

การออกแบบและพัฒนาไมโครอัจฉริยะแบบสามารถแยกโหนดในขณะใช้งานสำหรับที่อยู่อาศัย

รัฐกรณ์ ประเสริฐลักษณ์

TNI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

บัณฑิตวิทยาลัย สาขาเทคโนโลยีวิศวกรรม

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2564

A DESIGN AND DEVELOPMENT OF SMART METER LOAD PROFILE FOR RESIDENCES

Ruttgartorn Prasertlux



A Thesis Submitted in Partial Fulfillment of Requirements
for the Degree of Master of Engineering Program in Engineering Technology
Graduate School

Thai-Nichi Institute of Technology

Academic Year 2021

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาไมโครอัจฉริยะแบบสามารถแยกโพลด
	ในขณะใช้งานสำหรับที่อยู่อาศัย
โดย	รัฐกรณ์ ประเสริฐลักษณ์
สาขาวิชา	เทคโนโลยีวิศวกรรม
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.ไพศาล สดวิสัย

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)
วันที่เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.กฤษณ์ชนม์ ภูมิภิตติพิชญ์)

..... กรรมการ
(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)
..... กรรมการ
(ดร.ดอน แก้วดก)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ดร.ไพศาล สดวิสัย)

รัฐกรณ์ ประเสริฐลักษณ์: การออกแบบและพัฒนามิเตอร์อัจฉริยะแบบสามารถแยกโหลด
 ในขณะที่ใช้งานสำหรับที่อยู่อาศัย. อาจารย์ที่ปรึกษา : ผู้ตร.ไพศาล สุตวิสัย, 75 หน้า.

ความต้องการพลังงานไฟฟ้ามีแนวโน้มที่สูงขึ้นในทุกปี จากทั้งการขยายตัวทางเศรษฐกิจ และแนวโน้มที่อุปกรณ์สมัยใหม่จะหันมาใช้พลังงานไฟฟ้า เช่น รถยนต์ไฟฟ้า เป็นต้น เพื่อให้มีพลังงานไฟฟ้าใช้อย่างเพียงพอ การอนุรักษ์พลังงานจึงเป็นเรื่องสำคัญ แนวทางหนึ่งที่มีประสิทธิภาพก็คือ การตรวจวัดพลังงานในระดับเครื่องใช้ไฟฟ้าแต่ละเครื่อง เพราะข้อมูลดังกล่าวสามารถนำไปบริหารการใช้พลังงานทั้งในมุมของผู้ผลิตและผู้ใช้ วิธีการที่นิยมวิธีหนึ่งในการได้มาซึ่งข้อมูลนี้คือการติดตั้งมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะที่ใช้งานสำหรับที่อยู่อาศัย ในงานวิจัยนี้ได้มุ่งเน้นพัฒนามิเตอร์อัจฉริยะโดยอาศัยพื้นฐานจากระบบตรวจรู้สภาวะโหลดทางไฟฟ้าแบบไม่ล่วงล้ำชนิดมีผู้สอน โดยทำการพัฒนาระบบและทดสอบด้วยข้อมูลที่เก็บจากเครื่องใช้ไฟฟ้าจำนวน 10 เครื่อง ซึ่งมีบางเครื่องที่มีขนาดกำลังไฟฟ้าเท่ากัน เวลาที่ใช้ในการสอนอัลกอริทึมตรวจรู้สภาวะโหลดทางไฟฟ้าแบบไม่ล่วงล้ำในงานวิจัยนี้จะน้อยกว่าเวลาที่ใช้สอนอัลกอริทึมตรวจรู้สภาวะโหลดทางไฟฟ้าแบบการเรียนรู้เชิงลึก มิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะที่ใช้งานสำหรับที่อยู่อาศัยได้ถูกทดสอบค่าความเที่ยงตรงได้ผล 1.0 สำหรับเครื่องใช้ไฟฟ้าจำนวน 10 เครื่อง ซึ่งมีบางเครื่องที่มีขนาดกำลังไฟฟ้าเท่ากัน โดยประสิทธิภาพของมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะที่ใช้งานสำหรับที่อยู่อาศัยในงานวิจัยนี้ได้ถูกทดสอบบนสมมุติฐานที่ว่าอุปกรณ์ไฟฟ้าแต่ละตัวต้องไม่ถูกเปิดขึ้นในเวลาเดียวกัน จึงคาดหวังว่าผลลัพธ์ในงานวิจัยงานนี้จะสามารถผลักดันให้เกิดการพัฒนาแบบโครงข่ายไฟฟ้าอัจฉริยะเพื่อให้เกิดการใช้พลังงานที่ยั่งยืนในอนาคต

TNI

บัณฑิตวิทยาลัย

สาขาวิชา เทคโนโลยีวิศวกรรม

ปีการศึกษา 2564

ลายมือชื่อนักศึกษา.....

ลายมือชื่ออาจารย์ที่ปรึกษา.....

RUTTAGORN PRASERTLUX: A DESIGN AND DEVELOPMENT OF SMART METER
LOAD PROFILE FOR RESIDENCES. ADVISOR: DR. PHAISARN SUDWILAI, 75 PP.

The electricity consumption rate is increased every year. This is occurred from increasing of economic size and number of new trend electric equipment such as electric vehicle. To sustain electricity and reduce environment impact in the future, electric energy conservation is important. One way to achieve this topic is non-intrusived load monitoring that can show load profile of each electric equipment from total electricity consumption load. The electricity information of each equipment can help to manage electrical system supply and demand in view of electric supplier and user. In this research, supervised non-intrusived load monitoring algorithm was developed. This algorithm was tested through data that collected from 10 equipment's which some equipment had same power level. The time that used in training algorithm is less than NILM that based on deep learning .Also the smart meter load profile was developed and results were evaluated by accuracy of prediction load profile for monitoring of 10 equipment's. The accuracy score of smart meter system was 1.0 and the system can separated load profile of equipment that have same power level. The performance of smart meter load profile in this research is evaluate base on assumption that none of electrical equipment is ON at simultaneous time. The result of this research was improve smart grid technology that support to sustain electricity in the future.

Graduate School

Student's Signature.....

Field of Engineering of Technology

Advisor's Signature

Academic Year 2021

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จสมบูรณ์ได้จากความอนุเคราะห์จาก ดร.ไพศาล สุทวีชัย อาจารย์ที่ปรึกษาในการทำวิทยานิพนธ์ ที่ได้กรุณาให้ใช้พื้นที่และอุปกรณ์ในห้องวิจัยเชิงพลวัตเพื่อวิชาการทางไฟฟ้าและแม่เหล็กชั้นสูง เพื่อดำเนินการวิจัยจนสำเร็จลุล่วง อีกทั้งได้ให้ความช่วยเหลือในการให้คำแนะนำที่มีประโยชน์ในการนำมาปรับใช้ในวิทยานิพนธ์เล่มนี้ ตลอดจนการตรวจสอบข้อบกพร่องและแนะนำแก้ไขให้มีความสมบูรณ์มากยิ่งขึ้น อีกทั้งคณาจารย์ผู้เป็นกรรมการในการพิจารณาวิทยานิพนธ์ซึ่งประกอบด้วย รองศาสตราจารย์ ดร.กฤษณ์ชนม์ ภูมิกิตติพิชญ์ รองศาสตราจารย์ ดร.วรการ ศรีเชวงทรัพย์ ดร.ดอน แก้วดก และ ดร.ชาตรี บุชบาธ ที่ให้ข้อคิด และเสนอแนะข้อมูลที่เป็นประโยชน์ ทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น และขอขอบคุณอย่างยิ่งสำหรับครอบครัวที่คอยให้กำลังใจและการสนับสนุนที่ดีเสมอมา

ประโยชน์อันใดก็ตาม ที่เกี่ยวเนื่องมาจากวิทยานิพนธ์นี้ เป็นผลจากการช่วยเหลือและความกรุณาจากท่านข้างต้น ข้าพเจ้าจะระลึกอยู่เสมอ และขอกราบขอบพระคุณทุกท่านมา ณ โอกาสนี้

รัฐกรณ์ ประเสริฐลักษณ์

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	2
1.3 ขอบเขตของการศึกษา.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 แผนงานและระยะเวลาการดำเนินงาน.....	4
2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	5
2.1 ทฤษฎีโครงข่ายประสาทเทียมแบบ Multi-layers perceptron	7
2.2 การสื่อสารด้วยมาตรฐาน Modbus.....	13
2.3 การวัดประสิทธิภาพด้วย Confusion matrix.....	18
2.4 งานวิจัยที่เกี่ยวข้อง.....	20
3 ระเบียบวิธีการวิจัย.....	40
3.1 กรอบแนวคิดการวิจัย.....	40
3.2 ขั้นตอนการทดลอง.....	42
3.3 การดำเนินการวิจัยและเก็บข้อมูล.....	42
3.4 การทำการทดลอง.....	42

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการวิจัย.....	44
4.1 ส่วนประกอบฮาร์ดแวร์ของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลด ในขณะใช้งานสำหรับที่อยู่อาศัย.....	44
4.2 อัลกอริธึมสำหรับจำแนกโหลดในขณะใช้งาน.....	47
4.3 ส่วนโปรแกรมของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้ งานขณะใช้งานสำหรับที่อยู่อาศัย.....	49
4.4 การทดสอบและสรุปผลระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะ ในขณะใช้งานสำหรับที่อยู่อาศัย.....	49
5 สรุปผลการวิจัย.....	52
5.1 บทสรุป.....	52
5.2 ข้อเสนอแนะ.....	54
บรรณานุกรม.....	55
ภาคผนวก.....	58
ภาคผนวก ก. Meter and computer interface code (Arduino IDE).....	59
ภาคผนวก ข. Disaggregated load learning algorithm (Python).....	64
ภาคผนวก ค. Smart meter programing (Python).....	67
ประวัติย่อผู้วิจัย.....	75

สารบัญตาราง

ตาราง	หน้า
1.1 แผนงานและระยะเวลาในการดำเนินการศึกษาวิจัย.....	4
2.1 การประเมินประสิทธิภาพการแยกชนิดโหลดของอัลกอริทึมทั้ง 3 ชนิด.....	26
2.2 ผลการทดสอบในการจำแนกโหลดเครื่องใช้ไฟฟ้าทั้ง 3 ชนิด.....	27
2.3 ผลการทดสอบของอัลกอริทึมที่นำเสนอเทียบกับอัลกอริทึมที่ได้พัฒนาก่อน.....	28
2.4 ชุดข้อมูลอุปกรณ์ไฟฟ้าที่ใช้ในการสอนและการทดสอบระบบ.....	29
2.5 ผลการทดสอบโมเดลการแยกปริมาณโหลดไฟฟ้าเทียบกับโมเดลก่อนหน้า.....	30
2.6 ผลการทดสอบโมเดลการแยกปริมาณโหลดไฟฟ้าเทียบกับโมเดลก่อนหน้า.....	35
2.7 สรุปผลงานวิจัยที่เกี่ยวข้อง	36
4.1 รายการอุปกรณ์ไฟฟ้าที่จะนำมาใช้ในการพัฒนาอัลกอริทึมสำหรับจำแนกโหลด ในขณะใช้งาน.....	48
5.1 เปรียบเทียบระบบมิตเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานที่มีอยู่ใน ปัจจุบัน.....	54

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญรูป

รูป		หน้า
1.1	ความต้องการพลังงานไฟฟ้าที่เพิ่มขึ้นตามผลิตภัณฑ์มวลรวมของประเทศที่เพิ่มขึ้น.....	1
2.1	องค์ประกอบของมิเตอร์อัจฉริยะ.....	5
2.2	จำนวนการตีพิมพ์งานวิจัยการประยุกต์ใช้หลักการวิเคราะห์ข้อมูลกับมิเตอร์อัจฉริยะ...	6
2.3	ภาพรวมการเรียนรู้ของเครื่องจักร.....	8
2.4	สถาปัตยกรรมหรือโครงสร้างของโครงข่ายประสาทเทียมแบบ Multi layers perceptron.....	9
2.5	ลักษณะของโครงข่ายประสาทเทียมแบบ Multi layers perceptron.....	9
2.6	ความหมายของตัวแปรต่างๆที่เกี่ยวข้องกับการเรียนรู้ของโครงข่ายประสาทเทียม.....	10
2.7	การสื่อสารแบบอนุกรมด้วย RS-485 สำหรับ Modbus RTU.....	14
2.8	ชุดข้อมูลสำหรับการสื่อสาร Modbus RTU.....	15
2.9	รายละเอียดชุดข้อมูล Function Code.....	15
2.10	ตำแหน่งแอดเดรสใน Modbus RTU โดยแบ่งตามรูปแบบการทำงาน.....	16
2.11	ชุดคำสั่งสำหรับการอ่าน (Read Command).....	16
2.12	ชุดคำสั่งสำหรับการเขียน (Write Command).....	16
2.13	การสื่อสารแบบอีเทอร์เน็ตสำหรับ Modbus TCP.....	17
2.14	ส่วนประกอบชุดข้อมูลของ Modbus TCP เทียบกับ Modbus RTU.....	18
2.15	รายละเอียดของแต่ละ Field ในหนึ่งเฟรมของ Modbus TCP.....	18
2.16	เมตริกวัดประสิทธิภาพ (Confusion Matrix).....	19
2.17	โครงสร้างของระบบตรวจสอบรู้สภาวะโหลดแบบไม่รุกราน.....	20
2.18	การติดตั้งของระบบตรวจสอบรู้สภาวะโหลดแบบไม่รุกราน.....	21
2.19	โครงสร้างของระบบตรวจสอบรู้สภาวะโหลดแบบไม่รุกราน.....	22
2.20	อัลกอริทึมการแยกการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละเครื่อง.....	23
2.21	ผลการทดลองด้วยชุดข้อมูล BLUED.....	25
2.22	a) ระบบจำแนกชนิดของโหลดไฟฟ้า b) การติดตั้งระบบเข้ากับระบบไฟฟ้า.....	27
2.23	โครงสร้างของอัลกอริทึม Convolution Neural Network.....	28
2.24	โครงสร้างของอัลกอริทึม Convolution Neural Network.....	30

สารบัญรูป (ต่อ)

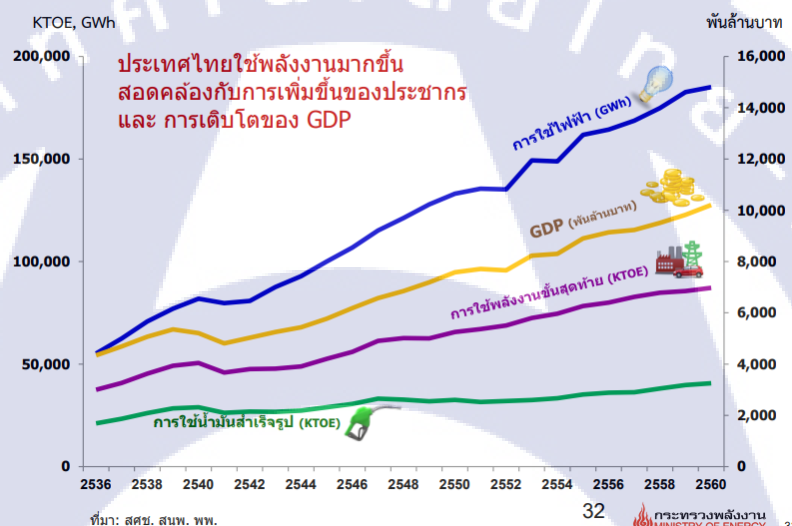
รูป	หน้า
2.25 ผลการทดสอบการแยกโหลดไฟฟ้าโหลดไฟฟ้ารวมที่อยู่อาศัยที่รู้ปริมาณและชนิดโหลดที่อยู่ภายใน.....	32
2.26 ผลการทดสอบการแยกโหลดไฟฟ้าโหลดไฟฟ้ารวมที่อยู่อาศัยที่ไม่รู้ปริมาณและชนิดโหลดที่อยู่ภายใน.....	33
2.27 โครงสร้างของอัลกอริทึม Unet – NILM.....	34
3.1 กรอบแนวความคิดงานวิจัยการออกแบบและพัฒนาไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย.....	40
3.2 ตำแหน่งติดตั้งไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน.....	43
3.3 แผนผังระบบไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย....	43
4.1 ไมเตอร์ไฟฟ้าแบบดิจิตอลยี่ห้อ Schneider รุ่น EasyLogic PM2230.....	44
4.2 หม้อแปลงกระแสขนาด 50/5A ยี่ห้อ QUBIX รุ่น MSQ-30.....	45
4.3 วงจรการต่อระบบไมเตอร์ไฟฟ้าแบบดิจิตอล.....	45
4.4 บอร์ดแปลงสัญญาณ RS-485 เป็น TTL(Serial).....	46
4.5 บอร์ด Arduino รุ่น Uno.....	46
4.6 วงจรของชุดส่งข้อมูล.....	46
4.7 อุปกรณ์ Power quality analyzer ยี่ห้อ Kyoritsu.....	47
4.8 ผลการทดสอบอัลกอริทึมสำหรับจำแนกโหลดในขณะใช้งาน.....	48
4.9 ลักษณะหน้าจอแสดงผล.....	49
4.10 ชุดทดสอบระบบไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน.....	50
4.11 ผลการทดสอบความแม่นยำของระบบไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน.....	51
5.1 แผนภาพกล่องของค่ากำลังไฟฟ้าจริง (P) ของโหลดที่ใช้จริงทั้ง 10 อุปกรณ์.....	52
5.2 แผนภาพกล่องของค่ากำลังไฟฟ้าปรากฏ (S) ของโหลดที่ใช้จริงทั้ง 10 อุปกรณ์.....	53

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

พลังงานไฟฟ้าถือเป็นพลังงานรูปแบบหนึ่งที่มีการใช้งานอย่างแพร่หลายมากที่สุดเมื่อเทียบกับพลังงานประเภทอื่น จากแนวโน้มในปัจจุบันที่ความต้องการพลังงานไฟฟ้าของประเทศไทยเพิ่มสูงขึ้นในทุกปี ตามแนวโน้มการเติบโตทางเศรษฐกิจดังแสดงด้วยผลิตภัณฑ์มวลรวมของประเทศไทยที่สูงขึ้นทุกปี แสดงได้จากข้อมูลของกระทรวงพลังงานในปี 2562[1] ดังภาพที่ 1.1



รูปที่ 1.1 ความต้องการพลังงานไฟฟ้าที่เพิ่มขึ้นตามผลิตภัณฑ์มวลรวมของประเทศไทยที่เพิ่มขึ้น

การผลิตไฟฟ้าในประเทศไทยปัจจุบัน ยังพึ่งพิงการใช้เชื้อเพลิงฟอสซิลเป็นหลัก การเพิ่มขึ้นของการใช้พลังงานไฟฟ้า จึงส่งผลต่อการเพิ่มปัญหาด้านสิ่งแวดล้อมของประเทศไทย หน่วยงานภาครัฐจึงได้จัดทำแผนแม่บทการพัฒนาระบบโครงข่ายสมาร์ททกริดของประเทศไทย ปี พ.ศ. 2558 – 2579 เพื่อส่งผลให้เกิดบูรณาการด้าน การจัดหาไฟฟ้าและการใช้ไฟฟ้าได้อย่างเพียงพอ มีประสิทธิภาพ ยั่งยืน มีคุณภาพบริการที่ดี และเกิดประโยชน์สูงสุดต่อประเทศ ในการที่จะพัฒนาระบบโครงข่ายสมาร์ททกริดให้เกิดขึ้นได้นั้นมิเตอร์อัจฉริยะหรือสมาร์ทมิเตอร์ถือเป็นปัจจัยสำคัญ มิเตอร์อัจฉริยะคือ มิเตอร์วัดพลังงานไฟฟ้าชนิดใหม่ที่ถูกพัฒนามาเพื่อแทนที่มิเตอร์ ไฟฟ้าแบบเดิมที่มีอยู่ (แบบจานหมุน) โดยการอ่านค่าพลังงานไฟฟ้าที่ถูกอ่านด้วยวงจรอิเล็กทรอนิกส์และส่งข้อมูลในรูปแบบดิจิทัลไปที่ระบบควบคุม (Control System) โดยมิเตอร์อัจฉริยะมีข้อดีที่เหนือกว่ามิเตอร์

แบบเดิมหลายข้อเช่น สามารถอ่านค่าพลังงานไฟฟ้าได้โดยอัตโนมัติไม่ต้องส่งพนักงานไปจดบันทึก, สามารถแจ้งเหตุผิดปกติทางไฟฟ้า เป็นต้น จากข้อมูลดังกล่าวข้างต้นผู้วิจัยเกิดแรงบันดาลใจที่จะพัฒนามิเตอร์อัจฉริยะให้สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย เพื่อให้สามารถรู้ได้ว่าอุปกรณ์ไฟฟ้าไหนภายใต้มิเตอร์อัจฉริยะนั้นทำงานอยู่ ส่งผลให้สามารถเฝ้าติดตามสถานะการทำงานของอุปกรณ์ไฟฟ้าแต่ละตัวว่ามีการทำงานที่ผิดปกติหรือไม่ รวมทั้งยังสามารถเพิ่มประสิทธิภาพในการจัดการพลังงานได้อีกด้วย ในการพัฒนามิเตอร์อัจฉริยะให้สามารถแยกโหลดในขณะใช้งานนั้น ทางผู้วิจัยได้ทำการพัฒนาโปรแกรมโดยใช้หลักทฤษฎีการเรียนรู้ของเครื่อง (Machine learning) โดยการเรียนรู้ของเครื่องนั้นแบ่งเป็น 2 วิธีคือ

1.การเรียนรู้แบบมีผู้สอน (Supervised learning) คือ มีการนำข้อมูลมาสอนให้กับโปรแกรมจนสามารถจำแนกข้อมูลได้ถูกต้องตามกำหนด ก่อนนำโปรแกรมไปใช้จำแนกข้อมูลจริง

2.การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) คือ การให้โปรแกรมจำแนกข้อมูลจริง โดยไม่มีการนำข้อมูลมาสอนให้กับโปรแกรมก่อน

ในการวิจัยนี้ได้เลือกใช้วิธีการเรียนรู้แบบมีผู้สอน โดยผู้วิจัยได้ทำการเก็บบันทึกค่าคุณลักษณะทางไฟฟ้าในส่วนของกำลังไฟฟ้าจริง (Real power: P) และกำลังไฟฟ้าปรากฏ (Apparent power: S) ของอุปกรณ์ไฟฟ้าจำนวน 10 รายการ จากนั้นจึงทำการสอนให้กับอัลกอริธึมการเรียนรู้ของเครื่องที่พัฒนาด้วยโปรแกรม Python โดยแสดงขั้นตอนการเรียนรู้ของเครื่องดังนี้

-การเก็บรวบรวมข้อมูล (Data acquisition) คือ ทำการวัดคุณลักษณะทางไฟฟ้าของเครื่องไฟฟ้าด้วยอุปกรณ์บันทึกข้อมูล (Data logger)

-การสร้างอัลกอริธึมการเรียนรู้ของเครื่องและการสอนด้วยข้อมูลที่เก็บรวบรวม (Machine learning algorithm)

จากนั้นจึงทำการประเมินความถูกต้องของโปรแกรมและนำโปรแกรมไปใช้ร่วมกับมิเตอร์อัจฉริยะ เพื่อให้มิเตอร์สามารถแยกโหลดเครื่องใช้ไฟฟ้าภายในบ้านในขณะใช้งานได้ โดยผู้วิจัยหวังว่างานวิจัยนี้จะเป็นส่วนหนึ่งในการช่วยผลักดันให้เกิดการพัฒนาโครงข่ายสมาร์ทกริด เป็นผลให้เกิดการใช้พลังงานที่มีประสิทธิภาพและนำไปสู่ประโยชน์ของประเทศต่อไปในอนาคต

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาและพัฒนามิเตอร์ไฟฟ้าอัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

1.2.2 เพื่อศึกษาและพัฒนาอัลกอริธึมการเรียนรู้ของเครื่องจักรแบบ Multi-layers Perceptron Neural Network (PNN) ที่ใช้ในการแยกโหลดสำหรับมิเตอร์อัจฉริยะ

1.2.3 เพื่อส่งเสริมให้เกิดการพัฒนาวิจัยในงานด้านโครงข่ายสมาร์ทกริด

1.3. ขอบเขตของการศึกษา

1.3.1 ศึกษาและพัฒนาไมเตอร์ไฟฟ้าอัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยด้วยวิธี Multi-layers Perceptron Neural Network (PNN)

1.3.2 ทดสอบความถูกต้องในการแยกโหลดของไมเตอร์ไฟฟ้าอัจฉริยะได้ถูกต้อง 100%

1.4. ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและระบุปัญหาที่เกี่ยวข้อง

1.4.2 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.4.3 เก็บรวบรวมข้อมูลโหลดเครื่องใช้ไฟฟ้าที่กำหนดจำนวน 10 อุปกรณ์

1.4.4 ออกแบบพัฒนาโปรแกรมในการแยกโหลดในขณะใช้งานสำหรับไมเตอร์อัจฉริยะ

1.4.5 ออกแบบวิธีการทดสอบ ตัวชี้วัดและเกณฑ์

1.4.6 จัดทำรายงานวิจัย บทที่ 1-3

1.4.7 พัฒนาไมเตอร์อัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

1.4.8 ทำการทดสอบและเก็บผลการทดสอบ

1.4.9 วิเคราะห์ข้อมูลและทำการสรุปผล

1.4.10 จัดทำรายงานวิจัย บทที่ 4-5

1.4.11 จัดทำเล่มวิทยานิพนธ์

1.4.12 จัดทำโปสเตอร์

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ผลการศึกษาระบบไมเตอร์อัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

1.5.2 ระบบไมเตอร์อัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

ตารางที่ 1.1 แผนงานและระยะเวลาในการดำเนินการศึกษาวิจัย

ขั้นตอนและวิธีการดำเนินการ		ปี พ.ศ. 2564								ปี พ.ศ. 2565			
		5	6	7	8	9	10	11	12	1	2	3	4
1	ศึกษาและระบุปัญหาที่เกี่ยวข้อง												
2	ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง												
3	เก็บรวบรวมข้อมูลโหลดเครื่องใช้ไฟฟ้าที่กำหนดจำนวน 10 อุปกรณ์												
4	ออกแบบพัฒนาโปรแกรมในการแยกโหลดในขณะใช้งานสำหรับมอเตอร์อัจฉริยะ												
5	ออกแบบวิธีการทดสอบ ตัวชี้วัดและเกณฑ์												
6	จัดทำรายงานวิจัย บทที่ 1-3												
7	พัฒนามอเตอร์อัจฉริยะที่สามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย												
8	ทำการทดสอบและเก็บผลการทดสอบ												
9	วิเคราะห์ข้อมูลและทำการสรุปผล												
10	จัดทำรายงานวิจัย บทที่ 4-5												
11	จัดทำเล่มวิทยานิพนธ์												
12	จัดทำโปสเตอร์												

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในปัจจุบันมิเตอร์อัจฉริยะมีการใช้งานกันอย่างแพร่หลายทั่วประเทศ ตั้งแต่เริ่มมีการใช้ครั้งแรกในปี พ.ศ. 2543 มิเตอร์อัจฉริยะถือเป็นอุปกรณ์สำคัญในการพัฒนาโครงข่ายไฟฟ้าสามารถกริดเพื่อรองรับการผลิตพลังงานไฟฟ้าจากแหล่งพลังงานหมุนเวียน รวมทั้งการใช้พลังงานไฟฟ้าที่มากขึ้น เช่นการรองรับยานยนต์ไฟฟ้า เป็นต้น มิเตอร์อัจฉริยะได้ถูกพัฒนาวิจัยในองค์ประกอบต่างๆ โดยสรุป 3 องค์ประกอบคือ

-การตรวจวัดและการจัดเก็บข้อมูลของมิเตอร์อัจฉริยะ

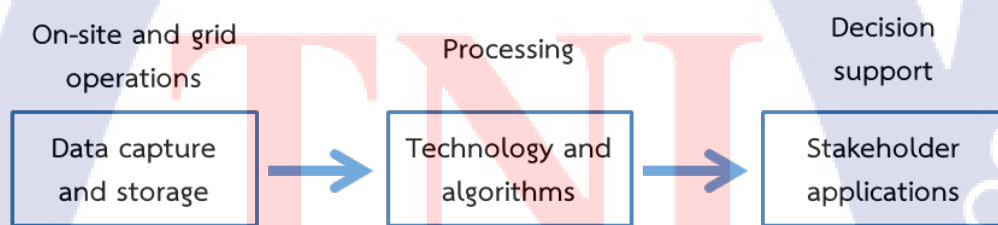
โดยในส่วนนี้จะเน้นการพัฒนาตัวอุปกรณ์ให้สามารถเก็บข้อมูลได้ตามเวลาจริงอย่างต่อเนื่อง และระบบในการส่งข้อมูลจากอุปกรณ์มายังหน่วยประมวลผลกลาง

-ระบบมิเตอร์อัจฉริยะ

โดยในส่วนนี้จะเน้นพัฒนาไปยังการวางรูปแบบของระบบในการส่งข้อมูลระหว่างตัวมิเตอร์อัจฉริยะกับส่วนประมวลผลกลาง รวมทั้งการส่งข้อมูลจากการประมวลผลมายังหน่วยแสดงผลในรูปแบบต่างๆ

-การนำข้อมูลมาพัฒนาเพื่อประยุกต์ใช้งานสำหรับผู้เกี่ยวข้อง

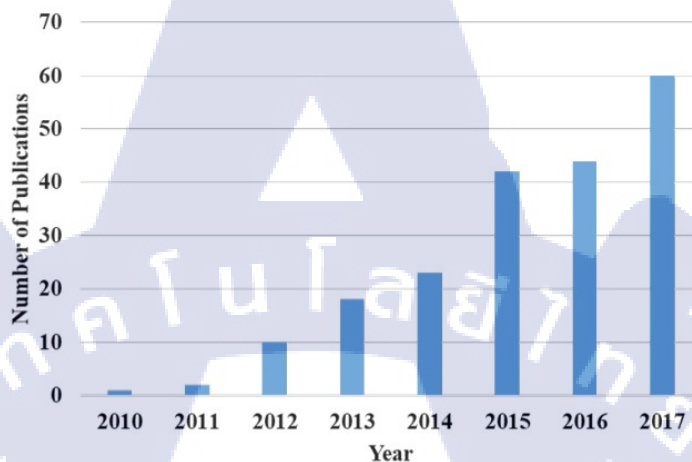
โดยในส่วนนี้จะเน้นพัฒนาไปยังการนำข้อมูลมาวิเคราะห์ (Data analytic) โดยการใช้การเรียนรู้ของเครื่อง (Machine learning) เพื่อนำผลที่วิเคราะห์ได้ไปประยุกต์ใช้งานในด้านต่างๆ โดยสามารถแสดงองค์ประกอบของมิเตอร์อัจฉริยะได้ดังรูปที่ 2.1



รูปที่ 2.1 องค์ประกอบของมิเตอร์อัจฉริยะ[1]

ในการนำแนวความคิดเกี่ยวกับทฤษฎีการวิเคราะห์ข้อมูล (Data analytic) มาใช้ในงานวิจัยมิเตอร์อัจฉริยะนั้น ได้พบว่าก่อนช่วงปี พ.ศ. 2554 มีการตีพิมพ์งานวิจัยด้านนี้ในอัตราที่ต่ำ แต่หลังจากปี พ.ศ. 2555 ถึงปี พ.ศ. 2560 กลับมีงานตีพิมพ์ที่เกี่ยวกับการประยุกต์ใช้หลักการ

วิเคราะห์ข้อมูลกับมิเตอร์อัจฉริยะสูงถึง 200 งานวิจัย โดยในปี พ.ศ. 2560 ปีเดียวมีการตีพิมพ์งานวิจัยด้านนี้ถึง 60 งานวิจัย ดังแสดงในรูปที่ 2.2 [2]



รูปที่ 2.2 จำนวนการตีพิมพ์งานวิจัยการประยุกต์ใช้หลักการวิเคราะห์ข้อมูลกับมิเตอร์อัจฉริยะ[2]

ที่เป็นเช่นนี้เนื่องจากมิเตอร์อัจฉริยะเริ่มต้นใช้งานในปี พ.ศ. 2543 จึงต้องใช้ระยะเวลาในการเก็บข้อมูลให้เพียงพอต่อการนำมาวิเคราะห์ การประยุกต์ใช้หลักการวิเคราะห์ข้อมูลกับมิเตอร์อัจฉริยะสามารถแบ่งออกเป็น 3 ด้านด้วยกันคือ

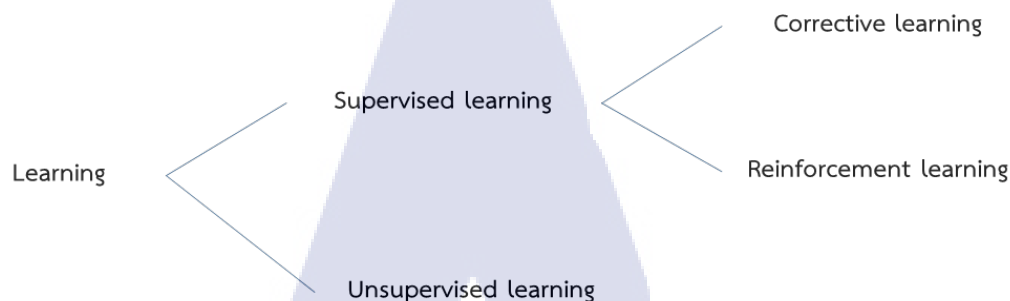
- การวิเคราะห์โหลดไฟฟ้า (Load Analysis) ในงานวิจัยด้านนี้จะเน้นไปยังการใช้การเรียนรู้ของเครื่องจักรในการวิเคราะห์หาความผิดปกติของข้อมูล ทั้งในรูปแบบที่ไม่มีแบบแผนซึ่งอาจเกิดจากข้อมูลที่ไม่มีความสมบูรณ์ (Bad Data Detection) และในรูปแบบที่มีแบบแผนซึ่งเกิดจากการโจรกรรมระบบไฟฟ้า (Energy Theft Detection) อีกส่วนงานวิจัยหนึ่งที่เกี่ยวข้องกับงานด้านนี้คือ การวิเคราะห์และแยกแยะข้อมูลของโหลด โดยใช้การเรียนรู้ของเครื่องจักรในการแยกแยะข้อมูลโหลดตามคุณลักษณะ การลดมิติของข้อมูล รวมไปถึงการวิเคราะห์ความแปรปรวนของข้อมูล
- การพยากรณ์โหลดไฟฟ้า (Load Forecasting) งานวิจัยด้านนี้จะเน้นการประยุกต์ใช้การเรียนรู้ของเครื่องจักร เพื่อพยากรณ์โหลดภายใต้เงื่อนไขแวดล้อมที่หลากหลาย
- การบริหารจัดการโหลดไฟฟ้า (Load Management) ในงานวิจัยด้านนี้ได้นำข้อมูลของโหลดไฟฟ้ามาวิเคราะห์โดยวิธีการเรียนรู้ของเครื่องจักรร่วมกับปัจจัยทางด้านสังคมวิทยา เพื่อหาแบบจำลองปัจจัยที่มีผลต่อการเปลี่ยนแปลงของโหลด รวมทั้งการกำหนดกลยุทธ์ เช่นการกำหนด

ราคาค่าพลังงานในแต่ละช่วง และทำการวิเคราะห์ผลของปัจจัยที่มีต่อการเปลี่ยนแปลงของโหลดไฟฟ้า เพื่อให้สามารถบริหารจัดการระบบไฟฟ้าได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

การจัดการบริหารระบบไฟฟ้าได้อย่างมีประสิทธิภาพนั้น ข้อมูลของโหลดไฟฟ้าในแง่มุมที่หลากหลายมีความจำเป็นต่อการวิเคราะห์และกำหนดกลยุทธ์ดำเนินการ หนึ่งในข้อมูลที่สำคัญ คือ การรู้สถานะ การทำงานของอุปกรณ์ไฟฟ้าแต่ละอุปกรณ์ ในการที่จะสถานะการทำงานของอุปกรณ์ไฟฟ้าได้นั้นสามารถดำเนินการได้ 2 แนวทาง โดยแนวทางแรก คือ การติดตั้งมิเตอร์เพื่อตรวจจับการทำงานให้กับอุปกรณ์ไฟฟ้าแต่ละอุปกรณ์ที่เรียกว่า การตรวจจับโหลดแบบรุกล้ำ (Intrusive Load Monitoring) และแนวทางที่สอง คือ การนำสัญญาณที่ได้จากมิเตอร์หลักมาทำการวิเคราะห์โดยวิธีการเรียนรู้ของเครื่องเพื่อสามารถบอกสถานะการทำงานของอุปกรณ์ไฟฟ้าแต่ละตัวได้ ซึ่งเรียกวิธีนี้ว่า การตรวจจับโหลดแบบไม่รุกล้ำ (Non-Intrusive Load Monitoring) สำหรับอัลกอริทึมที่ใช้ในการแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยในงานวิจัยนี้ จะมีทฤษฎีพื้นฐานที่เกี่ยวข้องในการออกแบบพัฒนา 3 ทฤษฎี ดังต่อไปนี้

2.1 ทฤษฎีโครงข่ายประสาทเทียมแบบ Multi-layers perceptron

โครงข่ายประสาทเทียมเป็นระบบคอมพิวเตอร์จากแบบจำลองทางคณิตศาสตร์ เพื่อจำลองการทำงานของโครงข่ายประสาทชีวภาพที่อยู่ในสมองของมนุษย์ เพื่อแก้ปัญหาแบบเดียวกับที่สมองมนุษย์สามารถทำได้ อาทิเช่น คอมพิวเตอร์วิทัศน์, การรู้จำคำพูด, การแปลภาษา, การกรองเนื้อหาโซเชียลมีเดีย, การเล่นเกมและการวินิจฉัยโรค โครงข่ายประสาทเทียมจัดเป็นอัลกอริทึมการเรียนรู้ของเครื่องชนิดหนึ่ง โดยเราสามารถแบ่งอัลกอริทึมการเรียนรู้ของเครื่องจักรได้เป็น 2 แบบ แบบแรกเป็นการเรียนรู้แบบมีผู้สอน (Supervised learning) คือมีการนำข้อมูลและผลลัพธ์ป้อนเข้าในอัลกอริทึมเพื่อทำการสอน และทำการประเมินความแม่นยำของแบบจำลอง เมื่อได้ผลความแม่นยำที่ยอมรับได้ แบบจำลองดังกล่าวก็สามารถใช้งานได้ ในขณะที่อีกแบบหนึ่งเป็นการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) โดยทำการป้อนข้อมูลเข้าไป อัลกอริทึมจะทำการจัดแยกข้อมูลออกเป็นกลุ่มโดยไม่ต้องมีการนำผลลัพธ์มาสอนไว้ก่อน นอกจากนี้อัลกอริทึมแบบการเรียนรู้แบบมีผู้สอนสามารถแบ่งย่อยได้เป็นอีก 2 แบบ คือการเรียนรู้แบบเสริมกำลัง (Reinforcement learning) โดยข้อมูลจะถูกป้อนเข้าไปในอัลกอริทึมและทำการประมวลผลผลลัพธ์ ผู้สอนจะทำการประเมินผลลัพธ์ว่าถูกหรือผิดและทำการส่งผลลัพธ์ดังกล่าวให้อัลกอริทึมเรียนรู้ วิธีการดังนี้ทำให้สามารถสอนอัลกอริทึมโดยใช้ข้อมูลเพียงอย่างเดียว ส่วนอีกแบบคือการเรียนรู้แบบแก้ไข (Corrective learning) โดยจะมีการวัดค่าความคลาดเคลื่อนของอัลกอริทึมและทำการปรับอัลกอริทึมเพื่อลดขนาดความคลาดเคลื่อนลง วิธีการเรียนรู้โครงข่ายประสาทเทียมแบบ Multi-layers perceptron จัดอยู่ในวิธีการเรียนรู้ในแบบดังกล่าว สรุปภาพรวมการเรียนรู้ของเครื่องจักรสามารถแสดงดังรูปที่ 2.3



รูปที่ 2.3 ภาพรวมการเรียนรู้ของเครื่องจักร

สำหรับในการศึกษาระบบโครงข่ายประสาทเทียมแบบ Multi-layers perceptron จะพิจารณาตามหัวข้อดังต่อไปนี้ 2.1.1. สถาปัตยกรรมหรือโครงสร้าง (Architecture), 2.1.2 ลักษณะโครงข่ายประสาทเทียมแบบ Multi-layers perceptron และ 2.1.3 วิธีการเรียนรู้แบบแพร่ย้อนกลับ (Back propagation method)

2.1.1สถาปัตยกรรมหรือโครงสร้าง (Architecture) การพิจารณาสถาปัตยกรรมหรือโครงสร้างของโครงข่ายประสาทเทียมนั้นจะแยกพิจารณาดังนี้

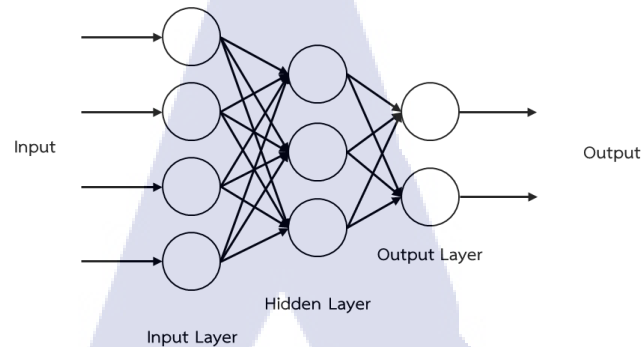
- จำนวนชั้น (Number of layers)

โครงข่ายประสาทเทียมแบบ Multi layers perceptron นั้นจะมีชั้นอินพุต (Input layer) จำนวน 1 ชั้น ชั้นซ่อน (Hidden layer) ต้องมีอย่างน้อย 1 ชั้น และชั้นเอาต์พุตจำนวน 1 ชั้น

- จำนวนโหนด (Numbers of nodes)

ในแต่ละชั้นจะประกอบด้วยโหนด (Nodes) ในชั้นอินพุตจะเรียกว่าโหนดชั้นอินพุต, ในชั้นซ่อนจะเรียกว่าโหนดชั้นซ่อน และชั้นเอาต์พุตจะเรียกว่าโหนดชั้นเอาต์พุต โดยจำนวนโหนดชั้นอินพุตจะเท่ากับตัวแปรชั้นอินพุต ในโหนดชั้นซ่อนไม่ได้มีเกณฑ์ที่แน่นอนในการกำหนดจำนวน ส่วนจำนวนโหนดชั้นเอาต์พุตจะขึ้นกับรูปแบบ (Pattern) ของตัวแปรเอาต์พุต

โดยแสดงสถาปัตยกรรมหรือโครงสร้างของโครงข่ายประสาทเทียมแบบ Multi layers perceptron ดังรูปที่ 2.4

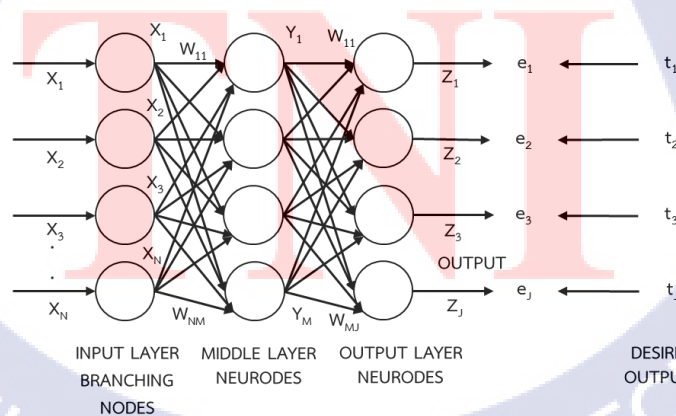


รูปที่ 2.4 สถาปัตยกรรมหรือโครงสร้างของโครงข่ายประสาทเทียมแบบ Multi layers perceptron[3]

2.1.2 ลักษณะโครงข่ายประสาทเทียมแบบ Multi-layers perceptron โครงข่ายประสาทเทียมแบบ Multi-layers perceptron ที่ใช้งานวิจัยนี้ มีหลักการดังนี้

- การเชื่อมต่อระหว่างชั้นต่าง ๆ นั้น จะมีลักษณะที่ทุกโหนดในชั้นอินพุตนั้นจะส่งสัญญาณไปทุกโหนดในชั้นซ่อน และทุกโหนดในชั้นซ่อนจะทำการส่งสัญญาณไปทุกโหนดในชั้นเอาต์พุต

- การทำงานของชั้นต่าง ๆ นั้น ในชั้นอินพุตจะไม่มีผลกระทบใดๆทั้งสิ้น จะทำหน้าที่เพียงแครับสัญญาณหรือข้อมูลเข้าแล้วกระจายออกไปยังแต่ละโหนดในชั้นซ่อน ส่วนชั้นซ่อนและชั้นเอาต์พุตนั้น จะเป็นชั้นที่มีการประมวลผล โดยรูปที่ 2.5 จะเป็นการแสดงถึงลักษณะของโครงข่ายประสาทเทียมแบบ Multi-layers perceptron



รูปที่ 2.5 ลักษณะของโครงข่ายประสาทเทียมแบบ Multi-layers perceptron[3]

2.1.3 วิธีการเรียนรู้แบบแพร่ย้อนกลับ (Back propagation method) [3]

จากโครงข่ายประสาทเทียมแบบ Multi-layers perceptron แสดงดังรูปที่ 2.5 และความหมายของตัวแปลต่างๆดังแสดงในภาพ 2.6 สามารถแบ่งขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียมแบบ Multi-layers perceptron ได้ดังนี้

x_n = อินพุตโหนดที่ n มีทั้งหมด N โหนด
 s_m = เอาต์พุตของชั้นซ่อน ก่อนทำการปรับค่า activation เป็น y_m
 y_m = เอาต์พุตของชั้นซ่อน หลังทำการปรับค่าของโหนดที่ m มีทั้งหมด M โหนด
 v_j = เอาต์พุตของชั้นซ่อน ก่อนทำการปรับค่า activation เป็น z_j
 z_j = ค่าเอาต์พุตที่ได้ทำการปรับค่าแล้วของชั้นเอาต์พุตโหนดที่ j มีทั้งหมด J โหนด
 t_j = ค่าเอาต์พุตที่ต้องการที่ชั้นเอาต์พุตโหนดที่ j มีทั้งหมด J โหนด
 w_{nm} = น้ำหนักของเส้นเชื่อมระหว่างชั้นอินพุต กับชั้นซ่อน
 w_{mj} = น้ำหนักของเส้นเชื่อมระหว่างชั้นซ่อน กับชั้นเอาต์พุต
 η = อัตราการเรียนรู้มีค่าอยู่ระหว่าง 0 ถึง 1
 r = จำนวนรอบที่จะทำการเรียนรู้ มี R เป็นจำนวนรอบที่กำหนด
 q = จำนวนชุดของข้อมูลตัวอย่าง มี Q เป็นตัวกำหนด
 $e^{(q)}$ = ค่าผิดพลาดของข้อมูลตัวอย่าง
 E = ค่าผิดพลาดรวมเฉลี่ยของข้อมูลตัวอย่าง

รูปที่ 2.6 ความหมายของตัวแปลต่างๆที่เกี่ยวข้องการเรียนรู้ของโครงข่ายประสาทเทียม [3]

2.1.3.1 กำหนดจำนวนโหนดในชั้นอินพุต (N), จำนวนโหนดในชั้นเอาต์พุต (J) และจำนวนโหนดในชั้นซ่อน (M) จำนวน 1 ชั้นซ่อน และกำหนดจำนวนข้อมูลอินพุตและข้อมูลเอาต์พุตต่อจากนั้นจะทำการกำหนดจำนวนรอบสูงสุดที่จะทำการเรียนรู้ (R) รวมถึงค่าผิดพลาดที่ยอมรับได้

2.1.3.2 กำหนดค่าพารามิเตอร์ของอัตราการเรียนรู้ (η) ที่อยู่ในช่วง $[0, 1]$

2.1 3.3 การสุ่มน้ำหนักเริ่มต้นให้กับทุกเส้นเชื่อมโยงภายในโครงข่ายประสาทเทียมในทั้ง 2 ชั้น โดยให้มีค่าอยู่ระหว่าง $[-1, 1]$

2.1.3.4 รับค่าอินพุตของข้อมูลชุดแรก เพื่อใช้ในการคำนวณหาค่าเอาต์พุตของโครงข่ายประสาทเทียม

2.1.3.5 คำนวณค่าผลรวมของโหนดในชั้นซ่อน (S_m) ก่อนทำการปรับค่ารวมด้วยฟังก์ชันกระตุ้น (Activation Function) ฟังก์ชันซิกมอยด์ ซึ่งจะได้ค่าของโหนดในชั้นซ่อนที่อยู่ในช่วง $[0, 1]$ โดยมีรายละเอียดดังสมการ 2.1 ถึงสมการ 2.3 ค่าเอาต์พุตของชั้นซ่อนก่อนทำการปรับค่า ดังสมการที่ 2.1

$$S_m = \sum_{n=1}^N x_n \times w_{nm} \quad (2.1)$$

ค่าเอาต์พุตของชั้นซ่อนหลังทำการปรับค่า ดังสมการที่ 2.2

$$y_m = f(s_m) \quad (2.2)$$

ฟังก์ชันที่ใช้ปรับค่า $f(x)$ ดังสมการที่ 2.3

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

2.1.3.6 คำนวณค่าเอาต์พุตของโหนดในชั้นเอาต์พุตด้วยสมการ 2.4 จากนั้นทำการปรับค่าผลรวมด้วยฟังก์ชันกระตุ้นซิกมอยด์ดังสมการที่ 2.3 ซึ่งจะได้ค่าของโหนดในชั้นเอาต์พุตที่อยู่ในช่วง $[0, 1]$ สำหรับค่าของผลลัพธ์ในโหนดชั้นเอาต์พุตนั้น หลังทำการปรับค่าแสดงดังสมการที่ 2.4

$$v_j = \sum_{m=1}^M y_m \times w_{mj} \quad (2.4)$$

ค่าเอาต์พุตของชั้นเอาต์พุตหลังทำการปรับค่า ดังสมการที่ 2.5

$$y_m = f(s_m) \quad (2.5)$$

2.1.3.7 การหาค่าความผิดพลาดของโหนดในชั้นเอาต์พุตและทำการปรับน้ำหนักนั้น โดยการนำเอาต์พุตที่คำนวณได้จริงเปรียบเทียบกับเอาต์พุตที่กำหนดไว้ เพื่อหาค่าความผิดพลาดของข้อมูล โดยถ้าค่าความผิดพลาดของข้อมูลน้อยกว่าค่าความผิดพลาดที่ยอมรับได้แล้วนั้น โครงข่ายประสาทเทียมก็จะทำการรับข้อมูลชุดต่อไปเข้าสู่โครงข่าย โดยถ้าไม่ได้ทำการปรับน้ำหนักแล้วนั้น โครงข่ายประสาทเทียมจะทำการรับข้อมูลแถวถัดไป และจะกลับไปทำข้อ 2.1.3.5 แต่ถ้าเป็นข้อมูลชุดสุดท้ายของข้อมูลแล้วนั้น จะไปทำข้อ 2.1.3.8

ค่าความผิดพลาดในแต่ละแถวของข้อมูล ดังสมการที่ 2.6

$$e^{(q)} = \frac{1}{2} \sum_{j=1}^J (t_j - z_j)^2 \quad (2.6)$$

การปรับน้ำหนักระหว่างโหนดในชั้นซ่อนและชั้นเอาต์พุต ดังสมการที่ 2.7

$$w_{mj}^{(r+1)} = w_{mj}^{(r)} + \eta \left\{ (t_j^{(q)} - z_j^{(q)}) \times [z_j^{(q)} (1 - z_j^{(q)})] \times y_m^{(q)} \right\} \quad (2.7)$$

การปรับน้ำหนักระหว่างโหนดในชั้นอินพุตและชั้นซ่อน ดังสมการที่ 2.8

$$w_{nm}^{(r+1)} = w_{nm}^{(r)} + \eta \left\{ \sum_{j=1}^J (t_j^{(q)} - z_j^{(q)}) [z_j^{(q)} (1 - z_j^{(q)})] w_{mj}^{(r)} \right\} \times [y_m^{(q)} (1 - y_m^{(q)})] [x_n^{(q)}] \quad (2.8)$$

2.1.3.8 คำนวณค่าความผิดพลาดรวมเฉลี่ย (Mean squared error : MSE) ในทุกแถวข้อมูล โดยนำค่าผิดพลาดของแต่ละแถวของชุดข้อมูลมารวมกัน แล้วทำการหาค่าเฉลี่ย เพื่อใช้ในการตรวจสอบว่าผลลัพธ์ของทุกข้อมูลในแต่ละรอบนั้นมีค่าน้อยกว่าค่าความผิดพลาดที่ยอมรับได้หรือไม่ โดยถ้าน้อยกว่าค่าความผิดพลาดที่ยอมรับได้ให้จบการเรียนรู้ แต่ถ้าค่าความผิดพลาดเฉลี่ยมากกว่าค่าที่ยอมรับได้ให้ตรวจสอบว่าได้ทำการเรียนรู้ครบตามจำนวนรอบที่กำหนดไว้

หรือไม่ ถ้าครบแล้วให้จบการเรียนรู้ แต่ถ้ายังไม่ครบให้กลับไปทำข้อ 2.1.3.4 ใหม่ ซึ่งก็คือเริ่มต้นการเรียนรู้รอบใหม่ค่าความผิดพลาดรวมเฉลี่ย ดังสมการที่ 2.9

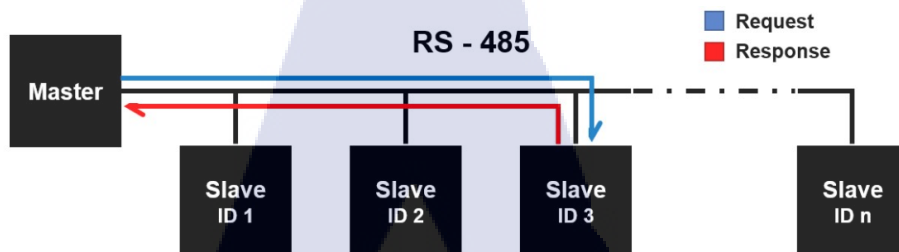
$$E = \frac{1}{Q} \sum_{q=1}^Q e^{(q)} \quad (2.9)$$

2.2 การสื่อสารด้วยมาตรฐาน Modbus (Modbus Protocol Communications) [4]

การสื่อสารด้วยมาตรฐาน Modbus เป็นหนึ่งในมาตรฐานการสื่อสารแบบอนุกรม (Serial communications protocol) ที่ใช้กันอย่างแพร่หลายในระบบอัตโนมัติอุตสาหกรรม (Industrial Automation Systems : IAS) เพื่อสร้างการเชื่อมโยงข้อมูลระหว่างอุปกรณ์ต่างๆ เช่น อุปกรณ์ควบคุมพีแอลซี (Programmable Logic Controllers : PLC) อุปกรณ์ตรวจวัด (Sensor) อุปกรณ์เครื่องกล อุปกรณ์ขับเคลื่อน (Actuator) หน่วยตรวจวัดระยะไกล (Remote Terminal Unit : RTU) รวมถึงระบบคอมพิวเตอร์ที่ใช้ในการควบคุมและแสดงสถานะของอุปกรณ์ต่างๆ (Supervisory Control and Data acquisition : SCADA)

Modbus ถูกพัฒนาขึ้นในปีค.ศ. 1979 โดยบริษัท Modicon (ปัจจุบันคือ Schneider Electric) เป็นโพรโทคอลที่ถูกใช้กันอย่างกว้างขวางในงานอุตสาหกรรมเนื่องจากความง่ายในการใช้งานและมีความน่าเชื่อถือ ในปัจจุบันนี้การสื่อสารสามารถแบ่งได้เป็น 2 ระบบคือ Modbus RTU และ Modbus TCP โดยความแตกต่างอยู่ที่โพรโทคอลการสื่อสารที่ใช้ ในระบบ Modbus RTU จะใช้โพรโทคอลการสื่อสารแบบอนุกรม (Serial-based Protocol) ในขณะที่ระบบ Modbus TCP จะใช้โพรโทคอลการสื่อสารแบบอีเทอร์เน็ต (Ethernet-based Protocol) ซึ่งทั้งสองแบบจะแตกต่างกันตรงที่ความเร็วและระยะทางในการรับส่งข้อมูล โดย Modbus RTU สามารถรับส่งได้ระยะทางสูงสุดถึง 1.2 กิโลเมตร (ที่ความเร็ว 57.6 kbps) ในขณะที่ Modbus TCP สามารถรับส่งได้ที่ความเร็วสูงสุดถึง 100 Mbps (ที่ระยะทาง 100 เมตร) โดยรายละเอียดมีดังต่อไปนี้

2.2.1 การสื่อสารแบบ Modbus RTU



รูปที่ 2.7 การสื่อสารแบบอนุกรมด้วย RS – 485 สำหรับ Modbus RTU

Modbus RTU คือ โพรโทคอลที่ใช้การสื่อสารแบบอนุกรม (Serial-based Protocol) ด้วยสถาปัตยกรรมการสื่อสารแบบ Master/Slave หรืออาจกล่าวได้ว่าอุปกรณ์ Slave จะไม่ส่งข้อมูล (Response) กลับมาจนกว่าจะมีการร้องขอ (Request) จากอุปกรณ์ Master ดังรูปที่ 2.7 Modbus RTU โดยทั่วไปจะใช้การสื่อสารในระดับกายภาพ (Physical Layer) แบบ RS-232 หรือ RS-485 ข้อมูลในโพรโทคอล Modbus จะถูกเก็บ 4 รูปแบบ คือ Output coils, Input contacts, Input registers และ Holding registers

โดย 2 แบบแรก Output coils และ Input contacts แต่ละแอดเดรสจะเก็บค่าเพียง 1 บิต หรือมีค่าได้แค่ “0” กับ “1” เปรียบเสมือนค่าการเปิดและปิดของอุปกรณ์รีเลย์และสวิตช์ที่พบได้ในระบบงานอัตโนมัติอุตสาหกรรม

ในขณะที่ 2 แบบหลัง Input registers และ Holding registers สามารถเก็บค่าเป็นตัวเลขได้ถึง 16 บิต เปรียบเสมือนค่าที่มาจากอุปกรณ์ตรวจวัดที่ส่งข้อมูลแบบอนาล็อก (Analog)

การสื่อสารของข้อมูลในระบบ Modbus RTU จะรับส่งเป็นชุดข้อมูล โดยที่ใน 1 ชุดข้อมูลนั้นจะประกอบด้วยส่วน 6 ส่วน ดังแสดงในรูปที่ 2.8 ซึ่งเริ่มต้นด้วยชุดบิตเริ่มต้น (Start bits) อ้างอิงถึงการเริ่มต้นชุดข้อมูล ตามด้วยค่าตำแหน่งแอดเดรส (Address) ของอุปกรณ์ที่ต้องการสื่อสารด้วยตามด้วยชุดสำหรับ Function Code และข้อมูลที่ต้องการ (Data) ต่อด้วยชุดข้อมูลตรวจสอบความผิดพลาด (Cyclic Redundancy Check : CRC) และชุดบิตปิดท้าย (End bits) อ้างอิงถึงการสิ้นสุดข้อมูล

Field Name	Bit length	Function
Start	28	At least 3.5 character times of silence (mark condition)
Address	8	Station address
Function	8	Indicates function code eg. Read coils/holding registers
Data	n x 8	Data + length will be filled depending on message type
CRC	16	Cyclic Redundancy Check
End	28	At least 3.5 character times of silence between frames

รูปที่ 2.8 ชุดข้อมูลสำหรับการสื่อสาร Modbus RTU

2.2.1.1 ฟังก์ชันการทำงานสำหรับ Modbus RTU (Function code)

ชุดฟังก์ชันการทำงานสามารถแบ่งหน้าที่ต่างๆได้ตามรหัส หรือ Function code รายละเอียดแสดงดังรูปที่ 2.9 โดยหลักๆแล้วจะมีฟังก์ชันการทำงานอยู่ 2 แบบ คือ การอ่าน (Read) และเขียน (Write) โดยสามารถเลือกที่จะอ่านหรือเขียนข้อมูลไปยัง Coils หรือ Contacts สำหรับข้อมูลแบบดิจิตอล (Digital) หรือ “0” กับ “1” และ Registers สำหรับอ่านหรือเขียนข้อมูลแบบแอนะล็อก โดยมีขนาด 16 บิต หรือ ตั้งแต่ 0000 ถึง FFFF

Function Code (DEC)	Action	Data Type	Object Type
01	Read	Single bit	Output Coils
05	Write Single	Single bit	Output Coils
15	Write Multiple	Single bit	Output Coils
02	Read	Single bit	Input Contacts
04	Read	Word (16bit)	Input Registers
03	Read	Word (16bit)	Holding Registers
06	Write Single	Word (16bit)	Holding Registers
16	Write Multiple	Word (16bit)	Holding Registers

รูปที่ 2.9 รายละเอียดชุดข้อมูล Function Code

2.2.1.2 ตำแหน่งแอดเดรสของ Modbus RTU (Address)

ตำแหน่งแอดเดรสใน Modbus RTU จะมีขนาด 16 บิต หรือ 65535 ตำแหน่ง ในแต่ละรูปแบบการทำงาน โดยที่ Output coils (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 000001) Input contacts (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 100001) Input registers (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 300001) และ Holding registers (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 400001) ดังรูปที่ 2.10

Register Number (DEC)	Register Address (HEX)	Extended Register Number (DEC)	Extended Register Address (HEX)	Type	Object Type
00001-09999	0000 to 270E	000001-065535	0000 to FFFF	Read-Write	Output Coils
10001-19999	0000 to 270E	100001-165535	0000 to FFFF	Read-Only	Input Contacts
30001-39999	0000 to 270E	300001-365535	0000 to FFFF	Read-Only	Input Registers
40001-49999	0000 to 270E	400001-465535	0000 to FFFF	Read-Write	Holding Registers

รูปที่ 2.10 ตำแหน่งแอดเดรสใน Modbus RTU โดยแบ่งตามรูปแบบการทำงาน

2.2.1.3 ชุดข้อมูล (Data)

ในส่วนชุดข้อมูล Data Field นั้นจะถูกแบ่งเป็น 2 ชุด คือ ชุดคำสั่งสำหรับการอ่าน (Read Command) ตามรูปที่ 2.11 และชุดคำสั่งสำหรับการเขียน (Write Command) ตามรูปที่ 2.12 โดยชุดคำสั่งทั้ง 2 จะถูกส่งจากอุปกรณ์ที่ทำหน้าที่เป็น Master เท่านั้น เพื่อส่งไปยังอุปกรณ์ Slave ที่ต้องการสื่อสาร

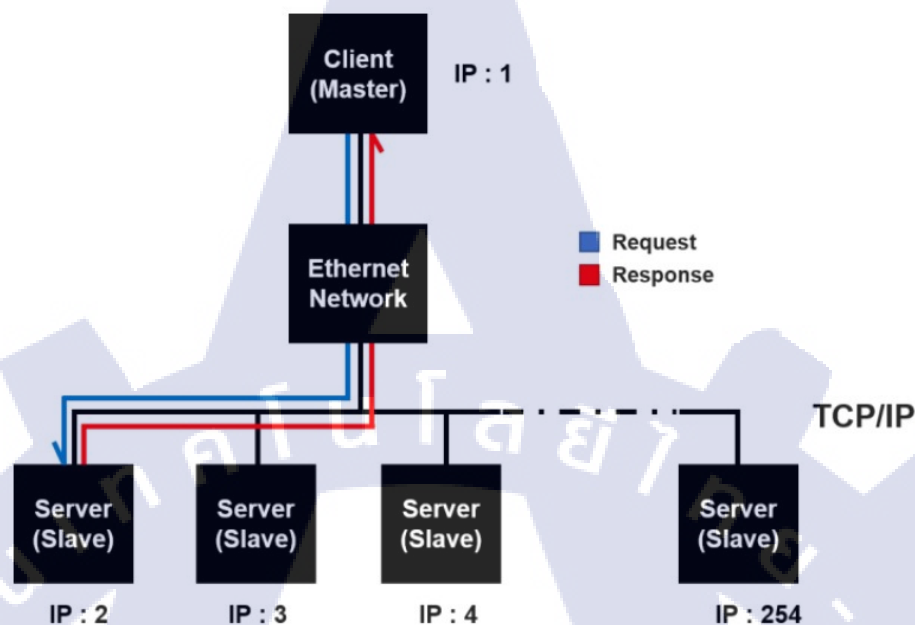
Read Command	
Request Message	start register address (2 bytes) + no. of registers (2 bytes)
Response Message	byte count (1 byte) + data (no. of registers*2 bytes)

รูปที่ 2.11 ชุดคำสั่งสำหรับการอ่าน (Read Command)

Write Command	
Request Message	start register address (2 bytes) + no. of registers (2 bytes) + byte count (1 byte) + data (no. of registers*2 bytes)
Response Message	start register address (2 bytes) + no. of registers (2 bytes)

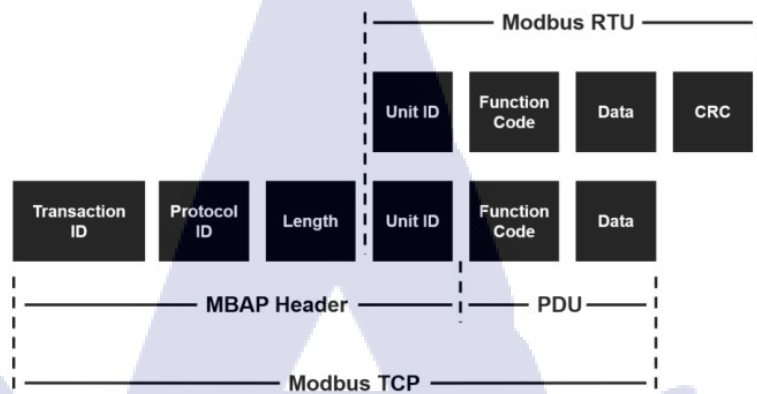
รูปที่ 2.12 ชุดคำสั่งสำหรับการเขียน (Write Command)

2.2.2 การสื่อสารแบบ Modbus TCP



รูปที่ 2.13 การสื่อสารแบบอีเทอร์เน็ตสำหรับ Modbus TCP

Modbus TCP คือ โพรโทคอลที่ครอบ Modbus RTU เพื่อใช้ในการสื่อสารแบบอีเทอร์เน็ต (Ethernet-based protocol) ด้วย TCP/IP (Transmission control protocol) ที่พอร์ต (Port) 502 แทนการใช้การสื่อสารแบบอนุกรม ดังรูปที่ 2.13 ทำให้อุปกรณ์สามารถสร้างการสื่อสารผ่านเครือข่ายเฉพาะบริเวณ (Local area network :LAN) หรือ เครือข่ายอินเทอร์เน็ต (Internet network) รวมไปถึงการเชื่อมต่อแบบไร้สาย (Wireless) โดยมีอุปกรณ์กระจายสัญญาณ (Router หรือ Access point) เป็นตัวกลางในการเชื่อมต่อ โดยชุดข้อความใน Modbus TCP มีรายละเอียดดังรูปที่ 2.14 และ 2.15 เริ่มต้นข้อมูลด้วย Modbus application protocol (MBAP) Header ซึ่งประกอบด้วย Transaction ID, Protocol ID, Length, Unit ID ซึ่งเพิ่มเติมขึ้นจาก Modbus RTU ส่วนชุดข้อมูล Function code และ Data จะยังคงเหมือนเดิม ยกเว้นชุดข้อมูล CRC สำหรับเช็คความผิดพลาดจะไม่มี แต่เปลี่ยนไปใช้ของ Ethernet ใน Data link layer แทน



รูปที่ 2.14 ส่วนประกอบชุดข้อมูลของ Modbus TCP เทียบกับ Modbus RTU

Area Name		Area Size	Description
MBAP header (MODBUS® application header)	Transaction ID	2 bytes	Used by the master for matching of the response message from the slave.
	Protocol ID	2 bytes	Indicates the protocol of the PDU (protocol data unit). Stores 0 in the case of MODBUS® /TCP.
	Message length	2 bytes	Stores the message size in byte unit. The message length after this field is stored. (See the above figure.)
	Module ID	1 byte	Used to specify the slave connected to the other line. E.g. MODBUS® serial protocol.
PDU (Protocol data unit)	Function code	1 byte	The master specifies the processing to be performed for the slave.
	Data	1 to 252 bytes	[When master sends request message to slave] Stores the requested processing. [When slave sends response message to master] Stores the result of processing execution.

รูปที่ 2.15 รายละเอียดของแต่ละ Field ในหนึ่งเฟรมของ Modbus TCP

2.3 การวัดประสิทธิภาพด้วย Confusion matrix

การวัดประสิทธิภาพด้วย Confusion Matrix เป็นการประเมินประสิทธิภาพการจำแนกข้อมูลของแบบจำลองการเรียนรู้ของเครื่องจักรและแสดงผลด้วยเมตริกเป็นส่วนสำคัญ ในขั้นตอนสุดท้ายของการทำแบบจำลอง เนื่องจากการวัดประสิทธิภาพของการจำแนกข้อมูล (Classifier) จะบอกถึงความน่าเชื่อถือของแบบจำลอง รูปแบบของเมตริกวัดประสิทธิภาพ (Confusion Matrix) แสดงได้ดังรูปที่ 2.15

	Actual Positive	Actual Negative
Predict Positive	TP	FP
Predict Negative	FN	TN

รูปที่ 2.16 เมตริกวัดประสิทธิภาพ (Confusion Matrix)

เมตริกวัดประสิทธิภาพเป็นผลสรุปของการประเมินความสามารถในการจำแนกข้อมูลของแบบจำลองการจำแนกข้อมูล โดยทดสอบด้วยชุดทดสอบ จากรูปที่ 2.15 TP หรือ True Positive คือ จำนวนข้อมูลทดสอบที่เป็นคลาส Positive และแบบจำลองทำการจำแนกได้ถูกต้องว่าเป็น Positive ค่า FP หรือ False Positive จำนวนข้อมูลทดสอบที่ไม่ใช่คลาส Positive แต่แบบจำลองทำการจำแนกผิดว่าเป็นคลาส Positive ค่า TN หรือ True Negative คือ จำนวนข้อมูลทดสอบที่เป็นคลาส Negative และแบบจำลองทำการจำแนกได้ถูกต้องว่าเป็นคลาส Negative ค่า FN หรือ False Negative คือจำนวนข้อมูลทดสอบที่เป็นคลาส Positive แต่แบบจำลองทำการจำแนกผิดว่าเป็นคลาส Negative ค่า TP, FP, TN และ FN สามารถนำมาใช้คำนวณมาตรวัดต่าง ๆ เพื่อประเมินประสิทธิภาพ การจำแนกข้อมูลของแบบจำลอง มาตรวัดที่นิยมใช้โดยทั่วไปประกอบด้วย ความแม่นยำ (P), ความถูกต้อง (Acc), Recall (R) และ F-measure (f) โดยแสดงสมการดังนี้

$$P = \frac{TP}{TP + FP} \quad (2.10)$$

$$R = \frac{TP}{TP + FN} \quad (2.11)$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.12)$$

$$f = \frac{2PR}{P + R} \quad (2.13)$$

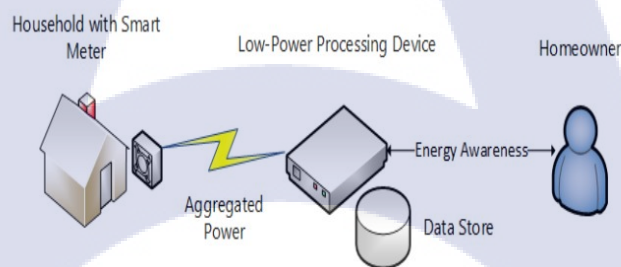
ในงานวิจัยนี้จะใช้ Confusion matrix และ มาตรฐานวัดความถูกต้อง (Acc) เป็นหลักในการประเมินประสิทธิภาพ

2.4 งานวิจัยที่เกี่ยวข้อง

โดยมีงานวิจัยมากมายที่เกี่ยวข้องกับการตรวจติดตามสถานการณ์ทำงานของอุปกรณ์ไฟฟ้า ดังแสดงต่อไปนี้

2.4.1 Low-Complexity Non-Intrusive Load Monitoring Using Unsupervised Learning and Generalized Appliance Models [5]

ในงานวิจัยนี้ได้พัฒนาอัลกอริทึมสำหรับแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละตัว จากข้อมูลกำลังไฟฟ้าที่เก็บได้จากมิเตอร์อัจฉริยะที่อัตราสุ่ม 1 Hz สำหรับโครงสร้างของระบบจะประกอบด้วยมิเตอร์อัจฉริยะและอุปกรณ์ที่ในการจัดการข้อมูลดังแสดงในรูปที่ 2.16 สำหรับโปรแกรมที่ใช้ในการแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าจะประกอบด้วยส่วนต่างๆดังนี้



รูปที่ 2.17 โครงสร้างของระบบตรวจรู้สถานะโหลดแบบไม่รุกร้า

2.4.1.1 การเก็บข้อมูล (Data acquisition) ข้อมูลจะถูกเก็บผ่านมิเตอร์อัจฉริยะ และส่งข้อมูลไปยังอุปกรณ์ประมวลผลซึ่งทำการประมวลผลตามอัลกอริทึมที่เขียนไว้ก่อนส่งข้อมูลที่ประมวลผลได้แสดงให้ผู้ใช้ไฟฟ้า

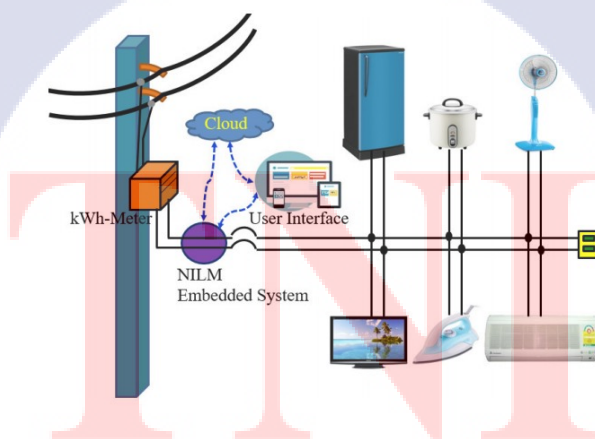
2.4.1.2 การตรวจจับเหตุการณ์และดึงคุณลักษณะที่กำหนดจากข้อมูล (Event detection and Feature extraction) คือกระบวนการในการตรวจจับการเปลี่ยนแปลงข้อมูล และดึงคุณลักษณะจากข้อมูลเพื่อทำการสร้างแบบจำลองที่ใช้อธิบายข้อมูลนั้น โดยคุณลักษณะที่ใช้ในงานวิจัยนี้คือ กำลังไฟฟ้า (P) หลังจากได้ข้อมูลคุณลักษณะที่ต้องการ ข้อมูลดังกล่าวจะผ่านกระบวนการแบ่งกลุ่มข้อมูล (Clustering) ด้วยเทคนิคแบบ Entropy Index Constraints Competitive Agglomeration (EICCA)

2.4.1.3 การเรียนรู้แบบจำลองและการอนุมาน ข้อมูลที่ได้จากขั้นตอนที่ 2 จะถูกนำมาสร้างเป็นแบบจำลองการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละชนิด โดยจะสามารถแบ่งแบบจำลองของเครื่องใช้ไฟฟ้าออกเป็น 2 ชนิดคือ 1) แบบ เปิด-ปิด 2) แบบปรับได้หลายระดับ ซึ่งแบบจำลองที่ได้นี้จะถูกนำไปใช้ในการอนุมานเพื่อแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้า

โดยในงานวิจัยนี้ได้ทดสอบความถูกต้องในการแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าของโปรแกรมผ่านดัชนีชี้วัดความถูกต้องหลายตัว แต่ตัวที่สำคัญคือ Total energy correctly assigned (TECA) ซึ่งเป็นค่าแสดงปริมาณพลังงานที่แยกได้ถูกต้องจากปริมาณพลังงานทั้งหมด โดยได้ทำการทดสอบโปรแกรมผ่านฐานข้อมูล Reference Energy Disaggregation Dataset (REDD) ใน 3 บริบท คือ ในแง่ของประสิทธิภาพของโปรแกรม ความสามารถในการเรียนรู้แบบจำลองและการอนุมานและความถูกต้องในการแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้ากำลังต่ำ ซึ่งจากการทดสอบพบว่าโปรแกรมสามารถให้ความถูกต้อง (TECA) มากกว่า 90% ทั้งในกรณีที่เครื่องใช้ไฟฟ้าปริมาณมากในวงจรเดียวและเครื่องใช้ไฟฟ้าที่กำลังต่ำ

2.4.2 Non-Intrusive Load Monitoring (NILM) Using an Artificial Neural Network in Embedded system with Low Sampling Rate [6]

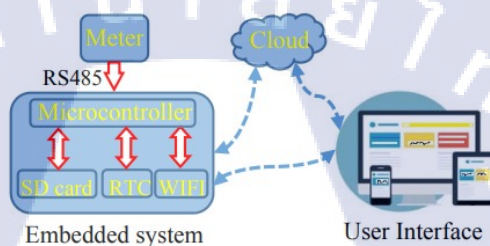
งานวิจัยนี้ได้พัฒนาระบบมิเตอร์วัดพลังงานที่สามารถแยกการใช้ไฟฟ้าของอุปกรณ์ไฟฟ้า (NILM) โดยติดตั้งมิเตอร์เพียงจุดเดียว ดังแสดงในรูปที่ 2.17 ในงานวิจัยจะทำการแยกปริมาณการใช้



รูปที่ 2.18 การติดตั้งของระบบตรวจสอบสถานะโหลดแบบไม่รุกราน

ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละเครื่องออกจากปริมาณการใช้ไฟฟ้ารวม โดยเครื่องใช้ไฟฟ้าที่ใช้ในงานวิจัยประกอบด้วย ตู้เย็น โทรทัศน์ เครื่องปรับอากาศและหม้อหุงข้าว สำหรับระบบการทำงานของมิเตอร์วัดพลังงานแบบ NILM มีทั้งหมด 4 ขั้นตอนดังนี้

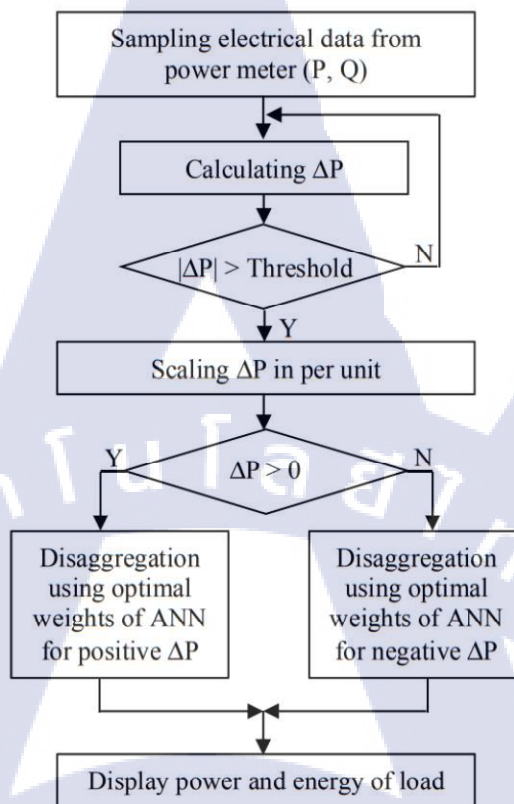
2.4.2.1 การเก็บข้อมูล (Data) ในงานวิจัยนี้ได้ใช้มิเตอร์ทำการวัดกำลังไฟฟ้าจริง ที่อัตราสุ่ม 1 Hz และส่งผ่านข้อมูลด้วยสัญญาณอนุกรม RS 485 ให้กับระบบสมองกลฝังตัว ที่ใช้ในการประมวลผล เก็บข้อมูลและส่งข้อมูลไปยังระบบคลาวด์ ทำให้สามารถแสดงผลไปยังผู้ใช้ผ่านอุปกรณ์โทรศัพท์เคลื่อนที่และคอมพิวเตอร์ โดยแสดงโครงสร้างระบบตามรูปที่ 2.18



รูปที่ 2.19 โครงสร้างของระบบตรวจสอบสถานะโหลดแบบไม่รุกร้า

2.4.2.2 การตรวจจับการเปลี่ยนแปลงของสัญญาณกำลังไฟฟ้า (Detection) จะทำการตรวจวัดการเปลี่ยนแปลงของกำลังไฟฟ้าโดยใช้ค่าเฉลี่ย 6 จุด ถ้าค่าเฉลี่ยทั้ง 6 จุด น้อยกว่าค่าเกณฑ์แสดงว่าไม่มีการเปลี่ยนแปลงของสัญญาณ ในขณะที่ถ้ามากกว่าค่าเกณฑ์แสดงว่ามีการเปลี่ยนแปลงของสัญญาณเกิดขึ้น โดยโปรแกรมดังกล่าวจะทำงานบนระบบสมองกลฝังตัว และข้อมูลที่ได้อาจบันทึกลงบนหน่วยความจำ SD card

2.4.2.3 การแยกการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละเครื่องออกจากการใช้ไฟฟ้ารวม (Data disaggregation) ในงานวิจัยนี้ได้ใช้อัลกอริทึมวิธี Artificial neural network (ANN) ในการแยกปริมาณการใช้ไฟฟ้าของตู้เย็น เครื่องปรับอากาศ โทรทัศน์และหม้อหุงข้าว ดังแสดงขั้นตอนวิธีการดังรูปที่ 2.19



รูปที่ 2.20 อัลกอริทึมการแยกการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละเครื่อง

2.4.2.4 การแสดงผล (Information presentation) ข้อมูลที่เก็บอยู่บนระบบคลาวด์จะถูกส่งไปแสดงผลยังอุปกรณ์สื่อสารผ่านระบบอินเทอร์เน็ตทุกทรัพย์สิน (ในงานวิจัยนี้ใช้ระบบ ThingSpeak)

หลังจากได้ทดสอบด้วยการจำลองผ่านโปรแกรม MATLAB พบว่าค่าเฉลี่ยของความถูกต้องในการแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละชนิดอยู่ที่ 98% ในขณะที่การทดสอบติดตั้งจริงด้วยระบบสมองกลอัจฉริยะจะมีความถูกต้องอยู่ที่ 95%

2.4.3 An Approach for Unsupervised Non-Intrusive Load Monitoring of Residential Appliances [7]

งานวิจัยนี้ได้พัฒนาระบบในการการแยกปริมาณการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าแต่ละชนิดจากปริมาณการใช้รวม โดยเน้นไปที่ระบบบนพื้นฐานตรวจจับเหตุการณ์ (Event-based) และใช้อัลกอริทึมการเรียนรู้แบบไม่มีผู้สอน ระบบจะประกอบด้วยส่วนต่างๆดังนี้

2.4.3.1 ส่วนตรวจจับเหตุการณ์ ในงานวิจัยนี้จะใช้วิธี ตรวจจับเหตุการณ์แบบ Clustering based คือการปรับขยายมิติที่ตรวจจับทั้งในด้านกำลังไฟฟ้าจริงและเสมือน (P,Q) รวมทั้งมิติด้านช่วงเวลา ตามเกณฑ์และอัลกอริทึมที่กำหนดไว้

2.4.3.2 การแยกคุณลักษณะ โดยคุณลักษณะที่เลือกใช้ในงานวิจัยนี้คือ การเปลี่ยนแปลงของกำลัง ไฟฟ้า ($\Delta\psi$) และการเปลี่ยนแปลงของสัญญาณกำลังไฟฟ้าแบบยอดแหลม (Spike) ($\delta\psi$) โดยกำลังไฟฟ้าประกอบด้วยกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือน ซึ่งแยกผ่านสมการดังนี้

$$\Delta\psi^x_i = \psi^x_i (N^{\psi_i} - 1) - \psi^x_i(0) \quad (2.14)$$

และ

$$\delta\psi^x_i = \text{sign}(\Delta\psi^x_i) [\max_n \psi^x_i(n) - \min_n \psi^x_i(n)] \quad (2.15)$$

2.4.3.3 การแบ่งกลุ่มข้อมูล ข้อมูลที่ได้จากข้อ 2 จะถูกทำการแบ่งกลุ่มผ่านอัลกอริทึม Mean-shift

2.4.3.4 การจับคู่ ในขั้นตอนนี้ข้อมูลที่ได้จากขั้นตอนที่ 3 จะถูกนำมาจับกลุ่มเพื่อสร้างเป็นแบบจำลองการใช้ไฟฟ้าของเครื่องใช้ไฟฟ้าเพื่อให้รวดเร็วในการทำงานของโปรแกรม

ได้ทำการทดสอบโปรแกรมด้วยชุดข้อมูล BLUED โดยทำการวัดความถูกต้องผ่านดัชนี True positive percentage (TPP) และความผิดพลาดผ่านดัชนี False positive percentage (FPP) โดยผลการทดลองแสดงดังรูปที่ 2.20

	TPP	FPP	Events E
Phase A	98.50%	0.55%	886
Phase A	70.50%	8.75%	1579

รูปที่ 2.21 ผลการทดลองด้วยชุดข้อมูล BLUED

2.4.4 Electric Load Disaggregation in Smart Metering Using a Novel Feature Extraction Method and Supervised Classification [8]

งานวิจัยนี้ได้พัฒนาระบบจำแนกโหลดไฟฟ้าโดยใช้เทคนิคการคัดกรองคุณลักษณะ (Feature Extraction) ร่วมกับการเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Machine Learning) ระบบจำแนกโหลดไฟฟ้าในงานวิจัยจะประกอบด้วย 4 ขั้นตอน คือ การรวบรวมข้อมูล (Data acquisition) การตรวจจับเหตุการณ์ (Event detection) การคัดกรองคุณลักษณะ (Feature extraction) และการคัดแยกชนิดโหลดทางไฟฟ้า (Classification) ในงานวิจัยนี้ได้ใช้โมเดลอัจฉริยะในการวัดข้อมูลกระแสไฟฟ้าของโหลด จากนั้นกระแสไฟฟ้างัดกล่าวจะเขียนอยู่ในรูปสมการ Linear combination of M cisoids (Complex-valued sinusoidal signals) ดังแสดงตามสมการที่ 2.16

$$i(t) \approx \sum_{m=1}^M r_m \exp\{(\alpha_m + j2\pi f_m)t\} + b(t) \quad (2.16)$$

โดยที่ค่า r_m เป็นค่า residue ของ m^{th} cisoids, α_m เป็นค่า attenuation factor, f_m เป็นค่าความถี่ และ $b(t)$ เป็นค่า Gaussian Noise จากนั้นใช้วิธี Matrix Pencil Method เพื่อเขียนสมการใหม่ให้อยู่ในรูปสมการที่ 2.17

$$i(k) \approx \sum_{m=1}^M r_m z_m^k + b(k) \quad (2.17)$$

โดย

$$z_m = \exp\{(\alpha_m + j2\pi f_m)t_s\}$$

โดยค่าของ z_m และ r_m จะถูกนำไปใช้ในขั้นตอนการแยกชนิดโหลดต่อไป ในการแยกชนิดโหลด ในงานวิจัยนี้ได้ใช้อัลกอริทึม 3 ชนิด คือ k-Nearest Neighbor, Naïve Bayes และ Support Vector Machine จากนั้นได้ทำการประเมินประสิทธิภาพดังแสดงผลในตาราง 2.1

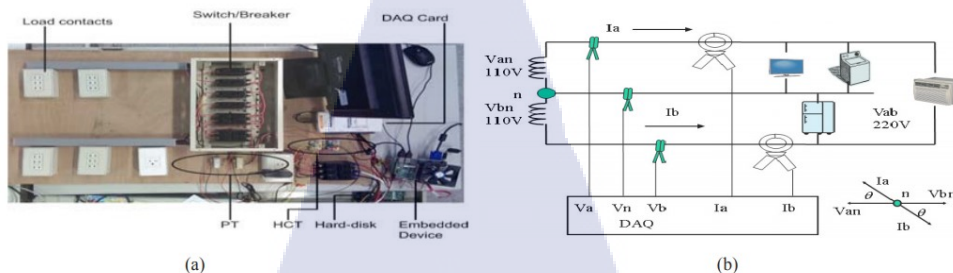
ตารางที่ 2.1 การประเมินประสิทธิภาพการแยกชนิดโหลดของอัลกอริทึมทั้ง 3 ชนิด

Classifier	k-NN	Naïve Bayes	SVM
Incandescent	100%	100%	100%
Halogen	100%	97%	100%
Economy	100%	99%	94%
Water Heater	100%	100%	100%
Convector	96%	100%	100%
Oven	100%	100%	100%
Hot Plate	97%	50%	73%
Television	100%	100%	98%
Computer	100%	100%	100%
Overall Accuracy	99.22%	94%	96.11%

โดยสามารถสรุปผลจากการทดลองได้ว่าวิธี k-Nearest Neighbor จะให้ประสิทธิภาพความแม่นยำมากที่สุด แต่อย่างไรก็ตามอีก 2 วิธีก็ให้ความแม่นยำที่สูงเช่นกัน อย่างไรก็ตามเมื่อพิจารณาจากข้อจำกัดในการนำไปประยุกต์ใช้กับอุปกรณ์ไมโครคอนโทรลเลอร์ พบว่ามีเพียงวิธี k-Nearest Neighbor และ Naïve Bayes ที่สามารถใช้งานได้เท่านั้น

2.4.5 A Non-Intrusive Load Monitoring System Using an Embedded System for Applications to Unbalanced Residential Distribution System [9]

ในงานวิจัยนี้ผู้วิจัยได้พัฒนาทั้งอุปกรณ์และโปรแกรมเพื่อทำการจำแนกชนิดของโหลดไฟฟ้าจากโหลดรวม โดยใช้หม้อแปลงแรงดันและหม้อแปลงกระแสชนิด Hall effect ทำการวัดสัญญาณแรงดันและกระแสตามลำดับ สัญญาณที่วัดได้จะถูกส่งไปประมวลผลเพื่อหาค่ากำลังไฟฟ้าจริง (P) และกำลังไฟฟ้าเสมือน (Q) โดยไมโครคอนโทรลเลอร์ Portwll รุ่น OQ7-C100XL ซึ่งแสดงระบบจำแนกชนิดของโหลดไฟฟ้าดังรูปที่ 2.21



รูปที่ 2.22 a) ระบบจำแนกชนิดของโหลดไฟฟ้า b) การติดตั้งระบบเข้ากับระบบไฟฟ้า

สำหรับอัลกอริทึมที่ใช้ในการจำแนกชนิดของโหลดไฟฟ้า ในงานวิจัยนี้ได้ใช้เทคนิค Particle swarm optimization (PSO) ในการหาค่าตัวแปรน้ำหนัก (W) และไบแอส ในขั้นตอนการสอนอัลกอริทึมการเรียนรู้ของเครื่องแบบ Neuron network โดยทำการทดสอบระบบด้วยการจำแนกโหลดเครื่องใช้ไฟฟ้า 3 ชนิด คือ เครื่องดูดฝุ่นสูญญากาศขนาด 900 วัตต์(1), เครื่องดูดฝุ่นสูญญากาศขนาด 1200 วัตต์(2) และเครื่องเป่าผมขนาด 600 วัตต์(3) ซึ่งผลการทดสอบแสดงดังตารางที่ 2.2

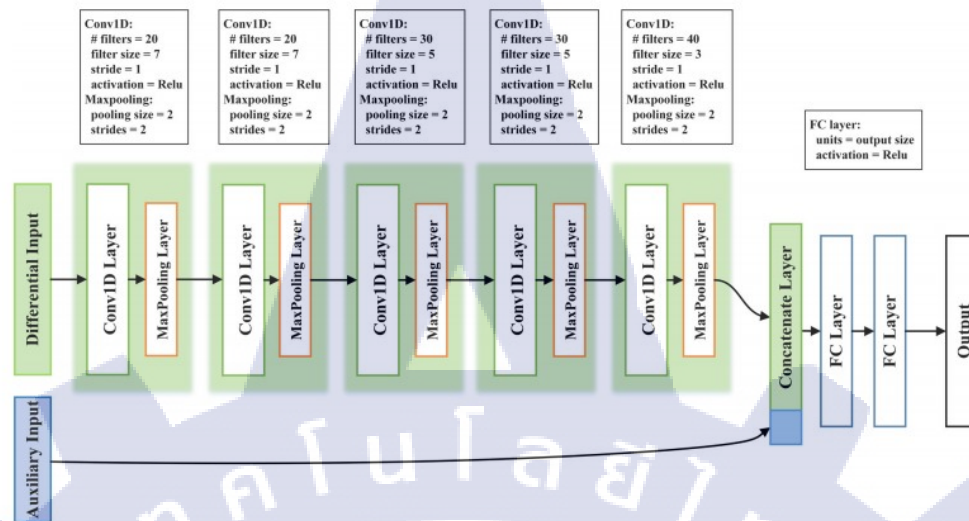
ตารางที่ 2.2 ผลการทดสอบในการจำแนกโหลดเครื่องใช้ไฟฟ้าทั้ง 3 ชนิด

System Voltage (V)	209	211.2	213.4	215.6	217.8	220	221.2	222.2	226.6	228.8	231
Load 1	1	1	1	1	1	1	1	1	1	1	1
Load 2	2	2	2	2	2	2	2	2	2	2	2
Load 3	3	3	3	3	3	3	3	3	3	3	3
Load 1+2	1+2	1+2	1+2	1+2	1+2	1+2	1+2	1+2	1+2	1+2	1+2
Load 1+3	1+3	1+3	1+3	1+3	1+3	1+3	1+3	1+3	1+3	1+3	1+3
Load 2+3	2+3	2+3	2+3	2+3	2+3	2+3	2+3	2+3	2+3	2+3	2+3
Load 1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3	1+2+3

จากผลการทดสอบสามารถสรุปได้ว่าระบบการจำแนกชนิดโหลดสามารถระบุสถานะของโหลดได้ถูกต้อง 100%

2.4.6 Non-intrusive Load Monitoring based on Convolution Neural Network with Differential Input [10]

งานวิจัยนี้ได้พัฒนาระบบในการแยกปริมาณโหลดไฟฟ้าจากโหลดรวม โดยอาศัยเทคนิคอัลกอริทึม Convolution Neural Network (CNN) ข้อมูลที่ใช้ในการป้อนเข้าอัลกอริทึม CNN ได้ใช้ผลต่างของกำลังไฟฟ้าจริง (ΔP) โดยโครงสร้างของอัลกอริทึม Convolution Neural Network แสดงดังรูปที่ 2.22



รูปที่ 2.23 โครงสร้างของอัลกอริทึม Convolution Neural Network

Mean square error และ Mean absolute error ได้ใช้ในการประเมินผลความแม่นยำของอัลกอริทึมที่นำเสนอ โดยได้ทำการทดลองโดยใช้ชุดข้อมูล Reference Energy Disaggregation Dataset และทดสอบเทียบกับโมเดล seq2point, seq2seq และ GLU-res ที่ได้มีการพัฒนามาก่อนหน้านี้ ซึ่งผลการทดสอบแสดงดังตารางที่ 2.3

ตารางที่ 2.3 ผลการทดสอบของอัลกอริทึมที่นำเสนอเทียบกับอัลกอริทึมที่ได้พัฒนามาก่อน

Metrics	Models	Fridge	Microwave	Dishwasher
MSE	seq2point [6]	2393.9	17483.5	15891.3
	seq2seq [6]	2151.9	19292.8	14172.6
	GLU-Res [8]	2197.4	25202.0	22301.1
	Proposed Model (no norm)	1622.8	17037.9	18658.5
	Proposed Model (norm)	1867.1	19195.6	15489.2
MAE	seq2point [6]	26.17	20.51	23.73
	seq2seq [6]	28.15	27.87	24.45
	GLU-Res [8]	23.52	28.41	33.37
	Proposed Model (no norm)	21.76	18.32	22.32
	Proposed Model (norm)	25.41	20.81	20.12

จากผลการทดสอบจะเห็นได้ว่าอัลกอริทึมที่นำเสนอมีความแม่นยำสูงกว่าเมื่อประเมินจาก MSE และ MAE ยกเว้นในกรณีแยกโหลดเครื่องล้างจานโดยประเมินจาก MSE และนอกจากนี้งานวิจัยยังชี้ให้เห็นว่าอัลกอริทึมที่นำเสนอมีขนาดเล็กกว่าอัลกอริทึมที่พัฒนามาก่อนหน้าเป็นอย่างมาก

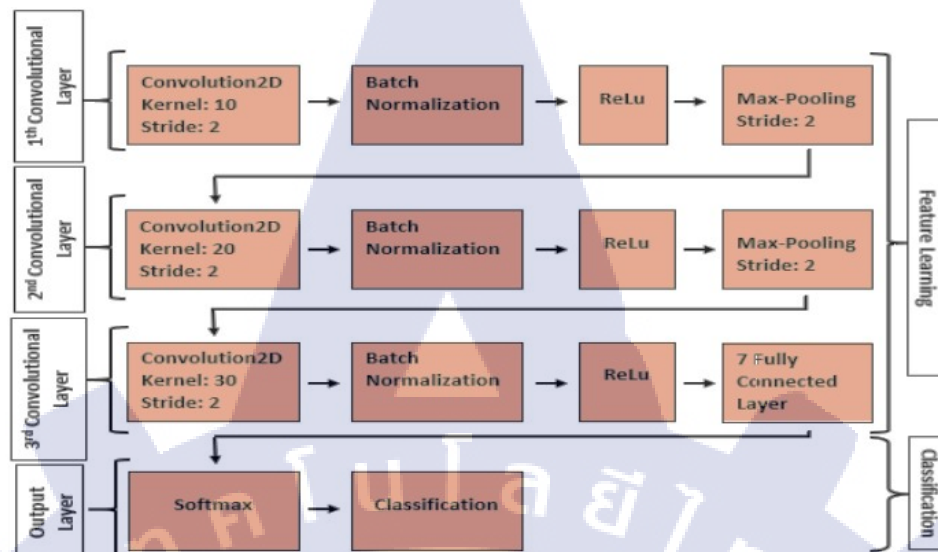
2.4.7 Home Appliance Identification for NILM System Based on Deep Neural Network [11]

ระบบสำหรับแยกปริมาณโหลดไฟฟ้าจากโหลดรวมได้ถูกพัฒนา โดยอาศัยเทคนิคอัลกอริทึม Convolution Neural Network (CNN) การเปลี่ยนแปลงของกำลังไฟฟ้าชั่วขณะจะถูกตรวจจับและนำไปใช้ในการแยกปริมาณโหลดไฟฟ้า ในการสอนและทดสอบระบบจะใช้ชุดข้อมูล Reference Energy Disaggregation Dataset (REDD) โดยได้ทำการเลือกอุปกรณ์ไฟฟ้า 6 เครื่องดังแสดงในตาราง 2.4

ตารางที่ 2.4 ชุดข้อมูลอุปกรณ์ไฟฟ้าที่ใช้ในการสอนและทดสอบระบบ

N°	Equipment	Trai	Valid	Test	Total
1	Refrigerator	60	20	20	100
2	Microwave	60	20	20	100
3	Oven	60	20	20	100
4	Dishwasher	60	20	20	100
5	Air conditioning	60	20	20	100
6	Washer / Dryer	60	20	20	100
#	Overall	360	120	120	600

โดยโครงสร้างของอัลกอริทึม Convolution Neural Network แสดงดังรูปที่ 2.23



รูปที่ 2.24 โครงสร้างของอัลกอริทึม Convolution Neural Network

ในการประเมินผลความแม่นยำได้ประเมินใน 4 ส่วน คือ 1) ความแม่นยำ (Prec), 2) ความถูกต้อง (Acc), 3) Sensitivity (Sens) และ 4) F-measure (f) รวมทั้งทำการเปรียบเทียบกับโมเดลที่ได้พัฒนามาแล้วก่อนหน้านี้ โดยผลการทดสอบแสดงดังตารางที่ 2.5

ตารางที่ 2.5 ผลการทดสอบโมเดลการแยกปริมาณโหลดไฟฟ้าเทียบกับโมเดลก่อนหน้านี้

Authors	Technique	N° of Appliance categories	Sens	Prec	F	Acc
This Study	CNN	6	0.933	0.936	0.930	0.93
Deyvison [29]	CNN	7	0.82	0.84	0.82	0.82
Kelly [7]	Autoencoder	5	0.80	0.58	0.55	0.91
Kelly [7]	LSTM	5	0.69	0.39	0.39	0.68
WONG [16]	PDT	6	0.77	0.76	0.73	----
Zhao [17]	GSP	8	0.51	0.89	0.64	0.77

[7] Uses long short-term memory; [16] Uses Particle-based Distribution Truncation (PDT) and [17] Uses Graph Signal Processing (GSP).

ผลการทดสอบแสดงให้เห็นว่าโมเดลการแยกปริมาณโหลดไฟฟ้าที่ได้นำเสนอในงานวิจัยนี้มีประสิทธิภาพสูงกว่าของเก่าที่นำเสนอมา

2.4.8 Neural NILM: Deep Neural Network Applied to Energy Disaggregation

[12]

งานวิจัยนี้ได้พัฒนาโปรแกรมในการแยกปริมาณโหลดไฟฟ้าจากโหลดรวม โครงสร้างของโปรแกรมจะประกอบไปด้วย 4 ส่วน คือ การรวบรวมข้อมูล (Data acquisition) การตรวจจับเหตุการณ์ (Event detection) การคัดกรองคุณลักษณะ (Feature extraction) และการคัดแยกชนิดโหลดทางไฟฟ้า (Classification)

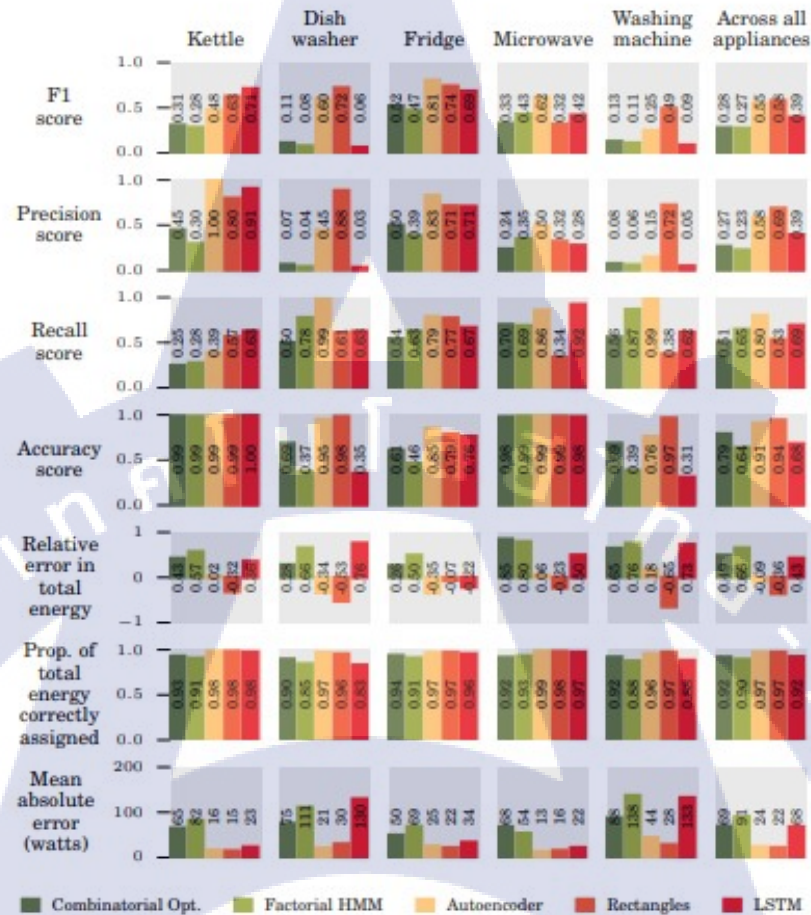
2.4.8.1 การรวบรวมข้อมูล (Data acquisition) ในงานวิจัยนี้ได้ใช้ชุดข้อมูล UK-DALE ในการสอนและทดสอบชุดโปรแกรมการแยกปริมาณโหลดไฟฟ้าจากโหลดรวม

2.4.8.2 การตรวจจับเหตุการณ์ (Event detection) ในการตรวจจับเหตุการณ์ได้ใช้เทคนิค Sliding window ในการสแกนข้อมูลเป็นช่วงๆ

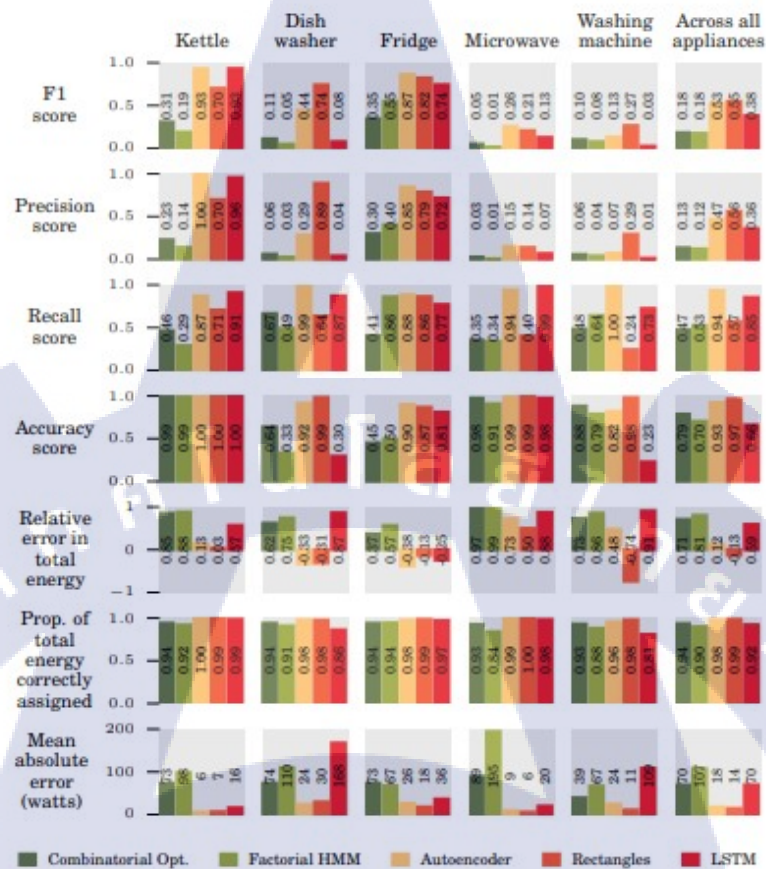
2.4.8.3 การคัดกรองคุณลักษณะ (Feature extraction) ใช้ชุดคำสั่งของ Library ของ NILMTK's ในโปรแกรม Python เพื่อคัดกรองคุณลักษณะ

2.4.8.4 การคัดแยกชนิดโหลดทางไฟฟ้า (Classification) ในงานวิจัยนี้ได้เสนออัลกอริทึมเพื่อใช้ในการคัดแยกชนิดโหลดทางไฟฟ้า 3 อัลกอริทึม คือ Long short-term memory (LSTM), Denoising Autoencoder และ Regress Start time, End time & Power

ในการประเมินประสิทธิภาพของโปรแกรมได้ทำการประเมินใน 7 ด้าน คือ Recall, Precision, F1, Accuracy, Relative error in total energy, Mean absolute error และ Proportion of total energy correctly assigned โดยในงานวิจัยนี้ได้ทำการทดสอบกับชุดข้อมูล UK-DALE ของโหลดไฟฟ้ารวมที่อยู่อาศัยที่รู้ปริมาณและชนิดโหลดที่อยู่ภายในและที่อยู่อาศัยที่ไม่รู้ปริมาณและชนิดโหลดที่อยู่ภายใน โดยผลการทดสอบแสดงดังรูปที่ 2.24 และ 2.25 สำหรับโหลดไฟฟ้ารวมที่อยู่อาศัยที่รู้ปริมาณและชนิดโหลดที่อยู่ภายในและที่อยู่อาศัยที่ไม่รู้ปริมาณและชนิดโหลดที่อยู่ภายในตามลำดับ



รูปที่ 2.25 ผลการทดสอบการแยกโหลดไฟฟ้าโหลดไฟฟ้ารวมที่อยู่อาศัยที่รู้ปริมาณและชนิดโหลดที่อยู่ภายใน

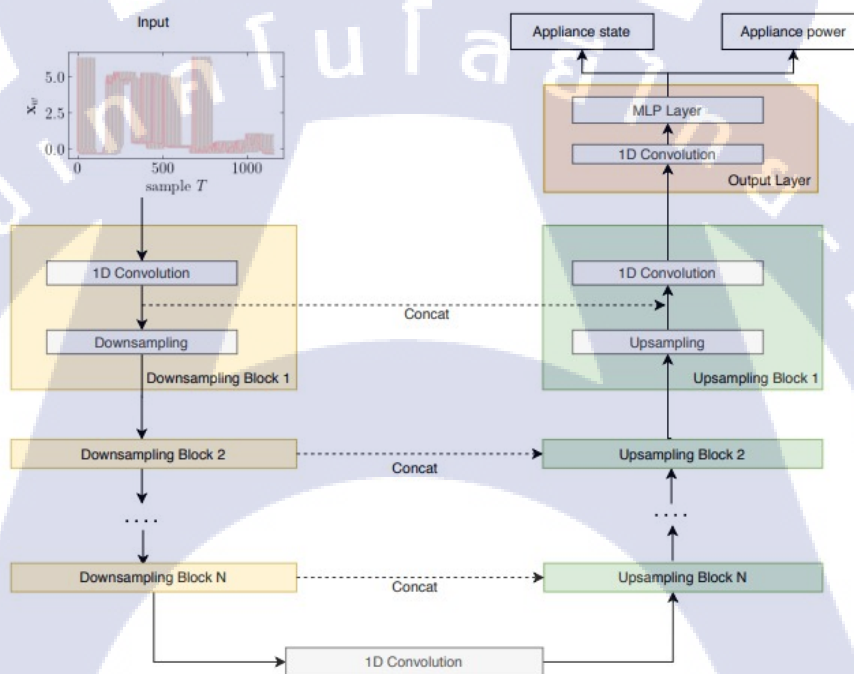


รูปที่ 2.26 ผลการทดสอบการแยกโหลดไฟฟ้าโหลดไฟฟ้ารวมที่อยู่อาศัยที่ไม่รู้ปริมาณและชนิดโหลดที่อยู่ภายใน

จากผลการทดสอบพบว่าอัลกอริทึม Denoising Autoencoder และ Regress Start time, End time & Power มีประสิทธิภาพดีกว่าอัลกอริทึมที่พัฒนามาก่อนหน้าอย่าง Combinatorial optimization และ Factorial hidden markov model ในเกือบทุกด้าน ส่วนอัลกอริทึม Long short-term memory ก็มีประสิทธิภาพเหนือกว่าอัลกอริทึม อย่าง Combinatorial optimization และ Factorial hidden markov model ในการแยกอุปกรณ์ไฟฟ้าแบบ On/Off แต่จะด้อยกว่าในการแยกอุปกรณ์ไฟฟ้าแบบ Multi-state

2.4.9 Unet-NILM: A Deep Neural Network for Multi-task Appliances State Detection and Power Estimation in NILM [13]

งานวิจัยนี้ได้พัฒนาระบบในการแยกปริมาณโหลดไฟฟ้าจากโหลรวม โดยอาศัยเทคนิค Multi-target quantile regression for power estimation และ Multi-label learning for state estimation แทนอัลกอริทึม Convolution Neural Network (CNN) แบบเดิม ข้อมูลที่ใช้ในการป้อนเข้าอัลกอริทึมได้ใช้กำลังไฟฟ้าจริง (P) โดยโครงสร้างของอัลกอริทึม Unet-NILM แสดงดังรูปที่ 2.26



รูปที่ 2.27 โครงสร้างของอัลกอริทึม Unet-NILM

Mean absolute error (MAE), Estimated accuracy (EAC) และ Normalized disaggregation error (NDE) ได้ใช้ในการประเมินผลความแม่นยำของอัลกอริทึมที่นำเสนอ โดยได้ทำการทดลองโดยใช้ชุดข้อมูล UK-DALE Dataset และทดสอบเทียบกับโมเดล 1D Convolution Neural Network (CNN) ที่ได้มีการพัฒนาก่อนหน้านี้ ซึ่งผลการทดสอบแสดงดังตารางที่ 2.6

ตารางที่ 2.6 ผลการทดสอบโมเดลการแยกปริมาณโหลดไฟฟ้าเทียบกับโมเดลก่อนหน้า

Appliance	EAC		MAE		NDE		exb - F ₁	
	1D-CNN	UNet-NILM	1D-CNN	UNet-NILM	1D-CNN	UNet-NILM	1D-CNN	UNet-NILM
KT	0.589 ± 0.003	0.677 ± 0.017	20.390 ± 0.169	16.003 ± 0.824	0.674 ± 0.010	0.429 ± 0.039	0.944	0.956
FRZ	0.923 ± 0.000	0.937 ± 0.000	18.583 ± 0.006	15.124 ± 0.014	0.073 ± 0.000	0.072 ± 0.000	0.964	0.962
DW	0.875 ± 0.000	0.914 ± 0.000	9.884 ± 0.012	6.764 ± 0.012	0.126 ± 0.000	0.080 ± 0.000	0.913	0.909
WM	0.875 ± 0.000	0.909 ± 0.000	15.758 ± 0.009	11.506 ± 0.006	0.111 ± 0.000	0.062 ± 0.000	0.954	0.963
MW	0.630 ± 0.002	0.753 ± 0.003	9.690 ± 0.055	6.475 ± 0.072	0.656 ± 0.007	0.334 ± 0.005	0.907	0.916
Average	0.778 ± 0.003	0.838 ± 0.004	14.86 ± 0.050	11.174 ± 0.186	0.328 ± 0.003	0.195 ± 0.009	0.937	0.941

ตารางที่ 2.7 สรุปผลงานวิจัยที่เกี่ยวข้อง

ที่	คณะผู้วิจัย (ปี)	หัวข้อวิจัย	วิธีวิจัย	ผลที่ได้รับ
1	Q. Liu et al.[5]	Low-Complexity Non-Intrusive Load Monitoring Using Unsupervised Learning and Generalized Appliance Models	งานวิจัยได้เลือกกำลังไฟฟ้าจริง (P) เป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม Entropy Index Constraints Competitive Agglomeration (EICCA) ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูล REDD และเลือกตัวชี้วัด Total energy correctly assigned (TECA)	Total energy correctly assigned (TECA) จากการทดลองมากกว่า 90%
2	S. Biansoongnern and B. Plangklang[6]	Nonintrusive Load Monitoring (NILM) Using an Artificial Neural Network in Embedded System with Low Sampling Rate	งานวิจัยได้เลือกกำลังไฟฟ้าจริง (P) เป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม Artificial neural network (ANN) ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบจากข้อมูลที่วัดด้วยมิเตอร์และเลือกตัวชี้วัดเป็นค่าความถูกต้องเทียบกับมิเตอร์	ความถูกต้อง 95%
3	K. S. Barsim et al.[7]	An Approach for Unsupervised Non-Intrusive Load Monitoring of Residential Appliances	งานวิจัยได้เลือกกำลังไฟฟ้าจริงและเสมือน (P,Q) เป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม Mean-shift ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูล BLUEED และเลือกตัวชี้วัดเป็น TPP	TPP = 95% สำหรับจำนวนตัวอย่าง 886 TPP = 70.5% สำหรับจำนวนตัวอย่าง 1579

ตารางที่ 2.7 สรุปผลงานวิจัยที่เกี่ยวข้อง(ต่อ)

ที่	คณะผู้วิจัย (ปี)	หัวข้อวิจัย	วิธีวิจัย	ผลที่ได้รับ
4	K. Chahine et al.[8]	Electric Load Disaggregation in Smart Metering Using a Novel Feature Extraction Method and Supervised Classification	งานวิจัยได้เลือกกระแสไฟฟ้า (A) เป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม k-Nearest Neighbor, Naïve Bayes และ Support Vector Machine ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูลจริงที่เก็บจากมิเตอร์อัจฉริยะกับอุปกรณ์ไฟฟ้า 9 เครื่อง และเลือกตัวชี้วัดเป็น Overall Accuracy	Overall Acc ของ Naïve Bayes =94%, Overall Acc ของ k-Nearest Neighbor =99.22%, Overall Acc ของ Support Vector Machine =96.11%,
5	H. H. Chang et al.[9]	A Non-Intrusive Load Monitoring System Using an Embedded System for Applications to Unbalanced Residential Distribution System	งานวิจัยได้เลือกกำลังไฟฟ้าจริง (P) และกำลังไฟฟ้าเสมือน (Q) เป็นคุณลักษณะที่ติดตามและใช้ Particle swarm optimization (PSO) ร่วมกับอัลกอริทึม Artificial neural network (ANN) ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบจากข้อมูลที่วัดด้วยไมโครคอนโทรลเลอร์ที่ทำขึ้นเอง และเลือกตัวชี้วัดเป็น Accuracy ของสถานะการทำงานอุปกรณ์ไฟฟ้า	ความถูกต้อง 100%

ตารางที่ 2.7 สรุปผลงานวิจัยที่เกี่ยวข้อง(ต่อ)

ที่	คณะผู้วิจัย (ปี)	หัวข้อวิจัย	วิธีวิจัย	ผลที่ได้รับ
6	Y. Zhang et al.[10]	Non-intrusive Load Monitoring based on Convolution Neural Network with Differential Input	งานวิจัยได้เลือกผลต่างกำลังไฟฟ้าจริง (ΔP) เป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม Convolution Neural Network (CNN) ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูล BLUED และเลือกตัวชี้วัดเป็น MSE และ MAE โดยทำการเทียบผลกับอัลกอริทึมก่อนหน้า	อัลกอริทึมที่นำเสนอให้ค่า MSE และ MAE ดีกว่าอัลกอริทึมก่อนหน้า
7	D. D. P. Penha and A. R. G. Castro[11]	Home Appliance Identification for NILM System Based on Deep Neural Network	งานวิจัยได้เลือกการเปลี่ยนแปลงกำลังไฟฟ้าจริงชั่วขณะเป็นคุณลักษณะที่ติดตามและใช้อัลกอริทึม Convolution Neural Network (CNN) ในการแบ่งข้อมูลและสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูล REDD และเลือกตัวชี้วัด ความแม่นยำ (Prec), ความถูกต้อง (Acc), Sensitivity (Sens) และ F-measure (f) โดยทำการเทียบผลกับอัลกอริทึมก่อนหน้า	อัลกอริทึมที่นำเสนอให้ค่า ความแม่นยำ (Prec), ความถูกต้อง (Acc), Sensitivity (Sens) และ F-measure (f) ดีกว่าอัลกอริทึมก่อนหน้า

ตารางที่ 2.7 สรุปผลงานวิจัยที่เกี่ยวข้อง(ต่อ)

ที่	คณะผู้วิจัย (ปี)	หัวข้อวิจัย	วิธีวิจัย	ผลที่ได้รับ
8	J. Kelly and W. Knottenbelt[12]	Neural NILM: Deep Neural Network Applied to Energy Disaggregation	งานวิจัยได้เลือกกำลังไฟฟ้าจริง (P) เป็นค่าติดตาม และใช้อัลกอริทึม Long short-term memory (LSTM), Denoising Autoencoder และ Regress Start time, End time & Power ในการแบ่งข้อมูล และสร้างโมเดล และทำการทดสอบด้วยชุดข้อมูล UK-DALE และเลือกตัวชี้วัดใน 7 ด้าน โดยทำการเทียบกับอัลกอริทึม Combinatorial optimization และ Factorial hidden markov model	อัลกอริทึมที่นำเสนอ ให้ผลการทดสอบทั้ง 7 ด้านดีกว่าอัลกอริทึม Combinatorial optimization และ Factorial hidden markov model
9	Anthony Faustine et al.[11]	Unet-NILM: A Deep Neural Network for Multi-task Appliances State Detection and Power Estimation in NILM	งานวิจัยได้เลือกกำลังไฟฟ้าจริง (P) เป็นค่าที่ติดตาม และใช้อัลกอริทึม Multi-target quantile regression for power estimation และ Multi-label learning for state estimation ในการแบ่งข้อมูลและสร้างโมเดล ทำการทดสอบด้วยข้อมูล UK-DALE และเลือกตัวชี้วัด Mean absolute error (MAE), Estimated accuracy (EAC) และ Normalized disaggregation error (NDE) เทียบกับ 1D Convolution Neural Network (CNN)	อัลกอริทึมให้ค่า Mean absolute error (MAE), Estimated accuracy (EAC) และ Normalized disaggregation error (NDE) ดีกว่าอัลกอริทึม Convolution Neural Network (CNN)



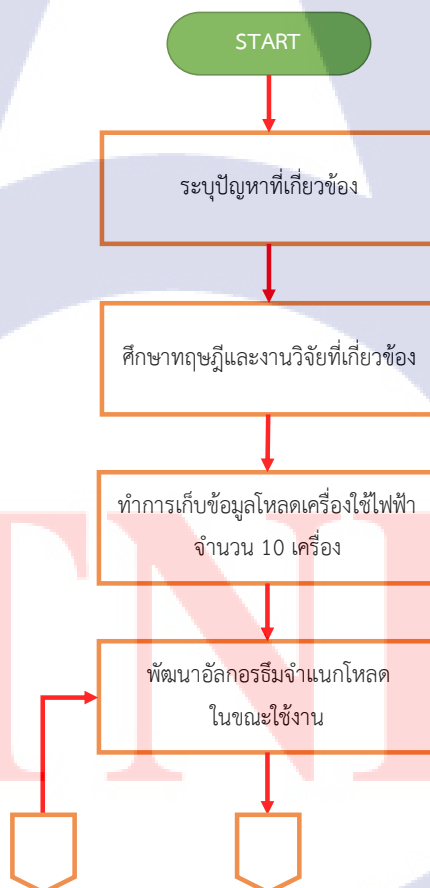
บทที่ 3

ระเบียบวิธีการวิจัย

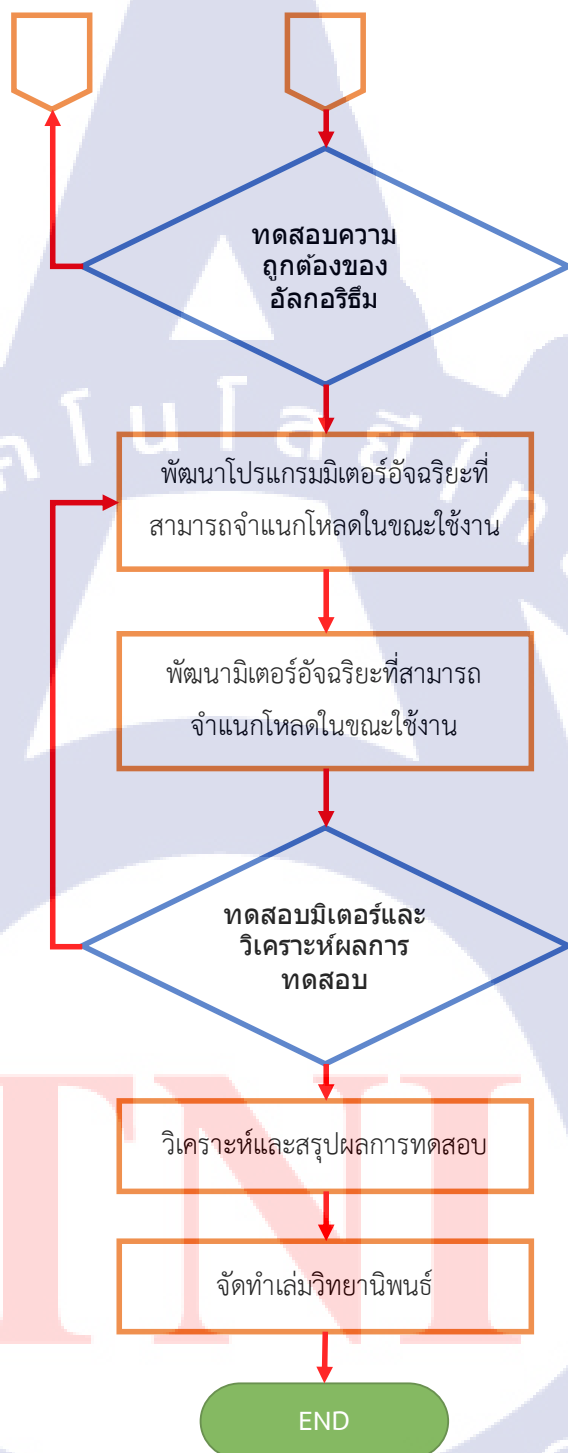
เนื้อหาในบทนี้กล่าวถึงระเบียบวิธีวิจัยเกี่ยวกับการออกแบบและพัฒนาไมโครอัจฉริยะแบบสามารถแยกโหนดในขณะใช้งานสำหรับที่อยู่อาศัย ได้แก่ กรอบแนวคิดการวิจัย ขั้นตอนการทดลอง การวิจัยและเก็บข้อมูล และอื่นๆ

3.1 กรอบแนวคิดการวิจัย

การออกแบบและพัฒนาไมโครอัจฉริยะแบบสามารถแยกโหนดในขณะใช้งานสำหรับที่อยู่อาศัย จะมีขั้นตอนการวิจัยแสดงดังรูปที่ 3.1



รูปที่ 3.1 กรอบแนวความคิดงานวิจัยการออกแบบและพัฒนาไมโครอัจฉริยะแบบสามารถแยกโหนดในขณะใช้งานสำหรับที่อยู่อาศัย



รูปที่ 3.1 กรอบแนวความคิดงานวิจัยการออกแบบและพัฒนามิเตอร์อัจฉริยะแบบสามารถแยก
โหลดในขณะใช้งานสำหรับที่อยู่อาศัย(ต่อ)

3.2 ขั้นตอนการทดลอง

ในการวิจัยออกแบบและพัฒนาไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย จะมีขั้นตอนการดำเนินการ 6 ขั้นตอนคือ

- 3.2.1 ทำการเก็บข้อมูลโหลดเครื่องใช้ไฟฟ้า
- 3.2.2 พัฒนาอัลกอริทึมในการแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย
- 3.2.3 ทำการกำหนดวิธีทดสอบ ตัวชี้วัดและเกณฑ์การทดสอบ
- 3.2.4 จัดทำระบบไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย
- 3.2.5 ทำการทดสอบระบบและวิเคราะห์ผลการทดสอบ
- 3.2.6 สรุปผลการทดสอบ

3.3 การดำเนินการวิจัยและเก็บข้อมูล

ในการดำเนินการวิจัยและเก็บข้อมูลจะแบ่งเป็น 2 ส่วนหลักคือ

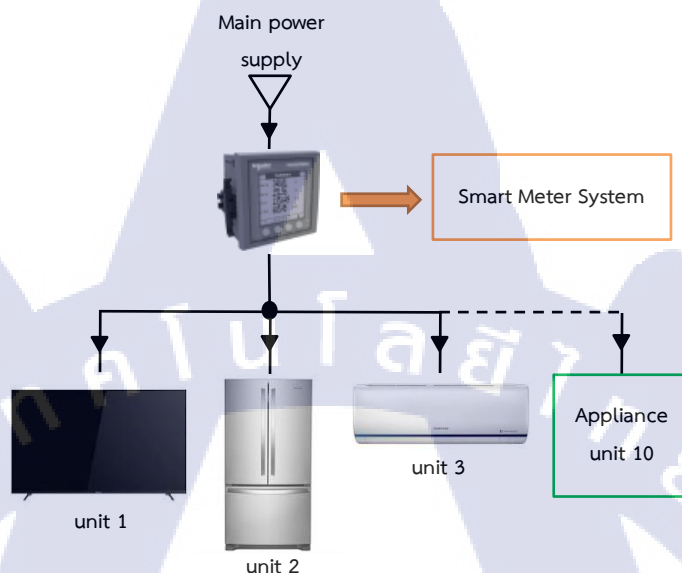
- 3.3.1 การเก็บข้อมูลสภาวะโหลดเครื่องใช้ไฟฟ้าเพื่อใช้ในการสร้างและทดสอบอัลกอริทึมการแยกโหลดเครื่องใช้ไฟฟ้าในขณะใช้งานสำหรับที่อยู่อาศัย
- 3.3.2 การเก็บและวิเคราะห์ข้อมูลผลการทดสอบของไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

3.4 การทำการทดลอง

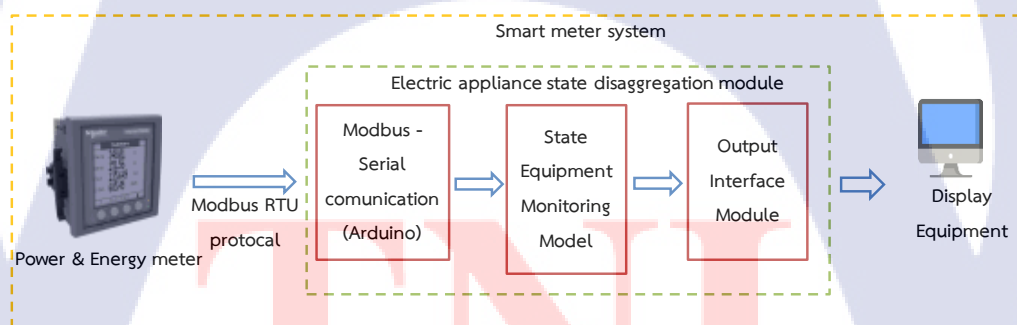
โดยรายละเอียดในการทดลองมีดังนี้

- 3.4.1 ทำการเก็บข้อมูลโหลดเครื่องใช้ไฟฟ้า โดยทำการเก็บข้อมูลการทำงานของเครื่องใช้ไฟฟ้าที่กำลังไฟฟ้าที่กำหนดจำนวน 10 เครื่อง โดยเก็บค่ากำลังไฟฟ้าจริง (Real power) และกำลังไฟฟ้าปรากฏ (Apparent power) แต่ละเครื่อง โดยเก็บแบบละ 25 นาที
- 3.4.2 นำข้อมูลมาสร้างแบบจำลองการแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยโดยใช้การเรียนรู้ของเครื่องจักรแบบ Multi-layers perceptron neural network
- 3.4.3 ทำการทดสอบความถูกต้องแม่นยำของแบบจำลองการแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยในข้อที่ 2 ด้วยวิธี Confusion matrix และมาตรวัดความถูกต้อง (Acc)
- 3.4.4 ทำการพัฒนาโปรแกรมในการอ่านค่ากำลังไฟฟ้าจริง (Real power) และกำลังไฟฟ้าปรากฏ (Apparent power) จากมิเตอร์ดิจิตอลและส่งข้อมูลไปจำแนกโหลดโดยแบบจำลองการแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยในข้อที่ 2 และทำการแสดงผล โดยในงานวิจัยนี้จะใช้โปรแกรม Python ในการดำเนินงาน

3.4.5 พัฒนามิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยตามแสดงตำแหน่งติดตั้งระบบและแผนผังระบบตามรูปที่ 3.2 และรูปที่ 3.3 ตามลำดับ



รูปที่ 3.2 ตำแหน่งติดตั้งมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน



รูปที่ 3.3 แผนผังระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

3.4.6 ทำการทดสอบระบบและวิเคราะห์ผลการทดสอบ โดยต้องมีความถูกต้อง (Acc) ในการตรวจรู้สภาวะการทำงานของโหลดเครื่องใช้ไฟฟ้า 100 %

3.4.7 สรุปผลการทดสอบ ทำการสรุปผลเทียบกับงานวิจัยอื่นที่เกี่ยวข้อง

บทที่ 4

ผลการวิจัย

เนื้อหาในบทนี้กล่าวถึงผลลัพธ์และการสรุปผลเบื้องต้นของงานวิจัยการออกแบบและพัฒนามิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยโดยแบ่งออกเป็นส่วนตัวๆ ได้แก่ อัลกอริธึมสำหรับจำแนกโหลดในขณะใช้งาน ส่วนโปรแกรมของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย ส่วนประกอบฮาร์ดแวร์ของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย การทดสอบและสรุปผลระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย และอื่นๆ

4.1 ส่วนประกอบฮาร์ดแวร์ของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

สำหรับส่วนประกอบฮาร์ดแวร์ของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยจะประกอบด้วย 3 ส่วนประกอบหลัก คือ

4.1.1 ระบบมิเตอร์ไฟฟ้าแบบดิจิตอล ในงานวิจัยนี้ได้ใช้มิเตอร์ไฟฟ้าแบบดิจิตอลยี่ห้อ Schneider รุ่น EasyLogic PM2230 แสดงดังรูปที่ 4.1 ในการวัดค่ากำลังจริง (P) และกำลังปรากฏ (S) จากระบบสายเมนของวงจรไฟฟ้าภายในที่อยู่อาศัย โดยต่อรับค่าแรงดันเข้ากับระบบโดยตรงและรับค่ากระแสผ่านหม้อแปลงวัดกระแส (CT) ขนาด 50/5A ยี่ห้อ QUBIX รุ่น MSQ-30 แสดงดังรูปที่

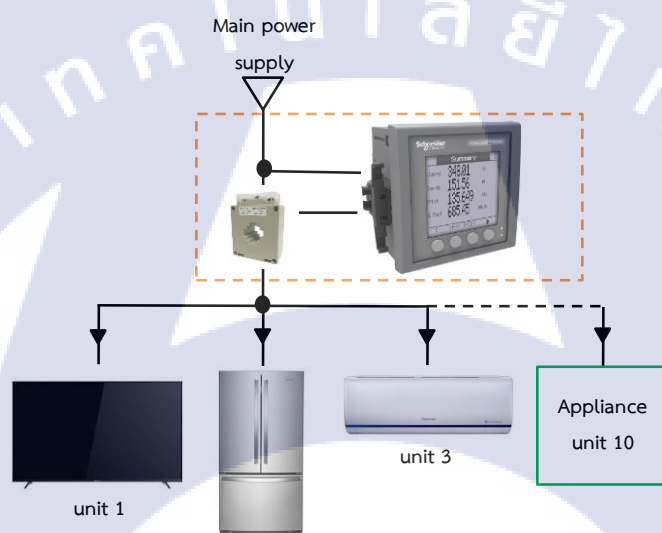
4.2 วงจรการต่อระบบมิเตอร์ไฟฟ้าแบบดิจิตอลแสดงดังภาพ 4.3



รูปที่ 4.1 มิเตอร์ไฟฟ้าแบบดิจิตอลยี่ห้อ Schneider รุ่น EasyLogic PM2230



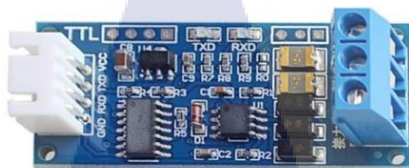
รูปที่ 4.2 หม้อแปลงกระแสขนาด 50/5A ยี่ห้อ QUBIX รุ่น MSQ-30



รูปที่ 4.3 วงจรการต่อระบบมิเตอร์ไฟฟ้าแบบดิจิทัล

4.1.2 ระบบส่งข้อมูลจากมิเตอร์มายังคอมพิวเตอร์ การส่งข้อมูลจากมิเตอร์ไปยังคอมพิวเตอร์เพื่อทำการจำแนกโหลดนั้น จะส่งผ่าน Modbus RTU protocol ไปยังชุดอุปกรณ์แปลงจาก Modbus RTU เป็น Serial และส่งข้อมูลแบบ Serial ไปยังคอมพิวเตอร์ สำหรับชุดอุปกรณ์แปลงจาก Modbus RTU เป็น Serial จะประกอบด้วย 2 ส่วนคือ

- บอร์ดแปลงสัญญาณ RS 485 เป็น TTL (Serial) ตามภาพ 4.4 ซึ่งทำหน้าที่แปลงสัญญาณ RS 485 (Recommended Standard no. 485) เป็นสัญญาณอนุกรม TTL (Transistor-transistor logic)

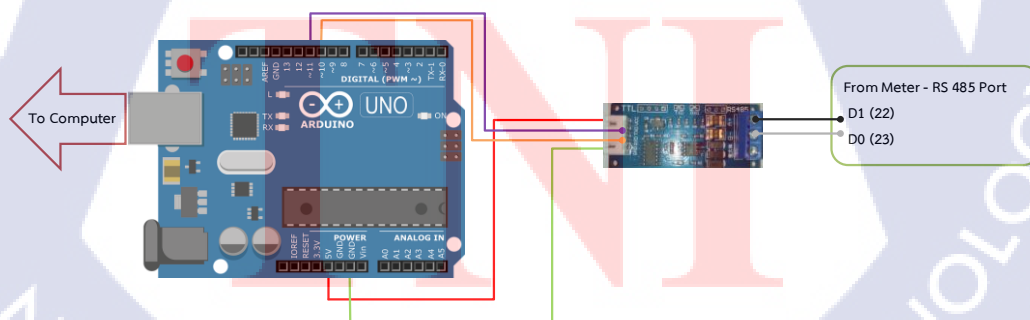


รูปที่ 4.4 บอร์ดแปลงสัญญาณ RS 485 เป็น TTL (Serial)

- บอร์ด Arduino รุ่น Uno ดังรูปที่ 4.5 โดยได้ดำเนินการเขียนโค้ดเพื่อให้บอร์ดทำการอ่านค่าที่ต้องการ (ค่า P และ S) จากมิเตอร์และทำการส่งค่าไปยังคอมพิวเตอร์ต่อไป โดยส่วนของโค้ดดังกล่าวจะอธิบายในหัวข้อ 4.3 วงจรของชุดส่งข้อมูลแสดงดังรูปที่ 4.6



รูปที่ 4.5 บอร์ด Arduino รุ่น Uno



รูปที่ 4.6 วงจรของชุดส่งข้อมูล

4.1.3 คอมพิวเตอร์ ข้อมูลจากมิเตอร์จะถูกส่งมาที่คอมพิวเตอร์เพื่อทำการแยกโพลต์และแสดงผล โดยคอมพิวเตอร์จะถูกโปรแกรมให้ทำการอ่านค่า แยกชนิดของโพลต์และแสดงผล บนพื้นฐานโปรแกรมภาษา Python ซึ่งจะกล่าวต่อในหัวข้อ 4.3

4.2 อัลกอริธึมสำหรับจำแนกโพลต์ในขณะใช้งาน

การพัฒนาไมเตอร์อัลกอริธึมสำหรับจำแนกโพลต์ในขณะใช้งานสำหรับที่อยู่อาศัย จะดำเนินการโดยการเก็บค่ากำลังจริง (P) และกำลังปรากฏ (S) โดยใช้อุปกรณ์ Power quality analyzer ยี่ห้อ Kyoritsu รุ่น KEW 6315 แสดงตามรูปที่ 4.7 เพื่อเก็บค่าสำหรับอุปกรณ์ไฟฟ้าที่จะนำมาใช้ในการทดสอบจำนวน 10 รายการ ตามแสดงในตารางที่ 4.1 โดยใช้ความถี่ในการเก็บค่าทุก 1 วินาที เป็นเวลา 25 นาที ในแต่ละอุปกรณ์



รูปที่ 4.7 อุปกรณ์ Power quality analyzer ยี่ห้อ Kyoritsu

ตารางที่ 4.1 รายการอุปกรณ์ไฟฟ้าที่จะนำมาใช้ในการพัฒนาอัลกอริทึมสำหรับจำแนกโหลดในขณะใช้งาน

No.	Appliance Name	Brand/Model	Rating (kW)	Voltage/Frequency
1	Hair dryer	Panasonic/ EH-NE72	1700 W	220Vac/50-60 Hz
2	LED Flood light	LED/-	50 W	85-265 Vac/50/60Hz
3	Blender	Philips/ HR2872	350 W	200-240 Vac/ 50-60 Hz
4	Fan	Hatari/ HT-S16M7	50 W	220 Vac/ 50 Hz
5	Toaster	Misawa /YT-2002	750 W	220 Vac/ 50 Hz
6	Electric kettle 1L	Seagull/ 150000712	1630 W	240 Vac/ 50/60Hz
7	Microwave oven	Sharp/R-220	1250 W	220 Vac/ 50 Hz
8	Refrigerator	Toshiba/ GR-A906Z,ZK	58 W/+15W	220 Vac/ 50 Hz
9	Television LCD 68"	Samsung/UA65RU7100	160 W	100-240 Vac/ 50/60Hz
10	Steam Iron	Philips/ GC2990	2000-2400 W	200-240 Vac/ 50-60 Hz

จากข้อมูลที่เก็บได้จากอุปกรณ์ไฟฟ้าทั้ง 10 รายการข้างต้น นำข้อมูลดังกล่าวมาสอนผ่านอัลกอริทึมสำหรับจำแนกโหลดในขณะใช้งานดังแสดงโค้ดใน Appendix B โดยใช้พื้นฐานจาก Multi-layer perceptron neural network และทำการทดสอบโดยวิธี Confusion matrix ซึ่งแสดงผลการทดสอบดังรูปที่ 4.8 โดยได้ผลการทดสอบความถูกต้อง (Acc) ที่ 1.0 สำหรับงานวิจัยนี้จะสร้างอัลกอริทึมผ่านโปรแกรม Python

```

Enter your file path: C:/Users/Admin/Desktop/Thesis_ruttagorn/Data/Raw_dataM1.csv
[0 1 3 ... 0 0 7]
Score -> 1.000
Accuracy Score: 1.0
[[2464  0  0  0  0  0  0  0  0  0  0]
 [  0 311  0  0  0  0  0  0  0  0  0]
 [  0  0 328  0  0  0  0  0  0  0  0]
 [  0  0  0 1235  0  0  0  0  1  1  0]
 [  0  0  0  0 335  0  0  0  0  0  0]
 [  0  0  0  0  0 441  0  0  0  0  0]
 [  0  0  0  0  0  0 429  0  0  0  0]
 [  0  0  0  0  0  0  0 356  0  0  0]
 [  0  0  0  0  0  0  0  0 539  0  0]
 [  0  0  0  0  0  0  0  0  0 393  0]
 [  0  0  0  0  0  0  0  0  0  0 83]]

```

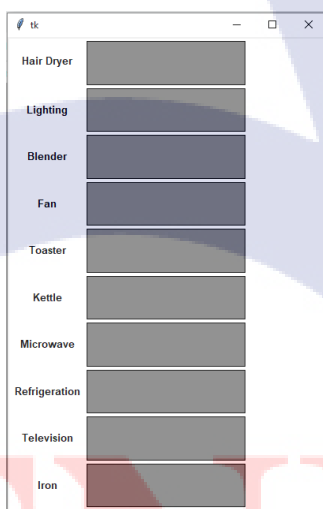
รูปที่ 4.8 ผลการทดสอบอัลกอริทึมสำหรับจำแนกโหลดในขณะใช้งาน

4.3 ส่วนโปรแกรมของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

ในงานวิจัยออกแบบและพัฒนาไมเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย จะมีโปรแกรมที่เกี่ยวข้องอยู่ 2 ส่วนคือ

4.2.1 โปรแกรมสำหรับสื่อสารข้อมูลจากมิเตอร์มายังคอมพิวเตอร์ ในการสื่อสารข้อมูลจากมิเตอร์มายังคอมพิวเตอร์นั้น ใช้บอร์ด Arduino รุ่น Uno ในการส่งข้อมูลตามอธิบายใน 4.1 โดยทำการโปรแกรมบอร์ดดังกล่าวได้แสดงในแสดง Appendix A ซึ่งโปรแกรมนี้นำหน้าที่ในการอ่านค่าที่ต้องการจากมิเตอร์และส่งค่าดังกล่าวไปยังคอมพิวเตอร์

4.2.2 โปรแกรมระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย ในการแยกโหลดของเครื่องใช้ไฟฟ้าและการแสดงผลลัพธ์นั้น จะแสดงโค้ดดังใน Appendix C ซึ่งโปรแกรมห้ทำหน้าที่แยกโหลดของเครื่องใช้ไฟฟ้าในขณะใช้งานและทำการแสดงผลลัพธ์ทางหน้าจอแสดงในรูปที่ 4.9



รูปที่ 4.9 ลักษณะหน้าจอแสดงผลลัพธ์

4.4 การทดสอบและสรุปผลระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัย

ในการทดสอบความแม่นยำของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยนั้น ได้ดำเนินการผ่านชุดทดสอบดังแสดงดังรูปที่ 4.10 โดยทำการทดสอบดังนี้

4.4.1 ทดสอบกับอุปกรณ์ครั้งละ 1 อุปกรณ์ จำนวนอุปกรณ์ละ 30 ครั้ง ให้ครบทั้ง 10 อุปกรณ์

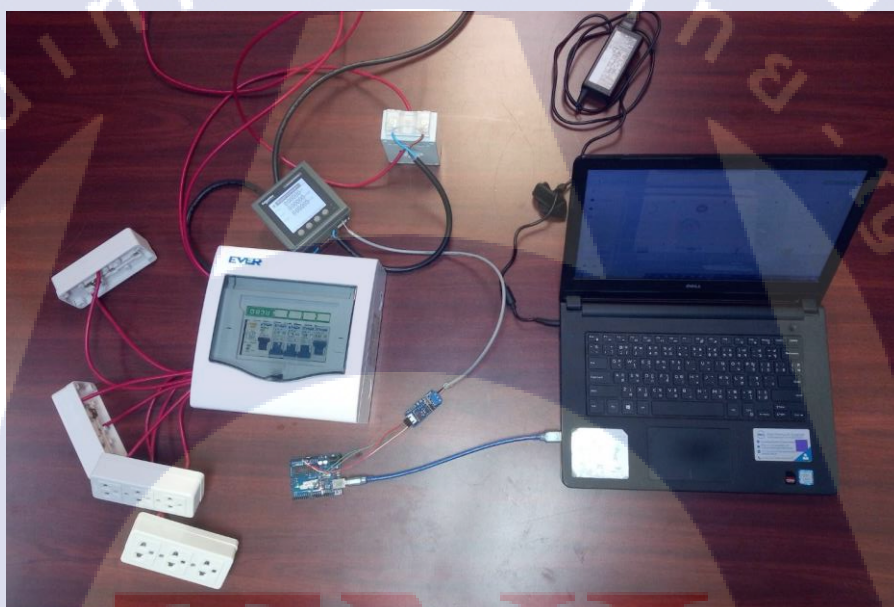
4.4.2 ทดสอบกับอุปกรณ์ทั้ง 10 อุปกรณ์ ใน 1 รอบ โดยทำการเปิดอุปกรณ์เข้าระบบทีละอุปกรณ์จนครบทั้ง 10 อุปกรณ์ ทำจนครบ 30 รอบ

4.4.3 ทำการคำนวณค่าความถูกต้อง (ACC) จากสูตร

$$\begin{aligned} \text{Accuracy Score} &= \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned}$$

ความถูกต้อง (Acc) ที่ได้มีค่าเท่ากับ 1.0

ผลการทดสอบดังข้างต้นแสดงในรูปแบบ Confusion matrix ดังรูปที่ 4.11



รูปที่ 4.10 ชุดทดสอบระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน

1	30	0	0	0	0	0	0	0	0	0	0	0 = No load
2	0	30	0	0	0	0	0	0	0	0	0	1 = Hair dryer
3	0	0	30	0	0	0	0	0	0	0	0	2 = Lighting
4	0	0	0	30	0	0	0	0	0	0	0	3 = Blender
5	0	0	0	0	30	0	0	0	0	0	0	4 = Fan
6	0	0	0	0	0	30	0	0	0	0	0	5 = Toaster
7	0	0	0	0	0	0	30	0	0	0	0	6 = Electric kettle
8	0	0	0	0	0	0	0	30	0	0	0	7 = Microwave
9	0	0	0	0	0	0	0	0	30	0	0	8 = Refrigeration
10	0	0	0	0	0	0	0	0	0	30	0	9 = Television
Total	0	0	0	0	0	0	0	0	0	0	30	10 = Steam iron
	1	2	3	4	5	6	7	8	9	10	Total	Total = 10 equipments

รูปที่ 4.11 ผลการทดสอบความแม่นยำของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งาน

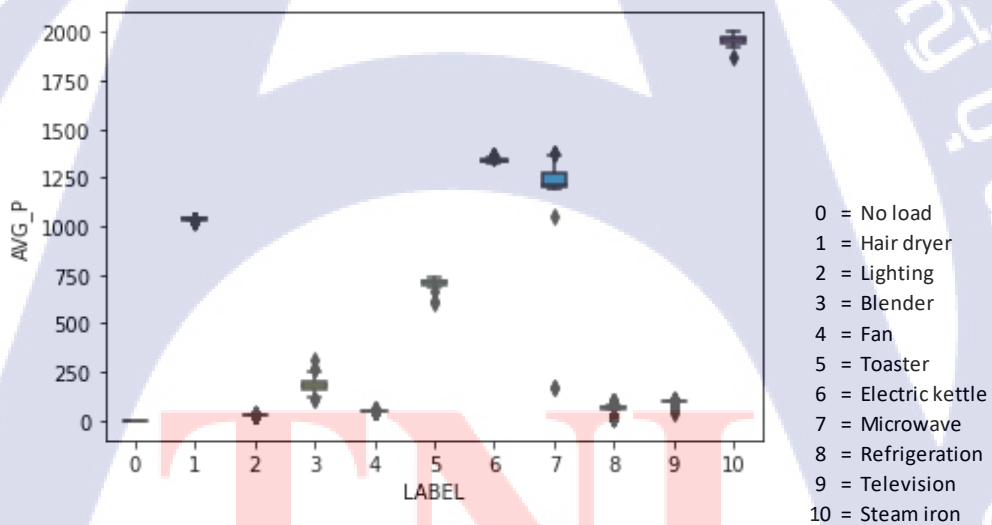


บทที่ 5

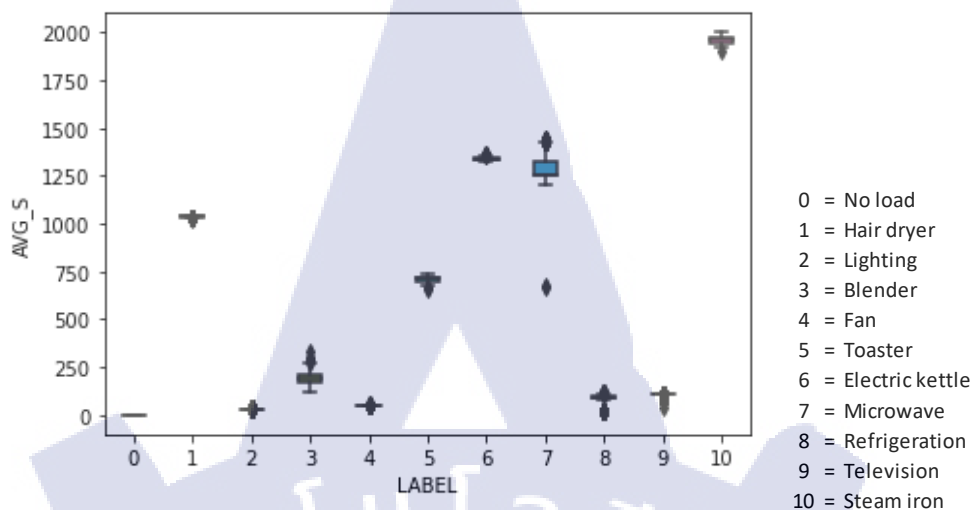
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

เมื่อเทียบกับระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานสำหรับที่อยู่อาศัยที่มีในปัจจุบัน จะพบว่าระบบมิเตอร์อัจฉริยะในงานวิจัยนี้ จะมีจุดเด่นในความสามารถในการแยกโหลดที่มีขนาดเท่ากันหรือใกล้เคียงกัน ดังสังเกตจากแผนภาพกล่อง (Box and whisker plot) ของค่ากำลังไฟฟ้าจริง (P) และค่ากำลังไฟฟ้าปรากฏ (S) ของโหลดที่ใช้จริงทั้ง 10 อุปกรณ์ ดังแสดงในรูปที่ 5.1 และ 5.2 ตามลำดับ โดยจากภาพจะเห็นได้ว่า หลอดไฟ พัดลม ตู้เย็นและโทรทัศน์จะมีช่วงกำลังไฟฟ้าที่ใช้งานทับซ้อนกันอยู่ ในกรณีเดียวกับกาน้ำไฟฟ้าและเตาไมโครเวฟที่มีช่วงกำลังไฟฟ้าที่ใช้งานทับซ้อนกันอยู่เช่นเดียวกัน ซึ่งระบบก็สามารถทำการแยกโหลดได้อย่างถูกต้องแม่นยำไม่ผิดพลาดแต่อย่างใด



รูปที่ 5.1 แผนภาพกล่องของค่ากำลังไฟฟ้าจริง (P) ของโหลดที่ใช้จริงทั้ง 10 อุปกรณ์



รูปที่ 5.2 แผนภาพกล่องของค่ากำลังไฟฟ้าปรากฏ (S) ของโหลดที่ใช้จริงทั้ง 10 อุปกรณ์

นอกจากนั้นเมื่อเทียบกับระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะที่ใช้งานที่มีพื้นฐานจากทฤษฎี Deep learning ที่มีความนิยมมากในปัจจุบันยังพบว่าระบบในงานวิจัยนี้ใช้ข้อมูลและเวลาในการ Training อัลกอริธึมสำหรับจำแนกโหลดน้อยกว่ามาก เช่นในงานวิจัย “Home Appliance Identification for NILM System Based on Deep Neural Network [11]” ได้ใช้ชุดข้อมูลของ REDD (Reference energy disaggregation data set) สำหรับ Training เพื่อสร้างอัลกอริธึมสำหรับจำแนกโหลดอุปกรณ์ไฟฟ้า 6 อุปกรณ์ ซึ่งใช้เวลาเก็บข้อมูลรวม 119 วัน สำหรับความถี่ในการเก็บข้อมูลที่ 1 วินาที ถ้าเทียบกับระบบในงานวิจัยนี้ใช้เวลาเก็บข้อมูลรวมทั้ง 10 อุปกรณ์ 4 ชั่วโมง 10 นาที ที่ความถี่ในการเก็บข้อมูลเดียวกัน

แต่อย่างไรก็ตามระบบมิเตอร์อัจฉริยะในงานวิจัยนี้ยังมีข้อด้อยที่ต้องปรับปรุงพัฒนาคือระบบดังกล่าวจะไม่สามารถตรวจจับการเปลี่ยนแปลงเมื่อมีการใช้งานอุปกรณ์ไฟฟ้าตั้งแต่ 2 อุปกรณ์ขึ้นไป พร้อมกัน โดยได้สรุปเปรียบเทียบข้อดีและข้อด้อยของระบบมิเตอร์อัจฉริยะแบบสามารถแยกโหลดในขณะที่ใช้งานเมื่อเปรียบเทียบกับระบบที่มีอยู่ในปัจจุบันดังแสดงในตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบระบบไมโครอัจฉริยะแบบสามารถแยกโหลดในขณะใช้งานที่มีอยู่ในปัจจุบัน

ลำดับ	คุณลักษณะในด้านต่างๆ	ชนิดของระบบไมโครอัจฉริยะที่เปรียบเทียบแบบสามารถแยกโหลดในขณะใช้งาน			
		ระบบไมโครอัจฉริยะในงานวิจัยนี้	ระบบไมโครอัจฉริยะแบบดิจิทัล เช่น FPGA	ระบบไมโครอัจฉริยะโดยวิธี Deep learning	ระบบไมโครอัจฉริยะโดยวิธี Deep learning
1	Accuracy score	1	1	1	0.95
2	เวลาที่ใช้ในการสร้างอัลกอริธึมแยกโหลด	น้อย	น้อย	มาก	น้อยมาก
3	ความยากในการสร้างอัลกอริธึมแยกโหลด	ปานกลาง	ง่าย	ยาก	ง่าย
4	ความสามารถในการแยกโหลดที่มากกว่า 2 อุปกรณ์ขึ้นไปทำงานพร้อมกันในเวลาเดียวกัน	ไม่สามารถทำได้	สามารถทำได้	สามารถทำได้	สามารถทำได้
5	ความสามารถในการแยกโหลดที่มีค่ากำลังไฟฟ้าเท่ากัน	สามารถทำได้	ไม่สามารถทำได้	สามารถทำได้	สามารถทำได้
6	ราคาของระบบ	ถูก	ปานกลาง	สูง	ปานกลาง

5.2 ข้อเสนอแนะ

จากข้อด้อยในประเด็นที่ระบบไมโครอัจฉริยะในงานวิจัยนี้ไม่สามารถตรวจจัดการเปลี่ยนแปลงเมื่อมีการใช้งานอุปกรณ์ไฟฟ้าตั้งแต่ 2 อุปกรณ์ขึ้นไปนั้น ทางผู้วิจัยได้เสนอแนวทางแก้ไขโดยการเพิ่มข้อมูลในการ Training ในส่วนของค่ากำลังจริง (P) และกำลังปรากฏ (S) ของอุปกรณ์ไฟฟ้าที่มีการใช้พร้อมกันตั้งแต่ 2 อุปกรณ์ขึ้นไปจนถึงใช้พร้อมกัน 10 อุปกรณ์ ซึ่งจะทำให้ต้องมีการเก็บข้อมูลเพื่อใช้ Training เพิ่มขึ้นอีก $2^{10} - 10$ หรือ 1,014 ชุดข้อมูล โดยเบื้องต้นสามารถเก็บชุดข้อมูลละ 25 นาที ในอัตราความถี่เก็บข้อมูล 1 วินาที ทำให้ต้องใช้เวลารวบรวมข้อมูลเพิ่มเป็น 17 วัน 18 ชั่วโมง 40 นาที แต่จะทำให้สามารถแก้ไขให้ระบบไมโครอัจฉริยะสามารถตรวจจัดการเปลี่ยนแปลงเมื่อมีการใช้งานอุปกรณ์ไฟฟ้าตั้งแต่ 2 อุปกรณ์ขึ้นไปได้

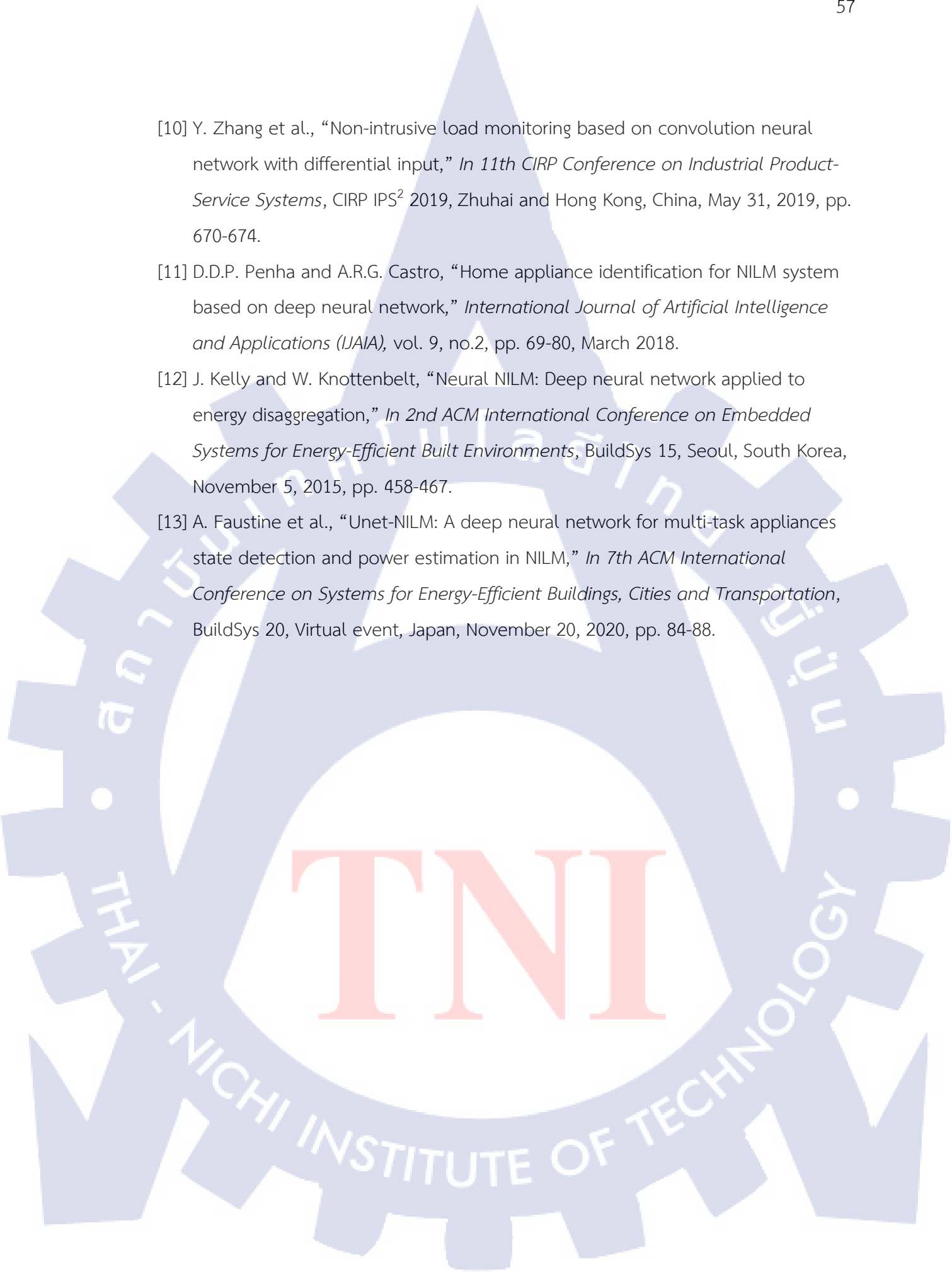
บรรณานุกรม

TNI

บรรณานุกรม

- [1] กระทรวงพลังงาน, แผนแม่บทการพัฒนาระบบโครงข่ายสมาร์ทกริดของประเทศไทย พ.ศ. 2558-2579, กรุงเทพฯ: สำนักงานนโยบายและแผนพลังงาน, 2558.
- [2] Y. Wang et al., "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions Smart Grid*, vol. 10, no. 3, pp. 3125-3148, May 2019.
- [3] R. Rojas, *Neural Networks*, Berlin: Springer-Verlag, 1996.
- [4] อีริเชษฐ์ สุรพันธ์ และณัฐพล ตันสังวร, "การสื่อสารในงานอุตสาหกรรมด้วยโพรโทคอล Modbus," [Online] Available: <https://www.nectec.or.th/news/news-public-document/modbus-protocol.html> [Accessed: 10 เมษายน 2564].
- [5] Q. Liu et al., "Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 28-37, February 2019.
- [6] S. Biansoongnern and B. Plangklang, "Nonintrusive load monitoring (NILM) using an artificial neural network in embedded system with low sampling rate," *In 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, ECTI-CON 2016, Chiang Mai, Thailand, June 28, 2016, pp. 66-70.
- [7] K.S. Barsim et al., "An approach for unsupervised non-intrusive load monitoring of residential appliances," *In 2nd Non-Intrusive Load Monitoring (NILM) Workshop*, NILM 14, Austin, USA, June 3, 2014, pp. 365-370.
- [8] K. Chahine et al, "Electric load disaggregation in smart metering using a novel feature extraction method and supervised classification," *Energy Procedia*, vol. 6, no. 1, pp. 627-632, December 2011.
- [9] H.H. Chang et al., "A non-intrusive load monitoring system using an embedded system for applications to unbalanced residential distribution system," *Energy Procedia*, Vol. 61, no. 1, pp. 146-150, January 2014.

- [10] Y. Zhang et al., “Non-intrusive load monitoring based on convolution neural network with differential input,” *In 11th CIRP Conference on Industrial Product-Service Systems*, CIRP IPS² 2019, Zhuhai and Hong Kong, China, May 31, 2019, pp. 670-674.
- [11] D.D.P. Penha and A.R.G. Castro, “Home appliance identification for NILM system based on deep neural network,” *International Journal of Artificial Intelligence and Applications (IJAIA)*, vol. 9, no.2, pp. 69-80, March 2018.
- [12] J. Kelly and W. Knottenbelt, “Neural NILM: Deep neural network applied to energy disaggregation,” *In 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys 15, Seoul, South Korea, November 5, 2015, pp. 458-467.
- [13] A. Faustine et al., “Unet-NILM: A deep neural network for multi-task appliances state detection and power estimation in NILM,” *In 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities and Transportation*, BuildSys 20, Virtual event, Japan, November 20, 2020, pp. 84-88.



TNI

ภาคผนวก

TNI

ภาคผนวก ก.

Meter and computer interface code (Arduino IDE)

TNI

ภาคผนวก ก.

Meter and computer interface code (Arduino IDE)

```

#include "REG_PM2200.h"
#include <ModbusMaster.h>
#include <SoftwareSerial.h>

//SoftwareSerial swSer(14, 12, false, 256);
SoftwareSerial swSer(10, 11); // RX, TX

ModbusMaster node;
//01 04 00 00 00 02 71 3F // Test 30001
//-----
// Convert 32bit to float
//-----
float HexTofloat(uint32_t x)
{
    return (*(float*)&x);
}

uint32_t FloatTohex(float x)
{
    return (*(uint32_t*)&x);
}
//-----

float Read_Meter_float(char addr , uint16_t REG)
{
    float i = 0;
    uint8_t j, result;

```

```

uint16_t data[2];
uint32_t value = 0;
node.begin(addr, swSer);
result = node.readHoldingRegisters (REG, 2); ///< Modbus function 0x03 Read
Holding Registers
delay(200);
if (result == node.ku8MBSuccess)
{
    for (j = 0; j < 2; j++)
    {
        data[j] = node.getResponseBuffer(j);
    }
    value = data[0];
    value = value << 16;
    value = value + data[1];
    i = HexToFloat(value);
    //Serial.println("Connect modbus Ok.");
    return i;
} else
{
    Serial.print("Connect modbus fail. REG >>> "); Serial.println(REG); // Debug
    delay(500);
    return 0;
}
}

void GET_METER()
{
    // Update read all data
    delay(500);
    for (char i = 0; i < Total_of_Reg ; i++)
    {

```

```

    DATA_METER[i] = Read_Meter_float(ID_meter, Reg_addr[i])*1000;
  }
}

//*****
*****

void setup()
{
  Serial.begin(9600);
  swSer.begin(9600);
}

void loop()
{
  //float x = Read_Meter_float(ID_meter,Reg_Volt);
  GET_METER();
  Serial.println(DATA_METER[0]);
  Serial.println(DATA_METER[1]);
  delay(1000);
}

```

REG_PM2200.h

```

#define ID_meter 1
#define Total_of_Reg 2

#define Reg_RP      3059    // 1.
#define Reg_AP      3075    // 2.

uint16_t Reg_addr[2] = {

```

```
Reg_RP,  
Reg_AP,  
};
```

```
float DATA_METER [Total_of_Reg] ;
```



ภาคผนวก ข.

Disaggregated load learning algorithm (Python)

TNI

ภาคผนวก ข.

Disaggregated load learning algorithm (Python)

```

import numpy as np
import pandas as pd
import serial

from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import pickle

from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler

raw = input('Enter your file path:')
cols = ['AVG_P','AVG_S','LABEL']
df = pd.read_csv(raw, usecols=cols)
df.AVG_S = pd.to_numeric(df.AVG_S,errors='coerce')
df.AVG_P = pd.to_numeric(df.AVG_P,errors='coerce')
df.dropna(inplace=True)
x = df[['AVG_P','AVG_S']]
target = df['LABEL']
y= target.ravel()
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

model = MLPClassifier(activation = 'logistic', solver = 'adam', alpha=0.00001,
max_iter=5000, learning_rate_init=0.001, hidden_layer_sizes=(300,))

```

```
model.fit(X_train, y_train)

filename = 'SmartMeter_model.sav'
pickle.dump(model, open(filename, 'wb'))

loaded_model = pickle.load(open(filename, 'rb'))
y_pred = loaded_model.predict(X_test)

print(y_pred)
print('Score -> {:.3f}'.format(model.score(X_test, y_test)))
print('Accuracy Score:', accuracy_score(y_test, y_pred).round(3))

cm = confusion_matrix(y_test, y_pred)
print(cm)
```



TNI

ภาคผนวก ค.

Smart meter programing (Python)

TNI

ภาคผนวก ค.

Smart meter programming (Python)

```
import serial
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import pickle
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
import tkinter
from threading import Thread
import queue
import time

class Serial(Thread):

    def __init__(self, queue):
        Thread.__init__(self)
        self.queue = queue

    def run(self):
        Data_P = []
        Data_S = []
        k=0
        m=0
```

```

n=0
sgn=1
AVG_P=0
AVG_S=0
y_pred=0
ser = serial.Serial(port='COM4',baudrate=9600,parity=serial.PARITY_NONE)
while True:
    line = ser.readline() # read a byte string
    if line: #Process serial
        string = line.decode() # convert the byte string to a unicode string
        num = int(float(string)) # convert the unicode string to an int
        print(num)
        # Form AVG_P and AVG_S
        if k%20 == 0:
            Data_P.append(num) # add int to data list
            if m>0:
                AVG_P = Data_P[m]-Data_P[m-1]
                m=m+1
            if m ==0:
                m=m+1
        elif k%20 == 1:
            Data_S.append(num) # add int to data list
            if n>0:
                AVG_S = Data_S[n]-Data_S[n-1]
                n=n+1
            if n == 0:
                n=n+1
        print(AVG_P)
        print(AVG_S)
        print(Data_P)
        print(Data_S)

```

```

print(m)
print(n)

# Define increase or decrease
if AVG_P<0:
    AVG_P=abs(AVG_P)
    sgn=-1
elif AVG_P>0:
    AVG_P=AVG_P
    sgn=1
if AVG_S<0:
    AVG_S=abs(AVG_S)
    sgn=-1
elif AVG_S>0:
    AVG_S=AVG_S
    sgn=1
print(sgn)

#Form dataframe
raw1 = [AVG_P]
raw2 = [AVG_S]
raw = list(zip(raw1,raw2))
print(raw)
cols = ['AVG_P','AVG_S']
df = pd.DataFrame(raw, columns=cols)
df.AVG_S = pd.to_numeric(df.AVG_S,errors='coerce')
df.AVG_P = pd.to_numeric(df.AVG_P,errors='coerce')
df.dropna(inplace=True)
x = df[['AVG_P','AVG_S']]
filename = 'SmartMeter_model.sav'
if k%20 ==1:

```

```

        loaded_model = pickle.load(open(filename, 'rb'))
        y_pred = loaded_model.predict(x)
        if sgn == -1:
            y_pred = -1*y_pred
        print(y_pred)
        k=k+1
        self.queue.put(y_pred)

class App(tkinter.Tk):

    def __init__(self):
        tkinter.Tk.__init__(self)

        # create canvas
        self.myCanvas = tkinter.Canvas(self, bg="white", height=600, width=400)

        # draw display
        HairDryer = self.myCanvas.create_rectangle(100,5,300,59.5, fill='gray',
tags=('HairDryer'))
        Lighting = self.myCanvas.create_rectangle(100,64.5,300,119, fill='gray',
tags=('Lighting'))
        Blender = self.myCanvas.create_rectangle(100,124,300,178.5, fill='gray',
tags=('Blender'))
        Fan = self.myCanvas.create_rectangle(100,183.5,300,238, fill='gray', tags=('Fan'))
        Toaster = self.myCanvas.create_rectangle(100,243,300,297.5, fill='gray',
tags=('Toaster'))
        Kettle = self.myCanvas.create_rectangle(100,302.5,300,357, fill='gray',
tags=('Kettle'))
        Microwave = self.myCanvas.create_rectangle(100,362,300,416.5, fill='gray',
tags=('Microwave'))

```



```

Refrigeration = self.myCanvas.create_rectangle(100,421.5,300,476, fill='gray',
tags=('Refrigeration'))

Television = self.myCanvas.create_rectangle(100,481,300,535.5, fill='gray',
tags=('Television'))

Iron = self.myCanvas.create_rectangle(100,540.5,300,595, fill='gray', tags=('Iron'))
self.myCanvas.create_text(50,29.75,font=('Helvetica','10','bold'), text="Hair Dryer")
self.myCanvas.create_text(50,91.75,font=('Helvetica','10','bold'), text="Lighting")
self.myCanvas.create_text(50,151.25,font=('Helvetica','10','bold'), text="Blender")
self.myCanvas.create_text(50,210.75,font=('Helvetica','10','bold'), text="Fan")
self.myCanvas.create_text(50,270.25,font=('Helvetica','10','bold'), text="Toaster")
self.myCanvas.create_text(50,329.75,font=('Helvetica','10','bold'), text="Kettle")
self.myCanvas.create_text(50,389.25,font=('Helvetica','10','bold'),
text="Microwave")
self.myCanvas.create_text(50,448.75,font=('Helvetica','10','bold'),
text="Refrigeration")
self.myCanvas.create_text(50,508.25,font=('Helvetica','10','bold'),
text="Television")
self.myCanvas.create_text(50,567.75,font=('Helvetica','10','bold'), text="Iron")

self.myCanvas.pack()
self.queue = queue.Queue()
thread = Serial(self.queue)
thread.start()
self.Process()

def Process(self):
    # Process display
    y_predi = self.queue.get()
    if y_predi == 1:
        self.myCanvas.itemconfig('HairDryer', fill='red')
    elif y_predi == 2:

```

```

        self.myCanvas.itemconfig('Lighting', fill='red')
    elif y_predi == 3:
        self.myCanvas.itemconfig('Blender', fill='red')
    elif y_predi == 4:
        self.myCanvas.itemconfig('Fan', fill='red')
    elif y_predi == 5:
        self.myCanvas.itemconfig('Toaster', fill='red')
    elif y_predi == 6:
        self.myCanvas.itemconfig('Kettle', fill='red')
    elif y_predi == 7:
        self.myCanvas.itemconfig('Microwave', fill='red')
    elif y_predi == 8:
        self.myCanvas.itemconfig('Refrigeration', fill='red')
    elif y_predi == 9:
        self.myCanvas.itemconfig('Television', fill='red')
    elif y_predi == 10:
        self.myCanvas.itemconfig('Iron', fill='red')
    elif y_predi == -1:
        self.myCanvas.itemconfig('HairDryer', fill='gray')
    elif y_predi == -2:
        self.myCanvas.itemconfig('Lighting', fill='gray')
    elif y_predi == -3:
        self.myCanvas.itemconfig('Blender', fill='gray')
    elif y_predi == -4:
        self.myCanvas.itemconfig('Fan', fill='gray')
    elif y_predi == -5:
        self.myCanvas.itemconfig('Toaster', fill='gray')
    elif y_predi == -6:
        self.myCanvas.itemconfig('Kettle', fill='gray')
    elif y_predi == -7:
        self.myCanvas.itemconfig('Microwave', fill='gray')

```

```
elif y_predi == -8:  
    self.myCanvas.itemconfig('Refrigeration', fill='gray')  
elif y_predi == -9:  
    self.myCanvas.itemconfig('Television', fill='gray')  
elif y_predi == -10:  
    self.myCanvas.itemconfig('Iron', fill='gray')  
  
self.after(10,self.Process)  
app = App()  
app.mainloop()
```



THAI-NICHI INSTITUTE OF TECHNOLOGY

TNI