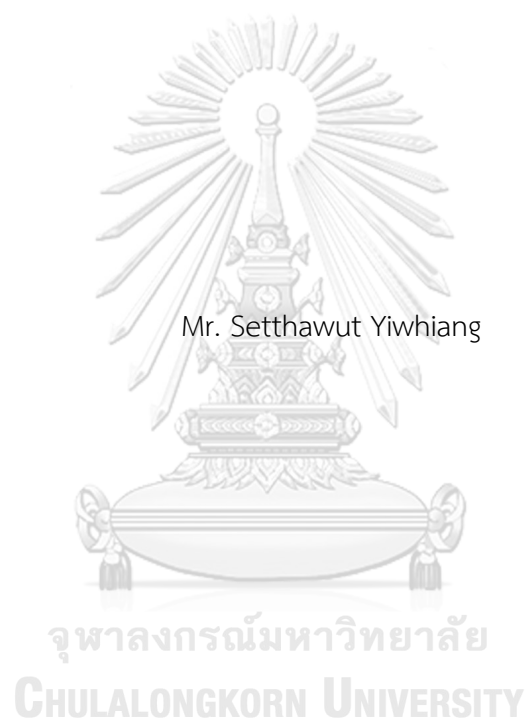


การตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้การเรียนรู้
ของเครื่อง



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

FAULT DETECTION AND CLASSIFICATION IN PHOTOVOLTAIC SYSTEMS USING MACHINE
LEARNING



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2022

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้การเรียนรู้ของเครื่อง
โดย	นายเสกฐฎฐุฒิ ยิวเหียง
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.วาทีต เบญจพลกุล
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รองศาสตราจารย์ ดร.สุรชัย ชัยทัศน์ีย์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของ
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(ผู้ช่วยศาสตราจารย์ ดร.ชาญณรงค์ บาลมงคล)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.วาทีต เบญจพลกุล)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(รองศาสตราจารย์ ดร.สุรชัย ชัยทัศน์ีย์)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ)	

เสถียรวุฒิ ยั่วเหียง : การตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้า
พลังงานแสงอาทิตย์โดยใช้การเรียนรู้ของเครื่อง. (FAULT DETECTION AND
CLASSIFICATION IN PHOTOVOLTAIC SYSTEMS USING MACHINE LEARNING) อ.
ที่ปรึกษาหลัก : ศ. ดร.วาทิต เบญจพลกุล, อ.ที่ปรึกษาร่วม : รศ. ดร.สุรชัย ชัยทัศนีย์

การใช้พลังงานหมุนเวียนโดยเฉพาะพลังงานแสงอาทิตย์มาผลิตกระแสไฟฟ้าได้รับความนิยมเพิ่มขึ้น แต่เนื่องจากพลังงานแสงอาทิตย์มีความผันผวนค่อนข้างสูงและเสี่ยงต่อการเกิดความผิดปกติ ทำให้มีความท้าทายในการรวมเข้ากับระบบไฟฟ้าที่ใช้พลังงานแบบดั้งเดิมซึ่งเป็นระบบไฟฟ้าหลัก การเกิดความผิดปกติในระบบไฟฟ้าพลังงานแสงอาทิตย์จึงเป็นหนึ่งในปัจจัยสำคัญที่ต้องพิจารณาเพื่อเพิ่มความน่าเชื่อถือให้ระบบไฟฟ้า วิทยานิพนธ์นี้จึงเสนอแนวทางการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ฝั่งกระแสตรงด้วยเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) โดยเปรียบเทียบความแม่นยำระหว่างอัลกอริทึม AdaBoost กับ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติ 5 ประเภท ได้แก่ การลัดวงจรระหว่างสายภายในสตริง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตริง โดยการฝึกและทดสอบอัลกอริทึมมี 4 พารามิเตอร์ที่เป็นอินพุต ได้แก่ กระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ซึ่งได้มาจากการจำลองความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง และเก็บรวบรวมข้อมูลด้วยการส่งข้อมูลแบบไร้สาย โดยใช้เทคโนโลยีสื่อสารไร้สาย LoRa เพื่อประหยัดและลดความยุ่งยากในการใช้สายไฟต่อจากอาเรย์เซลล์แสงอาทิตย์มายังศูนย์ควบคุม อีกทั้งยังช่วยลดปัญหาความผิดพลาดของข้อมูลเนื่องจากสายไฟแกว่งจากกระแสลม ผลการทดสอบเปรียบเทียบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติแสดงให้เห็นว่าโมเดลที่สร้างจาก Gradient Boosting มีความแม่นยำในการจำแนกประเภทความผิดปกติภายใต้ชุดข้อมูลความผิดปกติที่ได้จากการจำลองดังกล่าวมากกว่า AdaBoost อีกทั้งโมเดลที่สร้างจาก Gradient Boosting สามารถตรวจจับและระบุประเภทความผิดปกติได้อย่างแม่นยำและทำงานได้แบบเรียลไทม์

สาขาวิชา วิศวกรรมไฟฟ้า

ลายมือชื่อนิสิต

ปีการศึกษา 2565

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ลายมือชื่อ อ.ที่ปรึกษาร่วม

6270307621 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: fault detection in photovoltaic systems, machine learning, AdaBoost, Gradient Boosting, fault classification in photovoltaic systems

Setthawut Yiwhiang : FAULT DETECTION AND CLASSIFICATION IN PHOTOVOLTAIC SYSTEMS USING MACHINE LEARNING. Advisor: Prof. Watit Benjapolakul, Ph.D. Co-advisor: Assoc. Prof. SURACHAI CHAITUSANEY, Ph.D.

Solar energy is increasingly popular for generating electricity. But solar energy has a high fluctuation and risks a fault occurring. This causes a challenge for combining with conventional power systems. The fault occurring in the photovoltaic system is one of the essential factors to consider to increase the reliability of the power system. This thesis studies fault detection and classification on the DC-side of photovoltaic systems using machine learning and compares the accuracy of the AdaBoost and the Gradient Boosting algorithms in fault detection and classification. 5 Faults are classified: Intra string line-to-line fault, open circuit, partial shading, degradation, and Cross string line-line fault. The algorithm training and testing has 4 parameters as inputs: string current, system voltage, system power, and irradiance. The dataset of 4 parameters was obtained from the fault simulation on a stand-alone 180 W PV system (2 strings, 3 modules per string). LoRa wireless communication technology is used instead of wires to save and simplify data transmission. It also reduces the problem of data distortion caused by wires swaying from the wind. The results comparing the accuracy of fault detection and classification showed that the model generated using Gradient Boosting was more accurate in fault classification than AdaBoost. The model developed from Gradient Boosting accurately predicts fault types in real-time.

Field of Study: Electrical Engineering

Student's Signature

Academic Year: 2022

Advisor's Signature

Co-advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากผู้วิจัยได้รับการช่วยเหลือ การให้คำแนะนำ ตลอดจนการสนับสนุนเป็นอย่างดีมาโดยตลอดของ ศาสตราจารย์ ดร.วาทีต เบญจพลกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.ชาญณรงค์ บาลมงคล ประการกรมการ รองศาสตราจารย์ ดร.สุรชัย ชัยทัศน์ และรองศาสตราจารย์ ดร.ภูมิพัฒน์ แสงอุดมเลิศ ที่ให้เกียรติมาเป็นกรรมการสอบวิทยานิพนธ์ และชี้แนะแนวทางอันเป็นประโยชน์ต่องานวิจัยนี้

ขอขอบคุณการไฟฟ้าฝ่ายผลิตแห่งประเทศไทย ที่ได้อนุเคราะห์สถานที่ทำการทดลอง ณ โรงไฟฟ้าพลังงานแสงอาทิตย์ อ.ทับสะแก จ.ประจวบคีรีขันธ์ ตลอดจนบุคลากรที่ประจำที่โรงไฟฟ้าดังกล่าวที่ให้การช่วยเหลือและคำแนะนำต่างๆ ที่เป็นประโยชน์ต่องานวิจัย

ขอขอบคุณ คุณวัฒน์ศักดิ์ ศรีศิริ นิสิตระดับปริญญาตรีบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ความช่วยเหลือ คำแนะนำ ชี้แนะแนวทางแก้ไข ปัญหาตั้งแต่ผู้วิจัยเริ่มทำวิจัยจนงานวิจัยสำเร็จลุล่วง

สุดท้ายนี้ผู้วิจัยขอกราบขอบพระคุณครอบครัวของผู้วิจัยที่ให้การสนับสนุนตลอดการศึกษา และเป็นกำลังใจสำคัญให้ผู้วิจัยมาโดยตลอด



เสถียรวุฒิ ยิวเหียง

สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญรูป.....	ฎ
บทที่ 1	1
บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 การทบทวนวรรณกรรม.....	2
1.3 วัตถุประสงค์ของวิทยานิพนธ์.....	8
1.4 ขอบเขตของวิทยานิพนธ์.....	9
1.5 ขั้นตอนการดำเนินการ.....	9
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	10
บทที่ 2	11
ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง	11
2.1 ความผิดพร่องในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์.....	11
2.1.1 ความผิดพร่องทางไฟฟ้าที่อาร์เรย์เซลล์แสงอาทิตย์.....	13
2.1.2 ความผิดพร่องที่เกิดจากสิ่งแวดล้อม.....	15

2.1.3 ความผิดปกติของทางกายภาพ.....	17
2.2 วิธีการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์	17
2.2.1 Real-time Difference Measurement	18
2.2.2 Output Signal Analysis	18
2.2.3 Machine Learning.....	19
2.3 อัลกอริทึม Machine Learning.....	20
2.3.1 ต้นไม้ตัดสินใจ (Decision Tree).....	21
2.3.2 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine: SVM)	23
2.3.4 การหาเพื่อนบ้านใกล้ที่สุด (k-Nearest Neighbors: k-NN)	27
2.3.5 โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)	29
2.3.6 ป่าสุ่ม (Random Forest: RF)	31
2.3.7 Adaptive Boosting (Adaboost)	31
2.3.8 Gradient Boosting (GB)	33
2.4 การประเมินความถูกต้องของตัวจำแนกข้อมูล.....	33
2.4.1 ตัวชี้วัดความถูกต้องจากการจำแนกข้อมูล	34
2.4.2 การประเมินความถูกต้องของตัวจำแนกข้อมูล	36
2.5 การหา Hyperparameter ที่เหมาะสม.....	37
2.6 เทคโนโลยีสื่อสารแบบไร้สาย.....	39
2.6.1 Long Range (LoRa).....	39
บทที่ 3	40
การออกแบบและพัฒนาระบบตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงาน แสงอาทิตย์.....	40
3.1 ภาพรวมขั้นตอนการทำงานของวิธีการที่เสนอ	42
3.2 รายละเอียดการทำงานของระบบตรวจจับและระบุประเภทความผิดปกติ	43

3.3 การจำลองความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์	44
3.4 การเก็บรวบรวมข้อมูลสำหรับนำมาฝึกและทดสอบ Machine Learning.....	49
3.4.1 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้า.....	49
3.4.2 การวัดกระแสไฟฟ้าและแรงดันไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์	51
3.4.3 การส่งข้อมูลด้วยเทคโนโลยีสื่อสารแบบไร้สาย LoRa.....	52
3.5 การฝึก Machine Learning	55
3.6 การประเมินความแม่นยำของอัลกอริทึมและแนวทางการทดสอบโมเดลตรวจจับและระบุ	
ประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์	57
3.6.1 การประเมินความแม่นยำของอัลกอริทึม.....	57
3.6.2 แนวทางการทดสอบโมเดล	57
บทที่ 4	60
การทดสอบและผลการวิจัย	60
4.1 การทดสอบความสามารถของ AdaBoost และ Gradient Boosting ในการพัฒนาอัลกอริทึม	
ที่อ่อนแอให้มีความแม่นยำเพิ่มมากขึ้น.....	60
4.2 การทดสอบเปรียบเทียบความแม่นยำระหว่าง AdaBoost กับ Gradient Boosting ในการ	
ตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์.....	62
4.3 การทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้า	
พลังงานแสงอาทิตย์ของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก.....	64
4.4 การนำอัลกอริทึมที่เลือกมาทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิด	
พ้อรงด้วยชุดข้อมูลทดสอบ	65
4.5 การทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดพลาดใน	
ระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์.....	68
4.6 สรุปผลการทดสอบ	71
บทที่ 5	73
สรุปผลการวิจัยและข้อเสนอแนะ.....	73

5.1 สรุปผลการวิจัย.....	73
5.2 ข้อเสนอแนะงานวิจัยในอนาคต.....	74
บรรณานุกรม.....	76
ประวัติผู้เขียน.....	79



สารบัญตาราง

	หน้า
ตารางที่ 1 ประเภทของการบังเงาบางส่วน	16
ตารางที่ 2 ผลการจำแนกประเภทของข้อมูลตัวอย่าง x โดยที่มีข้อมูลตัวอย่างที่ทราบประเภท 9 ตัวอย่าง [12].....	28
ตารางที่ 3 ตัวอย่าง Confusion matrix สำหรับ Positive tuples และ Negative tuples [22]... 35	
ตารางที่ 4 ข้อมูลพิกัดทางไฟฟ้าของแผงเซลล์แสงอาทิตย์	44
ตารางที่ 5 การกระจายของข้อมูลความผิดพลาดชุดฝึกในแต่ละคลาส.....	56
ตารางที่ 6 การกระจายของข้อมูลความผิดพลาดชุดทดสอบในแต่ละคลาส	58
ตารางที่ 7 การกระจายของข้อมูลความผิดพลาดชุดทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์	59
ตารางที่ 8 Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุดเมื่อกำหนดให้ความลึกของ Decision Tree เท่ากับ 1.....	60
ตารางที่ 9 ผลการทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เมื่อกำหนดให้แต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1	61
ตารางที่ 10 Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุด	62
ตารางที่ 11 ผลการทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้ AdaBoost และ Gradient Boosting.....	63
ตารางที่ 12 ผลการทดสอบความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก	64
ตารางที่ 13 ประสิทธิภาพในการจำแนกประเภทของ Gradient Boosting ด้วยชุดข้อมูลทดสอบ... 68	
ตารางที่ 14 ผลการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์.....	70

สารบัญรูป

	หน้า
รูปที่ 1 ความผิดพลาดที่เกิดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ [2]	12
รูปที่ 2 ตัวอย่างการเกิดความผิดพลาดแบบลัดวงจรระหว่างสายในสตริง [2]	14
รูปที่ 3 ตัวอย่างการเกิดความผิดพลาดแบบลัดวงจรระหว่างสายข้ามสตริง [2]	14
รูปที่ 4 ตัวอย่างความผิดพลาดแบบเปิดวงจร [10]	15
รูปที่ 5 ตัวอย่างการบังเงาบางส่วน [2]	16
รูปที่ 6 เทคนิค RDM สำหรับตรวจจับความผิดพลาดในระบบเซลล์แสงอาทิตย์ [2]	18
รูปที่ 7 เทคนิค OSA สำหรับการตรวจจับความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ [2]	19
รูปที่ 8 เทคนิค ML สำหรับการตรวจจับความผิดพลาดในระบบเซลล์แสงอาทิตย์ [2]	20
รูปที่ 9 รูปแบบของต้นไม้ตัดสินใจ [12]	21
รูปที่ 10 ตัวอย่างต้นไม้ตัดสินใจแบบไบนารี [12]	22
รูปที่ 11 ตัวอย่างต้นไม้ตัดสินใจแบบหลายคลาส [12]	22
รูปที่ 12 ตัวอย่างระนาบเกินเชิงเส้นแบ่งข้อมูลเป็น 2 คลาส ในปริภูมิ 2 มิติ [12]	23
รูปที่ 13 ตัวอย่างบางส่วนของระนาบเกินที่สามารถแบ่งแยกข้อมูลเป็น 2 คลาสได้ [12]	24
รูปที่ 14 ระยะขอบและเวกเตอร์ซัพพอร์ต [12]	24
รูปที่ 15 แสดงอสมการของพื้นที่ในแต่ละคลาส [12]	26
รูปที่ 16 กราฟโลจิสติกฟังก์ชัน [13]	27
รูปที่ 17 ตัวอย่างการกำหนดค่า k [12]	28
รูปที่ 18 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบ Perceptron [12]	30
รูปที่ 19 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบหลายชั้น [12]	30
รูปที่ 20 โครงสร้างของป่าสุ่ม	31
รูปที่ 21 ตัวอย่างการจำแนกประเภทแบบ 2 คลาสด้วย Adaboost [19]	32

รูปที่ 22 ตัวอย่างการจำแนกประเภทด้วย Gradient Boosting [20].....	33
รูปที่ 23 การประเมินความถูกต้องด้วยวิธี Holdout [22]	36
รูปที่ 24 ตัวอย่างการประเมินความถูกต้องด้วยวิธี Cross-validation โดยกำหนดให้ $k = 5$	37
รูปที่ 25 การจำลองรูปแบบการทำงานของวิธี Grid Search [24].....	38
รูปที่ 26 การจำลองรูปแบบการทำงานของวิธี Random Search [24].....	38
รูปที่ 27 โครงสร้างของ LoRaWAN พร้อมทั้งแอปพลิเคชัน [25].....	39
รูปที่ 28 ภาพรวมขั้นตอนการทำงานของวิธีการที่เสนอ	42
รูปที่ 29 ขั้นตอนการทำงานของระบบตรวจจับและระบุประเภทความผิดปกติ	43
รูปที่ 30 แผงเซลล์แสงอาทิตย์ชนิด Monocrystalline ขนาด 30 W	45
รูปที่ 31 การจำลองความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์: การลัดวงจรระหว่างสายใน สตริง (a), การเปิดวงจร (b), การบังเงาบางส่วน (C), การเสื่อมสภาพของอุปกรณ์ (d), การลัดวงจร ระหว่างสายข้ามสตริง (e).....	46
รูปที่ 32 ระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง. 47	
รูปที่ 33 ตัวอย่างการจำลองความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จริง.....	48
รูปที่ 34 เซนเซอร์วัดกระแสไฟฟ้า	50
รูปที่ 35 เซนเซอร์วัดแรงดันไฟฟ้า	50
รูปที่ 36 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้า	50
รูปที่ 37 การต่อเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าเข้ากับระบบผลิตไฟฟ้าพลังงาน แสงอาทิตย์.....	52
รูปที่ 38 การส่งข้อมูลทางไฟฟ้าจากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์มายังศูนย์ควบคุม	53
รูปที่ 39 Dragino LoRaWAN Gateway LG308-AS923-TH-EC25 รุ่น 4G.....	53
รูปที่ 40 ตัวอย่าง End device ที่กำหนดไว้บนหน้าเว็บของ The Things Network	54
รูปที่ 41 ตัวอย่างกราฟของกระแสไฟฟ้าและแรงดันไฟฟ้าที่แสดงบน IoT Analytics-ThingSpeak Internet of Things	54

รูปที่ 42 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้าที่มี LoRa-RFM Shield ทำหน้าที่ส่งข้อมูลแบบไร้สาย 55

รูปที่ 43 ตัวอย่างข้อมูลที่ใช้ในการฝึกอัลกอริทึม 56

รูปที่ 44 แผนภูมิแสดงความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เมื่อกำหนดให้แต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1..... 61

รูปที่ 45 แผนภูมิแสดงความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้ AdaBoost และ Gradient Boosting 63

รูปที่ 46 กราฟแสดงผลการทดสอบความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก 65

รูปที่ 47 Confusion matrix แสดงผลการจำแนกความผิดปกติของชุดข้อมูลทดสอบด้วยการใช้ Gradient Boosting..... 66

รูปที่ 48 การบันทึกอัลกอริทึมที่ดีที่สุดเพื่อเป็นโมเดลสำหรับนำไปใช้งาน 69

รูปที่ 49 การเรียกโมเดลที่บันทึกไว้มาใช้งาน 69

รูปที่ 50 ขั้นตอนการทำงานของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ 69

รูปที่ 51 Confusion matrix แสดงผลการจำแนกความผิดปกติของชุดข้อมูลทดสอบการทำงานได้แบบเรียลไทม์ด้วยการใช้โมเดล Gradient Boosting 70

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากการตระหนักถึงสถานะโลกร้อนและความต้องการลดการใช้พลังงานจากฟอสซิลที่เป็นพลังงานดั้งเดิมในการผลิตไฟฟ้า ความนิยมในการใช้พลังงานหมุนเวียนจึงเพิ่มมากขึ้นโดยเฉพาะการใช้พลังงานแสงอาทิตย์ที่เติบโตอย่างรวดเร็วในช่วงทศวรรษที่ผ่านมา มีการสร้างโรงไฟฟ้าพลังงานแสงอาทิตย์ขนาดใหญ่ขึ้นมากมายทั่วโลกเพื่อลดภาระการผลิตไฟฟ้าด้วยพลังงานแบบดั้งเดิม จึงเป็นสาเหตุให้มีการศึกษาเกี่ยวกับการผลิตกระแสไฟฟ้าด้วยพลังงานแสงอาทิตย์อย่างจริงจังและแพร่หลาย ไม่ว่าจะเป็นการศึกษาด้านการปรับปรุงประสิทธิภาพของแผงเซลล์แสงอาทิตย์เพื่อพัฒนาแผงเซลล์แสงอาทิตย์ให้มีประสิทธิภาพสูงในราคาที่ถูกลง หรือการศึกษาเพื่อหาวิธีดึงพลังงานสูงสุดจากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยประเทศไทยมีเป้าหมายในการเพิ่มสัดส่วนการผลิตไฟฟ้าจากพลังงานหมุนเวียนด้วยเช่นกันจากการกำหนดแผนพัฒนากำลังผลิตไฟฟ้าของประเทศไทย พ.ศ. 2561-2580 ฉบับปรับปรุงครั้งที่ 1 (PDP2018 Revision 1) [1] ซึ่งมีเป้าหมายที่จะใช้พลังงานหมุนเวียนในการผลิตไฟฟ้าเป็นร้อยละ 26.3 ในปี 2575 โดยมีพลังงานแสงอาทิตย์เป็นหนึ่งในพลังงานหมุนเวียนหลัก มีวัตถุประสงค์เพื่อรองรับการเติบโตทางเศรษฐกิจของประเทศรวมถึงจำนวนประชากรที่เพิ่มขึ้น อีกทั้งยังเป็นแผนการสนับสนุนให้ประชาชนเข้าถึงพลังงานไฟฟ้าได้อย่างทั่วถึงในราคาที่เหมาะสม

อย่างไรก็ตามการผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ก็ยังมี ความท้าทายในการรวมเข้ากับระบบไฟฟ้าที่ใช้พลังงานแบบดั้งเดิมซึ่งเป็นระบบไฟฟ้าหลัก เนื่องจากความไม่แน่นอนของกำลังไฟฟ้าที่ได้จากพลังงานแสงอาทิตย์รวมถึงความเสี่ยงในการเกิดความผิดปกติ (Fault) ด้วยเหตุนี้จึงต้องมีการศึกษาเพื่อปรับปรุงประสิทธิภาพและปรับปรุงความเชื่อถือได้ในการผลิตไฟฟ้าจากพลังงานแสงอาทิตย์ ดังนั้นการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จึงเป็นหนึ่งในปัจจัยสำคัญที่จะต้องพิจารณา โดยทั่วไปโรงไฟฟ้าพลังงานแสงอาทิตย์ส่วนใหญ่จะมีระบบตรวจจับความผิดปกติทางฝั่งไฟฟ้ากระแสสลับที่มีประสิทธิภาพ แต่ระบบตรวจจับความผิดปกติทางฝั่งไฟฟ้ากระแสตรงยังต้องได้รับการพัฒนาเนื่องจากยังมีข้อจำกัดบางประการ [2] เช่น กระแสไฟฟ้าจากบางชนิดความผิดปกติต่ำกว่าที่อุปกรณ์ตรวจจับกระแสเกินจะตรวจจับได้ หรือตรวจไม่พบการที่แผงเซลล์แสงอาทิตย์ถูกบดบังการได้รับรังสีแสงอาทิตย์ เป็นต้น ซึ่งความผิดปกติที่เกิดขึ้นที่ฝั่งไฟฟ้ากระแสตรงของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์สามารถส่งผลกระทบต่อกำลังการผลิตไฟฟ้าหรือทำ

ให้ระบบเสียหายได้ ด้วยเหตุนี้การตรวจจับและระบุประเภทความผิดปกติของพลังงานไฟฟ้ากระแสตรงในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จึงเป็นหนึ่งในปัจจัยสำคัญที่จะต้องพิจารณาและพัฒนาให้มีความแม่นยำเพื่อสามารถแก้ไขความผิดปกติที่เกิดขึ้นได้อย่างรวดเร็ว

1.2 การทบทวนวรรณกรรม

จากการศึกษางานวิจัยในอดีตทำให้ทราบถึงแนวทางในการพัฒนาระบบตรวจจับและระบุประเภทความผิดปกติที่เกิดขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วยการใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นแหล่งข้อมูลอ้างอิงและเป็นประโยชน์สำหรับการพัฒนาระบบตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยงานวิจัยและวรรณกรรมที่เกี่ยวข้องมีดังนี้

Andre Eugenio Lazzaretti และคณะ [3] เสนอวิธีการตรวจจับการเกิดความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วยเทคนิค Real-time Difference Measurement (RDM) ซึ่งเป็นการเปรียบเทียบผลต่างของกำลังไฟฟ้าที่ผลิตได้จากโรงไฟฟ้าจริงกับกำลังไฟฟ้าจากโรงไฟฟ้าทางทฤษฎี โดยงานวิจัยนี้จำลองระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 5 kW ที่ติดตั้งแบบ 2 สตริง สตริงละ 8 แผงขึ้นบนโปรแกรม PSIM ส่วนตัวแปรด้านสิ่งแวดล้อม เช่น ความเข้มแสงอาทิตย์และอุณหภูมิสภาพแวดล้อมถูกควบคุมด้วย Matlab/Simulink โดยสำหรับวิธีการตรวจจับความผิดปกติจะนำค่ากำลังไฟฟ้าจากโรงไฟฟ้าที่จำลองขึ้นเปรียบเทียบเพื่อหาผลต่างกับกำลังไฟฟ้าที่ได้จากโรงไฟฟ้าจริง ณ เวลาเดียวกัน ซึ่งโรงไฟฟ้าจำลองใช้อัลกอริทึม Recursive Least Squares (RLS) ในการอัปเดตค่าพารามิเตอร์ใหม่ตามเวลาจริง โดยหากผลต่างของกำลังไฟฟ้าเกินค่าเกณฑ์ที่กำหนดไว้ ระบบจะแจ้งเตือนว่าเกิดความผิดปกติ จากผลการทดสอบวิธีตรวจจับความผิดปกติที่เสนอให้ความแม่นยำ 93.09% นอกจากนี้ยังมีการจำแนกประเภทความผิดปกติที่เกิดว่าเป็นการเปิดวงจร (Open-Circuit) การลัดวงจร (Short-Circuit) การบังเงาบางส่วน (Partial Shading) หรือความเสื่อมสภาพของอุปกรณ์ (Degradation) ด้วยเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) โดยอัลกอริทึม Artificial Neural Network (ANN) ที่มีชั้นซ่อน (Hidden Layer) เดียวและมีกระบวนการเรียนรู้แบบ backpropagation มีความแม่นยำสูงสุดเมื่อเทียบกับ k-Nearest Neighbors (KNN), Decision Tree และ Support Vector Machine (SVM) ภายใต้พารามิเตอร์ 4 ชนิด ได้แก่ ความเข้มแสงอาทิตย์ อุณหภูมิสิ่งแวดล้อม แรงดันไฟฟ้าแต่ละสตริง และกระแสไฟฟ้าแต่ละสตริง โดยข้อมูลชุดฝึกอัลกอริทึมได้มาจากโรงไฟฟ้าจำลอง ส่วนข้อมูลชุดทดสอบเป็นข้อมูลจากโรงไฟฟ้าจริง การจำแนกประเภทความผิดปกติด้วย ANN ให้ความแม่นยำ 95.44% สรุปคืองานวิจัยนี้นำเทคนิคตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ 2 เทคนิคมาผสมผสานกัน โดยขั้นตอน

การตรวจจับความผิดพลาดโดยใช้เทคนิค RDM และระบุประเภทความผิดพลาดที่ตรวจพบด้วยเทคนิค Machine Learning ระบบที่เสนอมีความแม่นยำรวมในการตรวจจับและจำแนกประเภทความผิดพลาด 92.64%

Elyes Garoudja และคณะ [4] เสนอเทคนิค Machine Learning ด้วยการใช้อัลกอริทึม Probabilistic neural network (PNN) เพื่อมาตรวจจับและจำแนกประเภทของความผิดพลาดทางด้านไฟฟ้ากระแสตรงในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ที่เชื่อมต่อกับโครงข่ายไฟฟ้า ความผิดพลาดที่พิจารณาในการจำแนกประเภทของงานวิจัยนี้ได้แก่ การลัดวงจร 3 แผลง การลัดวงจร 10 แผลง และการเปิดวงจรของสตริง โดยสร้างโมเดล PNN สำหรับตรวจจับและจำแนกประเภทความผิดพลาดจากข้อมูลการเกิดความผิดพลาดที่ได้จากระบบไฟฟ้าที่จำลองขึ้นด้วยโปรแกรม PSIM และใช้โปรแกรม Matlab คำนวณพารามิเตอร์ทางไฟฟ้า ซึ่งระบบไฟฟ้าจำลองดังกล่าวเป็นแบบจำลองจากระบบจริงขนาด 9.54 kW ซึ่งค่าพารามิเตอร์ของแผงเซลล์แสงอาทิตย์ที่จำเป็นต่อการสร้างแผงเซลล์แสงอาทิตย์จำลองไม่สามารถวัดได้จากแผงจริง จึงต้องสกัดค่าให้ใกล้เคียงกับแผงจริงมากที่สุดด้วยการนำค่าอุณหภูมิและแรงดันไฟฟ้าของแผงเซลล์แสงอาทิตย์จากระบบจริงมาสกัดพารามิเตอร์ที่ต้องการสำหรับสร้างแบบจำลองด้วยโมเดลแผงเซลล์แสงอาทิตย์ไดโอดเดี่ยวและใช้อัลกอริทึม best-so-far artificial bee colony (ABC) ในการลดความต่างระหว่างค่าพารามิเตอร์จากแผงจำลองกับแผงจริง โดยพารามิเตอร์จากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จำลองที่ใช้สำหรับฝึกโมเดล PNN ประกอบไปด้วย กระแสไฟที่จุดกำลังสูงสุดและแรงดันไฟฟ้าที่จุดกำลังสูงสุด ส่วนความเข้มแสงอาทิตย์และอุณหภูมิแผงได้จากการวัดระบบจริง โมเดล PNN ถูกสร้างมา 2 โมเดล โมเดลแรกสำหรับตรวจจับความผิดพลาด หากตรวจจับพบความผิดพลาดโมเดลที่สองก็จะระบุชนิดของความผิดพลาดที่ตรวจพบ โดยทั้งสองโมเดลเป็น PNN ที่มีชั้นซ่อน 1 ชั้น การทดสอบประสิทธิภาพของระบบที่นำเสนอแยกเป็น 2 กรณี 1. การตรวจสอบและการจำแนกประเภทของความผิดพลาดขณะไม่มีสัญญาณรบกวน ซึ่ง PNN มีประสิทธิภาพในการตรวจสอบความผิดพลาด 100% และมีประสิทธิภาพในการจำแนกประเภทของความผิดพลาดแม่นยำ 100% 2. การตรวจสอบและจำแนกประเภทของความผิดพลาดขณะมีสัญญาณรบกวน สัญญาณรบกวน เช่น ความคลาดเคลื่อนของเครื่องมือวัดและเซนเซอร์ ซึ่ง PNN มีประสิทธิภาพในการตรวจสอบความผิดพลาด 82.34% และมีประสิทธิภาพในการจำแนกชนิดของความผิดพลาดแม่นยำ 98.19% โดยในการทดสอบทุกกรณีได้มีการเปรียบเทียบกับวิธี ANN โดยในงานวิจัยนี้ PNN มีประสิทธิภาพในการตรวจสอบความผิดพลาดและจำแนกประเภทความผิดพลาดแม่นยำกว่า ANN ทุกกรณี แต่มีข้อจำกัดของวิธีที่เสนอคือไม่สามารถตรวจสอบความผิดพลาดแบบ

ลัดวงจรแผงเดียวสำหรับโรงไฟฟ้าขนาดใหญ่ได้ เนื่องจากสัญญาณมีพฤติกรรมไม่ต่างจากสภาวะการทำงานปกติมากพอที่จะตรวจจับ

Barun Basnet และคณะ [5] เสนอวิธีการจำแนกประเภทความผิดปกติทางไฟฟ้า 2 ประเภทออกจากสภาวะการทำงานปกติ ได้แก่ ความผิดปกติแบบเปิดวงจร และความผิดปกติแบบลัดวงจรระหว่างสาย ด้วยการใช้อัลกอริทึม PNN ความผิดปร้องดังกล่าวถูกพิจารณาบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 1.8 kW งานวิจัยนี้แสดงให้เห็นว่าลักษณะของสัญญาณกระแสไฟฟ้าในช่วงฤดูหนาวนั้นเป็นปัญหาในการตรวจจับความผิดปกติประเภทลัดวงจรระหว่างสาย เนื่องจากในช่วงนี้กระแสไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ในสภาวะการทำงานปกติช่วงที่มีเมฆมากกับสภาวะการเกิดลัดวงจรระหว่างสายมีลักษณะใกล้เคียงกันมาก และมีค่ากระแสต่ำกว่าในช่วงฤดูร้อน ทำให้ระบบตรวจจับความผิดปกติแบบทั่วไปที่ใช้การตรวจจับกระแสเกินของระบบหรือระบบตรวจจับความผิดปกติที่พัฒนามาด้วยเทคนิค Machine Learning แต่ไม่ได้พิจารณาพารามิเตอร์ทางสภาพแวดล้อมทำงานผิดพลาดบ่อยครั้ง ดังนั้นการพิจารณาพารามิเตอร์ทางสภาพแวดล้อมจึงเป็นการยกระดับการตรวจจับความผิดปกติด้วยเทคนิค Machine Learning ให้มีประสิทธิภาพมากยิ่งขึ้น โดยพารามิเตอร์ที่เป็นอินพุตให้โมเดลตรวจจับความผิดปกติที่สร้างจาก PNN ได้แก่ กระแสไฟฟ้าของสตริงที่ 1, กระแสไฟฟ้าของสตริงที่ 2, แรงดันไฟฟ้าสตริงที่ 1, แรงดันไฟฟ้าของสตริงที่ 2, ความเข้มแสงอาทิตย์, อุณหภูมิแผงเซลล์ และเงื่อนไขจากสภาพอากาศ (สาดใส ทึบมาก เมฆมาก และฝนตก) ในขั้นตอนการเตรียมข้อมูลมีการนำเทคนิคการทำเหมืองข้อมูล (Data-mining) มาใช้เพื่อคัดกรองข้อมูลก่อนนำไปฝึก PNN สำหรับโครงสร้างของ PNN นั้นมีชั้นซ่อน 2 ชั้น แต่ละชั้นซ่อนมี 12 ปม ใช้ฟังก์ชัน Rectified Linear Unit (ReLU) ในการกำหนดเอาต์พุตของเซลล์ประสาทเทียม และใช้ Adaptive Moment Estimation (Adam) ในการปรับค่าอัตราการเรียนรู้ของแต่ละพารามิเตอร์ ผลการทดสอบแสดงให้เห็นว่าโมเดลตรวจจับความผิดปกติที่เสนอมีความแม่นยำในการทำนายข้อมูลชุดทดสอบ 100% ซึ่งมากกว่าหลายงานวิจัยในอดีต

Muhammed Hussain และคณะ [6] เสนอวิธีการตรวจจับความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 2.2 kW ที่เป็นระบบแบบสตริงเดี่ยว 10 แผง ความผิดปกติที่พิจารณาคือ การหลุดการเชื่อมต่อของแผงเซลล์แสงอาทิตย์จากสตริงตั้งแต่ 1 แผง ไปจนกระทั่ง 9 แผง โดยใช้อัลกอริทึม ANN ที่มีโครงสร้างแบบ Radial Basis Function (RBF) ซึ่งเป็น ANN ที่มีชั้นซ่อนเพียงชั้นเดียว จุดเด่นของงานวิจัยนี้คือการใช้พารามิเตอร์อินพุตแค่ 2 ชนิด ได้แก่ ความเข้มแสงอาทิตย์ และกำลังไฟฟ้าเอาต์พุต ซึ่งข้อดีของพิจารณาพารามิเตอร์น้อยคือมีความง่ายในการจัดการข้อมูลก่อนนำไปฝึกอัลกอริทึม วิธีการ Quick prorogation ถูกนำมาใช้เพื่อเพิ่มประสิทธิภาพ

การทำงานของ ANN จากการทดสอบเพื่อหาปมซ่อนที่เหมาะสมกับชุดข้อมูลปรากฏว่า 10 ปมซ่อนเหมาะสมที่สุด สำหรับขั้นตอนการเตรียมข้อมูลก่อนนำไปฝึก ANN งานวิจัยนี้ได้ใช้วิธีสุ่มเลือกค่าความเข้มแสงอาทิตย์และกำลังไฟฟ้าเอาต์พุตมา หลังจากนั้นใช้วิธี Normalization ในการจัดการข้อมูลเพื่อลดความซ้ำซ้อนของข้อมูล แต่การใช้ Normalization อย่างเดียวอาจจะทำให้บางข้อมูลที่ใช้ได้หายไป ดังนั้นจึงใช้การ Mapping ความเข้มแสงอาทิตย์กับกำลังไฟฟ้าเอาต์พุตร่วมด้วย โดยใช้ความเข้มแสงในช่วง 0-1,000 W/m² นอกจากนี้งานวิจัยนี้ยังนำ ANN ที่มีโครงสร้างแบบ RBF ไปเทียบเทียบประสิทธิภาพกับ ANN ที่มีโครงสร้างแบบหลายชั้นซ่อนหรือที่เรียกว่า Multi-Layer Perceptron (MLP) โดยผลลัพธ์แสดงให้เห็นว่า RBF มีความแม่นยำน้อยกว่า MLP ที่มี 3 ชั้นซ่อนแค่ 0.5% แต่ใช้เวลาในการทำงานน้อยกว่ามาก โดย RBF ใช้เวลา 45 ms ส่วน MLP ใช้เวลาไป 290 ms โดยผู้เขียนงานวิจัยนี้ให้แนวคิดว่า การจะเลือกกระหว่าง RBF กับ MLP นั้นควรพิจารณาเปรียบเทียบ 3 องค์ประกอบร่วมกันคือ วัตถุประสงค์ของงาน ความแม่นยำ และเวลาที่ใช้ในการประมวลผล สำหรับบทสรุปที่งานวิจัยนี้เลือกใช้ RBF เนื่องจากความแม่นยำที่ไม่ต่างกันมาก แต่ใช้เวลาในการประมวลผลเร็วกว่ามาก ในส่วนสุดท้ายเป็นการทดสอบการตรวจจับความผิดปกติของโมเดลที่ผ่านการฝึกมาแล้ว โดยมีการทดสอบที่ 3 สภาพอากาศ ได้แก่ สภาพอากาศปกติ สภาพอากาศที่มีเมฆเป็นบางส่วน และสภาพอากาศที่มีดีคริม ผลการทดสอบปรากฏว่าที่สภาพอากาศปกติโมเดลมีความแม่นยำ 98.1% ส่วนที่สภาพอากาศมีเมฆบางส่วนโมเดลมีความแม่นยำ 97.9% และที่สภาพอากาศมีดีคริมโมเดลมีความแม่นยำลดลงเหลือ 96.5% นอกจากนี้ยังมีการทดสอบความยืดหยุ่นในการใช้งานของโมเดลที่เสนอด้วยการไปทดสอบตรวจจับความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 4.16 kW ที่ติดตั้งแบบ 4 สตริงละ 8 แผง ผลการทดสอบแสดงให้เห็นว่าโมเดลที่เสนอก็กังให้มีความแม่นยำโดยรวมที่สูงมากกว่า 97%

Saeed Ansari และคณะ [7] เสนอวิธีการตรวจจับและระบุตำแหน่งความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ที่จำลองขึ้นในโปรแกรม Matlab/Simulink ขนาด 250 kW ที่เชื่อมต่อกับกริด โดยระบบติดตั้งแบบ 66 สตริง สตริงละ 10 แผง และใช้เซนเซอร์วัดแรงดัน 1 ตัวต่อ 1 สตริง งานวิจัยนี้แบ่งการทำงานของโมเดลที่เสนอเป็น 3 ส่วน ดังนี้ ส่วนแรกเป็นการตรวจจับความผิดปกติที่เกิดจากการลัดวงจรระหว่างสายภายในสตริง การลัดวงจรระหว่างสายข้ามสตริง การลัดวงจรระหว่างสายกับสายดิน และการเปิดวงจร ในส่วนแรกนี้ใช้เทคนิค Real-Time Difference Measurement (RDM) ในการตรวจจับความผิดปกติ โดยที่จะพิจารณาผลต่างระหว่างค่าเฉลี่ยเคลื่อนที่ (Moving average) ของแรงดันไฟฟ้า ณ เวลาปัจจุบันกับค่าเฉลี่ยเคลื่อนที่ของแรงดันไฟฟ้าในรอบก่อนหน้า โดยจะคิดค่าเฉลี่ยๆ ทุกๆ 1 ms หากผลต่างเกินเกณฑ์ที่กำหนดไว้จะถือว่าระบบเกิด

ความผิดพลาดขึ้น ส่วนที่สองเป็นการแยกการเกิดบังเงาบางส่วนออกจากความผิดพลาดประเภทอื่นๆ เนื่องจากวิธีการในส่วนแรกนั้นใช้ค่าแรงดันไฟฟ้าในการพิจารณา แต่เมื่อเกิดการบังเงาบางส่วนขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จะทำให้แรงดันไฟฟ้าของระบบเปลี่ยนเช่นกัน ด้วยเหตุนี้อาจทำให้วิธีในส่วนแรกแจ้งเตือนผิดพลาดได้ สำหรับวิธีการในการแยกการบังเงาบางส่วนออกจากความผิดพลาดที่พิจารณาใช้หลักการวิเคราะห์การเปลี่ยนแปลงของแรงดัน โดยหากเป็นความผิดพลาดที่พิจารณาแรงดันไฟฟ้าจะเปลี่ยนทันทีทันใด แต่การเกิดบังเงาบางส่วนแรงดันไฟฟ้าจะค่อยๆ เปลี่ยนและเมื่อผ่านไปสักครู่แรงดันก็จะเปลี่ยนอีกครั้ง โดยในช่วงระยะนี้ความผิดพลาดที่พิจารณาจะใช้เวลาเปลี่ยนไปค่าน้อยกว่าการบังเงาบางส่วน โดยใช้วิธีการเปรียบเทียบค่าระหว่าง least square error ของสัญญาณแรงดันกับเกณฑ์ที่กำหนด หากน้อยกว่าค่าเกณฑ์ที่กำหนดจะถือว่าระบบเกิดบังเงาบางส่วน ส่วนสุดท้ายเป็นการเสนอวิธีการระบุตำแหน่งของแผงที่เกิดความผิดพลาดในสตริง โดยใช้คุณลักษณะเด่นที่สกัดได้จากการแปลงเวฟเล็ต (Wavelet transform) ของสัญญาณแรงดันไฟฟ้าเป็นอินพุตในการฝึกอัลกอริทึม ANN โดยพิจารณาเงื่อนไขการเกิดความผิดพลาดในสตริง เช่น การเกิดลัดวงจรระหว่างสายในสตริง 2 แผง สามารถเกิดระหว่างไหนได้บ้าง จากนั้นนำคุณลักษณะเด่นของแต่ละเงื่อนไขไปให้ ANN เรียนรู้ สำหรับผลการทดสอบโมเดลที่เสนอปรากฏว่าการตรวจจับความผิดพลาดมีความแม่นยำ 99.14% การแยกการเกิดการบังเงาบางส่วนออกจากความผิดพลาดที่พิจารณามีความแม่นยำ 99% และการระบุตำแหน่งมีความแม่นยำ 98.41%

Sherif S. M. Ghoneim และคณะ [8] เสนอวิธีการตรวจจับความผิดพลาดในโรงไฟฟ้าพลังงานแสงอาทิตย์ขนาด 250 kW ที่ติดตั้งแบบ 88 สตริง สตริงละ 7 แผง ด้วยการใช้เทคนิค Machine Learning โดยชุดข้อมูลความผิดพลาดที่ใช้ฝึกและทดสอบอัลกอริทึมได้มาจากการจำลองระบบผลิตไฟฟ้าดังกล่าวด้วยโปรแกรม Matlab/Simulink ความผิดพลาดที่พิจารณาในงานวิจัยนี้ได้แก่ การลัดวงจรภายในสตริงที่ 1 การลัดวงจรระหว่างสตริงที่ 1 กับสายดิน และการลัดวงจรระหว่างสตริงที่ 1 กับสตริงที่ 2 โดยงานวิจัยนี้ได้เสนอวิธีการวัดกระแสของสตริงแบบใช้เซนเซอร์วัดที่หัวสตริง 1 ตัว และใช้เซนเซอร์วัดที่ท้ายสตริง 1 ตัว รวมถึงมีการวัดแรงดันไฟฟ้าของระบบกำลังไฟฟ้าของระบบ อุณหภูมิสภาพแวดล้อม ความเข้มแสงอาทิตย์ และความต้านทานขณะเกิดความผิดพลาด โดยความเข้มแสงอาทิตย์ถูกกำหนดให้อยู่ในช่วง 100 W/m^2 ถึง 1000 W/m^2 และอุณหภูมิอยู่ในช่วง $10 \text{ }^{\circ}\text{C}$ ถึง $35 \text{ }^{\circ}\text{C}$ การจำลองความผิดพลาดแต่ละครั้งจะใช้เวลา 0.4 s และจะวัดค่าพารามิเตอร์ในช่วงตั้งแต่ 0.2 s ถึง 0.4 s งานวิจัยนี้คัดเลือกคุณลักษณะเด่นจากพารามิเตอร์ต่างๆ ด้วยการเปรียบเทียบค่า Information Gain ปรากฏว่ามี 4 คุณลักษณะเด่นจากทั้งหมด 30 คุณลักษณะที่มีอิทธิพลสูงที่สุดในการตรวจจับและจำแนกประเภทความผิดพลาด ได้แก่ ผลต่างระหว่าง

ค่าสูงสุดกับค่าต่ำสุดของกระแสที่หัวสตริงที่ 1, ผลต่างระหว่างค่าสูงสุดกับค่าต่ำสุดของกระแสที่หัวสตริงที่ 2, ผลต่างระหว่างกระแสที่หัวกับกระแสที่ท้ายของสตริงที่ 1 และผลต่างระหว่างกระแสที่หัวกับกระแสที่ท้ายของสตริงที่ 2 โดยงานวิจัยนี้เปรียบเทียบประสิทธิภาพในการจำแนกประเภทของอัลกอริทึม 6 ประเภท ได้แก่ Random Forest, Ridge-Regression, Lasso-Regression, Naive Bayes, CN2 rule induction และ AdaBoost การสร้างโมเดลจากอัลกอริทึมดังกล่าวดำเนินการบนโปรแกรม Orange สำหรับการประเมินประสิทธิภาพของแต่ละอัลกอริทึมใช้วิธี 10-fold cross validation ผลการประเมินประสิทธิภาพด้วยชุดข้อมูลฝึกปรากฏว่า AdaBoost มีประสิทธิภาพสูงสุดโดยมีค่าความแม่นยำโดยรวมที่ 100% และเมื่อนำไปทดสอบด้วยข้อมูลชุดทดสอบ โมเดลให้ความแม่นยำ 95% รวมถึงค่าตัวบ่งชี้ประสิทธิภาพอื่นๆ เช่น Precision, Recall และ F1-score ของผลการจำแนกประเภทของโมเดลที่สร้างจาก AdaBoost ก็สูงกว่าทุกอัลกอริทึม

Ceyhun Kapucu และคณะ [9] เสนอการตรวจจับความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วย Machine Learning ที่ใช้เทคนิค Ensemble Learning (EL) โดยงานวิจัยจำลองความผิดพลาดบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จริง ที่ติดตั้งแบบ 2 สตริง สตริงละ 2 แผง ซึ่งเป็นระบบขนาดเล็กที่ติดตั้งเพื่อการดำเนินงานวิจัย โดยเทคนิค EL คือการนำอัลกอริทึมทาง Machine Learning มาเรียนรู้ร่วมกันทั้งแบบที่เป็นอัลกอริทึมชนิดเดียวกันหรือต่างชนิดกันก็ได้ EL แบ่งตามลักษณะของอัลกอริทึมย่อยที่นำมาเรียนรู้ได้ 2 ประเภท ได้แก่ Heterogeneous ensemble เป็นการรวมกันของอัลกอริทึมย่อยต่างชนิดกันหรืออัลกอริทึมย่อยชนิดเดียวกันแต่มีการตั้งค่าพารามิเตอร์ต่างกันมาเรียนรู้ด้วยข้อมูลชุดเดียวกัน ประเภทที่ 2 เรียกว่า Homogeneous ensemble เป็นการรวมกันของอัลกอริทึมย่อยที่เหมือนกัน โดยที่อัลกอริทึมย่อยแต่ละตัวจะเรียนรู้ด้วยชุดข้อมูลย่อยที่ต่างกัน นอกจากนี้ประเภทของ EL ยังแบ่งได้ตามวิธีการสร้างโมเดลได้ 2 แบบ แบบแรกเรียกว่า Averaging methods เป็นการรวมกันของอัลกอริทึมย่อยที่เหมือนกันหรือต่างกันได้ โดยต่อละอัลกอริทึมจะอิสระต่อกัน โดยผลการทำนายจะมาจากการเฉลี่ยผลทำนายของทุกอัลกอริทึมย่อย ตัวอย่างของอัลกอริทึมที่สร้างจากวิธีนี้ได้แก่ Random Forest, Extra Trees และอัลกอริทึมที่ทำนายผลด้วยวิธีการ voting เป็นต้น แบบที่สองเรียกว่า Boosting methods เป็นการรวมกันของอัลกอริทึมย่อย โดยที่อัลกอริทึมย่อยแต่ละตัวจะทำงานต่อกันด้วยการปรับค่าน้ำหนักของรายการข้อมูลที่อัลกอริทึมย่อยก่อนหน้าทำนายผิด โดยอัลกอริทึมย่อยไหนทำนายผิดมาก ก็จะมีน้ำหนักคะแนนในการทำนายน้อย ตัวอย่างของอัลกอริทึมที่สร้างจากวิธีนี้เช่น AdaBoost, Gradient Boosting และ Extreme Gradient Boosting เป็นต้น นอกจากนี้งานวิจัยนี้ยังกล่าวถึงการเตรียมชุดข้อมูลสำหรับนำมาฝึก Machine Learning ว่าควรเลือกชุดข้อมูลที่เกี่ยวข้องกับประเภทของความผิด

พร้อมที่พิจารณาให้มากที่สุดและตัดข้อมูลที่ไมเกี่ยวข้องออกไป เพื่อลดปัญหาการที่โมเดลเกิด Overfitting คือการที่โมเดลทำนายข้อมูลชุดฝึกได้แม่นยำมาก แต่พอเจอข้อมูลใหม่โมเดลมีประสิทธิภาพต่ำ และปัญหา Underfitting คือการที่โมเดลเรียนรู้ชุดข้อมูลฝึกได้ต่ำและทำนายผิดพลาดสูงตั้งแต่ชุดข้อมูลฝึก โดยงานวิจัยนี้พิจารณาความผิดพลาด 3 แบบ ได้แก่ การบังเงาบางส่วน 1 แผง การบังเงาบางส่วน 2 แผง และการลัดวงจร 1 แผง จากการศึกษาพบว่า การบังเงาบางส่วน 1 แผงทำให้สูญเสียกำลังการผลิตประมาณ 35% การบังเงาบางส่วน 2 แผงทำให้สูญเสียกำลังการผลิตประมาณ 46% และการลัดวงจร 1 แผงทำให้สูญเสียกำลังการผลิตประมาณ 48% โดยการทดลองแรกเป็นการทดลองเปรียบเทียบความแม่นยำในการตรวจจับความผิดพลาดดังกล่าวด้วยอัลกอริทึม 6 ประเภท ได้แก่ Decision Tree, Extra Tree, Random Forest, quadratic discriminant analysis (QD), Logistic Regression และ Gradient Boosting โดยใช้วิธีการประเมินความแม่นยำแบบ 10-fold cross validation และแสดงค่าความแม่นยำเป็นค่า F1-score ปรากฏว่า Random Forest ที่วัด Impurity แบบ gini มีความแม่นยำสูงสุดที่ 96.93% จากนั้นงานวิจัยนี้ได้พัฒนาความแม่นยำของโมเดลจำแนกประเภทโดยการนำหลายอัลกอริทึมมาทำนายผลร่วมกัน จากผลการทดสอบปรากฏว่า โมเดลที่แม่นยำที่สุดเป็นการรวมกันของ QD, Extra Tree ที่วัด Impurity แบบ entropy และ Decision Tree โดยวิธีการทำนายผลรวมกันนั้นจะกำหนดน้ำหนักคะแนนของผลการทำนายจากแต่ละอัลกอริทึมดังนี้ QD = 1, Extra Tree = 2 และ Decision Tree = 2 โดยผลการทำนายของแต่ละอัลกอริทึมจะถูกคูณด้วยสัมประสิทธิ์คะแนนดังกล่าว คลาสไหนได้คะแนนมากที่สุดคลาสนั้นคือผลลัพธ์ของการทำนายของโมเดล โดยวิธีนี้ให้ความแม่นยำ 97.67%

จากตัวอย่างงานวิจัยที่ได้ศึกษาทำให้ทราบถึงวิธีการและอัลกอริทึมต่างๆ ที่ใช้ในการตรวจจับความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ วิทยานิพนธ์นี้จึงเสนอวิธีการตรวจจับและระบุประเภทความผิดพลาดด้วยการใช้เทคนิค Machine Learning รวมถึงใช้เทคโนโลยีสื่อสารแบบไร้สาย LoRa สำหรับการเก็บข้อมูลความผิดพลาดเพื่อนำมาฝึกอัลกอริทึม Machine Learning เพื่อเป็นแนวทางในการพัฒนาระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ให้มีความปลอดภัยและเชื่อถือได้มากยิ่งขึ้น

1.3 วัตถุประสงค์ของวิทยานิพนธ์

1. ศึกษาวิธีการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์
2. พัฒนาระบบตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ให้ทำงานได้แบบเรียลไทม์

3. ลดระยะเวลาในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ เพื่อการดำเนินการแก้ไขได้อย่างรวดเร็ว

1.4 ขอบเขตของวิทยานิพนธ์

1. พิจารณาเฉพาะความผิดปกติ 5 ประเภทได้แก่ ความผิดปกติแบบลัดวงจรภายในสตริง (Intra string line-line fault) ความผิดปกติเปิดวงจร (Open-circuit) ความผิดปกติจากการบังเงาบางส่วน (Partial shading) ความผิดปกติจากการเสื่อมสภาพของอุปกรณ์ระดับสตริง (String degradation) และความผิดปกติแบบลัดวงจรระหว่างสายข้ามสตริง (Cross string line-line fault) เนื่องจากความผิดปกติแบบลัดวงจรและการบังเงาบางส่วนเป็นความผิดปกติที่พบได้บ่อยในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ ส่วนความผิดปกติเปิดวงจรและการเสื่อมสภาพของอุปกรณ์ระดับสตริงก็เป็นความผิดปกติที่ส่งผลกระทบต่อประสิทธิภาพการผลิตไฟฟ้าโดยตรงเช่นกัน
2. พิจารณาเฉพาะการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เพื่อแจ้งเตือนให้ผู้ปฏิบัติงานทราบ
3. อัลกอริทึมสำหรับตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ได้มาจากการฝึกด้วยชุดข้อมูลความผิดปกติที่ประกอบด้วยพารามิเตอร์ 4 ชนิด ได้แก่ กระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ซึ่งพารามิเตอร์ดังกล่าวได้มาจากการจำลองความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 w ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง
4. ใช้ภาษา Python ทำงานบนโปรแกรม Jupyter Notebook เพื่อสร้างโมเดล AdaBoost และ Gradient Boosting สำหรับตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

1.5 ขั้นตอนการดำเนินการ

1. กำหนดวัตถุประสงค์และขอบเขตของงานวิจัย
2. ศึกษางานวิจัยที่เกี่ยวข้องสำหรับการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์
3. ศึกษาประเภทของความผิดปกติที่เกิดขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์
4. ศึกษาอัลกอริทึมการเรียนรู้ของเครื่องประเภทต่างๆ ที่สามารถนำมาประยุกต์ใช้ในการตรวจจับและจำแนกประเภทความผิดปกติได้

5. ศึกษาการใช้งานภาษา Python บนโปรแกรม Jupyter Notebook
6. ศึกษาเทคโนโลยีสื่อสารแบบไร้สาย LoRa
7. ออกแบบและพัฒนาระบบส่งข้อมูลผ่านเทคโนโลยีสื่อสารแบบไร้สาย LoRa
8. ติดตั้งระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W แบบ 2 สตริง สตริงละ 3 แผง เพื่อเก็บค่ากระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ขณะเกิดความผิดพลาดที่จำลองขึ้น โดยใช้เทคโนโลยีสื่อสารแบบไร้สาย LoRa ในการส่งข้อมูลมายังศูนย์ควบคุม
9. ออกแบบและเขียนโปรแกรมเพื่อสร้างอัลกอริทึมสำหรับการตรวจจับและจำแนกความผิดปกติด้วยเทคนิคการเรียนรู้ของเครื่อง
10. ฝึกและทดสอบ Adaboost และ Gradient Boosting ด้วยชุดข้อมูลที่ได้มาจาก ขั้นตอน ที่ 8 เพื่อเปรียบเทียบประสิทธิภาพสำหรับการตรวจจับและระบุประเภทความผิดปกติ
11. นำโมเดลการเรียนรู้ของเครื่องที่มีความแม่นยำที่สุดมาทดสอบการทำงานได้แบบเรียลไทม์
12. วิเคราะห์และสรุปผลการทดสอบของวิทยานิพนธ์
13. เรียบเรียงผลวิจัยและจัดทำรูปเล่มวิทยานิพนธ์
14. เสนอวิทยานิพนธ์กับกรรมการต่อไป

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. อัลกอริทึมที่พัฒนาสามารถตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ได้
2. เมื่อเกิดความผิดปกติขึ้นผู้ปฏิบัติงานสามารถทราบได้อย่างรวดเร็ว ทำให้ลดระยะเวลาในการตรวจสอบ
3. ลดความเสียหายของระบบจากความผิดปกติที่เกิดขึ้นได้ เพราะสามารถดำเนินการแก้ไขได้อย่างทันท่วงที
4. ประหยัดทรัพยากรบุคคลในการดำเนินการตรวจสอบเหตุผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

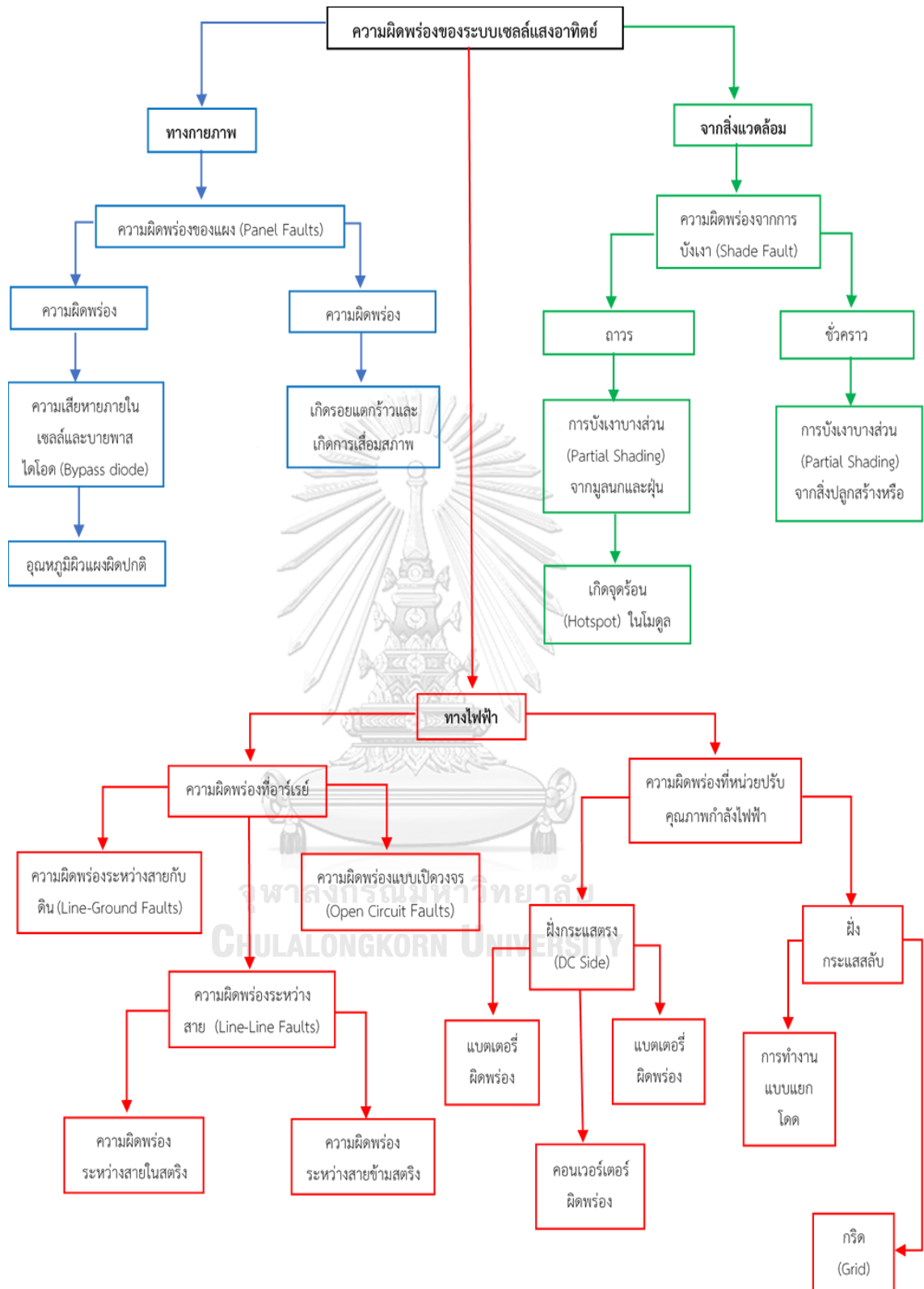
บทที่ 2

ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้จะกล่าวถึงหลักการและข้อมูลพื้นฐานเกี่ยวกับการตรวจจับและระบุประเภทความผิดปกติที่เกิดขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ ซึ่งประกอบไปด้วยความผิดปกติประเภทต่างๆ ที่เกิดขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ เทคนิคในการจำแนกความผิดปกติ เครื่องมือทาง Machine Learning ที่สามารถใช้ในการตรวจจับและจำแนกความผิดปกติได้ และเทคโนโลยีสื่อสารแบบไร้สาย เพื่อใช้เป็นแนวทางในการพัฒนาระบบตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ให้สามารถทำงานได้แบบเรียลไทม์และมีความแม่นยำ

2.1 ความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

ระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เป็นระบบไฟฟ้าที่มีความอ่อนไหวต่อการเกิดความผิดปกติ เนื่องจากเมื่อมีความผิดปกติเกิดขึ้นภายในระบบอาจทำให้ระบบได้รับผลกระทบเป็นวงกว้าง โดยความผิดปกติต่างๆ ที่สามารถเกิดขึ้นได้ภายในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์แสดงดังรูปที่ 1 [2] ได้แก่ ความผิดปกติทางกายภาพเป็นความผิดปกติของแผงเซลล์แสงอาทิตย์ ความผิดปกติทางไฟฟ้า และความผิดปกติที่เกิดจากสิ่งแวดล้อม ในงานวิจัยนี้จะกล่าวถึงเฉพาะความผิดปกติทางไฟฟ้าที่อาร์เรย์เซลล์แสงอาทิตย์ (PV Array) ความผิดปกติที่เกิดจากสิ่งแวดล้อม และความผิดปกติทางกายภาพที่เป็นการเสื่อมสภาพของสตริงเท่านั้น



รูปที่ 1 ความผิดปกติที่เกิดขึ้นในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ [2]

2.1.1 ความผิดพลาดทางไฟฟ้าที่อาร์เรย์เซลล์แสงอาทิตย์

ความผิดพลาดทางไฟฟ้าเป็นความผิดพลาดที่ส่งผลกระทบต่อความสามารถในการผลิตพลังงานไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยตรง โดยระดับความรุนแรงจะขึ้นอยู่กับชนิดและขนาดพื้นที่ของความผิดพลาด ความผิดพลาดทางไฟฟ้าที่อาร์เรย์เซลล์แสงอาทิตย์แบ่งออกเป็น 2 แบบ ได้แก่ การลัดวงจร (Short Circuit Faults) และการเปิดวงจร (Open Circuit Faults) [2]

2.1.1.1 ความผิดพลาดแบบลัดวงจร (Short Circuit Faults)

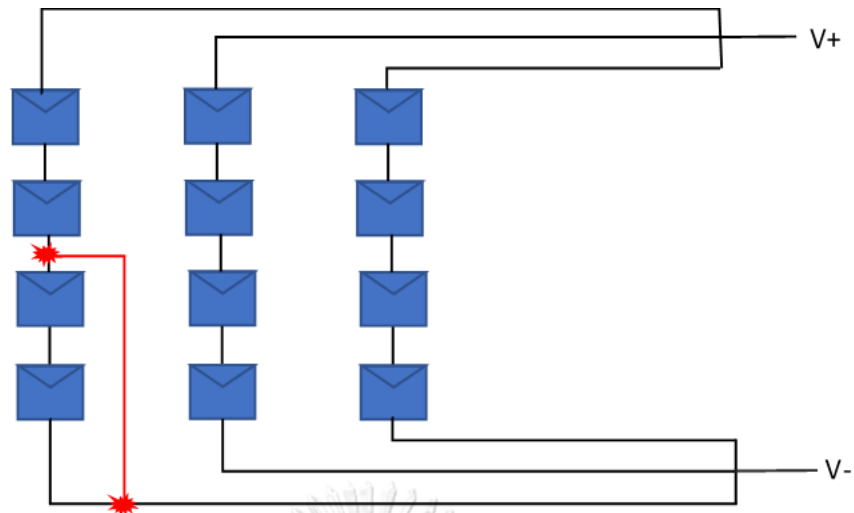
ความผิดพลาดแบบลัดวงจรมีโอกาสเกิดขึ้นไม่น้อยเมื่อเทียบกับความผิดพลาดประเภทอื่นๆ ในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ การลัดวงจรคือการไหลของกระแสไฟฟ้าแบบไม่ตั้งใจให้เกิดระหว่างสองจุดที่มีความต่างศักย์ไม่เท่ากัน ความผิดพลาดแบบลัดวงจรยังแบ่งออกเป็นการลัดวงจรระหว่างสาย (Line-Line Faults) และลัดวงจรระหว่างสายกับดิน (Line-Ground Faults) ความผิดพลาดประเภทนี้อาจมีความรุนแรงและยากต่อการตรวจจับ [2] ในงานวิจัยนี้พิจารณาความผิดพลาดแบบลัดวงจรระหว่างสายเท่านั้น

การลัดวงจรระหว่างสายเป็นการลัดวงจรที่เกิดขึ้นระหว่างสองจุดของสายที่มีความต่างศักย์ไม่เท่ากัน โดยในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์อาจเกิดขึ้นได้สองแบบ แบบแรกเป็นการลัดวงจรระหว่างสายภายในสตริง (Intra String Line-Line Fault) ดังแสดงในรูปที่ 2 แบบที่สองเป็นการลัดวงจรระหว่างสายข้ามสตริง (Cross String Line-Line Fault) ดังแสดงในรูปที่ 3 ความรุนแรงของการลัดวงจรระหว่างสายจะขึ้นอยู่กับจำนวนโมดูลที่ถูกลัดวงจรไป โดยยิ่งถูกลัดวงจรไปมากก็จะมี ความเสียหายมากขึ้น ดังนั้นแน่นอนว่าความผิดพลาดชนิดนี้จะส่งผลโดยตรงต่อการผลิตพลังงานไฟฟ้าของระบบไฟฟ้าพลังงานแสงอาทิตย์ [2] ซึ่งกระแสผิดพลาด (I_f) สูงสุดที่สามารถเกิดขึ้นเท่ากับ

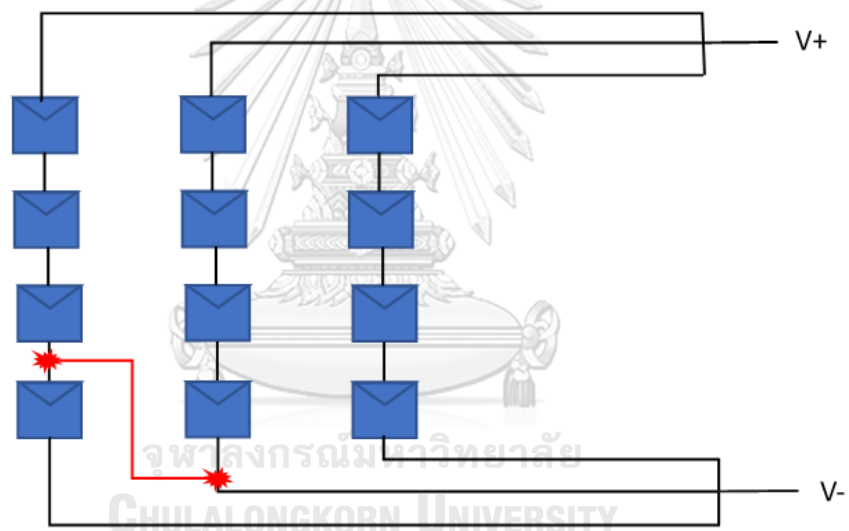
$$I_f = (n - 1)I_{sc} \quad (1)$$

เมื่อ n คือ จำนวนสตริง

I_{sc} คือ กระแสลัดวงจรของแต่ละโมดูล



รูปที่ 2 ตัวอย่างการเกิดความผิดปกติของแบบลัดวงจรระหว่างสายในสตรึง [2]

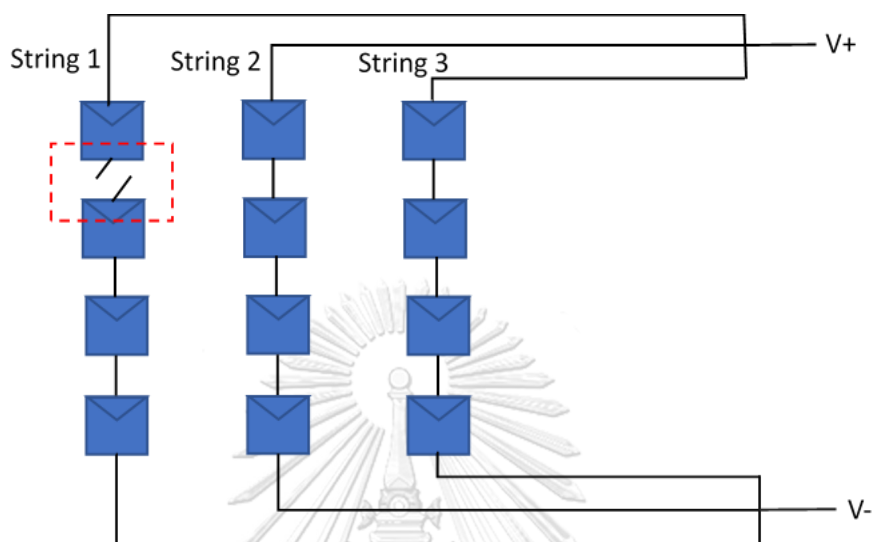


รูปที่ 3 ตัวอย่างการเกิดความผิดปกติของแบบลัดวงจรระหว่างสายข้ามสตรึง [2]

2.1.1.2 ความผิดปกติของแบบเปิดวงจร (Open Circuit Faults)

ความผิดปกติของแบบเปิดวงจรเกิดขึ้นเมื่อส่วนหนึ่งส่วนใดภายในอาร์เรย์เซลล์แสงอาทิตย์ถูกตัดการเชื่อมต่อ ทำให้แผงเซลล์แสงอาทิตย์ทุกแผงที่อยู่ในสตรึงที่มีการเปิดวงจรมีกระแสไฟฟ้าเป็นศูนย์ รูปที่ 4 แสดงตัวอย่างความผิดปกติของแบบเปิดวงจรในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ สมมติว่าแผงเซลล์แสงอาทิตย์ทุกแผงได้รับรังสีดวงอาทิตย์เท่ากัน เมื่อเกิดความผิดปกติของแบบเปิดวงจรที่สตรึงที่ 1 แรงดันไฟฟ้าของระบบจะไม่เปลี่ยนแปลง แต่กระแสไฟฟ้าจะลดลง เนื่องจากสตรึงที่ 1 มีกระแสไฟฟ้าเป็นศูนย์ ดังนั้นกำลังไฟฟ้าของระบบมีค่าลดลงหรือเกิดการสูญเสียกำลังไฟฟ้านั่นเอง

บางครั้งความผิดพลาดแบบเปิดวงจรก็เกิดจากผลกระทบจากความผิดพลาดประเภทอื่นๆ เนื่องจากการตัดวงจรของอุปกรณ์ป้องกันหลังจากตรวจพบความผิดพลาด [10]



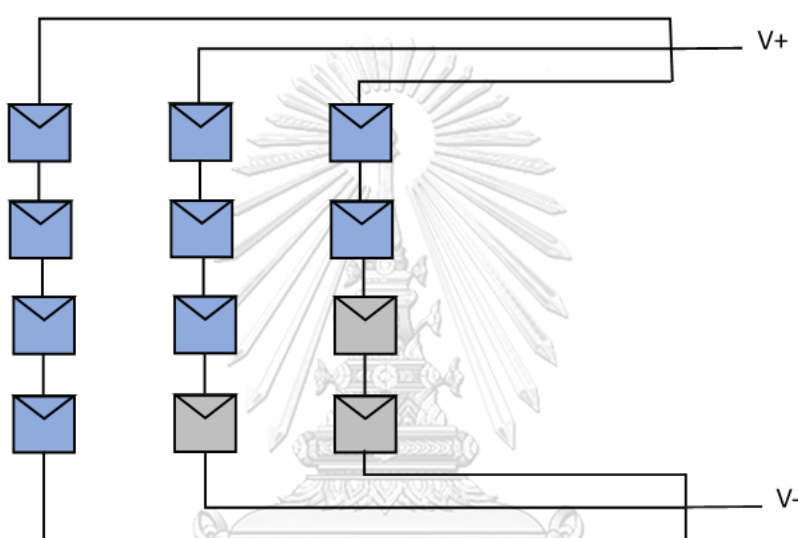
รูปที่ 4 ตัวอย่างความผิดพลาดแบบเปิดวงจร [10]

2.1.2 ความผิดพลาดที่เกิดจากสิ่งแวดลอม

ความผิดพลาดที่เกิดจากสิ่งแวดลอมในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จะเกิดจากการที่การฉายรังสีของดวงอาทิตย์ต่อแผงเซลล์แสงอาทิตย์ถูกบดบัง ส่งผลให้ความสามารถในการผลิตไฟฟ้าลดต่ำลง ความผิดพลาดชนิดนี้เรียกว่า “ความผิดพลาดจากการบังเงา (Shade Faults)” โดยเกิดได้ทั้งแบบสมดุลคือการที่ทั้งระบบได้รับรังสีดวงอาทิตย์ลดลงใกล้เคียงกัน และแบบไม่สมดุลคือการที่บางส่วนของระบบเซลล์แสงอาทิตย์ได้รับรังสีดวงอาทิตย์น้อยลงกว่าส่วนอื่นๆ ในระบบ เรียกเหตุการณ์แบบนี้ว่า “การบังเงาบางส่วน (Partial Shading)” ดังแสดงในรูปที่ 5 โมดูลสีฟ้าเป็นโมดูลที่ได้รับรังสีดวงอาทิตย์ปกติ ส่วนโมดูลสีเทาเป็นโมดูลที่เกิดการบังเงา การบังเงาบางส่วนเป็นความผิดพลาดที่เกิดขึ้นได้บ่อยและมีพฤติกรรมการเกิด 2 แบบ ดังตารางที่ 1 คือ การบังเงาบางส่วนแบบถาวร (Permanent Partial Shading) และการบังเงาบางส่วนแบบชั่วคราว (Temporary Partial Shading) นอกจากจะทำให้การผลิตไฟฟ้าลดลงแล้ว การบังเงาบางส่วนอาจทำให้เกิดจุดร้อน (Hotspots) ภายในแผงเซลล์แสงอาทิตย์ ซึ่งทำให้แผงเซลล์แสงอาทิตย์เกิดการเสื่อมสภาพได้ในระยะยาว [2]

ตารางที่ 1 ประเภทของการบังเงาบางส่วน

ประเภทการบังเงาบางส่วน	สาเหตุ
การบังเงาบางส่วนแบบถาวร	มูลนก, ฝุ่น, วัตถุปิดทับ
การบังเงาบางส่วนแบบชั่วคราว	การเคลื่อนตัวของก้อนเมฆ, เงาดันไม้, เงาสีงก่อสร้าง



รูปที่ 5 ตัวอย่างการบังเงาบางส่วน [2]

ในปัจจุบันการบังเงาบางส่วนก็ยังเป็นเรื่องที่ไม่ได้รับการแก้ไขอย่างสมบูรณ์ เนื่องจากยังไม่มีอุปกรณ์เฉพาะที่สามารถตรวจจับการบังเงาบางส่วนได้ แม้ว่าส่วนมากการบังเงาบางส่วนที่เกิดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จะมีพฤติกรรมแบบชั่วคราว แต่ก็เป็นเรื่องสำคัญที่ต้องได้รับการแก้ไข เนื่องจากเหตุผลดังต่อไปนี้ [2]

1. การตรวจจับการบังเงาบางส่วนจะทำให้ชะลอการเสื่อมสภาพของแผงเซลล์แสงอาทิตย์ได้
2. การตรวจจับการบังเงาบางส่วนจะช่วยลดความซับซ้อนในการทำงานของ MPPT (Maximum Power Point Tracking) ลงได้
3. ลักษณะสัญญาณทางไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ที่เกิดบังเงาบางส่วน คล้ายกับความผิดปกติประเภทอื่น จึงต้องสามารถแยกความผิดปกติจากการบังเงาบางส่วนออกมาให้ชัดเจน เพื่อป้องกันไม่ให้อุปกรณ์ป้องกันตัววงจรแบบไม่จำเป็น

2.1.3 ความผิดปกติทางกายภาพ

ความผิดปกติทางกายภาพของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์มีทั้งความผิดปกติที่เกิดภายใน ความผิดปกติที่เกิดภายนอก และการเสื่อมสภาพของอุปกรณ์ โดยงานวิจัยนี้จะกล่าวถึงเพียงความผิดปกติทางกายภาพที่เป็นการเสื่อมสภาพของอุปกรณ์ในระดับสตรีง โดยอาจจะเกิดที่แผงเซลล์แสงอาทิตย์ บริเวณขั้วต่อต่างๆ หรือเกิดที่สาย ซึ่งสาเหตุอาจจะมาจากการถูกกัดกร่อน ผลกระทบจากรังสีไอโอเล็ด และอื่นๆ ความเสื่อมสภาพเหล่านี้มักจะส่งผลทำให้ความต้านทานอนุกรมเพิ่มขึ้นจึงเป็นสาเหตุให้กำลังผลิตไฟฟ้าของระบบเซลล์แสงอาทิตย์ลดลง [11]

2.2 วิธีการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

วิธีที่ใช้ในการป้องกันความเสียหายจากความผิดปกติทางไฟฟ้าภายในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยทั่วไปมี 3 วิธี [2] ดังนี้

1. ใช้อุปกรณ์ป้องกันมาตรฐาน เช่น ฟิวส์
2. ใช้การรวมอุปกรณ์ป้องกันหลายตัวเข้าด้วยกันเพื่อป้องกันความเสียหายจากความผิดปกติหลายประเภท
3. ทดสอบระบบพลังงานแสงอาทิตย์ด้วยเครื่องมือทดสอบแบบออฟไลน์ (Offline) หรือเรียลไทม์ (Real-time) เพื่อตรวจหาความผิดปกติ

อย่างไรก็ตามวิธีการข้างต้นไม่สามารถรับประกันความปลอดภัยของระบบได้ รวมทั้งการใช้อุปกรณ์ป้องกันหลายตัวจะมีค่าใช้จ่ายที่ค่อนข้างสูง และการทดสอบระบบก็ต้องใช้กำลังคนและเวลามากขึ้น โดยเทคนิคและวิธีการสำหรับการตรวจจับความผิดปกติของระบบที่ดีต้องมีคุณสมบัติดังต่อไปนี้ [2]

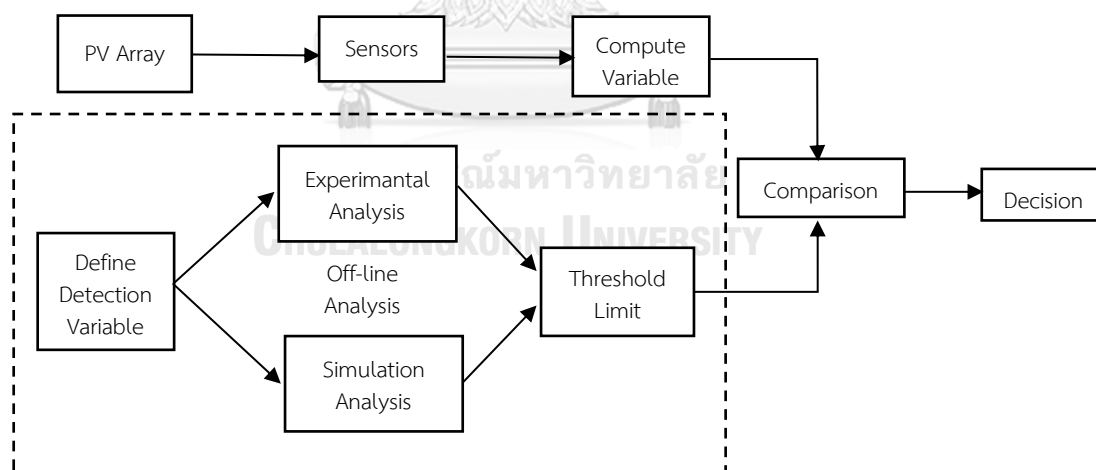
1. มีความสามารถในการตรวจจับความผิดปกติได้โดยไม่รบกวนการทำงานของระบบ
2. มีความสามารถในการแยกประเภทความผิดปกติและระบุตำแหน่งที่เกิดความผิดปกติได้
3. ประหยัดและมีความยืดหยุ่นสามารถติดตั้งเข้ากับระบบแบบดั้งเดิมได้
4. มีโครงสร้างไม่ซับซ้อน
5. สามารถใช้ได้กับระบบเซลล์แสงอาทิตย์ได้โดยไม่คำนึงถึงประเภทและขนาดของระบบ

เนื่องจากการเติบโตของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ในปัจจุบัน ผู้เกี่ยวข้องจึงมีความต้องการที่จะพัฒนาเทคนิคและอัลกอริทึมสำหรับตรวจจับความผิดปกติแบบออนไลน์ (Online) ที่มีความเชื่อถือได้และมีประสิทธิภาพ ด้วยเหตุนี้ทำให้ในช่วงไม่กี่ปีที่ผ่านมาได้มีการนำเทคนิคต่างๆ มาใช้ ได้แก่ Model Based Difference Measurement (MBDM), Real-time Difference

Measurement (RDM), Output Signal Analysis (OSA), Machine Learning (ML), Infrared Thermography (ITH), Hybrid Detection Techniques (HDT) เป็นต้น ที่ผ่านมามีเทคนิค MBDM เป็นเทคนิคที่มีการศึกษามากที่สุด โดย MBDM เป็นเทคนิคที่ใช้การเปรียบเทียบค่าพารามิเตอร์ระหว่างค่าที่วัดได้จากระบบจริงแบบเรียลไทม์กับค่าทางทฤษฎีจากแบบจำลอง ซึ่งที่ผ่านมามีการเสนอแบบจำลองระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จำนวนมากเพื่อนำมาตรวจจับการทำงานที่ผิดปกติ แต่ในบรรดาเทคนิคที่กล่าวมาข้างต้น มี 3 เทคนิคที่น่าสนใจที่จะพัฒนาต่อไป ได้แก่ RDM, OSA และ ML เพราะมีศักยภาพที่จะพัฒนาไปเป็นเทคนิคการตรวจจับความผิดปกติในระบบเซลล์แสงอาทิตย์ที่ปราศจากความผิดพลาดในการตรวจจับได้ [2]

2.2.1 Real-time Difference Measurement

เทคนิค Real-time Difference Measurement (RDM) เป็นเทคนิคที่เปรียบเทียบค่าพารามิเตอร์ที่วัดได้แบบเรียลไทม์จากระบบจริงกับเกณฑ์ (Threshold limit) ที่กำหนดไว้ โดยเกณฑ์ได้มาจากการวิเคราะห์ผลการทดลองร่วมกับแบบจำลอง รูปที่ 6 แสดงขั้นตอนการทำงานของเทคนิค RDM ซึ่งมีหลักการใกล้เคียงกับเทคนิค MBDM แต่ต่างกันว่า MBDM นำค่าพารามิเตอร์ที่วัดได้แบบเรียลไทม์จากระบบจริงเปรียบเทียบกับค่าพารามิเตอร์ที่ได้จากแบบจำลองเลย ทำให้มีความล่าช้ากว่าเทคนิค RDM เนื่องจากเวลาจากการประมวลผลของแบบจำลอง [2]

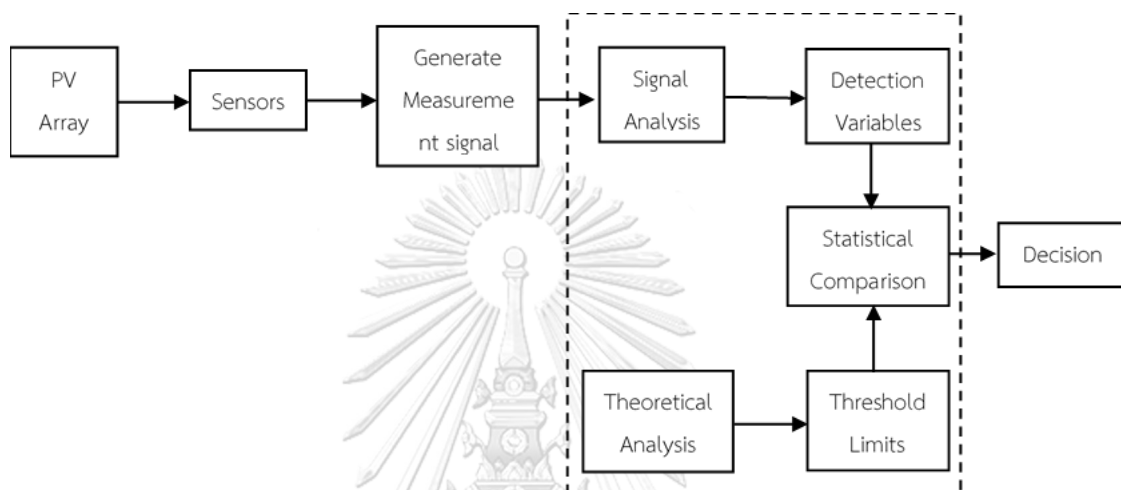


รูปที่ 6 เทคนิค RDM สำหรับตรวจจับความผิดปกติในระบบเซลล์แสงอาทิตย์ [2]

2.2.2 Output Signal Analysis

ความผิดปกติที่เกิดขึ้นในระบบเซลล์แสงอาทิตย์จะส่งผลกระทบต่อสัญญาณขาออก (Output) โดยเมื่อเกิดความผิดปกติ สัญญาณขาออกไม่ว่าจะเป็นกระแสไฟฟ้าหรือแรงดันจะมีลักษณะผิดเพี้ยนไป ด้วยเหตุนี้ Output Signal Analysis (OSA) จึงเป็นเทคนิคที่วิเคราะห์ลักษณะ

สัญญาณขาออกของระบบว่ามีความผิดปกติขึ้นไปจากลักษณะสัญญาณขาออกทางทฤษฎีหรือไม่ เพื่อความเข้าใจที่ง่ายขึ้นรูปที่ 7 ได้แสดงขั้นตอนการทำงานของเทคนิค OSA แม้ว่าโดยทั่วไปเทคนิค OSA มักจะถูกนำมาใช้ในการตรวจจับการเกิดอาร์ค (Arc Fault) แต่ไม่กี่ปีที่ผ่านมาได้มีการนำ OSA มาตรวจจับความผิดปกติประเภทอื่นๆ ด้วยเช่นกัน [2] ใน [9] ได้ใช้เทคนิค OSA เพื่อตรวจจับความผิดปกติแบบลัดวงจรของฝั่งกระแสตรงในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์



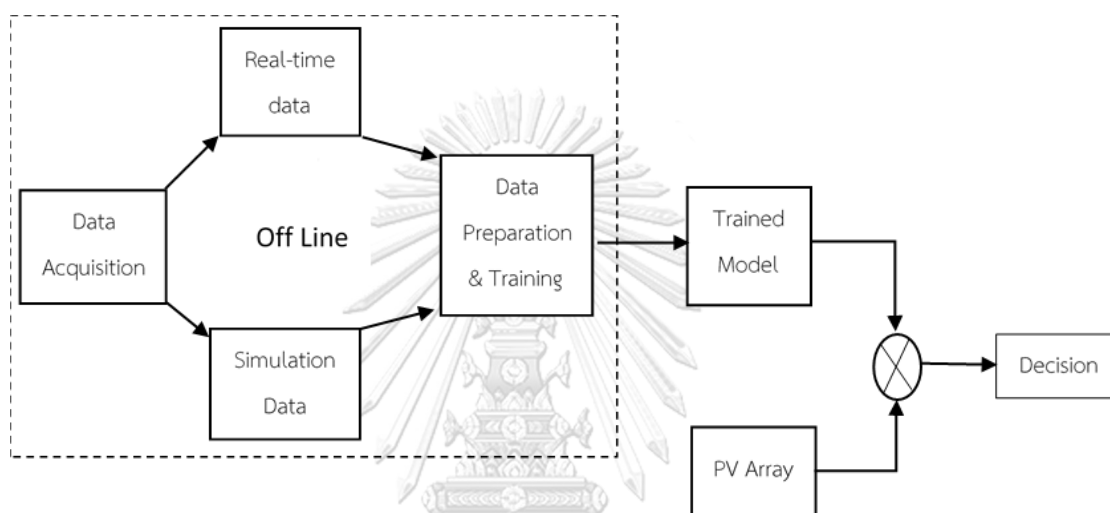
รูปที่ 7 เทคนิค OSA สำหรับการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ [2]

2.2.3 Machine Learning

Machine Learning (ML) หรือการเรียนรู้ของเครื่องเป็นเทคนิคที่มีหลักการโดยอาศัยการเรียนรู้ความสัมพันธ์ระหว่างข้อมูลขาเข้า (Input) กับข้อมูลขาออก (Output) อัลกอริทึมจะถูกฝึกฝนให้เรียนรู้ความสัมพันธ์นี้จนสามารถตัดสินใจเองได้ว่าข้อมูลที่เข้ามาจะให้ผลลัพธ์อย่างไร ซึ่งในการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ข้อมูลที่ใช้สำหรับการฝึกอัลกอริทึมสามารถใช้ได้ทั้งข้อมูลที่มาจากระบบจำลองและข้อมูลที่ได้จากการวัดระบบจริง รูปที่ 8 แสดงขั้นตอนการทำงานของเทคนิค ML นอกจากนี้การตรวจจับความผิดปกติด้วยเทคนิค ML ทำให้แก้ปัญหาคความยุ่งยากในการกำหนดเกณฑ์ (Threshold Limit) ได้ เป็นเทคนิคที่ง่ายและแม่นยำในการตรวจจับความผิดปกติ อย่างไรก็ตามเทคนิค ML ก็ยังมีข้อเสียและข้อจำกัดดังนี้ [2]

1. ความแม่นยำขึ้นอยู่กับคุณภาพและจำนวนของข้อมูลที่นำมาฝึกอัลกอริทึม
2. ในทางปฏิบัติเป็นเรื่องยากที่จะรวบรวมข้อมูลความผิดปกติที่เกิดขึ้นจริงในระบบเซลล์แสงอาทิตย์ให้ได้ตามต้องการสำหรับนำมาฝึกอัลกอริทึม

3. การนำข้อมูลที่ได้จากแบบจำลองมาฝึกอัลกอริทึมไม่ค่อยน่าเชื่อถือ เนื่องจากความแม่นยำจะขึ้นอยู่กับแบบจำลองที่นำมาใช้
4. ชุดข้อมูลที่นำมาฝึกอัลกอริทึมจำเป็นต้องใช้ข้อมูลจากระบบเซลล์แสงอาทิตย์ที่ต้องการจะตรวจจับความผิดปกติ ไม่มีชุดข้อมูลใดเป็นมาตรฐานที่สามารถใช้ฝึกอัลกอริทึมของระบบเซลล์แสงอาทิตย์ต่างๆ ได้



รูปที่ 8 เทคนิค ML สำหรับการตรวจจับความผิดปกติในระบบเซลล์แสงอาทิตย์ [2]

2.3 อัลกอริทึม Machine Learning

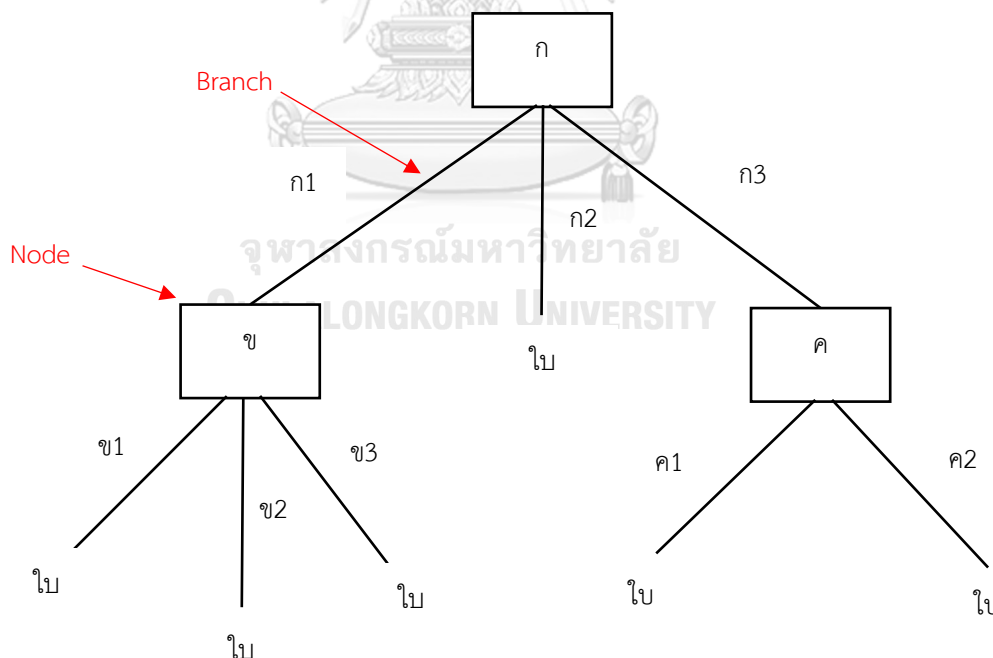
ปัจจุบันพลังงานหมุนเวียนได้รับความสนใจเป็นอย่างมากสำหรับการนำมาใช้ผลิตไฟฟ้า โดยเฉพาะพลังงานแสงอาทิตย์ แต่ก็ยังมีความท้าทายในเรื่องของความไม่แน่นอนในกำลังการผลิต ทำให้ภาคส่วนที่เกี่ยวข้องพยายามศึกษาหาวิธีแก้ไขประสิทธิภาพและความเชื่อถือได้ของระบบผลิตไฟฟ้าด้วยพลังงานแสงอาทิตย์ หนึ่งในเทคนิคที่ได้รับความสนใจอย่างมากในระยะหลังสำหรับการนำมาประยุกต์ใช้กับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์คือเทคนิค Machine Learning ไม่ว่าจะเป็นการพยากรณ์กำลังการผลิต การวิเคราะห์ความเสื่อมสภาพของอุปกรณ์ การควบคุมการผลิต และการตรวจสอบความผิดปกติ สำหรับอัลกอริทึม Machine Learning ที่ใช้ในวิทยานิพนธ์ฉบับนี้ได้แก่ AdaBoost และ Gradient Boosting ที่สร้างมาจากการทำงานต่อกันเป็นลูกโซ่ของต้นไม้ตัดสินใจ (Decision Tree)

2.3.1 ต้นไม้ตัดสินใจ (Decision Tree)

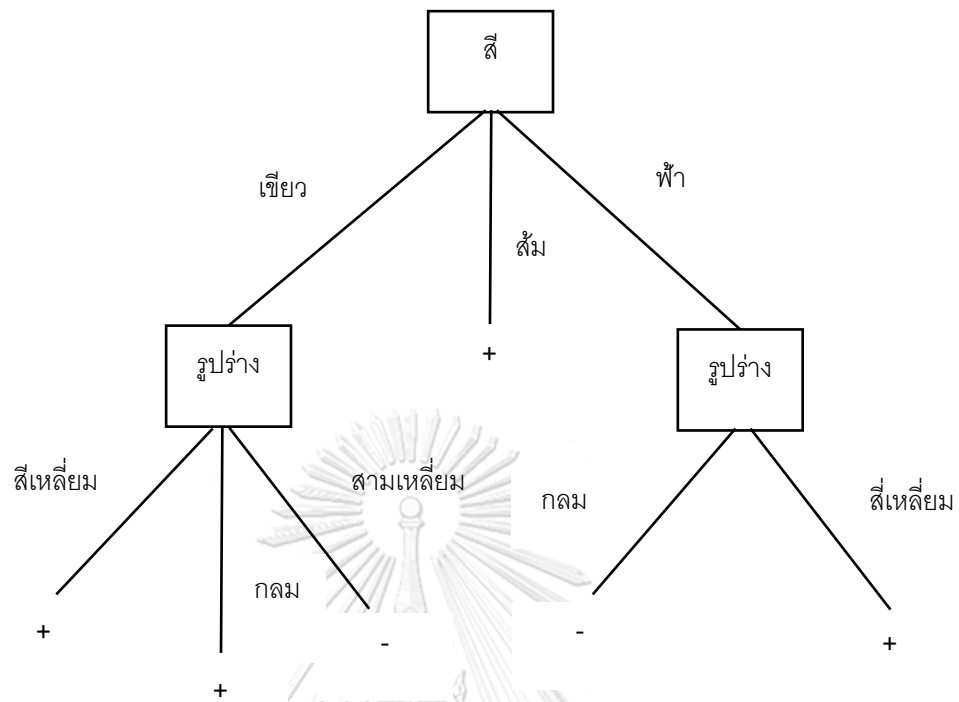
Decision Tree [12] คือ เครื่องมือที่มีโครงสร้างเป็นกราฟที่มีลักษณะคล้ายต้นไม้ดังรูปที่ 9 มาช่วยในงานการตัดสินใจและแปลผลเพื่อให้มนุษย์ทำงานได้ง่ายขึ้น Decision Tree สามารถนำมาช่วยในการจำแนกประเภท (Classification) ทั้งในปัญหาแบบไบนารี (Binary Classes) และปัญหาแบบหลายคลาส (multiple Classes) สำหรับการแก้ปัญหาแบบไบนารี ตำแหน่งใบจะมีเพียง 2 คลาสเท่านั้น เช่น บวกหรือลบดังตัวอย่างในรูปที่ 10 ส่วน Decision Tree ที่ใช้ในการแก้ปัญหาแบบหลายคลาสดำแหน่งใบจะมีได้ตั้งแต่ 2 คลาสขึ้นไปดังตัวอย่างในรูปที่ 11

ต้นไม้ตัดสินใจมีองค์ประกอบหลัก 3 ส่วน ได้แก่

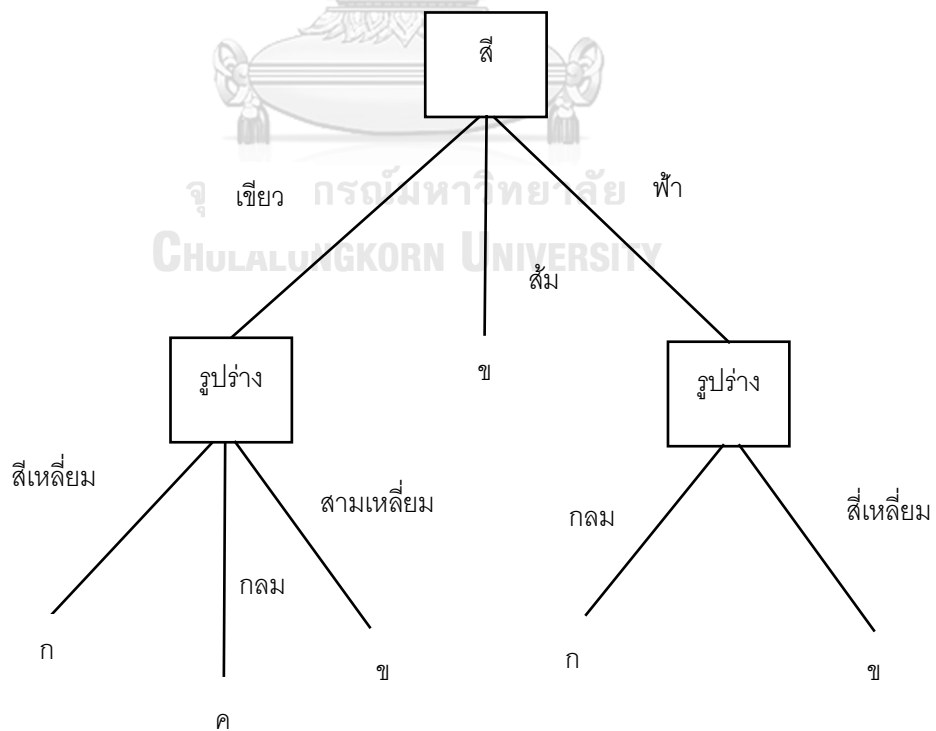
1. ปม (Node) ใช้สำหรับระบุชื่อคุณลักษณะประจำ (Attribute)
2. กิ่ง (Branch) ใช้สำหรับระบุค่าของแต่ละคุณลักษณะประจำของปมที่เหนือกิ่งกิ่งนั้น หรือเรียกว่าเส้นเชื่อม (Edge)
3. ใบ (Leaf) ใช้สำหรับระบุค่าเป้าหมาย เช่น คลาส (Class) ของปัญหา



รูปที่ 9 รูปแบบของต้นไม้ตัดสินใจ [12]



รูปที่ 10 ตัวอย่างต้นไม้ตัดสินใจแบบไบนารี [12]

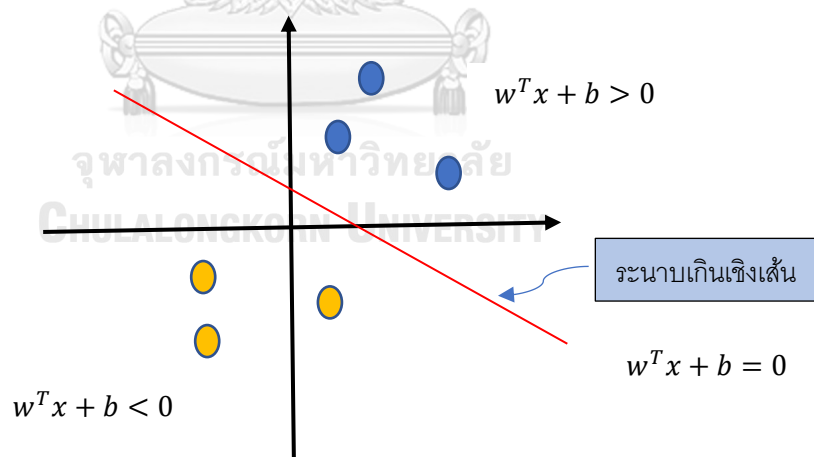


รูปที่ 11 ตัวอย่างต้นไม้ตัดสินใจแบบหลายคลาส [12]

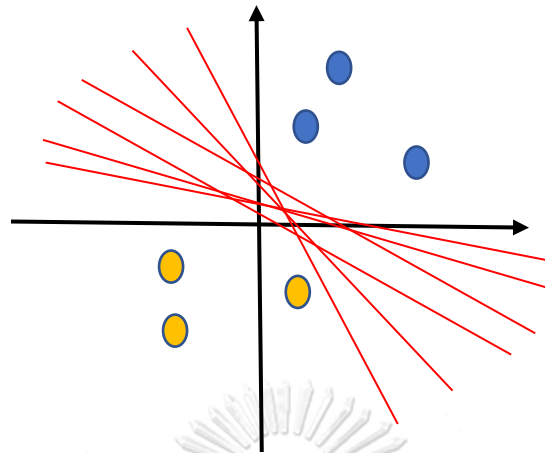
2.3.2 ซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine: SVM)

SVM [12] โดยทั่วไปเป็นเครื่องมือจำแนกประเภทแบบไบนารี (Binary Classes) แต่ก็มีหลายงานที่ประยุกต์ใช้งาน SVM ในการจำแนกประเภทแบบหลายคลาส (Multiple Classes) SVM สามารถใช้งานได้ทั้งกับข้อมูลที่เป็นเชิงเส้น (Linear) และไม่เป็นเชิงเส้น (Nonlinear) สำหรับชุดข้อมูลที่เป็นเชิงเส้น สามารถใช้เส้นตรงหรือที่เรียกว่า “ระนาบเกินเชิงเส้น” (Linear Hyperplane) แบ่งข้อมูลแต่ละคลาสได้ดังตัวอย่างรูปที่ 12 แต่ชุดข้อมูลที่ไม่เป็นเชิงเส้น ไม่สามารถใช้ระนาบเกินเชิงเส้นแบ่งข้อมูลแต่ละคลาสได้ โดยอาจจะใช้เส้นโค้งแบ่งคลาสของชุดข้อมูลได้ แต่ก็มี ความซับซ้อนมาก ดังนั้นวิธีการแก้ปัญหาที่ง่ายคือการแปลงชุดข้อมูลจากปริภูมิไม่เชิงเส้น (Nonlinear Space) เป็นปริภูมิเชิงเส้นก่อน โดยการใช้ฟังก์ชันเคอร์เนล (Kernel Function) ในการส่ง (Mapping) ข้อมูลจากปริภูมิไม่เชิงเส้นไปยังปริภูมิที่มีความเป็นเชิงเส้น

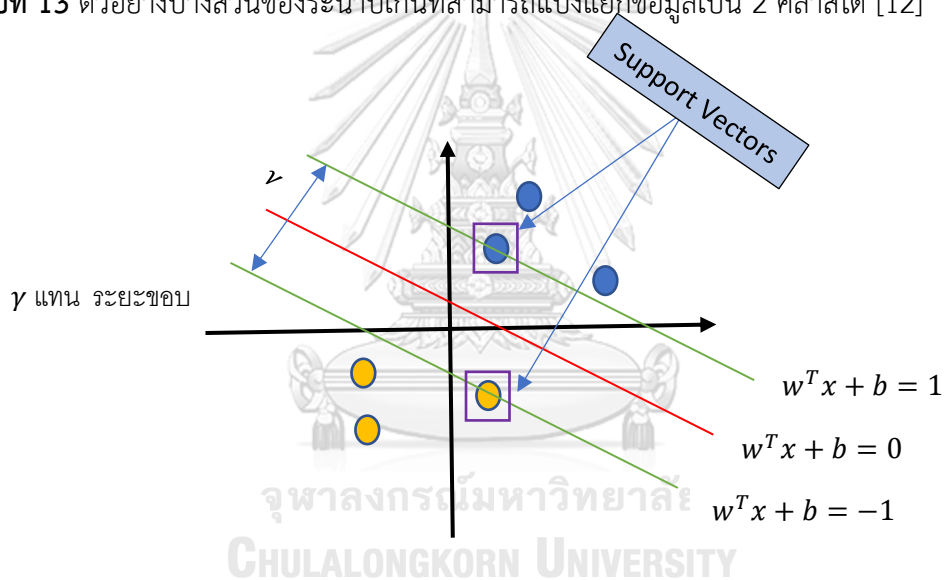
การแบ่งแยกข้อมูลแต่ละคลาสนั้น มีระนาบเกินจำนวนมากที่สามารถแบ่งแยกข้อมูลได้ดังรูปที่ 13 แต่ต้องเลือกใช้ระนาบเกินที่เหมาะสมที่สุด โดยวิธีการหาระนาบเกินที่เหมาะสมที่สุดทำได้โดยการหาระยะขอบ (Margin) ที่มากที่สุดระหว่างสองคลาสดังแสดงดังรูปที่ 14 ซึ่งข้อมูลเวกเตอร์ที่อยู่ที่ขอบของแต่ละคลาสจะถูกเรียกว่า “เวกเตอร์ซัพพอร์ต” (Support Vector) เป็นที่มาของชื่อวิธีการ “ซัพพอร์ตเวกเตอร์แมชชีน” นั่นเอง



รูปที่ 12 ตัวอย่างระนาบเกินเชิงเส้นแบ่งข้อมูลเป็น 2 คลาส ในปริภูมิ 2 มิติ [12]



รูปที่ 13 ตัวอย่างบางส่วนของระนาบเกินที่สามารถแบ่งแยกข้อมูลเป็น 2 คลาสได้ [12]



รูปที่ 14 ระยะขอบและเวกเตอร์ซัพพอร์ต [12]

หลังจากทราบวิธีการในการเลือกระนาบเกินที่จะใช้ในการแบ่งแยกข้อมูลแล้ว ต้องหาความสัมพันธ์ระหว่างเวกเตอร์ปกติ w ของระนาบเกินและระยะขอบ โดยเริ่มจากเวกเตอร์ซัพพอร์ตของแต่ละคลาส ดังนี้

$$w^T x + b = 1 \quad (2.1)$$

$$w^T x + b = -1 \quad (2.2)$$

นำ (2.1) ลบด้วย (2.2)

$$w^T(x_+ - x_-) = 2 \quad (2.3)$$

รูปที่ 15 แสดงให้เห็นว่า $x_+ - x_-$ เป็นเวกเตอร์ลิพธ์ระหว่างระนาบเกินของทั้งสองคลาส และทำมุม θ กับเวกเตอร์ปกติของระนาบเกิน

หาเวกเตอร์ที่มีขนาดเท่ากับระยะขอบและมีทิศทางเดียวกับเวกเตอร์ปกติ ดังนี้

$$(x_+ - x_-) \cos \theta = (x_+ - x_-) \left(\frac{w^T(x_+ - x_-)}{\|w\| \|x_+ - x_-\|} \right) \quad (2.4)$$

แทนค่า (2.3) ใน (2.4)

$$(x_+ - x_-) \frac{w^T(x_+ - x_-)}{\|w\| \|x_+ - x_-\|} = \frac{2}{\|w\|} \left(\frac{x_+ - x_-}{\|x_+ - x_-\|} \right) \quad (2.5)$$

จากสมการที่ (2.5) เมื่อ $\frac{(x_+ - x_-)}{\|x_+ - x_-\|}$ เป็นเวกเตอร์หนึ่งหน่วย (Unit Vector) ดังนั้นจะได้

$$\gamma = \frac{2}{\|w\|} \quad (2.6)$$

สมการที่ (2.6) แสดงความสัมพันธ์ระหว่างระยะขอบกับระนาบเกิน ซึ่งจากสมการนี้สามารถทำให้เรากำหนดปัญหาค่าที่เหมาะสมสำหรับระยะขอบที่มากที่สุดในการจำแนกแบบไบนารีได้ โดยใช้การแบ่งพื้นที่ของแต่ละคลาสดังรูปที่ 15 โดยอสมการของทั้งสองพื้นที่สามารถเขียนเป็นอสมการเดียวได้โดยคูณด้วยค่า γ ของแต่ละพื้นที่ จะได้อสมการดังนี้

$$y_i(w^T x_i + b) \geq 1 \quad (2.7)$$

สมการที่ (2.7) เป็นสมการที่เราต้องการ เมื่อพิจารณาร่วมกับสมการที่ (2.6) ทำให้ปัญหาการหาค่าที่เหมาะสมที่สุดเขียนได้ดังนี้

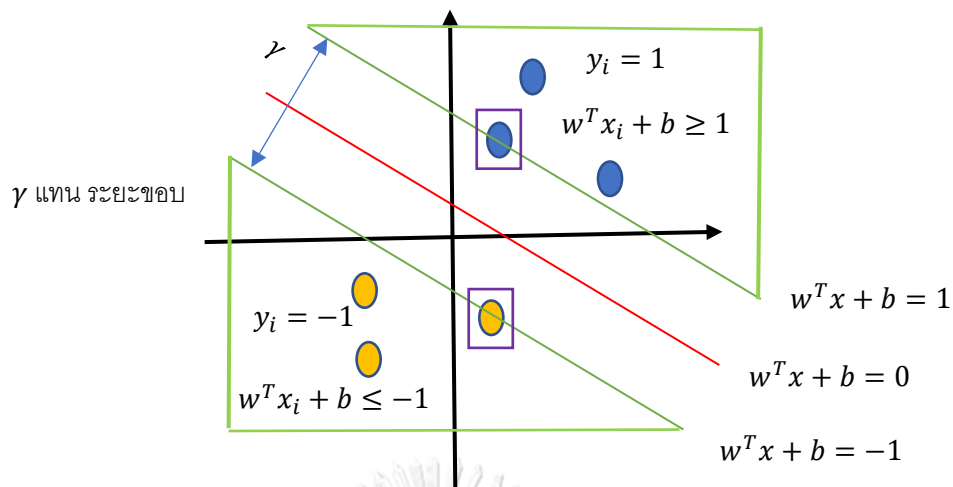
$$\text{Maximize:} \quad \gamma = \frac{2}{\|w\|}$$

$$\text{Subject to:} \quad y_i(w^T x_i + b) \geq 1$$

เปลี่ยนค่าระยะขอบมากที่สุดให้เป็นระยะขอบน้อยที่สุด ดังนั้นจะได้ปัญหาการหาค่าที่เหมาะสมที่สุดใหม่ดังนี้

$$\text{Minimize:} \quad \frac{1}{\gamma} = \frac{1}{2} w^T w$$

$$\text{Subject to:} \quad y_i(w^T x_i + b) \geq 1$$



รูปที่ 15 แสดงอสมการของพื้นที่ในแต่ละคลาส [12]

2.3.3 การถดถอยโลจิสติก (Logistic regression)

การถดถอยโลจิสติก [13, 14] เป็นเครื่องมือการจำแนกประเภทแบบไบนารี (Binary) โดยใช้ฟังก์ชันพื้นฐานเป็นฟังก์ชันโลจิสติก (Logistic function) ดังสมการที่ (2.8) และมีลักษณะกราฟดังรูปที่ 16 มาแก้ปัญหา การถดถอยโลจิสติกมีความคล้ายคลึงกับวิธีการถดถอยเชิงเส้น จะต่างกันตรงที่เอาต์พุต (Output) ของการถดถอยโลจิสติกจะมีค่าเป็นไบนารี คือ 0 หรือ 1 แทนที่จะเป็นค่าเชิงตัวเลข (Numeric value) และเขียนเป็นสมการการถดถอยโลจิสติกได้ดังสมการที่ (2.9)

$$F(x) = \frac{1}{1+e^{-x}} \quad (2.8)$$

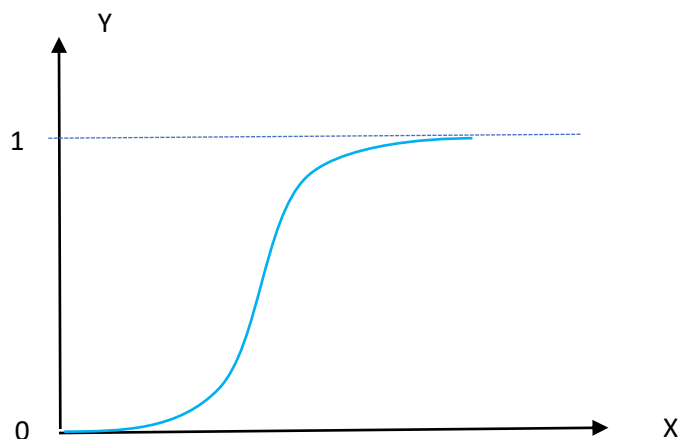
เมื่อ x คือ ตัวแปรอิสระ

แสดงสมการการถดถอยโลจิสติกได้ดังนี้

$$Y = \frac{e^{(\beta_0 + \beta_1 x)}}{1 + e^{\beta_0 + \beta_1 x}} \quad (2.9)$$

เมื่อ Y คือ เอาต์พุตหรือตัวแปรตาม

x คือ อินพุต (Input) หรือตัวแปรอิสระ

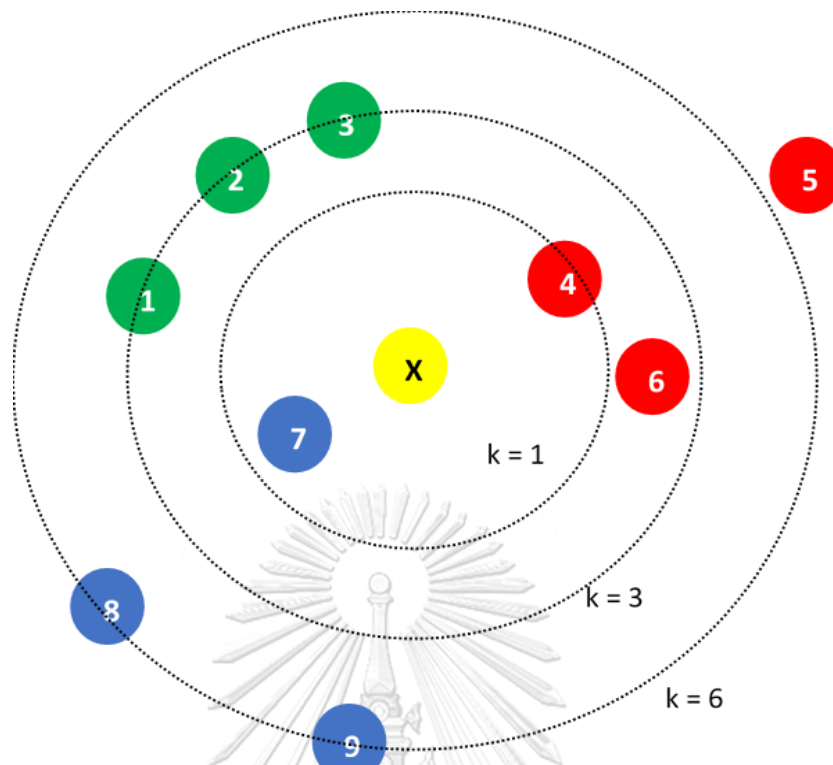


รูปที่ 16 กราฟโลจิสติกฟังก์ชัน [13]

การถดถอยโลจิสติกใช้กันอย่างแพร่หลายในหมู่นักสถิติ โดยแรกเริ่มใช้ในวงการสาธารณสุข ก่อนที่จะแพร่หลายไปยังสาขาอื่นๆ สำหรับในสาขาวิศวกรรมไฟฟ้าก็มีการนำการถดถอยโลจิสติกมาใช้ในงานเกี่ยวกับการจัดการความผิดพลาดในระบบไฟฟ้า ยกตัวอย่างเช่นใน [15] ใช้โมเดลการถดถอยโลจิสติก (Logistic regression model) มาจำแนกสาเหตุการเกิดความผิดพลาดในระบบจำหน่ายไฟฟ้ากำลังที่เกิดจากต้นไม้และจากสัตว์ โดยผลที่ได้จากโมเดลการถดถอยโลจิสติกมีความแม่นยำใกล้เคียงกับการใช้โครงข่ายประสาทเทียม (Artificial Neural Network)

2.3.4 การหาเพื่อนบ้านใกล้ที่สุด (k-Nearest Neighbors: k-NN)

k-NN [12] เป็นวิธีที่เข้าใจง่ายและไม่ซับซ้อน ปัจจุบันก็ยังได้รับความนิยมใช้ในการจำแนกประเภท (Classification) เนื่องจากเป็นอีกวิธีที่มีประสิทธิภาพสูง หลักการของวิธี k-NN จะใช้วิธีการหาระยะห่างระหว่างข้อมูลตัวอย่างที่ต้องการจำแนกประเภทกับข้อมูลตัวอย่างทั้งหมด โดยการกำหนดค่า k เป็นค่าตั้งแต่ 1 จนถึงค่าสุดท้ายคือค่าเท่ากับจำนวนตัวอย่างทั้งหมด เช่น ชุดข้อมูลมี 9 ตัวอย่าง ก็กำหนดค่า k ตั้งแต่ 1 ถึง 9 แล้วทดสอบความแม่นยำของการจำแนกประเภทว่าค่า k เท่าไรมีความแม่นยำที่สุด เพื่อความเข้าใจที่ง่ายขึ้นจะแสดงตัวอย่างของการจำแนกประเภทของชุดข้อมูลในรูปที่ 17 เป็นการจำแนกประเภทของ x ซึ่งเป็นข้อมูลตัวอย่างที่ไม่ทราบประเภท และมีข้อมูลอีก 9 ตัวอย่างที่ทราบประเภท ประกอบด้วยข้อมูลตัวอย่างที่ 1-3 เป็นประเภท G, ข้อมูลตัวอย่างที่ 4-6 เป็นประเภท R และข้อมูลตัวอย่างที่ 7-9 เป็นประเภท B วิธีการจำแนกคือในแต่ละค่า k จะหาข้อมูลตัวอย่างที่ใกล้กับ x ที่สุดจำนวน k ตัวอย่าง เช่น หาก $k = 3$ ต้องหาข้อมูลตัวอย่างที่ใกล้ที่สุด 3 ตัวอย่าง หากข้อมูลตัวอย่างประเภทไหนมีจำนวนมากที่สุด ใน 3 ตัวอย่างนี้ ให้จัดอยู่ในประเภทนั้น ผลการจำแนกประเภทแสดงดังตารางที่ 2



รูปที่ 17 ตัวอย่างการกำหนดค่า k [12]

ตารางที่ 2 ผลการจำแนกประเภทของข้อมูลตัวอย่าง x โดยที่มีข้อมูลตัวอย่างที่ทราบประเภท 9 ตัวอย่าง [12]

K	ผลการจำแนกประเภท
1	B หรือ R
2	R
3	R
4	R หรือ G
5	G
6	G
7	G หรือ B
8	B หรือ R หรือ G
9	

การเลือกค่า k ที่เหมาะสมที่สุดต้องทดสอบความแม่นยำในการจำแนกประเภทของแต่ละค่า k แต่โดยส่วนใหญ่ค่า k ที่เหมาะสมและนิยมใช้มากที่สุดคือ $k = 1$

ข้อเสียประการหนึ่งที่เกิดขึ้นได้สำหรับวิธี k -NN คือเมื่อมีจำนวนตัวอย่างที่ใช้ในการฝึกมาก จะทำให้ประมวลผลได้ช้า จึงนำวิธีที่เรียกว่า “Condensed Nearest Neighbor (CNN)” เข้ามาลดจำนวนตัวอย่างที่ใช้ในการฝึกให้น้อยลง เพื่อให้การประมวลผลเร็วขึ้น แต่ก็ยังคงไว้ซึ่งประสิทธิภาพที่ใกล้เคียงกับ k -NN โดยการสร้างเซตต้นแบบ (Prototype) ขึ้นจากจำนวนตัวอย่างทั้งหมด แต่จะมีจำนวนตัวอย่างน้อยลง

ขั้นตอนวิธีการสร้าง Function CNN

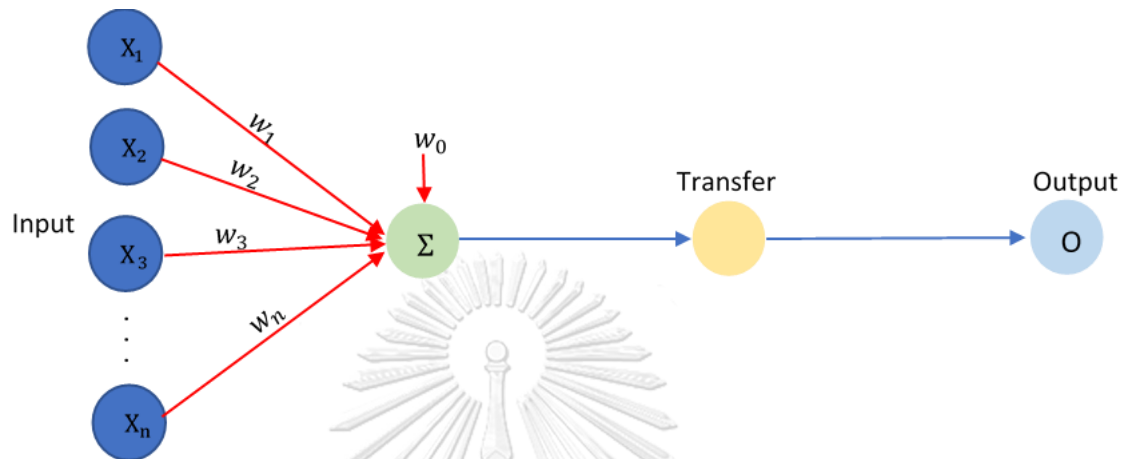
1. กำหนดเซตต้นแบบ P เป็นเซตว่าง
2. หาตัวอย่าง x ในเซตของตัวอย่างทั้งหมด S ที่ใกล้กับเซต P ที่สุด และเป็นคนละประเภทกัน
3. นำตัวอย่าง x ออกจากเซต S และนำไปเพิ่มในเซต P
4. ทำซ้ำตั้งแต่ขั้นตอนที่ 1 ถึง 3 จนกระทั่งไม่มีตัวอย่างไปเพิ่มในเซต P ได้อีก
5. RETURN เซต P

2.3.5 โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)

ANN [12] มีหลักการที่ประยุกต์มาจากการทำงานของเซลล์ประสาทของมนุษย์ เริ่มโดย Warren S. McCulloch และ Walter Pitts ซึ่งเป็นนักประสาทวิทยาและนักตรรกศาสตร์ตามลำดับ ทั้งคู่ได้เสนอแบบจำลองทางคณิตศาสตร์ สำหรับอธิบายการทำงานที่ซับซ้อนของสมองว่าเกิดจากการเชื่อมต่อกันของหน่วยย่อยพื้นฐานจำนวนมาก ต่อมานักวิจัยหลายท่านได้นำงานวิจัยของ Warren S. McCulloch และ Walter Pitts มาพัฒนาต่อจนกระทั่งสามารถแก้ปัญหาพื้นฐานของการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) และการเรียนรู้แบบมีผู้สอน (Supervised Learning) ได้สำเร็จ เริ่มแรกสำหรับการแก้ปัญหาของการเรียนรู้แบบมีผู้สอน ถูกเสนอด้วยวิธีการที่เรียกว่า “เพอร์เซปตรอน” (Perceptron) เป็นวิธีที่มีการเรียนรู้ได้จากตัวอย่างที่นำเข้ามา แต่ก็มีข้อจำกัดที่ Perceptron ไม่สามารถแก้เป็นหาได้ เช่น ปัญหาแบบ XOR (Exclusive OR) ต่อมาจึงมีการพัฒนาวิธีการเพื่อมาแก้ไขข้อจำกัดของ Perceptron และเป็นวิธีการที่นิยมใช้มาจนถึงปัจจุบัน วิธีการดังกล่าวถูกเรียกว่า “โครงข่ายประสาทเทียมแบบแพร่กลับ” (Backpropagation) สามารถแก้ปัญหา XOR ได้โดยการใช้โครงข่ายแบบหลายชั้น (Multi-Layer) ที่มีการคำนวณค่าความผิดพลาดระหว่างชั้นดังกล่าว ลักษณะพื้นฐานของโครงข่ายประสาทเทียมแบบ Perceptron แสดงดังรูปที่ 18 โดยมี x_i เป็นค่าป้อนเข้า (Input) มี w_0 เป็นไบอัส (Bias) ซึ่งมีค่าเป็น 1 เสมอ ส่งค่าที่ผ่านการไบอัสแล้วผ่าน

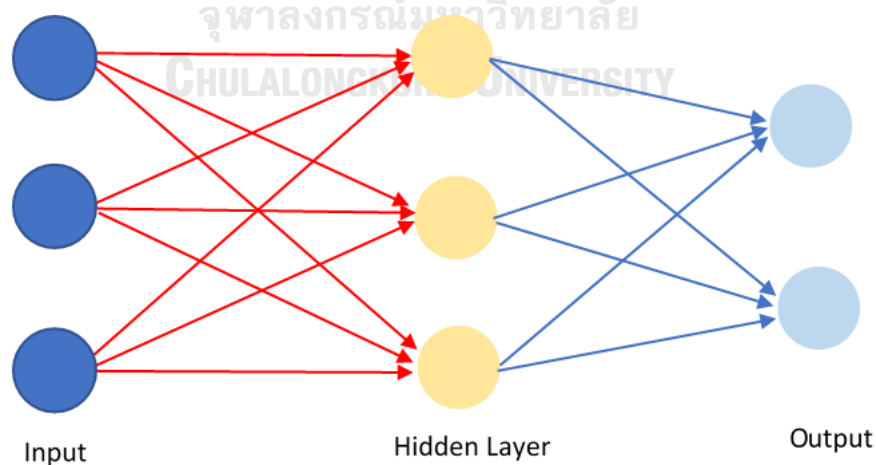
ฟังก์ชันถ่ายโอน (Transfer Function) ที่เป็นฟังก์ชันฮาร์ดลิมิตแบบสมมาตร (Symmetric Hard-Limit) ทำให้ค่าส่งออก (Output) มีค่า 1 หรือ -1 ดังนี้

$$o(x_i) = \begin{cases} 1, & w_0 + \sum_{i=1}^n w_i x_i > 0 \\ -1, & \text{Otherwise} \end{cases}$$



รูปที่ 18 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบ Perceptron [12]

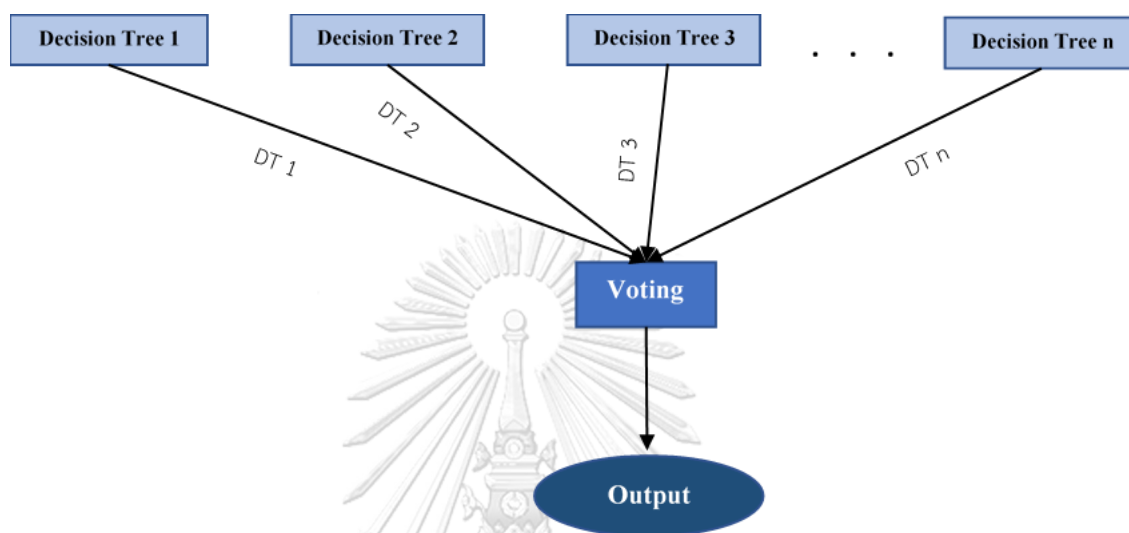
โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบหลายชั้นแสดงดังรูปที่ 19 ประกอบด้วย 3 ส่วนหลัก ได้แก่ ชั้นป้อนเข้า (Input Layer), ชั้นซ่อน (Hidden Layer), ชั้นส่งออก (Output Layer) โดยส่วนที่พิเศษและถือว่าเป็นเอกลักษณ์ของโครงข่ายประสาทเทียมแบบหลายชั้นคือ ชั้นซ่อนที่สามารถแทรกเข้าไปจำนวนกี่ชั้นก็ได้ โดยจะทำหน้าที่คล้ายการสกัดคุณลักษณะ (Feature Extraction) ทำให้สามารถแก้ปัญหาที่ไม่สามารถแบ่งเป็นเชิงเส้นได้



รูปที่ 19 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบหลายชั้น [12]

2.3.6 ป่าสุ่ม (Random Forest: RF)

ป่าสุ่ม [16] เป็นวิธีการที่พัฒนามาจากวิธีการต้นไม้ตัดสินใจ โดยเป็นการนำต้นไม้ตัดสินใจหลายๆ โมเดล มารวมกัน ดังแสดงในรูปที่ 20 โดยค่าส่งออก (Output) สุดท้ายมาจากประเภทที่ถูกจำแนกได้มากที่สุดที่ได้จากผลการทำนาย (Prediction) ของต้นไม้ตัดสินใจแต่ละโมเดล



รูปที่ 20 โครงสร้างของป่าสุ่ม

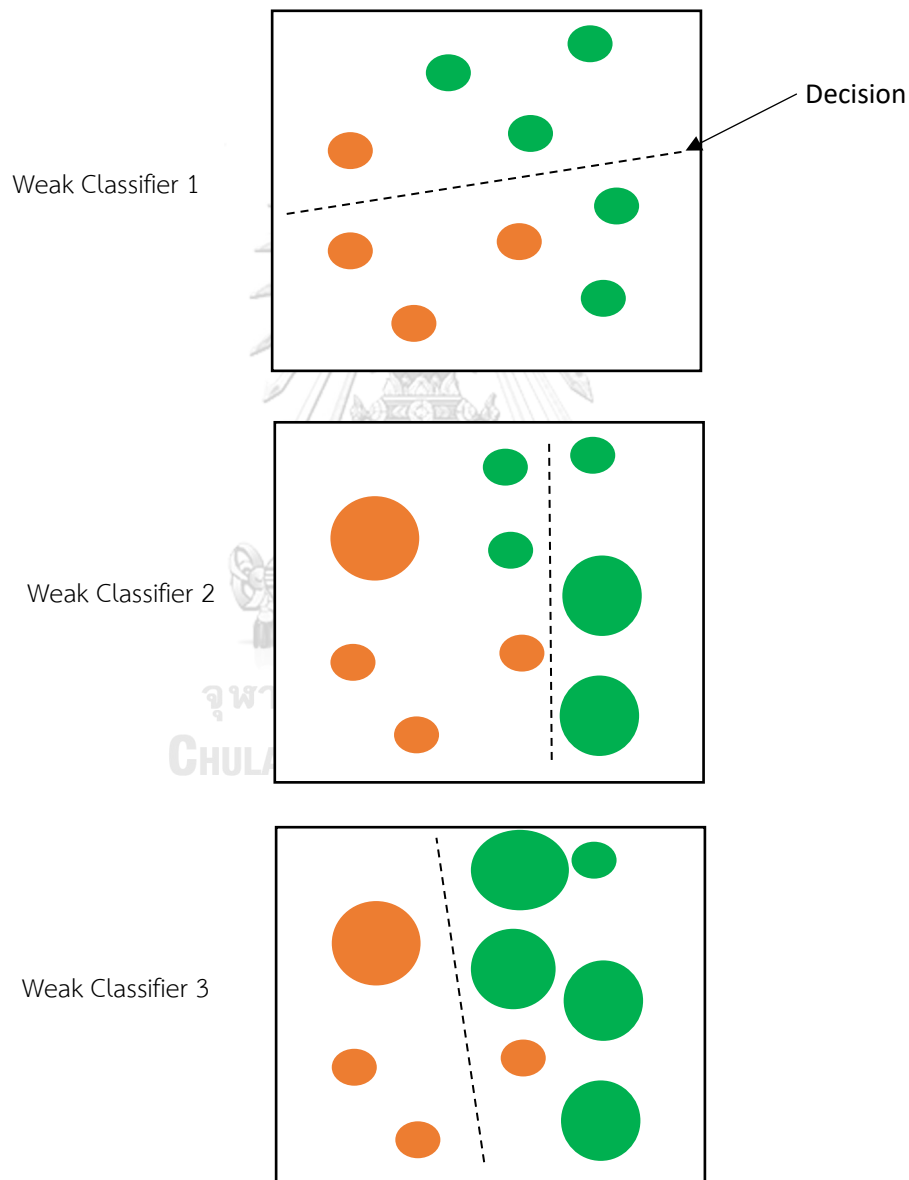
ต้นไม้ตัดสินใจแต่ละโมเดลจะมีชุดข้อมูลและคุณลักษณะ (Feature) ที่ไม่เหมือนกัน เพื่อทำให้เกิดความหลากหลายของการจำแนกประเภทของต้นไม้ตัดสินใจในแต่ละโมเดล

2.3.7 Adaptive Boosting (Adaboost)

Boosting เป็นหนึ่งในเทคนิคของ Ensemble Learning (EL) ที่สร้างเครื่องมือจำแนกประเภทที่แข็งแกร่งจากการใช้การรวมกันของเครื่องมือจำแนกประเภทอย่างง่ายหลายตัว โดยจะทำงานต่อกันในลักษณะเป็นลูกโซ่ ซึ่งแต่ละตัวจะแก้ไขข้อผิดพลาดที่เกิดจากแบบจำลองก่อนหน้านี้ เครื่องมือจำแนกประเภทอย่างง่ายจะถูกเพิ่มจำนวนเข้าไปจนกว่าจะได้รับการคาดการณ์ที่แม่นยำหรือเป็นค่าความแม่นยำที่เหมาะสม [17-19]

Adaboost [17-19] เป็นพื้นฐานของเทคนิค Boosting มักใช้กับเครื่องมือจำแนกประเภทที่ไม่ซับซ้อน เช่น ต้นไม้ตัดสินใจ มาฝึกต่อกันเป็นลูกโซ่ โดยการฝึกแต่ละรอบจะมีการกำหนดค่าน้ำหนักให้ข้อมูลแต่ละรายการ โดยถ้ามีการจำแนกประเภทผิดค่าน้ำหนักของข้อมูลรายการนั้นจะเพิ่มขึ้น ดังตัวอย่างการจำแนกประเภทแบบ 2 คลาสด้วย Adaboost ในรูปที่ 21 เริ่มต้นด้วยการให้ทุกข้อมูลมีค่าน้ำหนักเท่ากันหมด หลังจากนั้นจะสร้างเส้นแบ่งเกณฑ์ที่เรียกว่า Decision stump จากรูปจะเห็นว่าที่ Weak Classifier 1 มีข้อมูลสีเขียว 2 จุด และข้อมูลสีส้ม 1 จุด ที่ถูกจำแนกประเภทผิด หลังจาก

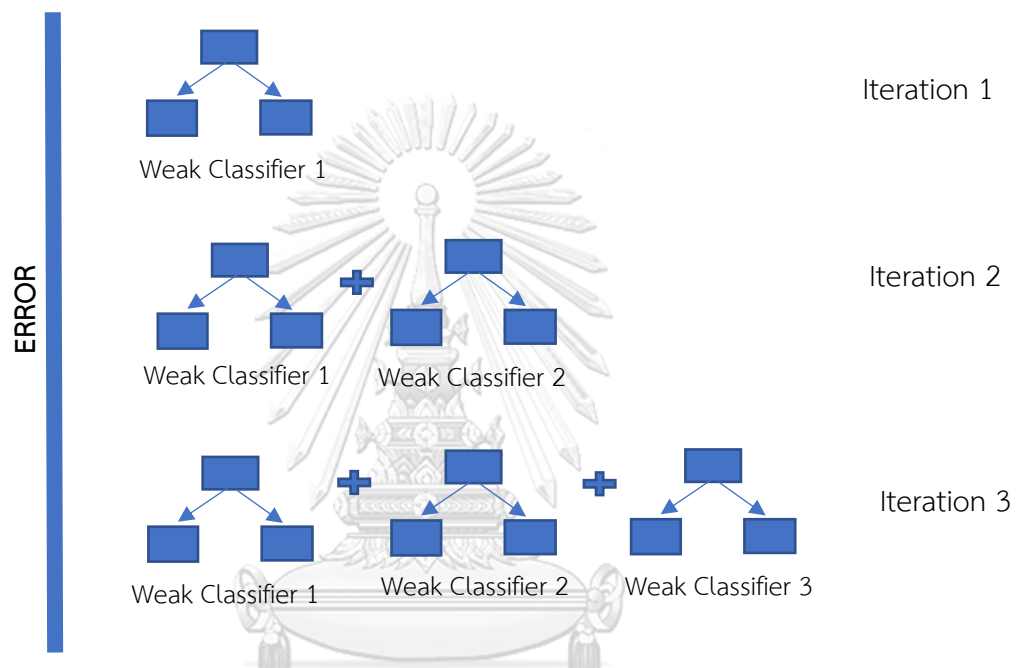
นั่นในรอบที่ 2 จะเห็นว่าที่ Weak Classifier 2 ข้อมูลสีเขียว 2 จุด และสีส้ม 1 จุดที่ถูกจำแนกผิดในรอบแรก ถูกปรับค่าน้ำหนักให้มากขึ้น ทำให้ Decision stump นั้นเปลี่ยนตำแหน่งและจำแนกได้ดีขึ้น เหลือเพียงข้อมูลสีเขียว 2 จุดเท่านั้นที่ถูกจำแนกผิด ต่อมาในรอบที่ 3 จะเห็นว่าที่ Weak Classifier 3 ข้อมูลที่ถูกจำแนกผิดในรอบที่ 2 ถูกปรับค่าน้ำหนักให้มากขึ้น ทำให้ Decision stump นั้นเปลี่ยนตำแหน่งและมีเพียงข้อมูลสีส้มสีเดียวที่ถูกจำแนกประเภทผิด โดยจะเพิ่มจำนวน Weak Classifier ไปจนกว่าจะจำแนกประเภทได้แม่นยำที่สุด



รูปที่ 21 ตัวอย่างการจำแนกประเภทแบบ 2 คลาสด้วย Adaboost [19]

2.3.8 Gradient Boosting (GB)

Gradient Boosting [20, 21] เป็นอีกอัลกอริทึมหนึ่งของเทคนิค Boosting สามารถใช้ได้กับการวิเคราะห์การถดถอยและการจำแนกประเภท โดยมีหลักการทำให้เครื่องมือจำแนกประเภทอย่างง่ายหรือ Weak Classifier แต่ละตัวเรียนรู้รู้ค่าความผิดพลาดสะสมของ Weak Classifier ก่อนหน้า โดยจะเพิ่ม Weak Classifier ไปเรื่อยๆ จนกว่าจะสามารถลดค่าความผิดพลาดรวมของการจำแนกประเภทให้น้อยที่สุด รูปที่ 22 แสดงตัวอย่างการจำแนกประเภทด้วย GB



รูปที่ 22 ตัวอย่างการจำแนกประเภทด้วย Gradient Boosting [20]

Gradient Boosting ประกอบด้วยองค์ประกอบที่สำคัญ 3 ส่วน ดังนี้

1. Loss function จะถูกปรับในแต่ละรอบการทำงาน
2. Weak Classifier คือเครื่องมือจำแนกประเภทอย่างง่าย
3. รูปแบบการทำงานของอัลกอริทึมเพื่อปรับ Loss function ให้น้อยที่สุด

2.4 การประเมินความถูกต้องของตัวจำแนกข้อมูล

เมื่อสร้างตัวจำแนกข้อมูลจากอัลกอริทึม Machine Learning แล้ว จำเป็นต้องประเมินความถูกต้องของตัวจำแนกข้อมูลเหล่านั้น เพื่อประเมินว่าตัวจำแนกข้อมูลที่สร้างมามีประสิทธิภาพและสามารถนำไปใช้งานได้หรือไม่ [22]

2.4.1 ตัวชี้วัดความถูกต้องจากการจำแนกข้อมูล

ตัวจำแนกข้อมูลที่ถูกสร้างมาจะต้องได้รับการตรวจสอบความถูกต้องเพื่อประเมินว่าค่าความถูกต้องของตัวจำแนกข้อมูลนั้นสามารถยอมรับได้หรือไม่ การตรวจสอบความถูกต้องนั้นจะใช้ข้อมูลชุดทดสอบ ซึ่งเป็นข้อมูลที่ไม่ได้อยู่ในชุดฝึกฝน เมื่อทำการทดสอบแล้วจะทราบเปอร์เซ็นต์หรือจำนวนของข้อมูลที่จำแนกได้ถูกต้อง นอกจากนั้นยังต้องพิจารณาถึงอัตราการจำแนกข้อมูลผิดหรืออัตราความผิดพลาด (Misclassification rate) ของตัวจำแนกข้อมูล M ใดๆ [22] โดยสามารถหาได้จากสมการที่ (2.10)

$$\text{Misclassification rate} = 1 - \text{acc}(M) \quad (2.10)$$

โดยที่ $\text{acc}(M)$ คือ ค่าความถูกต้องของตัวจำแนกข้อมูล M

ในการวิเคราะห์ความถูกต้องและความผิดพลาดในการจำแนกข้อมูลนิยมใช้ Confusion matrix ซึ่งเป็นเครื่องมือสำคัญในการประเมินผลการจำแนกข้อมูล โดยจะเป็นตารางขนาด $m \times m$ มีข้อมูลแต่ละเซลล์เป็น $cm_{i,j}$ หมายถึงข้อมูลจริงอยู่ในคลาส i แต่ถูกทำนายให้อยู่ในคลาส j โดยถ้าจำแนกข้อมูลถูกจะได้ $i = j$ เช่น $cm_{1,1}$ $cm_{2,2}$ ไปจนถึง $cm_{m,m}$ ตัวอย่างของ Confusion Matrix แสดงดังตารางที่ 3 ซึ่งเป็น Confusion Matrix ขนาด 2×2 โดยแบ่งข้อมูลออกเป็น 2 คลาส คือคลาสที่เป็น Positive tuples (เป็นคลาสของข้อมูลที่สนใจ เช่น คลาสของลูกค้าที่ซื้อโทรศัพท์) และคลาสที่เป็น Negative tuples (คลาสของลูกค้าที่ไม่ซื้อโทรศัพท์) จากนั้นเมื่อจำแนกข้อมูลจะได้ความหมายของข้อมูลแต่ละเซลล์ดังนี้ [22]

1. True Positive (TP) หมายถึง ทำนายข้อมูลว่าเป็น Positive tuples และข้อมูลจริงก็เป็น Positive tuples
2. True Negative (TN) หมายถึง ทำนายข้อมูลว่าเป็น Negative tuples และข้อมูลจริงก็เป็น Negative tuples
3. False Positive (FP) หมายถึง ทำนายข้อมูลว่าเป็น Positive tuples แต่ข้อมูลจริงเป็น Negative tuples
4. False Negative (FN) หมายถึง ทำนายข้อมูลว่าเป็น Negative tuples แต่ข้อมูลจริงเป็น Positive tuples

ตารางที่ 3 ตัวอย่าง Confusion matrix สำหรับ Positive tuples และ Negative tuples [22]

Actual \ Predicted	C ₁ (Positive tuples)	C ₂ (Negative tuples)
	C ₁ (Positive tuples)	True Positive (TP)
C ₂ (Negative tuples)	False Positive (FP)	True Negative (TN)

หลังจากได้ Confusion matrix แล้ว สามารถนำผลการจำแนกข้อมูลใน Confusion matrix มาวิเคราะห์หาค่าของตัวบ่งชี้ประสิทธิภาพได้แก่ Precision, Recall, F-1 score และ Accuracy โดยคำอธิบายของทั้ง 4 ตัวบ่งชี้ประสิทธิภาพมีดังต่อไปนี้

1. Precision เป็นอัตราส่วนของจำนวนผลการทำนายที่ถูกต้องของคลาสที่พิจารณาต่อผลการทำนายทั้งหมดของคลาสที่พิจารณา [8]
2. Recall เป็นอัตราส่วนของจำนวนผลการทำนายที่ถูกต้องของคลาสที่พิจารณาต่อข้อมูลจริงทั้งหมดของคลาสที่พิจารณา [8]
3. F1-score เป็นค่าเฉลี่ยแบบฮาร์โมนิกระหว่าง Precision และ Recall [8]
4. Accuracy เป็นอัตราส่วนของจำนวนผลการทำนายที่ถูกต้องทั้งหมดต่อจำนวนข้อมูลทั้งหมด [8]

จาก Confusion matrix สามารถหาค่า Precision, Recall, F1-score และ Accuracy ได้

ดังนี้

$$Precision = \frac{TP}{TP+FP} \quad (2.11)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.12)$$

$$F1 - score = \frac{TP}{TP+FP} \quad (2.13)$$

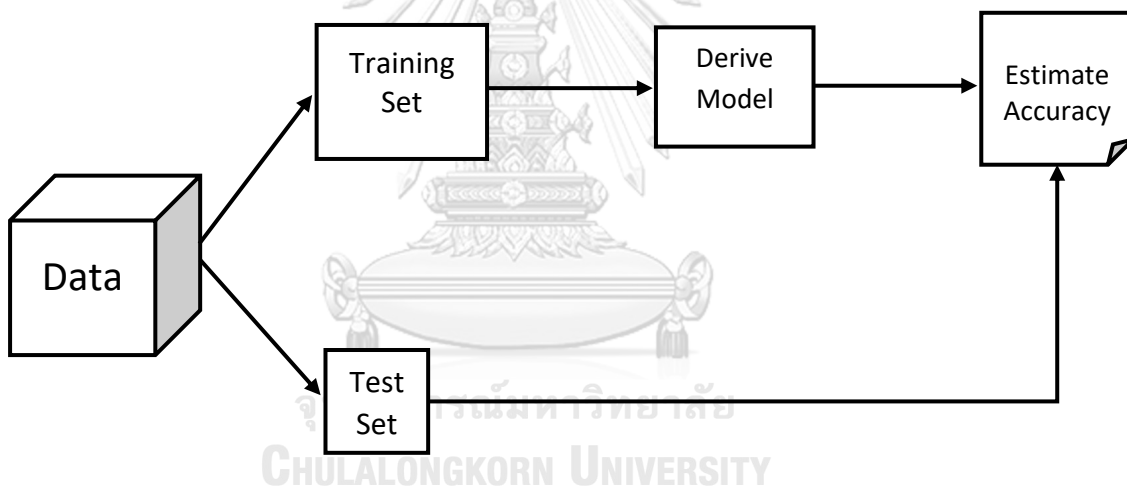
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.14)$$

2.4.2 การประเมินความถูกต้องของตัวจำแนกข้อมูล

จากตัวชี้วัดความถูกต้องของการจำแนกข้อมูล สามารถประยุกต์ใช้ตัวชี้วัดเหล่านั้นในการประเมินความถูกต้องของตัวจำแนกข้อมูลได้ โดยมีวิธีการที่นิยมสำหรับประเมินตัวจำแนกข้อมูล 2 วิธี ได้แก่ Holdout และ Cross-validation [22]

2.4.2.1 Holdout

วิธีการ Holdout [22] จะแบ่งข้อมูลออกเป็น 2 ชุดด้วยวิธีการสุ่ม ได้แก่ ชุดข้อมูลฝึก และชุดข้อมูลทดสอบ ซึ่งส่วนใหญ่จะแบ่งให้ชุดข้อมูลฝึกมีจำนวน 2 ใน 3 ของชุดข้อมูลทั้งหมด อีก 1 ใน 3 ของชุดข้อมูลทั้งหมดจะเป็นชุดข้อมูลทดสอบ โดยชุดข้อมูลฝึกจะถูกใช้ในการสร้างตัวจำแนกข้อมูล ส่วนชุดข้อมูลทดสอบจะใช้ในการทดสอบประสิทธิภาพของตัวจำแนกข้อมูลที่สร้างขึ้น ขั้นตอนการประเมินความถูกต้องของตัวจำแนกข้อมูลแสดงดังรูปที่ 23

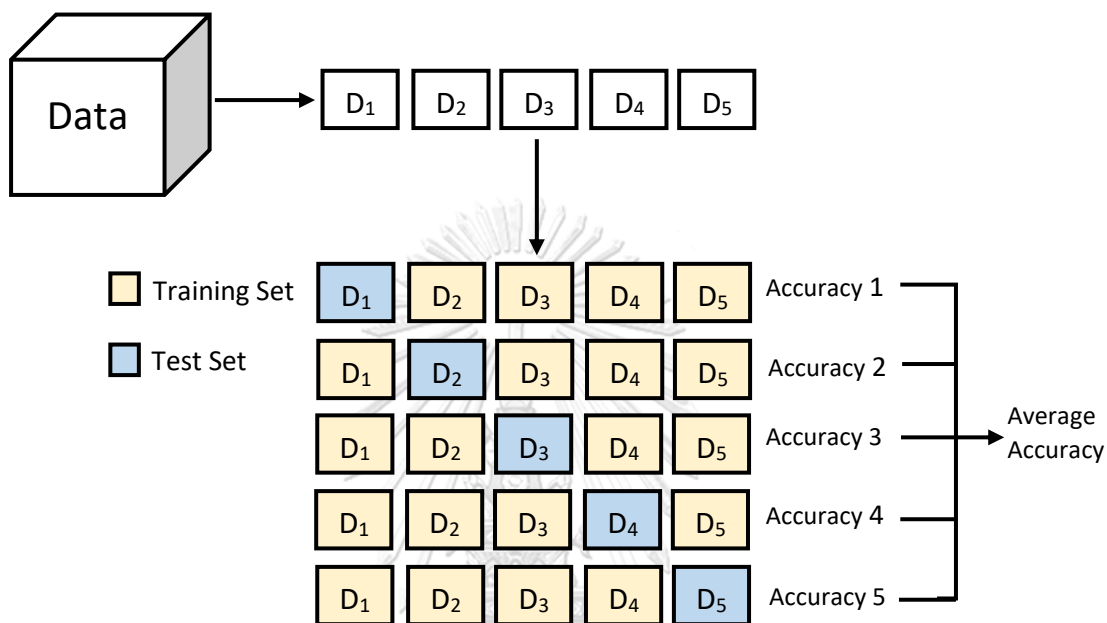


รูปที่ 23 การประเมินความถูกต้องด้วยวิธี Holdout [22]

2.4.2.2 Cross-validation

วิธีการ Cross-validation [22] หรือเรียกว่า “k-fold cross validation” จะเริ่มจากการแบ่งข้อมูลเป็น k ส่วนด้วยการสุ่ม ซึ่งแต่ละส่วนจะมีจำนวนข้อมูลเท่ากันแต่จะไม่มีข้อมูลซ้ำกัน จากนั้นจะทำการเรียนรู้และทดสอบตัวจำแนกข้อมูลทั้งสิ้น k ครั้ง โดยในการทำงานในรอบที่ i จะกำหนดให้ชุดข้อมูลย่อย D_i ให้เป็นชุดข้อมูลทดสอบ และชุดข้อมูลอื่นๆ ที่เหลือเป็นชุดข้อมูลฝึก ตัวอย่างเช่น กำหนดให้ $k=5$ ดังนั้นเรียกวิธีการประเมินนี้ว่า 5-fold cross validation จากนั้นแบ่งข้อมูลออกเป็น 5 ส่วนดังนี้ D_1, D_2, D_3, D_4 และ D_5 ในการทำงานรอบแรก ชุดข้อมูลย่อย D_1 เป็นชุดข้อมูลทดสอบ และชุดข้อมูลย่อย D_2, D_3, D_4 และ D_5 จะเป็นชุดข้อมูลสอน ในการทำงานรอบ

ที่สอง ชุดข้อมูลย่อย D_2 เป็นชุดข้อมูลทดสอบ และชุดข้อมูลย่อย D_1, D_3, D_4 และ D_5 จะเป็นชุดข้อมูลสอน โดยจะทำงานแบบนี้ไปจนครบ 5 รอบ โดยในแต่ละรอบจะให้ค่าความแม่นยำ (Accuracy) เมื่อทำงานครบ 5 รอบ สามารถสรุปผลการประเมินความแม่นยำได้ด้วยค่าเฉลี่ยของค่าความแม่นยำ ตัวอย่างการประเมินความถูกต้องด้วยวิธี Cross-validation แสดงดังรูปที่ 24



รูปที่ 24 ตัวอย่างการประเมินความถูกต้องด้วยวิธี Cross-validation โดยกำหนดให้ $k = 5$

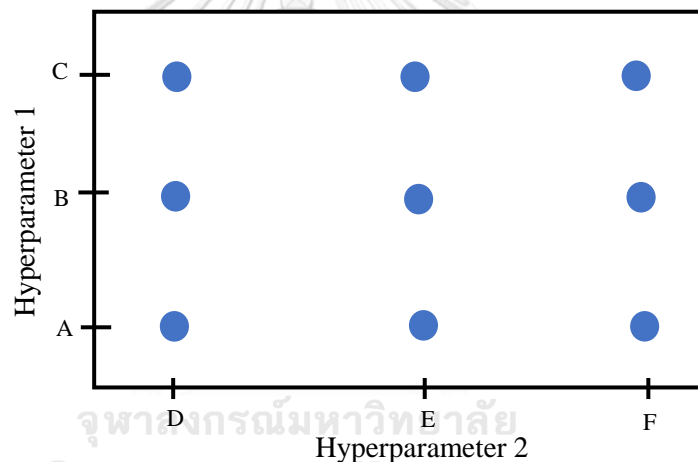
2.5 การหา Hyperparameter ที่เหมาะสม

Hyperparameter คือพารามิเตอร์ที่ไม่ได้มาจากกระบวนการเรียนรู้ของตัวจำแนกข้อมูลหรือตัวทำนายข้อมูล แต่เป็นพารามิเตอร์ที่ถูกปรับค่าจากภายนอกก่อนที่จะนำอัลกอริทึมไปฝึก โดยหากเลือกใช้ Hyperparameter ได้เหมาะสมกับชุดข้อมูลที่ใช้ฝึก จะทำให้โมเดลจำแนกข้อมูลหรือทำนายข้อมูลมีประสิทธิภาพสูง ตัวอย่างของ Hyperparameter เช่น จำนวนของ Decision Tree และความลึกของแต่ละ Decision Tree ที่ใช้สำหรับสร้างโมเดลด้วยอัลกอริทึม Random Forest เป็นต้น โดยวิธีการแบบดั้งเดิมที่ใช้หา Hyperparameter ที่เหมาะสม ได้แก่ Manual Search, Grid Search และ Random Search [23]

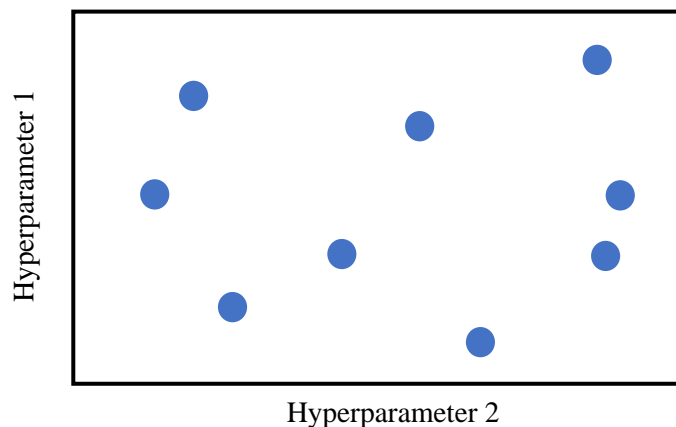
1. Manual Search เป็นวิธีที่เลือกค่า Hyperparameter จากประสบการณ์ของผู้สร้างโมเดล โดยผู้สร้างโมเดลใช้ประสบการณ์คาดการณ์ว่าช่วงไหนของแต่ละ Hyperparameter ที่ทำให้โมเดลที่สร้างมีความแม่นยำสูง โดยจะเลือก Hyperparameter มาสร้างโมเดลและประเมินความถูกต้องของโมเดลไปเรื่อยๆ จนกว่าจะได้ค่าความแม่นยำที่พอใจ [23]

2. Grid Search เป็นเทคนิคที่เข้าใจง่ายและตรงไปตรงมา โดยจะกำหนด Hyperparameter ไว้ทุกชุด และประเมินค่าความถูกต้องของโมเดลที่สร้างมาจาก Hyperparameter แต่ละชุดด้วยวิธี Cross-validation แล้วเลือกชุด Hyperparameter ที่ให้ค่าความแม่นยำสูงสุดหรือเหมาะสมที่สุดกับชุดข้อมูลที่น่ามาฝึก รูปแบบการทำงานจะคล้ายกริดและค่าทั้งหมดจะอยู่ในรูปเมทริกซ์ [23] ดังแสดงในรูปที่ 25

3. Random Search เป็นวิธีที่พัฒนามาจาก Grid Search เปลี่ยนจากการนำ Hyperparameter ทุกชุดมาประเมินความถูกต้อง เป็นการสุ่มเลือกชุด Hyperparameter จากกริดที่สร้างขึ้น ในการทำแค่กำหนดขอบเขตให้ Hyperparameter แต่ละตัว จากนั้นก็จะสุ่มชุดของ Hyperparameter มาประเมินความถูกต้องด้วยวิธี Cross-validation โดยวิธีนี้จะสร้างโมเดลเร็วกว่าวิธี Grid Search แต่ไม่รับประกันว่าจะได้โมเดลที่มีความแม่นยำสูงสุด ภาพการจำลองรูปแบบการทำงานของวิธี Random Search [23] แสดงดังรูปที่ 26



รูปที่ 25 การจำลองรูปแบบการทำงานของวิธี Grid Search [24]



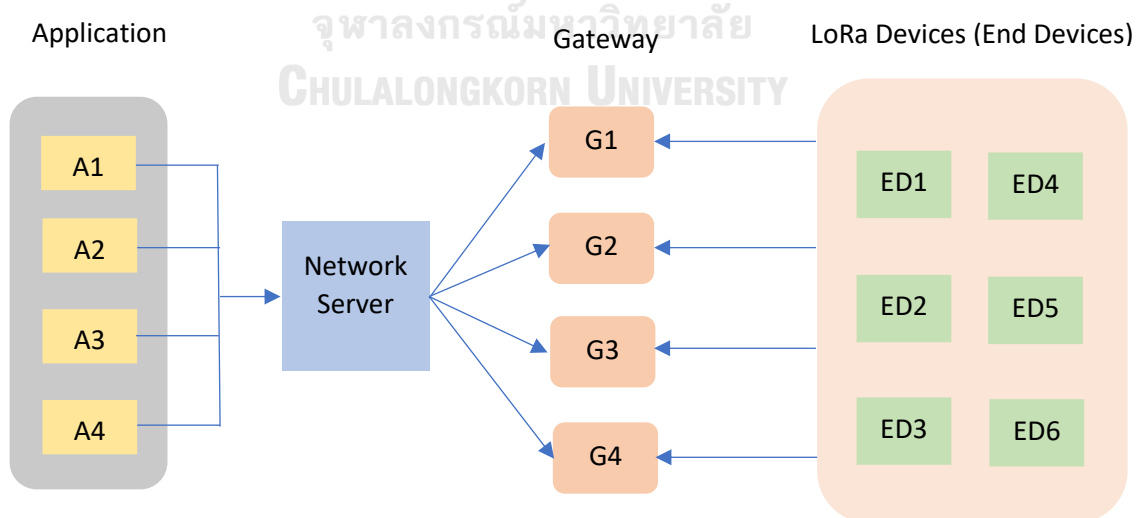
รูปที่ 26 การจำลองรูปแบบการทำงานของวิธี Random Search [24]

2.6 เทคโนโลยีสื่อสารแบบไร้สาย

การสื่อสารแบบไร้สายเป็นการแปลงสัญญาณจากไฟฟ้าไปเป็นสนามแม่เหล็กแล้วส่งไปยังภาครับผ่านสายอากาศ ในช่วงไม่กี่ปีที่ผ่านมาเทคโนโลยีการสื่อสารแบบไร้สายได้รับความสนใจอย่างมากทั้งในภาคอุตสาหกรรม วิทยาศาสตร์ การแพทย์ และภาคส่วนอื่นๆ ที่ให้ความสนใจในเรื่องของเทคโนโลยี IoT (Internet of Things) เนื่องจากเทคโนโลยีการสื่อสารแบบไร้สายเป็นเทคโนโลยีที่สำคัญของการใช้งาน IoT เทคโนโลยีที่โดดเด่นในการสื่อสารเพื่อรองรับ IoT ได้แก่ Zigbee, Bluetooth, NB-IoT และ LoRa [25]

2.6.1 Long Range (LoRa)

LoRa เป็นหนึ่งในเทคโนโลยีของ Low Power Wide Area (LPWA) ที่เกิดขึ้นใหม่เพื่อรองรับการสื่อสารในระยะไกลที่ใช้งานในช่วงความถี่ 1 GHz มีระยะสื่อสาร 15 กิโลเมตร ทั้งนี้เทคโนโลยี LoRa เริ่มแรกถูกพัฒนาขึ้นเพื่อใช้ในการจัดการอาหาร โดยใช้วิธีการมอดูเลชันสัญญาณด้วยเทคนิค Chirp Spread Spectrum (CSS) ทำให้มีความไวในการรับสัญญาณและสามารถป้องกันสัญญาณรบกวนได้ดี การส่งข้อมูล LoRa จะส่งผ่านเกตเวย์ โดยข้อมูลจะถูกส่งไปยังหลายเกตเวย์พร้อมกัน จากนั้นแต่ละเกตเวย์จะส่งต่อไปยังเน็ตเวิร์กเซิร์ฟเวอร์ผ่านช่องทางสื่อสารได้แก่ วิทยุอีเทอร์เน็ต หรือเซลลูลาร์ เน็ตเวิร์กเซิร์ฟเวอร์ยังทำหน้าที่ในการถอดความ (Decode) ของ แพ็กเกจจากอุปกรณ์รวมไปถึงการส่งแพ็กเกจส่งกลับไปยังอุปกรณ์ โดยโครงสร้างของ LoRa แสดงในรูปแบบที่เป็น LoRaWAN (Long Range Wide Area Network) [25] ดังแสดงในรูปที่ 27



รูปที่ 27 โครงสร้างของ LoRaWAN พร้อมทั้งแอปพลิเคชัน [25]

บทที่ 3

การออกแบบและพัฒนาระบบตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

งานวิจัย [3] เสนอวิธีการตรวจจับความผิดปกติภายในโรงไฟฟ้าพลังงานแสงอาทิตย์ขนาด 5 kW โดยใช้วิธีการเปรียบเทียบกำลังไฟฟ้าที่โรงไฟฟ้าจริงผลิตได้กับกำลังไฟฟ้าทางทฤษฎีจากโรงไฟฟ้าจำลอง และยังเสนอวิธีการจำแนกประเภทของความผิดปกติที่เกิดขึ้นด้วยเทคนิค Machine Learning แบบ ANN โดยสามารถจำแนกความผิดปกติได้ 4 ประเภทได้แก่ การเปิดวงจร, การลัดวงจร, การบังเงาบางส่วน และการเสื่อมสภาพของอุปกรณ์ พารามิเตอร์ที่ป้อนเข้า (Input) ให้ Machine Learning ได้แก่ ความเข้มแสง, อุณหภูมิสภาพแวดล้อม, กระแสไฟฟ้าและแรงดันไฟฟ้าแต่ละสตริง ในงานวิจัย [4] ได้เสนอเทคนิค Machine Learning แบบ PNN ที่สามารถจำแนกประเภทของความผิดปกติได้แก่ การลัดวงจร 3 แผงในสตริง, การลัดวงจร 10 แผงในสตริง, และการเปิดวงจรของสตริง โดยพารามิเตอร์ที่เป็น Input ให้ PNN ได้แก่ ความเข้มแสง อุณหภูมิแผงเซลล์แสงอาทิตย์ กระแสไฟฟ้าที่จุดกำลังสูงสุด และแรงดันไฟฟ้าที่จุดกำลังสูงสุด โดยที่พารามิเตอร์ดังกล่าวได้มาจากการจำลองระบบไฟฟ้าพลังงานแสงอาทิตย์ขนาด 9.54 kW ด้วยโปรแกรม PSIM และ Matlab ในงานวิจัย [6] ได้เสนอเทคนิค Machine Learning แบบ ANN ที่มีเลเยอร์ซ่อนเพียงเลเยอร์เดียวเพื่อตรวจจับความผิดปกติของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 2.2 kW ติดตั้งแบบสตริงเดี่ยว 10 แผง โดยการใช้พารามิเตอร์ป้อนเข้าให้ ANN เพียง 2 ตัว ได้แก่ ความเข้มแสงและกำลังไฟฟ้า เพื่อตรวจจับความผิดปกติในกรณีที่แผงเซลล์แสงอาทิตย์ถูกตัดการเชื่อมต่อจากสตริงตั้งแต่ 1 แผง จนกระทั่งตัดการเชื่อมต่อไป 9 แผง อีกทั้งยังมีการนำอัลกอริทึมที่เสนอไปทดสอบกับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 4.16 kW ติดตั้งแบบ 4 สตริง สตริงละ 8 แผง เพื่อทดสอบความยืดหยุ่นของอัลกอริทึม ซึ่งอัลกอริทึมที่เสนอมีความแม่นยำสูงกว่า 97% ในงานวิจัย [7] ได้เสนอวิธีการตรวจจับและจำแนกความผิดปกติด้วยเทคนิค RDM ด้วยการใช้เซนเซอร์วัดแรงดันไฟฟ้าติดที่แต่ละสตริง และระบุตำแหน่งความผิดปกติด้วยเทคนิค Machine Learning แบบ ANN

จากงานวิจัยที่กล่าวมาข้างต้น เป็นการเสนอวิธีการตรวจจับและจำแนกประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วยการผสมผสานเทคนิค Machine Learning กับเทคนิคอื่นๆ เช่น MBDM และ RDM เป็นต้น โดยพารามิเตอร์ที่นำมาฝึกและทดสอบอัลกอริทึม Machine Learning ได้มาจากระบบจำลองระบบจริงผ่านโปรแกรม Matlab หรือ PSIM และมาจากค่าพารามิเตอร์ที่วัดได้จากระบบจริง โดยอัลกอริทึมทาง Machine Learning ที่นิยมใช้ส่วนใหญ่จะเป็น Neural Network ไม่ว่าจะเป็น ANN หรือ PNN เป็นต้น จากงานวิจัย [26] ได้มีการรวบรวม

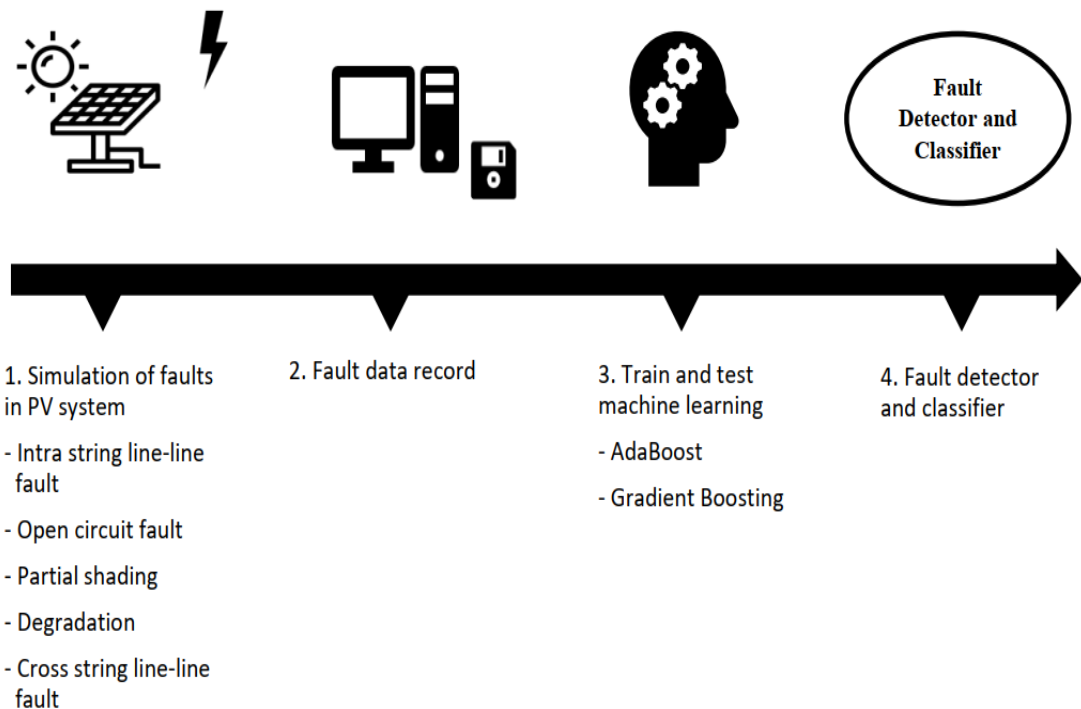
งานวิจัยที่ศึกษาการตรวจจับความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ไว้บางส่วน โดยเครื่องมือทาง Machine Learning ที่นิยมใช้กัน ได้แก่ Extreme learning machine (ELM), SVM, DT, Random Forest, KNN และในช่วงหลังมีการนำการเรียนรู้ของเครื่องแบบเชิงลึก (Deep Learning) มาใช้มากขึ้น ได้แก่ ANN, Long short-term memory (LSTM), Convolutional neural network (CNN) เป็นต้น ซึ่งมีหนึ่งเทคนิคที่งานวิจัยข้างต้นไม่ได้กล่าวคือการใช้เทคนิค Boosting ในการเพิ่มประสิทธิภาพของเครื่องมือทาง Machine Learning โดยเป็นการนำอัลกอริทึมทาง Machine Learning มาทำงานต่อกันเป็นลูกโซ่ทำให้สามารถเพิ่มประสิทธิภาพในงานจำแนกประเภทได้ โดยเครื่องมือที่ใช้เทคนิค Boosting ที่นิยมนำมาใช้ในงานจำแนกประเภทได้แก่ Adaboost และ Gradient Boosting (GB) ในงานวิจัย [8] ได้เสนอ Adaboost สำหรับการจำแนกประเภทความผิดปกติแบบลัดวงจร 3 แบบ ในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ ได้แก่ การลัดวงจรในสตริง, การลัดวงจรระหว่างสตริง, และการลัดวงจรจากสตริงถึงสายดิน ซึ่งพารามิเตอร์ที่เป็นอินพุตให้ Adaboost ได้แก่ กระแสไฟฟ้า แรงดันไฟฟ้าของระบบ กำลังไฟฟ้า ความเข้มแสง อุณหภูมิสิ่งแวดล้อม และความต้านทานขณะเกิดความผิดปกติ ในงานวิจัย [9] ได้มีการนำ Gradient Boosting มาร่วมทดสอบจำแนกประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ ได้แก่ การลัดวงจร และการบังเงาบางส่วน โดยพารามิเตอร์ที่เป็นอินพุตให้ GB ได้แก่ แรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า ความเข้มแสง อุณหภูมิพื้นผิวเซลล์แสงอาทิตย์แต่ละโมดูล อุณหภูมิพื้นผิวเซลล์แสงอาทิตย์เฉลี่ย และอุณหภูมิสภาพแวดล้อม โดยทั้งงานวิจัย [8] และ [9] ไม่ได้มีการนำ Adaboost และ Gradient Boosting มาเปรียบเทียบประสิทธิภาพการทำงานสำหรับการจำแนกประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยไม่สามารถสรุปได้ว่า Adaboost หรือ GB เครื่องมือใดจะจำแนกประเภทได้ดีกว่า และความเหนือกว่าในการจำแนกประเภทของเครื่องมือทั้งสองจะขึ้นอยู่กับชุดข้อมูลที่นำมาฝึก

ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอแนวทางการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยการใช้การเปรียบเทียบประสิทธิภาพระหว่าง Adaboost กับ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติ ได้แก่ การลัดวงจรระหว่างสายในสตริง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตริง โดยพารามิเตอร์ที่ใช้เป็นอินพุตให้แก่ Adaboost และ Gradient Boosting ได้แก่ กระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสง สำหรับชุดข้อมูลของพารามิเตอร์ดังกล่าวได้จากการจำลองความผิดปกติทั้ง 5 ประเภทข้างต้นบนระบบผลิต

ไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง โดยใช้เทคโนโลยีสื่อสารแบบไร้สาย LoRa ในการเก็บข้อมูล

3.1 ภาพรวมขั้นตอนการทำงานของวิธีการที่เสนอ

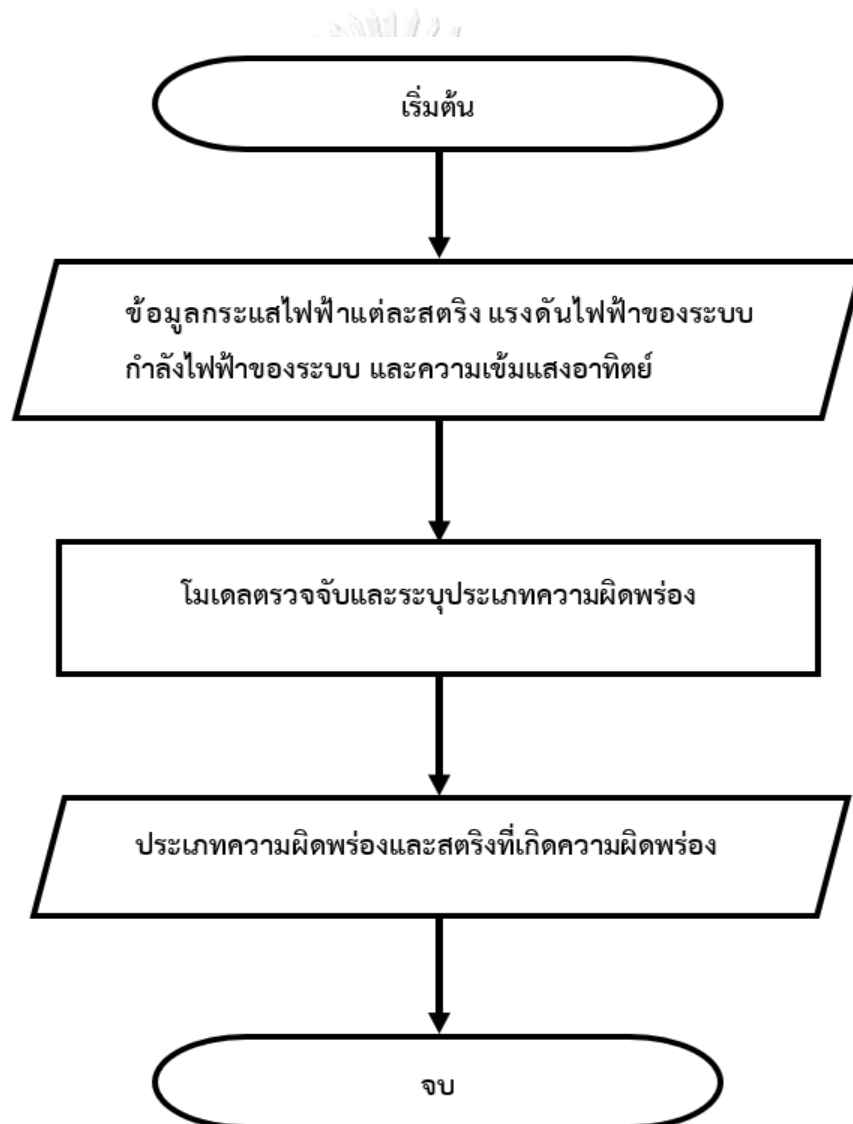
วิทยานิพนธ์นี้เสนอแนวทางการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้เทคนิค Machine Learning รูปที่ 28 แสดงขั้นตอนการทำงานของวิธีการที่เสนอ เริ่มจากการจำลองความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ 5 ประเภท ได้แก่ การลัดวงจรระหว่างสายในสตริง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตริง จากนั้นเก็บข้อมูลกระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ในขณะเกิดความผิดปกติ หลังจากนั้นนำข้อมูลเหล่านี้ไปฝึกและทดสอบอัลกอริทึม AdaBoost และ Gradient Boosting ซึ่งเป็นอัลกอริทึมทาง Machine Learning เพื่อสร้างโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยเปรียบเทียบความแม่นยำระหว่างโมเดล AdaBoost และ Gradient Boosting เพื่อเลือกโมเดลที่แม่นยำที่สุด



รูปที่ 28 ภาพรวมขั้นตอนการทำงานของวิธีการที่เสนอ

3.2 รายละเอียดการทำงานของระบบตรวจจับและระบุประเภทความผิดปกติ

วิทยานิพนธ์นี้ศึกษาและสร้างโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยรายละเอียดขั้นตอนการทำงานของระบบตรวจจับและระบุประเภทความผิดปกติแสดงดังรูปที่ 29 เริ่มต้นด้วยการเตรียมพารามิเตอร์ที่เป็นอินพุตได้แก่ กระแสไฟฟ้าสตรึงที่ 1 กระแสไฟฟ้าสตรึงที่ 2 แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ จากนั้นนำข้อมูลพารามิเตอร์ดังกล่าวป้อนเข้าสู่โมเดลตรวจจับและระบุประเภทความผิดปกติเพื่อประมวลผล โมเดลจะทำนายและแสดงผลออกมาว่าเป็นความผิดปกติประเภทใดและเกิดที่สตรึงใด



รูปที่ 29 ขั้นตอนการทำงานของระบบตรวจจับและระบุประเภทความผิดปกติ

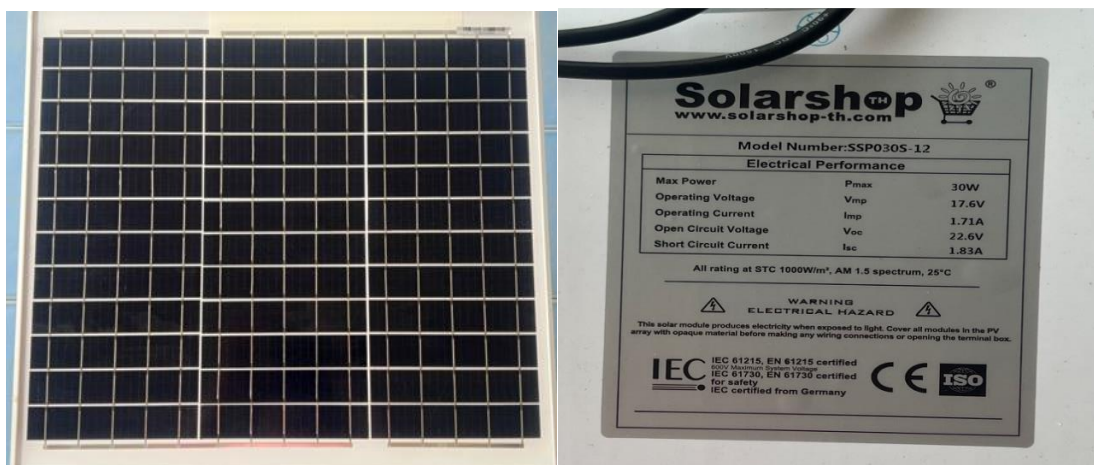
3.3 การจำลองความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

การจำแนกประเภทความผิดพลาดด้วย Machine Learning นั้นจำเป็นต้องใช้ข้อมูลที่มาจากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ที่จะตรวจจับ ไม่มีข้อมูลเป็นมาตรฐานเพื่อมาฝึกเครื่องมือทาง Machine Learning ให้เรียนรู้และสามารถตัดสินใจได้อย่างแม่นยำ [2]

งานวิจัยนี้ใช้พารามิเตอร์ 4 ชนิดสำหรับนำมาฝึกและทดสอบ AdaBoost และ Gradient Boosting ได้แก่ กระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ สำหรับชุดข้อมูลของพารามิเตอร์ดังกล่าวได้มาจากการจำลองความผิดพลาดบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง โดยแผงเซลล์แสงอาทิตย์ที่ใช้ในงานวิจัยนี้เป็นชนิด Monocrystalline ขนาด 30 W ดังแสดงในรูปที่ 30 และมีข้อมูลพิกัดทางไฟฟ้าของแผงเซลล์แสงอาทิตย์ดังตารางที่ 4

ตารางที่ 4 ข้อมูลพิกัดทางไฟฟ้าของแผงเซลล์แสงอาทิตย์

Name	Value
Peak Power Watts-PMAX	30 Wp
Power Output Tolerance-PMAX	0 ~ + 5 W
Maximum Power Voltage-VMPP	17.6 V
Maximum Power Current-IMPP	1.71 A
Open Circuit Voltage-VOC	22.6 V
Short Circuit Current-ISC	1.83 A
Cell Efficiency	17.6%

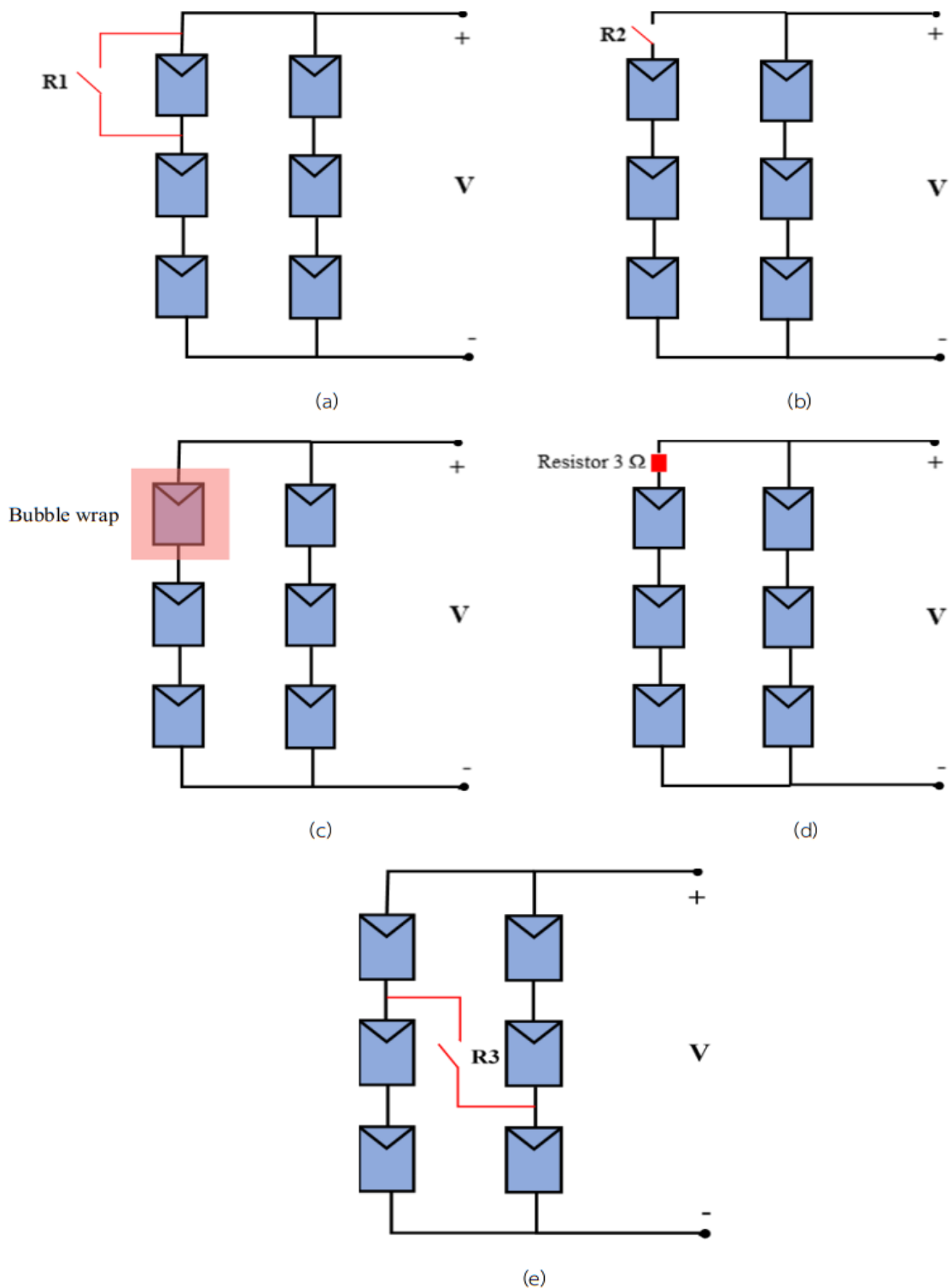


รูปที่ 30 แผงเซลล์แสงอาทิตย์ชนิด Monocrystalline ขนาด 30 W

วิทยานิพนธ์นี้พิจารณาความผิดพลาด 5 ประเภท ตามขอบเขตดังนี้

1. การลัดวงจรระหว่างสายในสตริง พิจารณาแค่การลัดวงจรของแผงที่ 1 ของสตริงที่ 1 และการลัดวงจรของแผงที่ 1 ของสตริงที่ 2 เท่านั้น
2. การเปิดวงจร พิจารณาทั้งการเปิดวงจรที่สตริงที่ 1 และการเปิดวงจรของสตริงที่ 2
3. การบังเงาบางส่วน พิจารณาแค่การเกิดการบังเงาที่แผงที่ 1 ของสตริงที่ 1 และการเกิดการบังเงาที่แผงที่ 1 ของสตริงที่ 2 เท่านั้น
4. การเสื่อมสภาพของอุปกรณ์ พิจารณาแค่การเสื่อมสภาพของสายของสตริงที่ 1 และการเสื่อมสภาพของสายของสตริงที่ 2 เท่านั้น
5. การลัดวงจรระหว่างสายข้ามสตริง พิจารณาแค่การลัดวงจรระหว่างขั้วลบของแผงที่ 1 ของสตริงที่ 1 กับขั้วลบของแผงที่ 2 ของสตริงที่ 2 เท่านั้น

การลัดวงจรระหว่างสายในสตริงถูกจำลองขึ้นด้วยการใช้สวิตช์รีเลย์ต่อคร่อมที่แผงที่ 1 ดังรูปที่ 31(a) เช่นเดียวกับการเปิดวงจรก็นำสวิตช์รีเลย์มาต่อแบบอนุกรมที่สตริงเพื่อตัดวงจรดังรูปที่ 31(b) ในส่วนของการบังเงาบางส่วนนั้นถูกจำลองขึ้นด้วยการใช้แผ่นกันกระแทกคลุมที่แผงที่ 1 ดังรูปที่ 31(c) สำหรับการเสื่อมสภาพของอุปกรณ์ถูกจำลองขึ้นด้วยการต่อตัวต้านทานขนาด 3 Ω อนุกรมเข้ากับสตริงดังรูปที่ 31(d) และการลัดวงจรระหว่างสายข้ามสตริงถูกจำลองขึ้นด้วยการใช้สวิตช์รีเลย์ต่อเข้าระหว่างขั้วลบของแผงที่ 1 ของสตริงที่ 1 กับขั้วลบของแผงที่ 2 ของสตริงที่ 2 ดังรูปที่ 31(e) ในส่วนของการลัดวงจรระหว่างสายในสตริง การเปิดวงจร การบังเงาบางส่วน และการเสื่อมสภาพของอุปกรณ์ ถูกจำลองขึ้นที่สตริงที่ 2 ในลักษณะเดียวกับสตริงที่ 1



รูปที่ 31 การจำลองความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์: การลัดวงจรระหว่างสายในสตริง (a), การเปิดวงจร (b), การบังเงาบางส่วน (c), การเสื่อมสภาพของอุปกรณ์ (d), การลัดวงจรระหว่างสายข้ามสตริง (e)

นอกจากชุดข้อมูลจากความผิดพลาดดังที่กล่าวมาแล้ว การฝึกและทดสอบ Machine Learning ให้สามารถตรวจจับความผิดพลาดจำเป็นต้องใช้ข้อมูลในสถานะการทำงานปกติมาฝึกด้วย เพื่อให้สามารถแยกความผิดพลาดประเภทต่างๆ ออกจากสถานะการทำงานปกติได้ รูปที่ 32 แสดงระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง โดยมีหลอดไฟฟ้ากระแสดตรงขนาด 15 W จำนวน 4 หลอดเป็นภาระไฟฟ้า (Load) รูปที่ 33 แสดงตัวอย่างการจำลองความผิดพลาดทั้ง 5 ประเภทบนระบบจริง



รูปที่ 32 ระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง



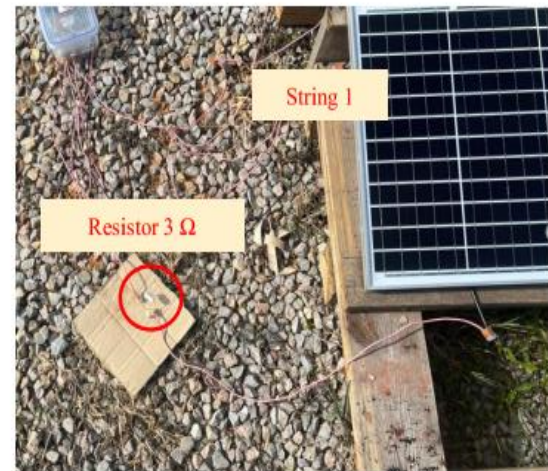
(a) Intra string line-line fault



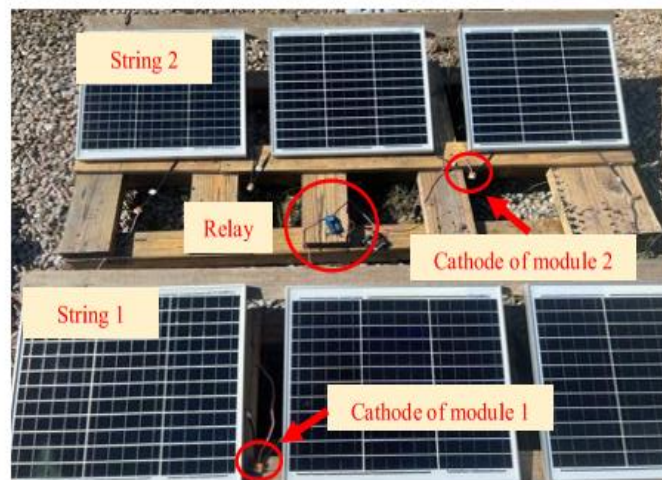
(b) Open circuit fault



(c) Partial shading



(d) Degradation



(e) Cross string line-line fault

รูปที่ 33 ตัวอย่างการจำลองความผิดพลาดบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์จริง

3.4 การเก็บรวบรวมข้อมูลสำหรับนำมาฝึกและทดสอบ Machine Learning

วิทยานิพนธ์นี้ใช้พารามิเตอร์ 4 ชนิด ในการฝึกและทดสอบ Machine Learning ได้แก่ กระแสไฟฟ้าแต่ละสตรึง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ดังนั้นจึงต้องเก็บข้อมูลของพารามิเตอร์ดังกล่าวทั้งในขณะทำงานปกติและขณะเกิดความผิดปกติประเภทต่างๆ โดยเก็บรวบรวมข้อมูลกระแสไฟฟ้าแต่ละสตรึง แรงดันไฟฟ้าของระบบ และกำลังไฟฟ้าของระบบด้วยการใช้เซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าต่อเข้ากับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตรึง สตรึงละ 3 แผง เมื่อได้ค่ากระแสไฟฟ้าแต่ละสตรึงและแรงดันไฟฟ้าของระบบมาแล้ว ทำให้สามารถหาค่ากำลังไฟฟ้าของระบบได้จากสมการที่ (3.1)

$$P = (I_1 + I_2) \times V \quad (3.1)$$

โดยที่ I_1 คือ ค่ากระแสไฟฟ้าของสตรึงที่ 1

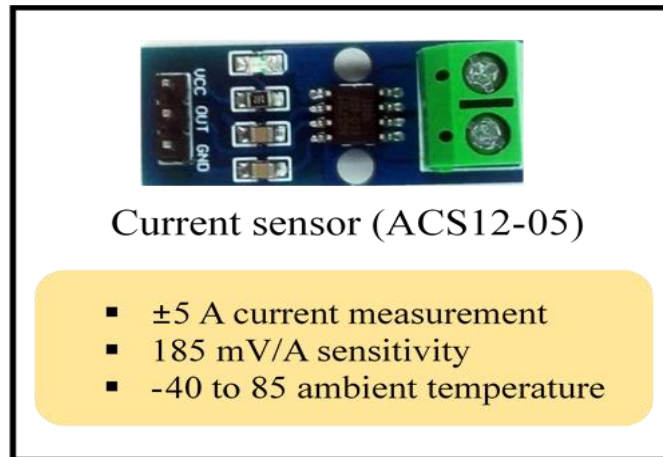
I_2 คือ ค่ากระแสไฟฟ้าของสตรึงที่ 2

V คือ ค่าแรงดันไฟฟ้าของระบบ

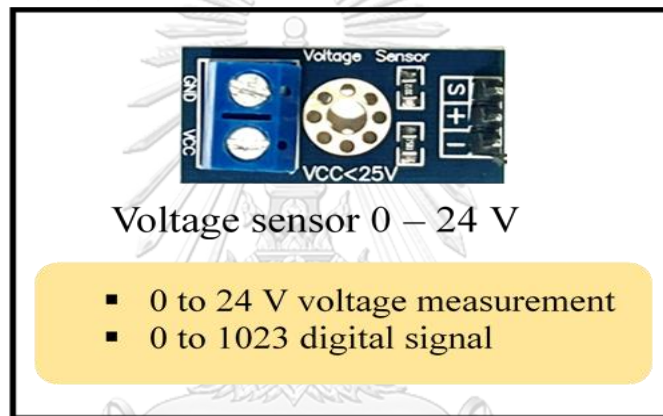
ในส่วนข้อมูลความเข้มแสงอาทิตย์นั้น ได้จากเซนเซอร์วัดความเข้มแสงของโรงไฟฟ้าพลังงานแสงอาทิตย์ อำเภอทับสะแก จังหวัดประจวบคีรีขันธ์ ซึ่งเป็นโรงไฟฟ้าในกำกับดูแลของการไฟฟ้าฝ่ายผลิตแห่งประเทศไทยที่ศูนย์วิเคราะห์สถานที่ในการทดลองงานวิทยานิพนธ์นี้

3.4.1 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้า

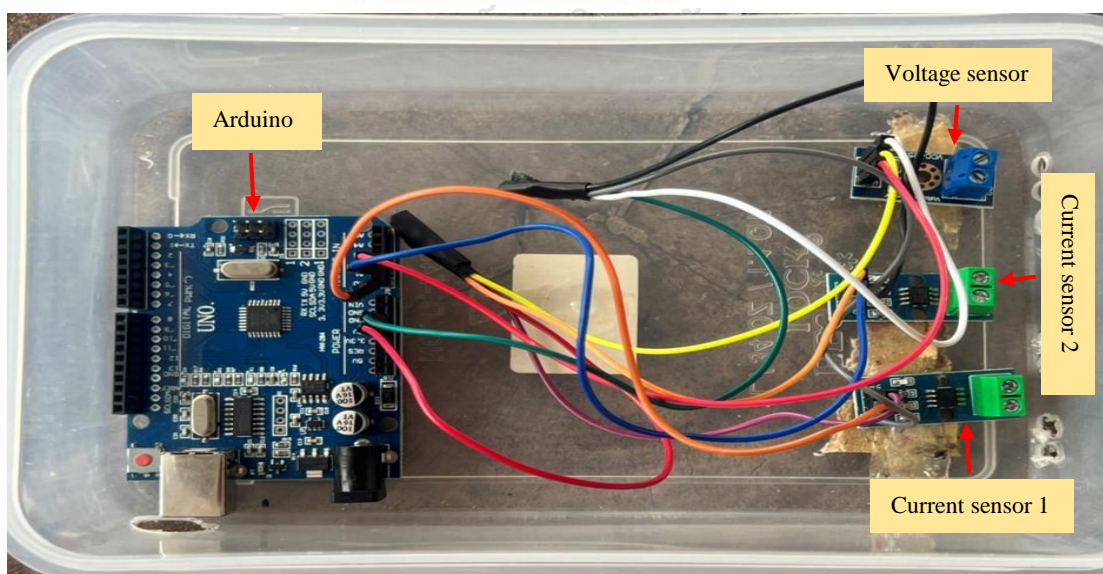
เครื่องมือสำหรับวัดกระแสไฟฟ้าและแรงดันไฟฟ้าสร้างจากเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าที่เก็บข้อมูลที่วัดได้ด้วยบอร์ดไมโครคอนโทรลเลอร์ชนิด Arduino Uno ซึ่งสะดวกในการใช้งานสำหรับการศึกษาเพราะเป็น open source และมีราคาต่ำ รูปที่ 34 แสดงเซนเซอร์วัดกระแสไฟฟ้าที่ใช้ในการเก็บค่ากระแสไฟฟ้าแต่ละสตรึง ซึ่งใช้เป็นเซนเซอร์ ACS712-05A เป็นเซนเซอร์ที่วัดกระแสไฟฟ้าได้ในช่วง -5 A ถึง 5 A โดยมีความไวของเอาต์พุตอยู่ที่ 185 mV/A หมายถึงแรงดันเอาต์พุตที่ออกมาจากเซนเซอร์ 185 mV จะเท่ากับกระแสไฟฟ้า 1 A ในส่วนของการวัดแรงดันไฟฟ้านั้น ใช้เซนเซอร์วัดแรงดันไฟฟ้าที่มีย่านการใช้งาน 0 V ถึง 24 V ดังแสดงในรูปที่ 35 โดยทั้งเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าใช้ไฟเลี้ยงในการทำงานที่ 5 V การรวมเข้าด้วยกันของเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้ากับบอร์ด Arduino Uno แสดงดังรูปที่ 36



รูปที่ 34 เซนเซอร์วัดกระแสไฟฟ้า



รูปที่ 35 เซนเซอร์วัดแรงดันไฟฟ้า



รูปที่ 36 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้า

3.4.2 การวัดกระแสไฟฟ้าและแรงดันไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

วิทยานิพนธ์นี้ต้องใช้ข้อมูลกระแสไฟฟ้าแต่ละสตรึงและข้อมูลแรงดันไฟฟ้าของระบบสำหรับนำไปฝึกและทดสอบ Machine Learning ดังนั้นในการวัดกระแสไฟฟ้าและแรงดันไฟฟ้าของระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตรึง สตรึงละ 3 แผง จำเป็นต้องใช้เซนเซอร์วัดกระแสไฟฟ้า 2 ตัว เพื่อวัดกระแสไฟฟ้า 2 สตรึง และใช้เซนเซอร์วัดแรงดันไฟฟ้า 1 ตัว เพื่อวัดแรงดันไฟฟ้ารวมของระบบ แต่เนื่องจากแผงเซลล์แสงอาทิตย์ 1 แผง มีแรงดันไฟฟ้าขณะเปิดวงจร (VOC) ที่ระบุในตารางที่ 4 อยู่ที่ 22.6 V ทำให้แรงดันไฟฟ้าขณะเปิดวงจรของระบบเท่ากับ 67.8 V ซึ่งมาจากการต่อกันแบบอนุกรมของแผงเซลล์แสงอาทิตย์ 3 แผง ด้วยเหตุนี้จึงส่งผลให้ไม่สามารถใช้เซนเซอร์วัดแรงดันไฟฟ้าตัวเดียววัดแรงดันไฟฟ้าของระบบได้โดยตรง ดังนั้นจึงแก้ปัญหานี้ด้วยการใช้วิธีของวงจรแบ่งแรงดัน (Voltage divider circuit) เพื่อลดทอนแรงดันไฟฟ้าให้อยู่ในย่านที่เซนเซอร์วัดแรงดันไฟฟ้ารับได้คือ 0 V ถึง 24 V โดยนำความต้านทานขนาด 50 Ω , 100 Ω และ 120 Ω มาต่ออนุกรมกัน จากนั้นนำตัวต้านทานที่ต่ออนุกรมกันนี้ไปต่อขนานเข้ากับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ และวัดแรงดันไฟฟ้าที่ตกคร่อมตัวต้านทานขนาด 50 Ω แทนการวัดแรงดันไฟฟ้าทั้งระบบโดยตรง โดยค่าแรงดันตกคร่อมตัวต้านทานขนาด 50 Ω หาได้จากสมการที่ (3.2)

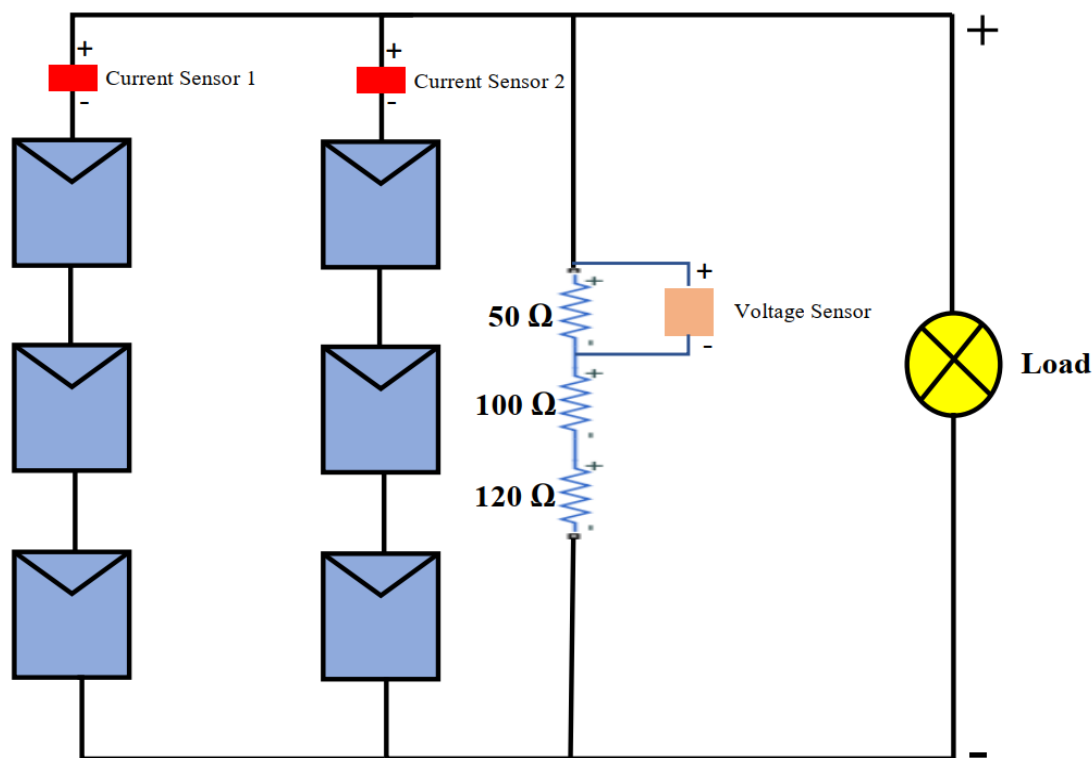
$$V_{out} = \left(\frac{R_1}{R_{total}} \right) \times V_{in} \quad (3.2)$$

โดยที่ V_{in} คือ แรงดันไฟฟ้าของระบบ

R_1 คือ ความต้านทานตัวที่เราต้องการทราบแรงดันตกคร่อม

R_{total} คือ ความต้านทานรวม

โดยแรงดันสูงสุดที่สามารถตกคร่อมตัวต้านทาน 50 Ω คือแรงดันในสภาวะที่ไม่มีโหลดมีค่า 12.56 V ทำให้ไม่เกินย่านการทำงานของเซนเซอร์วัดแรงดันไฟฟ้า โดยหากต้องการทราบค่าแรงดันไฟฟ้าของระบบสามารถใช้สมการที่ (3.2) แปลงกลับไปหา V_{in} ได้ รูปแบบการต่อเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าเข้ากับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์แสดงดังรูปที่ 37



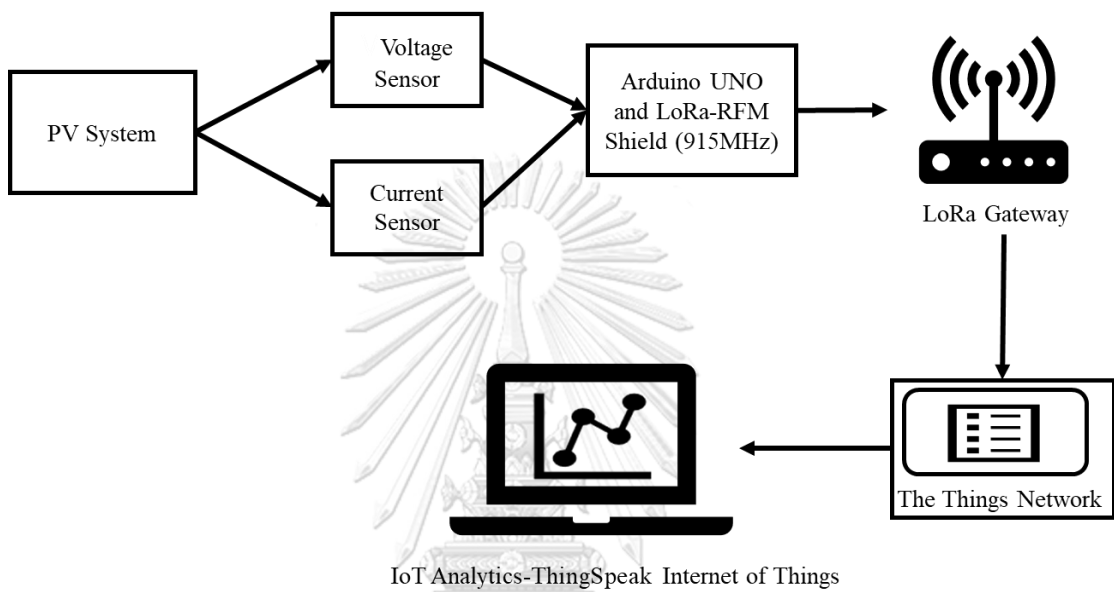
รูปที่ 37 การต่อเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าเข้ากับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

3.4.3 การส่งข้อมูลด้วยเทคโนโลยีสื่อสารแบบไร้สาย LoRa

การส่งค่ากระแสไฟฟ้าแต่ละสตรีงและแรงดันไฟฟ้าของระบบมายังคอมพิวเตอร์สำหรับเก็บรวบรวมข้อมูลนั้นใช้การส่งข้อมูลแบบไร้สายด้วยเทคโนโลยีสื่อสารไร้สาย LoRa ซึ่งใช้พลังงานต่ำและสามารถรับ-ส่งข้อมูลได้ในระยะทางไกลประมาณ 15 กิโลเมตร อีกทั้งยังมีความไวในการรับสัญญาณ และสามารถป้องกันสัญญาณรบกวนได้ดี ซึ่งการใช้เทคโนโลยี LoRa ทำให้ประหยัดและลดความยุ่งยากในการใช้สายไฟต่อจากอาเรียเซลล์แสงอาทิตย์มายังศูนย์ควบคุม อีกทั้งยังช่วยลดปัญหาความผิดพลาดของข้อมูลทางไฟฟ้า เนื่องจากการที่สายไฟแกว่งขณะกระแสลมแรง

ขั้นตอนการส่งข้อมูลแสดงดังรูปที่ 38 ต้นทางการส่งข้อมูลคือเซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์แรงดันไฟฟ้าที่ติดตั้งอยู่ที่ระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ ซึ่งส่งข้อมูลไปยัง Arduino โดยมี LoRa-RFM Shield ครอบ Arduino อยู่ LoRa-RFM Shield ทำหน้าที่เป็นลูกข่ายในการส่งข้อมูลด้วยสัญญาณ LoRa ไปยังเกตเวย์ หลังจากนั้นเกตเวย์จะยืนยันตัวตนของลูกข่ายและถอดรหัสสัญญาณด้วยแพลตฟอร์ม The Things Network ซึ่งทำหน้าที่บริหารจัดการโครงข่าย และสุดท้ายแสดงผลข้อมูลบน ThingSpeak Internet of Things โดยสามารถเก็บข้อมูลจาก ThingSpeak Internet of Things ในรูปแบบไฟล์ Excel ได้ทำให้สะดวกในการนำข้อมูลไปใช้งาน สำหรับเกตเวย์ที่

ใช้ในงานวิจัยนี้คือ Dragino LoRaWAN Gateway LG308-AS923-TH-EC25 รุ่น 4G ดังแสดงในรูปที่ 39 รูปที่ 40 แสดงตัวอย่างหน้าเว็บของ The Things Network รูปที่ 41 แสดงตัวอย่างกราฟของ กระแสไฟฟ้าและแรงดันไฟฟ้าที่แสดงบน ThingSpeak Internet of Things และเครื่องมือวัด กระแสไฟฟ้าและแรงดันไฟฟ้าที่มี LoRa-RFM Shield ทำหน้าที่ส่งข้อมูลแบบไร้สายแสดงดังรูปที่ 42



รูปที่ 38 การส่งข้อมูลทางไฟฟ้าจากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์มายังศูนย์ควบคุม



รูปที่ 39 Dragino LoRaWAN Gateway LG308-AS923-TH-EC25 รุ่น 4G

Applications > Solar Monitoring > End devices > Voltage and Current

Voltage and Current

ID: eui-70b3d57ed0053cc4

↑ 27 ↓ 18 • Last activity 18 hours ago

Overview Live data Messaging Location Payload formatters General settings

General information

End device ID: eui-70b3d57ed0053cc4

Frequency plan: Asia 915-928 MHz (AS923 Group 1) wi...

LoRaWAN version: LoRaWAN Specification 1.0.2

Regional Parameters version: RP001 Regional Parameters 1.0.2

Created at: Aug 3, 2022 11:47:16

Activation information

AppEUI: n/a

DevEUI: 70 B3 D5 7E D0 05 3C C4

Live data See all activity →

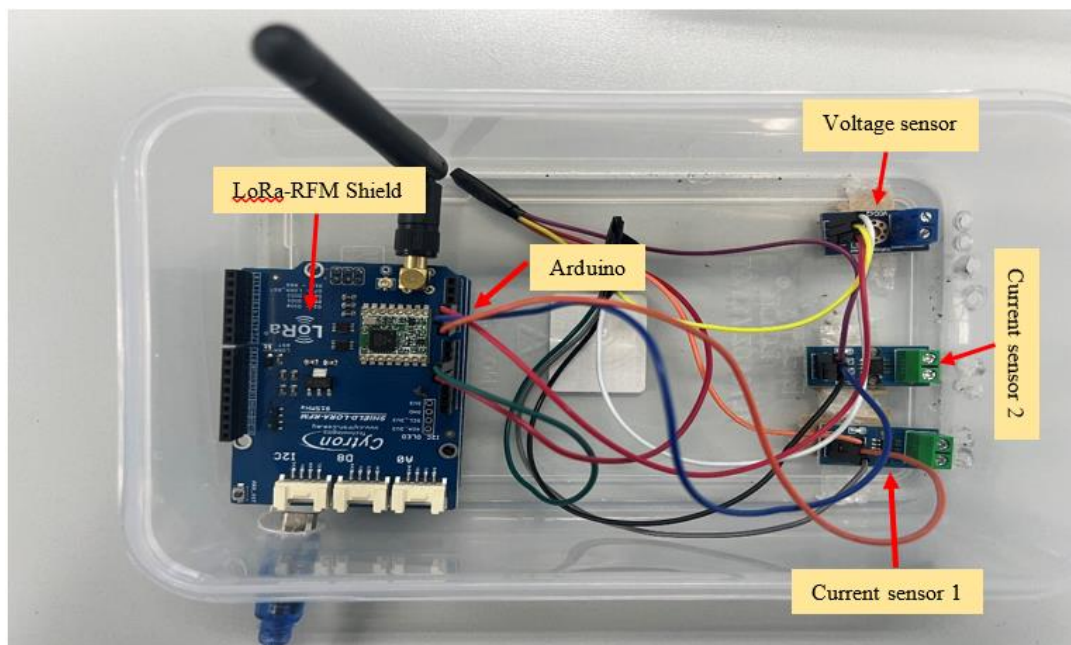
- ↓ 17:27:32 Schedule data downlink for transmission on Gateway
- ↑ 17:27:32 Forward uplink data message
- ↑ 17:27:32 Successfully processed data message
- ↓ 17:26:45 Failed to transmit downlink
- ↓ 17:26:44 Schedule data downlink for transmission on Gateway

Location Change location settings →

รูปที่ 40 ตัวอย่าง End device ที่กำหนดไว้บนหน้าเว็บของ The Things Network



รูปที่ 41 ตัวอย่างกราฟของกระแสไฟฟ้าและแรงดันไฟฟ้าที่แสดงบน IoT Analytics-ThingSpeak Internet of Things



รูปที่ 42 เครื่องมือวัดกระแสไฟฟ้าและแรงดันไฟฟ้าที่มี LoRa-RFM Shield ทำหน้าที่ส่งข้อมูลแบบไร้สาย

3.5 การฝึก Machine Learning

วิทยานิพนธ์นี้ใช้ภาษา Python บนโปรแกรม Jupyter Notebook เพื่อสร้างโมเดล Machine Learning โดยอัลกอริทึมทาง Machine Learning ที่ใช้สร้างโมเดล ได้แก่ AdaBoost และ Gradient Boosting สำหรับตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยการฝึกอัลกอริทึมทั้งสองใช้ข้อมูลกระแสไฟฟ้าแต่ละสตรึง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ในขณะที่งานปกติและขณะเกิดความผิดปกติ 5 ประเภท ได้แก่ การลัดวงจรระหว่างสายภายในสตรึง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตรึง โดยข้อมูลจะถูกจัดให้อยู่ในนามสกุลไฟล์ .csv ตัวอย่างข้อมูลที่ใช้การฝึกอัลกอริทึมแสดงดังรูปที่ 43 ใช้ข้อมูลชุดฝึกทั้งหมดทั้งสิ้น 1,732 กรณี แบ่งออกเป็น 10 คลาส โดยการกระจายของข้อมูลแสดงดังตารางที่ 5 ดังนี้ การทำงานปกติจำนวน 222 กรณี คิดเป็น 12.81%, การลัดวงจรระหว่างสายในสตรึงที่สตรึงที่ 1 จำนวน 169 กรณี คิดเป็น 9.76%, การเปิดวงจรที่สตรึงที่ 1 จำนวน 157 กรณี คิดเป็น 9.06%, การบังเงาบางส่วนจำนวนที่สตรึงที่ 1 จำนวน 161 กรณี คิดเป็น 9.30%, การเสื่อมสภาพของอุปกรณ์ที่สตรึงที่ 1 จำนวน 185 กรณี คิดเป็น 10.68% การลัดวงจรระหว่างสายในสตรึงที่สตรึงที่ 2 จำนวน 171 กรณี คิดเป็น 9.87%, การเปิดวงจรที่สตรึงที่ 2 จำนวน 158 กรณี คิดเป็น 9.12%, การบังเงาบางส่วนจำนวนที่

สตริงที่ 2 จำนวน 159 กรณี คิดเป็น 9.18%, การเสื่อมสภาพของอุปกรณ์ที่สตริงที่ 2 จำนวน 180 กรณี คิดเป็น 10.39% และการลัดวงจรระหว่างสายข้ามสตริงจำนวน 170 กรณี คิดเป็น 9.82%

	A	B	C	D	E	F	G	H	I	J	K	L
1	Current 1	Current 2	Voltage	Power	Irradiance	Fault						
2	0.250984	0.61205	62.3068	53.77289	699	Normal						
3	0.250984	0.559211	62.43824	50.58717	685	Normal						
4	0.224565	0.453534	62.70111	42.51752	689	Normal						
5	0.224565	0.559211	62.70111	49.14362	696	Normal						
6	0.198145	0.400695	62.83255	37.62666	719	Normal						
7	0.145307	0.61205	62.43824	47.28801	689	Normal						
8	0.303823	0.374275	63.09543	42.7849	678	Normal						
9	0.198145	0.427114	63.22686	39.5332	733	Normal						
10	0.171726	0.400695	59.28375	33.93525	648	Normal						
11	0.171726	0.400695	62.43824	35.74095	428	Normal						
12	0.277404	0.479953	61.78106	46.79028	455	Normal						
13	0.118887	0.400695	60.72956	31.55399	741	Normal						
14	0.250984	0.321437	58.23226	33.33336	758	Normal						
15	0.330242	0.559211	58.62657	52.1456	399	Normal						
16	0.330242	0.479953	63.88405	51.75855	833	Normal						
17	0.303823	0.479953	63.75261	49.96776	873	Normal						
18	0.250984	0.400695	61.25531	39.91879	864	Normal						
19	0.198145	0.427114	63.62117	39.77975	738	Normal						
20	0.250984	0.321437	60.07238	34.38668	423	Normal						
21	0.118887	0.400695	59.94094	31.14424	487	Normal						
22	0.224565	0.427114	60.07238	39.1479	457	Normal						
23	0.330242	0.347856	60.07238	40.73498	415	Normal						
24	0.356662	0.479953	62.17537	52.01682	428	Normal						
25	0.198145	0.506372	64.67267	45.56304	455	Normal						
26	0.198145	0.532792	61.64962	45.06199	616	Normal						
27	0.356662	0.479953	62.04393	51.90686	430	Normal						
28	0.303823	0.374275	60.861	41.26974	429	Normal						

รูปที่ 43 ตัวอย่างข้อมูลที่ใช้ในการฝึกอัลกอริทึม

ตารางที่ 5 การกระจายของข้อมูลความผิดปกติของชุดฝึกในแต่ละคลาส

Fault Type	class	Count	Percentage (%)
Normal	0	222	12.81
Intra string line-line fault, string 1	1	169	9.76
Open circuit fault, string 1	2	157	9.06
Partial shading, string 1	3	161	9.30
Degradation, string 1	4	185	10.68
Intra string line-line fault, string 2	5	171	9.87
Open circuit fault, string 2	6	158	9.12
Partial shading, string 2	7	159	9.18
Degradation, string 2	8	180	10.39
Cross string line-line fault	9	170	9.82
total	-	1,732	100

3.6 การประเมินความแม่นยำของอัลกอริทึมและแนวทางการทดสอบโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

3.6.1 การประเมินความแม่นยำของอัลกอริทึม

ก่อนการนำอัลกอริทึมไปพัฒนาเป็นโมเดลจำแนกประเภท จำเป็นที่จะต้องมีการประเมินความแม่นยำของอัลกอริทึมก่อน วิทยานิพนธ์นี้ใช้วิธี 5-fold cross validation ในการประเมินความแม่นยำของอัลกอริทึม โดยผลลัพธ์ของวิธีนี้จะแสดงค่าความแม่นยำของอัลกอริทึมจากการทดสอบด้วยข้อมูลชุดฝึก สำหรับการฝึกอัลกอริทึมทาง Machine Learning นั้น แต่ละชุดข้อมูลที่นำมาฝึกจะมี Hyperparameter ที่ให้ค่าความแม่นยำสูงสุดในแต่ละอัลกอริทึมไม่เหมือนกัน ดังนั้นในการเปรียบเทียบความแม่นยำของแต่ละอัลกอริทึม จึงต้องมีการหา Hyperparameter ที่ให้ค่าความแม่นยำสูงสุดของแต่ละอัลกอริทึม โดยวิทยานิพนธ์นี้ใช้วิธี Grid search

3.6.2 แนวทางการทดสอบโมเดล

วิทยานิพนธ์นี้มีการทดสอบเพื่อประเมินประสิทธิภาพของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ 5 การทดสอบ โดยแต่ละการทดสอบมีวิธีการดังนี้

3.6.2.1 วิธีการทดสอบความสามารถของ AdaBoost และ Gradient Boosting ในการพัฒนาอัลกอริทึมที่อ่อนแอให้มีความแม่นยำเพิ่มมากขึ้น

การทดสอบนี้เป็นการพิสูจน์ว่าการใช้เทคนิค AdaBoost และ Gradient Boosting สามารถปรับปรุงความแม่นยำในการจำแนกประเภทของตัวจำแนกประเภทที่อ่อนแอได้มากน้อยเพียงใด โดยตัวจำแนกประเภทที่อ่อนแอที่ใช้คือ Decision Tree ที่มีความลึกเท่ากับ 1 โดยใช้วิธี 5-fold cross validation ประเมินความแม่นยำของ Decision Tree, AdaBoost และ Gradient Boosting ในการจำแนกประเภท 10 คลาส ด้วยการใช้ชุดข้อมูลฝึกในตารางที่ 5 โดยกำหนดให้แต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1 และใช้วิธี Grid search สำหรับหา Hyperparameter ที่ให้ค่าความแม่นยำสูงสุดของแต่ละอัลกอริทึม

3.6.2.2 วิธีการทดสอบเปรียบเทียบความแม่นยำระหว่าง AdaBoost กับ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

การทดสอบนี้เป็นการเปรียบเทียบความแม่นยำระหว่าง AdaBoost และ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดย

เป็นการเปรียบเทียบเพื่อต้องการทราบว่า AdaBoost หรือ Gradient Boosting อัลกอริทึมใดมีความสามารถในการจำแนกประเภท 10 คลาสของความผิดพลาดได้ดีกว่ากัน ด้วยการใช้ชุดข้อมูลฝึกในตารางที่ 5 การทดสอบนี้ใช้วิธี 5-fold cross validation ประเมินความแม่นยำของทั้งสองอัลกอริทึม และใช้วิธี Grid search สำหรับหา Hyperparameter ที่ให้ค่าความแม่นยำสูงสุดของแต่ละอัลกอริทึม

3.6.2.3 วิธีการทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก

จากการทดสอบให้หัวข้อที่ 3.6.2.2 ซึ่งใช้ข้อมูลในตารางที่ 5 ซึ่งเป็นข้อมูลชุดฝึกจำนวน 100% การทดสอบนี้เป็นการศึกษาว่าจำนวนของข้อมูลชุดฝึกส่งผลอย่างไรต่อความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting โดยทดสอบความแม่นยำในการจำแนกประเภทของแต่ละอัลกอริทึมด้วยการลดจำนวนชุดฝึกลงเหลือ 90%, 80%, 70%, 60% และ 50% ตามลำดับ

3.6.2.4 วิธีการทดสอบความแม่นยำของอัลกอริทึมที่เลือกด้วยการใช้ชุดข้อมูลทดสอบ

ตารางที่ 6 การกระจายของข้อมูลความผิดพลาดชุดทดสอบในแต่ละคลาส

Fault Type	Class	Count	Percentage (%)
Normal	0	58	13.36
Intra string line-line fault, string 1	1	44	10.14
Open circuit fault, string 1	2	42	9.68
Partial shading, string 1	3	45	10.37
Degradation, string 1	4	46	10.60
Intra string line-line fault, string 2	5	46	10.60
Open circuit fault, string 2	6	34	7.83
Partial shading, string 2	7	36	8.29
Degradation, string 2	8	46	10.60
Cross string line-line fault	9	37	8.53
total	-	434	100

จากผลการทดสอบในหัวข้อที่ 3.6.2.1-3.6.2.3 ทำให้สามารถเลือกอัลกอริทึมที่ดีที่สุดในการตรวจจับและระบุประเภทความผิดพลาดในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ได้ การทดสอบนี้จะนำ

อัลกอริทึมที่ถูกเลือกและผ่านการฝึกแล้วมาทดสอบด้วยชุดข้อมูลทดสอบในตารางที่ 6 โดยแสดงผลการทดสอบเป็น Confusion matrix และตารางสรุปประสิทธิภาพตามตัวชี้วัดความแม่นยำ ได้แก่ Precision, Recall, F1-score และ Accuracy

3.6.2.5 วิธีการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

เมื่อได้อัลกอริทึมที่ดีที่สุดและผ่านการทดสอบด้วยชุดข้อมูลทดสอบแล้ว หลังจากนั้นก็นับที่กเป็นโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ การทดสอบนี้เป็นการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดล โดยจะทดสอบด้วยชุดข้อมูลความผิดปกติจำนวน 200 กรณี จากตารางที่ 7 การทดสอบเป็นการจำลองสถานการณ์ในการนำไปใช้งานว่าเมื่อมีข้อมูลอินพุตเข้ามาแบบอัตโนมัติอย่างต่อเนื่อง โมเดลจะสามารถทำนายผลออกมาได้ทันทีทันใดหรือไม่ โดยจะทดสอบป้อนอินพุตเข้าไปด้วยอัตราเร็วที่ต่างกันดังนี้ ป้อนอินพุตเข้าไปทุกๆ 5 วินาที, ป้อนอินพุตเข้าไปทุกๆ 2 วินาที และป้อนอินพุตเข้าไปทุกๆ 0.5 วินาที

ตารางที่ 7 การกระจายของข้อมูลความผิดปกติชุดทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

Fault Type	Class	Count	Percentage (%)
Normal	0	20	10
Intra string line-line fault, string 1	1	20	10
Open circuit fault, string 1	2	20	10
Partial shading, string 1	3	20	10
Degradation, string 1	4	20	10
Intra string line-line fault, string 2	5	20	10
Open circuit fault, string 2	6	20	10
Partial shading, string 2	7	20	10
Degradation, string 2	8	20	10
Cross string line-line fault	9	20	10
total	-	200	100

บทที่ 4

การทดสอบและผลการวิจัย

4.1 การทดสอบความสามารถของ AdaBoost และ Gradient Boosting ในการพัฒนาอัลกอริทึมที่อ่อนแอให้มีความแม่นยำเพิ่มมากขึ้น

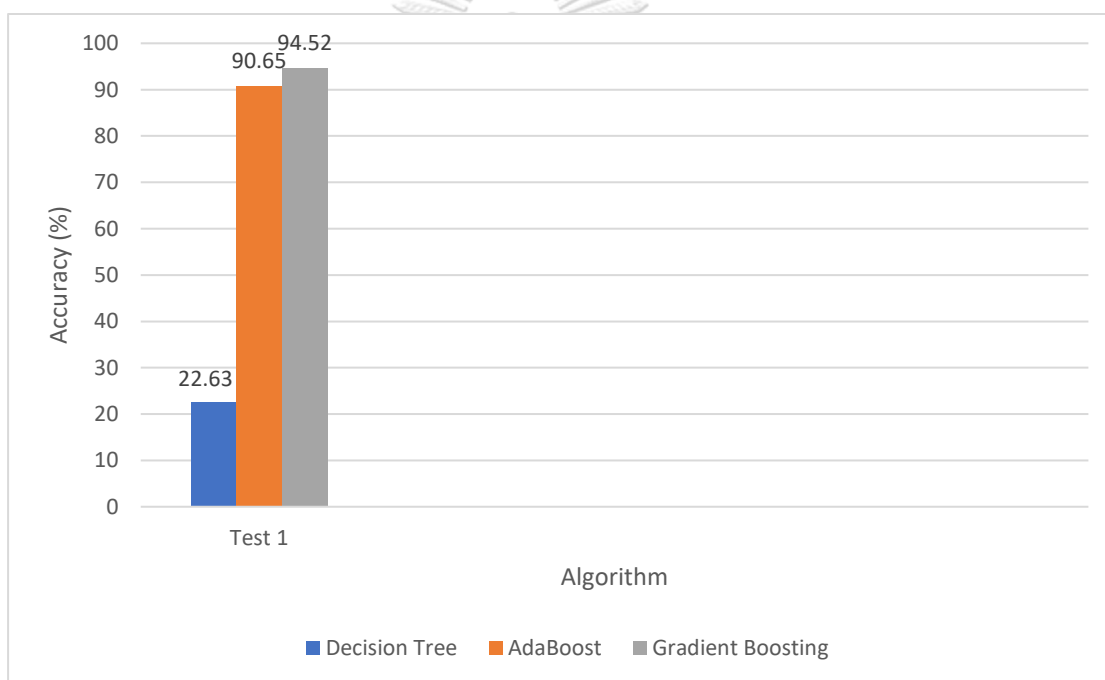
การทดสอบนี้กำหนดให้ Decision Tree ที่ใช้ในอัลกอริทึม AdaBoost และ Gradient Boosting มีความลึกเท่ากับ 1 เพื่อทดสอบว่าเมื่อนำ Decision Tree ที่มีความลึกเท่ากับ 1 ซึ่งถือว่าเป็นตัวจำแนกประเภทที่อ่อนแอมารพัฒนาด้วยการใช้เทคนิค AdaBoost และ Gradient Boosting จะทำให้มีความแม่นยำในการจำแนกประเภทเพิ่มมากขึ้นเพียงใด โดยในการทดสอบนี้เป็นการทดสอบจำแนกประเภท 10 คลาส ด้วยชุดข้อมูลในตารางที่ 5 ซึ่ง Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุดแสดงดังตารางที่ 8 โดยผลการทดสอบความแม่นยำด้วยวิธี 5-fold cross validation ของแต่ละอัลกอริทึมแสดงดังตารางที่ 9 และแสดงเป็นแผนภูมิดังรูปที่ 44

ตารางที่ 8 Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุดเมื่อกำหนดให้ ความลึกของ Decision Tree เท่ากับ 1

Algorithm	Hyperparameter
AdaBoost	<ul style="list-style-type: none"> - Estimator = Decision tree - Max dept = 1 - Number of estimator = 2,270 - Learning rate = 3.6 - Algorithm = SAMME
Gradient Boosting	<ul style="list-style-type: none"> - Estimator = Decision tree - Max dept = 1 - Number of estimator = 1500 - Learning rate = 0.04

ตารางที่ 9 ผลการทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เมื่อกำหนดให้แต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1

Algorithm	Accuracy (%)
Decision Tree	22.63
AdaBoost	90.65
Gradient Boosting	94.52



รูปที่ 44 แผนภูมิแสดงความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เมื่อกำหนดให้แต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1

จากผลการทดสอบในตารางที่ 9 และรูปที่ 44 แสดงให้เห็นว่าเมื่อนำ Decision Tree ที่มีความลึกเท่ากับ 1 ซึ่งถือว่าเป็นตัวจำแนกประเภทที่อ่อนแอมาดำเนินการพัฒนาด้วยการใช้เทคนิค AdaBoost และ Gradient Boosting ทำให้ความแม่นยำในการจำแนกประเภทเพิ่มขึ้นอย่างมาก โดย Decision Tree มีความแม่นยำเพียงแค่ 22.63% แต่เมื่อนำ Decision Tree ไปพัฒนาด้วยเทคนิค AdaBoost ให้ความแม่นยำถึง 90.65% และเมื่อนำไปพัฒนาด้วยเทคนิค Gradient Boosting ตัวจำแนกประเภทที่อ่อนแออย่าง Decision Tree ที่มีความลึกเท่ากับ 1 สามารถให้ความแม่นยำในการจำแนกประเภท

ได้สูงถึง 94.56% และมีความแม่นยำมากกว่า AdaBoost นอกจากนี้เมื่อพิจารณาจากตารางที่ 8 จะเห็นว่า Gradient Boosting สามารถลู่เข้าสู่ค่าความแม่นยำสูงสุดได้เร็วกว่า AdaBoost โดย Gradient Boosting ลู่เข้าสู่ค่าความแม่นยำสูงสุดด้วยการใช้ Decision Tree จำนวน 1500 ตัว ที่ Learning rate 0.04 ขณะที่ AdaBoost ต้องใช้ Decision Tree จำนวน 2,270 ตัว ที่ Learning rate 3.6 ในการเข้าสู่ค่าสูงสุด

4.2 การทดสอบเปรียบเทียบความแม่นยำระหว่าง AdaBoost กับ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

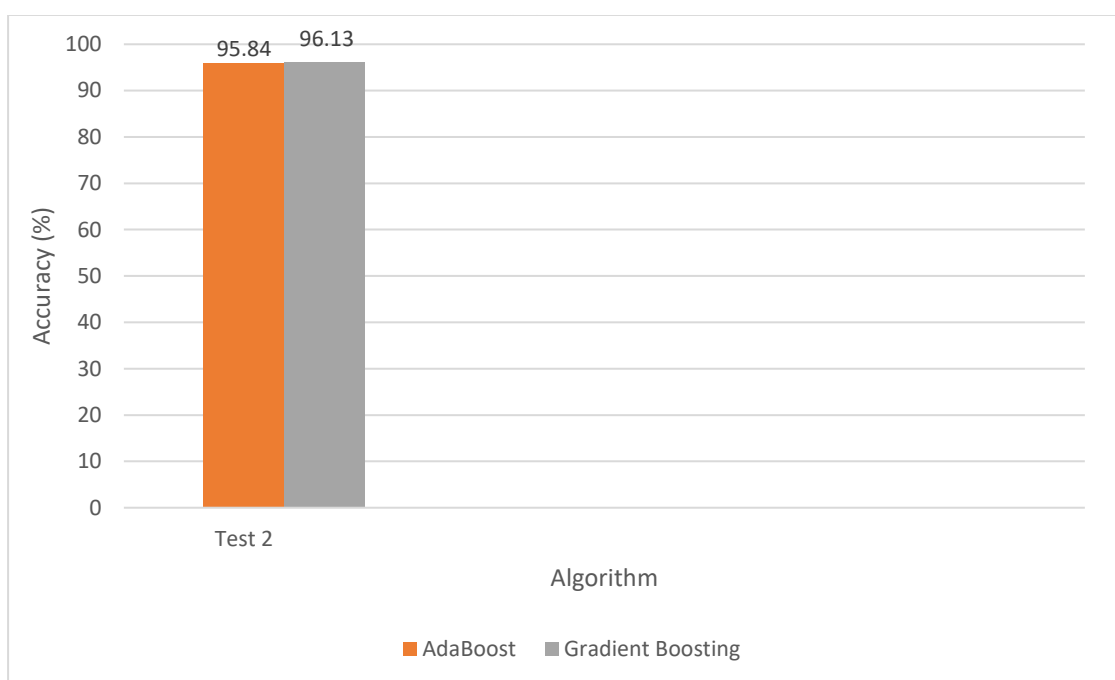
การทดสอบนี้เป็นการเปรียบเทียบความแม่นยำระหว่าง AdaBoost และ Gradient Boosting ในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยเป็นการทดสอบจำแนกประเภท 10 คลาส ด้วยชุดข้อมูลในตารางที่ 5 ซึ่งเป็นการทดสอบที่ต่างจากในหัวข้อ 4.1 คือจะไม่มีการกำหนดค่าความลึกของ Decision Tree ไว้ก่อน เพื่อให้ได้ AdaBoost และ Gradient Boosting ที่ทำงานเต็มประสิทธิภาพ โดย Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุดแสดงดังตารางที่ 10 สำหรับผลการทดสอบความแม่นยำด้วยวิธี 5-fold cross validation ของแต่ละอัลกอริทึมแสดงดังตารางที่ 11 และแสดงเป็นแผนภูมิดังรูปที่ 45

ตารางที่ 10 Hyperparameter ที่ทำให้แต่ละอัลกอริทึมมีความแม่นยำสูงสุด

Algorithm	Hyperparameter
AdaBoost	<ul style="list-style-type: none"> - Estimator = Decision tree - Max dept = 8 - Number of estimator = 825 - Learning rate = 0.8 - Algorithm = SAMME.
Gradient Boosting	<ul style="list-style-type: none"> - Estimator = Decision tree - Max dept = 7 - Number of estimator = 157 - Learning rate = 0.1

ตารางที่ 11 ผลการทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้ AdaBoost และ Gradient Boosting

Algorithm	Accuracy (%)
AdaBoost	95.84
Gradient Boosting	96.13



รูปที่ 45 แผนภูมิแสดงความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้ AdaBoost และ Gradient Boosting

จากผลการทดสอบในตารางที่ 11 และรูปที่ 45 แสดงให้เห็นว่า Gradient Boosting มีความแม่นยำมากกว่า AdaBoost ในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดย Gradient Boosting มีความแม่นยำ 96.13% ขณะที่ AdaBoost มีความแม่นยำ 95.84% และหากพิจารณาจากตารางที่ 10 จะเห็นว่า Gradient Boosting สามารถเข้าสู่ค่าความแม่นยำสูงสุดได้เร็วกว่า AdaBoost โดย Gradient Boosting เข้าสู่ค่าความแม่นยำสูงสุดด้วยการใช้ Decision Tree จำนวน 157 ตัว ความลึกของ Decision Tree เท่ากับ 7 ที่ Learning

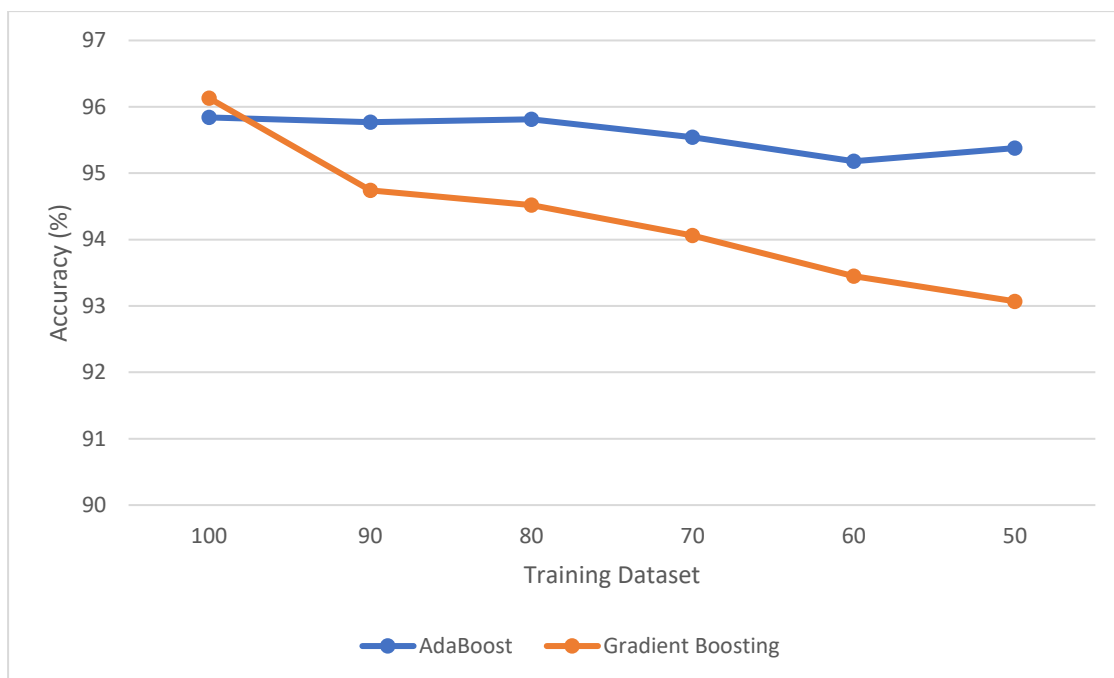
rate 0.1 ขณะที่ AdaBoost ต้องใช้ Decision Tree จำนวน 825 ตัว ความลึกของ Decision Tree เท่ากับ 8 ที่ Learning rate 0.8

4.3 การทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้า พลังงานแสงอาทิตย์ของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก

จากการทดสอบให้หัวข้อที่ 4.3 ใช้ข้อมูลในตารางที่ 5 ซึ่งเป็นข้อมูลชุดฝึกจำนวน 100% การทดสอบนี้เป็นการศึกษาว่าจำนวนของข้อมูลชุดฝึกส่งผลอย่างไรต่อความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting โดยทดสอบความแม่นยำในการจำแนกประเภทของแต่ละอัลกอริทึมด้วยการลดจำนวนชุดฝึกลงเหลือ 90%, 80%, 70%, 60% และ 50% ตามลำดับ ผลการทดสอบแสดงดังตารางที่ 12 และแสดงเป็นกราฟดังรูปที่ 46

ตารางที่ 12 ผลการทดสอบความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก

Training Dataset (%)	Accuracy (%)	
	AdaBoost	Gradient Boosting
100	95.84	96.13
90	95.77	94.74
80	95.81	94.52
70	95.54	94.06
60	95.18	93.45
50	95.38	93.07



รูปที่ 46 กราฟแสดงผลการทดสอบความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting ตามจำนวนข้อมูลชุดฝึก

จากผลการทดสอบในตารางที่ 12 และรูปที่ 46 จะเห็นว่า AdaBoost มีความเปลี่ยนแปลงของค่าความแม่นยำค่อนข้างน้อยเมื่อเทียบกับ Gradient Boosting เมื่อพิจารณาที่ AdaBoost หากจำนวนของข้อมูลฝึกใกล้เคียงกัน บางครั้งจำนวนข้อมูลฝึกที่น้อยกว่าก็ให้ค่าความแม่นยำสูงกว่าจำนวนข้อมูลฝึกที่มากกว่า เช่น จำนวนข้อมูลฝึก 80% ให้ค่าความแม่นยำสูงกว่าจำนวนข้อมูลฝึก 90% และจำนวนข้อมูลฝึก 50% ให้ค่าความแม่นยำสูงกว่าจำนวนข้อมูลฝึก 60% แต่หากจำนวนข้อมูลฝึกห่างกันมาก เช่น ห่างกันตั้งแต่ 20% จำนวนข้อมูลฝึกที่มากกว่าก็จะให้ค่าความแม่นยำสูงกว่า เมื่อพิจารณาที่ Gradient Boosting จะเห็นได้อย่างชัดเจนว่าเมื่อจำนวนข้อมูลฝึกลดลงค่าความแม่นยำก็น้อยลงไปด้วย

4.4 การนำอัลกอริทึมที่เลือกมาทดสอบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติด้วยชุดข้อมูลทดสอบ

จากการผลทดลองที่ในหัวข้อที่ 4.1-4.3 สามารถพิจารณาเลือกอัลกอริทึมที่ดีที่สุดสำหรับการตรวจจับและระบุประเภทความผิดปกติพร้อมได้แล้ว โดย Algorithm ที่ถูกเลือกคือ Gradient Boosting เพราะให้ค่าความแม่นยำสูงสุดเมื่อใช้ข้อมูลฝึกจำนวน 100% โดยมีความแม่นยำ 96.13%

การทดสอบนี้จะนำอัลกอริทึมที่ถูกเลือกและผ่านการฝึกแล้วมาทดสอบด้วยชุดข้อมูลทดสอบ ซึ่งเป็นชุดข้อมูลการเกิดความผิดพลาด 434 กรณี ในตารางที่ 6 ผลการจำแนกความผิดพลาดของชุดข้อมูลทดสอบด้วย Gradient Boosting แสดงดังรูปที่ 47

		Predicted										Σ
		0	1	2	3	4	5	6	7	8	9	
Actual	0	54	0	1	2	0	0	0	1	0	0	58
	1	0	42	0	0	0	0	0	0	0	2	44
	2	0	0	42	0	0	0	0	0	0	0	42
	3	0	0	0	44	1	0	0	0	0	0	45
	4	1	0	0	0	45	0	0	0	0	0	46
	5	0	0	0	0	0	46	0	0	0	0	46
	6	0	0	0	0	0	0	34	0	0	0	34
	7	1	0	0	0	0	0	0	35	0	0	36
	8	4	0	0	0	0	0	0	1	41	0	46
	9	0	3	0	0	0	0	0	0	0	34	37

Gradient Boosting

รูปที่ 47 Confusion matrix แสดงผลการจำแนกความผิดพลาดของชุดข้อมูลทดสอบด้วยการใช้ Gradient Boosting

จากรูปที่ 47 Gradient Boosting จำแนกประเภทความผิดพลาดผิดไป 17 กรณี จากการจำแนกทั้งหมด 434 กรณี ดังนั้นจึงจำแนกประเภทความผิดพลาดถูกต้อง 96.08% สังเกตว่าประเภทความผิดพลาดที่ Gradient Boosting จำแนกผิดค่อนข้างมากจะแบ่งเป็น 2 กลุ่ม กลุ่มแรกคือกลุ่มที่จำแนกผิดระหว่างการทำงานปกติ การบังเงาบางส่วน และการเสื่อมสภาพของอุปกรณ์ เนื่องจากทั้ง 3 ความผิดพลาดมีพฤติกรรมของพารามิเตอร์ทางไฟฟ้าไม่ต่างกันมาก ทำให้เมื่อเก็บค่าความเข้มแสงอาทิตย์ไม่ครอบคลุมมากพอ จึงเกิดการจำแนกผิดพลาดระหว่างกันค่อนข้างมากกว่าความผิดพลาดประเภทอื่น นอกจากนี้หากพิจารณาอย่างละเอียดจะเห็นว่า การจำแนกประเภทผิดระหว่างการทำงานปกติกับสถานะการทำงานปกติเกิดขึ้นบ่อยที่สุด เนื่องจากโดยปกติใน

สภาวะการทำงานปกติกระแสไฟฟ้าของสตริงที่ 1 จะน้อยกว่าสตริงที่ 2 ส่วนการเสื่อมสภาพของอุปกรณ์ที่สตริงที่ 2 ในช่วงแรงดันไฟฟ้าคงที่กระแสไฟฟ้าสตริงที่ 1 จะมากกว่าสตริงที่ 2 เล็กน้อย แต่ในช่วงที่มีการเปลี่ยนแปลงแรงดันไฟฟ้าโดยเฉพาะช่วงที่แรงดันตก ในช่วงแรกบางครั้งกระแสไฟฟ้าของสตริงที่ 1 จะน้อยกว่าสตริงที่ 2 เหมือนกับพฤติกรรมของกระแสที่สภาวะการทำงานปกติ ดังนั้นจึงเกิดการจำแนกผิดพลาดระหว่างกันบ่อยครั้ง กลุ่มที่จำแนกผิดค่อนข้างมากกลุ่มที่ 2 คือ กลุ่มที่จำแนกผิดระหว่างการลัดวงจรระหว่างสายในสตริงที่สตริงที่ 1 กับการลัดวงจรระหว่างสายข้ามสตริงเนื่องจาก 2 ความผิดพลาดนี้มีพฤติกรรมใกล้เคียงกันคือ กระแสไฟฟ้าของสตริงแรกจะติดลบและกระแสไฟฟ้าของสตริงที่ 2 จะเกิน 1 แต่ที่ความเข้มแสงอาทิตย์ใกล้เคียงกันแรงดันไฟฟ้าของการลัดวงจรระหว่างสายข้ามสตริงจะมากกว่าเล็กน้อย และในบางครั้งเมื่อความเข้มแสงขณะเกิดการลัดวงจรระหว่างสายข้ามสตริงตกลง ทำให้แรงดันไฟฟ้าตกไปใกล้เคียงกับช่วงแรงดันไฟฟ้าของการเกิดการลัดวงจรระหว่างสายในสตริงที่ 1 และเนื่องจากค่าความเข้มแสงอาทิตย์ที่เก็บมาไม่ครอบคลุมมากพอ ดังนั้นจึงทำให้เกิดการจำแนกผิดพลาดระหว่างกันบ่อยในช่วงดังกล่าว

ผลสรุปประสิทธิภาพในการจำแนกประเภทของแต่ละความผิดพลาดแสดงดังตารางที่ 13 โดยเป็นการนำผลการจำแนกประเภทความผิดพลาดจาก confusion matrix ในรูปที่ 47 มาคิดเป็นเปอร์เซ็นต์ของ Precision, Recall, F1-score และ Accuracy โดยค่าเฉลี่ย Precision, Recall และ F1-score คือ 96.4%, 96.2% และ 96.3% ตามลำดับ ขณะที่ค่าความแม่นยำในการจำแนกประเภทของอัลกอริทึมคือ 96.08%

ตารางที่ 13 ประสิทธิภาพในการจำแนกประเภทของ Gradient Boosting ด้วยชุดข้อมูลทดสอบ

Fault Type	Precision (%)	Recall (%)	F1-score (%)
Normal	90	93	92
Intra string line-line fault, string 1	93	95	94
Open circuit fault, string 1	98	100	99
Partial shading, string 1	96	98	97
Degradation, string 1	98	98	98
Intra string line-line fault, string 2	100	100	100
Open circuit fault, string 2	100	100	100
Partial shading, string 2	95	97	96
Degradation, string 2	100	89	94
Cross string line-line fault	94	92	93
Average	96.4	96.2	96.3
Accuracy = 96.08%			

4.5 การทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

เมื่อได้อัลกอริทึมที่ดีที่สุดและผ่านการทดสอบด้วยชุดข้อมูลทดสอบแล้ว หลังจากนั้นก็นำไปเป็นโมเดลเพื่อนำไปใช้งานด้วยนามสกุล .joblib โดยใช้คำสั่ง dump จากไลบรารี joblib ดังแสดงในรูปที่ 48

```
In [23]: from joblib import dump
         dump(GB_model, 'GB.joblib')
```

Out[23]: ['GB.joblib'] ← Fault detector and classifier

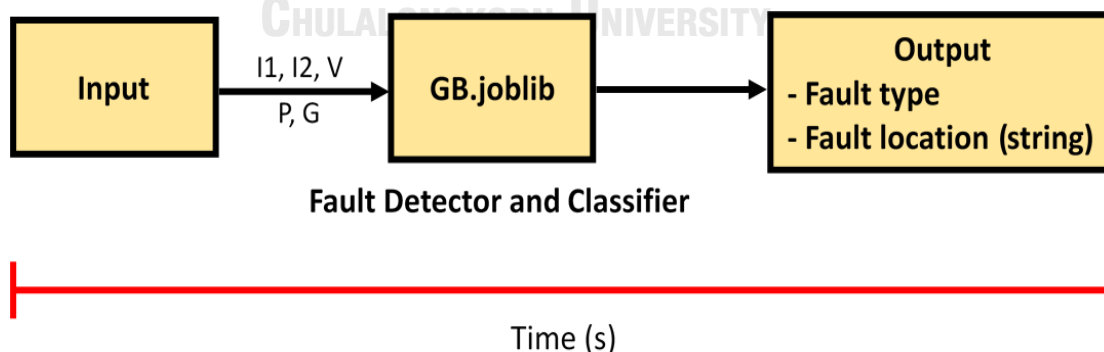
รูปที่ 48 การบันทึกอัลกอริทึมที่ดีที่สุดเพื่อเป็นโมเดลสำหรับนำไปใช้งาน

การเรียกโมเดลที่บันทึกไว้มาใช้งานทำได้ด้วยการใช้คำสั่ง load จากไลบรารี joblib ดังแสดงในรูปที่ 49

```
In [47]: from joblib import load
         GB = load('GB.joblib')
```

รูปที่ 49 การเรียกโมเดลที่บันทึกไว้มาใช้งาน

การทดสอบนี้เป็นการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ โดยจะทดสอบด้วยชุดข้อมูลความผิดปกติจำนวน 200 กรณี จากตารางที่ 7 ด้วยการป้อนข้อมูลอินพุต ได้แก่ กระแสไฟฟ้าแต่ละสตรึง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์เข้าไปเพื่อให้โมเดลตรวจจับและระบุประเภทความผิดปกติทำนายชนิดของความผิดปกติออกมาแบบอัตโนมัติอย่างต่อเนื่องจนครบ 200 กรณีความผิดปกติ ขั้นตอนการทำงานของโมเดลตรวจจับและระบุประเภทความผิดปกติ ดังรูปที่ 50 โดยจะทดสอบป้อนอินพุตเข้าไปทุกๆ 5 วินาที, 2 วินาที และ 0.5 วินาที ผลการทดสอบแสดงดังตารางที่ 14 และรูปที่ 51



รูปที่ 50 ขั้นตอนการทำงานของโมเดลตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

ตารางที่ 14 ผลการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลตรวจจับและระบุประเภทความผิดปกติพร้องในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์

Sampling Time (s)	Expect Time (s)	Test Time (S)	Delay(s)
5	1000	1002.518	2.518
2	400	402.765	2.765
0.5	100	102.924	2.924

		Predicted										
		0	1	2	3	4	5	6	7	8	9	Σ
Actual	0	15	0	1	0	0	0	0	1	3	0	20
	1	0	20	0	0	0	0	0	0	0	0	20
	2	0	0	20	0	0	0	0	0	0	0	20
	3	0	0	0	20	0	0	0	0	0	0	20
	4	0	0	0	0	20	0	0	0	0	0	20
	5	0	0	0	0	0	20	0	0	0	0	20
	6	0	0	0	0	0	0	20	0	0	0	20
	7	0	0	0	0	0	0	0	20	0	0	20
	8	2	0	0	0	0	0	0	0	18	0	20
	9	0	3	0	0	0	0	0	0	0	17	20

Gradient Boosting

รูปที่ 51 Confusion matrix แสดงผลการจำแนกความผิดปกติพร้องของชุดข้อมูลทดสอบการทำงานได้แบบเรียลไทม์ด้วยการใช้โมเดล Gradient Boosting

จากผลการทดสอบในตารางที่ 14 สังเกตว่าเมื่ออัตราเร็วในการป้อนข้อมูลอินพุตเพิ่มขึ้น ความล่าช้าในการทำนายผลก็จะเพิ่มขึ้นเล็กน้อย โดยป้อนข้อมูลอินพุตทุก 5 วินาที เกิดความล่าช้าขึ้น 2.518 วินาที เฉลี่ยแล้วล่าช้าไป 0.013 วินาทีต่อ 1 การทำนาย เมื่อป้อนข้อมูลอินพุตทุก 2 วินาที เกิดความล่าช้าขึ้น 2.765 วินาที เฉลี่ยแล้วล่าช้าไป 0.014 วินาทีต่อ 1 การทำนาย และเมื่อป้อนข้อมูลอินพุตทุก 0.5 วินาที เกิดความล่าช้าขึ้น 2.924 วินาที เฉลี่ยแล้วล่าช้าไป 0.015 วินาทีต่อ 1 การทำนาย โดยความล่าช้าที่เกิดขึ้นในแต่ละอัตราเร็วในการป้อนข้อมูลอินพุตถือว่าต่างกันน้อยมาก

ดังนั้นไม่ว่าจะป้อนข้อมูลด้วยอัตราเร็วที่น้อยหรือมาก โมเดลก็สามารถทำนายผลออกมาได้เร็วและสามารถทำงานได้แบบเรียลไทม์

นอกจากความเร็วในการทำนายผลของโมเดลแล้ว จากผลการจำแนกประเภทในรูปที่ 51 แสดงให้เห็นว่าโมเดลจำแนกประเภทความผิดปกติพร้อมผิดไป 10 กรณี จากทั้งหมด 200 กรณี คิดเป็นความแม่นยำ 95% ซึ่งถือว่ามีความแม่นยำในระดับที่ดี

4.6 สรุปผลการทดสอบ

การทดสอบในวิทยานิพนธ์ฉบับนี้เป็นการทดสอบเปรียบเทียบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ระหว่างอัลกอริทึม AdaBoost และ Gradient Boosting และทดสอบว่าโมเดลที่ถูกเลือกสามารถทำงานแบบเรียลไทม์ได้หรือไม่ ด้วยข้อมูลในสภาวะการทำงานปกติและข้อมูลจากการจำลองความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ที่ติดตั้งแบบ 2 สตริง สตริงละ 2 แผง ความผิดปกติที่พิจารณามี 5 ประเภท ได้แก่ การลัดวงจรระหว่างสายในสตริง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตริง

การทดสอบแบ่งเป็น 5 การทดสอบดังนี้

1. การทดสอบเปรียบเทียบความแม่นยำในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์โดยใช้ Decision Tree, AdaBoost และ Gradient Boosting เมื่อแต่ละอัลกอริทึมมีความลึกของ Decision Tree เท่ากับ 1 เป็นการทดสอบที่พิสูจน์ว่าเทคนิค AdaBoost และ Gradient Boosting สามารถปรับปรุงความแม่นยำของตัวจำแนกประเภทที่อ่อนแออย่าง Decision Tree ที่มีความลึกเท่ากับ 1 ได้ เมื่อ Decision Tree ที่มีความลึกเท่ากับ 1 มีความแม่นยำในการจำแนกประเภทแค่ 22.63% แต่เมื่อพัฒนาด้วยการใช้เทคนิค AdaBoost ให้ความแม่นยำเพิ่มขึ้นเป็น 90.65% และเมื่อพัฒนาด้วยเทคนิค Gradient Boosting สามารถให้ความแม่นยำได้ถึง 94.52%

2. เมื่อทราบแล้วว่า AdaBoost และ Gradient Boosting สามารถปรับปรุงความแม่นยำของตัวจำแนกประเภทที่อ่อนแอได้ จึงได้ทดลองเปรียบเทียบความแม่นยำในการจำแนกประเภทระหว่าง AdaBoost และ Gradient Boosting แบบให้ทั้งสองอัลกอริทึมทำงานอย่างเต็มประสิทธิภาพ ผลปรากฏว่า Gradient Boosting มีความแม่นยำ 96.13% ซึ่งมากกว่า AdaBoost ที่มีความแม่นยำ 95.84%

3. การทดสอบที่ 3 เป็นการทดสอบเพื่อต้องการทราบว่าจำนวนของข้อมูลชุดฝึกส่งผลอย่างไรต่อความแม่นยำในการจำแนกประเภทของ AdaBoost และ Gradient Boosting โดยทดสอบความแม่นยำในการจำแนกประเภทของแต่ละอัลกอริทึมด้วยจำนวนข้อมูลชุดฝึก 100%, 90%, 80%, 70%, 60% และ 50% ตามลำดับ ผลปรากฏว่า AdaBoost มีความเปลี่ยนแปลงของค่าความแม่นยำค่อนข้างน้อยเมื่อเทียบกับ Gradient Boosting และหากจำนวนข้อมูลชุดฝึกใกล้เคียงกัน ก็ไม่แน่นอนว่าจำนวนข้อมูลชุดฝึกที่มากกว่าจะให้ค่าความแม่นยำสูงกว่า แต่หากจำนวนข้อมูลชุดฝึกต่างกันมาก เช่น ต่างกัน 20% จำนวนข้อมูลชุดฝึกที่มากกว่าก็จะให้ค่าความแม่นยำสูงกว่า ส่วน Gradient Boosting ชัดเจนว่าหากจำนวนข้อมูลฝึกน้อยลงค่าความแม่นยำก็น้อยลงไปด้วย

4. เมื่อเลือกอัลกอริทึมที่ให้ค่าความแม่นยำสูงสุดในการจำแนกประเภทแล้ว ซึ่งก็คือ Gradient Boosting ที่ผ่านการฝึกด้วยจำนวนข้อมูลชุดฝึก 100% ในการทดสอบนี้จึงได้นำข้อมูลชุดทดสอบมาทดสอบโดยให้อัลกอริทึมที่เลือกทำนายผล ปรากฏว่ามีความแม่นยำถึง 96.08% และค่าเฉลี่ยของ Precision, Recall และ F1-score อยู่ที่ 96.4%, 96.2% และ 96.3% ตามลำดับ

5. หลังจากผ่านการทดสอบที่ 4 ซึ่งยืนยันว่าอัลกอริทึมมีความแม่นยำสูง จากนั้นบันทึกเป็นโมเดลเพื่อนำไปใช้งาน ในการทดสอบนี้เป็นการทดสอบการทำงานได้แบบเรียลไทม์ของโมเดลด้วยการป้อนข้อมูลอินพุต ได้แก่ กระแสไฟฟ้าแต่ละสตรึง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์เข้าไปเพื่อให้โมเดลทำนายชนิดและตำแหน่งสตรึงที่เกิดความผิดพลาดออกมาแบบอัตโนมัติอย่างต่อเนื่องจนครบ 200 กรณีความผิดพลาด โดยป้อนอินพุตเข้าไปทุกๆ 5 วินาที, 2 วินาที และ 0.5 วินาที ผลปรากฏว่าโมเดลสามารถทำนายความผิดพลาดได้อย่างรวดเร็วและทำงานได้แบบเรียลไทม์ รวมถึงมีความแม่นยำในการจำแนกประเภทความผิดพลาดได้ถูกต้อง 95%

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้นำเสนอแนวทางในการตรวจจับและระบุประเภทความผิดปกติในระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ด้วยการเปรียบเทียบความแม่นยำของอัลกอริทึมทาง Machine Learning 2 ชนิด ได้แก่ AdaBoost และ Gradient Boosting โดยพิจารณาความผิดปกติ 5 ประเภท ได้แก่ การลัดวงจรระหว่างสายในสตริง การเปิดวงจร การบังเงาบางส่วน การเสื่อมสภาพของอุปกรณ์ และการลัดวงจรระหว่างสายข้ามสตริง มีพารามิเตอร์ที่เป็นอินพุตให้กับอัลกอริทึม ได้แก่ กระแสไฟฟ้าแต่ละสตริง แรงดันไฟฟ้าของระบบ กำลังไฟฟ้าของระบบ และความเข้มแสงอาทิตย์ ซึ่งชุดข้อมูลของพารามิเตอร์ดังกล่าวได้มาจากการจำลองความผิดปกติบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ขนาด 180 W ติดตั้งแบบ 2 สตริง สตริงละ 3 แผง

การเก็บรวบรวมข้อมูลสำหรับนำไปฝึกและทดสอบ AdaBoost และ Gradient Boosting ใช้เซนเซอร์วัดกระแสไฟฟ้าและเซนเซอร์วัดแรงดันไฟฟ้าที่เก็บข้อมูลที่วัดได้ด้วยบอร์ดไมโครคอนโทรลเลอร์ชนิด Arduino Uno ซึ่งเหมาะในการใช้งานสำหรับการศึกษาเพราะเป็น open source และมีราคาต่ำ สำหรับการส่งข้อมูลที่วัดได้มายังศูนย์ควบคุมนั้น ใช้การส่งข้อมูลแบบไร้สายด้วยเทคโนโลยีสื่อสารไร้สาย LoRa เพื่อประหยัดและลดความยุ่งยากในการใช้สายไฟต่อจากอาเรียเซลล์แสงอาทิตย์มายังศูนย์ควบคุม อีกทั้งยังช่วยลดปัญหาความผิดพลาดของข้อมูลเนื่องจากสายไฟแกว่งเมื่อถูกลมพัด ข้อมูลที่เก็บได้จากระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์คือกระแสไฟฟ้าแต่ละสตริงและแรงดันไฟฟ้าของระบบ จากนั้นนำข้อมูลกระแสไฟฟ้าแต่ละสตริงและแรงดันไฟฟ้าของระบบมาหากำลังไฟฟ้าของระบบได้ ในส่วนของข้อมูลความเข้มแสงอาทิตย์ได้จากเซนเซอร์วัดความเข้มแสงของโรงไฟฟ้าพลังงานแสงอาทิตย์ อำเภอทับสะแก จังหวัดประจวบคีรีขันธ์ ซึ่งเป็นโรงไฟฟ้าในกำกับ การดูแลของการไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

ข้อมูลที่ใช้ในการฝึกและทดสอบ AdaBoost และ Gradient Boosting เป็นชุดข้อมูลความผิดปกติ 9 คลาส และสภาวะการทำงานปกติ 1 คลาส รวมเป็น 10 คลาส ได้แก่ การทำงานปกติ, การลัดวงจรระหว่างสายในสตริงที่สตริงที่ 1, การเปิดวงจรที่สตริงที่ 1, การบังเงาบางส่วนที่สตริงที่ 1, การเสื่อมสภาพของอุปกรณ์ที่สตริงที่ 1, การลัดวงจรระหว่างสายในสตริงที่สตริงที่ 2, การเปิดวงจรที่สตริงที่ 2, การบังเงาบางส่วนที่สตริงที่ 2, การเสื่อมสภาพของอุปกรณ์ที่สตริงที่ 2 และการลัดวงจรระหว่างสายข้ามสตริง โดยการตรวจจับและระบุประเภทความผิดปกติใช้วิธีการจำแนก

ประเภทความผิดพลาดทั้ง 9 คลาสนี้ออกจากสภาวะการทำงานปกติ ทำให้สามารถตรวจจับและระบุประเภทความผิดพลาดได้ในเวลาเดียวกัน ผลการทดสอบได้แสดงว่าเทคนิค AdaBoost และ Gradient Boosting สามารถเพิ่มความแม่นยำให้ตัวจำแนกประเภทที่อ่อนแอได้ โดย Decision Tree ที่มีความลึกเท่ากับ 1 มีความแม่นยำในการจำแนกประเภทความผิดพลาดเพียง 22.63% แต่เมื่อนำมาพัฒนาด้วยเทคนิค AdaBoost และ Gradient Boosting ให้ค่าความแม่นยำถึง 90.65% และ 94.52% ตามลำดับ เมื่อทดสอบเปรียบเทียบประสิทธิภาพระหว่าง AdaBoost กับ Gradient Boosting แบบเต็มประสิทธิภาพการทำงาน ผลปรากฏว่า Gradient Boosting มีความแม่นยำถึง 96.13% ซึ่งมากกว่า AdaBoost ที่มีความแม่นยำ 95.84% โดยจำนวนของชุดข้อมูลฝึกมีผลต่อทั้ง AdaBoost และ Gradient Boosting คือเมื่อจำนวนชุดข้อมูลฝึกน้อยลงก็จะทำให้ค่าความแม่นยำลดลง แต่มีข้อยกเว้นสำหรับ AdaBoost เพราะหากจำนวนข้อมูลชุดฝึกต่างกันไม่มาก เช่น ต่างกัน 10% บางครั้งจำนวนข้อมูลฝึกที่น้อยกว่าก็มีความแม่นยำมากกว่าเล็กน้อย ในการทดสอบสุดท้ายเป็นการนำโมเดลตรวจจับและระบุประเภทความผิดพลาดที่เลือกนั้นคือ Gradient Boosting มาทดสอบการทำงานได้แบบเรียลไทม์ ผลปรากฏว่าโมเดลทำนายผลได้อย่างรวดเร็วและสามารถทำงานได้แบบเรียลไทม์ รวมถึงมีความแม่นยำถึง 95%

จากผลการทดสอบทำให้โมเดลตรวจจับและระบุประเภทความผิดพลาดที่ได้ศึกษาและพัฒนาอาจจะนำไปใช้ประโยชน์กับระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ได้ ด้วยการใช้งานร่วมกับระบบตรวจติดตามของโรงไฟฟ้าพลังงานแสงอาทิตย์ที่รองรับการใช้งานอัลกอริทึมที่พัฒนาด้วยภาษา Python และจากการที่ได้สอบถามข้อมูลจากโรงไฟฟ้าพลังงานแสงอาทิตย์ของการไฟฟ้าฝ่ายผลิตแห่งประเทศไทย อ. ทับสะแก จ. ประจวบคีรีขันธ์ ได้ทราบว่าปัจจุบันระบบตรวจจับความผิดพลาดของโรงไฟฟ้าสามารถแจ้งเตือนได้แบบทันทีทันใดว่าเกิดความผิดพลาดและเกิดที่สตริงไหน แต่ยังไม่สามารถระบุได้ทันทีว่าเป็นความผิดพลาดประเภทใด อาจต้องใช้การวิเคราะห์จากค่าพารามิเตอร์ทางไฟฟ้าที่แสดงอยู่ หรือไปตรวจเช็คที่จุดเกิดความผิดพลาด โดยโมเดลที่ได้ศึกษาและพัฒนาในวิทยานิพนธ์นี้สามารถตรวจจับและระบุได้ทันทีว่าเป็นความผิดพลาดประเภทใดและเกิดที่สตริงใด ซึ่งจะช่วยประหยัดเวลาในการระบุวิธีแก้ไขในแต่ละประเภทของความผิดพลาด และช่วยประหยัดทรัพยากรบุคคลในการปฏิบัติงาน เช่น การไปตรวจเช็คประเภทความผิดพลาดที่จุดเกิดเหตุ อาจต้องใช้เวลา 2 คนขึ้นไปตามมาตรการความปลอดภัย

5.2 ข้อเสนอแนะงานวิจัยในอนาคต

1. ในขั้นตอนการเก็บรวบรวมข้อมูล หากใช้ข้อมูลการเกิดความผิดพลาดที่จำลองขึ้นในระบบจริง ควรพยายามเก็บข้อมูลความเข้มแสงอาทิตย์ให้ครอบคลุมมากที่สุด โดยเฉพาะในสภาวะปกติ การบังเงา

บางส่วน และการเสื่อมสภาพของอุปกรณ์ในระดับสตรีง เนื่องจากลักษณะของพารามิเตอร์ทางไฟฟ้า เช่น กระแสไฟฟ้าและแรงดันไฟฟ้า ของ 3 เหตุการณ์ดังกล่าวมีความคล้ายคลึงกัน ดังนั้น หากไม่เก็บข้อมูลความเข้มแสงอาทิตย์ให้ครอบคลุมตรงกัน จะทำให้เกิดความผิดพลาดในการ จำแนกประเภทระหว่าง 3 เหตุการณ์ดังกล่าวค่อนข้างสูง

2. ต่อยอดและปรับปรุงงานวิจัยได้โดยเพิ่มจำนวนเหตุการณ์ความผิดพลาดเข้าไปและพิจารณาความผิดพลาดบนระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์ที่ใหญ่ขึ้น รวมถึงเพิ่มระบบตรวจติดตามระบบผลิตไฟฟ้าพลังงานแสงอาทิตย์เข้าไปเพื่อให้สามารถทดสอบใช้งานโมเดลตรวจจับและระบุประเภทความผิดพลาดในสถานการณ์ใช้งานจริงได้



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บรรณานุกรม

1. สำนักงานนโยบายและแผนพลังงาน, ก., แผนพัฒนากำลังผลิตไฟฟ้าของประเทศไทย พ.ศ. 2561-2580 ฉบับปรับปรุงครั้งที่ 1. 2020.
2. Pillai, D.S. and N. Rajasekar, *A comprehensive review on protection challenges and fault diagnosis in PV systems*. Renewable and Sustainable Energy Reviews, 2018. 91: p. 18-40.
3. Lazzaretti, A.E., et al., *A monitoring system for online fault detection and classification in photovoltaic plants*. Sensors, 2020. 20(17): p. 4688.
4. Garoudja, E., et al., *An enhanced machine learning based approach for failures detection and diagnosis of PV systems*. Energy conversion and management, 2017. 151: p. 496-513.
5. Basnet, B., H. Chun, and J. Bang, *An intelligent fault detection model for fault detection in photovoltaic systems*. Journal of Sensors, 2020. 2020.
6. Hussain, M., et al., *Artificial neural network based photovoltaic fault detection algorithm integrating two bi-directional input parameters*. Renewable Energy, 2020. 155: p. 1272-1292.
7. Ansari, S., H. Samet, and T. Ghanbari, *Fault Location in Solar Farms*. IEEE Systems Journal, 2020.
8. Ghoneim, S.S., A.E. Rashed, and N.I. Elkalashy, *Fault Detection Algorithms for Achieving Service Continuity in Photovoltaic Farms*. INTELLIGENT AUTOMATION AND SOFT COMPUTING, 2021. 30(2): p. 467-479.
9. Kapucu, C. and M. Cubukcu, *A supervised ensemble learning method for fault diagnosis in photovoltaic strings*. Energy, 2021. 227: p. 120463.
10. Eduardo Nieto, A., F. Ruiz, and D. Patiño, *Characterization of electric faults in photovoltaic array systems*. Dyna, 2019. 86(211): p. 54-63.
11. Chen, Z., et al., *Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions*. Energy Conversion and Management, 2019. 198: p. 111793.
12. สงวนสัตย์, ป., *Artificial intelligence with machine learning : AI สร้างได้ด้วยแมชชีนเลิร์น*

- นึ่ง. พิมพ์ครั้งที่ 1 ed. Vol. 1. 2562, นนทบุรี: ไอทีซี พีริเมียร์. 366.
13. Brownlee, J., *Logistic Regression for Machine Learning*. 2020, Machine Learning Mastery.
 14. สุชีวะ, ศ., การวิเคราะห์ถดถอยโลจิสติก: แนวคิดการวิเคราะห์และการแปรความหมาย. วารสารวิธีวิทยาการวิจัย (Journal of Research Methodology: JRM), 2015. 8(1): p. 10-34.
 15. Xu, L. and M.-Y. Chow. *Power distribution systems fault cause identification using logistic regression and artificial neural network*. in *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. 2005. IEEE.
 16. Daroontham, W., เจาะลึก *Random Forest*. 2018.
 17. Brownlee, J., *Boosting and AdaBoost for Machine Learning*. 2020, Machine Learning Mastery.
 18. กิตตินราทร, ช., *Machine Learning* บทที่ 11: *Boosting*. 2020, guopai.github.io.
 19. Jaichuen, S., *AdaBoost Algorithm*. 2019, sirawichjaichuen.medium.com.
 20. Shetty, C., *Gradient Boosting Technique*. 2020, pub.towardsai.net.
 21. Brownlee, J., *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning*. 2020, Machine Learning Mastery.
 22. อัมพวัน, โ., *Data mining-การทำเหมืองข้อมูล*. 2562, Burapha University.
 23. ทินกร ม้าลายทอง, ป.ธ., เพิ่มประสิทธิภาพของ *Machine Learning Model* ด้วย *Hyperparameter Optimization*. 2021, Big Data Thailand.
 24. James Bergstra, Y.B., *Random Search for Hyper-Parameter Optimization*. *Journal of Machine Learning Research*, 2012. 13: p. 281-305.
 25. คุณบัว, ช., *IoT : สถาปัตยกรรมการสื่อสาร Internet of Things*. 2562, ซีไอเอ็มเคชั่น: กรุงเทพมหานคร. p. 272.
 26. Feng, C., Y. Liu, and J. Zhang, *A taxonomical review on recent artificial intelligence applications to PV integration into power grids*. *International Journal of Electrical Power & Energy Systems*, 2021. 132: p. 107176.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	เสฏฐวุฒิ ยิวเที่ยง
วัน เดือน ปี เกิด	6 กรกฎาคม 2538
สถานที่เกิด	ตราง ประเทศไทย
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	122/38 ถ.บ้านหนองยวน ต.ทับเที่ยง อ.เมือง จ.ตรัง 92000



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY