาพสูงสุด งานวิจัยนี้ได้ดำเนินการปรับแต่งพารามิเตอร์ของตัวแบบแต่ละเงื่อนไขการคัดเลือกตัวแปรอิสระ โดยใช้ Randomized Search ร่วมกับ K-fold Cross-Validation เพื่อค้นหาพารามิเตอร์ที่เหมาะสมที่สุด กระบวนการปรับแต่งพารามิเตอร์ดำเนินการผ่านการสุ่มเลือกค่าพารามิเตอร์ที่เป็นไปได้ ภายในค่าที่กำหนด พร้อมใช้ K-fold Cross-Validation เพื่อประเมินประสิทธิภาพของตัวแบบแต่ละค่าพารามิเตอร์ โดยตัวแบบ การปรับแต่งพารามิเตอร์ และคำอธิบาย ดังตารางที่ 3.8

ตารางที่ 3.8 การปรับแต่งพารามิเตอร์ตัวแบบการเรียนรู้

ตัวแบบ	การปรับแต่งพารามิเตอร์	คำอธิบาย
การถดถอยเชิงเส้นพหุคูณ	ใช้ค่าเริ่มต้นของ LinearRegression()	ไม่มีพารามิเตอร์ให้ปรับแต่งโดยตรง
การถดถอยเชิงเส้นแบบลาซโซ, การถดถอยเชิงเส้นแบบบริดจ์	alpha	ค่า alpha ควบคุมระดับ Regularization เพื่อป้องกัน Overfitting
การถดถอยแบบอิลาสติกเน็ต	alpha, l1_ratio	ปรับสมดุลระหว่าง L1 และ L2 Regularization
การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน	Kernel, C, epsilon	ปรับฟังก์ชันเคอร์เนล และค่าควบคุมขอบเขตของค่าคาดการณ์
การถดถอยแบบต้นไม้ตัดสินใจ	max_depth, min_samples_split, min_samples_leaf	ควบคุมขนาดและโครงสร้างของต้นไม้ตัดสินใจ

ตัวแบบ	การปรับแต่งพารามิเตอร์	คำอธิบาย
การถดถอยแบบป่าสุ่ม	n_estimators max_depth	ปรับจำนวนต้นไม้และปรับความลึกของต้นไม้
การถดถอยแบบเอ็กจีบูส, การถดถอยแบบแคทบูส, การถดถอยแบบโลทจีบีเอ็ม	n_estimators, learning_rate	ปรับจำนวนต้นไม้ และอัตราการเรียนรู้ (learning rate)
โครงข่ายประสาทเทียมเชิงลึก (DNN, RNN, LSTM, GRU)	dense layer, learning_rate, activation function, optimizer, batch size, epochs	ปรับโครงสร้างของโครงข่ายเพื่อเพิ่มประสิทธิภาพการเรียนรู้ของตัวแบบ

### 3.6 การวัดประสิทธิภาพตัวแบบการทำนาย

เพื่อให้ได้ตัวแบบที่ดีที่สุด งานวิจัยนี้ได้ทำการเปรียบเทียบประสิทธิภาพของแต่ละตัวแบบในแต่ละเงื่อนไขการคัดเลือกตัวแปรอิสระ งานวิจัยนี้ใช้ตัวชี้วัดหลักสองวิธี ได้แก่

- 1) ค่าผิดพลาดกำลังสองเฉลี่ย (MSE) ใช้วัดค่าความคลาดเคลื่อนของตัวแบบ ยิ่งค่า MSE ต่ำ หมายถึงตัวแบบสามารถพยากรณ์ค่าที่ใกล้เคียงกับค่าจริงมากขึ้น
- 2) ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ ) ซึ่งมีบทบาทสำคัญในการประเมินความแม่นยำและความสามารถของตัวแบบการทำนาย โดยหากค่าที่เข้าใกล้ 1 หมายถึงตัวแบบสามารถอธิบายความแปรปรวนของข้อมูลได้ดีมาก

## บทที่ 4

### ผลการวิจัยและอภิปรายผล

ในบทนี้นำเสนอผลการวิจัยที่ได้จากการศึกษาการทำนายการใช้พลังงานของเครื่องสูบน้ำ ป้อนหม้อไอน้ำโดยใช้วิธีการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก รวมถึงการวิเคราะห์เปรียบเทียบ ประสิทธิภาพของตัวแบบต่าง ๆ ที่พัฒนาในงานวิจัยนี้ โดยแสดงกระบวนการเลือกตัวแปรอิสระ การปรับแต่งพารามิเตอร์ และการวัดประสิทธิภาพของตัวแบบในแต่ละขั้นตอน และการอภิปรายผล การศึกษาเพื่อแสดงถึงข้อดีและข้อจำกัดของแต่ละตัวแบบ ทั้งในด้านความแม่นยำ ความซับซ้อน และความเหมาะสมของตัวแบบในบริบทของการใช้งานจริง

ในส่วนนี้จะแบ่งการนำเสนอผลการวิจัยออกเป็นหัวข้อย่อยตามตัวแบบที่พัฒนาขึ้น ได้แก่ การถดถอยเชิงเส้นพหุคูณ การถดถอยแบบเร็กคิวลาไรซ์เซชัน การถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน การถดถอยแบบต้นไม้ตัดสินใจ การถดถอยแบบป่าสุ่ม การถดถอยแบบเอ็กจิบูส การถดถอยแบบแคตบูส การถดถอยแบบไลทจีบีเอ็ม และโครงข่ายประสาทเทียม เพื่อแสดงถึงกระบวนการและผลการวิเคราะห์ในเชิงลึก พร้อมเปรียบเทียบประสิทธิภาพของแต่ละตัวแบบบนเกณฑ์การวัด เช่น ค่าผิดพลาดกำลังสองเฉลี่ย (MSE) และค่าสัมประสิทธิ์แสดงการตัดสินใจ (Coefficient of Determination:  $R^2$ )

การอภิปรายผลในแต่ละหัวข้อจะเชื่อมโยงข้อมูลจากตารางสรุปผลการทดลอง เพื่อให้เห็นภาพรวมของประสิทธิภาพของตัวแบบ รวมทั้งระบุข้อดีและข้อจำกัดของแต่ละวิธีอย่างชัดเจน ซึ่งสามารถนำไปใช้เป็นแนวทางในการเลือกใช้ตัวแบบที่เหมาะสมสำหรับงานวิจัยหรือการประยุกต์ใช้ในอนาคต

#### 4.1 การสร้างตัวแบบถดถอยเชิงเส้นพหุคูณ

ในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณมีวิธีการคัดเลือกตัวแปรอิสระที่จะนำเข้าไปสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่า P-Value  $\leq 0.05$
- ใช้ตัวแปรอิสระที่มีค่า P-Value  $\leq 0.05$  และ VIF  $\leq 10$

การคัดเลือกตัวแปรอิสระสำหรับการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณเริ่มต้นจากการพิจารณา 3 กรณีหลัก เพื่อให้ตัวแบบมีความแม่นยำและเหมาะสมที่สุด ในกรณีแรก ตัวแปรอิสระทั้งหมดจะถูกใช้ในการสร้างตัวแบบ โดยไม่มีการคัดกรองเพิ่มเติม ส่วนในกรณีที่สอง มีการคัดเลือกตัว

แปรอิสระโดยพิจารณาจากค่า P-Value  $\leq 0.05$  ซึ่งเป็นเกณฑ์ที่บ่งชี้ถึงความสัมพันธ์เชิงสถิติที่มีนัยสำคัญระหว่างตัวแปรตามและตัวแปรอิสระ ผลการคัดเลือกในกรณีนี้ปรากฏใน ตารางที่ 4.1 ซึ่งพบว่า BFW#1\_MOTOR\_BEARING\_TEMP#A, BFW#1\_MOTOR\_BEARING\_TEMP#E, BFW#1\_MOTOR\_BEARING\_TEMP#F และ BFW#1\_PUMP\_BEARING\_TEMP#B ถูกคัดออกจากตัวแปรอิสระเนื่องจากไม่มีนัยสำคัญทางสถิติต่อการอธิบายตัวแปรตามในตัวแบบ สำหรับกรณีที่สามมีการพิจารณาตัวแปรอิสระเพิ่มเติม โดยไม่เพียงแต่ต้องมีค่า P-Value  $\leq 0.05$  เท่านั้น แต่ยังต้องตรวจสอบค่าดัชนี VIF (Variance Inflation Factor) ซึ่งบ่งชี้ถึงปัญหาความสัมพันธ์ระหว่างตัวแปรอิสระที่อาจเกิดขึ้น ตัวแปรที่มีค่า VIF  $\leq 10$  จะถูกเลือกเข้าสู่การสร้างตัวแบบเพื่อหลีกเลี่ยงปัญหาความสัมพันธ์เชิงเส้นร่วม (Multicollinearity) ผลการคัดเลือกในขั้นตอนนี้แสดงใน ตารางที่ 4.2 พบว่า BFW#1\_MOTOR\_WINDING\_TEMP#A, BFW#1\_MOTOR\_WINDING\_TEMP#B, BFW#1\_MOTOR\_WINDING\_TEMP#D และ BFW#1\_BEARING\_HOUSING\_VIBR\_A ถูกคัดออกในรอบนี้ และตารางที่ 4.3 เป็นการตรวจสอบค่า VIF อีกครั้งเพื่อยืนยันว่าไม่มีปัญหาความสัมพันธ์ระหว่างตัวแปรอิสระ

**ตารางที่ 4.1** การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่า P-Value  $\leq 0.05$

ตัวแปรอิสระ	ค่า P-value	ดัชนี VIF
BFW#1_PUMP_BEARING_TEMP#A	0.0000	11.2210
BFW#1_PUMP_BEARING_TEMP#B	0.5630	10.8596
BFW#1_PUMP_BEARING_TEMP#C	0.0015	3.4026
BFW#1_MOTOR_BEARING_TEMP#A	0.1260	11.8249
BFW#1_MOTOR_BEARING_TEMP#B	0.0000	8.4596
BFW#1_BEARING_HOUSING_VIBR_A	0.0000	25.3843
BFW#1_BEARING_HOUSING_VIBR_B	0.0000	10.3919
BFW#1_MOTOR_WINDING_TEMP#A	0.0000	32.2614
BFW#1_MOTOR_WINDING_TEMP#B	0.0000	36.4709
BFW#1_MOTOR_WINDING_TEMP#C	0.0000	4.6728
BFW#1_MOTOR_WINDING_TEMP#D	0.0047	39.9747
BFW#1_MOTOR_WINDING_TEMP#E	0.2045	40.2824
BFW#1_MOTOR_WINDING_TEMP#F	0.6818	29.2201
BFW_PUMP_DISCHARGE_PRESSURE	0.0000	8.9362

ตัวแปรอิสระ	ค่า P-value	ดัชนี VIF
GTG11_POWER	0.0000	1.3497
GTG12_POWER	0.0000	1.3332

ตารางที่ 4.2 การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่าดัชนี VIF  $\leq 10$  (รอบที่ 1)

ตัวแปรอิสระ	ดัชนี VIF
BFW#1_PUMP_BEARING_TEMP#A	5.1647
BFW#1_PUMP_BEARING_TEMP#C	1.7207
BFW#1_MOTOR_BEARING_TEMP#B	5.2753
BFW#1_BEARING_HOUSING_VIBR_A	23.7778
BFW#1_BEARING_HOUSING_VIBR_B	9.5159
BFW#1_MOTOR_WINDING_TEMP#A	27.1304
BFW#1_MOTOR_WINDING_TEMP#B	30.1957
BFW#1_MOTOR_WINDING_TEMP#C	4.4980
BFW#1_MOTOR_WINDING_TEMP#D	34.6637
BFW_PUMP_DISCHARGE_PRESSURE	8.5965
GTG11_POWER	1.3365
GTG12_POWER	1.3328

ตารางที่ 4.3 การคัดเลือกตัวแปรอิสระในการสร้างตัวแบบการถดถอยเชิงเส้นพหุคูณด้วยค่าดัชนี VIF  $\leq 10$  (รอบที่ 2)

ตัวแปรอิสระ	ดัชนี VIF
BFW#1_PUMP_BEARING_TEMP#A	1.9687
BFW#1_PUMP_BEARING_TEMP#C	1.6983
BFW#1_MOTOR_BEARING_TEMP#B	2.6387
BFW#1_BEARING_HOUSING_VIBR_B	1.4113
BFW#1_MOTOR_WINDING_TEMP#C	3.6350
BFW_PUMP_DISCHARGE_PRESSURE	1.6990

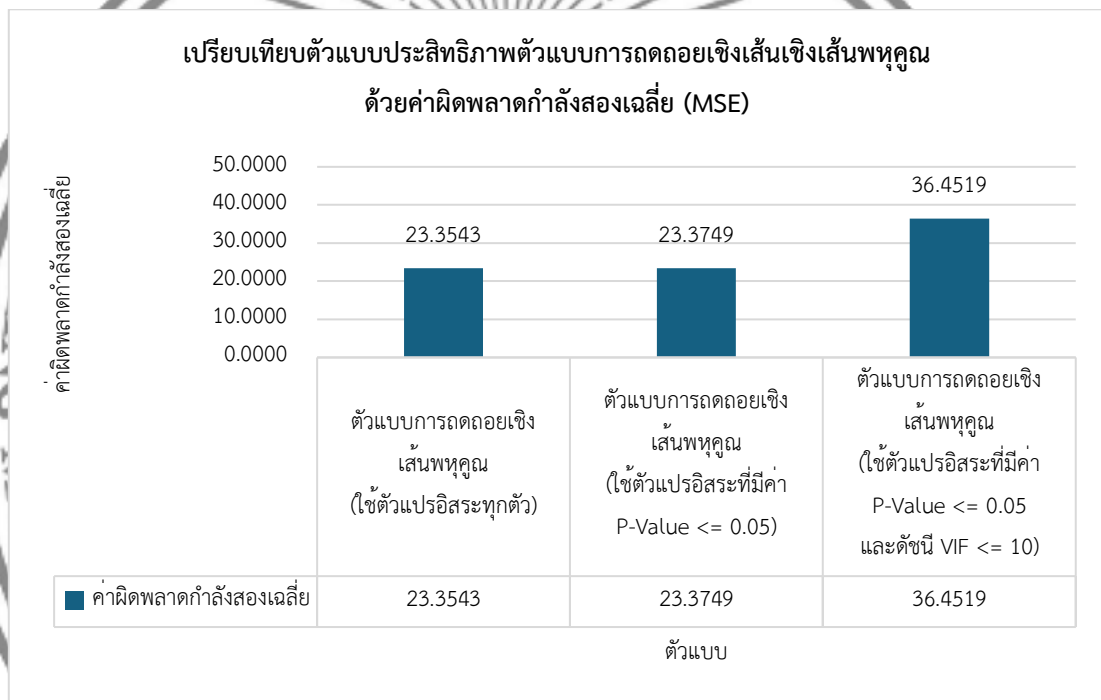
ตัวแปรอิสระ	ดัชนี VIF
GTG11_POWER	1.3172
GTG12_POWER	1.3846

ผลลัพธ์จากการเปรียบเทียบประสิทธิภาพของตัวแบบการถดถอยเชิงเส้นพหุคูณ (Multiple Linear Regression) ในการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ ดังตารางที่ 4.4 แสดงให้เห็นถึงความแตกต่างของค่าประสิทธิภาพระหว่างตัวแบบที่ใช้ตัวแปรอิสระในกรณีต่าง ๆ ตัวแบบที่ใช้ ตัวแปรอิสระทุกตัว มีค่า MSE เท่ากับ 23.3543 และค่า  $R^2$  เท่ากับ 0.9721 ซึ่งแสดงให้เห็นว่าตัวแบบสามารถทำนายค่าการใช้พลังงานได้อย่างแม่นยำโดยอาศัยตัวแปรอิสระทั้งหมด แม้ว่าความซับซ้อนของตัวแบบอาจเพิ่มขึ้นเนื่องจากจำนวนตัวแปรที่ใช้งาน สำหรับตัวแบบที่ใช้ ตัวแปรอิสระที่มีค่า P-Value  $\leq 0.05$  เท่านั้น มีค่า MSE ใกล้เคียงกับตัวแบบแรก คือ 23.3749 และค่า  $R^2$  เท่ากัน คือ 0.9721 ซึ่งชี้ให้เห็นว่าการคัดเลือกตัวแปรโดยพิจารณาจากความสัมพันธ์เชิงสถิติที่มีนัยสำคัญเพียงอย่างเดียว ไม่ได้ส่งผลกระทบต่อความแม่นยำของตัวแบบมากนัก แต่ช่วยลดจำนวนตัวแปรที่ไม่จำเป็น ทำให้ตัวแบบมีความง่ายขึ้นเล็กน้อย ในขณะที่ตัวแบบที่ใช้ ตัวแปรอิสระที่มีค่า P-Value  $\leq 0.05$  และดัชนี VIF  $\leq 10$  มีค่า MSE เพิ่มขึ้นเป็น 36.4519 และค่า  $R^2$  ลดลงเหลือ 0.9565 แสดงให้เห็นว่าการลดจำนวนตัวแปรอิสระลงเพื่อหลีกเลี่ยงปัญหาความสัมพันธ์เชิงเส้นร่วม (Multicollinearity) อาจส่งผลกระทบต่อความแม่นยำในการทำนายของตัวแบบ.

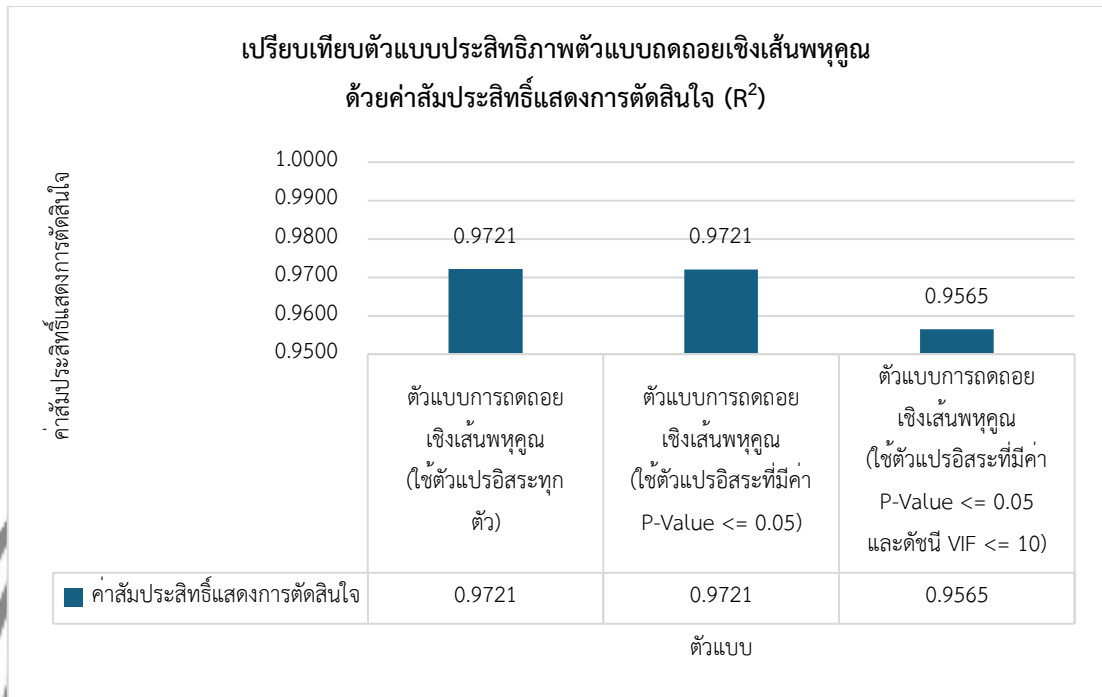
ตารางที่ 4.4 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยเชิงเส้นพหุคูณ

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบการถดถอยเชิงเส้นพหุคูณ (ใช้ตัวแปรอิสระทุกตัว)	23.3543	0.9721
ตัวแบบการถดถอยเชิงเส้นพหุคูณ (ใช้ตัวแปรอิสระที่มีค่า P-Value $\leq 0.05$ )	23.3749	0.9721
ตัวแบบการถดถอยเชิงเส้นพหุคูณ (ใช้ตัวแปรอิสระที่มีค่า P-Value $\leq 0.05$ และดัชนี VIF $\leq 10$ )	36.4519	0.9565

โดยรูปที่ 4.1 และ 4.2 แสดงให้เห็นถึงผลกระทบของการคัดเลือกตัวแปรอิสระต่อประสิทธิภาพของตัวแบบการถดถอยเชิงเส้นพหุคูณ โดยตัวแบบที่ใช้ตัวแปรอิสระทุกตัวและตัวแบบที่ใช้เกณฑ์  $P\text{-Value} \leq 0.05$  มีค่า MSE ใกล้เคียงกันและมีค่า  $R^2$  สูงสุดที่ 0.9721 ซึ่งแสดงถึงความแม่นยำสูงสุด ในขณะที่ตัวแบบที่ใช้เกณฑ์  $P\text{-Value} \leq 0.05$  และ  $VIF \leq 10$  มี MSE สูงขึ้น 36.4519 และค่า  $R^2$  ลดลง 0.9565 สะท้อนถึงผลกระทบของการลดจำนวนตัวแปรในตัวแบบที่อาจลดความแม่นยำลง



รูปที่ 4.1 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นเชิงเส้นพหุคูณด้วยค่า MSE



รูปที่ 4.2 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยเชิงเส้นพหุคูณด้วยค่า  $R^2$

#### 4.2 การสร้างตัวแบบการถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชัน

ในการสร้างตัวแบบเร็กคิวลาไรซ์เซชันแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ตัวแบบการถดถอยเชิงเส้นแบบลาซโซ
- ตัวแบบการถดถอยเชิงเส้นแบบบริดจ์
- ตัวแบบการถดถอยแบบอิลาสติกเน็ต

การสร้างตัวแบบการถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชันในงานวิจัยนี้ได้แบ่งการทดลองออกเป็น 3 กรณี ได้แก่ การถดถอยเชิงเส้นแบบลาซโซ, การถดถอยเชิงเส้นแบบบริดจ์ และการถดถอยเชิงเส้นแบบอิลาสติกเน็ต ซึ่งทั้งสามวิธีใช้แนวคิดการเพิ่มฟังก์ชันลงโทษ เพื่อลดความซับซ้อนของตัวแบบและควบคุมปัญหาความสัมพันธ์ระหว่างตัวแปร ในการเลือกพารามิเตอร์ การค้นหาแบบสุ่ม ถูกนำมาใช้ผลลัพธ์ ดังตารางที่ 4.5 พบว่าพารามิเตอร์ที่ดีที่สุดของตัวแบบเร็กคิวลาไรซ์เซชันมีค่า Alpha เท่ากับ 0.01 ทั้ง 3 ตัวแบบ และ ค่า  $l1\_ratio$  เท่ากับ 0.80 สำหรับตัวแบบการถดถอยเชิงเส้นแบบอิลาสติกเน็ต

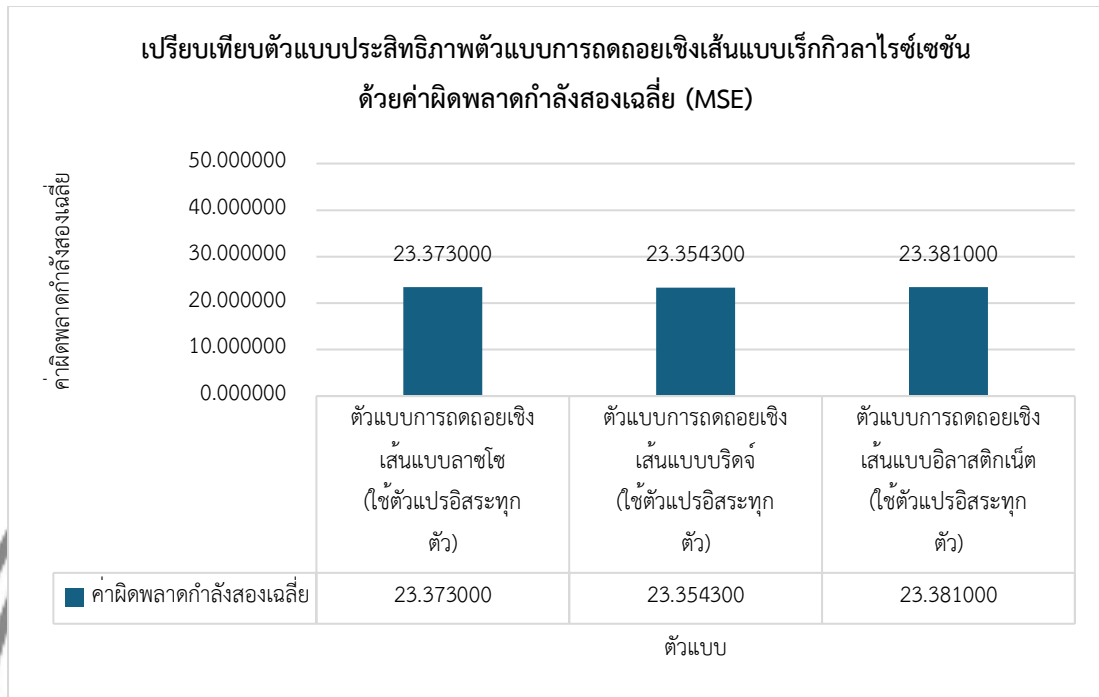
ตารางที่ 4.5 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยเชิงเส้นแบบเร็กกิวลาไรซ์เซชัน

ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบการถดถอยเชิงเส้นแบบลาซโซ (ใช้ตัวแปรอิสระทุกตัว)	Alpha	0.01, 0.1, 1.0, 10.0	0.01
ตัวแบบการถดถอยเชิงเส้นแบบบริดจ์ (ใช้ตัวแปรอิสระทุกตัว)	Alpha	0.01, 0.1, 1.0, 10.0	0.01
ตัวแบบการถดถอยเชิงเส้นแบบอิลาสติกเน็ต (ใช้ตัวแปรอิสระทุกตัว)	Alpha	0.01, 0.1, 1.0, 10.0	0.01
	l1_ratio	0.20, 0.50, 0.80	0.80

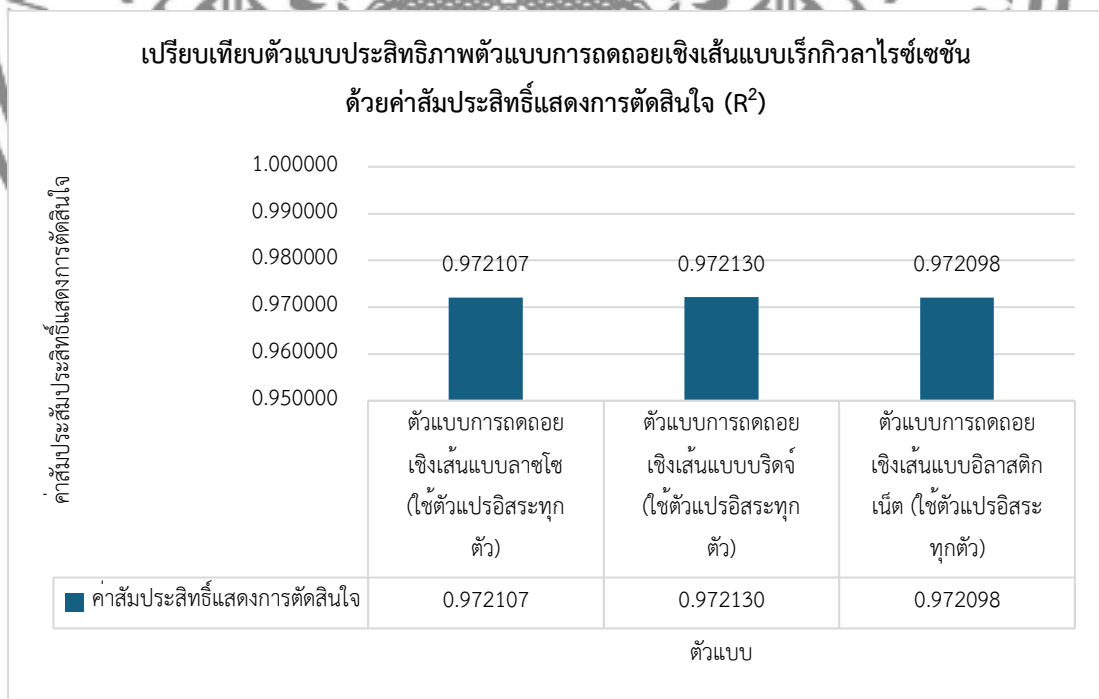
จากการเปรียบเทียบผลลัพธ์ประสิทธิภาพตัวแบบเร็กกิวลาไรซ์เซชันใน ตารางที่ 4.6 พบว่าตัวแบบการถดถอยเชิงเส้นแบบบริดจ์มีค่า MSE ต่ำที่สุด 23.3543 และค่า  $R^2$  สูงที่สุด 0.9721 เมื่อเทียบกับตัวแบบลาซโซและอิลาสติกเน็ต แสดงให้เห็นว่าบริดจ์สามารถสร้างตัวแบบที่มีความแม่นยำสูงกว่าเล็กน้อย ในขณะที่ตัวแบบลาซโซมีค่า MSE เท่ากับ 23.3730 และ  $R^2$  เท่ากับ 0.9721 ตัวแบบอิลาสติกเน็ต มีค่า MSE เท่ากับ 23.3810 และ  $R^2$  เท่ากับ 0.9721 ซึ่งทั้ง 3 ตัวแบบให้ผลลัพธ์ใกล้เคียงกัน ดังรูปที่ 4.3 และ 4.4 แต่มีความแตกต่างในด้านโครงสร้างตัวแบบ โดยลาซโซมีความสามารถในการคัดเลือกตัวแปรได้โดยอัตโนมัติ ตัวแบบอิลาสติกเน็ตให้ความยืดหยุ่นในการปรับใช้คุณสมบัติของทั้งลาซโซและบริดจ์ร่วมกัน

ตารางที่ 4.6 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยเชิงเส้นแบบเร็กกิวลาไรซ์เซชัน

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบการถดถอยเชิงเส้นแบบลาซโซ (ใช้ตัวแปรอิสระทุกตัว)	23.3730	0.9721
ตัวแบบการถดถอยเชิงเส้นแบบบริดจ์ (ใช้ตัวแปรอิสระทุกตัว)	23.3543	0.9721
ตัวแบบการถดถอยเชิงเส้นแบบอิลาสติกเน็ต (ใช้ตัวแปรอิสระทุกตัว)	23.3810	0.9721



รูปที่ 4.3 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชันด้วยค่า MSE



รูปที่ 4.4 เปรียบเทียบประสิทธิภาพตัวแบบการถดถอยเชิงเส้นแบบเร็กคิวลาไรซ์เซชันด้วยค่า  $R^2$

### 4.3 การสร้างตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน

ในการสร้างตัวแบบถดถอยซัพพอร์ตเวกเตอร์แมชชีนแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (SVR) ได้แบ่งการสร้างตัวแบบออกเป็น 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทุกตัว การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก โดยใช้ฟังก์ชัน coefficients โดยค่าความสำคัญของตัวแปรอิสระในแต่ละกรณีได้ถูกคำนวณและจัดอันดับดังแสดงในตารางที่ 4.7 ซึ่งพบว่า ตัวแปร BFW\_PUMP\_DISCHARGE\_PRESSURE มีค่าความสำคัญสูงสุดที่ 0.5958 ซึ่งแสดงว่าตัวแปรนี้มีผลกระทบต่อการทำนายค่าการใช้พลังงาน ในขณะที่ตัวแปร BFW#1\_BEARING\_HOUSING\_VIBR\_A และ BFW#1\_MOTOR\_WINDING\_TEMP#A มีค่าความสำคัญรองลงมา คือ 0.3693 และ 0.0526 ตามลำดับ ตัวแปรที่มีค่าความสำคัญน้อย เช่น GTG11\_POWER และ GTG12\_POWER 0.0030 และ 0.00544 ตามลำดับ มีผลกระทบต่อการทำนายค่าการใช้พลังงานน้อย

ตารางที่ 4.7 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีซัพพอร์ตเวกเตอร์แมชชีน

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	0.5958	1
BFW#1_BEARING_HOUSING_VIBR_A	0.3693	2
BFW#1_MOTOR_WINDING_TEMP#A	0.0526	3
BFW#1_MOTOR_WINDING_TEMP#D	0.0488	4
BFW#1_MOTOR_WINDING_TEMP#E	0.0468	5
BFW#1_MOTOR_WINDING_TEMP#F	0.0467	6
BFW#1_MOTOR_WINDING_TEMP#B	0.0440	7
BFW#1_BEARING_HOUSING_VIBR_B	0.0414	8
BFW#1_MOTOR_BEARING_TEMP#A	0.0284	9
BFW#1_PUMP_BEARING_TEMP#A	0.0231	10
BFW#1_MOTOR_WINDING_TEMP#C	0.0219	11

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW#1_MOTOR_BEARING_TEMP#B	0.0188	12
BFW#1_PUMP_BEARING_TEMP#C	0.0107	13
BFW#1_PUMP_BEARING_TEMP#B	0.0106	14
GTG12_POWER	0.0054	15
GTG11_POWER	0.0030	16

กระบวนการปรับพารามิเตอร์ของ SVR ใช้การค้นหาแบบสุ่ม สำหรับค่าพารามิเตอร์ C และ epsilon เพื่อหาพารามิเตอร์ที่เหมาะสมที่สุดในแต่ละกรณี โดยผลการค้นหาแสดงให้เห็นว่า ค่าที่เหมาะสมสำหรับทั้งสามกรณีคือ  $C = 100$  และ  $\epsilon = 0.5$  ดังแสดงในตารางที่ 4.8 ซึ่งสะท้อนถึงความสม่ำเสมอของผลลัพธ์จากการปรับแต่งพารามิเตอร์

ตารางที่ 4.8 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน

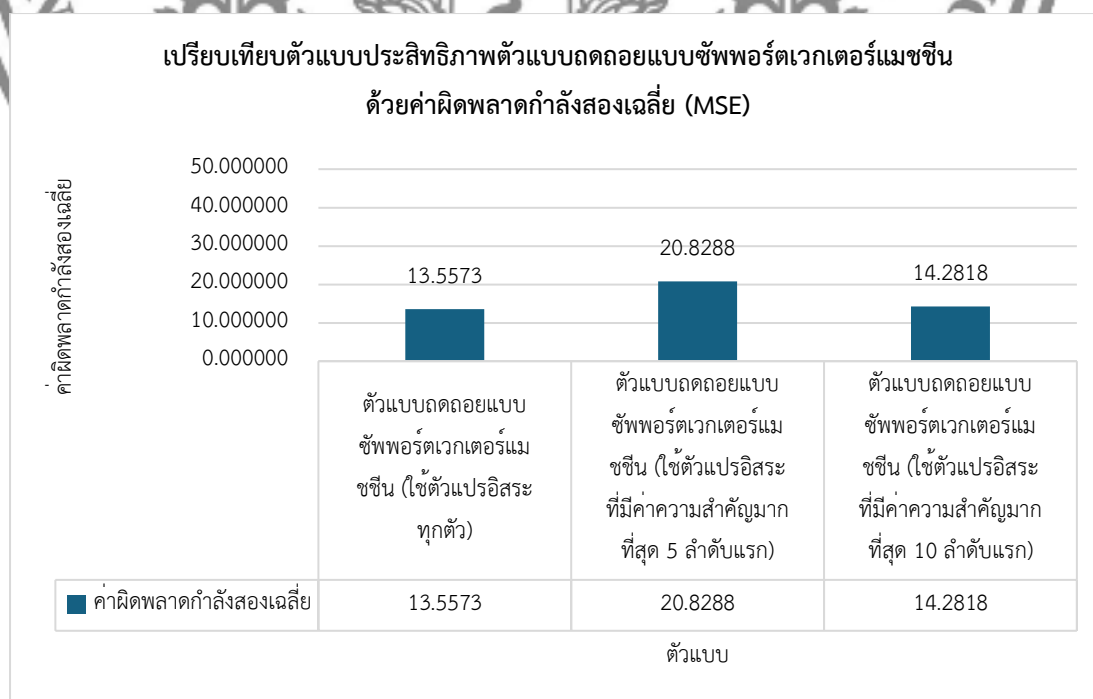
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระทุกตัว)	C	0.1, 1, 10, 100	100
	epsilon	0.1, 0.5	0.5
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	C	0.1, 1, 10, 100	100
	epsilon	0.1, 0.5	0.5
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	C	0.1, 1, 10, 100	100
	epsilon	0.1, 0.5	0.5

จากการเปรียบเทียบประสิทธิภาพของตัวแบบในตารางที่ 4.9 พบว่าตัวแบบที่ใช้ตัวแปรอิสระทุกตัวมีค่า MSE ต่ำที่สุดที่ 13.5573 และมีค่า  $R^2$  สูงที่สุดที่ 0.9838 แสดงให้เห็นถึงความแม่นยำสูงสุดในการทำนายการใช้พลังงาน ในขณะที่ตัวแบบที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก มี MSE ใกล้เคียงกันคือ 14.2818 และค่า  $R^2$  เท่ากับ 0.9830 ส่วนตัวแบบที่ใช้ตัวแปรอิสระ

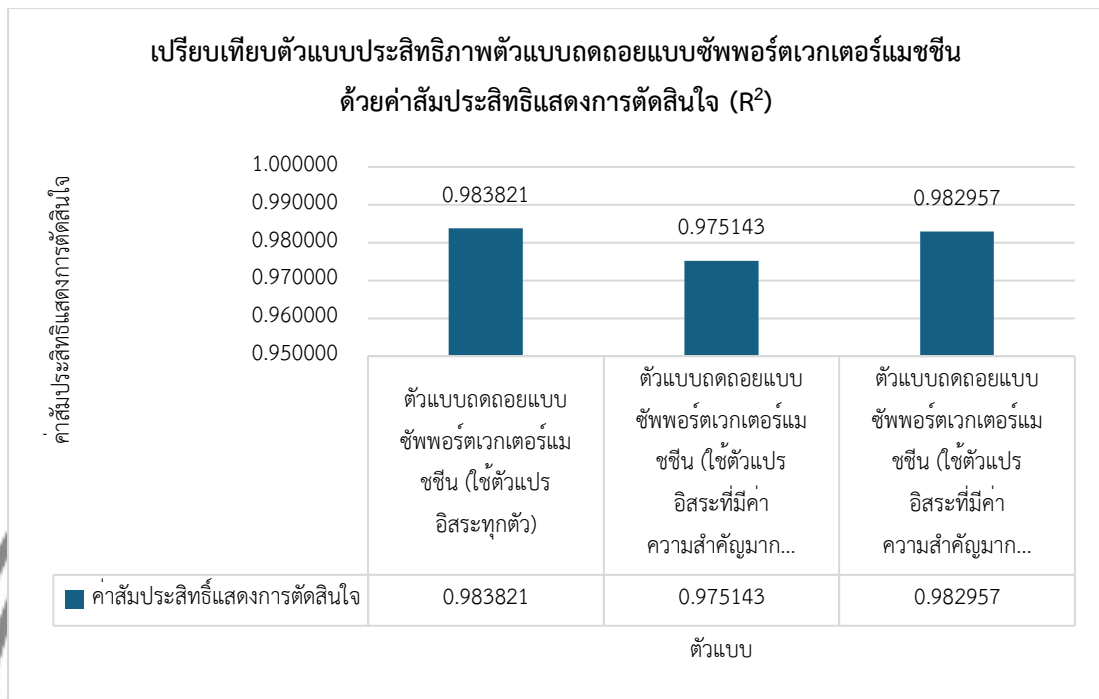
ที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก มีความแม่นยำลดลงเล็กน้อยด้วย MSE เท่ากับ 20.8288 และค่า  $R^2$  เท่ากับ 0.9751 ดังรูปที่ 4.5 และ 4.6 การเปรียบเทียบดังกล่าวเน้นให้เห็นถึงความสำคัญของการเลือกตัวแปรอิสระที่ครอบคลุมเพียงพอเพื่อให้ได้ตัวแบบที่มีความแม่นยำในการทำนายสูงสุด.

ตารางที่ 4.9 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยเชิงเส้นแบบซัพพอร์ตเวกเตอร์แมชชีน

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระทุกตัว)	13.5573	0.9838
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	20.8288	0.9751
ตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	14.2818	0.9830



รูปที่ 4.5 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีนด้วยค่า MSE



รูปที่ 4.6 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีนด้วยค่า MSE

#### 4.4 การสร้างตัวแบบถดถอยแบบต้นไม้ตัดสินใจ

ในการสร้างตัวแบบถดถอยแบบต้นไม้ตัดสินใจแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบต้นไม้ตัดสินใจ ได้แบ่งกระบวนการสร้างออกเป็น 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทุกตัว การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก โดยใช้ฟังก์ชัน `feature_importances` จากนั้น เปรียบเทียบประสิทธิภาพ ของตัวแบบในการทำนายพลังงานการใช้งานของเครื่องสูบน้ำป้อนหม้อไอน้ำ โดยค่าความสำคัญของตัวแปรอิสระในแต่ละกรณีได้ถูกคำนวณและแสดงในตารางที่ 4.10 ซึ่งพบว่าตัวแปร `BFW_PUMP_DISCHARGE_PRESSURE` มีค่าความสำคัญสูงสุดที่ 0.9369 แสดงถึงความสัมพันธ์ที่เด่นชัดกับตัวแปรตาม ในขณะที่ตัวแปร `BFW#1_MOTOR_WINDING_TEMP#E` และ `BFW#1_MOTOR_WINDING_TEMP#D` มีค่าความสำคัญต่ำที่สุดเท่ากันที่ 0.0006 แสดงถึงว่า อุณหภูมิของขดลวดมอเตอร์มีผลกระทบต่อการทำนายพลังงาน

ตารางที่ 4.10 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีต้นไม้ตัดสินใจ

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	0.9369	1
BFW#1_BEARING_HOUSING_VIBR_A	0.0434	2
BFW#1_BEARING_HOUSING_VIBR_B	0.0063	3
GTG12_POWER	0.0020	4
GTG11_POWER	0.0017	5
BFW#1_PUMP_BEARING_TEMP#B	0.0013	6
BFW#1_PUMP_BEARING_TEMP#A	0.0012	7
BFW#1_MOTOR_WINDING_TEMP#A	0.0011	8
BFW#1_MOTOR_WINDING_TEMP#C	0.0011	9
BFW#1_PUMP_BEARING_TEMP#C	0.0010	10
BFW#1_MOTOR_BEARING_TEMP#B	0.0008	11
BFW#1_MOTOR_BEARING_TEMP#A	0.0007	12
BFW#1_MOTOR_WINDING_TEMP#B	0.0007	13
BFW#1_MOTOR_WINDING_TEMP#F	0.0007	14
BFW#1_MOTOR_WINDING_TEMP#D	0.0006	15
BFW#1_MOTOR_WINDING_TEMP#E	0.0006	16

ในการปรับแต่งตัวแบบ กระบวนการค้นหาพารามิเตอร์แบบสุ่มถูกนำมาใช้ โดยพิจารณา ค่าพารามิเตอร์สำคัญ แสดงในตารางที่ 4.11 ซึ่งผลการค้นหาพารามิเตอร์ที่เหมาะสมที่สุดสำหรับตัวแบบการถดถอยแบบต้นไม้ตัดสินใจในแต่ละกรณี โดยพิจารณาพารามิเตอร์สำคัญ 3 ค่า ได้แก่ max\_depth ซึ่งกำหนดความลึกสูงสุดของต้นไม้, min\_samples\_split จำนวนตัวอย่างขั้นต่ำที่ต้องใช้ในการแบ่งโหนด และ min\_samples\_leaf จำนวนตัวอย่างขั้นต่ำในแต่ละใบ (Leaf) ของต้นไม้ การปรับแต่งนี้ช่วยให้ตัวแบบมีความสมดุลระหว่างความแม่นยำและความซับซ้อนของตัวแบบในแต่ละกรณี สำหรับกรณีที่ใช้ตัวแปรอิสระทุกตัว พบว่าพารามิเตอร์ที่ดีที่สุดคือ max\_depth เท่ากับ 10, min\_samples\_split เท่ากับ 5, และ min\_samples\_leaf เท่ากับ 5 ซึ่งเหมาะสมสำหรับตัวแบบที่มีตัวแปรจำนวนมากและต้องการควบคุมความซับซ้อน ในกรณีที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก พบว่า min\_samples\_split เพิ่มขึ้นเป็น 10 และ min\_samples\_leaf ลดลง

เหลือ 1 เพื่อเพิ่มความยืดหยุ่นของตัวแบบให้สามารถจัดการกับข้อมูลจำนวนน้อยลงได้ ในขณะที่กรณีที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก พบว่าพารามิเตอร์ที่ดีที่สุดเหมือนกับกรณีแรก (max\_depth เท่ากับ 10, min\_samples\_split เท่ากับ 5, min\_samples\_leaf เท่ากับ 5) แสดงให้เห็นว่าการเพิ่มจำนวนตัวแปรสำคัญช่วยให้ตัวแบบรักษาสมดุลระหว่างความแม่นยำและความซับซ้อนของโครงสร้างได้ดี

ตารางที่ 4.11 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบต้นไม้ตัดสินใจ

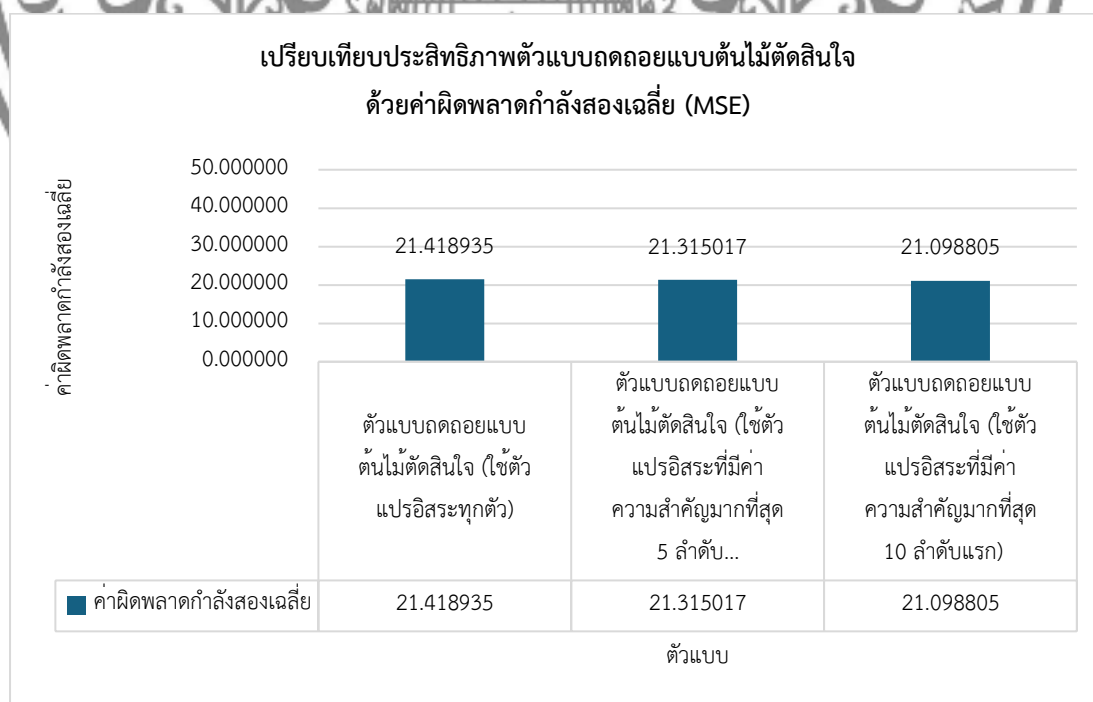
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระทุกตัว)	max_depth	5, 10, 20	10
	min_samples_split	2, 5, 10	5
	min_samples_leaf	1, 2, 5	5
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	max_depth	5, 10, 20	10
	min_samples_split	2, 5, 10	10
	min_samples_leaf	1, 2, 5	1
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	max_depth	5, 10, 20	10
	min_samples_split	2, 5, 10	5
	min_samples_leaf	1, 2, 5	5

จากผลการเปรียบเทียบประสิทธิภาพของตัวแบบในตารางที่ 4.12 พบว่าตัวแบบที่ใช้ตัวแปรอิสระทุกตัวมีค่า MSE เท่ากับ 21.4189 และค่า  $R^2$  เท่ากับ 0.9744 แสดงให้เห็นถึงประสิทธิภาพในการทำนายที่ดี อย่างไรก็ตาม การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรกมีค่า MSE เท่ากับ 21.3150 และ  $R^2$  เพิ่มขึ้นเป็น 0.9746 และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก ให้ผลลัพธ์ที่ดีกว่าเล็กน้อย โดยมีค่า MSE ลดลงเหลือ 21.0988 และค่า  $R^2$  เพิ่มขึ้นเป็น 0.9748 แสดงถึงการปรับลดจำนวนตัวแปรที่เหมาะสมสามารถลดความซับซ้อนของตัวแบบได้โดยไม่กระทบต่อความแม่นยำ จากรูปที่ 4.7 และ 4.8 แสดงให้เห็นถึงความแตกต่างด้านค่า MSE และค่า  $R^2$  ระหว่างตัวแบบในแต่ละกรณี ซึ่งเน้นย้ำถึงความสำคัญของการเลือกตัวแปรอิสระที่เหมาะสมเพื่อสร้างตัวแบบที่มีประสิทธิภาพสูงสุด ทั้งนี้ ตัวแบบการถดถอยแบบต้นไม้ตัดสินใจสามารถนำไปใช้ได้ดีใน

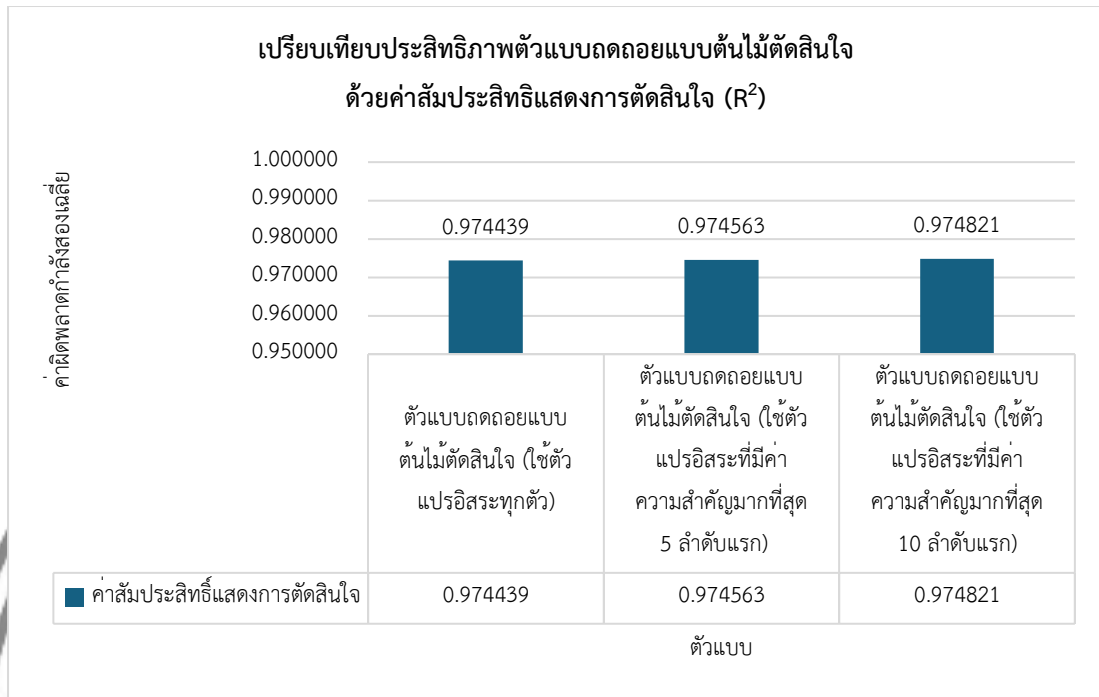
บริบทที่ต้องการความชัดเจนในโครงสร้างการตัดสินใจและความเข้าใจในความสัมพันธ์ของตัวแปรอิสระต่อการทำนาย

ตารางที่ 4.12 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจ

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระทุกตัว)	21.4189	0.9744
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	21.3150	0.9746
ตัวแบบถดถอยแบบต้นไม้ตัดสินใจ (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	21.0988	0.9748



รูปที่ 4.7 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจด้วยค่า MSE



รูปที่ 4.8 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบต้นไม้ตัดสินใจด้วยค่า  $R^2$

#### 4.5 การสร้างตัวแบบถดถอยแบบป่าสุ่ม

ในการสร้างตัวแบบถดถอยแบบป่าสุ่มแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบป่าสุ่ม ได้แบ่งกระบวนการสร้างตัวแบบออกเป็น 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทุกตัว การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก โดยการคัดเลือกตัวแปรอิสระนี้ดำเนินการจากค่าความสำคัญที่คำนวณได้จากโดยใช้ฟังก์ชัน `feature_importances` ดังที่แสดงในตารางที่ 4.13 พบว่า ตัวแปร `BFW_PUMP_DISCHARGE_PRESSURE` มีค่าความสำคัญสูงสุดที่ 0.9379 ซึ่งแสดงถึงการมีอิทธิพลต่อการทำนายค่าการใช้พลังงานของตัวแบบ ในขณะที่ ตัวแปร `BFW#1_MOTOR_WINDING_TEMP#E` และ `BFW#1_MOTOR_WINDING_TEMP#B` มีค่าความสำคัญต่ำสุดเท่ากัน 0.0006 เช่นเดียวกับวิธีการสร้างตัวแบบถดถอยแบบต้นไม้ตัดสินใจ ซึ่งแสดงว่าตัวแบบที่มีพื้นฐานการทำงานในลักษณะกลุ่มต้นไม้ ตัวแปรอุณหภูมิของขดลวดมอเตอร์มีผลกระทบน้อยต่อการทำนายพลังงาน

ตารางที่ 4.13 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีป่าสุ่ม

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	0.9379	1
BFW#1_BEARING_HOUSING_VIBR_A	0.0428	2
BFW#1_BEARING_HOUSING_VIBR_B	0.0058	3
GTG12_POWER	0.0019	4
GTG11_POWER	0.0017	5
BFW#1_PUMP_BEARING_TEMP#B	0.0015	6
BFW#1_PUMP_BEARING_TEMP#A	0.0012	7
BFW#1_MOTOR_WINDING_TEMP#C	0.0011	8
BFW#1_PUMP_BEARING_TEMP#C	0.0010	9
BFW#1_MOTOR_WINDING_TEMP#A	0.0010	10
BFW#1_MOTOR_BEARING_TEMP#B	0.0008	11
BFW#1_MOTOR_WINDING_TEMP#F	0.0008	12
BFW#1_MOTOR_BEARING_TEMP#A	0.0007	13
BFW#1_MOTOR_WINDING_TEMP#D	0.0007	14
BFW#1_MOTOR_WINDING_TEMP#B	0.0006	15
BFW#1_MOTOR_WINDING_TEMP#E	0.0006	16

ในการปรับแต่งพารามิเตอร์ ตัวแบบถดถอยแบบป่าสุ่มได้ใช้การค้นหาแบบสุ่ม โดยใช้ข้อมูลฝึกหัด และทำการใช้เทคนิค k-fold cross validation เพื่อหาค่าพารามิเตอร์ที่เหมาะสมที่สุดในแต่ละกรณี โดยค่าที่พิจารณาได้แก่ n\_estimators ซึ่งกำหนดจำนวนต้นไม้ และ max\_depth ซึ่งกำหนดความลึกสูงสุดของต้นไม้ ผลการปรับแต่งดังกล่าวแสดงในตารางที่ 4.14 พบว่า ค่าพารามิเตอร์ที่เหมาะสมที่สุดในทุกกรณีคือ n\_estimators เท่ากับ 200 และ max\_depth เท่ากับ 20 ซึ่งช่วยให้ตัวแบบสามารถปรับสมดุลระหว่างความซับซ้อนและความแม่นยำได้ดี

ตารางที่ 4.14 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบป่าสุ่ม

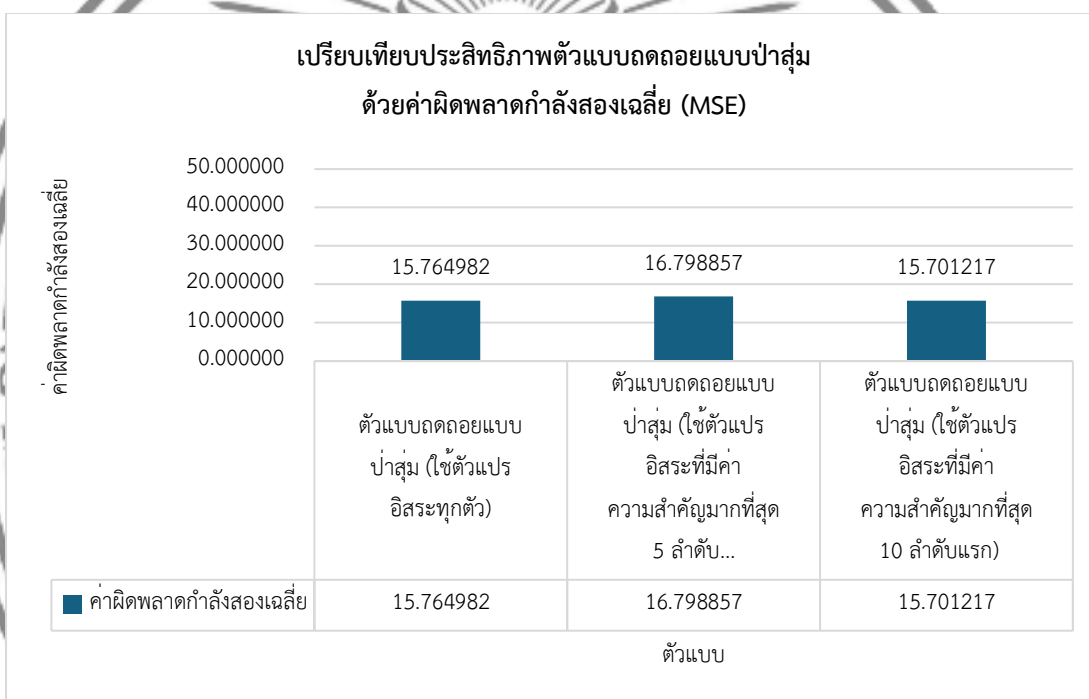
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบป่าสุ่ม(ใช้ตัวแปรอิสระทุกตัว)	n_estimators	100, 200	200
	max_depth	10, 20	20
ตัวแบบถดถอยแบบป่าสุ่ม(ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	n_estimators	100, 200	200
	max_depth	10, 20	20
ตัวแบบถดถอยแบบป่าสุ่ม (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	n_estimators	100, 200	200
	max_depth	10, 20	20

จากการเปรียบเทียบผลลัพธ์ของตัวแบบในตารางที่ 4.15 พบว่า ตัวแบบที่ใช้ตัวแปรอิสระทุกตัวมีค่า MSE เท่ากับ 15.7650 และค่า  $R^2$  เท่ากับ 0.9812 ขณะที่ตัวแบบที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก มีค่า MSE ลดลงเล็กน้อยเป็น 15.7012 และค่า  $R^2$  เพิ่มขึ้นเล็กน้อยเป็น 0.9813 ซึ่งแสดงถึงความแม่นยำสูงสุดในการทำนายการใช้พลังงานในกลุ่มตัวแบบถดถอยแบบป่าสุ่ม แสดงให้เห็นว่าการลดจำนวนตัวแปรอิสระบางตัวช่วยลดความซับซ้อนของตัวแบบโดยไม่กระทบต่อความแม่นยำ ส่วนตัวแบบที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรกมีค่า MSE สูงขึ้นเป็น 16.7989 และค่า  $R^2$  ลดลงเหลือ 0.9800 ดังแสดงในรูปที่ 4.9 และ 4.10 แสดงการเปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่ม การวิเคราะห์ดังกล่าวเน้นถึงประโยชน์ของการเลือกตัวแปรที่เหมาะสมเพื่อสร้างตัวแบบที่มีความสมดุลระหว่างความแม่นยำและความซับซ้อน

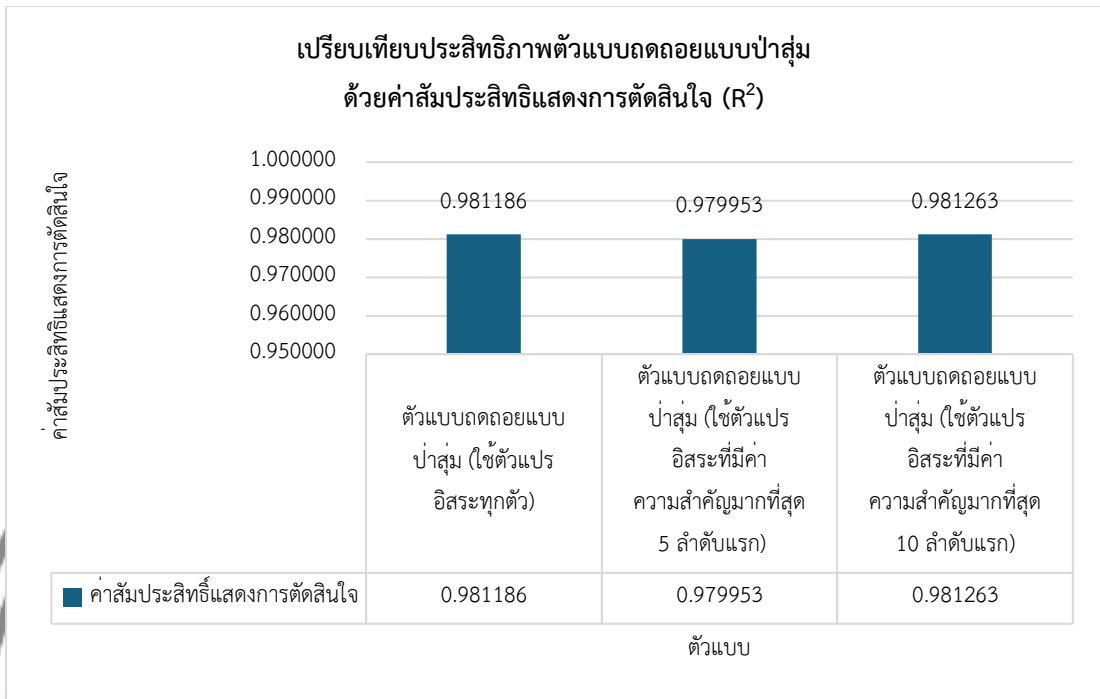
ตารางที่ 4.15 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่ม

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบป่าสุ่ม (ใช้ตัวแปรอิสระทุกตัว)	15.7650	0.9812
ตัวแบบถดถอยแบบป่าสุ่ม (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	16.7989	0.9800

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบป่าสุ่ม (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	15.7012	0.9813



รูปที่ 4.9 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่มด้วยค่า MSE



รูปที่ 4.10 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบป่าสุ่มด้วยค่า  $R^2$

#### 4.6 การสร้างตัวแบบถดถอยแบบเอ็กจีบิส

ในการสร้างตัวแบบถดถอยแบบเอ็กจีบิสแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบเอ็กจีบิส มีการทดลองสร้างตัวแบบ 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทุกตัว การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก โดยการคำนวณค่าความสำคัญของตัวแปรอิสระ ดำเนินการด้วยวิธี feature\_importances แบบ weight-based score เป็นการนับจำนวนครั้งที่ตัวแปร ถูกใช้ในการแบ่งข้อมูลในต้นไม้ของโมเดล ซึ่งผลลัพธ์ แสดงในตารางที่ 4.16 พบว่าตัวแปร BFW\_PUMP\_DISCHARGE\_PRESSURE มีค่าความสำคัญสูงสุด 0.9338 และตัวแปร BFW#1\_BEARING\_HOUSING\_VIBR\_A และ BFW#1\_BEARING\_HOUSING\_VIBR\_B มีค่าความสำคัญรองลงมาที่ 0.0433 และ 0.006 ตามลำดับ แต่ถือว่า BFW#1\_BEARING\_HOUSING\_VIBR\_B มีความสำคัญน้อยมากเมื่อเทียบกับ A ขณะที่ตัวแปรที่มีความสำคัญต่ำที่สุด ได้แก่ BFW#1\_MOTOR\_BEARING\_TEMP#A ซึ่งมีค่าความสำคัญ เท่ากับ 0.0008

ตารางที่ 4.16 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีเอ็กจิบูส

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	0.9338	1
BFW#1_BEARING_HOUSING_VIBR_A	0.0433	2
BFW#1_BEARING_HOUSING_VIBR_B	0.0060	3
GTG12_POWER	0.0023	4
BFW#1_MOTOR_WINDING_TEMP#A	0.0022	5
GTG11_POWER	0.0016	6
BFW#1_PUMP_BEARING_TEMP#B	0.0014	7
BFW#1_MOTOR_WINDING_TEMP#D	0.0013	8
BFW#1_PUMP_BEARING_TEMP#A	0.0012	9
BFW#1_MOTOR_WINDING_TEMP#B	0.0011	10
BFW#1_MOTOR_WINDING_TEMP#E	0.0011	11
BFW#1_MOTOR_WINDING_TEMP#F	0.0011	12
BFW#1_MOTOR_BEARING_TEMP#B	0.0010	13
BFW#1_PUMP_BEARING_TEMP#C	0.0009	14
BFW#1_MOTOR_WINDING_TEMP#C	0.0009	15
BFW#1_MOTOR_BEARING_TEMP#A	0.0008	16

การปรับแต่งพารามิเตอร์ของตัวแบบใช้วิธีการค้นหาแบบสุ่ม โดยพิจารณาค่า  $n\_estimators$  และ  $learning\_rate$  พร้อมใช้ข้อมูลฝึกหัดร่วมกับกระบวนการ K-fold cross-validation เพื่อลดความลำเอียงของตัวแบบ ผลการค้นหาพบว่า ค่าที่เหมาะสมที่สุดคือ  $n\_estimators = 200$  และ  $learning\_rate = 0.1$  ในทั้งสามกรณี ดังที่แสดงในตารางที่ 4.17

ตารางที่ 4.17 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบเอ็กจิบูส

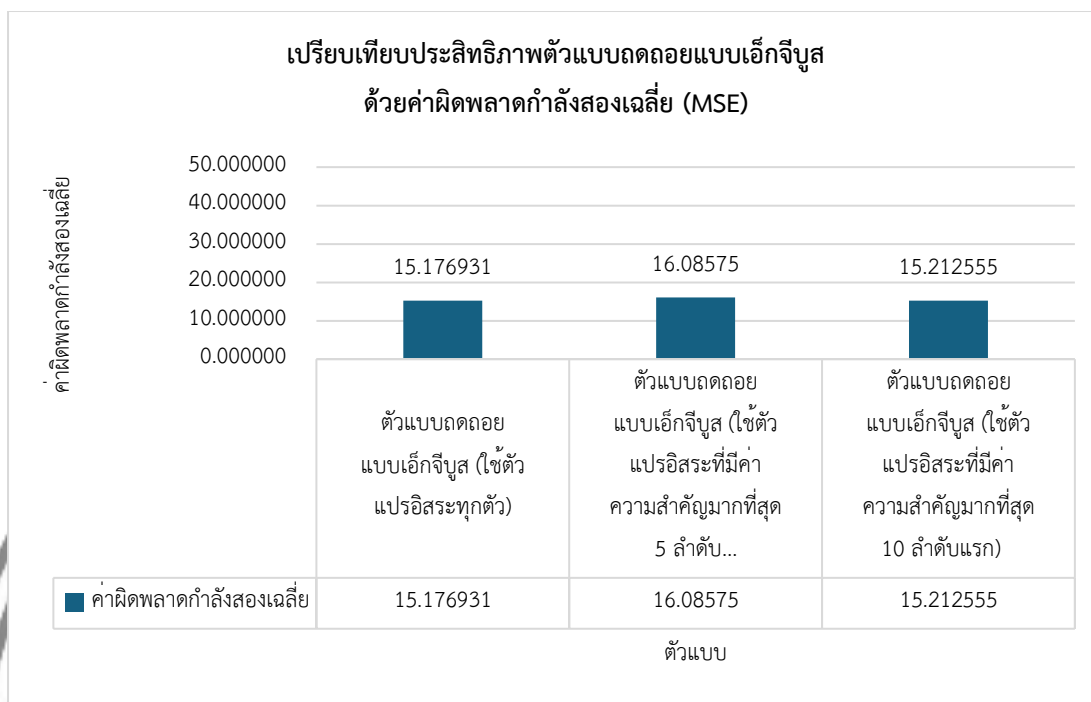
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ตัวแปรอิสระทุกตัว)	$n\_estimators$	100, 200	200
	$Learning\_rate$	0.01, 0.1	0.1

ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 5 ลำดับ แรก)	n_estimators	100, 200	200
	Learning_rate	0.01, 0.1	0.1
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 10 ลำดับแรก)	n_estimators	100, 200	200
	Learning_rate	0.01, 0.1	0.1

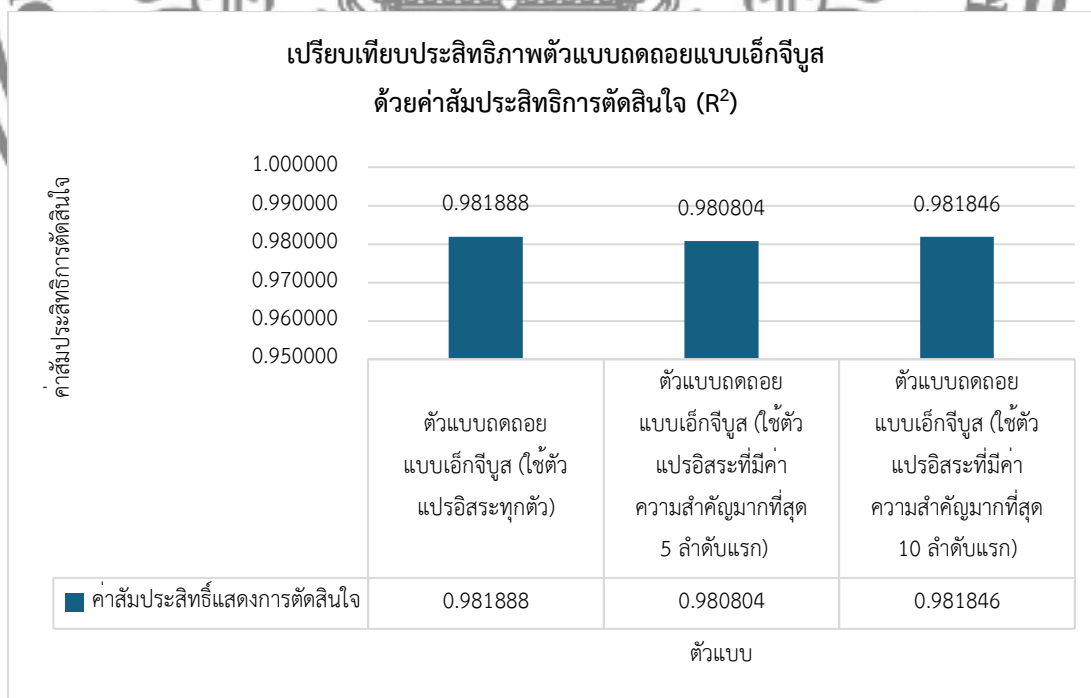
จากการเปรียบเทียบผลลัพธ์ในตารางที่ 4.18 พบว่า ตัวแบบที่ใช้ตัวแปรอิสระทุกตัวให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า MSE เท่ากับ 15.1769 และ  $R^2$  เท่ากับ 0.9819 ในกรณีตัวแบบที่ใช้ตัวแปรอิสระ 10 ลำดับแรกให้ผลลัพธ์ใกล้เคียงกันที่ MSE = 15.2126 และ  $R^2 = 0.9818$  การลดจำนวนตัวแปรเหลือเพียง 5 ลำดับแรกแม้จะลดความซับซ้อนของตัวแบบ แต่ทำให้ MSE เพิ่มขึ้นเป็น 16.0858 และ  $R^2$  ลดลงเล็กน้อยเป็น 0.9808 ดังแสดงในรูปที่ 4.11 และ 4.12 ซึ่งแสดงการเปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจิบูสด้วยค่า MSE และ  $R^2$  เน้นถึงความสำคัญของการเลือกตัวแปรที่ครอบคลุมเพื่อรักษาความแม่นยำสูงสุดในตัวแบบ

ตารางที่ 4.18 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจิบูส

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการ ตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ ตัวแปรอิสระทุกตัว)	15.1769	0.9819
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ ตัวแปรอิสระที่มีค่าความสำคัญ มากที่สุด 5 ลำดับแรก)	16.0858	0.9808
ตัวแบบถดถอยแบบเอ็กจิบูส (ใช้ ตัวแปรอิสระที่มีค่าความสำคัญ มากที่สุด 10 ลำดับแรก)	15.2126	0.9818



รูปที่ 4.11 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจิบูสด้วยค่า MSE



รูปที่ 4.12 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบเอ็กจิบูสด้วยค่า  $R^2$

#### 4.7 การสร้างตัวแบบถดถอยแบบแคทบูล

ในการสร้างตัวแบบถดถอยแบบแคทบูลแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบแคทบูล มีการพิจารณา 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทั้งหมด การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก การประเมินค่าความสำคัญของตัวแปรอิสระได้ดำเนินการโดยใช้เทคนิค feature\_importances ด้วยวิธี PredictionValuesChange เพื่อวิเคราะห์หีบหยาของตัวแปรแต่ละตัวว่าการเปลี่ยนแปลงค่าของตัวแปรนั้นจะส่งผลต่อการทำงานอย่างไรโดยเฉลี่ย หากค่าเฉลี่ยของการเปลี่ยนแปลงสูง แสดงว่าตัวแปรนั้นมีความสำคัญต่อการทำงาน ตารางที่ 4.19 แสดงให้เห็นว่า ตัวแปร BFW\_PUMP\_DISCHARGE\_PRESSURE มีค่าความสำคัญสูงสุดถึง 68.5997 ซึ่งบ่งชี้ถึงความสำคัญในการทำงานค่าพลังงานอย่างเด่นชัดขณะที่ ตัวแปร BFW#1\_MOTOR\_WINDING\_TEMP#E มีค่าความสำคัญต่ำสุดเพียง 0.1981 และแนวโน้มของกลุ่มตัวแปรอุณหภูมิของขดลวดมอเตอร์และตลับลูกปืน มีค่าความสำคัญต่ำซึ่งแสดงถึงผลกระทบที่น้อยต่อผลลัพธ์การทำงานการใช้พลังงาน

ตารางที่ 4.19 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีแคทบูล

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	68.5997	1
BFW#1_BEARING_HOUSING_VIBR_A	21.5584	2
BFW#1_BEARING_HOUSING_VIBR_B	2.5518	3
GTG12_POWER	1.0965	4
BFW#1_MOTOR_WINDING_TEMP#F	0.8663	5
BFW#1_PUMP_BEARING_TEMP#B	0.8560	6
GTG11_POWER	0.6864	7
BFW#1_MOTOR_WINDING_TEMP#C	0.6715	8
BFW#1_PUMP_BEARING_TEMP#A	0.6564	9
BFW#1_MOTOR_WINDING_TEMP#B	0.5735	10
BFW#1_MOTOR_WINDING_TEMP#A	0.5654	11

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW#1_PUMP_BEARING_TEMP#C	0.3277	12
BFW#1_MOTOR_BEARING_TEMP#B	0.2859	13
BFW#1_MOTOR_WINDING_TEMP#D	0.2784	14
BFW#1_MOTOR_BEARING_TEMP#A	0.2280	15
BFW#1_MOTOR_WINDING_TEMP#E	0.1981	16

ในการปรับแต่งพารามิเตอร์ ตัวแบบแคทพูลส์ใช้วิธีการค้นหาแบบสุ่ม โดยพิจารณา ค่าพารามิเตอร์  $n\_estimators$  และ  $learning\_rate$  ดังที่แสดงใน ตารางที่ 4.20 โดยค่าที่เหมาะสมที่สุดในทุกกรณีคือ  $n\_estimators = 200$  และ  $learning\_rate = 0.1$  การปรับแต่งนี้ดำเนินการร่วมกับเทคนิค K-fold Cross-validation บนชุดข้อมูลฝึกหัดเพื่อให้ตัวแบบมีความสามารถในการปรับตัวและลดปัญหาการเกิด overfitting

ตารางที่ 4.20 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบแคทพูลส์

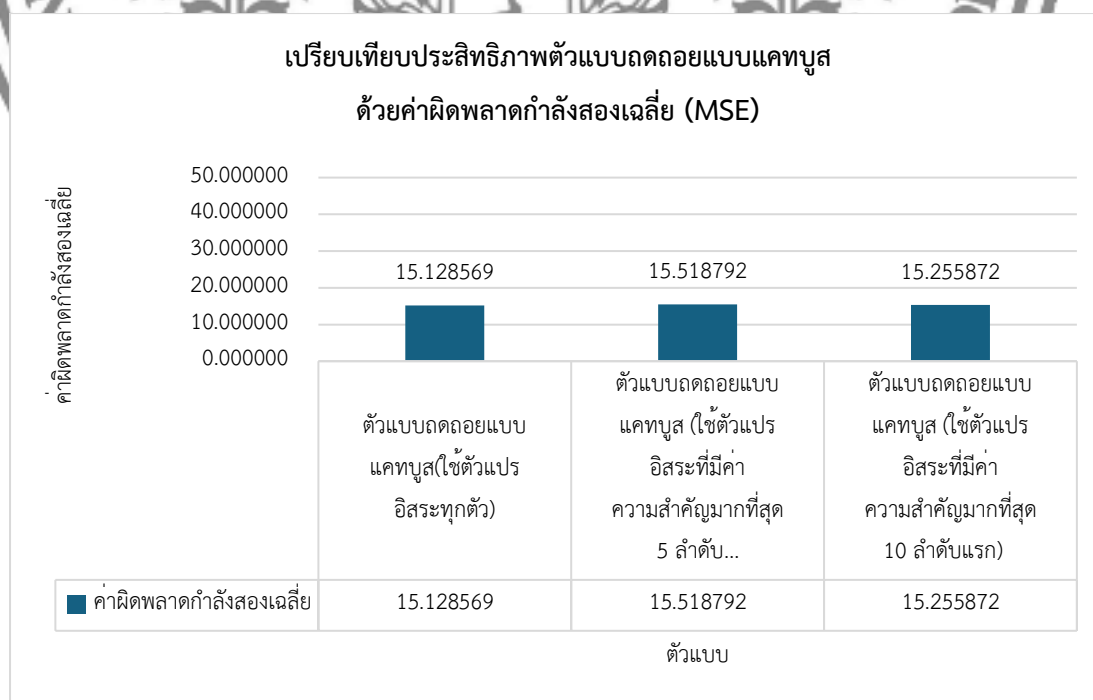
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบแคทพูลส์ (ใช้ตัวแปรอิสระทุกตัว)	$n\_estimators$	100, 200	200
	$Learning\_rate$	0.01, 0.1	0.1
ตัวแบบถดถอยแบบแคทพูลส์ (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 5 ลำดับ แรก)	$n\_estimators$	100, 200	200
	$Learning\_rate$	0.01, 0.1	0.1
ตัวแบบถดถอยแบบแคทพูลส์ (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 10 ลำดับแรก)	$n\_estimators$	100, 200	200
	$Learning\_rate$	0.01, 0.1	0.1

ผลการเปรียบเทียบประสิทธิภาพใน ตารางที่ 4.21 ซึ่งให้เห็นว่าตัวแบบที่ใช้ตัวแปรอิสระทั้งหมดให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า MSE เท่ากับ 15.1286 และ  $R^2$  เท่ากับ 0.9819 ขณะที่การลดจำนวนตัวแปรอิสระเหลือเพียง 10 ลำดับแรกให้ค่า MSE ใกล้เคียงกัน คือ 15.2559 และ  $R^2 = 0.9818$  ส่วนตัวแบบที่ใช้ 5 ตัวแปรสำคัญ มีค่า MSE เพิ่มขึ้นเป็น 15.5188 และ  $R^2$  ลดลงเป็น

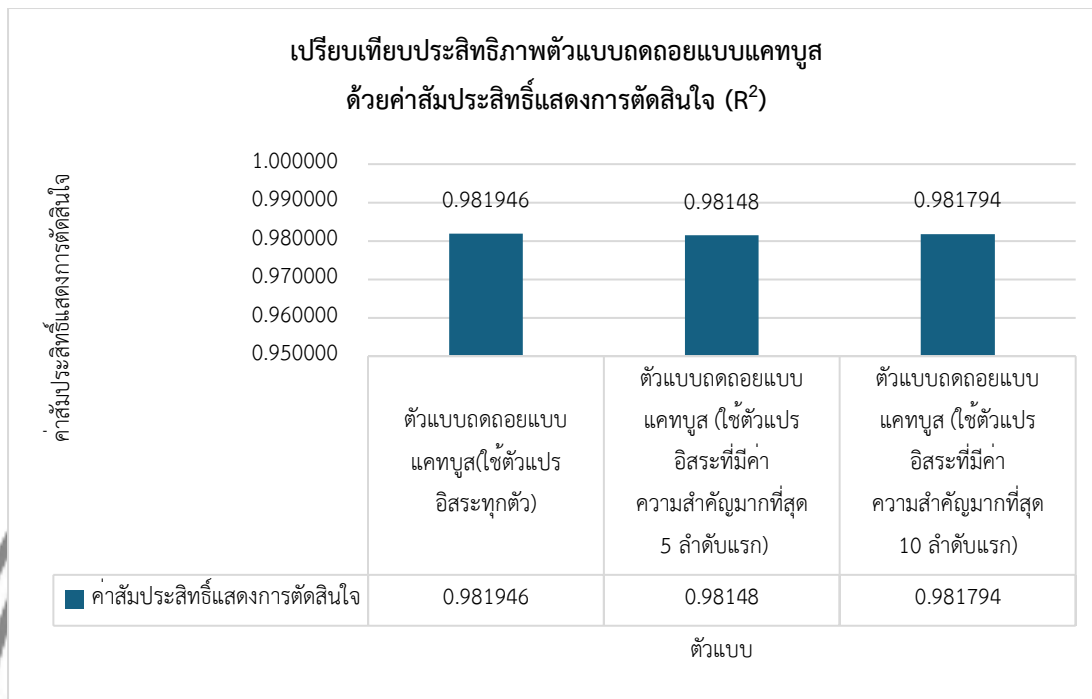
0.9815 ดังแสดงใน รูปที่ 4.13 และ 4.14 แสดงการเปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบ แคทบูลด้วยค่า MSE และ  $R^2$  ซึ่งช่วยยืนยันถึงความสำคัญของการเลือกตัวแปรอิสระที่เหมาะสมในการรักษาสมดุลระหว่างความซับซ้อนและความแม่นยำของตัวแบบ

ตารางที่ 4.21 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบแคทบูล

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดลิ้นใจ ( $R^2$ )
ตัวแบบถดถอยแบบแคทบูล(ใช้ตัวแปรอิสระทุกตัว)	15.1286	0.9819
ตัวแบบถดถอยแบบแคทบูล (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	15.5188	0.9815
ตัวแบบถดถอยแบบแคทบูล (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	15.2559	0.9818



รูปที่ 4.13 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบแคทบูลด้วยค่า MSE



รูปที่ 4.14 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบแคทบูลด้วยค่า  $R^2$

#### 4.7 การสร้างตัวแบบถดถอยแบบโลทิจิบีเอ็ม

ในการสร้างตัวแบบถดถอยแบบโลทิจิบีเอ็มแบ่งวิธีการสร้างตัวแบบ 3 กรณี ดังนี้

- ใช้ตัวแปรอิสระทุกตัวแปร
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก
- ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก

ในการสร้างตัวแบบถดถอยแบบโลทิจิบีเอ็ม ได้แบ่งกระบวนการสร้างตัวแบบออกเป็น 3 กรณี ได้แก่ การใช้ตัวแปรอิสระทุกตัว การใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก และการใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก การคัดเลือกตัวแปรอิสระดำเนินการโดยใช้เทคนิค Feature Importance ของโลทิจิบีเอ็ม ดังตารางที่ 4.22 พบว่าตัวแปร BFW\_PUMP\_DISCHARGE\_PRESSURE มีค่าความสำคัญสูงสุด เท่ากับ 1.3529 แสดงถึงความสัมพันธ์ที่เด่นชัดต่อการทำนายการใช้พลังงาน ขณะที่ตัวแปร BFW#1\_BEARING\_HOUSING\_VIBR\_A และ BFW#1\_BEARING\_HOUSING\_VIBR\_B มีความสำคัญรองลงมา ซึ่งมีค่าความสำคัญเท่ากับ 0.1585 และ 0.0234 ตามลำดับ ในขณะที่กลุ่มตัวแปรอุณหภูมิของขดลวดมอเตอร์ อย่างตัวแปร BFW#1\_MOTOR\_WINDING\_TEMP#A มีค่าความสำคัญต่ำสุด

เท่ากับ 0.0000 ซึ่งแสดงว่าตัวแปรอิสระของอนุกรมขดลวดจุด A ไม่มีผลกระทบต่อผลลัพธ์การทำนายการใช้พลังงาน

ตารางที่ 4.22 ค่าความสำคัญของตัวแปรอิสระด้วยวิธีไลท์จีบีเอ็ม

ตัวแปรอิสระ	ค่าความสำคัญ	ลำดับความสำคัญ
BFW_PUMP_DISCHARGE_PRESSURE	1.3529	1
BFW#1_BEARING_HOUSING_VIBR_A	0.1585	2
BFW#1_BEARING_HOUSING_VIBR_B	0.0234	3
BFW#1_PUMP_BEARING_TEMP#A	0.0033	4
BFW#1_MOTOR_WINDING_TEMP#A	0.0030	5
BFW#1_PUMP_BEARING_TEMP#B	0.0015	6
GTG12_POWER	0.0015	7
BFW#1_MOTOR_WINDING_TEMP#D	0.0010	8
GTG11_POWER	0.0006	9
BFW#1_MOTOR_WINDING_TEMP#F	0.0005	10
BFW#1_MOTOR_BEARING_TEMP#B	0.0004	11
BFW#1_MOTOR_WINDING_TEMP#E	0.0003	12
BFW#1_MOTOR_WINDING_TEMP#C	0.0003	13
BFW#1_MOTOR_WINDING_TEMP#B	0.0002	14
BFW#1_PUMP_BEARING_TEMP#C	0.0001	15
BFW#1_MOTOR_BEARING_TEMP#A	0.0000	16

การค้นหามิติเตอร์ใช้วิธีการค้นหาแบบสุ่ม ด้วยการวิธี K-fold cross-validation เพื่อลดความเอนเอียงของตัวแบบ และเพิ่มความสามารถในการคาดการณ์ค่าของชุดข้อมูลใหม่ ผลการค้นหาดังที่แสดงใน ตารางที่ 4.23 ค่าพารามิเตอร์ที่เหมาะสมที่สุดในทุกกรณีคือ  $n\_estimators = 200$  และ  $learning\_rate = 0.1$  ในทุกกรณี ซึ่งช่วยให้ตัวแบบสามารถรักษาความแม่นยำได้สูงสุดโดยไม่เกิด overfitting

ตารางที่ 4.23 เลือกพารามิเตอร์การค้นหาแบบสุ่มของตัวแบบถดถอยแบบโลทิจีบีเอ็ม

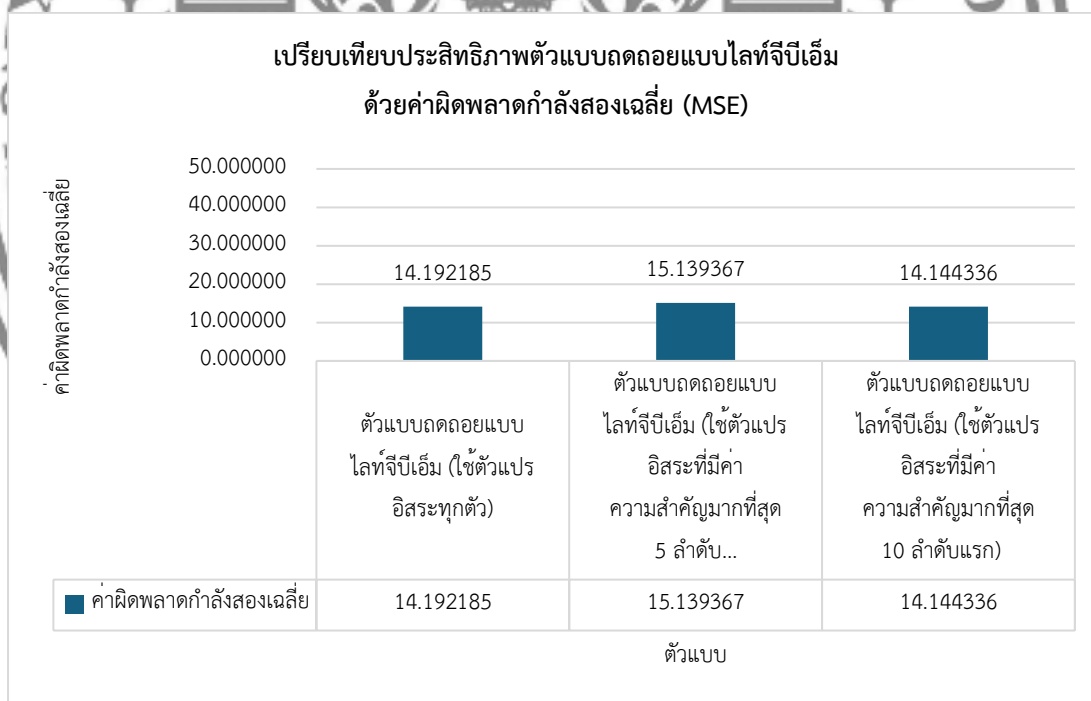
ตัวแบบ	พารามิเตอร์	ค่าค้นหาแบบสุ่ม	พารามิเตอร์ที่ดีที่สุด
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระทุกตัว)	n_estimators	100, 200	200
	Learning_rate	0.01, 0.1	0.1
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 5 ลำดับ แรก)	n_estimators	100, 200	200
	Learning_rate	0.01, 0.1	0.1
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระที่มีค่า ความสำคัญมากที่สุด 10 ลำดับแรก)	n_estimators	100, 200	200
	Learning_rate	0.01, 0.1	0.1

ผลการเปรียบเทียบประสิทธิภาพของตัวแบบจากตารางที่ 4.24 พบว่า ตัวแบบที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรกให้ผลลัพธ์ที่ดีที่สุด โดยมีค่า MSE เท่ากับ 14.1443 และค่า  $R^2$  เท่ากับ 0.9831 ซึ่งประสิทธิภาพตัวแบบใกล้เคียงกับตัวแบบที่ใช้ตัวแปรอิสระทุกตัว โดยมีค่า MSE เท่ากับ 14.1922 และค่า  $R^2$  เท่ากับ 0.9831 แสดงถึงการปรับเปลี่ยนจำนวนตัวแปรอิสระที่เหมาะสมไม่ส่งผลกระทบต่อความแม่นยำมากนัก ขณะที่การลดตัวแปรอิสระเหลือเพียง 5 ตัวแปรที่สำคัญที่สุด ส่งผลให้ค่า MSE เพิ่มขึ้นเป็น 15.1394 และ  $R^2$  ลดลงเป็น 0.9819 ดังแสดงในรูปที่ 4.15 และ 4.16 ซึ่งแสดงการเปรียบเทียบผลลัพธ์ของตัวแบบในเชิงประสิทธิภาพ โดยเน้นถึงความสำคัญของการเลือกตัวแปรที่ครอบคลุมเพื่อรักษาสมดุลระหว่างความแม่นยำและความซับซ้อนของตัวแบบ

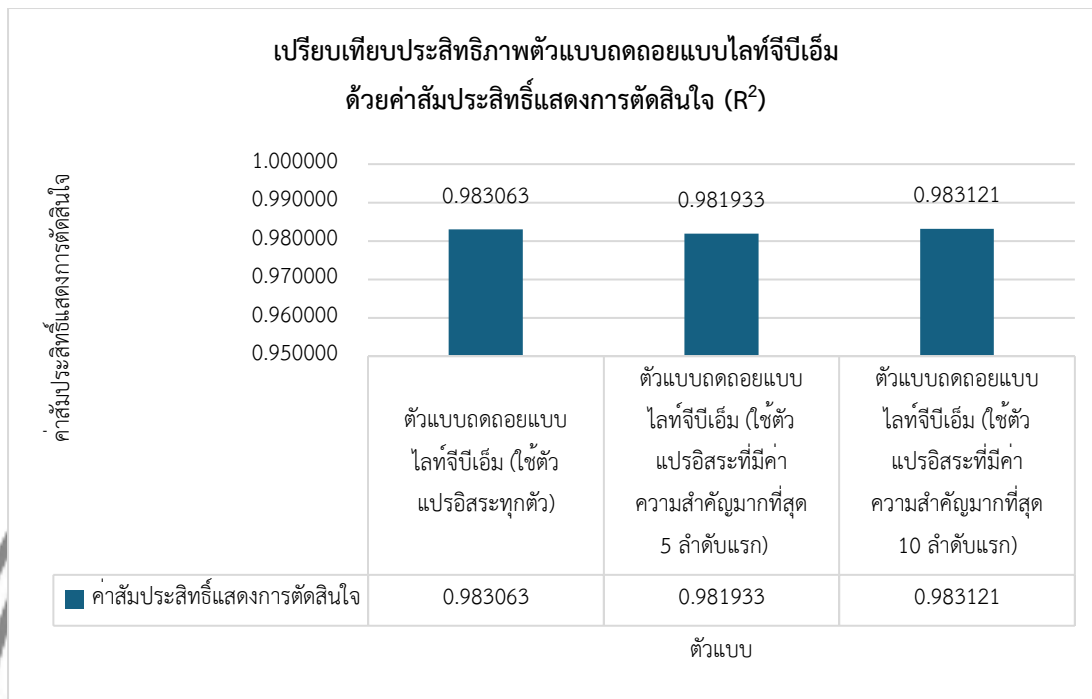
ตารางที่ 4.24 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบถดถอยแบบโลทิจีบีเอ็ม

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระทุกตัว)	14.1922	0.9831

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก)	15.1394	0.9819
ตัวแบบถดถอยแบบโลทิจีบีเอ็ม (ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก)	14.1443	0.9831



รูปที่ 4.15 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบโลทิจีบีเอ็มด้วยค่า MSE



รูปที่ 4.16 เปรียบเทียบประสิทธิภาพตัวแบบถดถอยแบบโลทิจีบีเอ็มด้วยค่า  $R^2$

#### 4.8 การสร้างตัวแบบโครงข่ายประสาทเทียม

ในการสร้างตัวแบบการเรียนรู้เชิงลึกแบ่งวิธีการสร้างตัวแบบ 4 รูปแบบ ดังนี้

- ตัวแบบโครงข่ายประสาทเทียมเชิงลึก
- ตัวแบบโครงข่ายประสาทเทียมวงกกลับ
- ตัวแบบโครงข่ายประสาทเทียมแบบ GRU
- ตัวแบบโครงข่ายประสาทเทียมแบบ LSTM

ในการสร้างตัวแบบโครงข่ายประสาทเทียม (Neural Network) สำหรับการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ ได้ทำการทดสอบ 4 รูปแบบหลัก ได้แก่ โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network: DNN), โครงข่ายประสาทเทียมวงกกลับ (Recurrent Neural Network: RNN), โครงข่ายประสาทเทียมแบบ GRU (Gated Recurrent Unit: GRU) และโครงข่ายประสาทเทียมแบบ LSTM (Long Short-Term Memory: LSTM) ทั้งหมดใช้โครงสร้างพื้นฐานคล้ายกัน เพื่อให้สามารถเปรียบเทียบประสิทธิภาพได้อย่างเป็นระบบ การปรับตั้งค่าพารามิเตอร์ของตัวแบบ ทุกตัวแบบใช้สถาปัตยกรรมเครือข่ายที่มีจำนวน 128 หน่วยประสาท ในชั้นแรก 64 หน่วยประสาท ในชั้นที่สอง และ 1 หน่วยประสาท สำหรับเอาต์พุต โดยใช้ฟังก์ชันการกระตุ้นแบบ ReLU (Rectified Linear Unit) ในชั้นซ่อน และใช้ตัวปรับค่า Adam Optimizer ที่มีอัตราการเรียนรู้

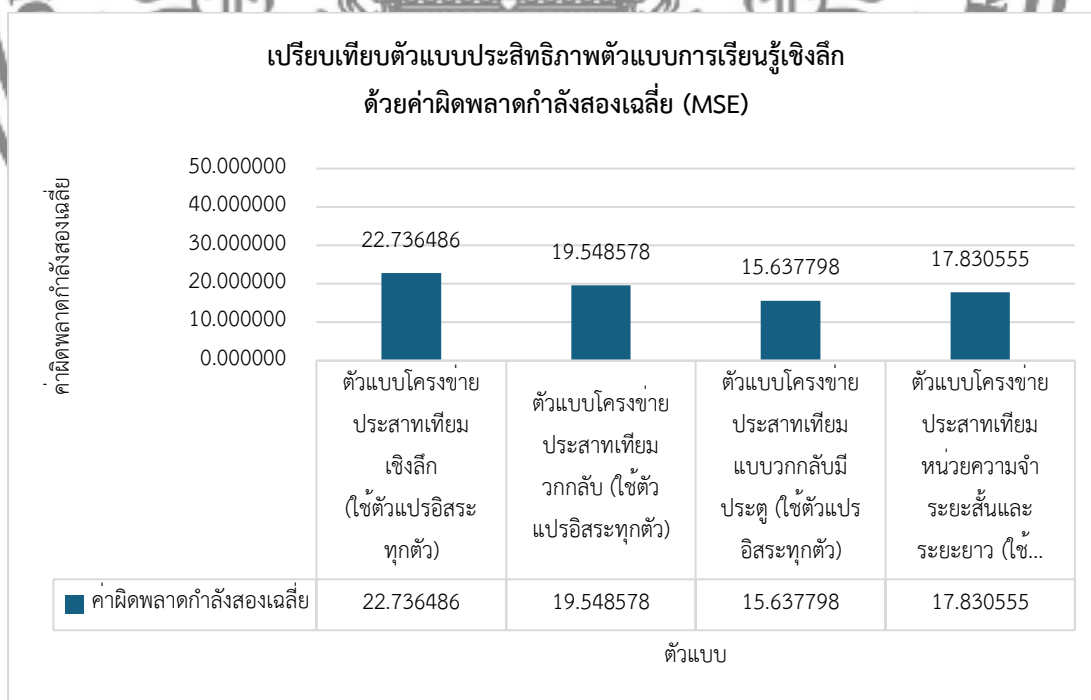
(Learning Rate) เท่ากับ 0.001 ตัวแบบถูกฝึกอบรมด้วย 100 Epochs และ Batch Size ขนาด 32 พร้อมชุดข้อมูลที่ถูกปรับขนาด (Scaled Data) เพื่อช่วยเพิ่มประสิทธิภาพในการเรียนรู้

ตัวแบบ DNN ใช้การเชื่อมต่อแบบ Feedforward โดยไม่มีการจัดการข้อมูลตามลำดับเวลา โครงสร้างของตัวแบบเป็นลำดับของชั้น Dense Layers ที่เรียนรู้ความสัมพันธ์ของตัวแปรอิสระเพื่อทำนายค่าตัวแปรตาม ผลลัพธ์จากตารางที่ 4.25 พบว่าตัวแบบ DNN ให้ค่า MSE เท่ากับ 22.7365 และค่า  $R^2$  เท่ากับ 0.9729 ในกรณีตัวแบบ RNN ถูกออกแบบมาให้สามารถใช้ข้อมูลในอดีตเพื่อทำนายค่าปัจจุบันได้ โดยใช้โครงสร้างแบบวนซ้ำ (Recurrent) ทำให้สามารถเก็บความสัมพันธ์เชิงเวลาได้ดีขึ้น แม้ว่าจะมีข้อจำกัดเรื่อง Vanishing Gradient ในบางกรณี ผลการทดสอบพบว่า RNN มีค่า MSE เท่ากับ 19.5486 และค่า  $R^2$  เท่ากับ 0.9767 ซึ่งแสดงให้เห็นว่าการใช้ข้อมูลลำดับเวลากับตัวแบบช่วยให้ผลลัพธ์แม่นยำขึ้นกว่าแบบดั้งเดิม กรณีตัวแบบ GRU เป็นเทคนิคที่พัฒนาต่อยอดจาก RNN โดยใช้โครงสร้างเกตควบคุม ซึ่งช่วยลดปัญหา Vanishing Gradient และทำให้ตัวแบบสามารถเรียนรู้ข้อมูลลำดับเวลาที่มีความยาวได้ดีขึ้น ในการทดลองพบว่า GRU ให้ผลลัพธ์ที่ดีที่สุดในกลุ่มตัวแบบทั้งหมด โดยมีค่า MSE เท่ากับ 15.6378 และค่า  $R^2$  เท่ากับ 0.9813 ตัวแบบ LSTM เป็นอีกรูปแบบหนึ่งของ RNN ที่ออกแบบมาเพื่อเก็บรักษาความสัมพันธ์ของข้อมูลในลำดับเวลาที่ยาวขึ้น โดยใช้โครงสร้างสามเกต ได้แก่ Forget Gate, Input Gate และ Output Gate ซึ่งช่วยควบคุมการจับเก็บและลืมข้อมูลในแต่ละช่วงเวลา อย่างไรก็ตาม เนื่องจากโครงสร้างของ LSTM มีความซับซ้อนกว่า GRU ทำให้ใช้เวลาฝึกนานขึ้น ในการทดลองพบว่า LSTM มีค่า MSE เท่ากับ 17.8306 และค่า  $R^2$  เท่ากับ 0.9787 ซึ่งแม้ว่าจะมีความแม่นยำสูง แต่ยังเป็นรอง GRU

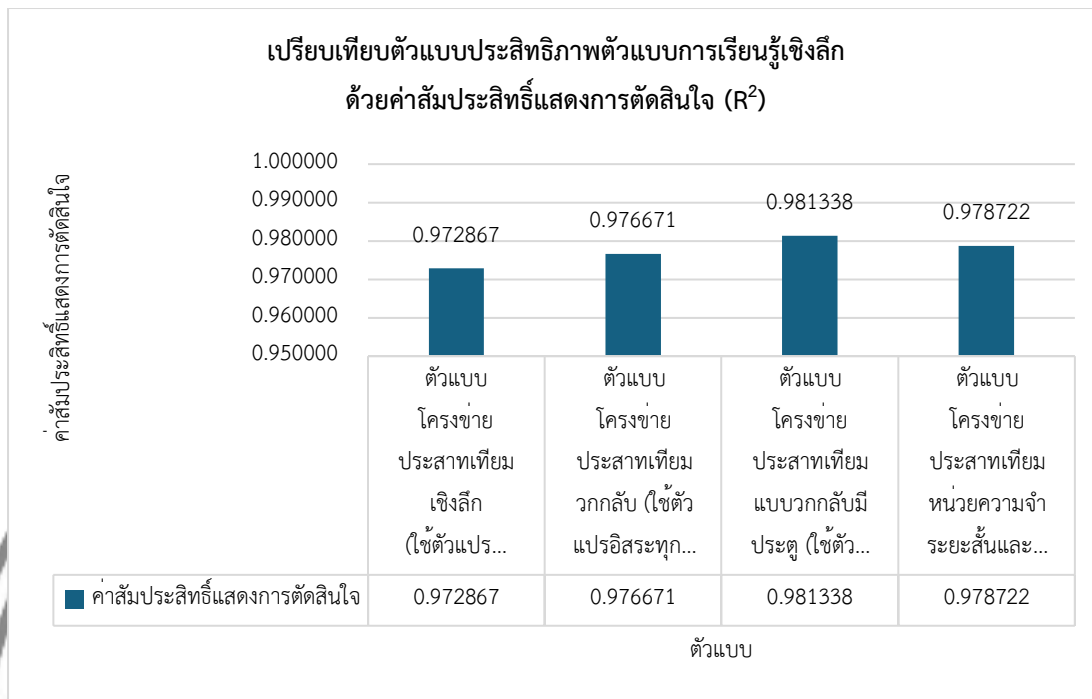
จาก ตารางที่ 4.25 และ รูปที่ 4.17 และ 4.18 พบว่าตัวแบบที่ใช้โครงสร้างแบบ Sequence Models อย่าง GRU และ LSTM สามารถจับความสัมพันธ์ของข้อมูลได้ดีกว่า DNN และ RNN โดย GRU ให้ผลลัพธ์ที่ดีที่สุด เนื่องจากสามารถเรียนรู้ลำดับข้อมูลได้ดีโดยใช้โครงสร้างที่เรียบง่ายกว่า LSTM ในขณะที่ LSTM แม้จะสามารถเรียนรู้ข้อมูลระยะยาวได้ดี แต่มีความซับซ้อนสูงและต้องใช้ทรัพยากรมากกว่า ในขณะที่ DNN แม้จะสามารถจับความสัมพันธ์เชิงซับซ้อนได้ดี แต่ไม่ได้ออกแบบมาให้รองรับข้อมูลลำดับเวลาโดยตรง

ตารางที่ 4.25 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึก

ตัวแบบ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
	ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
ตัวแบบโครงข่ายประสาทเทียมเชิงลึก (ใช้ตัวแปรอิสระทุกตัว)	22.7365	0.9729
ตัวแบบโครงข่ายประสาทเทียมวงกลับ (ใช้ตัวแปรอิสระทุกตัว)	19.5486	0.9767
ตัวแบบโครงข่ายประสาทเทียมแบบวงกลับมีประตู (ใช้ตัวแปรอิสระทุกตัว)	15.6378	0.9813
ตัวแบบโครงข่ายประสาทเทียมหน่วยความจำระยะสั้นและระยะยาว (ใช้ตัวแปรอิสระทุกตัว)	17.8306	0.9787



รูปที่ 4.17 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึกด้วยค่า MSE



รูปที่ 4.18 เปรียบเทียบตัวแบบประสิทธิภาพตัวแบบการเรียนรู้เชิงลึกด้วยค่า  $R^2$

## 4.9 การอภิปรายผลการวิจัย

### 4.9.1 ตัวแบบที่มีประสิทธิภาพมากที่สุด

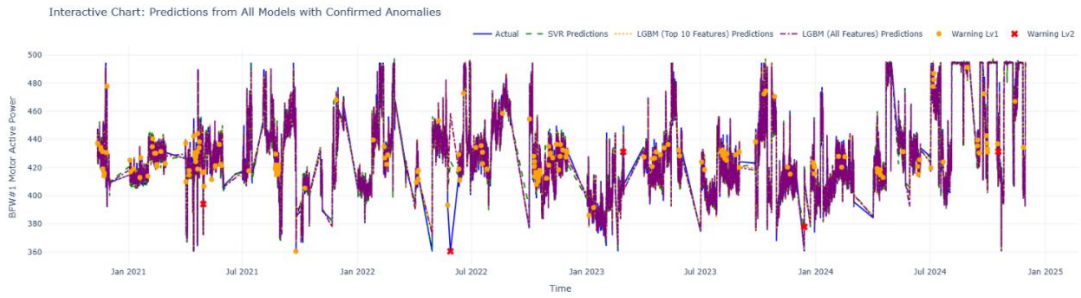
จากผลการเปรียบเทียบประสิทธิภาพการสร้างตัวแบบการเรียนรู้ ในตารางที่ 4.26 พบว่า ตัวแบบที่มีประสิทธิภาพสูงสุดในการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ 3 ลำดับแรก ได้แก่ ตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน ที่ใช้ตัวแปรอิสระทุกตัว , ตัวแบบการถดถอยแบบโลทจีบีเอ็มที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก , ตัวแบบการถดถอยแบบโลทจีบีเอ็มที่ใช้ตัวแปรอิสระทุกตัว ตามลำดับ โดยตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีนที่ใช้ตัวแปรอิสระทุกตัว มีค่า MSE ต่ำที่สุด เท่ากับ 13.5573 และค่า  $R^2$  สูงที่สุด เท่ากับ 0.9838 ซึ่งแสดงถึงความสามารถในการทำนายค่าพลังงานได้อย่างแม่นยำ ในขณะที่ ตัวแบบการถดถอยแบบโลทจีบีเอ็มที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก และ ตัวแบบการถดถอยแบบโลทจีบีเอ็มที่ใช้ตัวแปรอิสระทุกตัว มีค่าความผิดพลาดที่ใกล้เคียงกันที่ MSE เท่ากับ 14.1443 และ 14.1922 ตามลำดับ และค่า  $R^2$  เท่ากันคือ 0.9831 ซึ่งหมายความว่าทั้งสามตัวแบบมีประสิทธิภาพและความแม่นยำสูงและสามารถใช้งานได้ในบริบทของการทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำได้เป็นอย่างดี

#### 4.9.2 ตัวแปรที่ส่งผลต่อตัวแบบมากที่สุด

การวิเคราะห์ค่าความสำคัญของตัวแปรในแต่ละตัวแบบ พบว่า ตัวแปร BFW\_PUMP\_DISCHARGE\_PRESSURE หรือ แรงดันน้ำออกจากเครื่องสูบน้ำ มีค่าความสำคัญสูงสุดในทุกตัวแบบ แสดงถึงความสัมพันธ์โดยตรงกับการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ นอกจากนี้ในกลุ่มของตัวแปรค่าสั่นสะเทือนของตัวเรือนตลับลูกปืน ได้แก่ ตัวแปร BFW#1\_BEARING\_HOUSING\_VIBR\_A และ BFW#1\_BEARING\_HOUSING\_VIBR\_B มีค่าความสำคัญรองลงมาในหลายตัวกลุ่มแบบ ได้แก่ กลุ่มตัวแบบการถอดแบบเคอร์เนลเชิงเส้น กลุ่มตัวแบบการถอดแบบต้นไม้ตัดสินใจ และ กลุ่มตัวแบบการถอดแบบอองเซมเบล ซึ่งหมายความว่ากลุ่มของตัวแปรค่าสั่นสะเทือนของตัวเรือนตลับลูกปืน มีผลต่อทำนายการใช้พลังงาน ส่วนตัวแปรที่มีค่าความสำคัญต่ำสุด ได้แก่ กลุ่มตัวแปรอุณหภูมิของชุดมอเตอร์ ซึ่งมีอิทธิพลน้อยต่อการทำนายการใช้พลังงาน โดยสอดคล้องกันในหลายกลุ่มตัวแบบ

#### 4.9.3 การประยุกต์ตัวแบบตรวจจับความผิดปกติและระบบส่งอีเมลแจ้งเตือน

การตรวจจับความผิดปกติของการใช้พลังงานในระบบเครื่องสูบน้ำป้อนหม้อไอน้ำเป็นสิ่งสำคัญในการบริหารจัดการพลังงานอย่างมีประสิทธิภาพ เพื่อลดความสูญเสียและป้องกันความเสียหายที่อาจเกิดขึ้น ตัวแบบที่มีประสิทธิภาพสูงสุด 3 ลำดับแรก ถูกนำมาใช้ร่วมกันเพื่อตรวจจับความผิดปกติของพลังงานโดยใช้เทคนิค Interquartile Range (IQR) Method ในการคำนวณขอบเขตของค่าปกติและกำหนดช่วงขีดจำกัดสำหรับการตรวจจับค่าผิดปกติ แบ่งออกเป็น 2 ระดับ ได้แก่ ระดับเตือนที่ 1 และ ระดับเตือนที่ 2 พบว่า ช่วงขีดจำกัดสำหรับการตรวจจับค่าผิดปกติของตัวแบบการถอดแบบซัพพอร์ตเวกเตอร์แมชชีน ที่ใช้ตัวแปรอิสระทุกตัว เท่ากับ  $(-7.5079, +7.6733)$  และ  $(-13.2008, +13.3662)$  ตัวแบบการถอดแบบโลทิสติกที่ใช้ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก เท่ากับ  $(-7.5156, +7.5226)$  และ  $(-13.1550, +13.1620)$  ตัวแบบการถอดแบบโลทิสติกที่ใช้ตัวแปรอิสระทุกตัว เท่ากับ  $(-7.4368, +7.4404)$  และ  $(-13.0158, +13.0193)$  สำหรับ ระดับเตือนที่ 1 และ ระดับเตือนที่ 2 ตามลำดับ ระบบจะพิจารณาว่าหากค่าความผิดปกติของทั้งสามตัวแบบตรวจจับได้ในระดับเดียวกัน จะถือว่าเป็น ความผิดปกติที่ได้รับการยืนยัน ซึ่งพบว่าตรวจพบ ระดับเตือนที่ 1 จำนวน 183 เหตุการณ์ และ ระดับเตือนที่ 2 จำนวน 5 เหตุการณ์ ดังรูปที่ 4.19 ซึ่งสามารถนำไปใช้เป็นเกณฑ์แจ้งเตือนและวางแผนการตรวจเช็คซ่อมบำรุงล่วงหน้าเพื่อลดความเสี่ยงที่อาจเกิดขึ้น

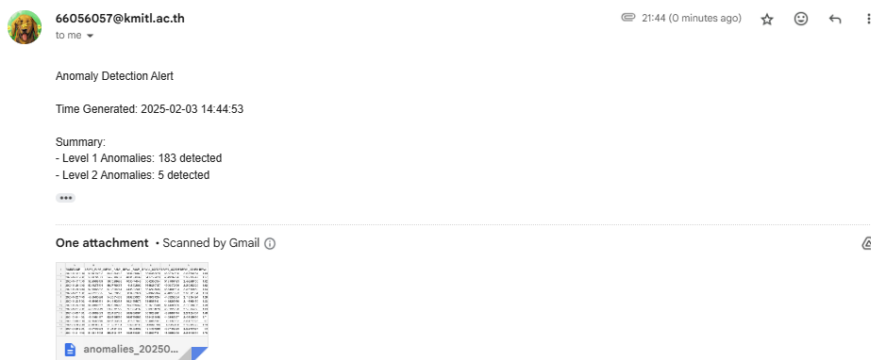


รูปที่ 4.19 ประยุกต์ตัวแบบตรวจจับความผิดปกติของการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ



รูปที่ 4.20 กระบวนการยืนยันการตรวจจับความผิดปกติด้วยตัวแบบการทำนายทั้งสามตัวแบบ

สำหรับระบบแจ้งเตือน เมื่อมีการตรวจพบ ระดับเตือนที่ 1 หรือ ระดับเตือนที่ 2 ระบบจะทำการบันทึกค่าความผิดปกติลงไฟล์ CSV และแนบไปกับอีเมลแจ้งเตือนโดยอัตโนมัติ ระบบดังกล่าวช่วยให้สามารถเฝ้าระวังสถานะการทำงานของเครื่องจักรได้แบบเรียลไทม์ และช่วยให้ผู้ดูแลสามารถดำเนินการมาตรการป้องกันได้อย่างทันท่วงที ซึ่งเป็นประโยชน์ต่อการลดการสูญเสียพลังงาน และช่วยให้กระบวนการทำงานของเครื่องสูบน้ำป้อนหม้อไอน้ำมีความเสถียรมากขึ้น



รูปที่ 4.21 การแจ้งเตือนเหตุการณ์ผิดปกติของการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ

การนำระบบนี้ไปใช้งานช่วยให้สามารถ คาดการณ์การใช้พลังงานและตรวจจับค่าความผิดปกติได้ล่วงหน้าของเครื่องสูบน้ำป้อนหม้อไอน้ำ ซึ่งเป็นปัจจัยสำคัญในการ บริหารจัดการพลังงานของเครื่องจักรให้มีเสถียรภาพ ลดต้นทุนจากการใช้พลังงานที่ไม่จำเป็น และช่วยให้สามารถวางแผนบำรุงรักษาเชิงคาดการณ์ (Predictive Maintenance) ได้อย่างมีประสิทธิภาพมากขึ้น



ตารางที่ 4.26 เปรียบเทียบประสิทธิภาพการสร้างตัวแบบการเรียนรู้

กลุ่มตัวแบบ	ตัวแบบ	การวัดประสิทธิภาพด้วยชุดข้อมูลทดสอบ	การวัดประสิทธิภาพด้วยชุดข้อมูลทดสอบ	
			ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ (R <sup>2</sup> )
กลุ่มตัวแบบการถดถอยเชิงเส้น	ตัวแปรอิสระ	ตัวแปรอิสระทุกตัว	23.3543	0.9721
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	23.3749	0.9721
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	36.4519	0.9565
กลุ่มตัวแบบการถดถอยเชิงเส้นแบบแรก	ตัวแปรอิสระ	ตัวแปรอิสระทุกตัว	23.3730	0.9721
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	23.3543	0.9721
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	23.3810	0.9721
กลุ่มตัวแบบการถดถอยเชิงเส้นแบบคอนโวลูชัน	ตัวแปรอิสระ	ตัวแปรอิสระทุกตัว	13.5573	0.9838
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	20.8288	0.9751
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	14.2818	0.9830
กลุ่มตัวแบบการถดถอยเชิงเส้นแบบไม่มีตัดต้นใจ	ตัวแปรอิสระ	ตัวแปรอิสระทุกตัว	21.4189	0.9744
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 5 ลำดับแรก	21.3150	0.9746
	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	ตัวแปรอิสระที่มีความสำคัญมากที่สุด 10 ลำดับแรก	21.0988	0.9748

กลุ่มตัวแบบ	ตัวแบบ	ตัวแปรอิสระ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
			ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
กลุ่มตัวแบบการถดถอยแบบอองเซนเบ็ค	ตัวแบบการถดถอยแบบป่าสุ่ม	ตัวแปรอิสระทุกตัว	15.7650	0.9812
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก	16.7989	0.9800
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก	15.7012	0.9813
		ตัวแปรอิสระทุกตัว	15.1769	0.9819
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก	16.0858	0.9808
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก	15.2126	0.9818
		ตัวแปรอิสระทุกตัว	15.1286	0.9819
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก	15.5188	0.9815
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก	15.2559	0.9818
		ตัวแปรอิสระทุกตัว	14.1922	0.9831
กลุ่มตัวแบบโครงข่ายประสาทเทียม	โครงข่ายประสาทเทียมเชิงลึก	ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 5 ลำดับแรก	15.1394	0.9819
		ตัวแปรอิสระที่มีค่าความสำคัญมากที่สุด 10 ลำดับแรก	14.1443	0.9831
กลุ่มตัวแบบโครงข่ายประสาทเทียม	โครงข่ายประสาทเทียมวงกลับ	ตัวแปรอิสระทุกตัว	22.7365	0.9729
		ตัวแปรอิสระทุกตัว	19.5486	0.9767

กลุ่มตัวแบบ	ตัวแบบ	ตัวแปรอิสระ	การวัดประสิทธิภาพตัวแบบด้วยชุดข้อมูลทดสอบ	
			ค่าผิดพลาดกำลังสองเฉลี่ย (MSE)	ค่าสัมประสิทธิ์แสดงการตัดสินใจ ( $R^2$ )
	โครงข่ายประสาทเทียมแบบ GRU	ตัวแปรอิสระทุกตัว	15.6378	0.9813
	โครงข่ายประสาทเทียมแบบ LSTM	ตัวแปรอิสระทุกตัว	17.8306	0.9787





## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้มุ่งเน้นพัฒนาและเปรียบเทียบตัวแบบการทำนายการใช้พลังงานของเครื่องสูบน้ำ ป้อนหม้อไอน้ำ โดยใช้เทคนิคการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก เพื่อเพิ่มประสิทธิภาพในการทำนายและสามารถนำไปประยุกต์ใช้ในการบริหารจัดการพลังงานเชิงอุตสาหกรรมและตรวจจับความผิดปกติของเครื่องสูบน้ำป้อนหม้อไอน้ำล่วงหน้าได้อย่างมีประสิทธิภาพ การวิจัยได้ดำเนินการทดลองและวิเคราะห์เปรียบเทียบตัวแบบต่าง ๆ เพื่อระบุแนวทางที่เหมาะสมสำหรับการนำไปใช้งานจริง ในบทนี้จะนำเสนอข้อสรุปของผลการวิจัย และข้อเสนอแนะสำหรับงานวิจัยในอนาคต

#### 5.1 สรุปผลการวิจัย

จากผลการวิจัยพบว่า ตัวแบบการเรียนรู้ของเครื่องและการเรียนรู้เชิงลึกที่พัฒนาขึ้นสามารถใช้ทำนายค่าการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำได้อย่างแม่นยำ จากการเปรียบเทียบค่าความผิดพลาดกำลังสองเฉลี่ย และค่าสัมประสิทธิ์การตัดสินใจ พบว่า ตัวแบบการถดถอยแบบซัพพอร์ตเวกเตอร์แมชชีน มีประสิทธิภาพและความแม่นยำในการทำนายการใช้พลังงานสูงสุด การคัดเลือกตัวแปรอิสระที่เหมาะสมโดยอาศัยการทดสอบสมมติฐานทางสถิติและการประเมินค่าความสำคัญของตัวแปรอิสระ พบว่า "แรงดันน้ำออกจากเครื่องสูบน้ำ" และ "ค่าสิ้นสะสมของตัวเรือนต้กลูกปืน" เป็นปัจจัยที่มีอิทธิพลสูงสุดต่อค่าการใช้พลังงาน ซึ่งแสดงให้เห็นถึงความสัมพันธ์ที่สำคัญระหว่างตัวแปรเหล่านี้กับปริมาณพลังงานที่ใช้ในระบบหม้อไอน้ำ การเลือกพารามิเตอร์โดยใช้การค้นหาแบบสุ่ม ช่วยเพิ่มความแม่นยำของตัวแบบและลดความซับซ้อนที่ไม่จำเป็น นอกจากนี้ ยังได้ประยุกต์ใช้เทคนิคการตรวจจับความผิดปกติร่วมกับตัวแบบที่มีประสิทธิภาพในการทำนายที่แม่นยำ เพื่อระบุสถานะที่อาจส่งผลกระทบต่อประสิทธิภาพการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำ อีกทั้งยังได้พัฒนาระบบแจ้งเตือนผ่านอีเมลเพื่อให้สามารถเฝ้าระวังและจัดการกับความผิดปกติของพลังงานได้แบบเรียลไทม์ ช่วยลดความเสี่ยงจากการใช้พลังงานที่ผิดปกติ และสนับสนุนการบำรุงรักษาเชิงพยากรณ์ให้มีประสิทธิภาพมากขึ้น

#### 5.2 ข้อจำกัดการวิจัย

- 1) ข้อจำกัดด้านข้อมูลและการประยุกต์ใช้งานกับโรงไฟฟ้าอื่นในเครือ

แม้ว่าข้อมูลที่ใช้ในการพัฒนาตัวแบบจะครอบคลุมช่วงเวลา 5 ปี แต่เป็นข้อมูลที่ได้จากโรงไฟฟ้าเพียงแห่งเดียวเท่านั้น จึงอาจไม่สามารถสะท้อนลักษณะการทำงานของเครื่องสูบน้ำป้อนหม้อไอน้ำภายใต้สภาพแวดล้อมและเงื่อนไขการดำเนินงานที่แตกต่างกันในโรงไฟฟ้าประเภทอื่นได้

อย่างครบถ้วน เพื่อยืนยันความถูกต้องและเพิ่มความแม่นยำของตัวแบบที่พัฒนาขึ้น จึงควรมีการนำตัวแบบไปทดสอบกับเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้าอื่นๆ ภายในเครือทั้ง 11 แห่ง โดยเปรียบเทียบผลการทำนายกับข้อมูลการใช้พลังงานจริงในแต่ละโรงไฟฟ้า ซึ่งจะช่วยประเมินประสิทธิภาพของตัวแบบภายใต้บริบทที่หลากหลาย และสามารถปรับแต่งพารามิเตอร์ให้เหมาะสมกับลักษณะเฉพาะของแต่ละโรงไฟฟ้าได้อย่างเหมาะสม ส่งผลให้สามารถขยายผลการใช้งานในเชิงปฏิบัติได้อย่างครอบคลุมและมีประสิทธิภาพยิ่งขึ้นในอนาคต

2) การทดสอบระบบตรวจจับความผิดปกติภายใต้สถานการณ์เหตุการณ์เครื่องจักรเสียหาย เครื่องสูบน้ำป้อนหม้อไอน้ำที่ใช้ในการศึกษาฯ ยังไม่มีประวัติการใช้งานจนถึงขั้นเกิดเหตุการณ์เสียหาย ทำให้ไม่สามารถยืนยันประสิทธิภาพของตัวแบบในตรวจจับความผิดปกติได้อย่างสมบูรณ์ เพื่อประเมินความสามารถระบุความผิดปกติที่สำคัญได้อย่างถูกต้อง

### 5.3 ข้อเสนอแนะ

1) ในการประยุกต์ใช้ตัวแบบที่พัฒนาขึ้นกับชุดข้อมูลอื่น

หากพบว่าประสิทธิภาพของการทำงานลดลง อาจพิจารณาปรับปรุงตัวแบบเพิ่มเติมโดยการเลือกตัวแปรอิสระที่เหมาะสมมากยิ่งขึ้น รวมถึงการปรับแต่งค่าพารามิเตอร์ของตัวแบบอย่างเป็นระบบ โดยสามารถเพิ่มการประยุกต์ใช้เทคนิคการปรับแต่งพารามิเตอร์อย่างละเอียด เช่น Bayesian Optimization หรือ Grid Search ซึ่งมีการใช้ทรัพยากรการประมวลผลที่สูงขึ้น

2) ในการศึกษาครั้งถัดไป

อาจพิจารณาเพิ่มตัวแปรอิสระที่อาจส่งผลต่อการใช้พลังงานของเครื่องสูบน้ำ เช่น ประวัติการบำรุงรักษา ระดับการสึกหรอของชิ้นส่วน หรือปัจจัยภายนอกอื่น ๆ ที่ไม่ได้รวมอยู่ในการศึกษาในครั้งนี้ เนื่องจากข้อจำกัดในการเข้าถึงข้อมูลเชิงปริมาณ

3) ในการศึกษาครั้งถัดไป

ผู้ที่ต้องการพัฒนาเว็บแอปพลิเคชันแบบเรียลไทม์ควรตระหนักว่าเป็นกระบวนการที่ต้องใช้เวลาในการออกแบบ ทดสอบ และตรวจสอบอย่างรอบคอบ เนื่องจากต้องให้ความสำคัญกับความปลอดภัยทางไซเบอร์เป็นพิเศษ โดยเฉพาะเมื่อมีการเชื่อมต่อกับระบบควบคุมในโรงไฟฟ้า ซึ่งต้องเป็นไปตามมาตรฐานอุตสาหกรรมอย่างเคร่งครัด ทั้งในด้านการควบคุมสิทธิ์การเข้าถึง การป้องกันข้อมูล และการลดความเสี่ยงต่อระบบหลัก

## บรรณานุกรม

- วัฒนา ถาวร. 2003. **โรงต้นกำลังไฟฟ้า**. สำนักพิมพ์สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), กรุงเทพมหานคร.
- อนิรุตต์ มัทธจักร. **วิศวกรรมโรงจักรต้นกำลัง**. สำนักพิมพ์ซีเอ็ดยูเคชั่น, กรุงเทพมหานคร.
- นवलวรรณ สุนทรภิชช. 2020. **การเรียนรู้ของเครื่อง**. สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์, กรุงเทพมหานคร.
- กิตติสุชาติ พสุภา. 2021. **ระบบอัจฉริยะขั้นสูง: ทฤษฎี อัลกอริทึม และการประยุกต์ใช้**, สำนักพิมพ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร.
- วิโรตม ตูจินดา. 2023. **โครงข่ายประสาทเทียม และการเรียนรู้เชิงลึก**, สำนักพิมพ์ มหาวิทยาลัยเกษตรศาสตร์, กรุงเทพมหานคร.
- กอบเกียรติ สระอุบล. 2020. **เรียนรู้ Data Science และ AI: Machine Learning ด้วย Python**, สำนักพิมพ์มีเดียเนทเวิร์ค, กรุงเทพมหานคร.
- กอบเกียรติ สระอุบล. 2020. **เรียนรู้ Data Science และ AI: Machine Learning ด้วย Python**, สำนักพิมพ์มีเดียเนทเวิร์ค, กรุงเทพมหานคร.
- กอบเกียรติ สระอุบล. 2022. **เรียนรู้ AI: Deep Learning ด้วย Python**, สำนักพิมพ์อินเทอร์เน็ตมีเดีย, กรุงเทพมหานคร.
- กัลยา วานิชย์บัญชา. 2017. **หลักสถิติ**, สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร.
- Moleda, M., Momot, A., & Mrozek, D. 2020. **Predictive Maintenance of Boiler Feed Water Pumps Using SCADA Data**. *Sensors*, 20(2), 571.
- J. Parraga-Alava and J. Rodas-Silva. 2024. **Enhancing Electric Power Consumption Prediction Using Machine Learning and Genetic Algorithms**. *IEEE Eighth Ecuador Technical Chapters Meeting (ETCM)*, Cuenca, Ecuador, 2024, pp. 1-6.

A. Satan, A. Khamzina, D. Toktarbayev, Z. Sotsial, I. Bapiyev and N. Zhakiyev. 2022. **Comparative LSTM and SVM Machine Learning Approaches for Energy Consumption Prediction: Case Study in Akmola.** International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, pp. 1-7.

Jemila, M. R. Mano, S, Pushpalatha K, R, Mithuna H, C, Supriya, Maheswari, K.T., and M, R. 2024. **A Novel Strategy to Estimate and Manage the Power Consumption of Household Appliances.** International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, pp. 1949-1954.

Al-Jamimi, Hamdi A., Binmakhshen, Galal M., Worku, Muhammed Y., and Hassan, Mohamed A. 2023. **Advancements in Household Load Forecasting: Deep Learning Model with Hyperparameter Optimization.** Electronics 12, no. 24: 4909.

Yin, Zhiqiang, Shi, Lin, Luo, Junru, Xu, Shoukun, Yuan, Yang, Tan, Xinxin, and Zhu, Jiaqun. 2023. **Pump Feature Construction and Electrical Energy Consumption Prediction Based on Feature Engineering and LightGBM Algorithm.** Sustainability, 15(1), 789.

Z. Attal, H. Noura, O. Salman and F. Vernier, 2024. **Predicting Power Consumption Using Machine Learning Techniques.** International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 2024, pp. 1522-1527.

Farsi, Behnam, Amayri, Manar, Bouguila, N., and Eicker, U.. 2021. **On Short-Term Load Forecasting Using Machine Learning Techniques and a Novel Parallel Deep LSTM-CNN Approach.** Energies 16, no. 23: 7878.

Kingma, D. P., & Ba, J. (2014). **Adam: A Method for Stochastic Optimization.** arXiv preprint arXiv:1412.6980.

- Tibshirani, R. 1996. **Regression Shrinkage and Selection via the Lasso**. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.
- Zou, H., & Hastie, T. 2005. **Regularization and Variable Selection via the Elastic Net**. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.
- Smola, A. J., & Schölkopf, B. 2004. **A Tutorial on Support Vector Regression**. Statistics and Computing, 14(3), 199-222.
- Drucker, H., et al. (1997). **Support Vector Regression Machines**. Advances in Neural Information Processing Systems, 9, 155-161.
- Chen, T., & Guestrin, C. 2016. **XGBoost: A Scalable Tree Boosting System**. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
- Dorogush, A.V., Ershov, V. and Gulin, A. 2018. **CatBoost: Gradient Boosting with Categorical Features Support**. arXiv preprint arXiv:1810.11363.
- Ke, G., Meng, Q., Finley, T., et al. 2017. **LightGBM: A Highly Efficient Gradient Boosting Decision Tree**. In Advances in Neural Information Processing Systems (pp. 3146-3154).
- Larry Drbal, Kayla Westra, Pat Boston. 1996. **Power Plant Engineering**. Springer.
- Mitchell, T. M. 1997. **Machine Learning**. McGraw-Hill.
- Hastie, T., Tibshirani, R., & Friedman, J. 2009. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. 2016. **Deep Learning**. MIT Press.
- Breiman, L., et al. 1984. **Classification and Regression Trees**. Wadsworth International Group.



ภาคผนวก

## ภาคผนวก ก

รายละเอียดเชิงเทคนิคของเครื่องสูบน้ำป้อนหม้อไอน้ำเครื่องสูบน้ำป้อนหม้อไอน้ำที่ใช้ในงานวิจัยนี้เป็นรุ่น HGC 4/5+1 ซึ่งออกแบบให้รองรับแรงดันและอุณหภูมิสูง เหมาะสำหรับการใช้งานในโรงไฟฟ้า โดยต้องใช้น้ำที่มีค่า pH >9.0 และปริมาณออกซิเจนต่ำกว่า 0.02 mg/l เพื่อป้องกันการกัดกร่อนของวัสดุ ซึ่งประกอบด้วยรายละเอียดที่สำคัญดังนี้

### 1. ข้อมูลทั่วไปของเครื่องสูบน้ำป้อนหม้อไอน้ำ

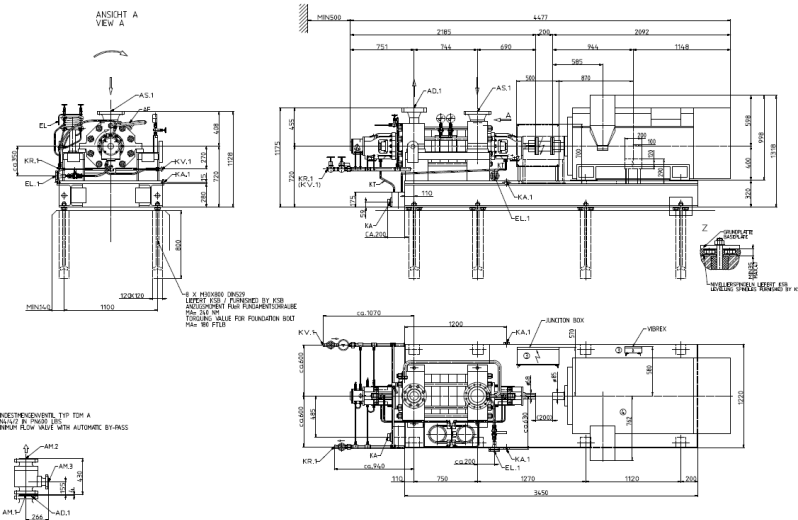
- Pump Type: Horizontal ring-section pump
- Pump Size / Stages: HGC 4/5+1
- Drive: Electric motor
- Manufacturing Company: KSB Pumps
- Application: High-pressure boiler feed water pump

### 2. ข้อมูลการทำงานของปั๊ม (Operating Data)

- pH Value at 25°C: >9.0
- Oxygen Content (O<sub>2</sub> mg/l): < 0.02
- Water Type: Alkaline composition with salt-free water
- ค่าการทำงานที่กำหนด (Operating Parameters)
- อุณหภูมิใช้งาน (Operating Temperature): 130.5°C
- ความหนาแน่น (Density): 0.9342 kg/dm<sup>3</sup>
- แรงดันในถังดูด (Pressure in Inlet Vessel): 2.797 kg/cm<sup>2</sup>
- แรงดันที่หัวดูด (Suction Pressure): 3.26 kg/cm<sup>2</sup>
- NPSH Available: 7.0 m
- NPSH Required at Qs: 4.74 m
- อัตราการไหล (Capacity): 205.0 m<sup>3</sup>/h (ต่ำสุด 50.64 m<sup>3</sup>/h)
- แรงม้ารวม (Total Head): 778.08 m (ต่ำสุด 899.28 m)
- แรงดันที่หัวจ่าย (Pump Discharge Pressure): 75.76 kg/cm<sup>2</sup>
- ประสิทธิภาพ (Efficiency acc. to Karassik): 74.34%
- กำลังไฟที่ใช้ (Pump Input Power): 543.33 kW
- ความเร็วรอบปั๊ม (Pump Speed): 2982 RPM
- ทิศทางการหมุน (Rotation Direction): ตามเข็มนาฬิกา (Clockwise)

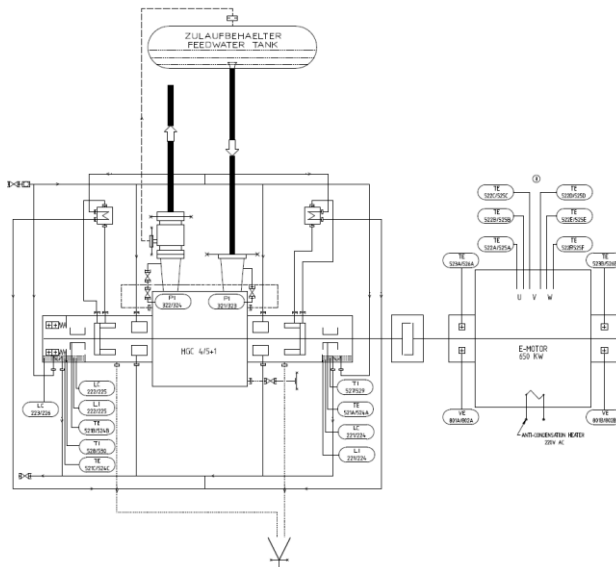
### 3. แบบแสดงรายละเอียดที่สำคัญ

#### 3.1 แผนการติดตั้งเครื่องสูบน้ำป้อนหม้อไอน้ำ



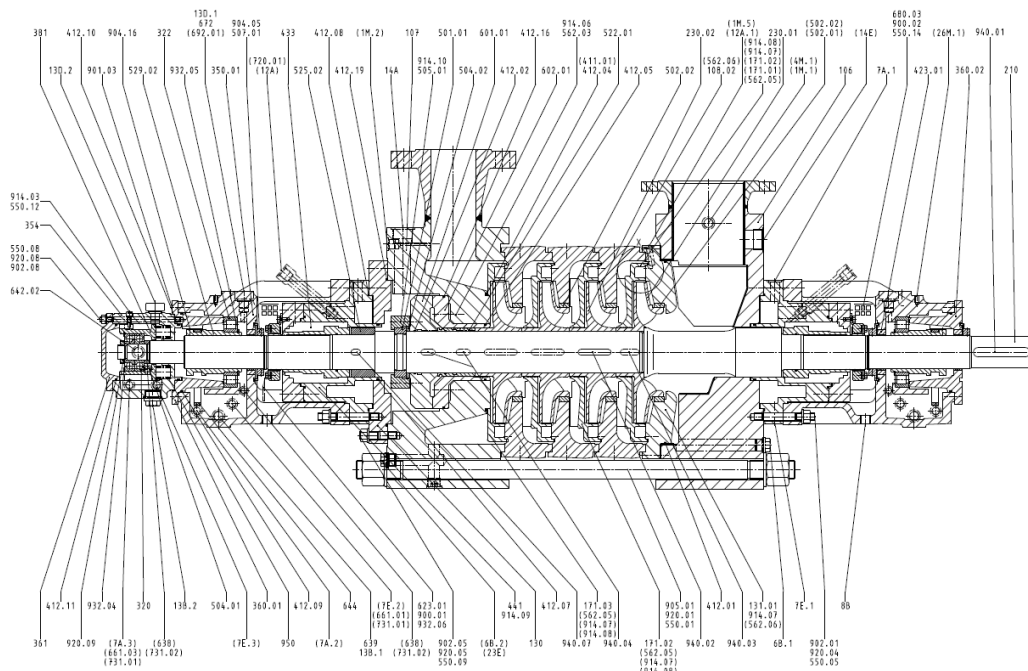
รูป ก.1 แบบแสดงแผนการติดตั้งเครื่องสูบน้ำป้อนหม้อไอน้ำ

#### 3.2 แบบสัญลักษณ์อุปกรณ์และเครื่องมือวัด



รูป ก.2 แบบสัญลักษณ์เพื่อแทนการแสดงรายละเอียดอุปกรณ์ต่างๆที่อยู่ในระบบท่อ และเครื่องมือวัดทางอุตสาหกรรมของเครื่องสูบน้ำป้อนหม้อไอน้ำ

### 3.3 แบบภาพตัดเครื่องสูบน้ำป้อนหม้อไอน้ำ



รูป ก.3 แบบภาพตัดเครื่องสูบน้ำป้อนหม้อไอน้ำ

## 4. รายการจุดวัดค่าสำหรับเครื่องสูบน้ำป้อนหม้อไอน้ำ

### 4.1 ตัวเรือนเครื่องสูบน้ำป้อนหม้อไอน้ำ

การวัดแรงดัน (Pressure Measurement)

- PI 321: วัดความดันหัวดูดของปั๊ม
- PI 322: วัดความดันหัวจ่ายของปั๊ม
- เครื่องมือวัดที่ใช้: WIKA pressure gauge พร้อมวาล์วเกจ

การวัดอุณหภูมิ (Temperature Measurement)

- TI 527, TI 530: วัดอุณหภูมิน้ำมันที่แบร์ริง
- TE 521A, TE 521B, TE 521C: วัดอุณหภูมิที่แบร์ริงและ Lift-off device
- เครื่องมือวัดที่ใช้: SIKA และ RECKMANN พร้อมสัญญาณเตือน (Alarm, Trip)

การวัดระดับน้ำมัน (Oil Level Measurement)

- LI 221, LI 222, LI 224, LI 225: วัดระดับน้ำมันที่แบร์ริง
- LC 221, LC 222, LC 223: ควบคุมระดับน้ำมันอัตโนมัติ

### 4.2 ตัวเรือนมอเตอร์ (Motor Housing)

การวัดอุณหภูมิขดลวดมอเตอร์ (Motor Winding Temperature Measurement)

- TE 522B, TE 522C, TE 522D, TE 522E, TE 522F: วัดอุณหภูมิขดลวดของมอเตอร์ในแต่ละเฟส (U, V, W)
- เครื่องมือวัดที่ใช้: Resistance thermometer พร้อมระบบเตือน (Alarm, Trip)

การวัดแรงสั่นสะเทือนของตลับลูกปืน (Bearing Vibration Measurement)

- VE 801 A, VE 801 B: วัดแรงสั่นสะเทือนของตลับลูกปืนที่ด้านขับ (DE-Side) และด้านไม่ขับ (NDE-Side)
- เครื่องมือวัดที่ใช้: VIBREX Vibration Monitor พร้อม Transmitter



## ภาคผนวก ข

การศึกษาวิจัยเรื่อง "การทำนายการใช้พลังงานของเครื่องสูบน้ำป้อนหม้อไอน้ำในโรงไฟฟ้า โดยใช้การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก" เขียนโดยโปรแกรมภาษาไพธอน โดยมีชุดคำสั่งประกอบด้วย 7 ส่วน ดังนี้

### 1. การติดตั้งและนำเข้าไลบรารี

```
!pip install catboost
!pip install lightgbm
!pip uninstall -y scikit-learn
!pip install scikit-learn==1.3.1
!pip install scikeras==0.10.0
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
# Machine Learning
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor
# Deep Learning
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
```

```
# Model Selection
from sklearn.model_selection import RandomizedSearchCV, train_test_split, KFold,
cross_val_score
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score,
make_scorer
import warnings
warnings.filterwarnings('ignore')
```

## 2. การตรวจสอบและปรับปรุงโครงสร้างข้อมูล

```
#load data
df = pd.read_excel('/content/BFW-A.xlsx', sheet_name='Sheet1')
# display data
df.head(10)
df.tail()
df.shape
df.info()
# Rename columns for easier handling
df.columns = [col.strip().replace(' ', '_').upper() for col in df.columns]
df.columns
# Rename column 'Unnamed: 0' to 'TIMESTAMP' and Convert timestamp column to
datetime i
df.rename(columns={'UNNAMED: _0': 'TIMESTAMP'}, inplace=True)
df['TIMESTAMP'] = pd.to_datetime(df['TIMESTAMP'], errors='coerce')
#Change all columns to numeric datatype except TIMESTAMP
for col in df.columns:
    if col != 'TIMESTAMP':
        df[col] = pd.to_numeric(df[col], errors='coerce')
```

### 3. การจัดการคุณภาพข้อมูล

```

# Check for missing or invalid data
missing_data = df.isnull().sum()
missing_data
# Display summary of missing values
missing_summary = pd.DataFrame({
    'Column': df.columns,
    'Missing Values': missing_data,
    'Percentage Missing': (missing_data / len(df)) * 100
}).sort_values(by='Percentage Missing', ascending=False)
missing_summary
# show missing values
df[df.isnull().any(axis=1)].head(10)
# Visualization
numerical_columns = df.select_dtypes(include=[np.number]).columns
plt.figure(figsize=(20, len(numerical_columns) * 4))
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(len(numerical_columns), 1, i)
    plt.plot(df['TIMESTAMP'], df[col], label=col, alpha=0.7)
    plt.title(f"Time Series of {col}")
    plt.xlabel("Time")
    plt.ylabel(col)
    plt.legend()
    plt.grid(True)
plt.tight_layout()
plt.show()
# Plot all parameters overlaid in a single graph
plt.figure(figsize=(20, 10))
for col in numerical_columns:
    plt.plot(df['TIMESTAMP'], df[col], label=col, alpha=0.7)
plt.title("Time Series")

```

```

plt.xlabel("Time")
plt.ylabel("Value")
plt.legend(loc="upper right", bbox_to_anchor=(1.2, 1))
plt.grid(True)
plt.tight_layout()
plt.show()

#Remove missing values
df = df.dropna()
df.shape
#Check duplicate data
df.duplicated().sum()
#If "BFW#1_MOTOR_ACTIVE_POWER" == 0, Remove
df_clean_1 = df[df['BFW#1_MOTOR_ACTIVE_POWER'] != 0]
df_clean_1.shape
#Exploratory Data Analysis (EDA):
df_clean_1.info()
df_clean_1.describe()
#Visualize the data
sns.pairplot(df_clean_1)
plt.show()
#Visualize the Correlation
df_clean_1.corr()
#Correlation Heatmap
plt.figure(figsize=(15, 8))
sns.heatmap(df_clean_1.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
#Handle Outliers with a Boxplot
columns_to_check = df_clean_1.drop(columns=['TIMESTAMP']).columns
for column in columns_to_check:
    plt.figure(figsize=(8, 6)) # Adjust figure size as needed

```

```

sns.boxplot(x=df_clean_1[column])
plt.title(f'Box Plot of {column}')
plt.show()

# Function to handle outliers using IQR
def handle_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    # Replace with bounds (Winsorization/Clipping):
    data[column] = np.clip(data[column], lower_bound, upper_bound)
    return data

for column in columns_to_check:
    df_clean = handle_outliers_iqr(df_clean_1, column)

# Handle Outliers with a Boxplot
for column in columns_to_check:
    plt.figure(figsize=(8, 6)) # Adjust figure size as needed
    sns.boxplot(x=df_clean[column])
    plt.title(f'Box Plot of {column}')
    plt.show()

```

#### 4. การคัดเลือกตัวแปรอิสระ (Feature Selection)

```

# Define the target variable and features
target_column = 'BFW#1_MOTOR_ACTIVE_POWER'
X = df_clean.drop(columns=['TIMESTAMP', target_column])
y = df_clean[target_column]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

```

```

# Standardize the features
scaler = StandardScaler()
X_train_scaled = pd.DataFrame(scaler.fit_transform(X_train),
columns=X_train.columns)
X_test_scaled = pd.DataFrame(scaler.transform(X_test), columns=X_test.columns)
# 1.1 กรองตัวแปรด้วย VIF <= 10
from statsmodels.stats.outliers_influence import variance_inflation_factor
def calculate_vif(X):
    vif_data = pd.DataFrame()
    vif_data["feature"] = X.columns
    vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
    print(vif_data)
    return vif_data[vif_data["VIF"] <= 10]["feature"]
vif_features = calculate_vif(X_train_scaled)
X_train_vif = X_train_scaled[vif_features]
X_test_vif = X_test_scaled[vif_features]
# 1.2 กรองตัวแปรด้วย P-Value <= 0.05
import statsmodels.api as sm
# Reset index for y_train to match X_train_scaled
y_train_reset = y_train.reset_index(drop=True)
X_train_const = sm.add_constant(X_train_scaled)
model = sm.OLS(y_train_reset, X_train_const).fit()
p_values = model.pvalues[model.pvalues <= 0.05].index
p_value_features = [feature for feature in p_values if feature != 'const']
X_train_pvalue = X_train_scaled[p_value_features]
X_test_pvalue = X_test_scaled[p_value_features]
model.summary2()
# selected_col =
['BFW#1_PUMP_BEARING_TEMP#A','BFW#1_PUMP_BEARING_TEMP#C','BFW#1_MOTOR
_BEARING_TEMP#B','BFW#1_BEARING_HOUSING_VIBR_A','BFW#1_BEARING_HOUSING_VI
BR_B','BFW#1_MOTOR_WINDING_TEMP#A','BFW#1_MOTOR_WINDING_TEMP#B','BFW#1

```

```

_MOTOR_WINDING_TEMP#C','BFW#1_MOTOR_WINDING_TEMP#D','BFW_PUMP_DISCHA
RGE_PRESSURE','GTG11_POWER','GTG12_POWER']
# vif_cal = calculate_vif(X_train_scaled[selected_col])
# print(vif_cal)
selected_col_2 =
['BFW#1_PUMP_BEARING_TEMP#A','BFW#1_PUMP_BEARING_TEMP#C','BFW#1_MOTOR
_BEARING_TEMP#B','BFW#1_BEARING_HOUSING_VIBR_B','BFW#1_MOTOR_WINDING_TE
MP#C','BFW_PUMP_DISCHARGE_PRESSURE','GTG11_POWER','GTG12_POWER']
# vif_cal_2 = calculate_vif(X_train_scaled[selected_col_2])
# print(vif_cal_2)
print('-----VIF Score-----')
print(f'vif features: {vif_features}')
print('-----P-Values-----')
print(model)
print(f'p-values features: {p_value_features}')
from sklearn.inspection import permutation_importance
def get_feature_importance(model, X_train, y_train, top_n=10):
    # Fit the model
    model.fit(X_train, y_train)
    # Check if the model has feature importances or coef_
    if hasattr(model, "feature_importances_"):
        importances = model.feature_importances_
    elif hasattr(model, "coef_"):
        importances = np.abs(model.coef_)
    else:
        # Use permutation importance as a fallback
        perm_importance = permutation_importance(model, X_train, y_train,
n_repeats=10, random_state=42)
        importances = perm_importance.importances_mean
    # Create a DataFrame of feature importances

```

```

importance_df = pd.DataFrame({'Feature': X_train.columns, 'Importance':
importance_df.sort_values(by='Importance', ascending=False)
print(importance_df)
# Return top_n features
return importance_df.head(top_n)["Feature"].values
from sklearn.model_selection import RandomizedSearchCV
# Define models and parameter grids
models = {
    "Linear": LinearRegression(),
    "Ridge": Ridge(),
    "Lasso": Lasso(),
    "Elasticnet": ElasticNet(),
    "Support Vector Regressor": SVR(),
    "Decision Tree": DecisionTreeRegressor(),
    "Random Forest": RandomForestRegressor(),
    "XGBoost": XGBRegressor(),
    "CatBoost": CatBoostRegressor(verbose=0),
    "LightGBM": LGBMRegressor()
}
param_grids = {
    "Ridge": {'alpha': [0.01,0.1, 1.0, 10.0]},
    "Lasso": {'alpha': [0.01,0.1, 1.0, 10.0]},
    "Elasticnet": {'alpha': [0.01,0.1, 1.0, 10.0], 'l1_ratio': [0.2, 0.5, 0.8]},
    "Support Vector Regressor": {'C': [0.1, 1, 10,100], 'epsilon': [0.1, 0.5]},
    "Decision Tree": {'max_depth': [5, 10, 20], 'min_samples_split': [2, 5,
10], 'min_samples_leaf': [1, 2, 5]},
    "Random Forest": {'n_estimators': [100, 200], 'max_depth': [10, 20]},
    "XGBoost": {'n_estimators': [100, 200], 'learning_rate': [0.01, 0.1]},
    "CatBoost": {'iterations': [100, 200], 'learning_rate': [0.01, 0.1]},
    "LightGBM": {'n_estimators': [100, 200], 'learning_rate': [0.01, 0.1]}

```

```

}
# Store results
results = []

```

### 5. การพัฒนาตัวแบบการเรียนรู้และการประเมินผล

```

# Loop through models and datasets
for model_name, model in models.items():
    print(f"Processing model: {model_name}")
    # Prepare Feature Importance datasets for the current model
    top_5_features = get_feature_importance(model, X_train_scaled, y_train, top_n=5)
    print(f"top_5_features: {top_5_features}")
    top_10_features = get_feature_importance(model, X_train_scaled, y_train,
    top_n=10)
    print(f"top_10_features: {top_10_features}")
    datasets = {
        "All": (X_train_scaled, X_test_scaled),
        "VIF <= 10": (X_train_vif, X_test_vif),
        "P <= 0.05": (X_train_pvalue, X_test_pvalue),
        "Feature Importance Top 5": (X_train_scaled[top_5_features],
X_test_scaled[top_5_features]),
        "Feature Importance Top 10": (X_train_scaled[top_10_features],
X_test_scaled[top_10_features])
    }
    # Loop through datasets
    for data_name, (X_tr, X_te) in datasets.items():
        print(f"Evaluating {model_name} with {data_name}...")
        # If the model has hyperparameters to optimize
        if model_name in param_grids:
            search = RandomizedSearchCV(
                model, param_distributions=param_grids[model_name],

```

```

n_iter=10, scoring='neg_mean_squared_error', cv=3, random_state=42,
n_jobs=-1)
search.fit(X_tr, y_train)
best_model = search.best_estimator_
print(f"Best parameters for {model_name}: {search.best_params_}")
else:
    best_model = model
    best_model.fit(X_tr, y_train)
# Evaluate the model
y_pred = best_model.predict(X_te)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
# Store the results
results.append({
    "Model": model_name,
    "Dataset": data_name,
    "MSE": mse,
    "R2": r2
})
# Convert results to DataFrame
results_df = pd.DataFrame(results)
# Display all results
print("Experiment Results:")
print(results_df)
# Find the best model and configuration
best_result = results_df.loc[results_df['R2'].idxmax()]
print("Best Model and Configuration:")
print(best_result)
#Regression (P-Value & VIF)
X_train_vif = X_train_scaled[selected_col_2]
X_test_vif = X_test_scaled[selected_col_2]

```

```

model_reg = LinearRegression()
model_reg.fit(X_train_vif, y_train)
y_pred_reg = model_reg.predict(X_test_vif)
mse_reg = mean_squared_error(y_test, y_pred_reg)
r2_reg = r2_score(y_test, y_pred_reg)
print(mse_reg, r2_reg)
#Best Model
best_model = SVR(C = 100, epsilon= 0.5)
best_model.fit(X_train_scaled, y_train)
y_pred = best_model.predict(X_test_scaled)
best_mse = mean_squared_error(y_test, y_pred)
best_r2 = r2_score(y_test, y_pred)
print(f'MSE: {best_mse}, R2 SCORE: {best_r2}')
#Deep Learning Model
#Neural Network
model_nn = Sequential()
model_nn.add(Dense(128, activation='relu', input_dim=X_train_scaled.shape[1]))
model_nn.add(Dense(64, activation='relu'))
model_nn.add(Dense(1))
# Compile the model
model_nn.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mse'])
# Early stopping to prevent overfitting
# early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
# Train the model
history_nn = model_nn.fit(
    X_train_scaled, y_train,
    validation_data=(X_test_scaled, y_test),
    epochs=100,
    batch_size=32,
    #callbacks=[early_stopping],

```

```

    verbose=1)

# Evaluate the model
loss, mse = model_nn.evaluate(X_test_scaled, y_test, verbose=0)
predictions_model_nn = model_nn.predict(X_test_scaled)
r2_model_nn = r2_score(y_test, predictions_model_nn)
print(f"Model 1 - Neural Network: Loss = {loss}, MSE = {mse}, R2 = {r2_model_nn}")

# Plot training and validation loss
plt.figure(figsize=(10, 6))
plt.plot(history_nn.history['loss'], label='Training Loss')
plt.plot(history_nn.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

#Deep Learning Model - RNN
from tensorflow.keras.layers import SimpleRNN, GRU, LSTM, Dense, Dropout
# Prepare data for RNN (reshape to 3D: samples, timesteps, features)
X_train_rnn = X_train_scaled.values.reshape((X_train_scaled.shape[0], 1,
X_train_scaled.shape[1]))
X_test_rnn = X_test_scaled.values.reshape((X_test_scaled.shape[0], 1,
X_test_scaled.shape[1]))
# Build the RNN model
model_rnn = Sequential()
model_rnn.add(SimpleRNN(128, activation='relu', input_shape=(X_train_rnn.shape[1],
X_train_rnn.shape[2])))
model_rnn.add(Dense(64, activation='relu'))
model_rnn.add(Dense(1)) # Single output for regression
# Compile the RNN model
model_rnn.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mse'])
# Train the RNN model

```

```

history_rnn = model_rnn.fit(
    X_train_rnn, y_train,
    validation_data=(X_test_rnn, y_test),
    epochs=100,
    batch_size=32,
    #callbacks=[early_stopping],
    verbose=1
)
# Evaluate the RNN model
loss_rnn, mse_rnn = model_rnn.evaluate(X_test_rnn, y_test, verbose=0)
predictions_rnn = model_rnn.predict(X_test_rnn)
r2_rnn = r2_score(y_test, predictions_rnn)
print(f"GRU Model: Loss = {loss_rnn}, MSE = {mse_rnn}, R2 = {r2_rnn}")
#Deep Learning Model - GRU
# Prepare data for GRU (reshape to 3D: samples, timesteps, features)
X_train_gru = X_train_scaled.values.reshape((X_train_scaled.shape[0], 1,
X_train_scaled.shape[1]))
X_test_gru = X_test_scaled.values.reshape((X_test_scaled.shape[0], 1,
X_test_scaled.shape[1]))
# Build the GRU model
model_gru = Sequential()

model_gru.add(GRU(128, activation='relu', input_shape=(X_train_gru.shape[1],
X_train_gru.shape[2])))

model_gru.add(Dense(64, activation='relu'))

model_gru.add(Dense(1)) # Single output for regression

# Compile the GRU model
model_gru.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mse'])

```

```
# Train the GRU model

history_gru = model_gru.fit(

    X_train_gru, y_train,

    validation_data=(X_test_gru, y_test),

    epochs=100,

    batch_size=32,

    #callbacks=[early_stopping],

    verbose=1)

# Evaluate the GRU model

loss_gru, mse_gru = model_gru.evaluate(X_test_gru, y_test, verbose=0)

predictions_gru = model_gru.predict(X_test_gru)

r2_gru = r2_score(y_test, predictions_gru)

print(f"GRU Model: Loss = {loss_gru}, MSE = {mse_gru}, R² = {r2_gru}")

#Deep Learning Model - LSTM

# Prepare data for LSTM (reshape to 3D: samples, timesteps, features)

X_train_lstm = X_train_scaled.values.reshape((X_train_scaled.shape[0], 1,

X_train_scaled.shape[1]))

X_test_lstm = X_test_scaled.values.reshape((X_test_scaled.shape[0], 1,

X_test_scaled.shape[1]))

# Build the LSTM model

model_lstm = Sequential()

model_lstm.add(LSTM(128, activation='relu', return_sequences=False,

input_shape=(X_train_lstm.shape[1], X_train_lstm.shape[2])))
```

```

model_lstm.add(Dense(64, activation='relu'))

model_lstm.add(Dense(1)) # Single output for regression

# Compile the LSTM model

model_lstm.compile(optimizer=Adam(learning_rate=0.001), loss='mse',
metrics=['mse'])

# Train the LSTM model

history_lstm = model_lstm.fit(
    X_train_lstm, y_train,
    validation_data=(X_test_lstm, y_test),
    epochs=100,
    batch_size=32,
    #callbacks=[early_stopping],
    verbose=1)

# Evaluate the LSTM model

loss_lstm, mse_lstm = model_lstm.evaluate(X_test_lstm, y_test, verbose=0)

predictions_lstm = model_lstm.predict(X_test_lstm)

r2_lstm = r2_score(y_test, predictions_lstm)

print(f"LSTM Model: Loss = {loss_lstm}, MSE = {mse_lstm}, R2 = {r2_lstm}")

# Comparison of predictions from the best Deep Learning model

predictions_dl = model_1.predict(X_test_scaled)

plt.figure(figsize=(10, 6))

plt.scatter(y_test, predictions_dl, alpha=0.5)

plt.plot(y_test, y_test, color='red')

```

```

plt.xlabel("Actual Values")

plt.ylabel("Predicted Values")

plt.title("Actual vs Predicted Values for Deep Learning Model")

plt.show()

df_clean['PREDICTED_DL'] = model_1.predict(scaler.transform(X)).flatten()

# Sort data by TIMESTAMP for consistency
df_clean.sort_values('TIMESTAMP', inplace=True)

```

## 6. การตรวจจับค่าผิดปกติ (Anomaly Detection)

```

#Combine Top 3 Model
df_predict.head()

# 1. SVR (All Features)
svr_model = SVR(C=100, epsilon=0.5)
svr_model.fit(X_train_scaled, y_train)
df_predict['SVR_PREDICTED'] = svr_model.predict(scaler.transform(X))

# 2. LGBM (Top 10 Features)
# Fit a new scaler on the top 10 features
scaler_top10 = StandardScaler()
top_10_features = ['BFW_PUMP_DISCHARGE_PRESSURE',
'BFW#1_BEARING_HOUSING_VIBR_A',
'BFW#1_BEARING_HOUSING_VIBR_B','GTG12_POWER','BFW#1_PUMP_BEARING_TEMP#B'
,'BFW#1_PUMP_BEARING_TEMP#A','BFW#1_MOTOR_WINDING_TEMP#A','GTG11_POWER
','BFW#1_MOTOR_WINDING_TEMP#D','BFW#1_MOTOR_BEARING_TEMP#B']

X_train_top10_scaled = pd.DataFrame(
    scaler_top10.fit_transform(X_train[top_10_features]),
    columns=top_10_features)

X_test_top10_scaled = pd.DataFrame(
    scaler_top10.transform(X_test[top_10_features]),
    columns=top_10_features)

```

```

lgbm_top10_model = LGBMRegressor(n_estimators = 200, learning_rate = 0.1)
lgbm_top10_model.fit(X_train_top10_scaled, y_train)
df_predict["LGBM_TOP10_PREDICTED"] = lgbm_top10_model.predict(
    pd.DataFrame(scaler_top10.transform(X[top_10_features]),
    columns=top_10_features))
# 3. LGBM (All Features)
lgbm_all_model = LGBMRegressor(n_estimators = 200, learning_rate = 0.1)
lgbm_all_model.fit(X_train_scaled, y_train)
df_predict["LGBM_ALL_PREDICTED"] = lgbm_all_model.predict(scaler.transform(X))
#Calculate residuals for each model
df_predict["SVR_RESIDUALS"] = df_predict[target_column] -
df_predict["SVR_PREDICTED"]
df_predict["LGBM_TOP10_RESIDUALS"] = df_predict[target_column] -
df_predict["LGBM_TOP10_PREDICTED"]
df_predict["LGBM_ALL_RESIDUALS"] = df_predict[target_column] -
df_predict["LGBM_ALL_PREDICTED"]
# Define IQR-based anomaly detection function
def detect_anomalies(residuals):
    Q1 = residuals.quantile(0.25)
    Q3 = residuals.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound_lv1 = Q1 - 1.5 * IQR
    upper_bound_lv1 = Q3 + 1.5 * IQR
    lower_bound_lv2 = Q1 - 3 * IQR
    upper_bound_lv2 = Q3 + 3 * IQR
    return lower_bound_lv1, upper_bound_lv1, lower_bound_lv2, upper_bound_lv2
# Apply IQR anomaly detection for each model
def classify_anomaly(residual, lower_lv1, upper_lv1, lower_lv2, upper_lv2):
    if residual < lower_lv2 or residual > upper_lv2:
        return 'Warning Lv2'
    elif residual < lower_lv1 or residual > upper_lv1:

```

```

        return 'Warning Lv1'
    else:
        return 'Normal'

# Anomaly detection for SVR
svr_bounds = detect_anomalies(df_predict['SVR_RESIDUALS'])
df_predict['SVR_ANOMALY'] = df_predict['SVR_RESIDUALS'].apply(
    lambda x: classify_anomaly(x, *svr_bounds))
print(svr_bounds)

# Anomaly detection for LGBM (Top 10 Features)
lgbm_top10_bounds = detect_anomalies(df_predict['LGBM_TOP10_RESIDUALS'])
df_predict['LGBM_TOP10_ANOMALY'] = df_predict['LGBM_TOP10_RESIDUALS'].apply(
    lambda x: classify_anomaly(x, *lgbm_top10_bounds))
print(lgbm_top10_bounds)

# Anomaly detection for LGBM (All Features)
lgbm_all_bounds = detect_anomalies(df_predict['LGBM_ALL_RESIDUALS'])
df_predict['LGBM_ALL_ANOMALY'] = df_predict['LGBM_ALL_RESIDUALS'].apply(
    lambda x: classify_anomaly(x, *lgbm_all_bounds))
print(lgbm_all_bounds)

df_predict[['SVR_ANOMALY', 'LGBM_TOP10_ANOMALY', 'LGBM_ALL_ANOMALY']]
# ANOMALY CONFIRMED based on all three models' agreement
def confirm_anomaly(row):
    # If all three models agree on Warning Lv2
    if (row['SVR_ANOMALY'] == 'Warning Lv2' and
        row['LGBM_TOP10_ANOMALY'] == 'Warning Lv2' and
        row['LGBM_ALL_ANOMALY'] == 'Warning Lv2'):
        return 'Warning Lv2'

    # If all three models agree on Warning Lv1
    elif (
        row['SVR_ANOMALY'] == 'Warning Lv1' and
        row['LGBM_TOP10_ANOMALY'] == 'Warning Lv1' and
        row['LGBM_ALL_ANOMALY'] == 'Warning Lv1'):

```

```

        return 'Warning Lv1'
    # Otherwise, mark as normal
else:
    return 'Normal'
# Apply the logic to classify anomalies
df_predict['ANOMALY_CONFIRMED'] = df_predict.apply(confirm_anomaly, axis=1)
# Count anomalies
print("Anomaly counts:")
print(df_predict['ANOMALY_CONFIRMED'].value_counts())
import plotly.graph_objects as go
# Create a Plotly figure
fig = go.Figure()
# Plot actual values
fig.add_trace(
    go.Scatter(
        x=df_predict['TIMESTAMP'],
        y=df_predict[target_column],
        mode='lines',
        name='Actual',
        line=dict(color='blue', width=2),
    ))
# Plot SVR predictions
fig.add_trace(
    go.Scatter(
        x=df_predict['TIMESTAMP'],
        y=df_predict['SVR_PREDICTED'],
        mode='lines',
        name='SVR Predictions',
        line=dict(color='green', dash='dash', width=2),))
# Plot LGBM (Top 10 Features) predictions
fig.add_trace(

```

```

go.Scatter(
    x=df_predict['TIMESTAMP'],
    y=df_predict['LGBM_TOP10_PREDICTED'],
    mode='lines',
    name='LGBM (Top 10 Features) Predictions',
    line=dict(color='orange', dash='dot', width=2,))
# Plot LGBM (All Features) predictions
fig.add_trace(
    go.Scatter(
        x=df_predict['TIMESTAMP'],
        y=df_predict['LGBM_ALL_PREDICTED'],
        mode='lines',
        name='LGBM (All Features) Predictions',
        line=dict(color='purple', dash='dashdot', width=2,))
# Plot Warning Lv1
warning_lv1 = df_predict[df_predict['ANOMALY_CONFIRMED'] == 'Warning Lv1']
fig.add_trace(
    go.Scatter(
        x=warning_lv1['TIMESTAMP'],
        y=warning_lv1[target_column],
        mode='markers',
        name='Warning Lv1',
        marker=dict(color='orange', size=8, symbol='circle'),)
)
# Plot Warning Lv2
warning_lv2 = df_predict[df_predict['ANOMALY_CONFIRMED'] == 'Warning Lv2']
fig.add_trace(
    go.Scatter(
        x=warning_lv2['TIMESTAMP'],
        y=warning_lv2[target_column],
        mode='markers',

```

```

name='Warning Lv2',
marker=dict(color='red', size=10, symbol='x'))

# Update layout for better visualization
fig.update_layout(
    title="Interactive Chart: Predictions from All Models with Confirmed Anomalies",
    xaxis_title="Time",
    yaxis_title="BFW#1 Motor Active Power",
    legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right", x=1),
    template="plotly_white",)
# Show the chart
fig.show()

```

## 7. ระบบการส่งอีเมลอัตโนมัติเมื่อเจอค่าผิดปกติ

```

import yagmail
import pandas as pd
import os
from typing import Optional
import logging
from datetime import datetime
from google.colab import files # For Colab file handling
class EmailNotifier:
    def __init__(self, sender_email: str, app_password: str):
        """
        Initialize EmailNotifier with email credentials.
        Use app-specific password instead of actual password for security.
        """
        self.sender_email = sender_email
        self.app_password = app_password
        self.setup_logging()
    def setup_logging(self):
        """Setup logging configuration for Colab environment"""

```

```

logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(message)s')
self.logger = logging.getLogger(__name__)
def prepare_email_content(self, anomalies_lv1: pd.DataFrame, anomalies_lv2:
pd.DataFrame) -> str:
    """Prepare the email content with detailed information"""
    return f"""
Anomaly Detection Alert
Time Generated: {datetime.now().strftime('%Y-%m-%d-%H:%M:%S')}
Summary:
- Level 1 Anomalies: {len(anomalies_lv1)} detected
- Level 2 Anomalies: {len(anomalies_lv2)} detected
Please review the attached CSV file for detailed information.
This is an automated message. Please do not reply.
"""
def save_anomalies_to_file(self,
    anomalies_lv1: pd.DataFrame,
    anomalies_lv2: pd.DataFrame) -> Optional[str]:
    """Save anomalies to a CSV file in Colab environment"""
    try:
        timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
        filename = f"/content/anomalies_{timestamp}.csv"

        # Combine anomalies with a level identifier
        anomalies_lv1['level'] = 'Level 1'
        anomalies_lv2['level'] = 'Level 2'
        combined_anomalies = pd.concat([anomalies_lv1, anomalies_lv2])
        # Save to file
        combined_anomalies.to_csv(filename, index=False)
        return filename

```

```

except Exception as e:
    self.logger.error(f"Error saving anomalies to file: {str(e)}")
    return None

def send_notification(self,
                      receiver_email: str,
                      anomalies_lv1: pd.DataFrame,
                      anomalies_lv2: pd.DataFrame) -> bool:
    """
    Send email notification with anomaly detection results.
    Returns True if successful, False otherwise.
    """
    try:
        # Initialize yagmail SMTP object
        with yagmail.SMTP(self.sender_email, self.app_password) as yag:
            # Prepare email content
            subject = f"Anomaly Detection Alert - {datetime.now().strftime('%Y-%m-%d')}"
            content = self.prepare_email_content(anomalies_lv1, anomalies_lv2)
            # Save anomalies to file
            attachment_file = self.save_anomalies_to_file(anomalies_lv1,
                                                         anomalies_lv2)
            if not attachment_file:
                raise Exception("Failed to create attachment file")
            # Send email
            yag.send(
                to=receiver_email,
                subject=subject,
                contents=content,
                attachments=attachment_file)
            # Clean up temporary file
            os.remove(attachment_file)

```

```

        self.logger.info(f"Email sent successfully to {receiver_email}")
        return True
    except Exception as e:
        self.logger.error(f"Failed to send email: {str(e)}")
        return False

# Create sample anomaly data
# Filter Level 1 and Level 2 anomalies
anomalies_lv1 = df_predict[df_predict['ANOMALY_CONFIRMED'] == 'Warning Lv1']
anomalies_lv2 = df_predict[df_predict['ANOMALY_CONFIRMED'] == 'Warning Lv2']
# Combine both types of anomalies into a single dataframe for review
all_anomalies = pd.concat([anomalies_lv1, anomalies_lv2])
# Sort anomalies by timestamp for easier review
all_anomalies = all_anomalies.sort_values(by='TIMESTAMP')
# Initialize the notifier
notifier = EmailNotifier(
    sender_email="",
    app_password=""
)
# Send notification
success = notifier.send_notification(
    receiver_email="",
    anomalies_lv1=anomalies_lv1,
    anomalies_lv2=anomalies_lv2)
if success:
    print("Email notification sent successfully!")
else:
    print("Failed to send email notification. Check the logs for details.")

```

## ประวัติผู้เขียน

ชื่อ-สกุล	พลวุฒิ คำฝอย
วัน เดือน ปี เกิด	3 พฤศจิกายน 2539
สถานที่เกิด	กรุงเทพมหานคร
ประวัติการศึกษา	(2562) วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง เกรดเฉลี่ย 3.42
ที่อยู่ปัจจุบัน	601 ซอย อ่อนนุช 30 แยก 7 ถนน อ่อนนุช แขวง สวนหลวง เขต สวนหลวง กรุงเทพมหานคร 10250
ผลงานตีพิมพ์	ไม่มี
รางวัลที่ได้รับ	ไม่มี

