การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐาน IEEE1888

นายเขตต์นนท์ ชูพุทธิพงศ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF DATA ANALYTIC PROGRAM FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSTEM

MR. KHETNON CHOOPUTTIPONG

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2016
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐาน IEEE1888
โดย	นายเขตต์นนท์ ชูพุทธิพงศ์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร.แนบบุญ หุนเจริญ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร)

.....กรรมการ
(อาจารย์ ดร.ภาณุวัฒน์ จันทร์ภักดี)

.....กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ภูมิพัฒ แสงอุดมเลิศ)

เขตต์นนท์ ชูพุทธิพงศ์ : การพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานไฟฟ้าภายในอาคารบนพื้นฐาน IEEE1888 (DEVELOPMENT OF DATA ANALYTIC PROGRAM FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSTEM)
อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร. เชาวน์ดิศ อัศวกุล, 116 หน้า

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับใช้ในระบบการจัดการพลังงานภายในอาคารที่มีพื้นฐานการสื่อสารข้อมูลด้วยมาตรฐาน IEEE1888 งานวิจัยมีเป้าหมายที่จะเพิ่มขีดความสามารถให้กับระบบในด้านการสร้างมโนภาพข้อมูลจากตัวรับรู้สภาพแวดล้อมประเภทต่าง ๆ ซึ่งมีติดตั้งภายในระบบ วิทยานิพนธ์ฉบับนี้ได้อาศัยทรัพยากรข้อมูลของโครงการ CUBEMS (Chulalonkorn university building energy management system) เพื่อพัฒนาต้นแบบทดสอบการพัฒนาโปรแกรมได้เลือกใช้คลัง Pandas ของภาษา python วิธีการทำสำเนาข้อมูลไว้ที่เครื่องคอมพิวเตอร์ที่รับผิดชอบโปรแกรมวิเคราะห์ข้อมูลได้ถูกพัฒนาขึ้นเพื่อลดระยะเวลาการร้องขอข้อมูลชุดเดิมจากหน่วยเก็บข้อมูลของ CUBEMS การสื่อสารเพื่อร้องขอข้อมูลทุกอย่างสอดคล้องตามมาตรฐาน IEEE1888 ด้วยโพรโทคอล FETCH และผลลัพธ์จากการคำนวณของโปรแกรมวิเคราะห์ข้อมูลถูกเขียนกลับลงไปยังหน่วยเก็บข้อมูลของ CUBEMS ด้วยโพรโทคอล WRITE ตามมาตรฐาน IEEE1888 นอกจากนี้วิทยานิพนธ์ฉบับนี้ได้พัฒนาโปรแกรมวิเคราะห์ความสัมพันธ์ของข้อมูลระหว่างตัวรับรู้การเคลื่อนไหว และตัวรับรู้อุณหภูมิ ร่วมกับข้อมูลการบริโภคพลังงาน ตัวกรองข้อมูลต่าง ๆ จากตัวรับรู้การเคลื่อนไหวได้ถูกนำเสนอไว้ในงานวิจัยนี้เพื่อเพิ่มความถูกต้องของการตรวจจับว่ามีผู้ใช้อยู่ในบริเวณที่ติดตั้งตัวรับรู้หรือไม่ ข้อมูลการอยู่ของผู้ใช้อาคารนี้ได้นำมาใช้เพื่อนิยามตัวชี้วัดความสูญเปล่าของการใช้พลังงานในระบบปรับอากาศ กล่าวคือ จะนับว่ามีการสูญเปล่าที่ไม่จำเป็นของการใช้พลังงานเมื่อมีการเปิดให้ระบบปรับอากาศทำงานในบริเวณที่ไม่มีผู้ใช้งานอยู่เลย หรือเมื่อมีการปรับอุณหภูมิให้ต่ำเกินไป ผลการวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศได้ถูกนำเสนอที่จอแสดงผลของโครงการ CUBEMS เว็บไซต์การวิเคราะห์ข้อมูล และการแจ้งเตือนแบบอัตโนมัติผ่านโปรแกรมประยุกต์ไลน์ ผลการวิเคราะห์ข้อมูลเบื้องต้นเป็นระยะเวลา 1 ปีในบริเวณชั้น 13 อาคารเจริญวิศวกรรมที่มีเครื่องปรับอากาศ 22 เครื่อง และตัวรับรู้สภาพแวดล้อม 43 ตัว แสดงให้เห็นว่ามีการสูญเปล่าของพลังงานมากถึง 24,085 กิโลวัตต์ชั่วโมง หรือคิดเป็นจำนวนเงิน 134,395 บาทต่อปี และสำหรับบริเวณชั้น 12 ของอาคารเดียวกันพบว่ามีการสูญเ ปล่าของพลังงานเท่ากับ 6,124 กิโลวัตต์ชั่วโมง หรือคิดเป็น 34,175 บาทต่อปี ผลวิเคราะห์นี้แสดงหลักฐานเชิงตัวเลขซึ่งเสนอว่าอาจมีการปรับลดการบริโภคพลังงานได้อย่างมากและเป็นรูปธรรมเพียงการสร้างให้เกิดการตระหนักรู้วินัยการใช้พลังงานของผู้ใช้อาคารได้ต่อไปในอนาคต

ภาควิชา วิศวกรรมไฟฟ้า **ลายมือชื่อนิสิต**.....
สาขาวิชา วิศวกรรมไฟฟ้า **ลายมือชื่อ อ.ที่ปรึกษาหลัก**.....
ปีการศึกษา ....2559.....

5770129921 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: DATA ANALYTIC/MOTION SENSOR/ WASTED ENERGY/ BUILDING ENERGY MANAGEMENT.

MR. KHETNON CHOOPUTTIPONG: DEVELOPMENT OF DATA ANALYTIC PROGRAM FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSTEM ADVISOR: ASSOC. PROF. CHAODIT ASWAKUL, Ph.D., 116 pp.

This thesis presents the development of data analytic program for the building energy management system with data communication based on IEEE1888 standard. The research is aimed at enhancing the system capability with the visualization of data from the installed sensors inside the system. This thesis has relied on data resources of the CUBEMS (Chulalonkorn university building energy management system) project to develop the prototype testbed. The program development has chosen the Pandas library of python language. A data caching method at the computer responsible for the data analytics has been developed to decrease the time period of the query for previously requested data from the CUBEMS data storage. All data requests are compliant with IEEE1888 FETCH protocol. And the computational output from the data analytics is updated into the CUBEMS storage with IEEE1888 WRITE protocol. In addition, this thesis has developed programs to analyze the relationship between the motion and temperature sensor data together with the energy data from smart meters. Data filters from motion sensors have been proposed in this research to improve the accuracy of identifying the presence of people at the sensor location. The presence data of users in the building is then used to define an indicator of wasted energy in air conditioning systems; that is, the energy usage is counted as an unnecessary energy waste when the air conditioning system is turned on without any detectable people movement or when the sensed temperature is lower than a minimum threshold allowable. The wasted energy output results are shown at CUBEMS interactive in-house displays, a data analytics website and an automatic Line-app notification. Based on the analysis of recent data over a one-year period on 13th floor, Engineering 4 buildings, with 22 air conditioning systems and 43 CUBEMS sensors suggests that there has been a wasted energy of up to 24,085 kWh or equivalently 134,395 Bath per year. And for 12th floor of the same building, the wasted energy has been found to be up to 6,124 kWh or equivalently 34,175 Bath per year. This analytics result shows a quantitative evidence with a strong suggestion towards tangible much energy savings by merely provoking people awareness in energy usage disciplines in the future.

Department	: Electrical Engineering	**Student's Signature**	.....
Field of Study	: Electrical Engineering	**Advisor's Signature**	.....
Academic Year	: 2016		

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล ที่ช่วยถ่ายทอดความรู้ทั้งทางด้านวิจัยและด้านวิชาการ ให้คำปรึกษา และคำแนะนำ รวมถึงช่วยตรวจทานวิทยานิพนธ์ฉบับนี้อย่างดีเสมอมา ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้ และขอขอบพระคุณ รองศาสตราจารย์ ดร.แนบบุญ หุนเจริญ ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร อาจารย์ ดร.ภาณุวัฒน์ จันทร์ภักดี และรองศาสตราจารย์ ดร.ภูมิพัฒ แสงอุดมเลิศ กรรมการสอบวิทยานิพนธ์ ที่สละเวลาตรวจสอบ รวมถึงให้คำแนะนำสำหรับวิทยานิพนธ์ฉบับนี้เพื่อความสมบูรณ์ยิ่งขึ้น

งานวิจัยนี้ได้รับการสนับสนุนจากหน่วยปฏิบัติการวิจัยโครงข่ายไร้สาย และอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Unit) กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย

ขอขอบคุณกลุ่มวิจัยโครงข่ายไร้สายและอินเทอร์เน็ตอนาคตภายใต้การดูแลของ รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล และผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร ที่จัดกิจกรรมเพื่อส่งเสริมการเรียนรู้ การทำงาน และทักษะเพื่อเป็นผู้วิจัยที่มีประสิทธิภาพที่ดียิ่งขึ้น

ขอขอบคุณเพื่อน พี่ น้องนักวิจัย เจ้าหน้าที่ บุคลากร และคณาจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้คำแนะนำ ความช่วยเหลือเรื่องต่าง ๆ และการสนับสนุนที่ดีเสมอมา

สุดท้ายนี้ขอขอบคุณครอบครัวของผู้วิจัย ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจให้แก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สารบัญ

หน้า

บทคัดย่อภาษาไทย . . . . . ง
บทคัดย่อภาษาอังกฤษ . . . . . จ
กิตติกรรมประกาศ . . . . . ฉ
สารบัญ . . . . . ช
สารบัญตาราง . . . . . ญ
สารบัญรูป . . . . . ฏ
บทที่
1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิทยานิพนธ์ 4
1.3 ขอบเขตวิทยานิพนธ์ 4
1.4 ขั้นตอน และวิธีดำเนินงาน 5
1.5 ประโยชน์ที่คาดว่าจะได้รับ 5
1.6 ประมวลวิทยานิพนธ์ 6
2 ทฤษฎีพื้นฐาน 7
2.1 มาตรฐาน IEEE1888 7
2.1.1 สถาปัตยกรรม IEEE1888 7
2.1.2 การสื่อสารพื้นฐานในมาตรฐาน IEEE1888 8
2.2 ระบบการจัดการพลังงานภายในอาคาร จุฬาลงกรณ์มหาวิทยาลัย 10
2.3 การสร้างมโนภาพข้อมูลในระบบการจัดการพลังงานภายในอาคาร 14
3 สถาปัตยกรรมของระบบและการทดสอบเบื้องต้น 17
3.1 สถาปัตยกรรมของระบบ 17
3.2 การทดสอบเชื่อมต่อโปรแกรมประยุกต์กับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 18
3.2.1 การทดสอบร้องขอข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 โดยอาศัยโพรโทคอล FETCH 18
3.2.2 การทดสอบส่งข้อมูลไปที่หน่วยเก็บข้อมูลมาตรฐาน IEEE1888 โดยอาศัยโพรโทคอล WRITE 23
3.3 การออกแบบและพัฒนาโปรแกรมประยุกต์ต้นแบบ 24
3.3.1 การพัฒนาส่วนต่อประสานของโปรแกรมประยุกต์ 24
3.3.2 การออกแบบฟังชันก์การสร้างมโนภาพข้อมูลของโปรแกรมประยุกต์ 31
4 การสร้างและทดสอบฟังก์ชันวิเคราะห์ข้อมูลของโปรแกรมประยุกต์ 34
4.1 การออกแบบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 34
4.2 การออกแบบชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ 37
4.2.1 ตัวกรองการจัดกลุ่ม 37
4.2.2 ตัวกรองการเดินผ่าน 40
4.2.3 ตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว 42

หน้า

4.2.4 ตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ 43
4.3 การทดสอบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 51
4.3.1 การทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน 51
4.3.2 การทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในลักษณะการปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ . . . 54
4.4 การวิเคราะห์ผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 58
4.4.1 พื้นที่ต่าง ๆ และการสูญเปล่าของพลังงานในระบบปรับอากาศ 58
4.4.2 อุณหภูมิภายนอกและการสูญเปล่าของพลังงานในระบบปรับอากาศ 61
4.4.3 การเปิดปิดภาคการศึกษาและการสูญเปล่าของพลังงานในระบบปรับอากาศ 63
4.5 การประยุกต์ใช้งานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ . 65
4.6 การประเมินสมรรถนะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 70
4.6.1 การประเมินสมรรถนะการทำงานของฟังก์ชันในรูปแบบการคำนวณรายห้อง . . . 71
4.6.2 การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ 73
4.6.3 การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายชั้น 75
5 บทสรุปและข้อเสนอแนะ 77
5.1 บทสรุป 77
5.2 ข้อเสนอแนะ 78
บรรณานุกรม 79
ภาคผนวก 80
ก ชุดคำสั่งสำหรับจัดการ PointID 82
ก.1 ชุดคำสั่งสำหรับสร้าง PointID ในรูปแบบเฟรมข้อมูล 82
ก.2 ชุดคำสั่งค้นหา PointID จากลักษณะเฉพาะ 82
ก.3 ชุดคำสั่งสำหรับค้นหาชื่อห้อง 84
ก.4 ชุดคำสั่งสำหรับค้นหาชื่อห้องที่ถูกแบ่งออกเป็นกลุ่ม 85
ข ชุดคำสั่งสำหรับติดต่อกับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 87
ข.1 ชุดคำสั่งเตรียมคลัง (โปรแกรม) ที่จำเป็นสำหรับติดต่อกับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 87
ข.2 ชุดคำสั่งสำหรับสร้าง XML เพื่อร้องขอข้อมูล 87
ข.3 ชุดคำสั่งสำหรับสร้าง XML เพื่อร้องขอข้อมูลแบบเฉพาะ 88
ข.4 ชุดคำสั่งสำหรับสร้าง XML เพื่อเขียนข้อมูล 88
ข.5 ชุดคำสั่งสำหรับส่งคำสั่งร้องขอข้อมูล 88
ข.6 ชุดคำสั่งสำหรับส่งคำสั่งเขียนข้อมูล 90
ข.7 ชุดคำสั่งสำหรับตรวจเช็คค่าเวลาแรกสุดหรือล่าสุด 90
ค ชุดคำสั่งการเตรียมข้อมูลของโปรแกรมประยุกต์ 92
ค.1 ชุดคำสั่งสำหรับตรวจเช็คเวลา 92
ค.2 ชุดคำสั่งสำหรับจัดการเฟรมข้อมูลให้แถวลำดับอยู่ในรูปแบบวันที่และเวลาใด ๆ . . 92
ค.3 ชุดคำสั่งสำหรับตรวจเช็คการอยู่ในช่วงเพื่อระบุรูปแบบการร้องขอ 92

หน้า

ค.4 ชุดคำสั่งสำหรับตรวจเช็คการมีอยู่ของไฟล์ PointID 92
ค.5 ชุดคำสั่งสำหรับตรวจเหตุการณ์ในการเตรียมข้อมูลของส่วนต่อประสาน 92
ค.6 ชุดคำสั่งสำหรับเตรียมข้อมูลให้อยู่ในรูปแบบเฟรมข้อมูล 93
ค.7 ชุดคำสั่งสำหรับเตรียมข้อมูลประเภทต่าง ๆ 94
ง ชุดคำสั่งสำหรับปรับปรุงข้อมูล 100
ง.1 ชุดคำสั่งสำหรับแปลงแถวลำดับของเฟรมข้อมูลให้อยู่ในลักษณะเวลาตามแถวตั้ง และวันตามแถวนอน 100
ง.2 ชุดคำสั่งสำหรับสร้างเฟรมข้อมูลจากข้อมูลประเภท Dictionary 100
ง.3 ชุดคำสั่งสำหรับเลือก key จากข้อมูลประเภท Dictionary 100
จ ชุดคำสั่งฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 101
จ.1 ชุดคำสั่งตัวกรองการเดินผ่าน 101
จ.2 ชุดคำสั่งตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว 101
จ.3 ชุดคำสั่งตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้หายไปจากพื้นที่ 101
จ.4 ชุดคำสั่งการคำนวณการสูญเปล่าของพลังงานในระบบปรับอากาศ 102
ฉ ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม 112
ฉ.1 ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม 1 112
ฉ.2 ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม 2 112
ช ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของพลังงาน ในระบบปรับอากาศ 114
ช.1 ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของ พลังงานในระบบปรับอากาศ 1 114
ช.2 ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของ พลังงานในระบบปรับอากาศ 2 114
ประวัติผู้เขียนวิทยานิพนธ์ 116

สารบัญตาราง

หน้า

ตารางที่ 2.1 รายละเอียดชนิดข้อมูลที่ถูกตรวจวัดในโครงการ CUBEMS 11
ตารางที่ 2.2 รายละเอียด PointID ในพื้นที่แต่ละบริเวณของโครงการ CUBEMS 12
ตารางที่ 3.1 คุณลักษณะของคอมพิวเตอร์ที่ใช้ในการทดสอบโปรแกมประยุกต์ 18
ตารางที่ 3.2 โครงสร้างข้อมูลตามมาตรฐาน IEEE1888 [15] 19
ตารางที่ 3.3 โครงสร้างข้อมูลตามมาตรฐาน IEEE1888 (ต่อ) [15] 20
ตารางที่ 3.4 เหตุการณ์ที่เกิดขึ้นในขั้นตอนการเตรียมข้อมูล 27
ตารางที่ 4.1 หลักการตรวจวัดข้อมูลของตัวรับรู้การเคลื่อนไหวชนิด PIR ของโครงการ CUBEMS 36
ตารางที่ 4.2 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ก่อนและหลังผ่านตัวกรองข้อมูล 47
ตารางที่ 4.3 หลักเกณฑ์การระบุเหตุการณ์การสูญเปล่าของพลังงานในระบบปรับบอากาศ . . . 50
ตารางที่ 4.4 รายละเอียดจำนวนระบบและตัวรับรู้สภาพแวดล้อมของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร 51
ตารางที่ 4.5 รายละเอียดตัวแปรขาเข้าของการทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน 51
ตารางที่ 4.6 การสูญเปล่าของพลังงานในระบบปรับอากาศของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 53
ตารางที่ 4.7 รายละเอียดจำนวนระบบและตัวรับรู้สภาพแวดล้อมของห้องแต่ละห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรม 55
ตารางที่ 4.8 รายละเอียดตัวแปรขาเข้าของการทดสอบปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ของฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศน 55
ตารางที่ 4.9 การทดสอบปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 56
ตารางที่ 4.10 การทดสอบปรับเปลี่ยนค่าขีดสุดอุณหภูมิของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 57
ตารางที่ 4.11 ผลรวมการสูญเปล่าของพลังงานและค่าใช้จ่ายจากพลังงานที่สูญเปล่าในระบบปรับอากาศบริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560 60
ตารางที่ 4.12 พื้นที่ที่มีผลรวมการสูญเปล่าของพลังงาน และค่าใช้จ่ายจากพลังงานสูญเปล่าในระบบปรับอากาศสูงสุด 5 อันดับแรกในเดือน กุมภาพันธ์ และ เมษายน พ.ศ. 2559 63
ตารางที่ 4.13 ตัวอย่างการเปรียบเทียบค่าใช้จ่ายของพลังงานที่สูญเปล่าในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษา 65
ตารางที่ 4.14 รูปแบบผลลัพธ์ของ PointID 66
ตารางที่ 4.15 รูปแบบข้อกำหนดของ PointID 66

หน้า

ตารางที่ 4.16 รายละเอียดและจำนวนข้อมูลของห้องพักอาจารย์ cak/cpp/lwk วันที่ 10 พฤษภาคม พ.ศ. 2560 72
ตารางที่ 4.17 รายละเอียดและจำนวนข้อมูลห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร วันที่ 10 พฤษภาคม พ.ศ. 2560 73
ตารางที่ 4.18 รายละเอียดและจำนวนข้อมูลชั้น 13 อาคารเจริญวิศวกรรม วันที่ 10 พฤษภาคม พ.ศ. 2560 75

สารบัญรูป

หน้า

รูปที่ 1.1 การประเมินระบบการจัดการพลังงานภายในอาคารโดยทีม Navigant research [11] 3
รูปที่ 2.1 สถาปัตยกรรม IEEE1888 [15] 7
รูปที่ 2.2 โพรโทคอล FETCH 8
รูปที่ 2.3 โพรโทคอล WRITE 9
รูปที่ 2.4 โพรโทคอล TRAP 9
รูปที่ 2.5 มาตราวัดอัจฉริยะของโครงการ CUBEMS 10
รูปที่ 2.6 ตัวรับรู้สภาพแวดล้อมของโครงการ CUBEMS 10
รูปที่ 2.7 เกตเวย์ตัวรับรู้สภาพแวดล้อมของโครงการ CUBEMS 10
รูปที่ 2.8 หน่วยเก็บข้อมูลของโครงการ CUBEMS 10
รูปที่ 2.9 โครงสร้าง PointID ของโครงการ CUBEMS 11
รูปที่ 2.10 ตัวอย่าง PointID ของโครงการ CUBEMS 11
รูปที่ 2.11 การแสดงผลบนเว็บไซต์ของโครงการ CUBEMS 12
รูปที่ 2.12 การแสดงผลบนโปรแกรมประยุกต์ระบบปฏิบัติการแอนดรอยด์โครงการ CUBEMS 13
รูปที่ 2.13 จอแสดงผลของโครงการ CUBEMS [19] 13
รูปที่ 2.14 การสร้างมโนภาพข้อมูลการบริโภคพลังงานของโครงการมหาวิทยาลัยโตเกียวสี [16] . 14
รูปที่ 2.15 การสร้างมโภาพข้อมูลเพื่อนำเสนอสัดส่วนการบริโภคพลังงานในรูปแบบแผนภูมิวงกลมของโครงการ CUBEMS 14
รูปที่ 2.16 การสร้างมโภาพข้อมูลเพื่อนำเสนอข้อมูลสภาพแวดล้อมในระดับเขตพื้นที่ของโครงการ CUBEMS 15
รูปที่ 2.17 การสร้างมโภาพข้อมูลร่วมกับการใช้ระดับสีเพื่อบ่งบอกถึงสัดส่วนการบริโภคพลังงานเทียบกับค่าขีดสุดของโครงการ CUBEMS 15
รูปที่ 2.18 แผงหน้าปัดระบบการจัดการพลังงานภายในอาคารบริษัท Verdigris [24] 16
รูปที่ 2.19 การสร้างมโนภาพข้อมูลการบริโภคพลังงานของบริษัท Ramotion [25] 16
รูปที่ 3.1 สถาปัตยกรรมของระบบ 17
รูปที่ 3.2 WSDL ที่ได้ตอบกลับจากหน่วยเก็บข้อมูลโครงการ CUBEMS 21
รูปที่ 3.3 การร้องขอข้อมูลจากโปรแกรมประยุกต์ไปยังหน่วยเก็บข้อมูลโดยโพรโทคอล FETCH 21
รูปที่ 3.4 การตอบกลับพร้อมข้อมูลจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล FETCH 22
รูปที่ 3.5 ค่าเฉลี่ยระยะเวลาการร้องขอข้อมูลอุณหภูมิจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 จำนวน 1 PointID ที่แปรผันตามจำนวนวันที่ร้องขอ 22
รูปที่ 3.6 การส่งข้อมูลไปยังหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 23
รูปที่ 3.7 การตอบกลับกระบวนการส่งข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 23
รูปที่ 3.8 สถาปัตยกรรมของระบบหลังถูกปรับปรุง 24
รูปที่ 3.9 ผังงานการทำงานของส่วนต่อประสานสำหรับตีความคำสั่งเพื่อกำหนดแหล่งที่มาของข้อมูล 25
รูปที่ 3.10 PointID ของระบบ (บางส่วน) ในรูปแบบแถวลำดับ 25
รูปที่ 3.11 PointID ของระบบ (บางส่วน) ที่ถูกจัดแบ่งในรูปแบบเฟรมข้อมูล 26

หน้า

รูปที่ 3.12 ผลลัพธ์การตีความคำสั่งเพื่อกำหนด PointID ของระบบปรับอากาศบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร เขตพื้นที่ย่อย 4 26
รูปที่ 3.13 ผลลัพธ์การตีความคำสั่งเพื่อกำหนด PointID ของตัวรับรู้การเคลื่อนไหวบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร เขตพื้นที่ย่อย 4 26
รูปที่ 3.14 เหตุการณ์ที่เกิดขึ้นในขั้นตอนการเตรียมข้อมูล 27
รูปที่ 3.15 ไฟล์ข้อมูล PointID (บางส่วน) ในลักษณะไฟล์ csv 28
รูปที่ 3.16 ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของทุกระบบบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบเฟรมข้อมูล 29
รูปที่ 3.17 ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของช่วงเวลาแต่ละช่วงเวลาบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบเฟรมข้อมูล 29
รูปที่ 3.18 ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรมในรูปแบบเฟรมข้อมูล 30
รูปที่ 3.19 ข้อมูลการตรวจจับการเคลื่อนไหว ค่าอุณหภูมิ และค่าการบริโภคพลังงานของระบบปรับอากาศในหน่วยวัตต์ชั่วโมงในรูปแบบเฟรมข้อมูล 30
รูปที่ 3.20 ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบแผนภูมิความร้อนตลอดเดือนมกราคม 31
รูปที่ 3.21 ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของทุกระบบในรูปแบบกราฟวงกลมบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารแยกระบบ 31
รูปที่ 3.22 ค่าเฉลี่ยการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเดือนมกราคม พ.ศ. 2560 32
รูปที่ 3.23 การแสดงผลบนเว็บไซต์วิเคราะห์ข้อมูล 32
รูปที่ 3.24 การแสดงผลบนเว็บไซต์วิเคราะห์ข้อมูล (เพิ่มเติม) 33
รูปที่ 3.25 การแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์เมื่อเกิดเหตุการณ์ที่ตัวรู้มีสถานะการทำงานผิดปกติ 33
รูปที่ 4.1 ผังงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 35
รูปที่ 4.2 ตัวอย่างและรายระเอียดการตรวจวัดข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4 36
รูปที่ 4.3 ชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ 37
รูปที่ 4.4 แผนผังแสดงเขตพื้นที่บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร ชั้น 13 อาคารเจริญวิศวกรรม 37
รูปที่ 4.5 ผังงานตัวกรองการจัดกลุ่ม 38
รูปที่ 4.6 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ของตัวรับรู้ที่ 1 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4 39
รูปที่ 4.7 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ของตัวรับรู้ที่ 2 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4 39
รูปที่ 4.8 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการจัดกลุ่ม 39
รูปที่ 4.9 ผังงานตัวกรองการเดินผ่าน 40
รูปที่ 4.10 ตัวอย่างการทำงานของตัวกรองการเดินผ่าน 41
รูปที่ 4.11 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการเดินผ่าน 41

หน้า

รูปที่ 4.12 ผังงานตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว 42
รูปที่ 4.13 ตัวอย่างการทำงานของตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวโดยกำหนดค่าขีด-สุดเท่ากับ 2 นาที 43
รูปที่ 4.14 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวโดยกำหนดค่าขีดสุดเท่ากับ 2 นาที 43
รูปที่ 4.15 ผังงานตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ 44
รูปที่ 4.16 ตัวอย่างการทำงานของตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 4 นาที 45
รูปที่ 4.17 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 5 นาที 45
รูปที่ 4.18 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 15 นาที 46
รูปที่ 4.19 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 30 นาที 46
รูปที่ 4.20 ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 60 นาที 46
รูปที่ 4.21 ตัวอย่างภาพรวมกระบวนการกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ 48
รูปที่ 4.22 ตัวอย่างข้อมูลฟังก์ชันวิเคราะห์การสูญเปล่าในระบบปรับอากาศ 49
รูปที่ 4.23 ตัวอย่างข้อมูลกรณีเกิดการสูญเปล่าของพลังงานจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ 49
รูปที่ 4.24 ตัวอย่างข้อมูลกรณีเกิดการสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น 50
รูปที่ 4.25 ข้อมูลการบริโภคพลังงานสะสมตั้งแต่วันแรกของเดือนในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 ของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร 52
รูปที่ 4.26 ค่าเฉลี่ยการสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน กุมภาพันธ์ พ.ศ. 2560 ของพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร 53
รูปที่ 4.27 แผนผังแสดงเขตพื้นที่บริเวณชั้น 13 อาคารเจริญวิศวกรรม 54
รูปที่ 4.28 สัดส่วนการสูญเปล่าของพลังงานในระบบปรับอากาศเมื่อมีการปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ 58
รูปที่ 4.29 ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศของห้องแต่ละห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560 59
รูปที่ 4.30 ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศของห้องแต่ละห้องบริเวณชั้น 12 อาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560 59
รูปที่ 4.31 ค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศบริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560 61

หน้า

รูปที่ 4.32 ค่าเฉลี่ยอุณหภูมิในช่วงเวลาต่าง ๆ ของวันในเดือนกุมภาพันธ์ และเมษายน พ.ศ. 2559 61
รูปที่ 4.33 ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างเดือนเมษายน และกุมภาพันธ์ พ.ศ. 2559 ของเขตพื้นที่ต่าง ๆ บริเวณชั้น 13 อาคารเจริญวิศวกรรม . . . 62
รูปที่ 4.34 ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างเดือนเมษายน และกุมภาพันธ์ พ.ศ. 2559 ของเขตพื้นที่ต่าง ๆ บริเวณชั้น 12 อาคารเจริญวิศวกรรม . . . 62
รูปที่ 4.35 ค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศระหว่างเดือนเมษายน และ กุมภาพันธ์ พ.ศ. 2559 บริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรม 62
รูปที่ 4.36 การเปรียบเทียบผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาของเขตพื้นที่ต่าง ๆ บริเวณชั้น 13 อาคารเจริญวิศวกรรม . . 64
รูปที่ 4.37 การเปรียบเทียบผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาของเขตพื้นที่ต่าง ๆ บริเวณชั้น 12 อาคารเจริญวิศวกรรม . . 64
รูปที่ 4.38 การเปรียบเทียบค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาบริเวณห้องพักอาจารย์ชั้น 12 และ 13 อาคารเจริญวิศวกรรม 64
รูปที่ 4.39 ลักษณะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 65
รูปที่ 4.40 รูปแบบการระบุ PointID ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 65
รูปที่ 4.41 การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานทุก 15 นาที 66
รูปที่ 4.42 การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศในรูปแบบสัดส่วนการสูญเปล่าของพลังงานตลอดวัน 67
รูปที่ 4.43 การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานตลอดวันจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ 67
รูปที่ 4.44 การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานตลอดวันจากการตั้งค่าอุณหภูมิเกินความจำเป็น 67
รูปที่ 4.45 การแสดงผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบนจอแสดงผลภายโครงการ CUBEMS 68
รูปที่ 4.46 การแสดงผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบนเว็บไซต์การวิเคราะห์ข้อมูล 68
รูปที่ 4.47 การแจ้งเตือนผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศผ่านโปรแกรมประยุกต์ไลน์ 69
รูปที่ 4.48 การแจ้งเตือนผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศผ่านโปรแกรมประยุกต์ไลน์ (เพิ่มเติม) 69

หน้า

รูปที่ 4.49 ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบในขณะดำเนินงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 70
รูปที่ 4.50 ระดับการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของของเครื่องคอมพิวเตอร์ทดสอบในขณะดำเนินงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ 71
รูปที่ 4.51 รูปแบบการดำเนินพร้อมกันของโปรแกรมประยุกต์เพื่อคำนวณแบบแยกเขตพื้นที่ . . 71
รูปที่ 4.52 ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้อง 72
รูปที่ 4.53 ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้อง 72
รูปที่ 4.54 ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ 74
รูปที่ 4.55 ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ 74
รูปที่ 4.56 ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายชั้น 76
รูปที่ 4.57 ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายชั้น 76

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

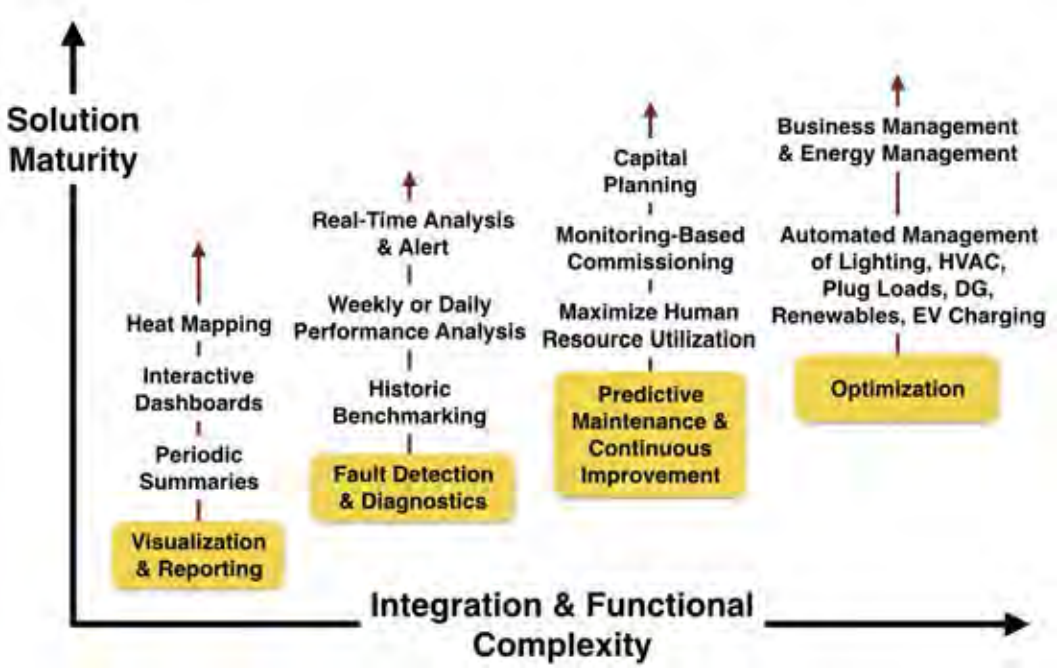
จากรายงานสถิติการบริโภคพลังงานภายในประเทศไทย [1] และสถิติการบริโภคพลังงานโดยรวมของโลก [2] พบว่าปริมาณการบริโภคพลังงานไฟฟ้ามีแนวโน้มสูงขึ้นเมื่อเปรียบเทียบกับปีที่ผ่านมา สอดคล้องกับอัตราค่าไฟฟ้าที่มีแนวโน้มสูงขึ้น [3] อันเนื่องมาจากสาเหตุต่าง ๆ เช่น จำนวนประชากรที่เพิ่มขึ้น ภาวะโลกร้อน ทรัพยากรธรรมชาติที่ใช้วัตถุดิบในการผลิตพลังงานมีอยู่อย่างจำกัด ตลอดจนการพัฒนาเทคโนโลยีในปัจจุบันที่เป็นไปอย่างรวดเร็ว ทำให้ปัจจุบันเกิดเป็นกระแสความตื่นตัวในเรื่องการบริหารจัดการพลังงาน (energy management) ทั้งนี้เพื่อให้การบริหารจัดการพลังงานไฟฟ้าเป็นไปอย่างมีประสิทธิภาพควรมีการบันทึกข้อมูลการบริโภคพลังงานด้วยระบบสารสนเทศ สำหรับใช้คำนวณค่าใช้จ่ายด้านพลังงานไฟฟ้า รวมถึงใช้ศึกษาลักษณะการบริโภคพลังงานในสภาพแวดล้อมรูปแบบต่าง ๆ ซึ่งระบบโครงข่ายไฟฟ้าที่มีอยู่เดิมไม่สามารถตอบสนองความต้องการข้างต้นได้ ด้วยการนำเทคโนโลยีสมาร์ทกริด (Smart Grid) หรือโครงข่ายไฟฟ้าอัจฉริยะ ซึ่งเป็นเทคโนโลยีด้านการบริหารจัดการพลังงานไฟฟ้าอาศัยการบูรณาการ (integration) โครงสร้างพื้นฐานของระบบผลิตไฟฟ้าที่เดิมเข้ากับโครงสร้างพื้นฐานด้านเทคโนโลยีสารสนเทศ และการสื่อสาร (information and communication technology) ทำให้สามารถตรวจวัด ควบคุม จัดเก็บ รวมถึงจัดสรรพลังงานไฟฟ้าได้อย่างมีประสิทธิภาพ ส่งผลให้คุณภาพการใช้ชีวิตดีขึ้น ลดผลกระทบต่อสิ่งแวดล้อม ในบทความ [4] อธิบายความหมายของสมาร์ทกริดเป็นระบบโครงข่ายที่ส่งจ่ายพลังงานไฟฟ้าแบบครบวงจรโดยอาศัยเทคโนโลยีดิจิทัลเพื่อจัดสรรพลังงานไฟฟ้าจากผู้ให้บริการไปยังผู้บริโภคด้วยระบบการสื่อสารสองทาง (two-way communication) สำหรับควบคุมการทำงานของอุปกรณ์ที่บริโภคพลังงานไฟฟ้าให้เป็นไปอย่างมีประสิทธิภาพ โดยผู้ให้บริการระบบส่งจ่ายไฟฟ้าสมาร์ทกริดจะติดตั้งโปรแกรมประยุกต์พร้อมอุปกรณ์ตรวจวัดไว้ที่สิ่งปลูกสร้างของผู้บริโภคเพื่อใช้ตรวจสอบปริมาณการบริโภคพลังงานไฟฟ้า รวมถึงช่วยคำนวณการแจกจ่ายพลังงานไฟฟ้าในแต่ละพื้นที่เพื่อเพิ่มความเสถียรของระบบส่ งจ่ายไฟฟ้า ลดปัญหาไฟฟ้าดับ (blackout) หรือไฟตกในช่วงที่มีความต้องการพลังงานไฟฟ้าสูง รวมถึงทำให้ผู้บริโภคทราบถึงพฤติกรรมการบริโภคพลังงานไฟฟ้าของตน คาดหวังให้เกิดการตระหนักรู้ (awareness) ในเรื่องการบริโภคพลังงานอย่างมีประสิทธิภาพ ในบทความ [5] กล่าวถึงการจัดการพลังงานสามารถแบ่งออกเป็นหลายระดับตามลักษณะของสิ่งปลูกสร้าง เรียกทั้งหมดว่า ระบบจัดการพลังงานชุมชน (community energy management systems, CEMS) โดยแบ่งเป็นสามระดับประกอบไปด้วย ระบบการจัดการพลังงานภายในที่อยู่อาศัย (house energy management systems, HEMS) ระบบการจัด-การพลังงานภายในอาคาร (building energy management systems, BEMS) และระบบการจัดการพลังงานภายในโรงงาน (factory energy management systems, FEMS) ที่มีจุดประสงค์ วิธีการโครงสร้างของระบบ รวมถึงผลลัพธ์ที่แตกต่างกันออกไป ซึ่งในปัจจุบันสิ่งปลูกสร้างประเภทอาคารมีสัดส่วนการบริโภคพลังงานไฟฟ้าโดยรวมของโลกสูงถึง 40% [6] และมีแนวโน้มเพิ่มสูงขึ้นสวนทางกับสิ่งปลูกสร้างประเภทที่อยู่อาศัยที่มีแนวโน้มการบริโภคพลังงานลดลง [7] โดยเมื่อพิจารณาร่วมกับข้อจำกัดของระบบการจัดการพลังงานประเภทโรงงานที่มีความซับซ้อนมากที่สุดเนื่องจากลักษณะการบริโภคพลังงานไฟฟ้าจะแตกต่างกันออกไปตามกระบวนการผลิตของแต่ละโรงงาน [8] จึงเกิดเป็น

กระแสผลักดันในเรื่องการจัดการพลังงานภายในอาคารอย่างมีประสิทธิภาพและยั่งยืน

ในบทความของบริษัทจัดการพลังงาน Prive [9] อธิบายถึงแนวทางการประหยัดพลังงานของสิ่งปลูกสร้างประเภทอาคารไว้สามวิธี ประกอบไปด้วย การปรับปรุงสิ่งปลูกสร้าง การติดตั้งระบบเฝ้าสังเกตเพื่อบริหารจัดการพลังงาน และการติดตั้งระบบควบคุมอาคารอัตโนมัติ (building automation system, BAS) จะเห็นว่าการปรับปรุงสิ่งปลูกสร้าง และการติดตั้งระบบควบคุมอาคารอัตโนมัติ เป็นวิธีการที่อาศัยเงินทุนจำนวนมาก อันเนื่องมาจากต้องมีการลงทุนในเรื่องการปรับเปลี่ยนโครงสร้างของอาคาร ตลอดจนอุปกรณ์ภายในอาคารเพื่อตอบสนองต่อการประหยัดพลังงาน นอกจาก่นั้นการติดตั้งระบบควบคุมอาคารอัตโนมัติต้องอาศัยระบบควบคุมที่ซับซ้อนสำหรับใช้ควบคุมอุปกรณ์ ซึ่งในความเป็นจริงสิ่งปลูกสร้างประเภทอาคารจะประกอบด้วยระบบที่แยกกันออกเป็นหลายส่วน ยกตัวอย่างเช่น ระบบแสงสว่าง (lighting system) ระบบปรับอากาศ (air conditioning system) ระบบระบายอากาศ (ventilation system) เป็นต้น ทั้งหมดนี้ในทางปฏิบัติอาจจะมีรูปแบบการทำงาน ความสามารถในการควบคุม รวมถึงโครงสร้างระบบที่แตกต่างกันออกไป เพื่อบรรลุเป้าหมายในหัวข้อการบริโภคพลังงานอย่างมีประสิทธิภาพ การควบคุมให้ทุกระบบสามารถติดต่อหรือแลกเปลี่ยนข้อมูลซึ่งกันและกันสำหรับใช้กำหนดลักษณะการทำงานแบบภาพรวมจึงเป็นสิ่งจำเป็น ซึ่งส่งผลโดยตรงต่อค่าใช้จ่ายในการติดตั้งระบบที่สูงขึ้น รวมถึงในบางกรณีเกิดเหตุการณ์ที่ระบบควบคุมอัตโนมัติรบกวนความสบายของผู้ใช้ (user comfort) อันเนื่องมาจากระบบควบคุมอัตโนมัติส่วนมากถูกตั้งค่าให้ทำตามตารางเวลา (scheduling) ซึ่งส่งผลโดยตรงต่อการยอมรับของผู้ใช้ [10] จากเหตุผลข้างต้นจะเห็นว่าการติดตั้งระบบเฝ้าสังเกตเพื่อบริหารจัดการพลังงาน เป็นวิธีที่เหมาะสม และสะดวกต่อการนำไปประยุกต์ใช้งานกับสิ่งปลูกสร้างประเภทอาคารที่มีอยู่เดิมโดยไม่จำเป็นต้องลงทุนมากเท่ากับการติดตั้งระบบควบคุมอาคารอัตโนมัติ

ระบบการจัดการพลังงานภายในอาคารที่ติดตั้งระบบเฝ้าสังเกตเพื่อตรวจวัดค่าการบริโภคพลังงานไฟฟ้ากับข้อมูลสภาพแวดล้อม และนำเข้าสู่ขั้นตอนการวิเคราะห์ข้อมูล (data analysis) ได้ผลลัพธ์เป็นข่าวสารไว้แสดงผล ซึ่งผลลัพธ์ที่ได้จากการวิเคราะห์มีหลากหลายรูปแบบตั้งแต่ข่าวสารเบื้องต้น เช่น ข่าวสารค่าพลังงานไฟฟ้า (energy consumption) ข่าวสารความต้องการพลังไฟฟ้า (demand power) ข่าวสารอุณหภูมิ (temperature) ในลักษณะกราฟบนแกนอนุกรมเวลา หรือในรูปแบบการสรุปเป็นช่วงเวลา (periodic summaries) ไปจนถึงข่าวสารเชิงลึกที่ต้องอาศัยกระบวนการวิเคราะห์ที่ซับซ้อน เช่น การพยากรณ์ความต้องการไฟฟ้า (load forecast) สำหรับจัดสรรแหล่งที่มาของพลังงานในช่วงเวลาต่าง ๆ ของวัน เป็นต้น นอกจากความแตกต่างของรูปแบบการวิเคราะห์ที่ให้ผลลัพธ์ออกมาในลักษณะที่แตกต่างกัน ประเภท จำนวน และคาบการตรวจวัดข้อมูลดิบ ล้วนเป็นส่วนสำคัญที่จะกำหนดรูปแบบของผลลัพธ์ สมรรถนะในการประมวลผล รวมถึงความแม่นยำในการวิเคราะห์ สอดคล้องกับทีม Navigant research ได้แจกแจงรูปแบบการวิเคราะห์ข้อมูลในระบบการจัดการพลังงานภายในอาคารเปรียบเทียบกับความซับซ้อนในการติดตั้งระบบ เพื่อใช้ประเมินศักยภาพของระบบการจัดการพลังงานภายในอาคาร [11] ดังแสดงในรูปที่ 1.1 กล่าวโดยสรุปคือความสามารถในการแก้ปัญหา (solution maturity) ของการวิเคราะห์แต่ละรูปแบบจะแปรผันตามความซับซ้อนของการออกแบบ และติดตั้งระบบเรียงลำดับตั้งแต่รูปแบบการวิเคราะห์ข้อมูลในลักษณะการสร้างมโนภาพข้อมูล (data visualization) เพื่อรายงานข่าวสาร (reporting) เป็นรูปแบบพื้นฐานของการวิเคราะห์ข้อมูลในระบบการจัดการพลังงานภายในอาคารที่ประสิทธิภาพ โดยอาศัยเพียงชุดข้อมูลที่ถูกตรวจวัดมารายงานผลในลักษณะทันทีทันใด หรือในลักษณะการสรุปแบบช่วงเวลาควบคู่ไปกับการสร้างมโนภาพข้อมูลที่ง่ายต่อการเข้าใจเพื่อให้ผู้ใช้อาคารทราบถึงลักษณะการบริโภคพลังงานของตนเอง ซึ่งแตกต่างจากการวิเคราะห์ข้อมูลในรูปแบบการพยากรณ์การซ่อมบำรุง (predictive maintenance) และ

การวิเคราะห์ข้อมูลในรูปแบบการตรวจสอบเหตุเสียและวินิจฉัยข้อบกพร่อง (fault detection and diagnostics) ที่อาศัยชุดประวัติข้อมูลหลายชนิดเพื่อศึกษาลักษณะการทำงานของแต่ละอุปกรณ์ในสภาพแวดล้อมต่าง ๆ เพื่อตรวจส อบประสิทธิภาพการทำงานเพื่อให้สะดวกต่อการวางแผนซ่อมบำรุง รวมถึงแจ้งเตือนในกรณีที่เกิดเหตุการณ์ผิดปกติ อย่างไรก็ตามรูปแบบของระบบการจัดการพลังงานภายในอาคารที่เหมาะสม และมีประสิทธิภาพของแต่ละอาคารจะแตกต่างกันออกไปขึ้นกับโครงสร้างและระบบที่มีอยู่เดิมของอาคาร ขอบเขตความต้องการของเจ้าของ งบประมาณในการติดตั้ง รวมถึงลักษณะการบริโภคพลังงานของผู้ใช้อาคาร



รูปที่ 1.1: การประเมินระบบการจัดการพลังงานภายในอาคารโดยทีม Navigant research [11]

เมื่อพิจารณาข้อจำกัดเรื่องความสามารถในการตรวจวัดข้อมูล และความหลากหลายของอุปกรณ์ภายในอาคารที่ได้กล่าวไว้ข้างต้น มีงานวิจัยในอดีตที่ได้พยายามออกแบบการวิเคราะห์ข้อมูลเพื่อใช้กับระบบการจัดการพลังงานโดยอาศัยชุดข้อมูลที่ง่ายต่อการตรวจวัด และไม่ผูกมัดกับชนิดของอุปกรณ์ แต่ได้ผลลัพธ์ออกมาในมุมมองรูปแบบใหม่ เช่น งานวิจัยของมหาวิทยาลัยอินเดียน่า [12] ใช้การสร้างมโนภาพข้อมูลในรูปแบบแผนที่ความร้อน (heat map) อาศัยการไล่ระดับสีในการแสดงถึงปริมาณกำลังงานไฟฟ้าในช่วงเวลาแต่ละช่วงของวัน สำหรับใช้ศึกษารูปแบบการบริโภคพลังงานเพื่อหาความสัมพันธ์ระหว่างกำลังงานสูงสุดในช่วงเวลาแต่ละช่วงของอาคารภายในมหาวิทยาลัยเพื่อใช้ลำดับความสำคัญในการควบคุมระบบในสถานการณ์ต่าง ๆ และในงานวิจัย [13] ประยุกต์ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ ควบคู่ไปกับข้อมูลความเข้มแสงเพื่อประเมินประสิทธิภาพระบบควบคุมแสงสว่างอัตโนมัติในรูปแบบตัวชี้วัดความสูญเปล่าของพลังงาน (wasted energy)

สถาบันวิจัยพลังงาน จุฬาลงกรณ์มหาวิทยาลัย คาดหวังที่จะจัดทำโครงการระบบการจัดการพลังงานจึงได้ร่วมมือกับภาควิชาวิศวกรรมศาสตร์ไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เพื่อออกแบบระบบการจัดการพลังงานไฟฟ้าภายในอาคารโดยอาศัยพื้นที่บริเวณภาควิชาวิศวกรรมไฟฟ้าเป็นพื้นที่ต้นแบบภายใต้ชื่อ Chulalongkorn University Building Energy Management system หรือ CUBEMS [14] โดยได้รับการถ่ายทอดเทคโนโลยีทางวิศวกรรมจากมหาวิทยาลัยโตเกียว

(University of Tokyo) ด้านการจัดการพลังงานภายในอาคารด้วยมาตรฐาน IEEE1888 [15] ซึ่งเป็นมาตรฐานที่ได้มีการนำไปประยุกต์ใช้ในโครงการมหาวิทยาลัยโตเกียวสีเขียว (Green University of Tokyo Project, GUTP) [16] โดยประยุกต์มาตรฐาน IEEE1888 เข้ากับระบบการจัดการพลังงานที่ติดตั้งอยู่เดิมภายในมหาวิทยาลัย เป็นผลให้สามารถเชื่อมโยงเทคโนโลยีต่างชนิดเข้าด้วยกันและสามารถสื่อสารเพื่อทำงานร่วมกันได้อย่างมีประสิทธิภาพ ทำให้มหาวิทยาลัยโตเกียวสามารถลดการบริโภคพลังงานไฟฟ้าได้ถึง 31% ทั้งนี้ทางภาควิชาวิศวกรรมไฟฟ้าได้ตั้งจุดประสงค์ของโครงการ CUBEMS คือใช้สำหรับเฝ้าสังเกตการบริโภคพลังงาน และสภาพแวดล้อมภายในอาคาร โดยออกแบบมาตราวัดอัจฉริยะ [17] ตัวรับรู้สภาพแวดล้อม [18] และจอแสดงผลภายในอาคาร [19] ผนวกเข้ากับมาตรฐาน IEEE1888 ตรวจวัด และนำข้อมูลมาแสดงผลในรูปแบบต่าง ๆ เพื่อสร้างการตระหนักรู้แก่ผู้ใช้อาคาร คาดหวังให้เกิดการปรับเปลี่ยนพฤติกรรมในเรื่องการบริโภคพลังงานไปในแนวทางที่มีประสิทธิภาพ

อย่างไรก็ตาม จากการประเมินผลทดสอบของผู้ใช้จอแสดงผลภายในอาคาร รวมถึงค่าผลลัพธ์การบริโภคพลังงานไฟฟ้าของโครงการ CUBEMS พบว่าการคาดหวังให้ผู้ใช้อาคารปรับเปลี่ยนพฤติกรรมการบริโภคพลังงานหลังรับรู้ข่าวสารค่อนข้างเป็นไปได้ยาก ซึ่งอาจจะเกิดจากปัจจัย ยกตัวอย่างเช่น อุปนิสัยของผู้ใช้อาคาร ช่องทางการแสดงข่าวสารที่เข้าถึงได้ยาก หรือรูปแบบของข่าวสารที่นำเสนอยากต่อการทำความเข้าใจและไม่ชัดเจนพอที่จะโน้มน้าวให้ผู้ใช้อาคารเกิดการตระหนักรู้ จากข้อจำกัดของระบบที่ได้พัฒนาไว้เพื่อเป็นระบบตั้งต้นในโครงการ CUBEMS ดังกล่าว วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอการออกแบบโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลในระบบการจัดการพลังงานภายในอาคารโดยเลือกใช้คลัง (โปรแกรม) Pandas ที่ถูกพัฒนาขึ้นบนภาษา python ในการจัดการข้อมูลที่เก็บในรูปแบบวันที่และเวลา และอาศัยทรัพยากรข้อมูลของโครงการ CUBEMS ตามมาตรฐาน IEEE1888 เพื่อพัฒนาต้นแบบทดสอบขึ้น ในงานวิจัยที่ดำเนินการนี้มีเป้าหมายที่จะเพิ่มขีดความสามารถด้านการสร้างมโนภาพข้อมูลแก่ผู้ใช้อาคาร โดยอาศัยการวิเคราะห์สหสัมพันธ์ (correlations analysis) ระหว่างข้อมูลพฤติกรรมของผู้ใช้ร่วมกับข้อมูลประเภทต่าง ๆ ให้เกิดเป็นข่าวสารเพื่อสร้างการตระหนักรู้ให้แก่ผู้ใช้อาคาร ซึ่งในอนาคตคาดหวังให้เกิดเป็นการปรับตัวระยะสั้นในลักษณะการปรับเปลี่ยนเปลี่ยนพฤติกรรม (habits modification) หรือในลักษณะการปรับตัวระยะยาวในรูปแบบนโยบาย (policy) ของการบริโภคพลังงานภายในอาคารได้ต่อไป

1.2 วัตถุประสงค์ของงานวิทยานิพนธ์

พัฒนาโปรแกรมประยุกต์สำหรับการวิเคราะห์ข้อมูลเพื่อเพิ่มขีดความสามารถด้านการสร้างมโนภาพในส่วนความสัมพันธ์ระหว่างข้อมูลแต่ละประเภท และผสานเข้ากับระบบการจัดการพลังงานภายในอาคาร บริเวณภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย (CUBEMS) บนพื้นฐานการติดต่อสื่อสารข้อมูลต่าง ๆ ด้วยมาตรฐาน IEEE1888

1.3 ขอบเขตวิทยานิพนธ์

1. ออกแบบโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลบนภาษา python [20] และใช้คลัง (โปรแกรม) Pandas [21]
2. การสื่อสารระหว่างโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลกับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 [15] ใช้เพียงโพรโทคอล FETCH และ WRITE

3. ออกแบบฟังก์ชันของโปรแกรมประยุกต์เพื่อใช้ในการสร้างมโนภาพของข้อมูลบนภาษา python และใช้คลัง (โปรแกรม) plotly [22]
4. ออกแบบฟังก์ชันของโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลการบริโภคพลังงานร่วมกับข้อมูลพฤติกรรมของผู้ใช้
5. ออกแบบการผสานผลลัพท์ที่ได้เข้ากับจอแสดงผลภายในอาคาร [19] ของโครงการ CUBEMS
6. ทดสอบและประเมินการใช้งานโปรแกรมประยุกต์

1.4 ขั้นตอน และวิธีดำเนินงาน

1. ศึกษาบทความ งานวิจัย และวิทยานิพนธ์ในอดีตที่เกี่ยวข้องกับการวิเคราะห์ข้อมูล และการสร้างมโนภาพในระบบบริหารจัดการพลังงานภายในอาคาร
2. ศึกษาโครงการ CUBEMS และมาตรฐาน IEEE1888
3. ศึกษาคุณลักษณะของภาษา python รวมถึงคลัง (โปรแกรม) Pandas และ plotly
4. ออกแบบ และสร้างโปรแกรมประยุกต์สำหรับร้องขอ และวิเคราะห์ข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888
5. ออกแบบ และสร้างฟังก์ชันของโปรแกรมประยุกต์เพื่อใช้ในการสร้างมโนภาพในโครงการ CUBEMS
6. วิเคราะห์ และประเมิณประสิทธิภาพของโปรแกรมประยุกต์
7. เผยแพร่การวิจัยในบทความ และจัดทำเล่มวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. องค์ความรู้ในการพัฒนาโปรแกรมวิเคราะห์ข้อมูล และการสร้างมโนภาพให้ตอบสนองต่อความคาดหวังที่จะเพิ่มความเข้าใจของผู้ใช้ในเรื่องพฤติกรรมการบริโภคพลังงาน สำหรับนำไปใช้ในการวางแผนบริหารจัดการพลังงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้นผ่านการสร้างมโนภาพ ในส่วนการแสดงความสัมพันธ์ข้อมูลหลากประเภทในหลาย ๆ มุมมอง
2. ต้นแบบโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลที่สามารถทำงานร่วมกับหน่วยเก็บข้อมูล และจอแสดงผลภายในโครงการ CUBEMS ตามมาตรฐาน IEEE1888
3. ต้นแบบฟังก์ชันวิเคราะห์ข้อมูลของโปรแกรมประยุกต์สำหรับระบบการจัดการพลังงานภายในอาคารที่อาศัยการวิเคราะห์สหสัมพันธ์ระหว่างข้อมูลพฤติกรรมของผู้ใช้กับข้อมูลการบริโภคพลังงานภายในระบบปรับอากาศ

1.6 ประมวลวิทยานิพนธ์

บทที่ 1 บทนำ : กล่าวถึงความเป็นมา และความสำคัญของการพัฒนาโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลในระบบการจัดการพลังงานภายในอาคาร รวมถึงนำเสนองานวิจัยในอดีตที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลในระบบการจัดการพลังงานภายในอาคาร วัตถุประสงค์ ขอบเขต ขั้นตอน และวิธีการดำเนินงานของวิทยานิพนธ์ และกล่าวถึงประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีพื้นฐาน : กล่าวถึงทฤษฎี และหลักการพื้นฐานต่าง ๆ ที่นำมาใช้ในวิทยานิพนธ์ฉบับนี้ ซึ่งประกอบด้วย มาตรฐาน IEEE1888 ระบบการจัดการพลังงานภายในอาคารโครงการ CUBEMS และตัวอย่างการสร้างมโนภาพข้อมูลในระบบการจัดการพลังงานภายในอาคาร

บทที่ 3 สถาปัตยกรรมของระบบและการทดสอบเบื้องต้น : กล่าวถึงโครงสร้าง การทำงาน และการประเมินประสิทธิภาพการทำงานของโปรแกรมประยุกต์ต้นแบบในหัวข้อการเชื่อมต่อเข้ากับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 แนวทางการออกแบบ และการพัฒนาส่วนต่อประสาน รวมถึงส่วนการสร้างมโนภาพข้อมูลของโปรแกรมประยุกต์ข้างต้น

บทที่ 4 การสร้างและทดสอบฟังก์ชันวิเคราะห์ข้อมูลของโปรแกรมประยุกต์ : กล่าวถึงแนวทางการออกแบบ การทดสอบ และการประยุกต์ใช้งานจริงของฟังก์ชันวิเคราะห์ข้อมูลที่ถูกพัฒนาขึ้น โดยมีการทดสอบในรูปแบบต่าง ๆ เพื่อศึกษาถึงพฤติกรรมของการบริโภคพลังงานของผู้ใช้อาคารภายในพื้นที่บริเวณทดสอบและการส่งผลของตัวแปรภายนอกต่อผลลัพธ์ของฟังก์ชันวิเคราะห์ข้อมูล รวมถึงมีการประเมินสมรรถนะการทำงานของฟังก์ชันวิเคราะห์ข้อมูลในรูปแบบต่าง ๆ เพื่อเป็นข้อมูลตอนนำไปประยุกต์ใช้จริง

บทที่ 5 บทสรุปและข้อเสนอแนะ : กล่าวถึงบทสรุปของงานวิจัยทั้งหมดในวิทยานิพนธ์ฉบับนี้ และเสนอแนวทางในการพัฒนางานวิจัยต่อไป

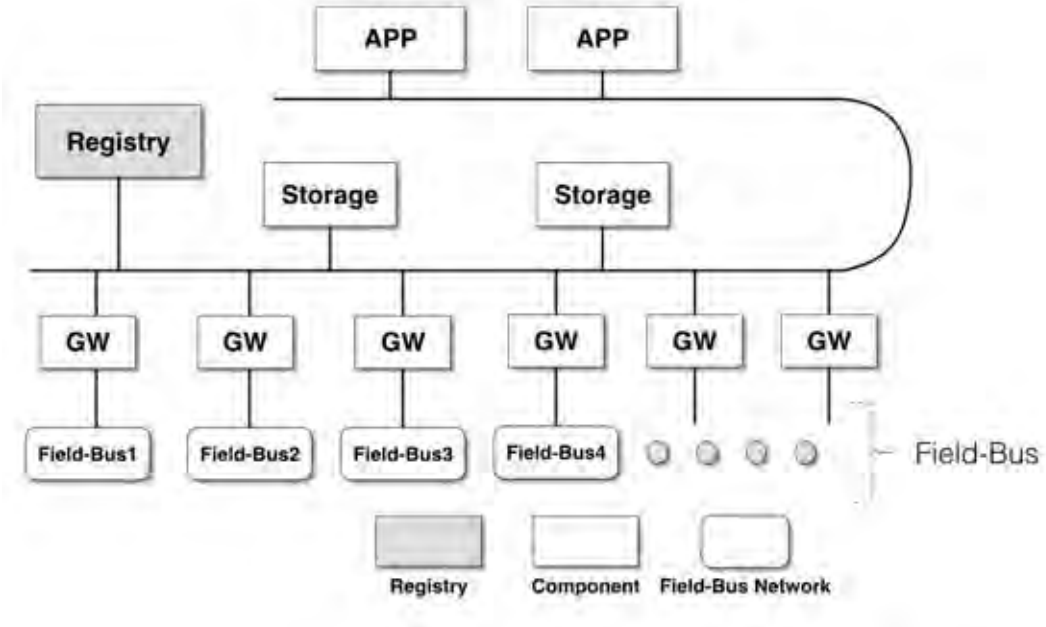
บทที่ 2
ทฤษฎีพื้นฐาน

2.1 มาตรฐาน IEEE1888

มาตรฐาน IEEE1888 ได้ถูกนำเสนอขึ้นในปี ค.ศ. 2011 โดยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (institute of electrical and electronics engineers, IEEE) เป็นมาตรฐานใหม่ที่เกี่ยวข้องกับการสื่อสารสำหรับระบบการจัดการพลังงาน โดยใช้ชื่อว่าโพรโทคอลโครงข่ายการควบคุมชุมชนสีเขียวอย่างแพร่หลาย (ubiquitous greencommunity control network protocol, UGCCNet) หรือ IEEE1888 มีวัตถุประสงค์เพื่อแก้ไข้ข้อจำกัดในเรื่องความหลากหลายของมาตรฐานในอุปกรณ์ให้สามารถเชื่อมโยงเพื่อแลกเปลี่ยนข้อมูลระหว่างกันได้ และประยุกต์ใช้ในระบบการจัดการพลังงานภายในอาคาร

2.1.1 สถาปัตยกรรม IEEE1888

สถาปัตยกรรมมาตรฐาน IEEE1888 จำแนกตามลักษณะการทำงานออกเป็น 2 ระนาบคือ ระนาบข้อมูล (data plane) และระนาบควบคุม (control plane) โดยในระนาบข้อมูลแบ่งเป็น 3 องค์ประกอบที่มีส่วนต่อประสาน (interface) แบบเดียวกันคือ เกตเวย์ หน่วยเก็บข้อมูล และโปรแกรมประยุกต์ สำหรับระนาบควบคุมจะประกอบไปด้วย รีจิสทรี (registry) ซึ่งทั้งสองระนาบนี้จะสื่อสารกันบนพื้นฐานของโครงข่ายทีซีพี/ไอพีดังแสดงในรูปที่ 2.1 และองค์ประกอบต่าง ๆ มีหน้าที่หลักดังนี้



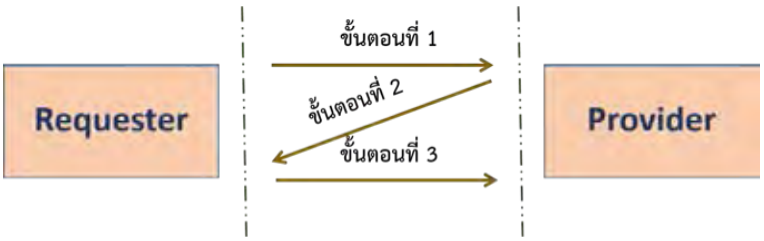
รูปที่ 2.1: สถาปัตยกรรม IEEE1888 [15]

1. เกตเวย์ ทำหน้าที่เปรียบเสมือนตัวกลางในกลางเชื่อมต่อระหว่างโครงข่ายทีซีพี/ไอพี กับโครงข่ายฟิลด์บัส (field-bus) ซึ่งจะเชื่อมโยงต่อไปยังอุปกรณ์อื่น ๆ เช่น ตัวรับรู้ (sensor) และตัวกระตุ้น (actuator) ตามมาตรฐานการสื่อสารของอุปกรณ์
2. หน่วยเก็บข้อมูล ทำหน้าที่บันทึกประวัติข้อมูล
3. โปรแกรมประยุกต์ ทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้งานเพื่อแสดงผลข้อมูล
4. รีจิสทรี ทำหน้าที่เป็นตัวกลางควบคุมการทำงานการเชื่อมต่อขององค์ประกอบข้างต้นทั้งสามส่วน

2.1.2 การสื่อสารพื้นฐานในมาตรฐาน IEEE1888

การสื่อสารกันระหว่างองค์ประกอบต่าง ๆ ในมาตรฐาน IEEE1888 จะแยกออกเป็นสองแบบ คือ การสื่อสารกันระหว่างองค์ประกอบ (component-to-component communication protocol) และการสื่อสารกันระหว่างองค์ประกอบกับรีจิสทรี (component-to-registry comunication protocol) โดยข้อความสื่อสารด้วยโพรโทคอลเข้าถึงอ็อบเจกต์อย่างง่าย (simple object access protocol, SOAP) ซึ่งประกอบด้วยโพรโทคอลย่อยทั้งสิ้น 3 ชนิด คือ

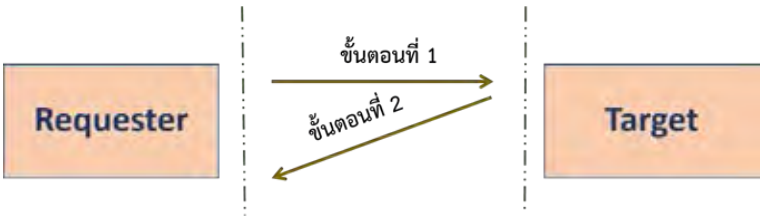
1. โพรโทคอล FETCH ใช้สำหรับร้องขอข้อมูลจากองค์ประกอบอื่น มีการทำงานตามรูปที่ 2.2 ซึ่งแสดงการรับส่งข้อมูลระหว่างผู้ร้องขอข้อมูลกับผู้ให้บริการข้อมูล



รูปที่ 2.2: โพรโทคอล FETCH

- ขั้นตอนที่ 1 : ผู้ร้องขอส่งข้อมูลการร้องขอไปยังผู้ให้บริการข้อมูล
- ขั้นตอนที่ 2 : ผู้ให้บริการข้อมูลตอบกลับข้อมูลทั้งหมดที่ได้ร้องขอเข้ามาถ้าข้อมูลมีขนาดเล็ก แต่ถ้าข้อมูลมีขนาดใหญ่จนผู้ร้องขอไม่สามารถรองรับได้ในการส่งครั้งเดียว ทางผู้ให้บริการข้อมูลจะส่งข้อมูลเป็นช่วงของข้อมูล โดยในแต่ละครั้งจะมีการส่งตัวชี้ตำแหน่ง (cursor) มาเพื่อใช้เป็นตัวระบุรอยต่อของข้อมูล
- ขั้นตอนที่ 3 หากข้อมูลที่ส่งกลับมาในขั้นตอนที่ 2 มีการแนบตัวชี้ตำแหน่ง แสดงให้เห็นว่าทางผู้ให้บริการข้อมูลไม่สามารถตอบกลับข้อมูลมาได้ทั้งหมดตามที่ร้องขอ ด้านผู้ร้องขอจะร้องขอข้อมูลต่อโดยแนบตัวชี้ตำแหน่งเข้าไปกับการร้องขอข้อมูลครั้งต่อไป และรอรับข้อมูลที่เหลืออยู่จนกระทั่งได้รับข้อมูลที่ร้องขอครบถ้วน

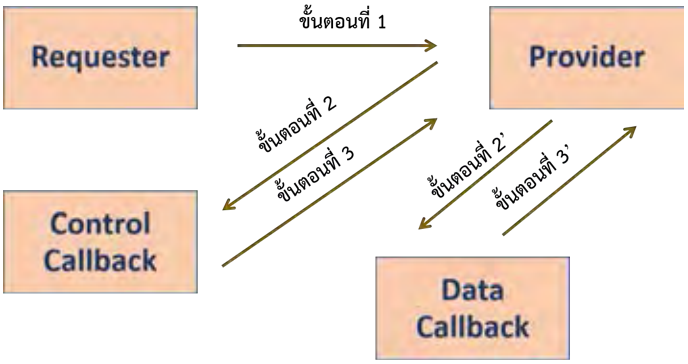
2. โพรโทคอล WRITE ใช้สำหรับถ่ายโอนข้อมูลไปยังองค์ประกอบอื่น มีขั้นตอนการทำงานตามรูปที่ 2.3 ซึ่งแสดงการรับส่งข้อมูลระหว่างผู้ร้องขอข้อมูล และเป้าหมาย



รูปที่ 2.3: โพรโทคอล WRITE

- ขั้นตอนที่ 1 : ผู้ร้องขอข้อมูลส่งข้อมูลพร้อมคำสั่งเขียนข้อมูลไปยังเป้าหมาย
- ขั้นตอนที่ 2 : เป้าหมายตอบกลับไปยังผู้ร้องขอว่ากระบวนการสำเร็จหรือล้มเหลว

3. โพรโทคอล TRAP (TRAP protocol) ใช้สำหรับแจ้งเตือนเมื่อเกิดเหตุการณ์ที่ได้จดทะเบียนไว้ โดยมีองค์ประกอบที่เกี่ยวข้องตามรูปที่ 2.4



รูปที่ 2.4: โพรโทคอล TRAP

- ขั้นตอนที่ 1 : ผู้ร้องขอข้อมูลจดทะเบียนเหตุการณ์ที่สนใจโดยส่งคำสั่งร้องขอไปยังผู้ให้บริการข้อมูล พร้อมกับระบุอายุของคำสั่งในหน่วย วินาที รวมทั้งระบุยูอาร์ไอของผู้เรียกกลับข้อมูลและผู้เรียกกลับของส่วนควบคุม
- ขั้นตอนที่ 2 : ผู้ให้บริการข้อมูลตอบกลับไปยังยูอาร์ไอของผู้เรียกกลับข้อมูล เมื่อเกิดเหตุการณ์ที่ตรงกับเหตุการณ์ที่ได้จะทะเบียนไว้
- ขั้นตอนที่ 3 : ผู้เรียกกลับข้อมูลทำการตอบกลับไปยังผู้ให้บริการข้อมูลว่าการส่งข้อมูลสำเร็จหรือเกิดข้อผิดพลาด
- ขั้นตอนที่ 2' : เมื่อเกิดข้อผิดพลาดในขั้นตอนที่ 3 ผู้ให้บริการข้อมูลจะประกาศข้อผิดพลาดของการสื่อสารไปยังยูอาร์ไอของผู้เรียกกลับส่วนควบคุม
- ขั้นตอนที่ 3' : ผู้เรียกกลับส่วนควบคุมตอบกลับไปยังผู้ให้บริการข้อมูลว่าการส่งข้อมูลสำเร็จหรือเกิดข้อผิดพลาด

2.2 ระบบการจัดการพลังงานภายในอาคาร จุฬาลงกรณ์มหาวิทยาลัย

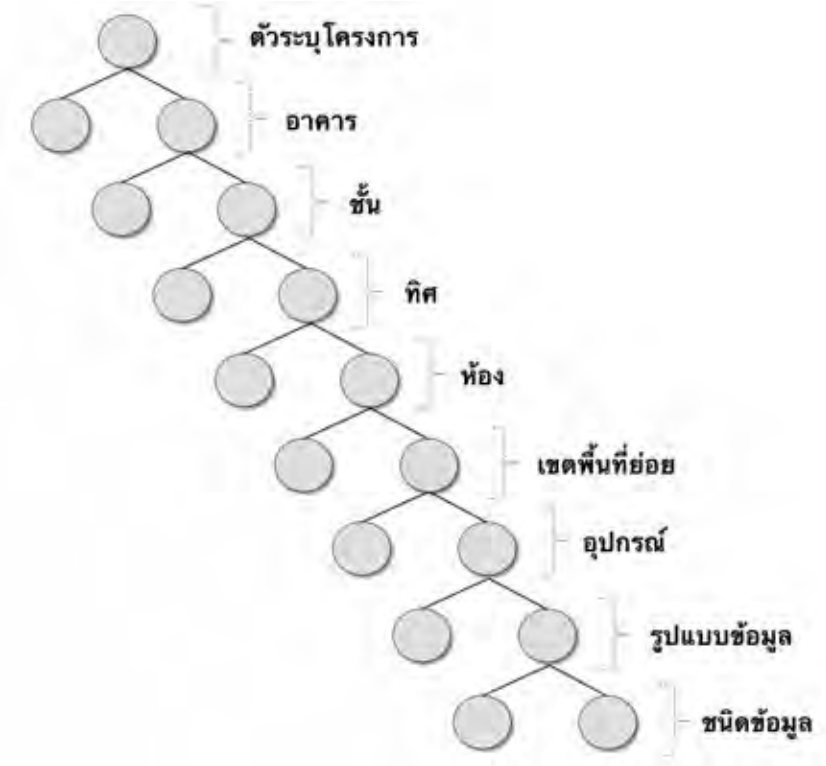
โครงการ CUBEMS มีจุดประสงค์เพื่อเฝ้าสังเกตการบริโภคพลังงาน และข้อมูลสภาพแวดล้อมภายในอาคาร บริเวณภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย โดยนำสถาปัตยกรรม และโพรโทคอลสื่อสารตามมาตรฐาน IEEE1888.1 มาประยุกต์ใช้ ประกอบไปด้วย เกตเวย์ หน่วยเก็บข้อมูล และโปรแกรมแกรมประยุกต์ตามมาตรฐาน IEEE1888 [15] รวมถึงมีการออกแบบมาตรวัดอัจฉริยะ [17] เกตเวย์ตัวรับรู้ และตัวรับรู้สภาพแวดล้อม [18] ดังแสดงในรูปที่ 2.5 - 2.7 ซึ่งตัวรับรู้สภาพแวดล้อมประกอบไปด้วยตัวรับรู้ 4 ชนิด คือ ตัวรับรู้อุณหภูมิ ตัวรับรู้ความชื้นสัมพัทธ์ ตัวรับรู้ความเข้มแสง และตัวรับรู้การเคลื่อนไหวของคน โดยมีรายละเอียดชนิด ลักษณะการตรวจวัดแสดงในตารางที่ 2.1 ข้อมูลที่ถูกตรวจวัดจะถูกส่งไปยังเกตเวย์ที่ติดตั้งในบริเวณรอบ ๆ ผ่านโครงข่ายเทคโนโลยีไร้สาย zigbee และมีการระบุแหล่งที่มาในรูปแบบตัวชี้แหล่งในอินเทอร์เน็ต (uniform resource locator, URL) หรือยูอาร์แอลที่เรียกว่า PointID ที่มีโครงสร้างในลักษณะต้นไม้ (tree) ดังแสดงในรูปที่ 2.9 และ 2.10 ลำดับถัดไปเกตเวย์จะทำการส่งข้อมูลตามมาตรฐาน IEEE1888 เข้าไปที่หน่วยเก็บข้อมูลของโครงการที่ถูกติดตั้งไว้บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร ชั้น 13 อาคารเจริญวิศวกรรม ดังแสดงในรูปที่ 2.8 สำหรับข้อมูลการบริโภคพลังงานจะแบ่งออกเป็นสามส่วน ประกอบไปด้วย ระบบปรับอากาศ ระบบแสงสว่าง และระบบเต้าเสียบ ซึ่งจำนวนของ PointID ที่ระบุถึงระบบข้างต้นจะแตกต่างกันออกไปในพื้นที่แต่ละบริเวณดังแสดงในตารางที่ 2.2

รูปที่ 2.5: มาตราวัดอัจฉริยะของโครงการ CUBEMS

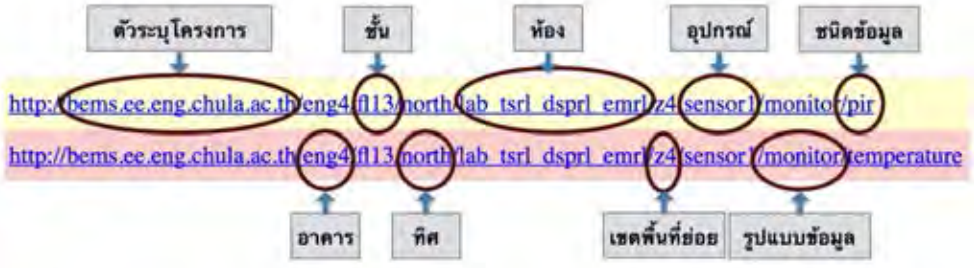
รูปที่ 2.6: ตัวรับรู้สภาพแวดล้อมของโครงการ CUBEMS

รูปที่ 2.7: เกตเวย์ตัวรับรู้สภาพแวดล้อมของโครงการ CUBEMS

รูปที่ 2.8: หน่วยเก็บข้อมูลของโครงการ CUBEMS



รูปที่ 2.9: โครงสร้าง PointID ของโครงการ CUBEMS



รูปที่ 2.10: ตัวอย่าง PointID ของโครงการ CUBEMS

ตารางที่ 2.1: รายละเอียดชนิดข้อมูลที่ถูกตรวจวัดในโครงการ CUBEMS

แหล่งที่มา	รายละเอียดเข้อมูล	หน่วย	ลักษณะการตรวจวัด
มาตรวัดอัจฉริยะ	ค่าพลังงาน	วัตต์ชั่วโมง	รายคาบ 1 นาที
ตัวรับรู้อุณหภูมิ	ค่าอุณหภูมิ	องศาเซลเซียส	รายคาบ 1 นาที
ตัวรับรู้ความชื้นสัมพัทธ์	ค่าความชื้นสัมพัทธ์	%	รายคาบ 1 นาที
ตัวรับรู้ความเข้มแสง	ค่าความเข้มแสง	ลูเมน	รายคาบ 1 นาที
ตัวรับรู้การเคลื่อนไหวของคน	ค่าการตรวจจับการเคลื่อนไหว ON/OFF	-	รายคาบ และเมื่อมีการเปลี่ยนสถานะ

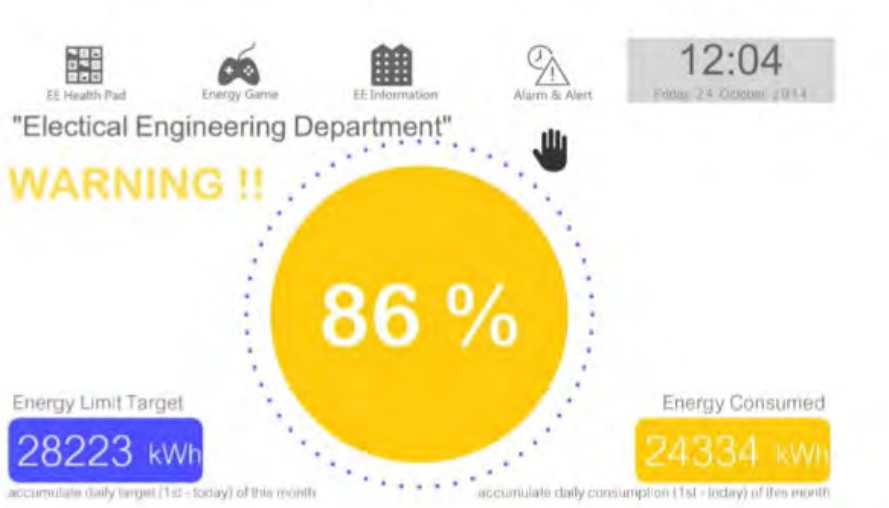
ตารางที่ 2.2: รายละเอียด PointID ในพื้นที่แต่ละบริเวณของโครงการ CUBEMS

ชนิด PointID	อาคารเจริญวิศวกรรม	อาคารภาควิชาวิศวกรรมไฟฟ้า	อาคารไฟฟ้าแรงสูง	อาคารชารลเอมสันเกเวอรด	ห้องสมุดอิเล็กทรอนิกส์และไมโครคอมพิวเตอร์
ระบบปรับอากาศ	73	60	1	1	1
ระบบแสงสว่าง	38	57			
ระบบเต้าเสียบ	41	86			
ตัวรับรู้อุณหภูมิ	103	78			
ตัวรับรู้ความชื้นสัมพัทธ์	103	78			
ตัวรับรู้ความเข้มแสง	103	78			
ตัวรับรู้การเคลื่อนไหวของคน	101	0			
รวม	567	467	1	1	1

นอกจากนี้โครงการ CUBEMS ได้ออกแบบช่องทางการแสดงผลไว้ทั้งสิ้นสามช่องทางประกอบไปด้วย การแสดงผลบนเว็บไซต์ การแสดงผลบนโปรแกรมประยุกต์ระบบปฏิบัติการแอนดรอยด์ และการแสดงผลบนจอแสดงผลภายในโครงการ [19] ดังแสดงในรูปที่ 2.11 - 2.13 ตามลำดับ

รูปที่ 2.11: การแสดงผลบนเว็บไซต์ของโครงการ CUBEMS

รูปที่ 2.12: การแสดงผลบนโปรแกรมประยุกต์ระบบปฏิบัติการแอนดรอยด์โครงการ CUBEMS

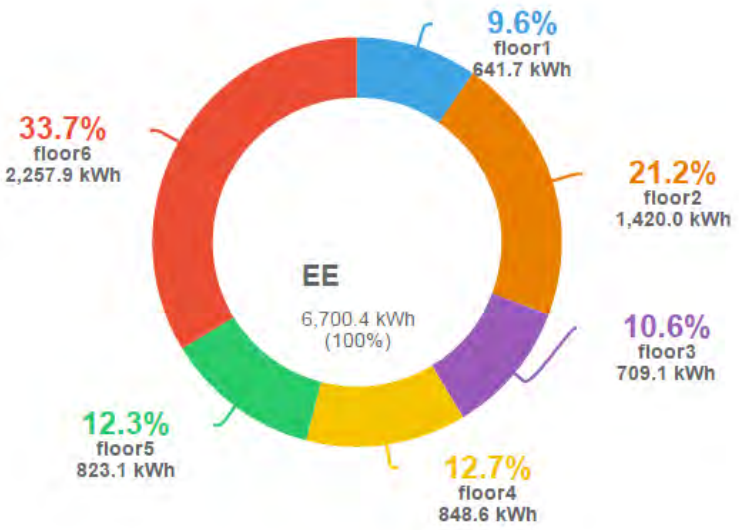


รูปที่ 2.13: จอแสดงผลของโครงการ CUBEMS [19]

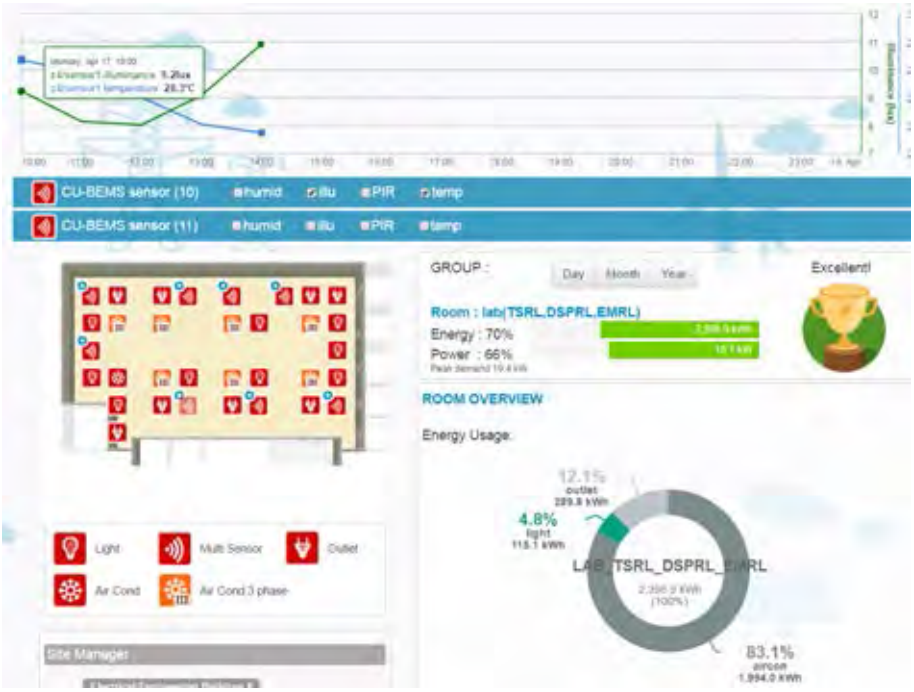
2.3 การสร้างมโนภาพข้อมูลในระบบการจัดการพลังงานภายในอาคาร

การสร้างมโนภาพของข้อมูล คือ เทคนิคการนำเสนอข้อมูลที่เป็นตัวเลข หรือตัวอักษรออกมาในลักษณะ ภาพ แผนผัง หรือภาพการเคลื่อนไหว มีจุดประสงค์เพื่อแสดงความสัมพันธ์ระหว่างข้อมูลเกิดเป็นมุมมองรูปแบบใหม่ [23] เช่น โครงการมหาวิทยาลัยโตเกียวสีเขียวมีการสร้างมโนภาพเพื่อแสดงข้อมูลการบริโภคพลังงานในรูปแบบกราฟแท่งบนแกนอนุกรมเวลาดังแสดงในรูปที่ 2.14 เช่นเดียวกับโปรแกรมประยุกต์ของโครงการ CUBEMS มีรูปแบบการนำเสนอโดยอาศัยกราฟเส้น แผนภูมิแท่ง แผนภูมิวงกลม เพื่อนำเสนอข้อมูลการบริโภคพลังงานรวมถึงข้อมูลสภาพแวดล้อมของอาคารแยกตามชั้น ห้อง และเขตพื้นที่ ร่วมกับการใช้ระดับสีเพื่อบ่งบอกถึงปริมาณการบริโภคพลังงานเป็นสัดส่วนเทียบกับค่าขีดสุด (threshold) ที่ถูกตั้งไว้เพื่อดึงดูดความสนใจของผู้ใช้อาคารดังแสดงในรูปที่ 2.15 - 2.17

รูปที่ 2.14: การสร้างมโนภาพข้อมูลการบริโภคพลังงานของโครงการมหาวิทยาลัยโตเกียวสี [16]



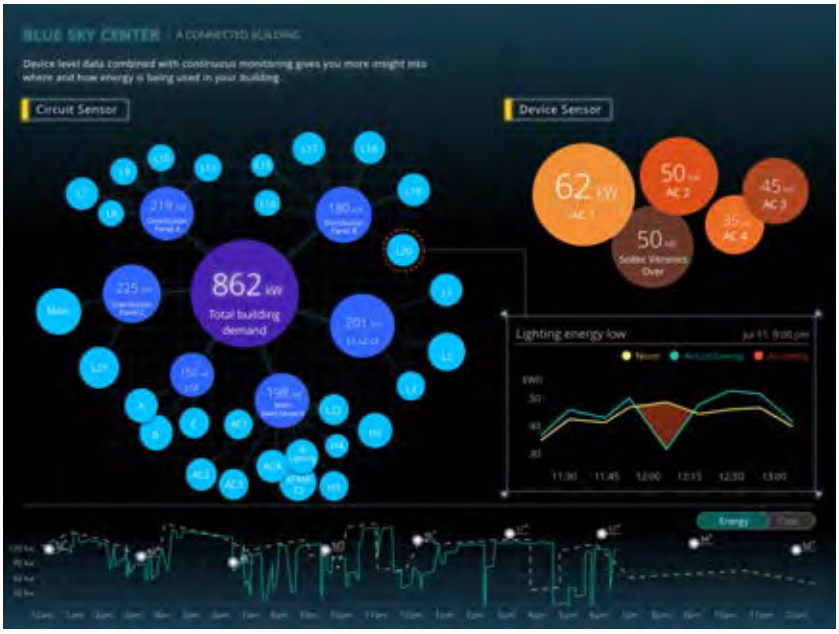
รูปที่ 2.15: การสร้างมโภาพข้อมูลเพื่อนำเสนอสัดส่วนการบริโภคพลังงานในรูปแบบแผนภูมิวงกลมของโครงการ CUBEMS



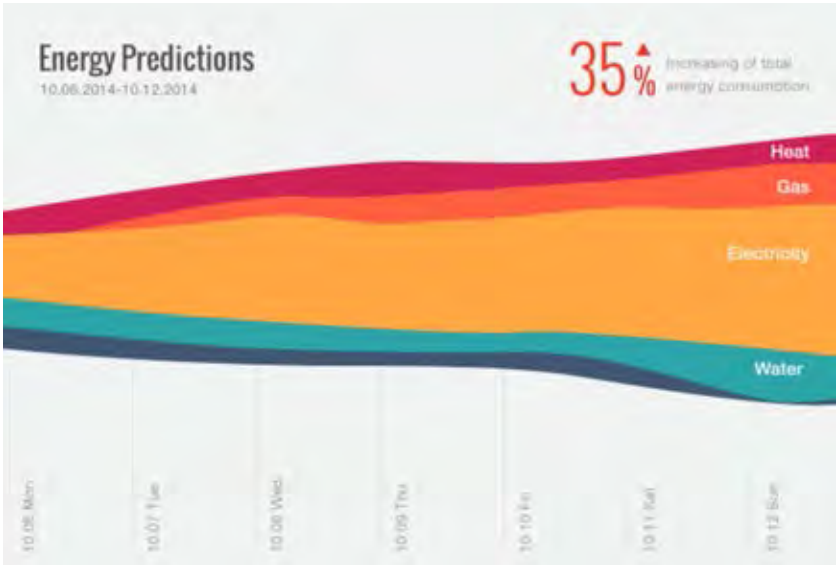
รูปที่ 2.16: การสร้างมโภาพข้อมูลเพื่อนำเสนอข้อมูลสภาพแวดล้อมในระดับเขตพื้นที่ของโครงการ CUBEMS

รูปที่ 2.17: การสร้างมโภาพข้อมูลร่วมกับการใช้ระดับสีเพื่อบ่งบอกถึงสัดส่วนการบริโภคพลังงานเทียบกับค่าขีดสุดของโครงการ CUBEMS

แผงหน้าปัดในรูปที่ 2.18 สำหรับใช้ในระบบการจัดการพลังงานภายในอาคารที่ถูกออกแบบโดยบริษัท Verdigris [24] มีการสร้างมโนภาพข้อมูลเพื่อแสดงข้อมูลการบริโภคพลังงานตั้งแต่ระดับภาพรวมโครงการ รายอาคาร แยกย่อยจนถึงระดับอุปกรณ์ และในรูปที่ 2.19 แสดงการสร้างมโนภาพที่ถูกออกแบบโดยบริษัท Ramotion [25] สำหรับข้อมูลพยากรณ์การบริโภคพลังงานในช่วงเวลาต่าง ๆ ของวันดังแสดงในรูปที่ 2.19



รูปที่ 2.18: แผงหน้าปัดระบบการจัดการพลังงานภายในอาคารบริษัท Verdigris [24]



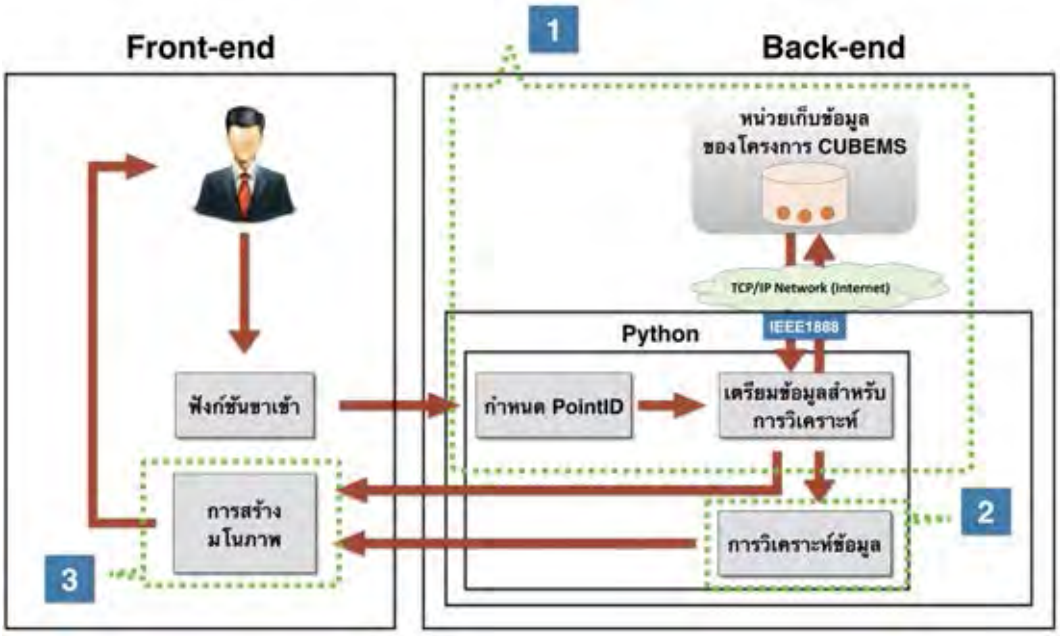
รูปที่ 2.19: การสร้างมโนภาพข้อมูลการบริโภคพลังงานของบริษัท Ramotion [25]

บทที่ 3

สถาปัตยกรรมของระบบและการทดสอบเบื้องต้น

3.1 สถาปัตยกรรมของระบบ

สถาปัตยกรรมของระบบที่ใช้ในการพัฒนาโปรแกรมประยุกต์สำหรับวิเคราะห์ข้อมูลโดยแบ่งออกเป็นสามส่วนดังแสดงในรูปที่ 3.1 ประกอบไปด้วย



รูปที่ 3.1: สถาปัตยกรรมของระบบ

1. ส่วนต่อประสาน มีหน้าที่รับคำสั่งจากผู้ใช้สำหรับตีความเพื่อกำหนดชุด PointID ที่จำเป็นในการเรียกใช้ฟังก์ชัน เพื่อร้องขอข้อมูลจากหน่วยเก็บข้อมูลโครงการ CUBEMS
2. ส่วนการวิเคราะห์ข้อมูล มีหน้าที่นำข้อมูลที่ถูกส่งต่อจากส่วนต่อประสานเข้าสู่ขั้นตอนวิเคราะห์ข้อมูลตามรูปแบบของฟังก์ชันที่ได้รับคำสั่งมา
3. ส่วนการสร้างมโนภาพข้อมูล มีหน้าที่นำผลลัพธ์จากกระบวนการข้างต้นมาสร้างมโนภาพข้อมูลเพื่อแสดงผล

เมื่อผู้ใช้งานเลือกรูปแบบฟังก์ชันการวิเคราะห์ข้อมูล โปรแกรมประยุกต์จะส่งคำสั่งเข้ามาที่ส่วนต่อประสานเพื่อระบุ PointID ของข้อมูลที่จำเป็นต้องใช้ในการวิเคราะห์ สำหรับการร้องขอข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 อาศัยโพรโทคอล FETCH ที่ถูกเขียนขึ้นในโปรแกรมประยุกต์บนพื้นฐานภาษา python รวมถึงจัดรูปแบบให้อยู่ในลักษณะเฟรมข้อมูลเพื่อให้สะดวกต่อขั้นตอนการวิเคราะห์ข้อมูลในลำดับถัดไป โดยผลลัพธ์ที่ได้การวิเคราะห์จะถูกเขียนกลับเข้าไปที่หน่วยเก็บข้อมูลโครงการ CUBEMS อาศัยโพรโทคอล WRITE ตามมาตรฐาน IEEE1888 เพื่อ

ลดความซ้ำซ้อนในการวิเคราะห์ชุดข้อมูลเดิม รวมถึงเพิ่มความหลากหลายของช่องทางในการแสดงผลลัพธ์ที่ได้จากกระบวนการวิเคราะห์ข้อมูล ในส่วนของการสร้างมโนภาพข้อมูลจะมีรูปแบบผลลัพธ์ที่แตกต่างกันออกไปขึ้นกับฟังก์ชันการวิเคราะห์ที่เลือกใช้ และช่องทางการแสดงผลลัพธ์ซึ่งประกอบไปด้วย จอแสดงผลภายในโครงการ CUBEMS [19] เว็บไซต์การวิเคราะห์ข้อมูล และการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์

3.2 การทดสอบเชื่อมต่อโปรแกรมประยุกต์กับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

เป็นขั้นตอนที่โปรแกรมประยุกต์ทำการเชื่อมต่อเข้ากับหน่วยเก็บข้อมูลโครงการ CUBEMS เพื่อใช้ในการร้องขอข้อมูลที่จำเป็นสำหรับกระบวนการวิเคราะห์ รวมถึงขั้นตอนการเขียนผลลัพธ์ที่ได้จากกระบวนการข้างต้นกลับเข้าไปที่หน่วยเก็บข้อมูล โดยโปรแกมประยุกต์ต้นแบบที่ถูกพัฒนาขึ้นติดตั้งอยู่บนคอมพิวเตอร์ ซึ่งมีคุณลักษณะดังแสดงในตารางที่ 3.1 และการทดสอบจะดำเนินงานบนโปรแกรมจูไพเตอร์โน็ตบุ้ค (Jupyter Notebook) [26]

ตารางที่ 3.1: คุณลักษณะของคอมพิวเตอร์ที่ใช้ในการทดสอบโปรแกมประยุกต์

Model	**description**
Computer	Windows 7 Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz processor 12.0 GB

3.2.1 การทดสอบร้องขอข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 โดยอาศัยโพรโทคอล FETCH

การร้องขอข้อมูลจากหน่วยเก็บข้อมูลโครงการ CUBEMS ต้องอาศัยโพรโทคอล FETCH ที่ได้กล่าวไว้ในหัวข้อ 2.1.2 ซึ่งเป็นข้อบังคับสำหรับการสื่อสารระหว่างองค์ประกอบในมาตรฐาน IEEE1888 ที่กำหนดให้การส่งข้อมูลต้องอยู่ในรูปแบบของโพรโทคอล SOAP โดยวิทยานิพนธ์ฉบับนี้อาศัยคลัง (โปรแกรม) suds [27] ที่สร้างขึ้นบนพื้นฐานภาษา python สำหรับใช้ประกาศขั้นตอนที่จำเป็นสำหรับเรียกใช้เซอร์วิส SOAP โดย suds จะส่งคำสั่งไปที่หน่วยเก็บข้อมูลโครงการเพื่อร้องขอผ่านบริการเว็บ หรือ WSDL (web services description language) ซึ่งนิยามในรูปแบบ XML ที่อธิบายรายละเอียดของเว็บเซอร์วิสสำหรับให้โปรแกรมที่ต้องการเรียกใช้ทราบถึงรูปแบบของบริการ รวมถึงวิธีการเรียกใช้เพื่อสร้างข้อความร้องขอ (request message) สำหรับใช้ร้องขอบริการ รูปที่ 3.2 แสดงผลลัพธ์ที่ได้จากการร้องขอ WSDL จากหน่วยเก็บข้อมูลโครงการ CUBEMS สำหรับใช้ในการสื่อสารระหว่างองค์ประกอบในมาตรฐาน IEEE1888 ขั้นตอนต่อไปคือการกำหนดตัวแปรที่จำเป็นสำหรับใช้ในการร้องขอข้อมูลให้กับ WSDL ตามโครงสร้างข้อมูล (data structure) มาตรฐาน IEEE1888 โดยมีรายละเอียดตามตารางที่ 3.2 และ 3.3

รูปที่ 3.3 แสดงตัวอย่าการร้องขอข้อมูลค่าอุณหภูมิตามโครงสร้างข้อมูลมาตรฐาน IEEE1888 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร เขตพื้นที่ย่อย 1 ชั้น 13 อาคารเจริญวิศวกรรม วันที่ 16 กุมภาพันธ์ ตั้งแต่เวลา 13:00 ถึง 13:30 จากหน่วยเก็บข้อมูลโครงการ CUBEMS โดยการตอบกลับจากหน่วยเก็บข้อมูลจะอยู่ในรูปแบบข้อมูลตามเวลาดังแสดงในรูปที่ 3.4 เนื่องจากข้อจำกัดในการ

ส่งข้อมูลตามมาตรฐาน IEEE1888 สามารถรองรับได้เพียง 5000 ค่าต่อรอบการส่ง ในกรณีที่มีการร้องขอข้อมูลเกิน 5000 ค่าระบบจะทำการแบ่งผลลัพธ์ในการตอบกลับออกเป็นหลายชุด โดยในแต่ละรอบการตอบกลับจะมีการระบุตัวชี้ตำแหน่ง (cursor) ดังแสดงในรูปที่ 3.4 เพื่อใช้สำหรับแสดงตัวตนในการตอบกลับผลลัพธ์ของการร้องขอในรอบถัดไป และในรูปที่ 3.5 แสดงค่าเฉลี่ยระยะเวลาการดึงข้อมูลอุณหภูมิ 100 รอบจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

ตารางที่ 3.2: โครงสร้างข้อมูลตามมาตรฐาน IEEE1888 [15]

Class	**Object**	**Attribute**
Transport	**Header, Body**	-
Header	**Query, OK, Error**	-
Body	**PointSet, Point**	-
Key	-	id: the target id of Point or PointSet
		attrName: the attribute name for the following
		eq: this predicate becomes true if the key attribute value is equal to the specified value, otherwise becomes false
		neq: this predicate becomes true if the key attribute value is not equal to the specified value, otherwise becomes false
		lt: this predicate becomes true if the key attribute value is less than the specified value
		gt: this predicate becomes true if the key attribute value is greater than the specified value
		lteq: this predicate becomes true if the key attribute value is less than or equal to the specified value
		gteq: this predicate becomes true if the key attribute value is greater than or equal to the specified value
		select: for selection of {maximum, minimum} attribute value
		trap: for event detection {changed} (valid if the query type="stream")

ตารางที่ 3.3: โครงสร้างข้อมูลตามมาตรฐาน IEEE1888 (ต่อ) [15]

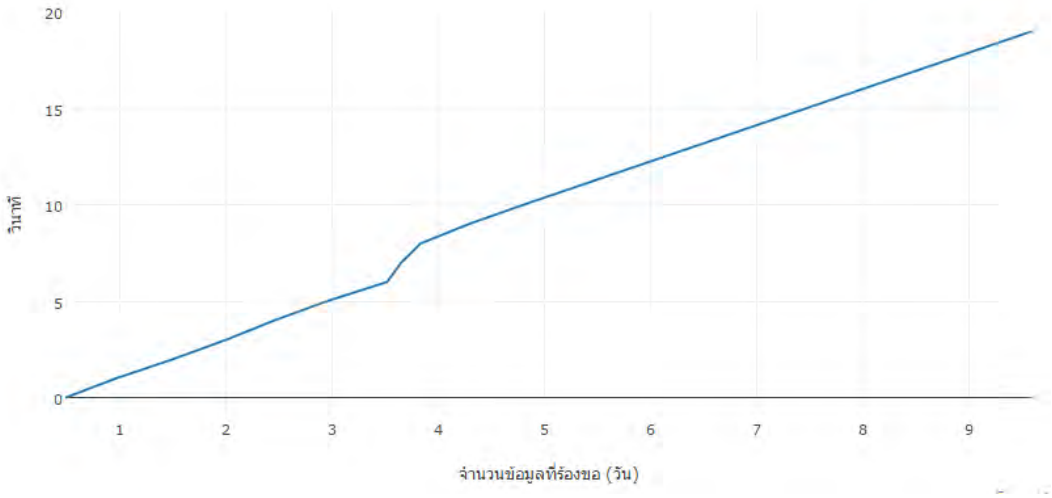
Class	Object	Attribute
Query	**Key**	id: the identifier of this query (UUID)
		type: the type of this query: (storage or stream)
		cursor: the cursor for sequential dataset retrieval (valid when type="storage")
		acceptableSize: Receiver's maximum acceptable size of value objects at one RPC
		ttl: validity time of query at the Provider (valid when type ="stream")
		callbackData: the URI of data callback in TRAP protocol (valid when type="stream")
		callbackControl: the URI of control-signal callback in TRAP protocol (valid when type="stream")
Ok	**-**	-
Error	**-**	type: category of error
PointSet	**PointSet Point**	id: URI-based identifier for this PointSet.
Point	**Value**	id: identifier of the Point (i.e., PointID)
Value	**-**	time: generated time (INPUT case) or scheduled time (OUTPUT case) in W3C Timestamp format

```
(transport){
  header =
    (header){
      OK = None
      error =
        (error){
          value = None
          _type = ""
        }
      query =
        (query){
          key[] = <empty>
          _id = ""
          _type = ""
          _cursor = ""
          _ttl = ""
          _acceptableSize = ""
          _callbackData = ""
          _callbackControl = ""
        }
    }
  body =
    (body){
      pointSet[] = <empty>
      point[] = <empty>
    }
}
```

รูปที่ 3.2: WSDL ที่ได้ตอบกลับจากหน่วยเก็บข้อมูลโครงการ CUBEMS

รูปที่ 3.3: การร้องขอข้อมูลจากโปรแกรมประยุกต์ไปยังหน่วยเก็บข้อมูลโดยโพรโทคอล FETCH

รูปที่ 3.4: การตอบกลับพร้อมข้อมูลจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล FETCH



รูปที่ 3.5: ค่าเฉลี่ยระยะเวลาการร้องขอข้อมูลอุณหภูมิจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 จำนวน 1 PointID ที่แปรผันตามจำนวนวันที่ร้องขอ

3.2.2 การทดสอบส่งข้อมูลไปที่หน่วยเก็บข้อมูลมาตรฐาน IEEE1888 โดยอาศัยโพรโทคอล WRITE

เช่นเดียวกับการร้องขอข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 ที่ได้กล่าวไว้ในหัวข้อที่ 3.2.1 การส่งข้อมูลไปที่หน่วยเก็บข้อมูลโครงการ CUBEMS ต้องอาศัยโพรโทคอล WRITE ที่ได้กล่าวไว้ในหัวข้อที่ 2.1.2 ซึ่งมีกระบวนการคล้ายกับโพรโทคอล FETCH คือมีการส่งคำสั่งไปที่หน่วยเก็บข้อมูลโครงการเพื่อการร้องขอ WSDL แต่มีจุดแตกต่างอยู่ที่ขั้นตอนการกำหนดตัวแปร โดยมีรายละเอียดตามตารางที่ 3.2 และ 3.3 และในรูปที่ 3.6 แสดงการส่งมูลข้อมูลชนิดตัวเลขตามโครงสร้างข้อมูลมาตรฐาน IEEE1888 เข้าไปที่ PointID 'http://pong.test.kahnn/ps/point_int_100' ในวันที่ 21 กุมภาพันธ์ เวลา 12:39:03 ซึ่งหน่วยเก็บข้อมูลของโครงการจะตอบกลับผลลัพธ์ของการส่งข้อมูลดังแสดงในรูปที่ 3.7

```
(transport){
  header =
    (header){
      OK = None
      error =
        (error){
          value = None
          _type = ""
        }
      query =
        (query){
          key[] = <empty>
          _id = ""
          _type = ""
          _cursor = ""
          _ttl = ""
          _acceptableSize = ""
          _callbackData = ""
          _callbackControl = ""
        }
    }
  body =
    (body){
      pointSet[] = <empty>
      point[] =
        (point){
          value[] =
            (value){
              value = 10
              _time = 2017-04-21 12:39:03.013000
            },
          _id = "http://pong.test.kahnn/ps/point_int_100"
        },
    }
}
```

รูปที่ 3.6: การส่งข้อมูลไปยังหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

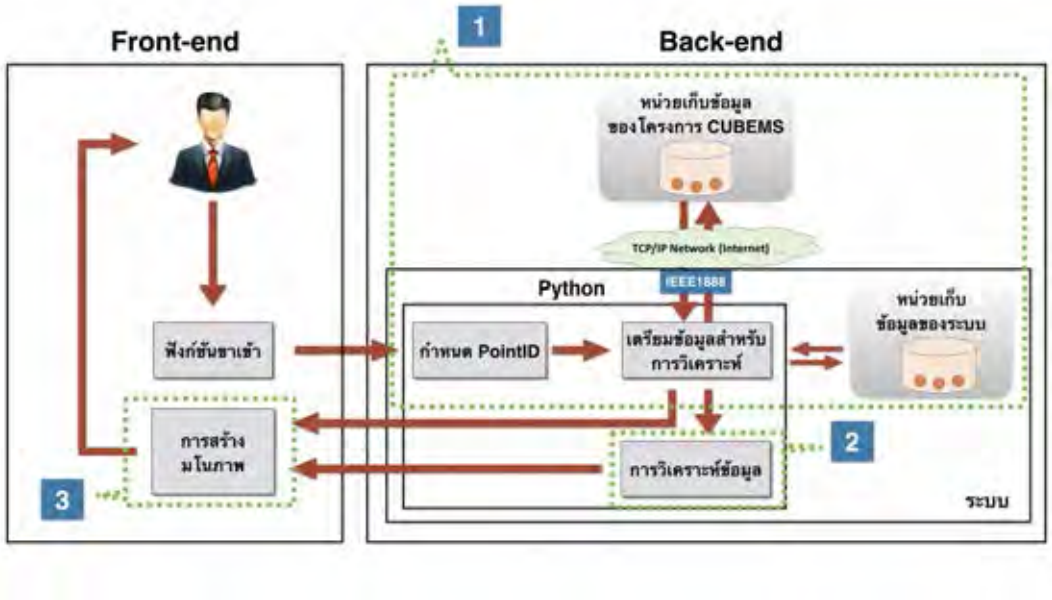
```
(transport){
  header =
    (header){
      OK = ""
    }
}
```

รูปที่ 3.7: การตอบกลับกระบวนการส่งข้อมูลจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

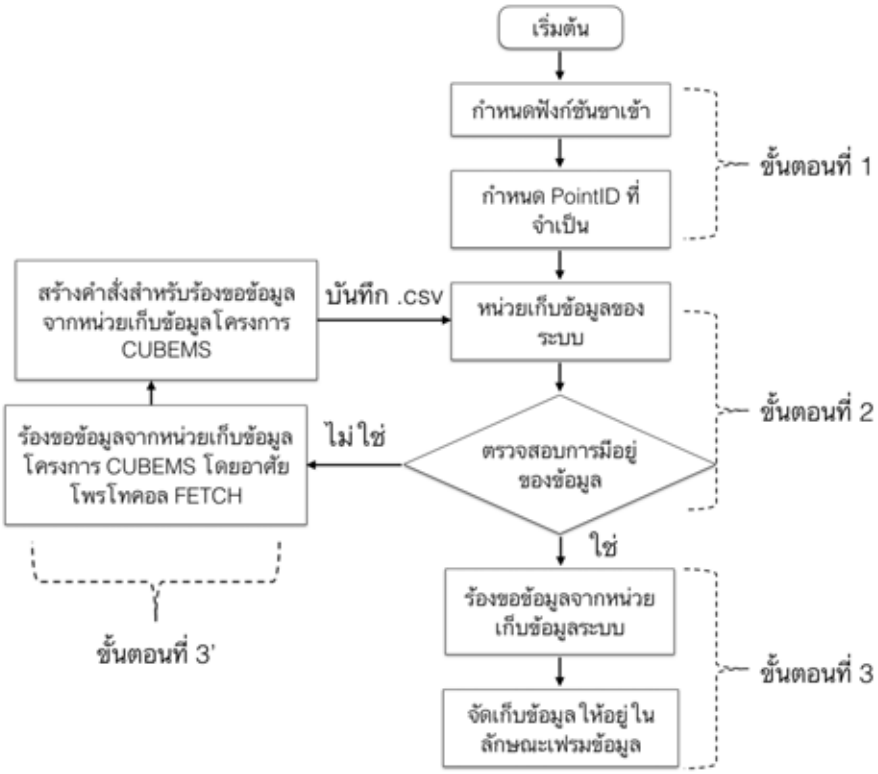
3.3 การออกแบบและพัฒนาโปรแกรมประยุกต์ต้นแบบ

3.3.1 การพัฒนาส่วนต่อประสานของโปรแกรมประยุกต์

เพื่อตอบสนองในเรื่องความยืดหยุ่นของโปรแกรมประยุกต์ในกรณีที่มีการติดตั้งอุปกรณ์เพิ่มเข้ามาในระบบการจัดการพลังงานที่มีอยู่เดิมเดิม การออกแบบโปรแกรมประยุกต์ให้สามารถกำหนด PointID ให้ตรงกับความต้องการของฟังก์ชันรูปแบบต่าง ๆ ได้อย่างอัตโนมัติจึงเป็นสิ่งสำคัญ และเมื่อพิจารณาร่วมกับระยะเวลาที่ใช้ในการร้องขอข้อมูลจากหน่วยเก็บข้อมูลของโครงการ CUBEMS ตามมาตรฐาน IEEE1888 ดังแสดงในรูปที่ 3.5 พบว่าต้องใช้เวลาถึง 20 วินาที สำหรับร้องขอข้อมูลการบริโภคพลังงานจำนวน 10 วัน หรือ 14,400 ข้อมูล ซึ่งในความเป็นจริงการวิเคราะห์ข้อมูลหนึ่งฟังก์ชันต้องอาศัยข้อมูลจาก PointID หลายตัว ยกตัวอย่างเช่นในการแสดงภาพรวมการบริโภคพลังงานตลอดวันของห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารต้องอาศัยข้อมูลจาก PointID จำนวน 90 ตัว หรือประมาณ 1,296,000 ข้อมูล ซึ่งต้องใช้เวลาในการร้องขอข้อมูลมากถึง 30 นาที ทำให้ยากต่อการนำไปประยุกต์ใช้งานจริง ผู้วิจัยจึงได้ปรับปรุงสถาปัตยกรรมของระบบข้างต้นในส่วนต่อประสานให้มีการทำสำเนาข้อมูล (copy) ไปที่หน่วยเก็บข้อมูลสำรองซึ่งอยู่ในคอมพิวเตอร์ที่ใช้ดำเนินงานในรูปแบบไฟล์ CSV (comma separated value) เป็นไฟล์ข้อความใช้สำหรับเก็บข้อมูลในรูปแบบตารางโดยอาศัยเครื่องหมายจุลภาคในการแบ่งแนวตั้ง (column) แต่ละแถว โดยมีการกำหนดเงื่อนไขเพื่อตรวจสอบการมีอยู่ของข้อมูลในหน่วยเก็บข้อมูลสำรองดังแสดงรูปที่ 3.8 ทั้งหมดนี้มีเป้าหมายเพื่อลดความซ้ำซ้อนในการร้องขอข้อมูลที่เวลาเดิมจากหน่วยเก็บข้อมูลมาตรฐาน IEEE1888 โดยมีผังงานแสดงในรูปที่ 3.9



รูปที่ 3.8: สถาปัตยกรรมของระบบหลังถูกปรับปรุง



รูปที่ 3.9: ผังงานการทำงานของส่วนต่อประสานสำหรับตีความคำสั่งเพื่อกำหนดแหล่งที่มาของข้อมูล

- ขั้นตอนที่ 1 : การตีความคำสั่งเพื่อกำหนดชุด PointID ที่จำเป็น เมื่อส่วนต่อประสานของโปรแกรมประยุกต์ได้รับคำสั่งขาเข้าในรูปแบบการเรียกใช้ฟังก์ชัน โปรแกรมประยุกต์จะทำการตีความคำสั่งเพื่อกำหนดชุด PointID และค่าลำดับที่ที่จำเป็นต้องใช้ในการดำเนินงานฟังก์ชันข้างต้น โดยอาศัยหลักการกรองข้อมูลจากชุด PointID ทั้งหมดของโครงการที่อยู่ในลักษณะรูปแบบแถวลำดับ (array) ดังแสดงในรูปที่ 3.10 และทำการจัดแบ่ง (classified) ให้อยู่ในรูปแบบเฟรมข้อมูลที่มีการแบ่งรายละเอียดในแนวตั้งออกเป็น ลำดับที่ ยูอาร์แอล อาคาร ชั้น ทิศ ห้อง เขตพื้นที่ หน้าที่ และชนิดข้อมูล ดังแสดงในรูปที่ 3.11 ทำให้โปรแกรมประยุกต์สามารถ ค้นหาลักษณะเฉพาะของ PointID ที่ต้องการได้ และในรูปที่ 3.12 - 3.13 แสดงผลลัพธ์ที่ได้จากการตีความคำสั่ง โดยจะอยู่ในรูปแบบรายการ (list) ที่บรรจุแถวลำดับของ PointID และค่าลำดับที่

```
PointIDD = ["http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/aircon_3ph1/monitor/energy_r",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/aircon_3ph1/monitor/energy_s",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/aircon_3ph1/monitor/energy_t",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/light1/monitor/energy",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/outlet1/monitor/energy",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/humidity",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/illuminance",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/pir",
"http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/temperature",
```

รูปที่ 3.10: PointID ของระบบ (บางส่วน) ในรูปแบบแถวลำดับ

	http	place	fl	direction	room	zone	kind	duty	value
0	http://bems.ee.eng.chula.ac.th	eng4	fl13	north	lab_tsrl_dsprl_emrl	z1	aircon_3ph1	monitor	energy_r
1	http://bems.ee.eng.chula.ac.th	eng4	fl13	north	lab_tsrl_dsprl_emrl	z1	aircon_3ph1	monitor	energy_s
2	http://bems.ee.eng.chula.ac.th	eng4	fl13	north	lab_tsrl_dsprl_emrl	z1	aircon_3ph1	monitor	energy_t
3	http://bems.ee.eng.chula.ac.th	eng4	fl13	north	lab_tsrl_dsprl_emrl	z1	light1	monitor	energy
4	http://bems.ee.eng.chula.ac.th	eng4	fl13	north	lab_tsrl_dsprl_emrl	z1	outlet1	monitor	energy

รูปที่ 3.11: PointID ของระบบ (บางส่วน) ที่ถูกจัดแบ่งในรูปแบบเฟรมข้อมูล

```
In [200]: Khetnon.Find_PointID(room='lab_tsrl_dsprl_emrl',zone='z4',kind='Air')

Out[200]: {'IND': [49, 50, 51],
           'Point_Name': ['http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z4/aircon_3ph1/monitor/energy_r',
            'http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z4/aircon_3ph1/monitor/energy_s',
            'http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z4/aircon_3ph1/monitor/energy_t']}
```

รูปที่ 3.12: ผลลัพธ์การตีความคำสั่งเพื่อกำหนด PointID ของระบบปรับอากาศบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร เขตพื้นที่ย่อย 4

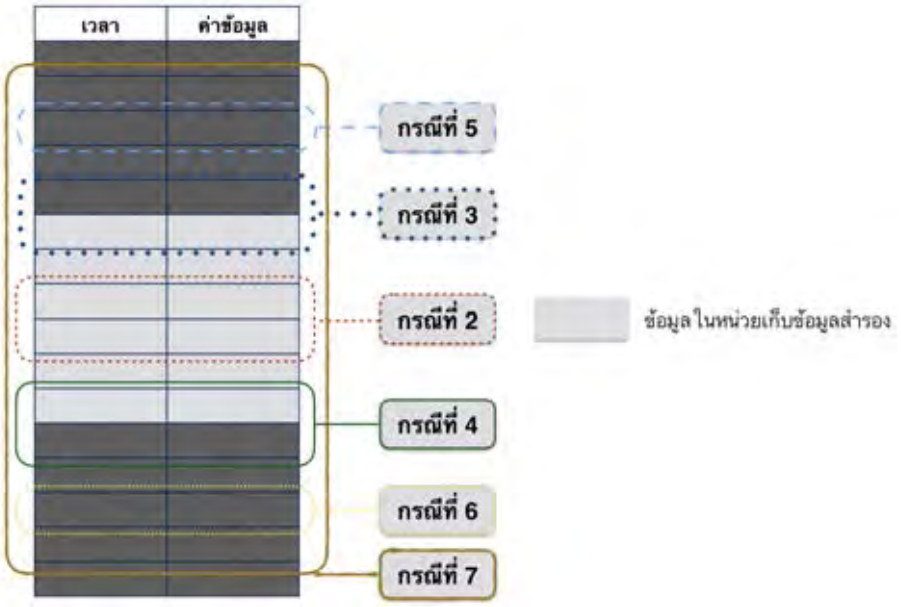
```
In [206]: Khetnon.Find_PointID(room='lab_tsrl_dsprl_emrl',zone='z4',value='pir')

Out[206]: {'IND': [56, 60],
           'Point_Name': ['http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z4/sensor1/monitor/pir',
            'http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z4/sensor2/monitor/pir']}
```

รูปที่ 3.13: ผลลัพธ์การตีความคำสั่งเพื่อกำหนด PointID ของตัวรับรู้การเคลื่อนไหวบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร เขตพื้นที่ย่อย 4

- ขั้นตอนที่ 2 : การเข้าถึงหน่วยเก็บข้อมูลสำรองเพื่อตรวจสอบการมีอยู่ของข้อมูล โปรแกรมประยุกต์จะใช้ค่าลำดับที่ ซึ่งเป็นผลลัพธ์จากขั้นตอนที่ 1 เพื่อใช้ตรวจสอบการมีอยู่ของไฟล์ PointID ในหน่วยเก็บข้อมูลสำรอง ในกรณีที่ไม่พบไฟล์ของ PointID ข้างต้น โปรแกรมประยุกต์จะกำหนดแหล่งที่มาของข้อมูลเป็นหน่วยเก็บข้อมูลโครงการ CUBEMS และเข้าสู่ขั้นตอนที่ 3’ แต่ถ้าโปรแกรมประยุกต์ตรวจพบไฟล์ของ PointID จะทำการอ่านไฟล์ข้อมูลเข้ามาเก็บในรูปแบบเฟรมข้อมูล และใช้ค่าเวลาเริ่มต้นกับค่าเวลาสุดท้ายสำหรับตรวจสอบการมีอยู่ของข้อมูลในช่วงเวลาที่ร้องขอ รวมถึงระบุเหตุการณ์ที่จะเกิดขึ้นเพื่อกำหนดขั้นตอนต่อไป ตารางที่ 3.4 แจกแจงเหตุการณ์ทั้งหมดที่มีโอกาสเกิดขึ้นในขั้นตอนการเตรียมข้อมูล ในกรณีที่โปรแกรมประยุกต์ตีความให้เหตุการณ์อยู่ในกรณีที่ 2 จะแสดงถึงการมีอยู่ของข้อมูลในหน่วยเก็บข้อมูลสำรองที่ครบถ้วน โปรแกรมประยุกต์จะกำหนดให้การทำงานเข้าสู่ขั้นตอนที่ 3 เพื่อทำการค้นคืน (retrieve) ข้อมูลจากหน่วยเก็บข้อมูลสำรอง แต่ถ้าโปรแกรมประยุกต์ตีความเหตุการณ์อยู่ในกรณีที่ 3 เป็นต้นไป จนถึงกรณีที่ 7 จะแสดงถึงการขาดหายของข้อมูลที่หน่วยเก็บข้อมูลสำรอง โดยโปรแกรมประยุกต์จะกำหนดแหล่งที่มาของข้อมูลเป็นหน่วยเก็บข้อมูลโครงการ CUBEMS สำหรับทำการร้องขอข้อมูลตามมาตรฐาน IEEE1888 ในขั้นตอนที่ 3’

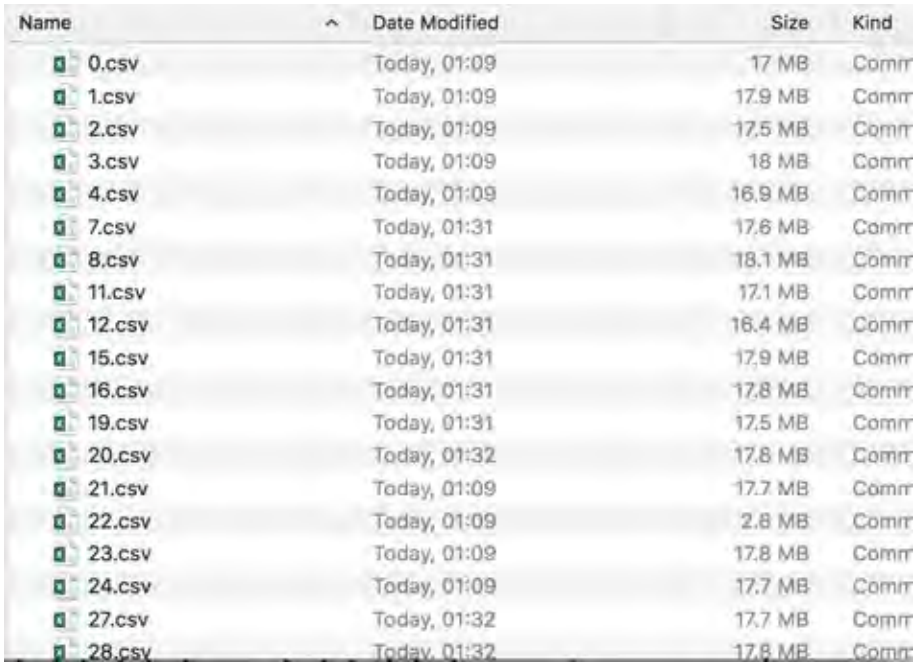
ตารางที่ 3.4: เหตุการณ์ที่เกิดขึ้นในขั้นตอนการเตรียมข้อมูล

เหตุการณ์	รายละเอียด
กรณีที่ 1	ไม่พบไฟล์ข้อมูล
กรณีที่ 2	ตรวจพอข้อมูลในช่วงเวลาที่ร้องขอทั้งหมด
กรณีที่ 3	ตรวจพบข้อมูลเพียงบางส่วนในช่วงเวลาท้ายของการร้องขอ
กรณีที่ 4	ตรวจพบข้อมูลเพียงบางส่วนในช่วงเวลาเริ่มต้นของการร้องขอ
กรณีที่ 5	ไม่พบข้อมูลเนื่องจากช่วงเวลาที่ร้องขออยู่ก่อนเวลาเริ่มต้นของข้อมูลในหน่วยเก็บข้อมูลสำรอง
กรณีที่ 6	ไม่พบข้อมูลเนื่องจากช่วงเวลาที่ร้องขออยู่หลังเวลาสุดท้ายของข้อมูลในหน่วยเก็บข้อมูลสำรอง
กรณีที่ 7	ตรวจพบข้อมูลเพียงบางส่วนระหว่างช่วงเวลาที่ร้องขอ



รูปที่ 3.14: เหตุการณ์ที่เกิดขึ้นในขั้นตอนการเตรียมข้อมูล

- ขั้นตอนที่ 3’ : การร้องขอข้อมูลจากหน่วยเก็บข้อมูลโครงการ CUBEMS ในกรณีที่ไม่พบข้อมูลตามช่วงเวลาที่ร้องขอในหน่วยเก็บข้อมูลสำรอง โปรแกรมประยุกต์จะทำการเขียนคำสั่งสำหรับร้องขอข้อมูลไปที่หน่วยเก็บข้อมูลโครงการ CUBEMS ตามมาตรฐาน IEEE1888 ดังที่กล่าวไว้ในหัวข้อที่ 3.2.1 โดยข้อมูลที่ถูกตอบกลับจากหน่วยเก็บข้อมูลจะถูกกรองให้ออกมาเฉพาะข้อมูลที่เวลาใด ๆ รวมถึงจัดรูปแบบให้อยู่ในลักษณะเฟรมข้อมูล ขั้นตอนถัดไปคือการนำผลลัพธ์ข้างต้นผสานเข้ากับข้อมูลที่มีอยู่ในหน่วยเก็บข้อมูลสำรองเพื่อปรับให้เป็นปัจจุบัน (update) หลังจากนั้นจะทำการบันทึกข้อมูลที่เกิดจากการผสานเข้าไปในหน่วยเก็บข้อมูลสำรองในรูปแบบไฟล์ csv ที่ประยุกต์ใช้ค่าลำดับที่จากขั้นตอนที่ 1 ในการระบุตัวตนของไฟล์ PointID แต่ละตัวดังแสดงในรูปที่ 3.15



Name	Date Modified	Size	Kind
0.csv	Today, 01:09	17 MB	Comm
1.csv	Today, 01:09	17.9 MB	Comm
2.csv	Today, 01:09	17.5 MB	Comm
3.csv	Today, 01:09	18 MB	Comm
4.csv	Today, 01:09	16.9 MB	Comm
7.csv	Today, 01:31	17.6 MB	Comm
8.csv	Today, 01:31	18.1 MB	Comm
11.csv	Today, 01:31	17.1 MB	Comm
12.csv	Today, 01:31	16.4 MB	Comm
15.csv	Today, 01:31	17.9 MB	Comm
16.csv	Today, 01:31	17.8 MB	Comm
19.csv	Today, 01:31	17.5 MB	Comm
20.csv	Today, 01:32	17.8 MB	Comm
21.csv	Today, 01:09	17.7 MB	Comm
22.csv	Today, 01:09	2.8 MB	Comm
23.csv	Today, 01:09	17.8 MB	Comm
24.csv	Today, 01:09	17.7 MB	Comm
27.csv	Today, 01:32	17.7 MB	Comm
28.csv	Today, 01:32	17.8 MB	Comm

รูปที่ 3.15: ไฟล์ข้อมูล PointID (บางส่วน) ในลักษณะไฟล์ csv

- ขั้นตอนที่ 3 : การค้นคืนข้อมูลจากหน่วยเก็บข้อมูลสำรอง โปรแกรมประยุกต์จะเรียกไฟล์ของ PointID ที่ถูกเก็บอยู่ในหน่วยเก็บข้อมูลสำรอง โดยอาศัยค่าลำดับที่ที่ได้จากขั้นตอนที่ 1 ในการระบุตัวตนของไฟล์แต่ละไฟล์ และคัดกรองเฉพาะข้อมูลตามช่วงเวลาที่ร้องขอ

ตามข้อกำหนดของมาตรฐาน IEEE1888 ที่กำหนดให้ข้อมูลจะถูกจัดเก็บในรูปแบบวันที่และเวลา ฉะนั้นการออกแบบให้โปรแกรมประยุกต์สามารถเข้าถึงประวัติข้อมูลในช่วงเวลาต่าง ๆ ได้อย่างง่ายได้จึงเป็นสิ่งจำเป็น และเมื่อพิจารณาเรื่องความหลากหลายของชนิดข้อมูลที่เป็นข้อมูลขาเข้าวิทยานิพนธ์ฉบับนี้เลือกใช้การจัดรูปแบบข้อมูลทั้งหมดให้อยู่ในรูปแบบเฟรมข้อมูล หรือเป็นการเก็บข้อมูลให้อยู่ในรูปแบบลักษณะคล้ายตาราง คือมีแถวนอน (row) และแถวตั้ง (column) ทำให้สะดวกต่อการนำไปประยุกต์ใช้กับการวิเคราะห์ข้อมูลที่อยู่ในรูปแบบข้อมูล ณ วันที่และเวลาใด ๆ (datetime) ยกตัวอย่างเช่นในรูปที่ 3.16 แสดงการจัดรูปแบบข้อมูลผลรวมการบริโภคพลังงานทุก 15 นาทีในลักษณะเฟรมข้อมูล โดยกำหนดให้แถวนอนแสดงค่าวันที่และเวลาใด ๆ ส่วนในแถวตั้งแยกออกเป็นระบบย่อยที่มีการบริโภคพลังงาน ซึ่งประกอบไปด้วยการบริโภคพลังงานใน

ระบบแสงสว่าง การบริโภคพลังงานในระบบเต้าเสียบ และการบริโภคพลังงานในระบบปรับอากาศ เพื่อให้สะดวกต่อการนำไปใช้เปรียบสัดส่วนการบริโภคพลังงานของแต่ละระบบ รูปที่ 3.17 แสดงการจัดรูปแบบข้อมูลผลรวมการบริโภคพลังงานในห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารบริเวณชั้น 13 อาคารเจริญวิศวกรรมในลักษณะเฟรมข้อมูลที่กำหนดให้แถวตั้งเป็นค่าเวลา ส่วนในแถวนอนแยกออกเป็นวันแต่ละวันในสัปดาห์ การจัดลักษณะรูปแบบนี้ทำให้ง่ายต่อการเปรียบเทียบค่าการบริโภคพลังงาน ณ ช่วงเวลาเดิมของวันในแต่ละสัปดาห์ ในรูปที่ 3.18 แสดงการจัดรูปแบบข้อมูลผลรวมการบริโภคพลังงานทุก 15 นาทีของบริเวณพื้นที่ชั้น 13 อาคารเจริญวิศวกรรมในลักษณะเฟรมข้อมูลที่กำหนดให้แถวตั้งเป็นค่าวันที่และเวลาใด ๆ ส่วนในแถวนอนแยกออกเป็นห้องบริเวณชั้น 13 และในรูปที่ 3.19 แสดงการจัดรูปแบบข้อมูลในลักษณะเฟรมข้อมูลที่กำหนดให้แถวตั้งเป็นค่าวันที่และเวลาใด ๆ ส่วนในแถวนอนแยกออกเป็นชนิดข้อมูลประเภทต่าง ๆ สะดวกต่อการวิเคราะห์ข้อมูลในรูปแบบการหาความสัมพันธ์ของข้อมูลประเภทต่าง ๆ

	AIR (Wh)	LIGHT (Wh)	OUTLET (Wh)
TIME			
2017-01-01 00:00:00	-1.376	0.0	21.478
2017-01-01 00:15:00	-1.820	0.0	20.051
2017-01-01 00:30:00	-0.685	0.0	21.514
2017-01-01 00:45:00	-0.488	0.0	21.522
2017-01-01 01:00:00	-0.519	0.0	21.503

รูปที่ 3.16: ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของทุกระบบบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบเฟรมข้อมูล

	2016-09-20	2016-09-21	2016-09-22	2016-09-23	2016-09-24	2016-09-25	2016-09-26	2016-09-27	2016-09-28	2016-09-29	...
00:00:00	8.794	376.896	237.158	354.811	318.685	339.985	295.618	283.556	255.147	308.747	...
00:15:00	323.736	274.324	287.582	347.558	302.070	265.427	321.924	280.780	306.049	405.735	...
00:30:00	356.296	199.210	337.942	366.301	478.521	325.413	308.244	252.982	282.626	310.480	...
00:45:00	297.218	272.348	364.302	268.153	385.020	310.442	236.753	400.056	300.874	381.934	...
01:00:00	272.796	311.719	335.553	219.351	284.937	313.703	222.874	351.119	350.619	245.483	...
01:15:00	321.528	340.509	351.169	309.566	258.621	369.964	279.134	345.276	368.012	299.149	...
01:30:00	326.101	417.608	309.702	321.420	474.477	273.418	340.293	332.799	319.015	352.240	...
01:45:00	340.409	222.020	282.471	360.167	320.441	254.650	299.801	316.919	304.192	331.276	...
02:00:00	269.758	205.483	306.190	340.043	312.951	285.888	324.057	315.282	272.001	331.317	...
02:15:00	316.096	328.120	335.383	239.429	505.272	249.878	264.240	372.728	284.678	316.569	...

รูปที่ 3.17: ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของช่วงเวลาแต่ละช่วงเวลาบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบเฟรมข้อมูล

	room_csc_jpl_ccr	lab_tsrl_dspri_emri	room_server	room_pkp_pjp_was	room_sav_dwc_nts	lectureroom_dsp	lectureroom_2
TIME							
2017-03-20 15:15:00	37.431	4766.871	435.306	6.472	35.847	56.940	1.689
2017-03-20 15:30:00	37.546	4839.913	390.656	6.515	36.816	56.867	1.702
2017-03-20 15:45:00	37.624	4844.297	410.893	6.612	91.649	57.581	1.709
2017-03-20 16:00:00	37.766	3943.816	468.963	6.658	591.838	56.862	1.727
2017-03-20 16:15:00	37.576	4344.808	367.209	6.598	662.159	56.932	1.743
2017-03-20 16:30:00	26.325	4417.574	414.662	7.867	655.388	56.870	1.746

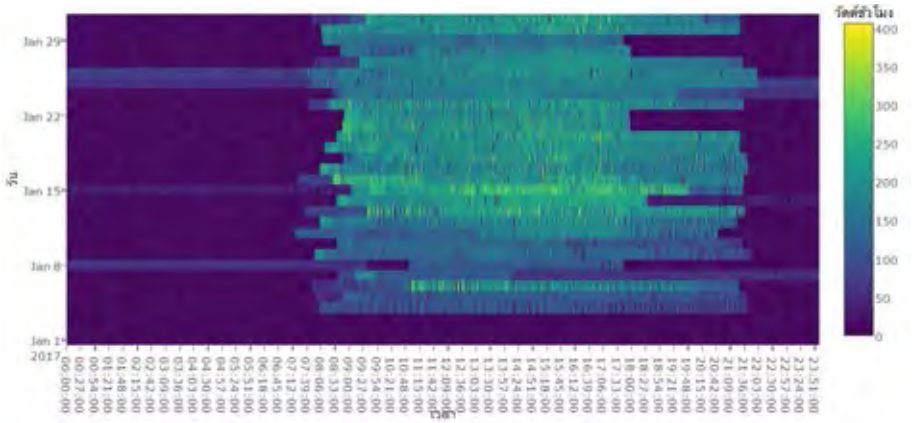
รูปที่ 3.18: ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรมในรูปแบบเฟรมข้อมูล

	Air (Wh)	Temp (celsius)	PIR
TIME			
2017-04-06 15:35:00	60.878	23.970	ON
2017-04-06 15:36:00	61.747	23.970	ON
2017-04-06 15:37:00	62.009	23.970	ON
2017-04-06 15:38:00	62.014	23.910	ON
2017-04-06 15:39:00	61.045	24.025	ON
2017-04-06 15:40:00	61.853	24.025	ON
2017-04-06 15:41:00	62.317	23.965	ON

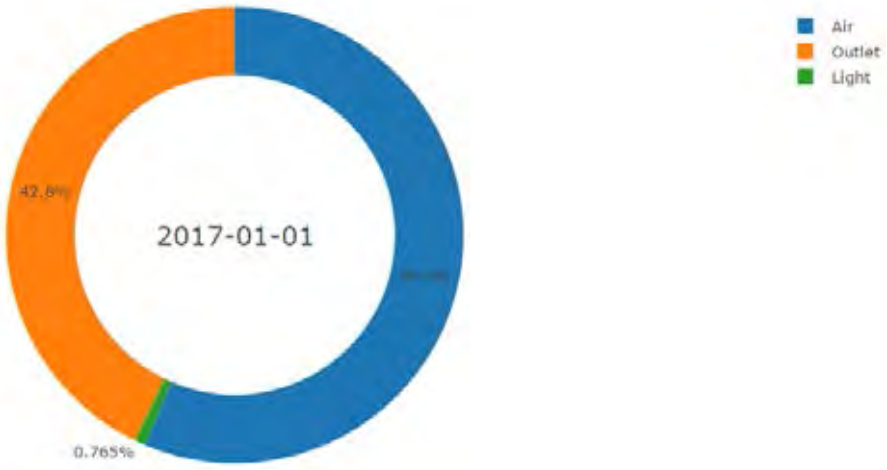
รูปที่ 3.19: ข้อมูลการตรวจจับการเคลื่อนไหว ค่าอุณหภูมิ และค่าการบริโภคพลังงานของระบบปรับอากาศในหน่วยวัตต์ชั่วโมงในรูปแบบเฟรมข้อมูล

3.3.2 การออกแบบฟังชันก์การสร้างมโนภาพข้อมูลของโปรแกรมประยุกต์

วิทยานิพนธ์ฉบับนี้ออกแบบฟังก์ชันพื้นฐานของโปรแกรมประยุกต์ให้สามารถแสดงประวิติข้อมูลครอบคลุมระดับภาพรวมทั้งโครงการจนถึงระดับเขตพื้นที่ย่อย และเพื่อตอบสนองความต้องการในกรณีที่ต้องการข้อมูลเฉพาะ (spcific data) รูปแบบต่าง ๆ ยกตัวอย่างเช่น รูปที่ 3.20 แสดงข้อมูลการบริโภคพลังงานบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบแผนภูมิความร้อนตลอดเดือนมกราคม ซึ่งแสดงให้เห็นถึงแนวโน้มการบริโภคพลังงานในช่วงเวลาต่าง ๆ ของวัน และรูปที่ 3.21 แสดงสัดส่วนการบริโภคพลังงานบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในวันที่ 1 มกราคม พ.ศ. 2560 ในลักษณะการสรุปเป็นช่วงเวลาโดยแยกออกเป็น ระบบแสงสว่าง ระบบปรับอากาศ และระบบเต้าเสียบ

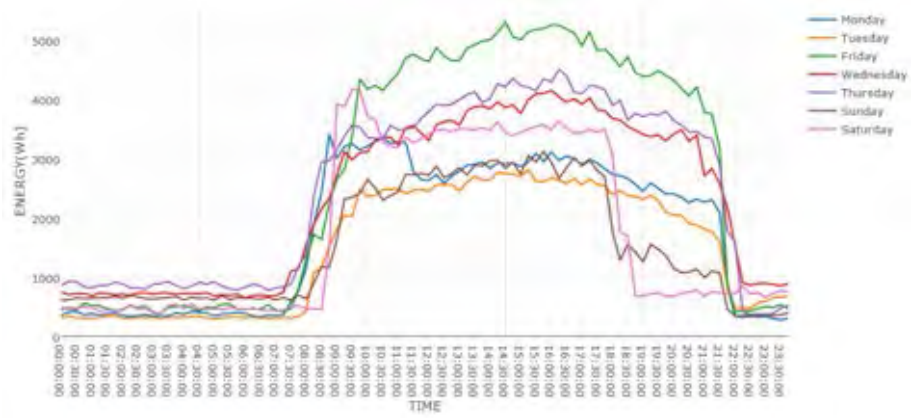


รูปที่ 3.20: ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารในรูปแบบแผนภูมิความร้อนตลอดเดือนมกราคม



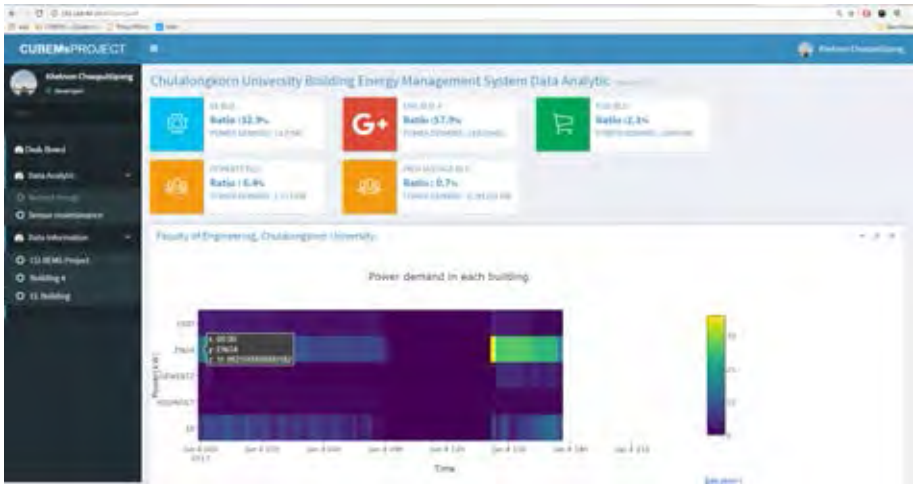
รูปที่ 3.21: ข้อมูลการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงของทุกระบบในรูปแบบกราฟวงกลมบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารแยกระบบ

ในความเป็นจริงสิ่งปลูกสร้างประเภทอาคารล้วนมีตารางการใช้งานเป็นไปในแนวทางที่คล้ายเดิมในทุกสัปดาห์ วิทยานิพนธ์ฉบับนี้ได้ออกแบบฟังก์ชันการสร้างมโนภาพเพื่อแสดงประวัติการบริโภคพลังงานของวันแต่ละวันในสัปดาห์ ยกตัวอย่างเช่น ในรูปที่ 3.22 แสดงค่าเฉลี่ยการบริโภคพลังงานย้อนหลังตลอดเดือนมกราคมบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร จะเห็นได้ชัดเจนว่าจะเริ่มมีการบริโภคพลังงานตั้งแต่เวลา 7 นาฬิกาของทุกวัน และลงลงในเวลา 22 นาฬิกา และ 18 นาฬิกาในวันจันทร์ถึงศกร์ และเสาร์ถึงอาทิตย์ตามลำดับ ซึ่งตรงกับเวลาเปิดปิดของห้องปฏิบัติการ

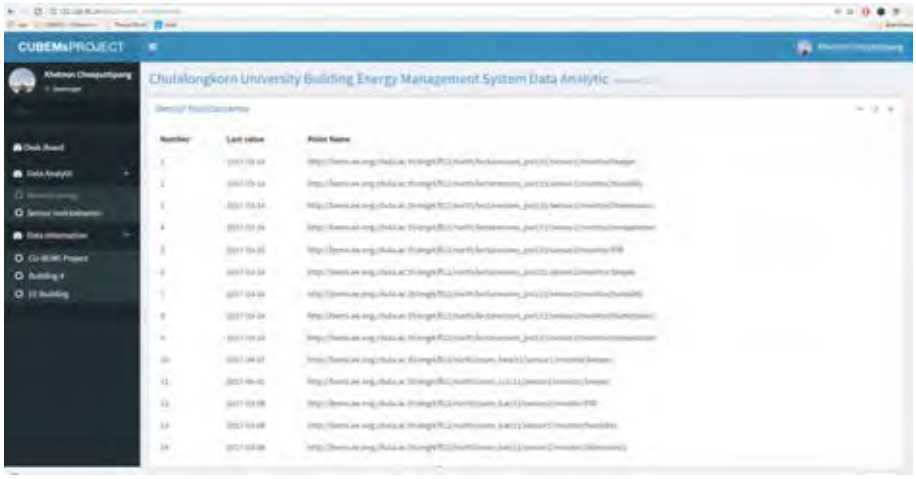


รูปที่ 3.22: ค่าเฉลี่ยการบริโภคพลังงานในหน่วยวัตต์ชั่วโมงบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเดือนมกราคม พ.ศ. 2560

เพื่อแก้ไขปัญหาในเรื่องช่องทางการแสดงผลที่มีอยู่อย่างจำกัดในโครงการ CUBEMS วิทยานิพนธ์ฉบับได้ออกแบบเว็บไซต์ที่ใช้สำหรับแสดงข้อมูลรูปแบบต่าง ๆ ยกตัวอย่างเช่น รูปที่ 3.23 แสดงสัดส่วนความต้องการพลังงานของอาคารแต่ล ะอาคารในโครงการ CUBEMS รูปที่ 3.24 แสดงข้อมูลการบริโภคพลังงานย้อนหลัง 6 เดือนของห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร รวมถึงออกแบบฟังก์ชันสำหรับแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์เมื่อเกิดเหตุการณ์ที่ตัวรับรู้มีสถานะการทำงานผิดปกติดังแสดงในรูปที่ 3.25



รูปที่ 3.23: การแสดงผลบนเว็บไซต์วิเคราะห์ข้อมูล



รูปที่ 3.24: การแสดงผลบนเว็บไซต์วิเคราะห์ข้อมูล (เพิ่มเติม)

รูปที่ 3.25: การแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์เมื่อเกิดเหตุการณ์ที่ตัวรู้มีสถานะการทำงานผิดปกติ

บทที่ 4

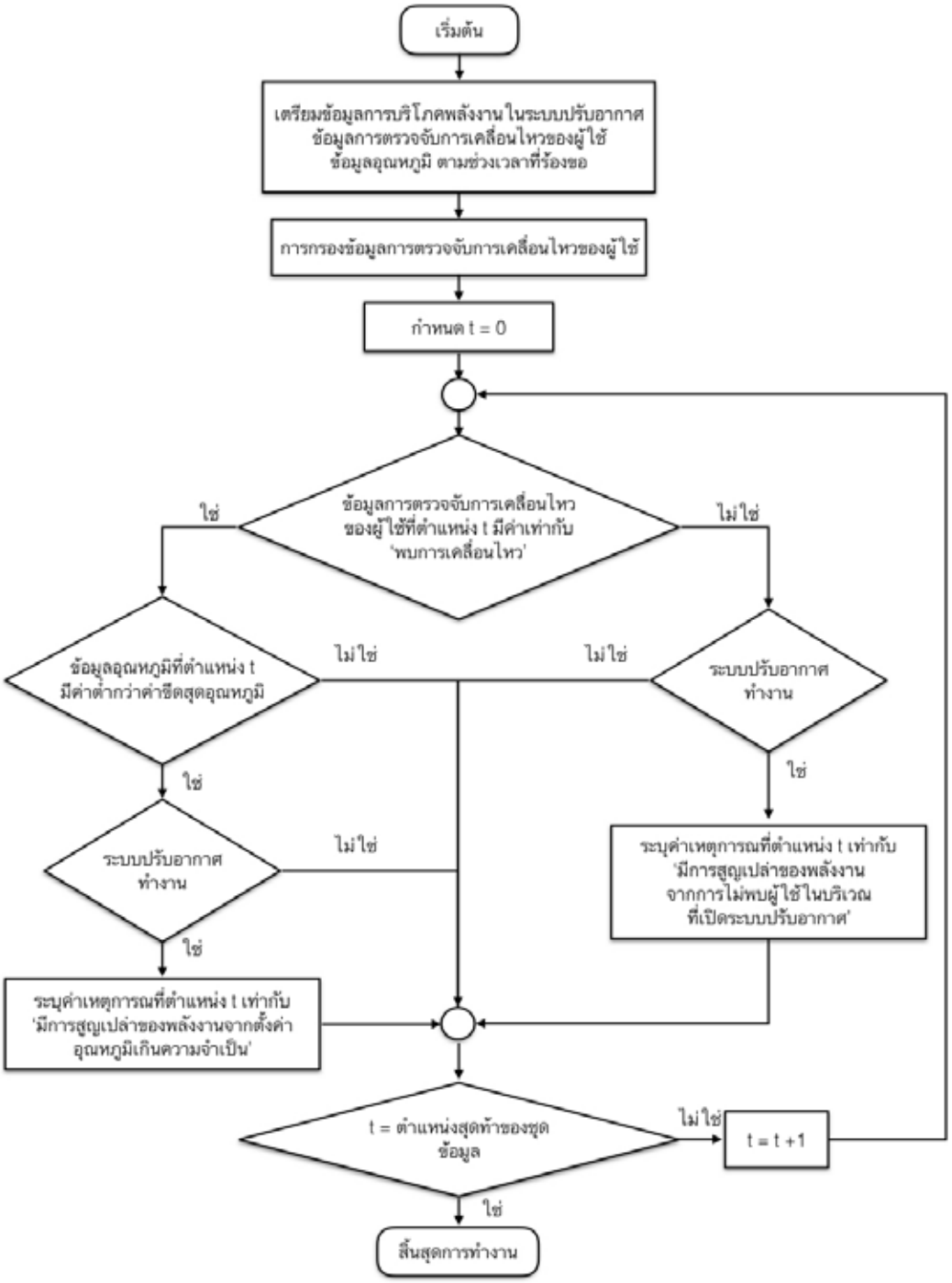
การสร้างและทดสอบฟังก์ชันวิเคราะห์ข้อมูล ของโปรแกรมประยุกต์

วิทยานิพนธ์ฉบับนี้นำเสนอการออกแบบฟังก์ชันวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานภายในอาคาร โดยอาศัยหลักการวิเคราะห์สหสัมพันธ์ระหว่างข้อมูลพฤติกรรมการเคลื่อนไหวของผู้ใช้ร่วมกับข้อมูลประเภทต่าง ๆ รวมถึงผู้วิจัยสนใจอาคารที่ตั้งอยู่ในพื้นที่เขตร้อน ซึ่งระบบปรับอากาศมีสัดส่วนการบริโภคพลังงานมากที่สุดดังแสดงในรูปที่ 3.21 วิทยานิพนธ์ฉบับนี้จึงได้มุ่งออกแบบและพัฒนาฟังก์ชันของโปรแกรมประยุกต์สำหรับใช้วิเคราะห์ข้อมูลเพื่อหาความสัมพันธ์ระหว่างข้อมูลการบริโภคพลังงาน ข้อมูลอุณหภูมิ และข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ เพื่อนำเสนอตัวชี้วัดความสูญเปล่าของพลังงานในระบบปรับอากาศเนื่องจากมีการเปิดให้ระบบทำงานในบริเวณที่ไม่มีผู้ใช้งานอยู่ หรือมีการปรับอุณหภูมิให้ต่ำเกินไป ผลลัพธ์ที่ออกมาในรูปแบบข่าวสารชนิดใหม่ที่สามารถมีส่วนช่วยสร้างการตระหนักรู้แก่ผู้ใช้อาคาร เกิดเป็นการปรับตัวระยะสั้นในรูปแบบการปรับเปลี่ยนพฤติกรรม หรือการปรับตัวระยะยาวในรูปแบบนโยบายของอาคารต่อไปในอนาคต

4.1 การออกแบบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับ-อากาศ

ฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศจะอาศัยการตีความข้อมูลการบริโภคพลังงาน ข้อมูลอุณหภูมิ และข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ เพื่อระบุประเภทของเหตุการณ์ที่ได้จากการวิเคราะห์ในช่วงเวลาต่าง ๆ ซึ่งประกอบไปด้วย การสูญเปล่าของพลังงานจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ และการสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น โดยมีผังงานแสดงในรูปที่ 4.1

วิทยานิพนธ์ฉบับนี้ได้เลือกใช้ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ในการอธิบายถึงพฤติกรรมการเคลื่อนไหวในช่วงเวลาต่าง ๆ ของเขตพื้นที่ที่พิจารณา แต่เนื่องจากลักษณะเฉพาะของข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ในโครงการ CUBEMS [18] ที่ถูกตรวจวัดโดยตัวรับรู้การเคลื่อนไหวชนิด PIR ซึ่งมีหลักการตรวจวัดเป็นรายคาบทั้งสิ้นสามรูปแบบดังแสดงในตารางที่ 4.1 และรูปที่ 4.2 ทำให้ในบางกรณีเกิดความผิดพลาดในการตรวจวัดข้อมูล เนื่องจากผู้ใช้ไม่ขยับร่างกายเป็นระยะเวลามากกว่าคาบการตรวจวัด รวมถึงเมื่อพิจารณาร่วมกับข้อจำกัดในเรื่องจำนวนตัวรับรู้ที่แตกต่างกันในแต่เขตพื้นที่ย่อยแต่ละเขต ซึ่งยากต่อการระบุว่าผู้ใช้มีกิจกรรมในเขตพื้นที่ดังกล่าว หรือเพียงเดินผ่านเพื่อไปยังพื้นที่อีกบริเวณ เพื่อแก้ไขปัญหาข้างต้นวิทยานิพนธ์ฉบับนี้ได้ออกแบบชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ที่ประกอบด้วย ตัวกรองการจัดกลุ่ม ตัวกรองการเดินผ่าน และตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว คาดหวังให้ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองข้างต้น สามารถอธิบายถึงพฤติกรรมการเคลื่อนไหวของผู้ใช้ในพื้นที่แต่ละบริเวณได้ชัดเจนมากยิ่งขึ้น รวมถึงออกแบบตัวกรองข้อมูลเพิ่มอีกหนึ่งชั้น สำหรับใช้คัดกรองเหตุการณ์ที่ผู้ใช้หายจากไปจากพื้นที่เพียงช่วงระยะเวลาหนึ่ง ซึ่งในความเป็นจริงการปิดระบบปรับอากาศในช่วงเวลาสั้น ๆ จะเป็นการสิ้นเปลืองพลังงานมากกว่าการเปิดระบบแบบต่อเนื่องเพราะระบบต้องทำการลดอุณหภูมิในช่วงแรกของจากการเปิดใช้อีกรอบ



รูปที่ 4.1: ผังงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

ตารางที่ 4.1: หลักการตรวจวัดข้อมูลของตัวรับรู้การเคลื่อนไหวชนิด PIR ของโครงการ CUBEMS

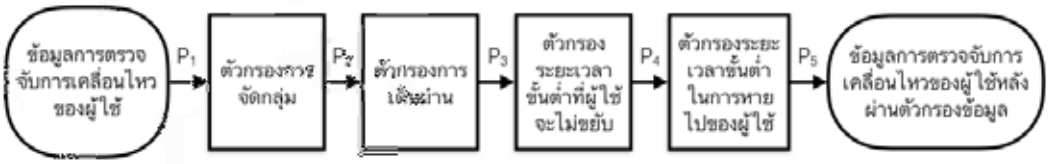
	เหตุการณ์
รูปแบบที่ 1	ตรวจวัดเป็นรายคาบ 60 วินาที
รูปแบบที่ 2	เมื่อมีการตรวจพบการเคลื่อนไหว ตัวรับรู้จะทำการจัดเก็บค่าข้อมูล ณ เวลาดังกล่าวแบบทันทีทันใด
รูปแบบที่ 3	หากเกิดเหตุการณ์ที่ตัวรับรู้ตรวจพบการเคลื่อนไหว ตัวรับรู้จะทำการตรวจวัดใหม่ในเวลา 40 วินาทีถัดไป

	TIME	VALUE	รูปแบบการตรวจวัด
0	2017-02-14 08:46:22	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
1	2017-02-14 08:47:21	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
2	2017-02-14 08:48:12	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 1
3	2017-02-14 08:48:20	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 2
4	2017-02-14 08:48:57	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 3
5	2017-02-14 08:49:18	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
6	2017-02-14 08:49:23	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 1
7	2017-02-14 08:50:14	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
8	2017-02-14 08:50:17	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 3
9	2017-02-14 08:51:16	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
10	2017-02-14 08:52:14	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
11	2017-02-14 08:53:06	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 1
12	2017-02-14 08:53:13	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 2
13	2017-02-14 08:53:51	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 3
14	2017-02-14 08:54:11	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
15	2017-02-14 08:55:01	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 1
16	2017-02-14 08:55:11	ON	เหตุการณ์ตรวจวัดรูปแบบที่ 2
17	2017-02-14 08:55:42	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 3
18	2017-02-14 08:56:09	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
19	2017-02-14 08:57:07	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
20	2017-02-14 08:58:06	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2
21	2017-02-14 08:59:04	OFF	เหตุการณ์ตรวจวัดรูปแบบที่ 2

รูปที่ 4.2: ตัวอย่างและรายระเอียดการตรวจวัดข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4

4.2 การออกแบบชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้

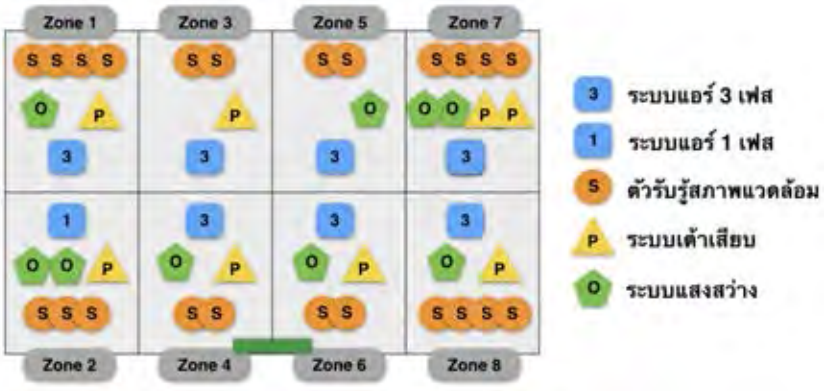
รูปที่ 4.3 แสดงชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ ซึ่งประกอบไปด้วยตัวกรองการจัดกลุ่ม ตัวกรองการเดินผ่าน ตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว และตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ และนิยามตัวแปร P_1, P_2, P_3, P_4, P_5 แทนชุดข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ซึ่งเป็นตัวแปรขาเข้าหรือตัวแปรขาออกของตัวกรองข้อมูล



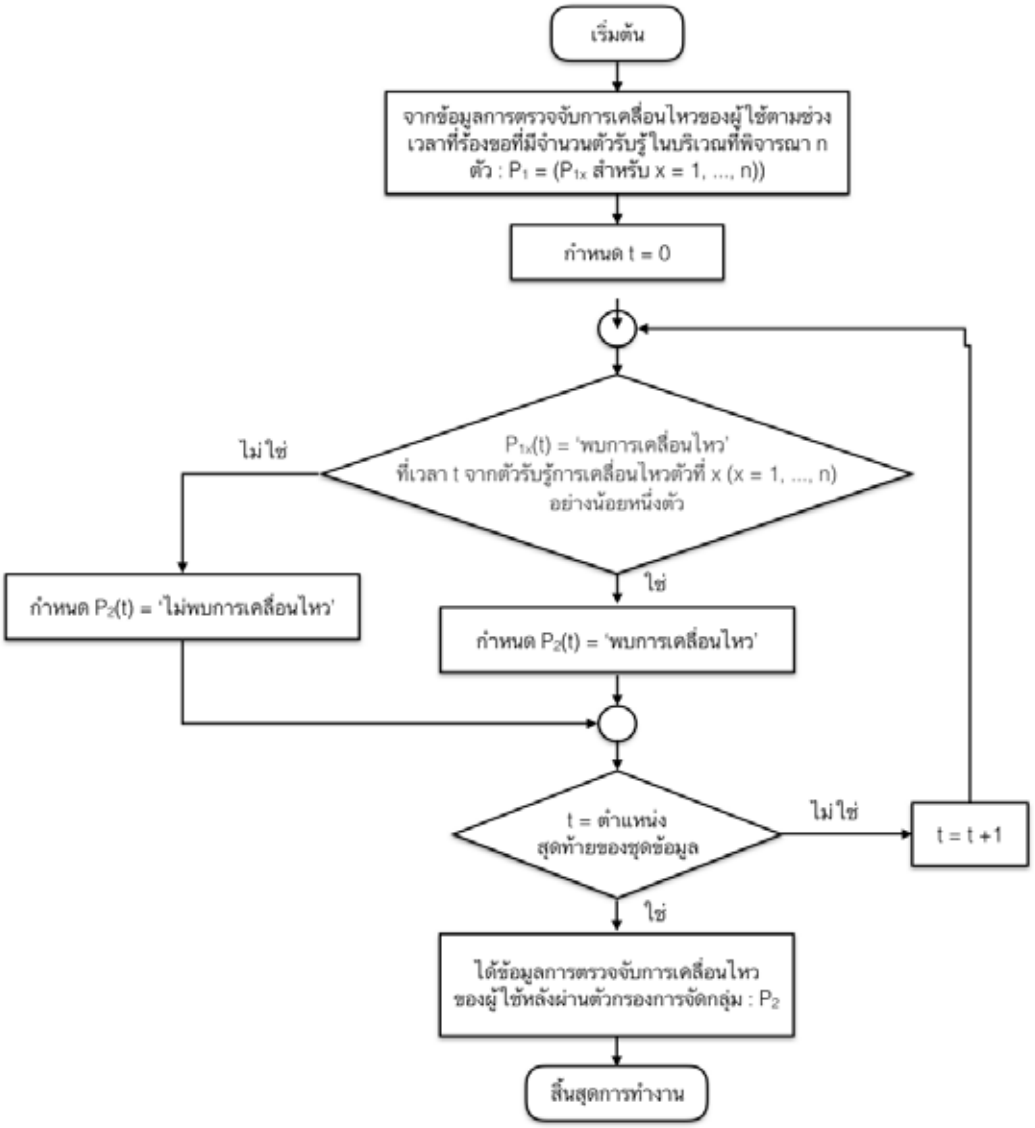
รูปที่ 4.3: ชุดตัวกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้

4.2.1 ตัวกรองการจัดกลุ่ม

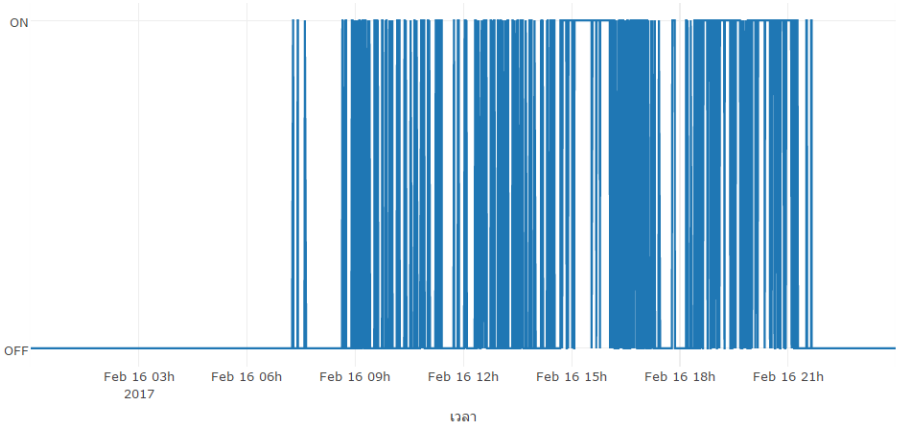
เพื่อแก้ไขปัญหาเรื่องความแตกต่างของคาบการตรวจวัดที่มีรูปแบบเฉพาะของตัวรับรู้แต่ละประเภท รวมถึงจำนวนของตัวรับรู้ที่แตกต่างกันในเขตพื้นที่ย่อยแต่ละเขตของโครงการ CUBEMS ดังแสดงในตารางที่ 2.1 และ 2.2 วิทยานิพนธ์ฉบับนี้ได้นำเสนอตัวกรองการจัดกลุ่มซึ่งมีผังงานแสดงในรูปที่ 4.5 เพื่อใช้ระบุค่าเหตุการณ์การตรวจจับการเคลื่อนไหวของผู้ใช้ในช่วงเวลาต่าง ๆ ของเขตพื้นที่ย่อยที่พิจารณา โดยมีหลักการทำงานคือ โปรแกรมประยุกต์จะระบุค่าเหตุการณ์ในช่วงเวลาดังกล่าวว่าพบการเคลื่อนไหวก็ต่อเมื่อตัวรับรู้อย่างน้อยหนึ่ง ตัวในเขตพื้นที่ย่อยที่พิจาณามีการระบุค่าเหตุการณ์ว่าตรวจพบการเคลื่อนไหว โดยในรูปที่ 4.6 และ 4.7 แสดงข้อมูลการตรวจจับการเคลื่อนไหวของตัวรับรู้ที่ 1 และ 2 ในวันที่ 16 กุมภาพันธ์ พ.ศ. 2560 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4 ที่มีแผนผังแสดงในรูปที่ 4.4 ซึ่งข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการจัดกลุ่มจะมีช่วงเวลาที่ระบุถึงเหตุการณ์ที่ตัวรับรู้ตรวจพบการเคลื่อนไหวลดลงเมื่อเทียบกับชุดข้อมูลก่อนผ่านตัวกรองดังแสดงในรูปที่ 4.8



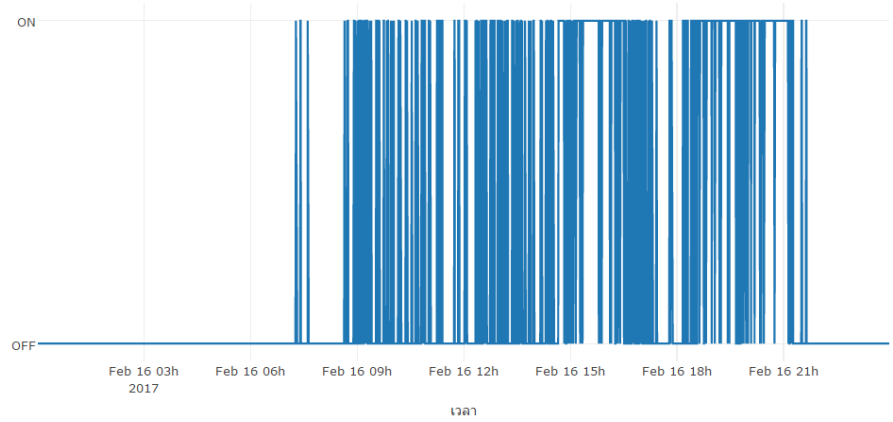
รูปที่ 4.4: แผนผังแสดงเขตพื้นที่บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร ชั้น 13 อาคารเจริญวิศวกรรม



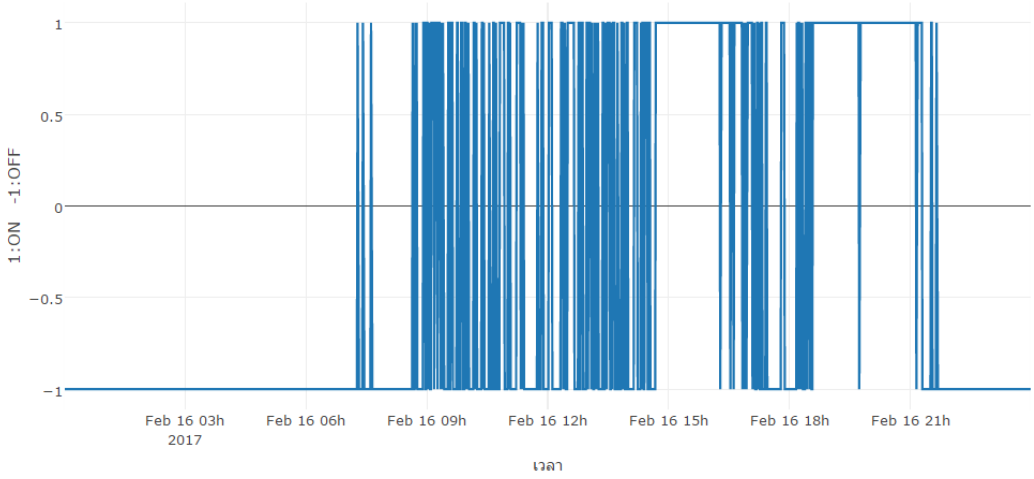
รูปที่ 4.5: ผังงานตัวกรองการจัดกลุ่ม



รูปที่ 4.6: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ของตัวรับรู้ที่ 1 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4



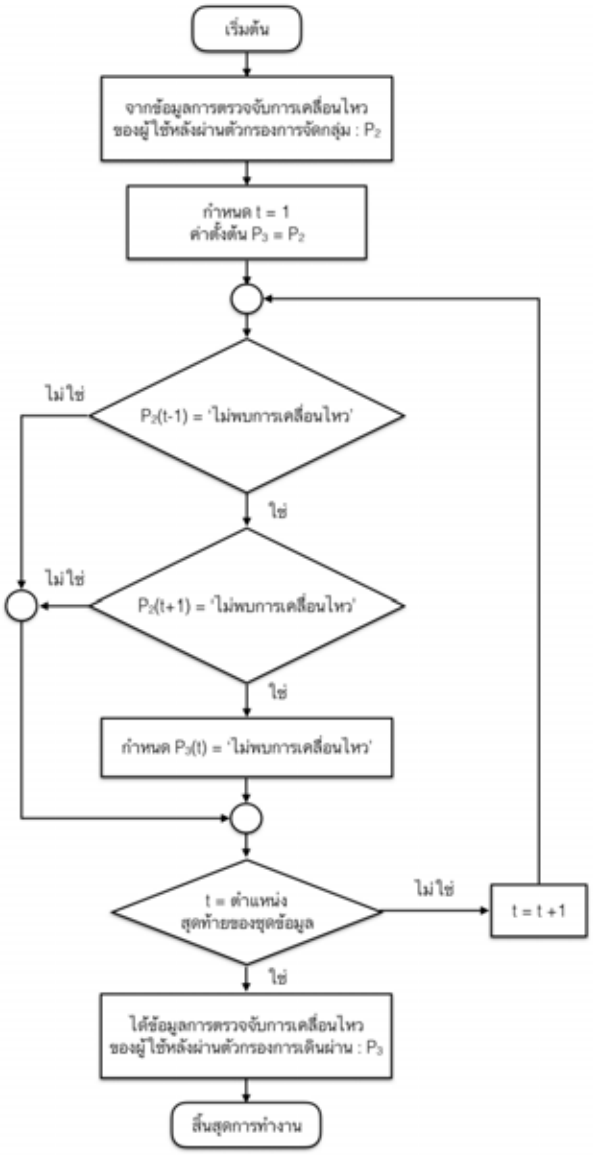
รูปที่ 4.7: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ของตัวรับรู้ที่ 2 บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเขตพื้นที่ย่อย 4



รูปที่ 4.8: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการจัดกลุ่ม

4.2.2 ตัวกรองการเดินผ่าน

รูปที่ 4.9 แสดงผังงานตัวกรองการเดินผ่านที่ใช้สำหรับปรับปรุงข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการจัดกลุ่มเพื่อคัดกรองเหตุการณ์ที่ผู้ใช้จำเป็นต้องเดินผ่านพื้นที่บริเวณอื่นเพื่อไปยังบริเวณที่ต้องการ โดยโปรแกรมประยุกต์จะปรับเปลี่ยนค่าเหตุการณ์ในทุกช่วงเวลาที่ถูกระบุเท่ากับ "พบการเคลื่อนไหว" เป็น "ไม่พบการเคลื่อนไหว" ถ้าค่าเหตุการณ์ในช่วงเวลาก่อนหน้า และหลังที่พิจารณามีค่าเท่ากับ "ไม่พบการเคลื่อนไหว" โดยในรูปที่ 4.10 แสดงตัวอย่างการทำงานของตัวกรองการเดินผ่าน ซึ่งข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการเดินผ่านจะมีช่วงเวลาที่ระบุถึงเหตุการณ์ที่ตัวรับรู้ตรวจพบการเคลื่อนไหวลดลง เมื่อเปรียบเทียบกับชุดข้อมูลก่อนผ่านตัวกรองดังแสดงในรูปที่ 4.11



รูปที่ 4.9: ผังงานตัวกรองการเดินผ่าน

2017-02-14 08:52:00	OFF
2017-02-14 08:53:00	OFF
2017-02-14 08:54:00	ON
2017-02-14 08:55:00	OFF
2017-02-14 08:56:00	ON
2017-02-14 08:57:00	OFF

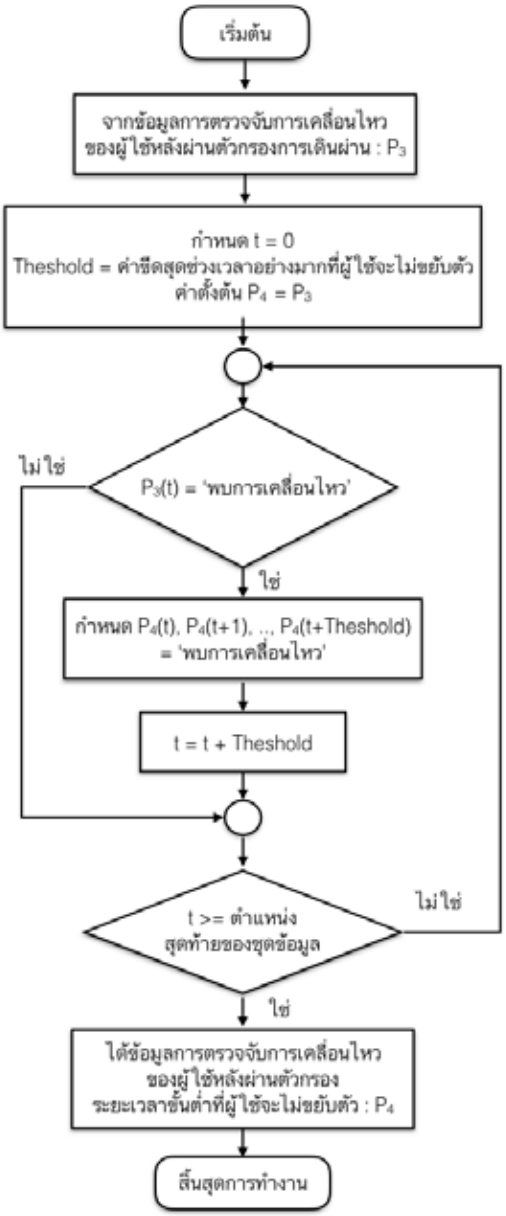
2017-02-14 08:52:00	OFF
2017-02-14 08:53:00	OFF
2017-02-14 08:54:00	OFF
2017-02-14 08:55:00	OFF
2017-02-14 08:56:00	OFF
2017-02-14 08:57:00	OFF

รูปที่ 4.10: ตัวอย่างการทำงานของตัวกรองการเดินผ่าน

รูปที่ 4.11: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองการเดินผ่าน

4.2.3 ตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว

เช่นเดียวกับในงานวิจัยที่ [13] เพื่อแก้ไขเหตุการณ์ที่ตัวรับรู้การเคลื่อนไหวชนิด PIR ตรวจวัดข้อมูลผิดพลาดเนื่องจากผู้ใช้ไม่ขยับร่างกายเป็นระยะเวลาที่มากกว่าคาบการตรวจวัด วิทยานิพนธ์ฉบับนี้ได้ออกแบบตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวที่มีผังงานแสดงในรูปที่ 4.12 โดยมีหลักการทำงานคือในทุกช่วงเวลาที่ตัวรับรู้ระบุค่าเหตุการณ์เท่ากับ "พบการเคลื่อนไหว" โปรแกรมประยุกต์จะปรับเปลี่ยนค่าเหตุการณ์ตลอดช่วงเวลาถัดไปตามค่าขีดสุดให้กลายเป็น "พบการเคลื่อนไหว" ดังแสดงในรูปที่ 4.13 ซึ่งข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวจะมีช่วงเวลาที่ระบุถึงเหตุการณ์ที่ตัวรับรู้ตรวจพบการเคลื่อนไหวเพิ่มขึ้น เมื่อเปรียบเทียบกับชุดข้อมูลก่อนผ่านตัวกรองดังแสดงในรูปที่ 4.14

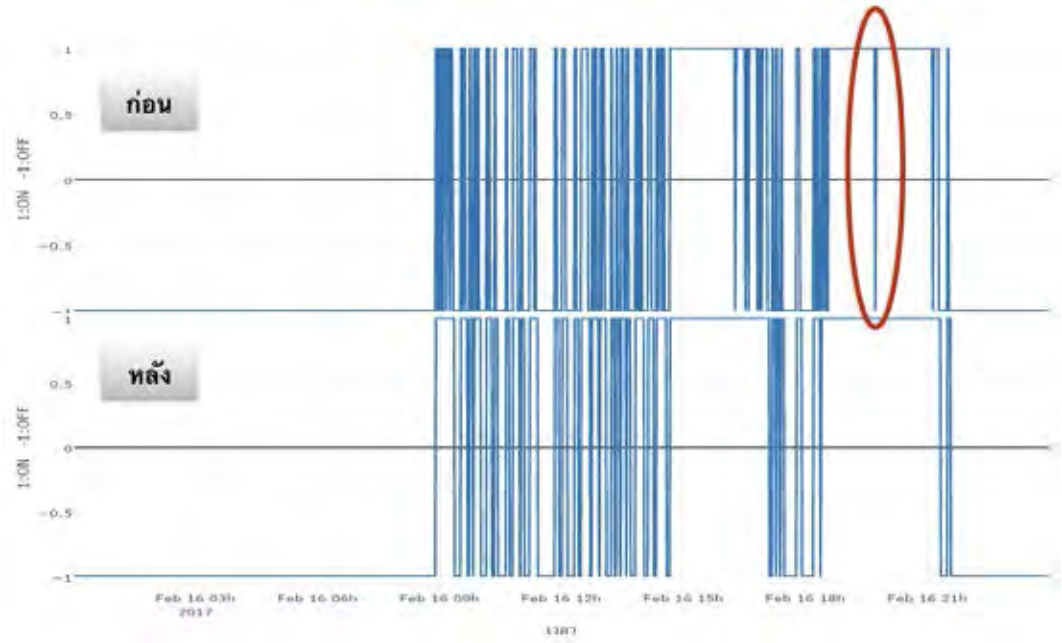


รูปที่ 4.12: ผังงานตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว

2017-02-14 08:50:00	ON
2017-02-14 08:51:00	OFF
2017-02-14 08:52:00	OFF
2017-02-14 08:53:00	OFF
2017-02-14 08:54:00	OFF

2017-02-14 08:50:00	ON
2017-02-14 08:51:00	ON
2017-02-14 08:52:00	ON
2017-02-14 08:53:00	OFF
2017-02-14 08:54:00	OFF

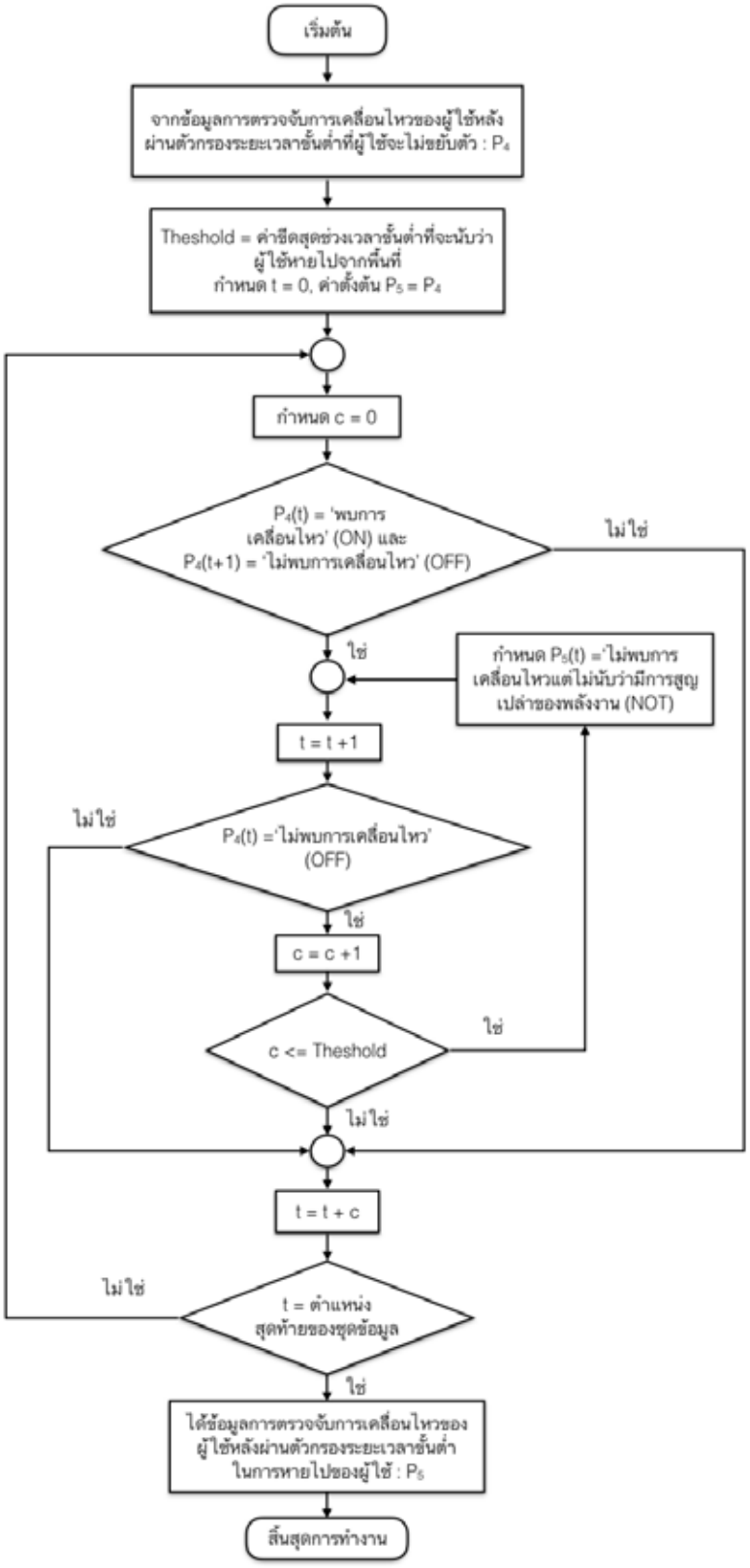
รูปที่ 4.13: ตัวอย่างการทำงานของตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวโดยกำหนดค่าขีดสุดเท่ากับ 2 นาที



รูปที่ 4.14: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัวโดยกำหนดค่าขีดสุดเท่ากับ 2 นาที

4.2.4 ตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้

เพื่อให้ผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศมีความชัดเจนมากยิ่งขึ้น ผู้วิจัยจึงได้ออกแบบตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้สำหรับปรับเปลี่ยนข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ โดยมีผังงานแสดงในรูปที่ 4.15 กำหนดตราเวลาที่จะเริ่มคำนวณการสูญเปล่าของพลังงานหลังจากผู้ใช้อาคารหายไปจากพื้นที่เป็นระยะเวลาหนึ่งมีหลักการทำงานคือ ในทุกตราเวลาที่ตัวรับรู้เปลี่ยนค่าเหตุการณ์จาก "พบการเคลื่อนไหว" เป็น "ไม่พบการเคลื่อนไหว" โปรแกรมประยุกต์จะปรับเปลี่ยนเหตุการณ์ตลอดช่วงเวลาถัดไปตามค่าขีดสุดให้กลายเป็น "ไม่มีการสูญเปล่าของพลังงาน" จนกระทั่งพบตราเวลาที่ระบุค่าค่าเหตุการณ์เท่ากับ "พบการเคลื่อนไหว" ดังแสดงในรูปที่ 4.16 ซึ่งผลลัพธ์จากกระบวนการข้างต้นจะทำให้ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ถูกแบ่งออกเป็นสามกรณีประกอบไปด้วย พบการเคลื่อนไหว ไม่พบการเคลื่อนไหว และไม่มีการสูญเปล่าของพลังงาน



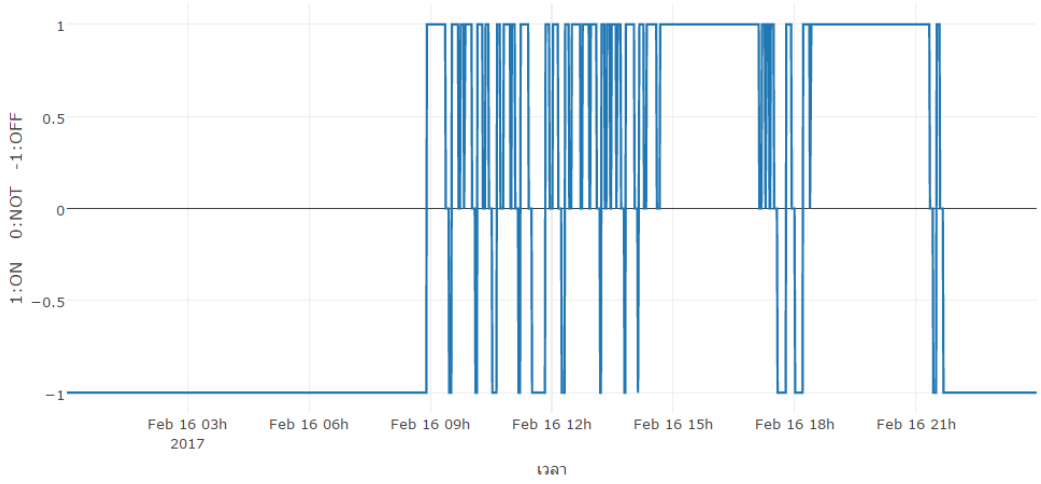
รูปที่ 4.15: ผังงานตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้

2017-02-14 08:51:00	ON
2017-02-14 08:52:00	ON
2017-02-14 08:53:00	OFF
2017-02-14 08:54:00	OFF
2017-02-14 08:55:00	OFF
2017-02-14 08:56:00	OFF
2017-02-14 08:57:00	OFF

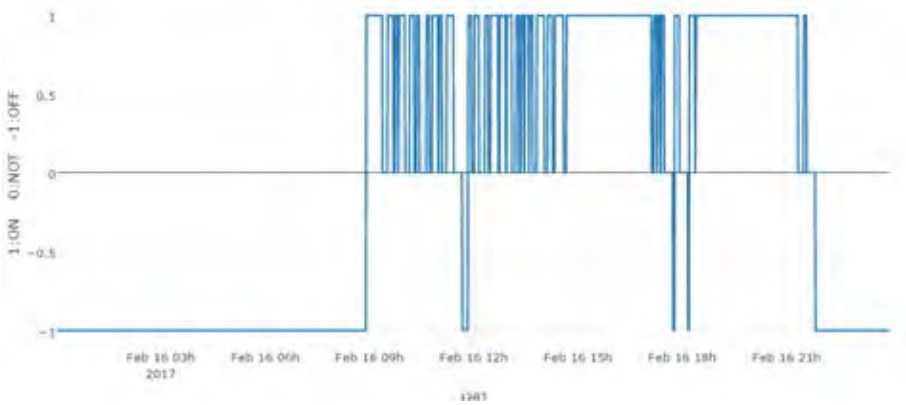
2017-02-14 08:51:00	ON
2017-02-14 08:52:00	ON
2017-02-14 08:53:00	NOT
2017-02-14 08:54:00	NOT
2017-02-14 08:55:00	NOT
2017-02-14 08:56:00	NOT
2017-02-14 08:57:00	OFF

รูปที่ 4.16: ตัวอย่างการทำงานของตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 4 นาที

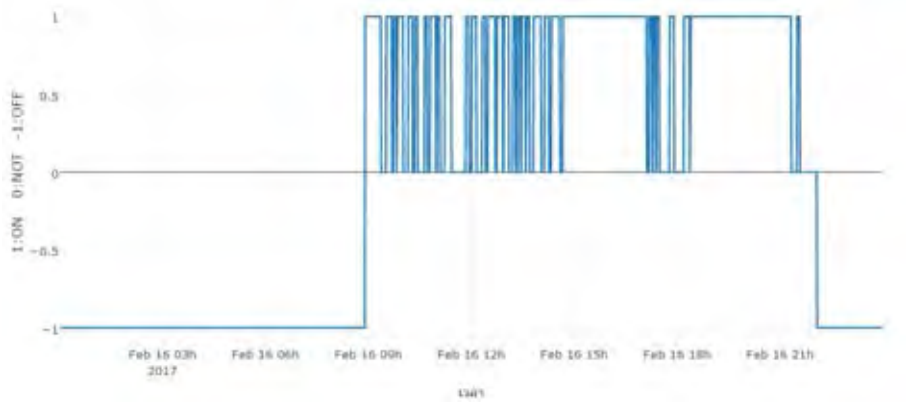
รูปที่ 4.17 - 4.20 แสดงข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ที่กำหนดค่าขีดสุดที่ผู้ใช้จะไม่ขยับตัวเท่ากับ 2 นาที และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 5 15 30 และ 60 นาที ตามลำดับ จะเห็นว่าจำนวนเหตุการณ์ที่ตัวรับรู้ไม่พบการเคลื่อนไหวของผู้ใช้มีจำนวนลดลงลดลงเมื่อเปรียบเทียบกับชุดข้อมูลก่อนผ่านตัวกรองโดยมีรายละเอียดแสดงในตารางที่ 4.2 และในรูปที่ 4.21 แสดงตัวอย่างภาพรวมกระบวนการกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ บริเวณเขตพื้นที่ 4 ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร ชั้น 13 อาคารเจริญวิศวกรรมในวันที่ 16 กุมภาพันธ์ พ.ศ. 2560 โดยกำหนดค่าขีดสุดที่ผู้ใช้จะไม่ขยับตัวเท่ากับ 2 นาที และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 5 นาที



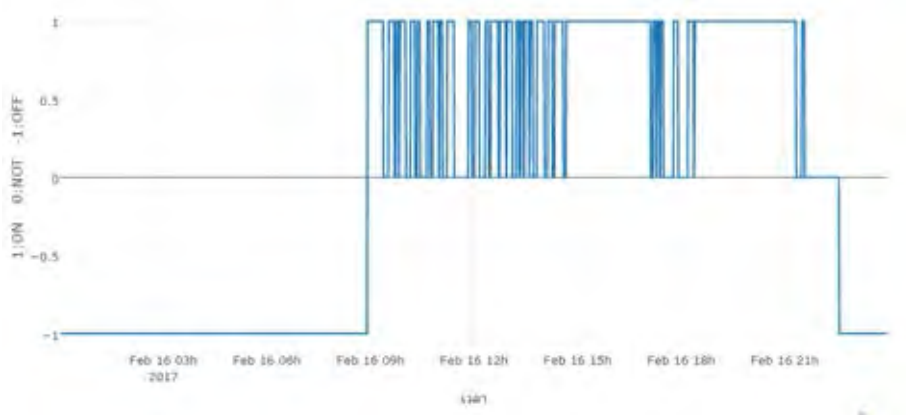
รูปที่ 4.17: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 5 นาที



รูปที่ 4.18: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 15 นาที



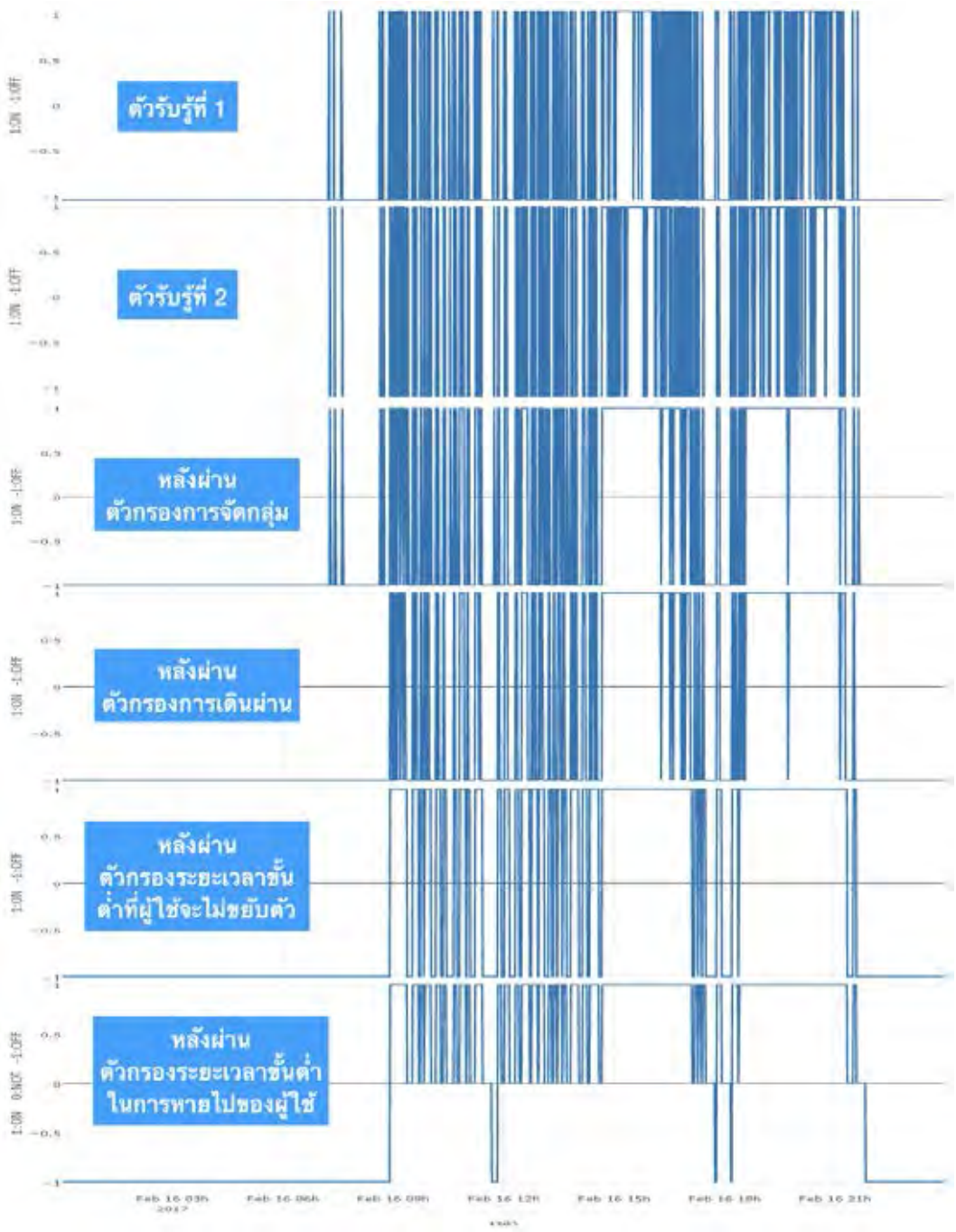
รูปที่ 4.19: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 30 นาที



รูปที่ 4.20: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้โดยกำหนดค่าขีดสุดเท่ากับ 60 นาที

ตารางที่ 4.2: ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ก่อนและหลังผ่านตัวกรองข้อมูล

ข้อมูลการตรวจจับการเคลื่อนไหว	พบการเคลื่อนไหว (ON)	ไม่พบการเคลื่อนไหวและจะนับว่ามีการสูญเปล่าของพลังงานหากเปิดระบบปรับอากาศ (OFF)	ไม่พบการเคลื่อนไหวแต่ไม่นับว่ามีการสูญเปล่าของพลังงาน (NOT)
ตัวรับรู้ ตัวที่ 1	542	1272	0
ตัวรับรู้ ตัวที่ 2	541	1254	0
ข้อมูลหลังผ่านตัวกรองการจัดกลุ่ม	469	970	0
ข้อมูลหลังผ่านตัวกรองการเดินผ่าน	438	1001	0
ข้อมูลหลังผ่านตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว	572	867	0
ข้อมูลหลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ที่ค่าขีดสุด 5 นาที	572	750	117
ข้อมูลหลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ที่ค่าขีดสุด 15 นาที	572	678	189
ข้อมูลหลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ที่ค่าขีดสุด 30 นาที	572	648	219
ข้อมูลตัวหลังผ่านตัวกรองระยะเวลาขั้นต่ำในการหายไปของผู้ใช้ที่ค่าขีดสุด 60 นาที	572	618	249



รูปที่ 4.21: ตัวอย่างภาพรวมกระบวนการกรองข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้

ขั้นตอนต่อไปคือการนำข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้หลังผ่านกระบวนการกรองข้อมูลข้างต้น ผสานเข้ากับข้อมูลการบริโภคพลังงาน และข้อมูลอุณหภูมิดังแสดงในรูปที่ 4.22 เพื่อระบุเหตุการณ์ของการวิเคราะห์ในแต่ละช่วงเวลาโดยอาศัยหลักเกณฑ์ตามตารางที่ 4.3 โดยในรูปที่ 4.23 - 4.24 แสดงตัวอย่างช่วงเวลาที่เกิดการสูญเปล่าของพลังงานในระบบปรับอากาศจากการหายไปของผู้ใช้อาคาร และตัวอย่างข้อมูลการสูญเปล่าของพลังงานในระบบปรับอากาศจากการตั้งค่าอุณหภูมิตามลำดับ

	Air (Wh)	Temp (celsius)	PIR
TIME			
2017-02-16 09:20:00	52.107	25.255	ON
2017-02-16 09:21:00	51.746	25.255	ON
2017-02-16 09:22:00	4.167	25.255	ON
2017-02-16 09:23:00	2.710	25.200	NOT
2017-02-16 09:24:00	2.704	25.200	NOT
2017-02-16 09:25:00	37.109	25.200	NOT
2017-02-16 09:26:00	55.423	25.200	NOT
2017-02-16 09:27:00	54.043	25.140	NOT
2017-02-16 09:28:00	52.187	25.140	OFF

รูปที่ 4.22: ตัวอย่างข้อมูลฟังก์ชันวิเคราะห์การสูญเปล่าในระบบปรับอากาศ

	Air (Wh)	Temp (celsius)	PIR
TIME			
2017-02-16 08:39:00	18.845	26.080	OFF
2017-02-16 08:40:00	45.842	26.080	OFF
2017-02-16 08:41:00	49.746	26.080	OFF
2017-02-16 08:42:00	50.206	26.080	OFF
2017-02-16 08:43:00	52.418	26.020	OFF
2017-02-16 08:44:00	52.050	25.965	OFF
2017-02-16 08:45:00	8.487	25.965	OFF
2017-02-16 08:46:00	2.733	25.905	OFF

รูปที่ 4.23: ตัวอย่างข้อมูลกรณีเกิดการสูญเปล่าของพลังงานจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ

ตารางที่ 4.3: หลักเกณฑ์การระบุเหตุการณ์การสูญเปล่าของพลังงานในระบบปรับบอากาศ

เงื่อนไข		**ข้อมูลการตรวจจับการเคลื่อนไหวของผู้ใช้ภายหลังจากผ่านตัวกรองทุกตัว : P_5**		
		(ON)	**(NOT)**	**(OFF)**
ระบบปรับอากาศทำงาน	**อุณหภูมิเฉลี่ยต่ำกว่าค่าขีดสุดอุณหภูมิ**	มีการสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น	ไม่มีเกิดการสูญเปล่าของพลังงาน	มีการสูญเปล่าของพลังงานจากการไม่พบผู้ใช้ในบริเวณที่เปิดระบบปรับอากาศ
	อุณหภูมิเฉลี่ยสูงกว่าค่าขีดสุดอุณหภูมิ	ไม่เกิดการสูญเปล่าของพลังงาน	ไม่เกิดการสูญเปล่าของพลังงาน	มีการสูญเปล่าของพลังงานจากการไม่พบผู้ใช้ในบริเวณที่เปิดระบบปรับอากาศ
ระบบปรับอากาศไม่ทำงาน	**อุณหภูมิเฉลี่ยต่ำกว่าค่าขีดสุดอุณหภูมิ**	ไม่เกิดการสูญเปล่าของพลังงาน		
	อุณหภูมิเฉลี่ยสูงกว่าค่าขีดสุดอุณหภูมิ			

	Air (Wh)	Temp (celsius)	PIR
TIME			
2017-02-16 08:55:00	51.700	25.555	ON
2017-02-16 08:56:00	51.996	25.495	ON
2017-02-16 08:57:00	52.285	25.495	ON
2017-02-16 08:58:00	52.013	25.495	ON
2017-02-16 08:59:00	52.511	25.495	ON
2017-02-16 09:00:00	52.619	25.495	ON
2017-02-16 09:01:00	52.478	25.495	ON
2017-02-16 09:02:00	51.489	25.495	ON
2017-02-16 09:03:00	52.448	25.495	ON

รูปที่ 4.24: ตัวอย่างข้อมูลกรณีเกิดการสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น

4.3 การทดสอบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

4.3.1 การทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน

การทดสอบจะอาศัยชุดข้อมูลบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 ซึ่งประกอบไปด้วยแปดเขตพื้นที่ย่อยดังแสดงในรูปที่ 4.4 รายละเอียดของจำนวนระบบที่มีการบริโภคพลังงาน รวมถึงจำนวนตัวรับรู้สภาพแวดล้อมของพื้นที่ทดสอบจะถูกอธิบายในตารางที่ 4.4 โดยกำหนดตัวแปรขาเข้าของชุดคำสั่งในภาคผนวกที่ จ.4 ตามตารางที่ 4.5

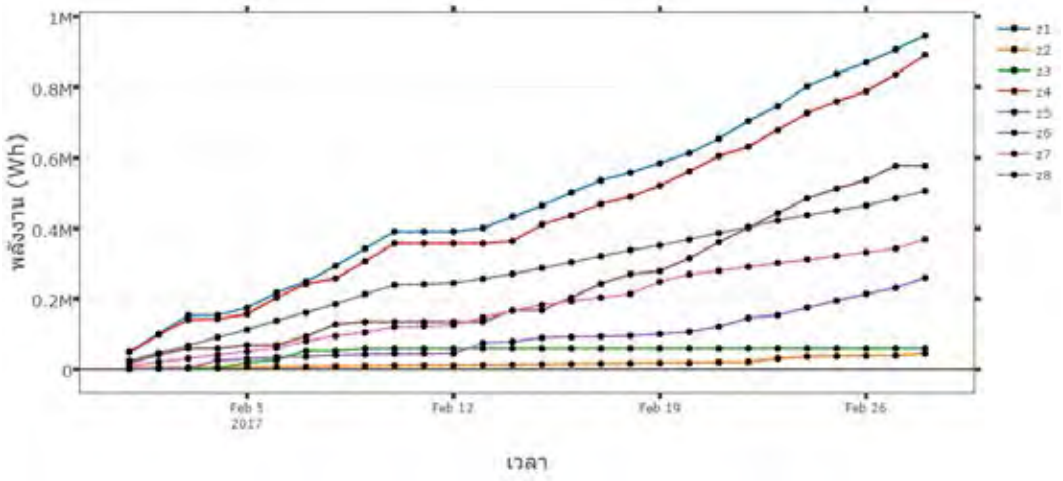
ตารางที่ 4.4: รายละเอียดจำนวนระบบและตัวรับรู้สภาพแวดล้อมของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร

ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	ระบบปรับอากาศ	ระบบแสงสว่าง	ระบบเต้าเสียบ	ตัวรับรู้สภาพแวดล้อม
เขตพื้นที่ย่อย 1	1 (3 เฟส)	1	1	4
เขตพื้นที่ย่อย 2	1 (1 เฟส)	2	1	3
เขตพื้นที่ย่อย 3	1 (3 เฟส)	0	1	2
เขตพื้นที่ย่อย 4	1 (3 เฟส)	1	1	2
เขตพื้นที่ย่อย 5	1 (3 เฟส)	1	0	2
เขตพื้นที่ย่อย 6	1 (3 เฟส)	1	1	2
เขตพื้นที่ย่อย 7	1 (3 เฟส)	2	2	4
เขตพื้นที่ย่อย 8	1 (3 เฟส)	1	1	4

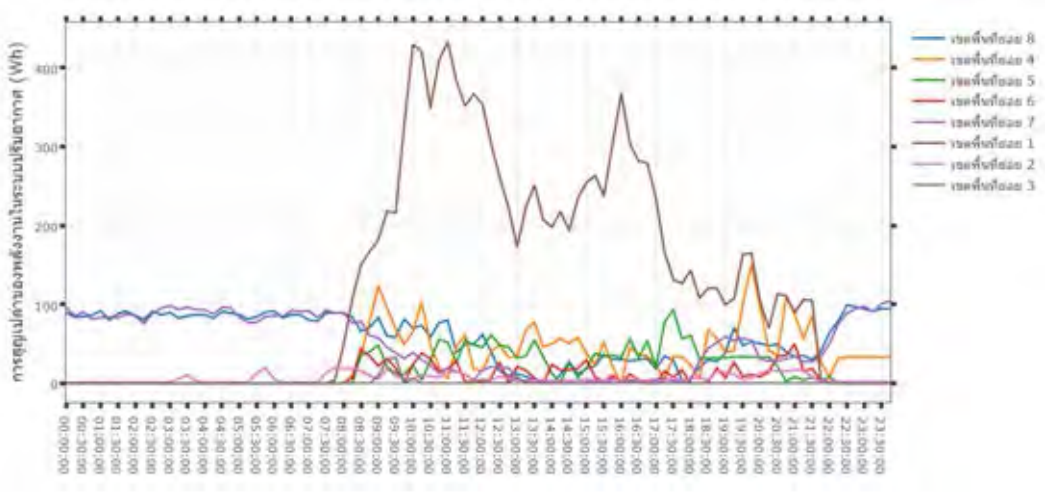
ตารางที่ 4.5: รายละเอียดตัวแปรขาเข้าของการทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน

ตัวแปรในชุดคำสั่ง	คำอธิบายตัวแปร	ค่าตัวแปร
Scale	ระดับพื้นที่การคำนวณ	room
qt	เวลาเริ่มต้นการวิเคราะห์	2017-02-01 00:00:00
lt	เวลาสิ้นสุดการวิเคราะห์	2017-02-28 23:59:00
Room	ห้องที่ทำการวิเคราะห์	lab_tsrl_dsprl_emrl
Temp_thres	ค่าขีดสุดอุณหภูมิ	24
thres	ค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่	15

รูปที่ 4.25 แสดงข้อมูลการบริโภคพลังงานในระบบปรับอากาศของเขตพื้นที่ย่อยแต่ละเขต ซึ่งผลลัพธ์ของการวิเคราะห์จะถูกแสดงในตารางที่ 4.6 จะเห็นว่าเขตพื้นที่ย่อย 2 มีสัดส่วนการสูญเปล่าของพลังงานในระบบปรับอากาศสูงเป็นอันดับหนึ่งคือ 38.7% แต่มีผลรวมการบริโภคพลังงานในระบบปรับอากาศตลอดเดือนต่ำเพียง 43.9 kWh ซึ่งอาจจะเกิดการชำรุดของระบบปรับอากาศในเขตพื้นที่ย่อยดังกล่าว และเมื่อพิจารณาเขตพื้นที่ย่อย 1 ที่มีการสูญเปล่าของพลังงานในระบบปรับอากาศมากเป็นอันดับหนึ่งคือ 340.1 kWh หรือคิดเป็นสัดส่วนมากถึง 35.9% ของการบริโภคพลังงานในระบบปรับอากาศของบริเวณดังกล่าว ในขณะที่เขตพื้นที่ย่อย 4 มีปริมาณการบริโภคพลังงานในระบบปรับอากาศสูงเป็นอันดับสองแต่มีสัดส่วนการสูญเปล่าของพลังงานเพียง 13.6% จากข้อมูลข้างต้นแสดงให้เห็นว่าเขตพื้นที่ย่อย 1 มีพฤติกรรมการบริโภคพลังงานไปในแนวทางสูญเปล่าเมื่อเปรียบเทียบกับเขตพื้นที่ย่อย 4 แต่เมื่อพิจารณาการบริโภคพลัง-งานในระบบปรับอากาศของเขตพื้นที่ย่อย 6 ซึ่งมีปริมาณการบริโภคพลังงานสูงเป็นอันดับสามรองจากเขตพื้นที่ย่อย 4 กลับมีสัดส่วนการสูญเปล่าของพลังงานเพียง 4.2% หรือคิดเป็น 21.5 Wh เป็นการแสดงให้เห็นว่าเขตพื้นที่ย่อย 6 มีพฤติกรรมการบริโภคพลังงานไปในแนวทางที่คุ้มค่ามากที่สุดเมื่อเปรียบเทียบกับเขตพื้นที่ย่อยอื่นในห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร และเมื่อพิจารณาค่าเฉลี่ยการสูญเปล่าของพลังงานในระบบปรับอากาศดังแสดงในรูปที่ 4.26 จะเห็นว่าเขตพื้นที่ย่อย 7 และ 8 มีค่าเฉลี่ยการสูญเปล่าของพลังงานสูงในช่วงเวลา 22:00 ถึง 08:00 ซึ่งเป็นช่วงเวลาที่ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารปิดทำการ และสอดคล้องกับสัดส่วนการสูญเปล่าของพลังงานในเขตพื้นที่ย่อยทั้งสองที่มากถึง 36.2% และ 31.1% จากเหตุผลข้างต้นแสดงให้เห็นว่าเกิดการชำรุดของระบบปรับอากาศบริเวณเขตพื้นที่ย่อยดังกล่าว



รูปที่ 4.25: ข้อมูลการบริโภคพลังงานสะสมตั้งแต่วันแรกของเดือนในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 ของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร



รูปที่ 4.26: ค่าเฉลี่ยการสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือน กุมภาพันธ์ พ.ศ. 2560 ของพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร

ตารางที่ 4.6: การสูญเปล่าของพลังงานในระบบปรับอากาศของเขตพื้นที่ย่อยแต่ละเขตบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารตลอดเดือนกุมภาพันธ์ พ.ศ. 2560

ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	การบริโภคพลังงานในระบบปรับอากาศ (kWh)	การสูญเปล่าของพลังงานจากการหายไปของผู้ใช้ (kWh)	การสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิ (kWh)	ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศ (kWh)
เขตพื้นที่ย่อย 1	946.7	340.1	0	340.1
เขตพื้นที่ย่อย 2	43.9	17	0	17
เขตพื้นที่ย่อย 3	59.3	3.2	0	3.2
เขตพื้นที่ย่อย 4	892.4	84.9	36.3	121.2
เขตพื้นที่ย่อย 5	259.5	52.1	0.3	52.3
เขตพื้นที่ย่อย 6	577.4	21.5	2.9	24.4
เขตพื้นที่ย่อย 7	368.6	133.8	0	133.8
เขตพื้นที่ย่อย 8	506.1	157.2	0	157.2

4.3.2 การทดสอบฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในลักษณะการปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่

หัวข้อนี้จะทดสอบปรับเปลี่ยนค่าขีดสุดอุณหภูมิ และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ คาดหวังว่าผลลัพธ์ของฟังก์ชันสามารถแสดงพฤติกรรมของผู้ใช้ได้ชัดเจนมากยิ่งขึ้น โดยการทดสอบจะอาศัยชุดข้อมูลบริเวณชั้น 13 อาคารเจริญวิศวกรรมตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 มีแผนผังแสดงในรูปที่ 4.27 รายละเอียดของจำนวนเขตพื้นที่ย่อย ระบบที่มีการบริโภคพลังงานไฟฟ้าและตัวรับรู้สภาพแวดล้อมประเภทต่าง ๆ จะถูกอธิบายในตารางที่ 4.7 โดยกำหนดตัวแปรขาเข้าของชุดคำสั่งในภาคผนวกที่ จ.4 ตามตารางที่ 4.8

รูปที่ 4.27: แผนผังแสดงเขตพื้นที่บริเวณชั้น 13 อาคารเจริญวิศวกรรม

จากการทดสอบจะเห็นว่าการเพิ่มค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ ส่งผลโดยตรงต่อการลดลงของสัดส่วนการสูญเปล่าของพลังงานโดยมีรายละเอียดตามตารางที่ 4.9 ทั้งนี้ความแต่งต่างของสัดส่วนการสูญเปล่าที่ลดลงสามารถแสดงถึงพฤติกรรมเชิงลึกของผู้ใช้ในพื้นที่ย่อยแต่ละเขตได้ชัดเจนมากยิ่งขึ้นดังแสดงในรูปที่ 4.28 ยกตัวอย่างเช่น ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารที่มีการสูญเปล่าของพลังงานเมื่อกำหนดค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 15 นาทีเป็นสัดส่วนมากถึง 22.3% โดยเมื่อทดลองปรับเปลี่ยนค่าขีดสุดเป็น 120 นาทีส่งผลให้สัดส่วนการสูญเปล่าของพลังงานลดเหลือ 12.1% ในขณะที่ห้องพักอาจารย์ pkp/pjp/was ที่มีสัดส่วนการสูญเปล่าของพลังงานใกล้เคียงกันคือ 21.3% เมื่อกำหนดค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 15 นาที แต่เมื่อปรับเปลี่ยนค่าขีดสุดเป็น 120 นาทีส่งผลให้สัดส่วนการสูญเปล่าของพลังงานลดเหลือเพียง 1.3% จะเห็นว่าการเปลี่ยนแปลงของสัดส่วนการสูญเปล่าของพลังงานแตกต่างจากกรณีห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารอย่างเห็นได้ชัด ชี้ให้เห็นว่าผู้ใช้บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารมีพฤติกรรมการเปิดระบบปรับอากาศทิ้งไว้เป็นเวลานานมากกว่าผู้ใช้บริเวณห้องพักอาจารย์ pkp/pjp/was ในส่วนของการทดสอบปรับเปลี่ยนค่าขีดสุดอุณหภูมิระหว่าง 24 และ 25 องศาเซลเซียสจะส่งผลต่อการเพิ่มขึ้นของผลรวมพลังงานที่สูญเปล่าในระบบปรับอากาศเพียงบางบริเวณดังแสดงในตารางที่ 4.10 ยกตัวอย่างเช่น การสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารเพิ่มขึ้น 47% จาก 811.9 kWh เป็น 1,199.73 kWh ในขณะที่ผลรวมการสูญเปล่าของพลังงานในระบบปรับ อากาศบริเวณห้องพักอาจารย์ทั้งหมดไม่มีการเปลี่ยนแปลงหลังจากปรับเปลี่ยนค่าขีดสุดอุณหภูมิ

ตารางที่ 4.7: รายละเอียดจำนวนระบบและตัวรับรู้สภาพแวดล้อมของห้องแต่ละห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรม

ชั้น 13 อาคารเจริญวิศวกรรม	จำนวนเขตพื้นที่ย่อย	ระบบปรับอากาศ	ระบบแสงสว่าง	ระบบเต้าเสียบ	ตัวรับรู้สภาพแวดล้อม
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	8	7 (3 เฟส) 1 (1 เฟส)	1	1	23
ห้องปฏิบัติการ AIS	1	1 (1 เฟส)	0	0	1
ห้องพักอาจารย์ pkp/pjp/was	1	1 (1 เฟส)	1	1	3
ห้องปฏิบัติการย่อยโทรคมนาคม	1	1 (1 เฟส)	0	0	1
ห้องกาแฟ	1	1 (1 เฟส)	1	0	1
ห้องพักอาจารย์ csc/jpl/ccr	1	1 (1 เฟส)	2	1	2
ห้องเรียน 2	1	1 (3 เฟส)	1	1	1
ห้องพักอาจารย์ sav/dwc/nts	1	1 (1 เฟส) 1 (3 เฟส)	2	2	3
ห้องพักอาจารย์ cak/cpp/lwk	1	1 (1 เฟส)	2	2	3
ห้องเรียนดีเอสพี	2	2 (3 เฟส)	1	3	4
ห้องเซิร์ฟเวอร์	1	3 (1 เฟส)	0	4	1

ตารางที่ 4.8: รายละเอียดตัวแปรขาเข้าของการทดสอบปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ของฟังก์ชันชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศน

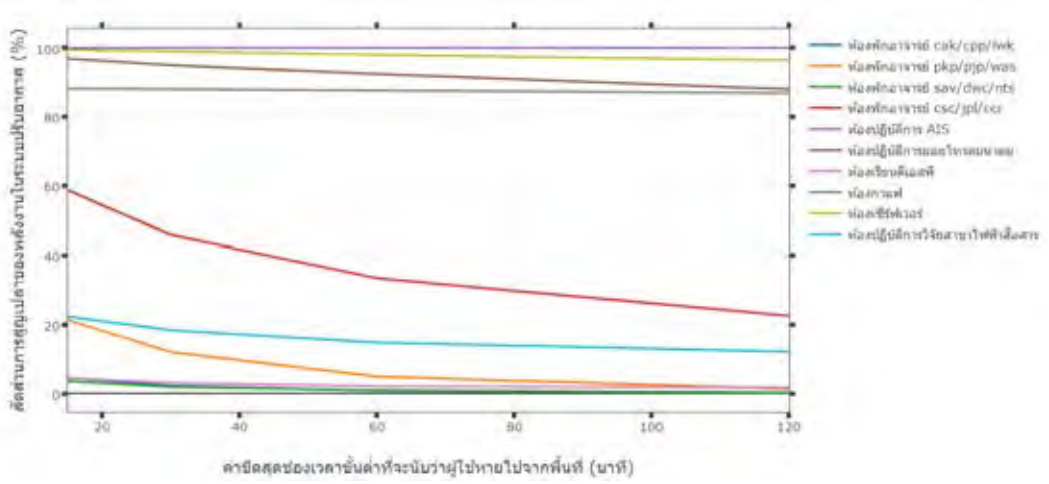
ตัวแปรในชุดคำสั่ง	คำอธิบายตัวแปร	ค่าตัวแปร
Scale	ระดับพื้นที่การคำนวณ	fl13
qt	เวลาเริ่มต้นการวิเคราะห์	2017-02-01 00:00:00
lt	เวลาสิ้นสุดการวิเคราะห์	2017-02-28 23:59:00
Temp_thres	ค่าขีดสุดอุณหภูมิ	24, 25
thres	ค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่	15, 30, 60, 120

ตารางที่ 4.9: การทดสอบปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560

ชั้น 13 อาคารเจริญวิศวกรรม	**การบริโภคพลังงานในระบบปรับอากาศ (kWh)**	**สัดส่วนการสูญเปล่าของพลังงานในระบบปรับอากาศ (พลังงานที่สูญเปล่า*100/พลังงานทั้งหมด)**			
		ค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่			
		15 นาที	**30 นาที**	**60 นาที**	**120 นาที**
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	3,645.65	22.3%	18.3%	14.8%	12.1%
ห้องเรียนดีเอสพี	521.06	4.4%	3.1%	2.1%	1.7%
ห้องพักอาจารย์ cak/cpp/lwk	500.56	4.5%	2.3%	0.8%	0.2%
ห้องเรียน 2	266.89	89.6%	88.4%	85.6%	82.6%
ห้องพักอาจารย์ csc/jpl/ccr	256.56	58.7%	45.8%	33.4%	22.5%
ห้องพักอาจารย์ sav/dwc/nts	158.66	3.5%	1.9%	0.9%	0.2%
ห้องพักอาจารย์ pkp/pjp/was	86.8	21.3%	12%	4.9%	1.3%
ห้องปฏิบัติการ AIS	84.08	100%	100%	100%	100%
ห้องกาแฟ	82.51	88.2%	88%	87.6%	86.9%
ห้องปฏิบัติการย่อยโทรคมนาคม	74.22	96.9%	95%	92.4%	88%

ตารางที่ 4.10: การทดสอบปรับเปลี่ยนค่าขีดสุดอุณหภูมิของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตลอดเดือนกุมภาพันธ์ พ.ศ. 2560

ชั้น 13 อาคารเจริญวิศวกรรม	การบริโภคพลังงานในระบบปรับอากาศ (kWh)	การสูญเปล่าของพลังงานในระบบปรับอากาศ			
		ค่าขีดสุดอุณหภูมิเท่ากับ 24 องศาเซลเซียส		ค่าขีดสุดอุณหภูมิเท่ากับ 25 องศาเซลเซียส	
		การสูญเปล่าของพลังงานจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ (kWh)	การสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น (kWh)	การสูญเปล่าของพลังงานจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ (kWh)	การสูญเปล่าของพลังงานจากการตั้งค่าอุณหภูมิเกินความจำเป็น (kWh)
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	3645.65	808.71	3.19	808.71	391.02
ห้องเรียนดีเอสพี	521.06	22.81	0	22.81	0
ห้องพักอาจารย์ cak/cpp/lwk	500.56	22.76	0	22.76	0
ห้องเรียน 2	266.89	239.17	0	239.17	0
ห้องพักอาจารย์ csc/jpl/ccr	256.56	150.65	0	150.65	0
ห้องพักอาจารย์ sav/dwc/nts	158.66	5.57	0	5.57	0
ห้องพักอาจารย์ pkp/pjp/was	86.8	18.45	0	18.45	0
ห้องปฏิบัติการ AIS	84.08	84.08	0	84.08	0
ห้องกาแฟ	82.51	72.7	0.03	72.7	0.81
ห้องปฏิบัติการย่อยโทรคมนาคม	74.22	71.91	0	71.91	0



รูปที่ 4.28: สัดส่วนการสูญเปล่าของพลังงานในระบบปรับอากาศเมื่อมีการปรับเปลี่ยนค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่

4.4 การวิเคราะห์ผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

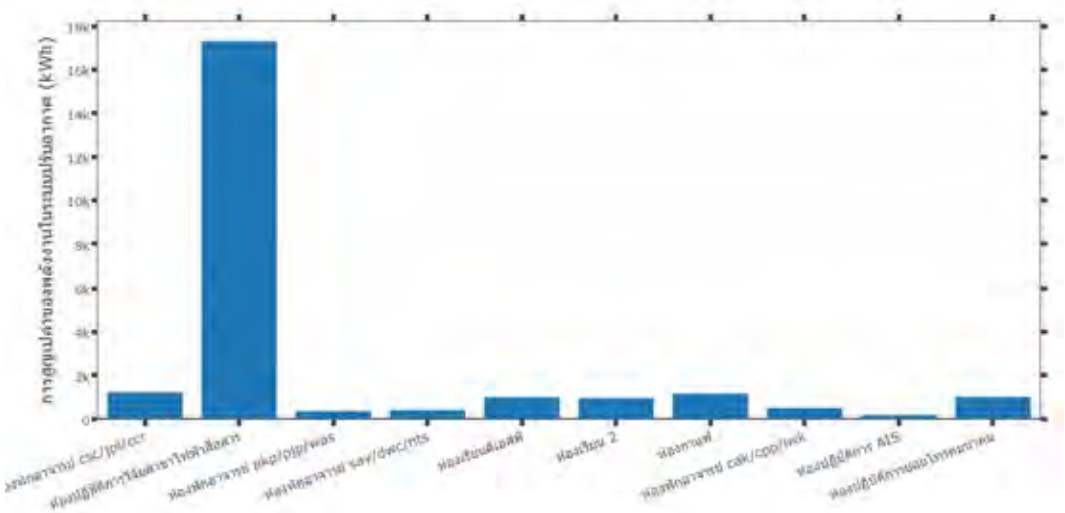
จากหัวข้อที่ 4.3 ได้แสดงผลทดสอบการทำงานเบื้องต้นของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ เพื่อให้การวิเคราะห์มีความชัดเจนมากยิ่งขึ้น ผู้วิจัยจึงได้ขยายขอบเขตช่วงเวลาในการวิเคราะห์ รวมถึงออกแบบการทดสอบเพื่อสังเกตตัวแปรที่จะส่งผลต่อการสูญเปล่าของพลังงานในระบบปรับอากาศ ประกอบไปด้วย

1. พื้นที่ต่าง ๆ ซึ่งมีผู้ใช้งานประจำที่แตกต่างกัน
2. อุณหภูมิของสภาพอากาศภายนอกห้องปรับอากาศในบริเวณที่ตั้งของเครื่องคอมเพรสเซอร์ระบบปรับอากาศ
3. การเปิดปิดของภาคการศึกษา

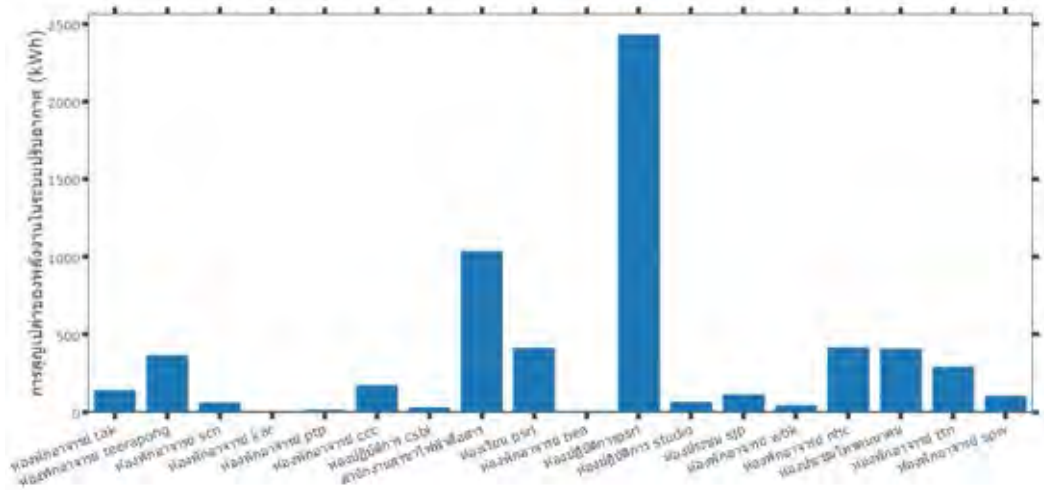
4.4.1 พื้นที่ต่าง ๆ และการสูญเปล่าของพลังงานในระบบปรับอากาศ

ในหัวข้อนี้จะทำการทดสอบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณชั้น 12 และ 13 อาคารเจริญวิศวกรรม ตลอดระยะเวลา 1 ปีเริ่มตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 จนถึงวันที่ 28 กุมภาพันธ์ พ.ศ. 2560 โดยกำหนดค่าขีดสุดอุณหภูมิเท่ากับ 24 องศาเซลเซียส และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 15 นาที ผลลัพธ์ของการวิเคราะห์ข้างต้นจะถูกแสดงในรูปที่ 4.29- 4.31 จะเห็นได้ว่าห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารบริเวณชั้น 13 มีผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศสูงที่สุด และจากรูปที่ 4.31 แสดงค่าเฉลี่ยการสูญเปล่าของพลังงานในช่วงเวลาต่าง ๆ ของวัน จะเห็นว่าค่าเฉลี่ยการสูญเปล่าของพลังงานส่วนมากจะมาจากพื้นที่บริเวณชั้น 13 โดยจะเริ่มมีค่าสูงขึ้นตั้งแต่เวลา 7 นาฬิกา ซึ่งเป็นเวลาเปิดทำการของห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร สอดคล้องกับข้อกำหนดการเปิดระบบ

ปรับอากาศของพื้นที่บริเวณดังกล่าว ซึ่งต้องเปิดระบบปรับอากาศเป็นจำนวนขั้นต่ำ 4 เครื่อง เพื่อป้องกันการทำงานเกินพิกัด ส่งผลให้ค่าเฉลี่ยการสูญเปล่าของพลังงานเพิ่มขึ้นอย่างต่อเนื่อง จนกระทั่งตอน 10 นาฬิกา ค่าเฉลี่ยการสูญเปล่าของพลังงานจะเริ่มลดลง เนื่องจากเป็นเวลาที่ผู้ใช้อาคารส่วนมากเริ่มเข้ามาปฏิบัติหน้าที่ จนกระทั่งเวลา 18 นาฬิกา ค่าเฉลี่ยการสูญเปล่าของพลังงานจะเข้าใกล้ค่าเฉลี่ยก่อนหน้าเวลา 7 นาฬิกา จากผลลัพธ์ข้างต้นจะเห็นได้ว่ายังมีโอกาสของการปรับลดการบริโภคพลังงานได้อีกมากเพียงการปรับวินัยของผู้ใช้อาคาร ยกตัวอย่างเช่น ผลรวมการสูญเปล่าของพลังงานตลอดปีบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารมีค่าสูงถึง 17,306 kWh หรือคิดเป็น 96,744 บาท ดังแสดงในตารางที่ 4.11



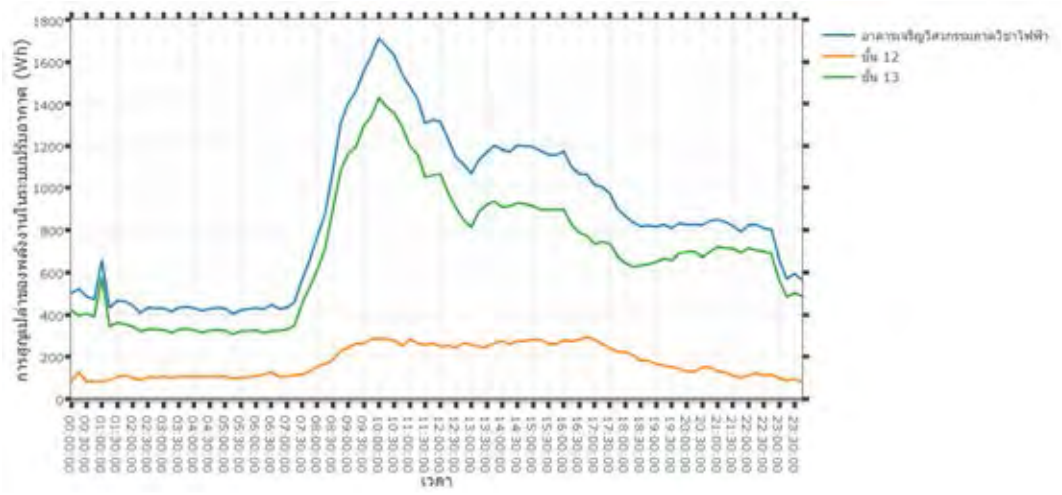
รูปที่ 4.29: ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศของห้องแต่ละห้องบริเวณชั้น 13 อาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560



รูปที่ 4.30: ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศของห้องแต่ละห้องบริเวณชั้น 12 อาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560

ตารางที่ 4.11: ผลรวมการสูญเปล่าของพลังงานและค่าใช้จ่ายจากพลังงานที่สูญเปล่าในระบบปรับอากาศบริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560

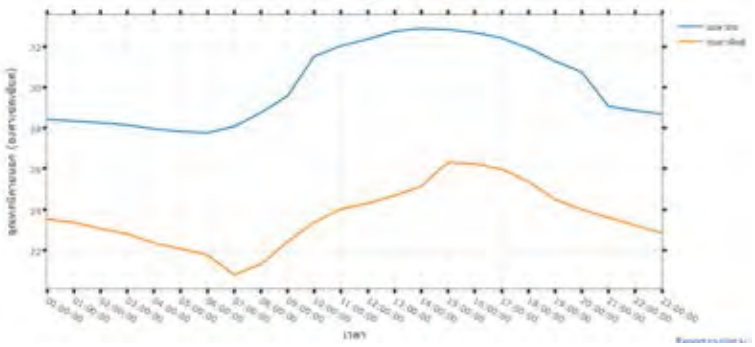
อาคารเจริญวิศวกรรม	การสูญเปล่าของพลังงานในระบบปรับอากาศ(kWh)	ค่าใช้จ่ายจากพลังงานที่สูญเปล่า (5.58 บาท ต่อ 1 kWh)
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	17,307	96,571
ห้องปฏิบัติการ psrl	2,434	13,581
ห้องพักอาจารย์ csc/jpl/ccr	1,219	6,800
ห้องกาแฟ	1,162	6,486
สำนักงานสาขาไฟฟ้าสื่อสาร	1,038	5,790
ห้องปฏิบัติการย่อยโทรคมนาคม	1,000	5,580
ห้องเรียนดีเอสพี	999	5,574
ห้องเรียน 2	954	5,321
ห้องพักอาจารย์ cak/cpp/lwk	500	2,791
ห้องพักอาจารย์ nhc	418	2,333
ห้องเรียน psrl	414	2,311
ห้องประชุมโทรคมนาคม	408	2,278
ห้องพักอาจารย์ sav/dwc/nts	406	2,264
ห้องพักอาจารย์ pkp/pjp/was	366	2,040
ห้องพักอาจารย์ teerapong	365	2,038
ห้องพักอาจารย์ ttn	293	1,638
ห้องพักอาจารย์ ccc	174	972
ห้องปฏิบัติการ AIS	174	968
ห้องพักอาจารย์ tak	143	796
ห้องประชุม sjp	116	647
ห้องพักอาจารย์ spw	104	581
ห้องปฏิบัติการ studio	67	374
ห้องพักอาจารย์ scn	62	344
ห้องพักอาจารย์ wbk	45	252
ห้องปฏิบัติการ csbl	30	170
ห้องพักอาจารย์ ptp	10	55
ห้องพักอาจารย์ bea	2	14
ห้องพักอาจารย์ kar	0	1
รวม	30,210	168,570



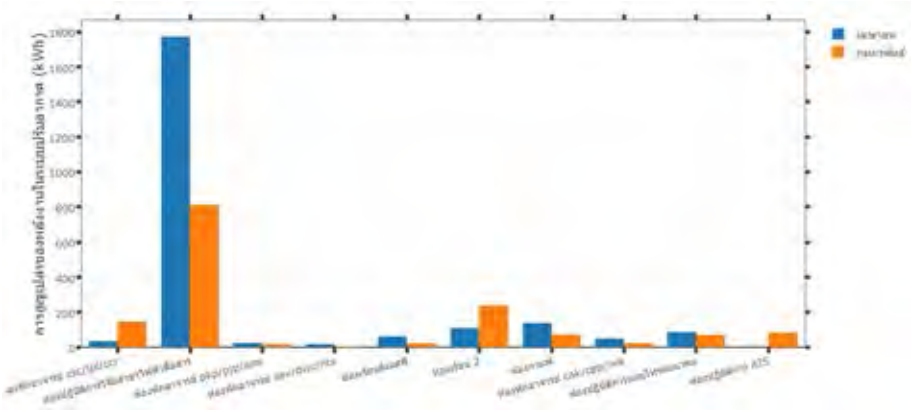
รูปที่ 4.31: ค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศบริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรมตั้งแต่วันที่ 1 มีนาคม พ.ศ. 2559 ถึง 28 กุมภาพันธ์ พ.ศ. 2560

4.4.2 อุณหภูมิภายนอกและการสูญเปล่าของพลังงานในระบบปรับอากาศ

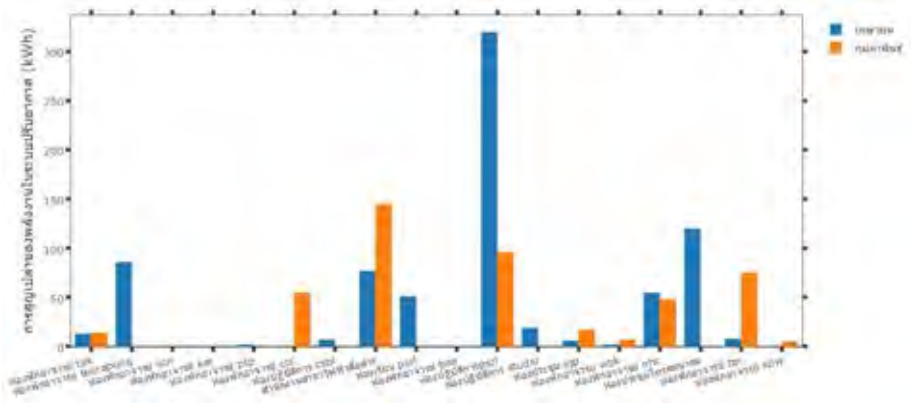
ในหัวข้อนี้จะทำการทดสอบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณชั้น 12 และ 13 อาคารเจริญวิศวกรรม เพื่อเปรียบเทียบการสูญเปล่าของพลังงานระหว่างเดือนกุมภาพันธ์ และเมษายน ซึ่งมีอุณหภูมิภายนอกเฉลี่ยทั้งปีต่ำและสูงสุดตามลำดับ โดยในรูปที่ 4.32 แสดงค่าเฉลี่ยอุณหภูมิในช่วงเวลาต่าง ๆ ของวันในเดือนกุมภาพันธ์ และเมษายน พ.ศ. 2559 และกำหนดค่าขีดสุดอุณหภูมิเท่ากับ 24 องศาเซลเซียส และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 15 นาที ผลลัพธ์ของการวิเคราะห์ข้างต้นจะถูกแสดงในรูปที่ 4.33 - 4.35 จะเห็นว่าผลรวม และค่าเฉลี่ยการสูญเปล่าของพลังงานในระบบปรับอากาศในเดือนเมษายน มีค่าสูงกว่าเมื่อเปรียบเทียบกับเดือนกุมภาพันธ์อย่างเห็นได้ชัด โดยเฉพาะบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารที่มีผลรวมการสูญเปล่าของพลังงานตลอดเดือนเมษายนสูงกว่าเดือนกุมภาพันธ์ถึง 964 kWh โดยในตารางที่ 4.12 แสดงตัวอย่างการเปรียบเทียบพลังงานที่สูญเปล่าในระบบปรับอากาศในรูปแบบค่าใช้จ่าย



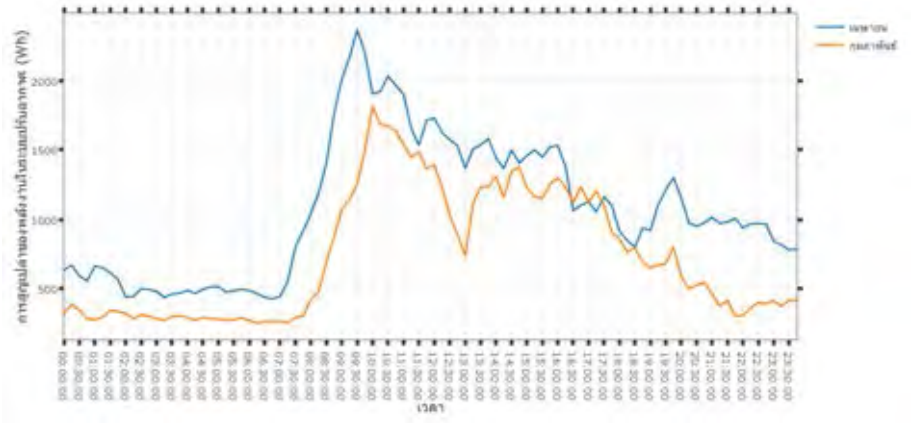
รูปที่ 4.32: ค่าเฉลี่ยอุณหภูมิในช่วงเวลาต่าง ๆ ของวันในเดือนกุมภาพันธ์ และเมษายน พ.ศ. 2559



รูปที่ 4.33: ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างเดือนเมษายน และกุมภาพันธ์ พ.ศ. 2559 ของเขตพื้นที่ต่าง ๆ บริเวณชั้น 13 อาคารเจริญวิศวกรรม



รูปที่ 4.34: ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างเดือนเมษายน และกุมภาพันธ์ พ.ศ. 2559 ของเขตพื้นที่ต่าง ๆ บริเวณชั้น 12 อาคารเจริญวิศวกรรม



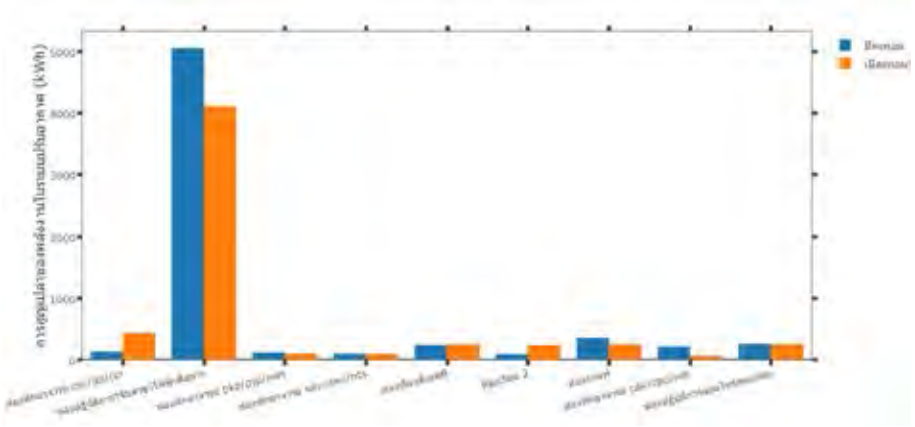
รูปที่ 4.35: ค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศระหว่างเดือนเมษายน และ กุมภาพันธ์ พ.ศ. 2559 บริเวณพื้นที่ภาควิชาวิศวกรรมไฟฟ้าในอาคารเจริญวิศวกรรม

ตารางที่ 4.12: พื้นที่ที่มีผลรวมการสูญเปล่าของพลังงาน และค่าใช้จ่ายจากพลังงานสูญเปล่าในระบบปรับอากาศสูงสุด 5 อันดับแรกในเดือน กุมภาพันธ์ และ เมษายน พ.ศ. 2559

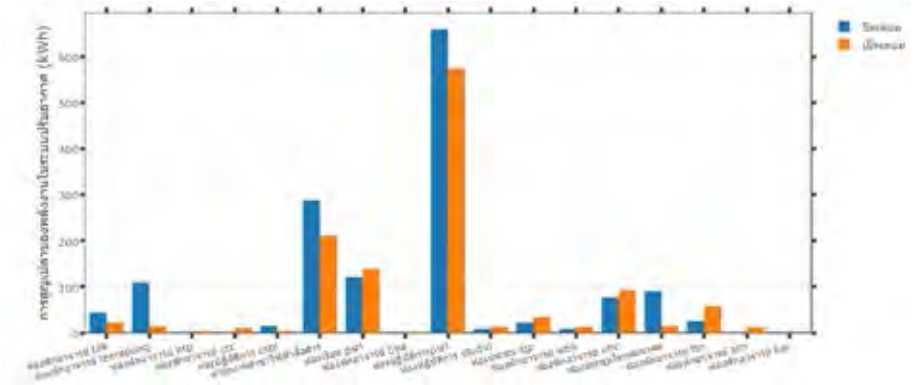
อาคารเจริญวิศวกรรม	ค่าใช้จ่ายจากพลังงานที่สูญเปล่าในเดือน กุมภาพันธ์ พ.ศ. 2559 (5.58 บาท ต่อ 1 kWh)	ค่าใช้จ่ายจากพลังงานที่สูญเปล่าในเดือน เมษายน พ.ศ. 2559 (5.58 บาท ต่อ 1 kWh)
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	4,531	9,921
ห้องปฏิบัติการ psrl	538	1,791
ห้องเรียน 2	1,337	624
สำนักงานสาขาไฟฟ้าสื่อสาร	811	433
ห้องกาแฟ	407	782

4.4.3 การเปิดปิดภาคการศึกษาและการสูญเปล่าของพลังงานในระบบปรับอากาศ

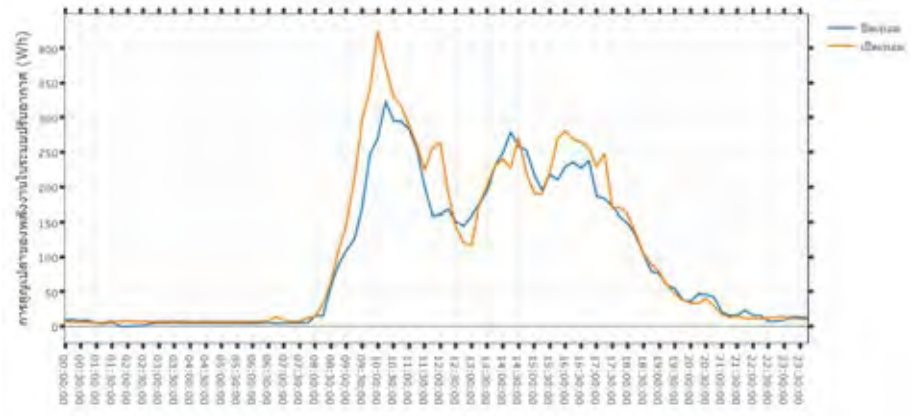
ในหัวข้อนี้จะทำการทดสอบฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณชั้น 12 และ 13 อาคารเจริญวิศวกรรม เพื่อเปรียบเทียบการสูญเปล่าของพลังงานระหว่างช่วงเวลาเปิดและปิดภาคการศึกษา โดยกำหนดช่วงเวลาตั้งแต่เดือน กันยายน ถึงพฤศจิกายน พ.ศ. 2559 เป็นช่วงเปิดภาคการศึกษา และกำหนดช่วงเวลาตั้งแต่เดือนพฤษภาคม ถึงกรกฎาคม พ.ศ. 2559 ให้เป็นช่วงปิดภาคการศึกษา รวมถึงกำหนดค่าขีดสุดอุณหภูมิ และค่าขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่เท่ากับ 24 องศาเซลเซียส และ 15 นาทีตามลำดับ ผลลัพธ์ของการวิเคราะห์ข้างต้นจะถูกแสดงในรูปที่ 4.36- 4.37 จะเห็นว่าผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศในช่วงปิดภาคการศึกษามีค่ามากกว่าในช่วงเปิดภาคการศึกษาอย่างเห็นได้ชัด ในขณะที่ค่าเฉลี่ยการสูญเปล่าของพลังงานในช่วงเวลาต่าง ๆ ของวัน จะมีลักษณะที่แตกต่างกันออกไปตามประเภทของห้อง ยกตัวอย่างเช่น เมื่อพิจารณาห้องพักอาจารย์ในช่วงเปิดภาคการศึกษา ซึ่งผู้ใช้ในพื้นที่บริเวณดังกล่าวจะมีการปฏิบัติหน้าที่ ส่งผลให้รวมการสูญเปล่าของพลังงานในระบบปรับอากาศตลอดช่วงเปิดภาคการศึกษามีค่ามากกว่าช่วงปิดภาคการศึกษาเท่ากับ 91 kWh ซึ่งสอดคล้องกับค่าเฉลี่ยการสูญเปล่าของพลังงานในช่วงเวลาต่าง ๆ ของวันในช่วงเปิดภาคการศึกษาที่มีค่าสูงกว่าช่วงปิดภาคการศึกษาดังแสดงในรูปที่ 4.38 ซึ่งอาจจะเกิดสาเหตุที่ผู้ใช้ในแต่ละบริเวณลืมปิดระบบปรับอากาศก่อนออกไปปฏิบัติหน้าที่ เป็นต้น และในตารางที่ 4.13 แสดงตัวอย่างการเปรียบเทียบพลังงานที่สูญเปล่าระบบปรับอากาศในรูปแบบค่าใช้จ่าย



รูปที่ 4.36: การเปรียบเทียบผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาของเขตพื้นที่ต่าง ๆ บริเวณชั้น 13 อาคารเจริญวิศวกรรม



รูปที่ 4.37: การเปรียบเทียบผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาของเขตพื้นที่ต่าง ๆ บริเวณชั้น 12 อาคารเจริญวิศวกรรม



รูปที่ 4.38: การเปรียบเทียบค่าเฉลี่ยการสูญเปล่าของพลังงานทุกช่วงเวลา 15 นาทีในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษาบริเวณห้องพักอาจารย์ชั้น 12 และ 13 อาคารเจริญวิศวกรรม

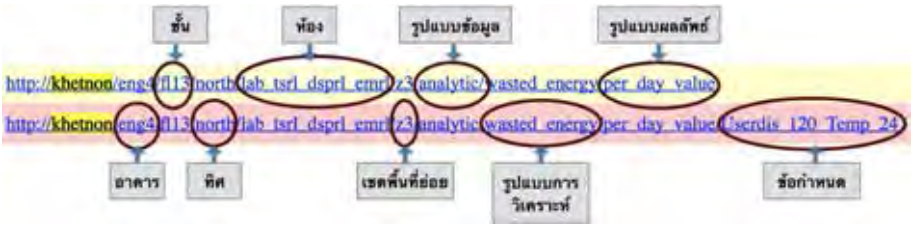
ตารางที่ 4.13: ตัวอย่างการเปรียบเทียบค่าใช้จ่ายของพลังงานที่สูญเปล่าในระบบปรับอากาศระหว่างช่วงเปิดและปิดภาคการศึกษา

อาคารเจริญวิศวกรรม	ค่าใช้จ่ายจาก พลังงานที่สูญเปล่า ช่วงเปิดภาคการศึกษา (5.58 บาท ต่อ 1 kWh)	ค่าใช้จ่ายจาก พลังงานที่สูญเปล่า ช่วงปิดภาคการศึกษา (5.58 บาท ต่อ 1 kWh)
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	22,967	28,253
ห้องปฏิบัติการ psrl	3,214	3,685
ห้องพักอาจารย์ csc/jpl/ccr	2,416	733
ห้องกาแฟ	1,392	1,966
สำนักงานสาขาไฟฟ้าสื่อสาร	1,180	1,611

4.5 การประยุกต์ใช้งานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

วิทยานิพนธ์ฉบับนี้กำหนดรูปแบบการทำงานของโปรแกรมประยุกต์ให้อยู่ในลักษณะการทำงานตามตารางเวลาสำหรับคำนวณการสูญเปล่าของพลังงานในระบบปรับอากาศรายวันของแต่ละเขตพื้นที่ย่อยในโครงการ CUBEMS โดยมีการปรับเปลี่ยนค่าขีดสุดประเภทต่าง ๆ ซึ่งลักษณะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศจะเริ่มคำนวณในระดับเขตพื้นที่ย่อยของห้องแต่ละห้องจนครบทั้งโครงการดังแสดงในรูปที่ 4.39 ผลลัพธ์ที่ได้จากกระบวนการวิเคราะห์จะถูกเขียนกลับเข้าไปที่หน่วยเก็บข้อมูลโครงการ CUBEMS ที่มีรูปแบบการระบุ PointID ที่คล้ายเดิมดังแสดงในรูปที่ 4.40 โดยรูปแบบผลลัพธ์ รวมถึงข้อกำหนดการปรับเปลี่ยนค่าขีดสุดของ PointID จะถูกอธิบายในตารางที่ 4.14 และ 4.15 ตามลำดับ

รูปที่ 4.39: ลักษณะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ



รูปที่ 4.40: รูปแบบการระบุ PointID ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

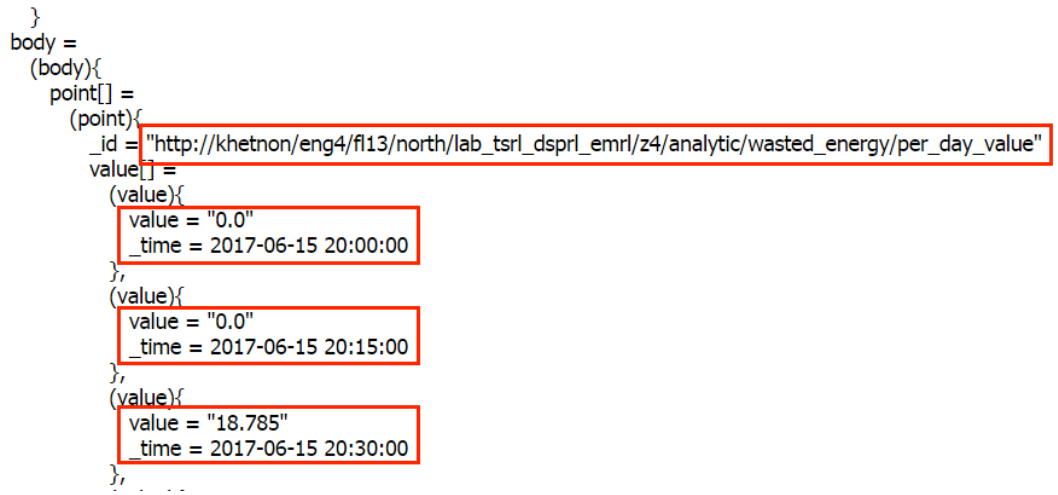
ตารางที่ 4.14: รูปแบบผลลัพธ์ของ PointID

รูปแบบผลลัพธ์ (Wh)	คำอธิบาย
per_day_value	ผลรวมการสูญเปล่าของพลังงานทุก 15 นาที
per_day	สัดส่วนการสูญเปล่าของพลังงานตลอดวัน
per_day_from_User_motion	ผลรวมการสูญเปล่าของพลังงานตลอดวัน จากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ
per_day_from_Area_temperature	ผลรวมการสูญเปล่าของพลังงานตลอดวัน จากการตั้งค่าอุณหภูมิเกินความจำเป็น

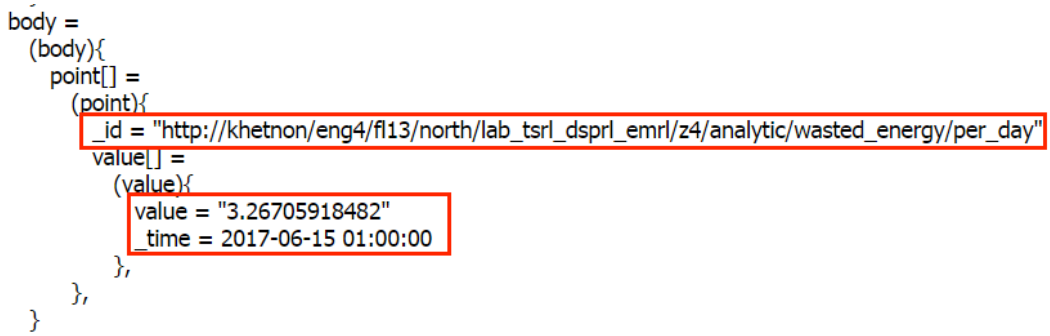
ตารางที่ 4.15: รูปแบบข้อกำหนดของ PointID

ข้อกำหนด	ค่าขีดสุดอุณหูมิ	ค่าขีดสุดที่ผู้ใช้หายไปจากพื้นที่
-	24	15
/Userdis_30_Temp_24	24	30
/Userdis_120_Temp_24	24	120
/Userdis_15_Temp_25	25	15
/Userdis_30_Temp_25	25	30
/Userdis_120_Temp_25	25	120

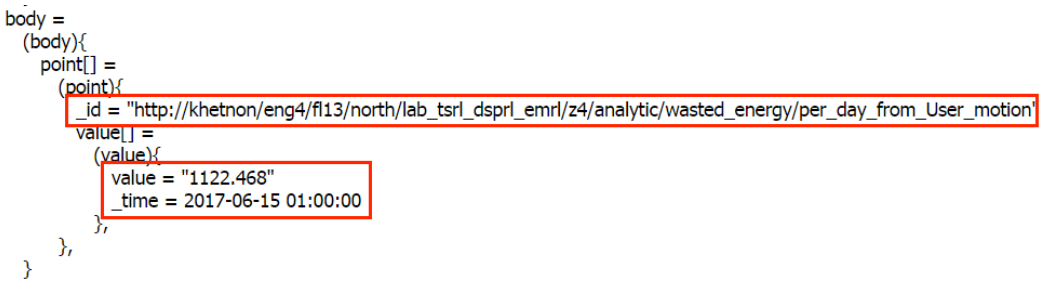
รูปที่ 4.41- 4.44 แสดงการตอบกลับผลลัพธ์ทั้ง 4 รูปแบบของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ



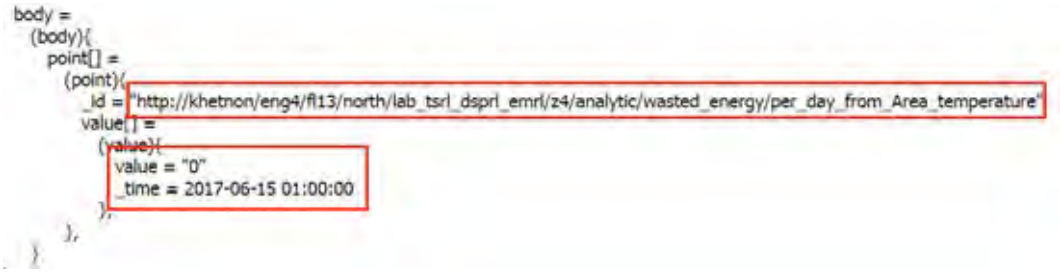
รูปที่ 4.41: การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานทุก 15 นาที



รูปที่ 4.42: การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในรูปแบบสัดส่วนการสูญเปล่าของพลังงานตลอดวัน

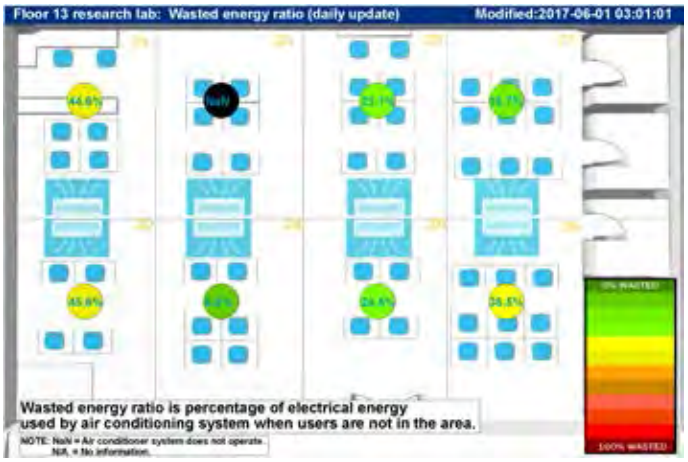


รูปที่ 4.43: การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานตลอดวันจากการไม่พบผู้ใช้อยู่ในบริเวณที่เปิดระบบปรับอากาศ

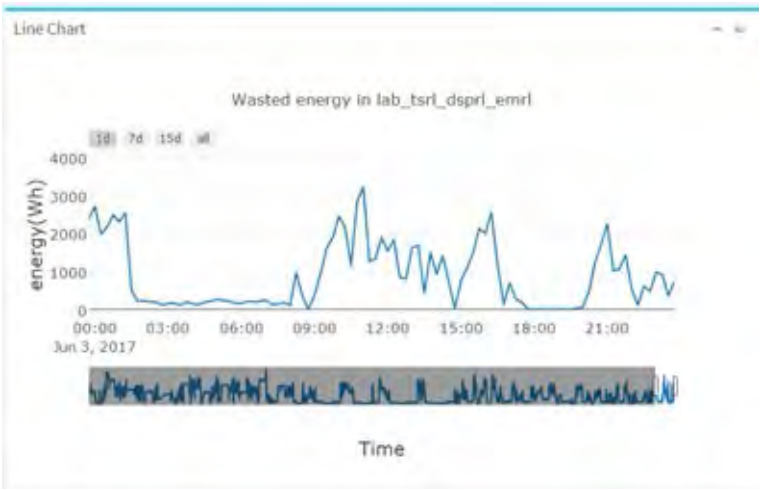


รูปที่ 4.44: การตอบกลับผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศในรูปแบบผลรวมการสูญเปล่าของพลังงานตลอดวันจากการตั้งค่าอุณหภูมิเกินความจำเป็น

การแสดงผลลัพธ์แต่ละรูปแบบของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศที่เหมาะสม จะแตกต่างกันออกไปขึ้นกับช่องทางการแสดงผล ยกตัวอย่างเช่น จอแสดงผลโครงการ CUBEMS ซึ่งถูกติดตั้งอยู่บริเวณโถงทางเดินหน้าลิฟท์ในแต่ละชั้นจะใช้ข้อมูลสัดส่วนการสูญเปล่าของพลังงานตลอดวันของพื้นที่แต่ละบริเวณมาแสดงผลเพื่อให้เกิดการเปรียบเทียบกับพื้นที่ข้างเคียงดังแสดงในรูปที่ 4.45 แตกต่างจากการแสดงผลบนเว็บไซต์การวิเคราะห์ข้อมูลที่เหมาะสมต่อแสดงผลลัพธ์ทุกรูปแบบ จนถึงข้อมูลการสูญเปล่าของพลังงานในรูปแบบเฉพาะเจาะจงดังแสดงในรูป 4.46 รวมถึงมีการออกให้โปรแกรมประยุกต์แจ้งเตือนการสูญเปล่าของพลังงานในระบบปรับอากาศในรูปแบบการสรุปเป็นช่วงเวลาดังแสดงในรูปที่ 4.47 และ 4.48



รูปที่ 4.45: การแสดงผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบนจอแสดงผลภายโครงการ CUBEMS



รูปที่ 4.46: การแสดงผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบนเว็บไซต์การวิเคราะห์ข้อมูล

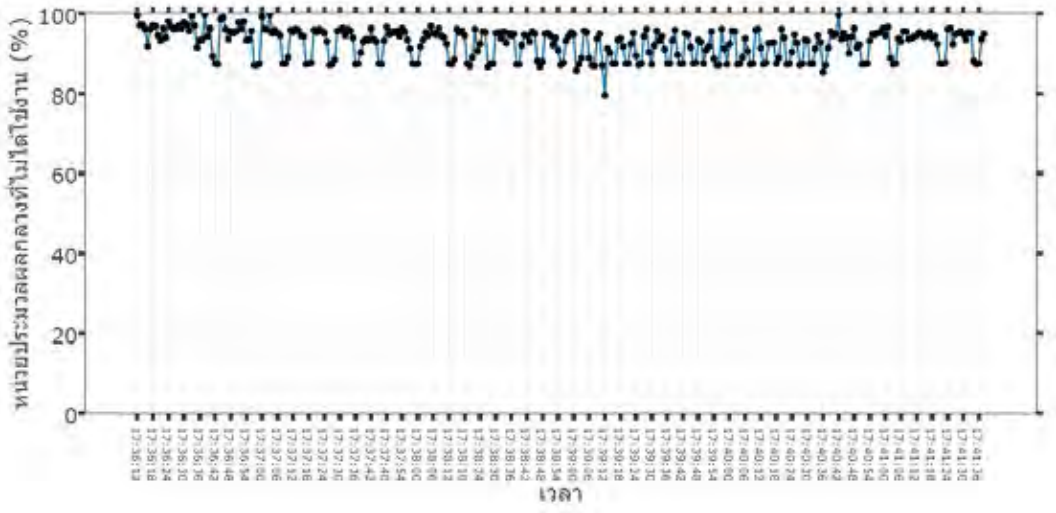
รูปที่ 4.47: การแจ้งเตือนผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศผ่านโปรแกรมประยุกต์ไลน์

รูปที่ 4.48: การแจ้งเตือนผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศผ่านโปรแกรมประยุกต์ไลน์ (เพิ่มเติม)

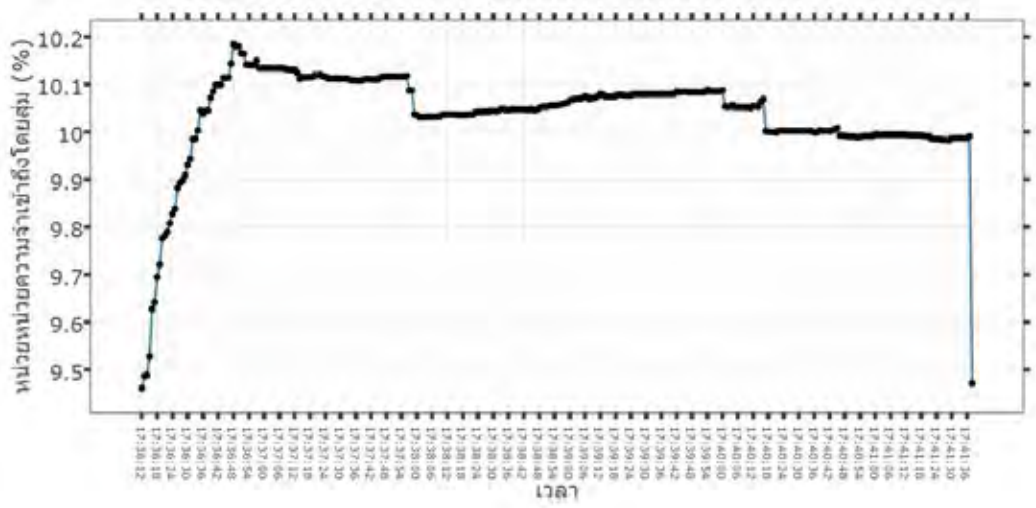
4.6 การประเมินสมรรถนะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

รูปที่ 4.49 และ 4.50 แสดงระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานและระดับการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของของเครื่องคอมพิวเตอร์ทดสอบที่มีคุณลักษณะดังแสดงในตารางที่ 3.1 ในขณะดำเนินงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศตามตารางเวลา โดยในหัวข้อที่ 4.5 อธิบายลักษณะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศว่าอยู่ในลักษณะ ไล่ลำดับการทำงานตั้งแต่ระดับเขตพื้นที่ย่อยของแต่ละห้องในโครงการ CUBEMS ซึ่งส่งผลโดยตรงต่อเวลาที่ใช้สำหรับการประมวลผล เพื่อแก้ไขปัญหาข้างต้นผู้วิจัยได้ปรับเปลี่ยนลักษณะการทำงานของฟังก์ชันวิเคราะห์ให้อยู่ในรูปแบบการดำเนินงานพร้อมกันของโปรแกรมประยุกต์หลายตัว สำหรับคำนวณแบบแยกเขตพื้นที่ดังแสดงในรูปที่ 4.51 ทั้งนี้จำนวนข้อมูลขาเข้าของกระบวนการวิเคราะห์จะแตกต่างกันออกไปขึ้นกับลักษณะพื้นที่ในแต่ละบริเวณ เพื่อให้เป็นมาตรฐานในการประเมินสมรรถนะการทำงานของฟังก์ชันการวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ ผู้วิจัยได้แบ่งรูปแบบของการประเมินสมรรถนะการทำงานตามลักษณะพื้นที่ทดสอบออกเป็นทั้งสิ้นสามรูปแบบประกอบไปด้วย

1. การประเมินสมรรถนะการทำงานของฟังก์ชันในรูปแบบการคำนวณรายห้อง
2. การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ
3. การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายชั้น



รูปที่ 4.49: ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบในขณะดำเนินงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ



รูปที่ 4.50: ระดับการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของของเครื่องคอมพิวเตอร์ทดสอบในขณะดำเนินงานฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

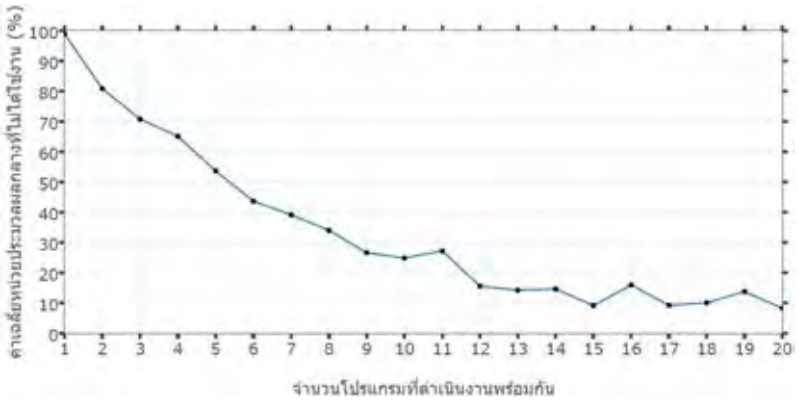
รูปที่ 4.51: รูปแบบการดำเนินพร้อมกันของโปรแกรมประยุกต์เพื่อคำนวณแบบแยกเขตพื้นที่

4.6.1 การประเมินสมรรถนะการทำงานของฟังก์ชันในรูปแบบการคำนวณรายห้อง

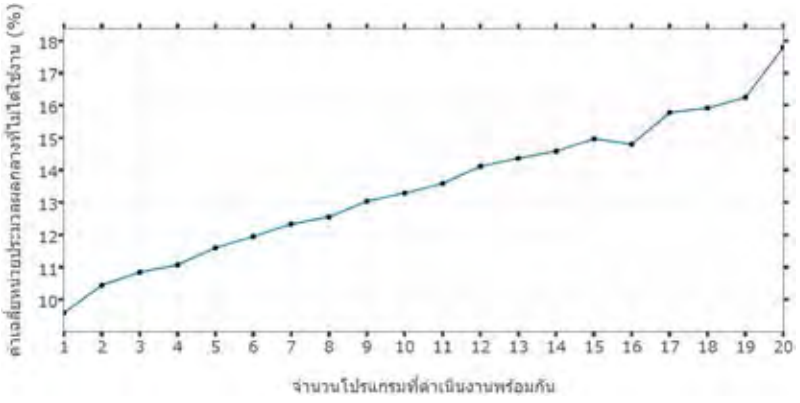
พื้นที่ทดสอบจะอยู่บริเวณห้องพักอาจารย์ cak/cpp/lwk ชั้น 13 อาคารเจริญวิศวกรรมในวันที่ 10 พฤษภาคม พ.ศ. 2560 โดยมีรายละเอียดจำนวนข้อมูลแต่ละประเภทแสดงในตารางที่ 4.16 การประเมินสมรรถนะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศจะอยู่ในรูปแบบการปรับเปลี่ยนจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์โดยเริ่มต้นจาก 1 โปรแกรมจนถึง 20 โปรแกรม ผลปรากฏว่าการเพิ่มจำนวนโปรแกรมประยุกต์จะส่งผลให้ระดับการทำงานของหน่วยประมวลผลกลาง ที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบลดต่ำลงจนกระทั่งจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์มากกว่า 15 โปรแกรมเป็นต้นไป ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบจะอิ่มตัวที่ระดับ 10 เปอร์เซ็นต์ดังแสดงในรูปที่ 4.52 ขณะที่การใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ทดสอบจะเพิ่มขึ้นจาก 8.8 เปอร์เซ็นต์ตอนจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 1 โปรแกรมไป 17.9 เปอร์เซ็นต์เมื่อจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 20 โปรแกรม ดังแสดงในรูปที่ 4.53

ตารางที่ 4.16: รายละเอียดและจำนวนข้อมูลของห้องพักอาจารย์ cak/cpp/lwk วันที่ 10 พฤษภาคม พ.ศ. 2560

ชนิดข้อมูล		จำนวนข้อมูล (ข้อมูล)
มาตรวัดอัจฉริยะ	ข้อมูลการบริโภคพลังงาน	1,438
ตัวรับรู้สภาพแวดล้อม 1	อุณหภูมิ	1,379
	การตรวจจับการเคลื่อนไหว	1,439
ตัวรับรู้สภาพแวดล้อม 2	อุณหภูมิ	1,370
	การตรวจจับการเคลื่อนไหว	1,523
ตัวรับรู้สภาพแวดล้อม 3	อุณหภูมิ	1,342
	การตรวจจับการเคลื่อนไหว	1,444
ผลรวม		**9,935**



รูปที่ 4.52: ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้อง



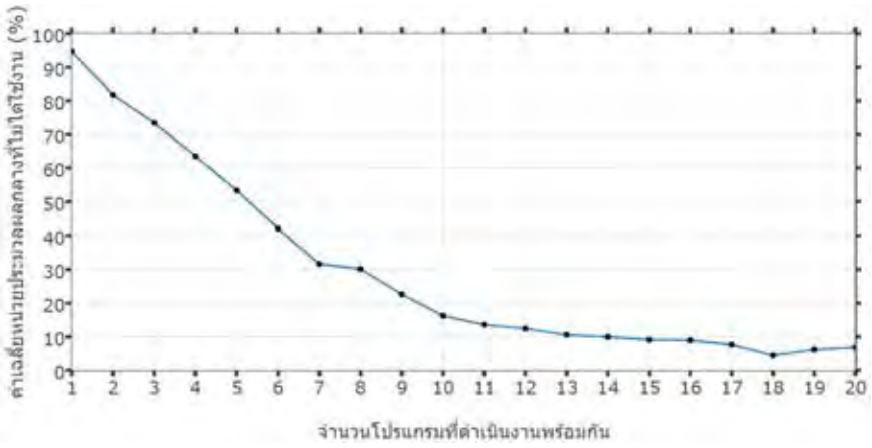
รูปที่ 4.53: ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้อง

4.6.2 การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ

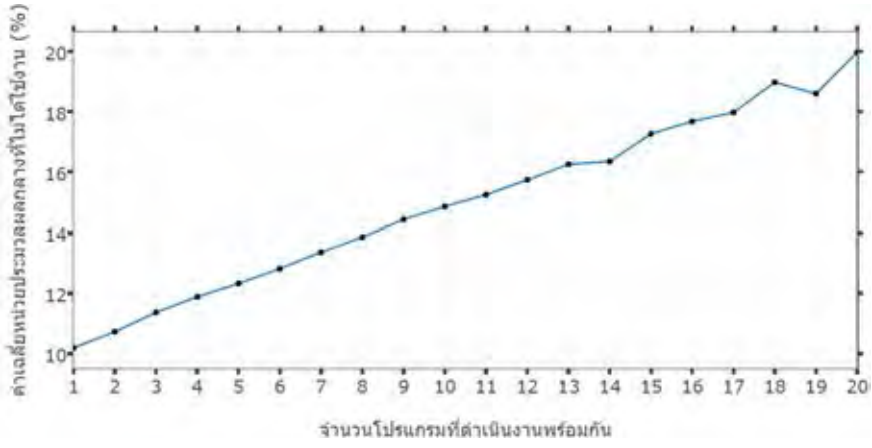
พื้นที่ทดสอบจะอยู่บริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร ชั้น 13 อาคารเจริญวิศวกรรมใน วันที่ 10 พฤษภาคม พ.ศ. 2560 โดยมีรายละเอียดจำนวนข้อมูลแต่ละประเภทแสดงในตารางที่ 4.17 ซึ่งการประเมินสมรรถนะการทำงานอยู่ในลักษณะเช่นเดียวกับหัวข้อที่ 4.6.1 ผลปรากฏว่าการเพิ่มจำนวนโปรแกรมประยุกต์จะส่งผลให้ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบลดต่ำลง จนกระทั้งจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์มากกว่า 13 โปรแกรม ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบจะอิ่มตัวที่ระดับ 10 เปอร์เซ็นต์ดังแสดงในรูปที่ 4.54 ขณะที่การใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ทดสอบจะเพิ่มขึ้นจาก 11 เปอร์เซ็นต์เมื่อจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 1 โปรแกรมไป 20 เปอร์เซ็นต์เมื่อจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 20 โปรแกรมดังแสดงในรูปที่ 4.55

ตารางที่ 4.17: รายละเอียดและจำนวนข้อมูลห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร วันที่ 10 พฤษภาคม พ.ศ. 2560

ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	จำนวนข้อมูล (ข้อมูล)		
	มาตรวัดอัจฉริยะ	ตัวรับรู้สภาพแวดล้อม	
	ข้อมูลการบริโภคพลังงาน	อุณหภูมิ	การตรวจจับการเคลื่อนไหว
เขตพื้นที่ย่อย 1	4,314	5,837	5,960
เขตพื้นที่ย่อย 2	1,438	4,383	4,495
เขตพื้นที่ย่อย 3	4,314	2,901	3,265
เขตพื้นที่ย่อย 4	4,314	2,925	3,188
เขตพื้นที่ย่อย 5	4,314	2,914	3,280
เขตพื้นที่ย่อย 6	4,314	2,916	3,457
เขตพื้นที่ย่อย 7	4,233	5,825	6,951
เขตพื้นที่ย่อย 8	4,233	5,836	7,126
ผลรวม	102,733		



รูปที่ 4.54: ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ



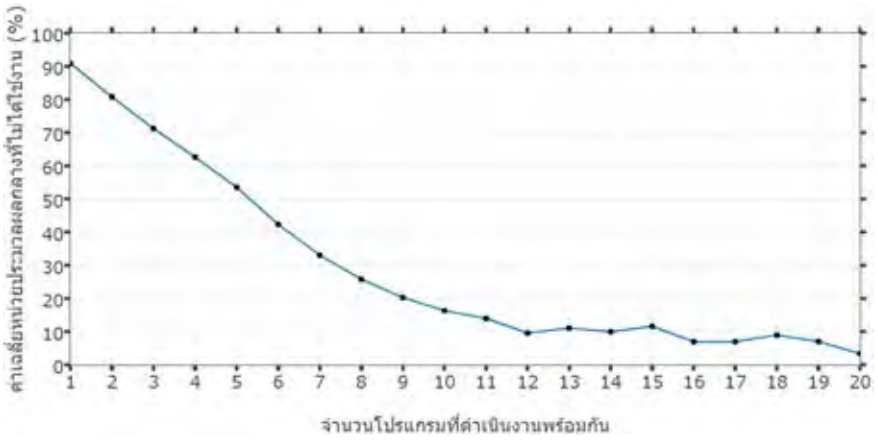
รูปที่ 4.55: ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายห้องปฏิบัติการ

4.6.3 การประเมินสมรรถนะการทำงานในรูปแบบการคำนวณรายชั้น

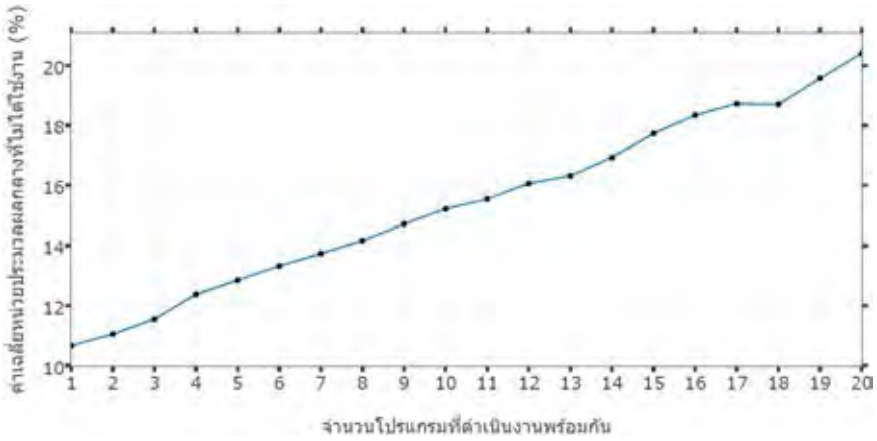
พื้นที่ทดสอบจะอยู่บริเวณ ชั้น 13 อาคารเจริญวิศวกรรมในวันที่ 10 พฤษภาคม พ.ศ. 2560 โดยมีรายละเอียดจำนวนข้อมูลแต่ละประเภทแสดงในตารางที่ 4.18 การประเมินสมรรถนะการทำงานของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ กระทำโดยการปรับเปลี่ยนจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์โดยเริ่มต้นจาก 1 โปรแกรมจนถึง 20 โปรแกรม ผลปรากฏว่าการเพิ่มจำนวนโปรแกรมประยุกต์จะส่งผลให้ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบจะลดต่ำลง จนกระทั้งจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์มากกว่า 12 โปรแกรม ระดับการทำงานของหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ทดสอบจะอิ่มตัวที่ระดับ 10 เปอร์เซ็นต์ดังแสดงในรูปที่ 4.56 ขณะที่การใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์จะเพิ่มขึ้นจาก 10.8 เปอร์เซ็นต์เมื่อจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 1 โปรแกรมไป 20.4 เปอร์เซ็นต์เมื่อจำนวนการดำเนินงานพร้อมกันของโปรแกรมประยุกต์เท่ากับ 20 โปรแกรมดังแสดงในรูปที่ 4.57

ตารางที่ 4.18: รายละเอียดและจำนวนข้อมูลชั้น 13 อาคารเจริญวิศวกรรม วันที่ 10 พฤษภาคม พ.ศ. 2560

ชั้น 13 อาคารเจริญวิศวกรรม	**ชนิดข้อมูล (ข้อมูล)**		
	มาตรวัดอัจฉริยะ	**ตัวรับรู้สภาพแวดล้อม**	
	ข้อมูลการบริโภคพลังงาน	**อุณหภูมิ**	**การตรวจจับการเคลื่อนไหว**
ห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสาร	31,474	33,537	37,722
ห้องเรียนดีเอสพี	8,616	5,489	5,489
ห้องพักอาจารย์ cak/cpp/lwk	1,438	4,091	4,406
ห้องเรียน 2	4,314	1,379	1,379
ห้องพักอาจารย์ csc/jpl/ccr	1,438	2,737	2,804
ห้องพักอาจารย์ sav/dwc/nts	5,752	4,092	4,159
ห้องพักอาจารย์ pkp/pjp/was	1,438	4,115	4,147
ห้องปฏิบัติการ AIS	1411	1,462	1,462
ห้องกาแฟ	1,411	1,458	1,458
ห้องปฏิบัติการย่อยโทรคมนาคม	1,411	1,458	1,460
ผลรวม	**188,841**		



รูปที่ 4.56: ค่าเฉลี่ยหน่วยประมวลผลกลางที่ไม่ได้ใช้งานของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายชั้น



รูปที่ 4.57: ค่าเฉลี่ยการใช้งานหน่วยความจำเข้าถึงโดยสุ่มของเครื่องคอมพิวเตอร์ขณะฟังก์ชันทำงานในรูปแบบการคำนวณรายชั้น

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาโปรแกรมวิเคราะห์ข้อมูลสำหรับระบบการจัดการพลังงานภายในอาคาร มีเป้าหมายที่จะเพิ่มขีดความสามารถในส่วนของการสร้างมุมมองใหม่ ๆ ในการวิเคราะห์ข้อมูลผ่านการเพิ่มการตระหนักรู้ในเรื่องพฤติกรรมการบริโภคพลังงาน และเป็นอีกหนึ่งทางเลือกสำหรับใช้เป็นเครื่องมือในการจัดการพลังงานของอาคารที่มีอยู่เดิมโดยไม่รบกวนในเรื่องความสบายของผู้ใช้ วิทยานิพนธ์ฉบับนี้อาศัยทรัพยากรข้อมูลของโครงการ CUBEMS เพื่อพัฒนาต้นแบบทดสอบ โดยเลือกใช้คลัง (โปรแกรม) Pandas ที่ถูกพัฒนาขึ้นบนพื้นฐานภาษา python ในการจัดการข้อมูล และออกแบบให้โปรแกรมประยุกต์มีการทำสำเนาข้อมูลเพื่อลดระยะเวลาในการร้องขอข้อมูลชุดเดิม การสื่อสารระหว่างโปรแกรมประยุกต์ที่พัฒนาขึ้นกับหน่วยเก็บข้อมูลของโครงการได้นำโพรโทคอล FETCH มาใช้ในการร้องขอประวัติข้อมูล รวมถึงประยุกต์ใช้ในการตรวจเช็คสถานะการทำงานของอุปกรณ์ตรวจวัดภายในโครงการ และใช้โพรโทคอล WRITE ในการเขียนผลลัพธ์กลับเข้าในหน่วยเก็บข้อมูลหลักของโครงการ CUBEMS เพื่อเพิ่มช่องทางการแสดงผล รวมถึงลดความซ้ำซ้อนในการวิเคราะห์ข้อมูลชุดเดิม นอกจากนี้วิทยานิพนธ์ฉบับนี้นำเสนอการออกแบบฟังก์ชันวิเคราะห์ข้อมูลเพื่อหาความสัมพันธ์ระหว่างข้อมูลจากตัวรับรู้การเคลื่อนไหว และตัว-รับรู้อุณหภูมิ ร่วมกับข้อมูลการบริโภคพลังงาน โดยข้อมูลที่ตรวจวัดจากตัวรับรู้การเคลื่อนไหวชนิด PIR จะถูกนำเข้ากระบวนการกรองข้อมูลเพื่อแก้ไขข้อผิดพลาดของการตรวจวัดข้อมูลกรณีต่าง ๆ ผลลัพธ์ของฟังก์ชันวิเคราะห์ข้างต้นจะออกมาในรูปแบบตัวชี้วัดความสูญเปล่าของพลังงานในระบบปรับอากาศเนื่องจากมีการเปิดให้ระบบทำงานในบริเวณที่ไม่มีผู้ใช้งานอยู่ หรือมีการปรับอุณหภูมิให้ต่ำเกินไป ซึ่งประกอบด้วย การสูญเปล่าของพลังงานในช่วงเวลาต่าง ๆ สัดส่วนการสูญเปล่าของพลังงานตลอดวัน การสูญเปล่าของพลังงานที่เกิดจากเคลื่อนไหวของผู้ใช้ และผลรวมของพลังงานที่เกิดจากการตั้งค่าอุณหภูมิเกินความจำเป็น ซึ่งจะถูกนำเสนอในช่องทางการแสดงผลที่ประกอบด้วยจอแสดงผลโครงการ CUBEMS เว็บไซต์การวิเคราะห์ข้อมูล และการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์

การทดสอบเพื่อวิเคราะห์การทำงานของโปรแกรมประยุกต์ และฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศที่พัฒนาขึ้น รวมถึงเพื่อศึกษาการส่งของตัวแปรภายนอกต่อการสูญเปล่าของพลังงาน พบว่า

1. ผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณห้องปฏิบัติการวิจัยสาขาไฟฟ้าสื่อสารตลอดเดือนกุมภาพันธ์ พ.ศ. 2560 สามารถแสดงถึงพฤติกรรมการบริโภคพลังงานของผู้ใช้ในแต่ละเขตพื้นที่ได้ชัดเจน ยกตัวอย่างเช่น เขตพื้นที่ย่อย 1 ที่มีการสูญเปล่าของพลังงานมากที่สุดเป็นจำนวนถึง 340 kWh หรือคิดเป็น 36% ของการบริโภคพลังงานในเขตพื้นที่ย่อยดังกล่าว ในขณะที่เขตพื้นที่ย่อย 4 ซึ่งมีปริมาณการบริโภคพลังงานใกล้เคียงกัน แต่มีการสูญเปล่าขอพลังงานเพียง 121.2 kWh หรือ 13.6% ของการบริโภคพลังงานในบริเวณดังกล่าว

2. ผลลัพธ์ของฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศที่มีการปรับเพิ่มค่า

ขีดสุดช่วงเวลาขั้นต่ำที่จะนับว่าผู้ใช้หายไปจากพื้นที่ สามารถแสดงถึงพฤติกรรมเชิงลึกของผู้ใช้ในแต่ละบริเวณได้ชัดเจนมากยิ่งขึ้น

3. ผลลัพธ์ของฟังก์ชันวิเคราะห์ที่อาศัยชุดข้อมูลตลอด 1 ปีพบว่าในบริเวณภาควิชาวิศวกรรมศาสตร์ไฟฟ้าในอาคารเจริญวิศวกรรม มีผลรวมการสูญเปล่าของพลังงานมากถึง 30,210 กิโลวัตต์ชั่วโมง หรือคิดเป็น 168,570 บาทสำหรับค่าใช่จ่ายจากพลังงานที่สูญเปล่า สื่อว่ายังมีโอกาสของการปรับลดการบริโภคพลังงานได้อีกมากเพียงการปรับวินัยของผู้ใช้อาคาร

4. การสูญเปล่าของพลังงานในระบบปรับอากาศของห้องเซิร์ฟเวอร์ จะเป็นข้อยกเว้นในการนำเสนอตัวชี้วัดความสูญเปล่าของพลังงานในระบบปรับอากาศเนื่องจากถูกออกแบบมาสำหรับติดตั้งเครื่องบริการ ซึ่งโดยปกติแล้วจะไม่มีผู้ใช้ในบริเวณดังกล่าว

5. ผลรวมการสูญเปล่าของพลังงานในระบบปรับอากาศบริเวณห้องพักอาจารย์ในช่วงเปิดภาคการศึกษาสูงกว่าในช่วงปิดภาคการศึกษาเท่ากับ 91 kWh ซึ่งตรงข้ามกับพื้นที่บริเวณอื่นที่มีการสูญเปล่าของพลังงานในช่วงปิดภาคการศึกษาสูงกว่าช่วงเปิดภาคการศึกษา

5.2 ข้อเสนอแนะ

การพัฒนาเพื่อเพิ่มประสิทธิภาพการทำงานของระบบให้ดีขึ้นในอนาคต ได้แก่

1. พัฒนารูปแบบการนำเสนอในเว็บไซต์การวิเคราะห์ข้อมูลให้มีความหลากหลาย และน่าสนใจมากยิ่งขึ้น

2. การเพิ่มเติมฟังก์ชันวิเคราะห์รูปแบบใหม่เข้าไปส่วนของการแสดงผล

3. ประยุกต์ใช้ข้อมูลการตรวจจับการเคลื่อนไหวที่ผ่านการกรองในการวิเคราะห์รูปแบบอื่น ๆ

4. พัฒนาฟังก์ชันในการตรวจสอบเหตุการณ์ รวมถึงแจ้งเตือน ในกรณีที่ผู้ใช้ในบางบริเวณหายจากพื้นที่เป็นระยะเวลาหนึ่ง และลืมปิดระบบที่มีการบริโภคพลังงานไฟฟ้า

บรรณานุกรม

[1] รายงานสถานการณ์การจำหน่ายไฟฟ้า การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย, ประจำปี 2558.

[2] Global energy statistical yearbook 2015 [Online]. Available: https://yearbook.enerdata.net/electricity-domestic-consumption-data-by-region.html [Accessed: June, 2016].

[3] รายงานสถานการณ์การจำหน่ายไฟฟ้า , การไฟฟ้าฝ่ายผลิต แห่งประเทศไทย กองอัตราค่าไฟฟ้าและพยากรณ์พลังไฟฟ้า ฝ่ายเศรษฐกิจพลังไฟฟ้า, ตารางที่ 21 เปรียบเทียบราคาขายปลีก ขายส่งเฉลี่ย, มกราคม 2557

[4] Hassan, R., and Radman, G. (2010, March). Survey on smart grid. In IEEE SoutheastCon 2010 (SoutheastCon), pp. 210-213, 2010.

[5] แผนแม่บทการพัฒนาโครงข่ายสมาร์ทกริดของประเทศไทย กระทรวงพลังงาน, พ.ศ. 2258-2579.

[6] United nations environment programme [Online]. Available: http://www.unep.org/sbci/AboutSBCI/Background.asp [Accessed: February, 2015].

[7] 2011 Buildings energy data boo. US Department of Energy, Washington, D. C., 2012.

[8] ระบบบริหารจัดการพลังงาน สำนักนโยบายและแผนพลังงาน กระทรวงพลังงาน [Online]. Available: http://www.thai-smartgrid.com/ระบบบริหารจัดการพลังงาน [Accessed: June, 2016].

[9] Energy management and performance management, Priva UK Ltd [Online]. Available: http://www.priva.co.uk/solutions-products/building-management/energy-management-and-performance-management/ [Accessed: June, 2016].

[10] Nguyen, T. A., and Aiello, M. Energy intelligent buildings based on user activity: A survey. Energy and Buildings 56 (2013): 244-257.

[11] Next-Generation building energy management systems, Navigant Research [Online]. Available: http://www.daikinapplied.com/download/Navigant_BEMS_White_Paper.pdf [Accessed: June, 2016].

[12] Yarbrough, I., Sun, Q., Reeves, D. C., Hackman, K., Bennett, R., and Henshel, D. S. Visualizing building energy demand for building peak energy analysis. Energy and Buildings 91 (2015): 10-15.

[13] Declan T. Delaney, Gregory M. P. O'Hare, Antonio G. Ruzzelli. Evaluation of energy-efficiency in lighting systems using sensor networks. Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pp. 61-66, 2009.

[14] รายงานฉบับสิ้นสุดโครงการ โครงงานวิจัย และพัฒนาเทคโนโลยีระบบโครงข่ายไฟฟ้าอัจฉริยะเพื่อบริหารจัดการการใช้พลังงานไฟฟ้าของอาคาร จุฬาลงกรณ์มหาวิทยาลัย (สถาบันวิจัยพลังงาน) เสนอกองทุนเพื่อส่งเสริมการอนุรักษ์พลังงานสำนักงานนโยบาย และแผนพลังงาน, กันยายน 2557.

[15] IEEE. Standard for ubiquitous green community control network. IEEE1888-2011, 2011.

[16] Esaki, H., and Ochiai, H. GUTP and IEEE1888 for smart facility system using internet architecture framework. In 1st IEEE Workshop on Holistic Building Intelligence through Sensing Systems, 2011.

[17] Le, D. H., and Pora, W. Development of smart meter for building energy management system based on the IEEE 1888 standard with Wi-Fi communication. In Proceedings of International Conference Electronics, Information and Communication (ICEIC 2014), Kota Kinabalu, Malaysia, January 2014.

[18] Inthasut, T., and Aswakul, C. ZigBee wireless sensor network with IEEE1888 gateway for building energy management system. In Proceedings of International Conference Electronics, Information and Communication (ICEIC 2014), Kota Kinabalu, Malaysia, January 2014.

[19] Khawsa-Ard, P., and Aswakul, C. IEEE1888 interactive display as a service (IDaaS): Example in building energy management system. IEEE 39th Computer Software and Applications Conference (COMPSAC), pp. 517-522, 2015.

[20] Van Rossum, G. Python programming language. In USENIX Annual Technical Conference 41 (June, 2007): 36.

[21] McKinney, W. Data structures for statistical computing In python. In Proceedings of the 9th Python in Science Conference 445 (June, 2010): 51-56.

[22] Plotly [Online]. Available: https://plot.ly/python/ [Accessed: May, 2017].

[23] Fry, B. Visualizing data. O'Reilly Media, 2008.

[24] Verdigris technologies, Inc [Online]. Available: https://verdigris.co/ [Accessed: May, 2017].

[25] Ramotion [Online]. Available: https://inspirationmobile.tumblr.com/post/102269248239/energy-predictions-ramotioncom [Accessed: May, 2017].

[26] Jupyter notebook [Online]. Available: http://jupyter.org/ [Accessed: May, 2017].

[27] suds. lightweight SOAP python client for consuming web services [Online]. Available: https://fedorahosted.org/suds/ [Accessed: May 2017].

ภาคผนวก

ก ชุดคำสั่งสำหรับจัดการ PointID

ก.1 ชุดคำสั่งสำหรับสร้าง PointID ในรูปแบบเฟรมข้อมูล

สำหรับแปลงข้อมูล PointID ให้อยู่ในรูปแบบเฟรมข้อมูลที่แบ่งลักษณะเฉพาะตามแถวนอน

```
1 def Create_AllPoint():
2     global All_Point_ID,coll
3     All_Point_ID = pd.read_csv("Data/Point_ID.csv")
4     All_Point_ID = All_Point_ID['0'].values
5     All_Point_ID = pd.Series(All_Point_ID)
6     All_Point_ID = All_Point_ID.str.rsplit('/', expand=True,n=8)
7     coll = ['http', 'place','fl','direction','room','zone','kind','
          duty','value']
8     All_Point_ID.columns = coll
```

ก.2 ชุดคำสั่งค้นหา PointID จากลักษณะเฉพาะ

สำหรับค้นหา PointID จากลักษณะเฉพาะ ที่มีตัวแปรขาเข้าประกอบไปด้วย อาคาร (place) ชั้น (fl) ทิศ (direction) ห้อง (room) เขตพื้นที่ย่อย (zone) ชนิดของอุปกรณ์ (kind) ชนิดของข้อมูล (value)

```
1 def Find_PointID (place="",fl="",direction="",room="",zone="",kind="
    ",value=""):
2     Point,Pointid = [],{}
3     if place == 'esid':
4         Pointid["Point_Name"] = ['http://bems.ee.eng.chula.ac.th/
              esid/monitor/energy']
5         Pointid["IND"] = ['esid']
6     elif place == 'highvoltage':
7         Pointid["Point_Name"] = ['http://bems.ee.eng.chula.ac.th/
              highvoltage/monitor/energy']
8         Pointid["IND"] = ['gewertz']
9     elif place == 'gewertz':
10        Pointid["Point_Name"] = ['http://bems.ee.eng.chula.ac.th/
              gewertz/monitor/energy']
11        Pointid["IND"] = ['gewertz']
12    else:
13        Drop_Number= []
14        for x in All_Point_ID['kind'].unique().tolist():
15            Drop_Number.append(''.join([i for i in x if not i.
                  isdigit()]))
16        Drop_Number_drop_duplicate = list(OrderedDict.fromkeys(
              Drop_Number))
```

```
17          Table = pd.DataFrame(data=All_Point_ID['kind'].unique().
                tolist(),columns=['All'])
18          Table['Drop_Number'] = Drop_Number
19          result = All_Point_ID
20          if place =='':pass
21          elif place in list(result.place.unique()):
22              result = result.loc[(result.place==place)]
23          else:
24              print 'Please define variable *place only in %s.'%str(
                    list(result.place.unique()) +['esid','highvoltage','
                    gewertz'])
25              result = result.loc[(result.place==place)]
26          if fl == "":pass
27          elif fl in list(result.fl.unique()):
28              result = result.loc[(result.fl==fl)]
29          else:
30              print 'Please define variable *fl only in %s.'%str(list(
                    result.fl.unique()))
31              result = result.loc[(result.fl==fl)]
32          if direction == "":pass
33          elif direction in list(result.direction.unique()):
34              result = result.loc[(result.direction==direction)]
35          else:
36              print 'Please define variable *direction only in %s.'%
                    str(list(result.direction.unique()))
37              result = result.loc[(result.direction==direction)]
38          if room == "":pass
39          elif room in list(result.room.unique()):
40              result = result.loc[(result.room==room)]
41          else:
42              print 'Please define variable *room only in %s.'%str(
                    list(result.room.unique()))
43              result = result.loc[(result.room==room)]
44          if zone == "":pass
45          elif zone in list(result.zone.unique()):
46              result = result.loc[(result.zone==zone)]
47          else:
48              print 'Please define variable *zone only in %s.'%str(
                    list(result.zone.unique()))
49              result = result.loc[(result.zone==zone)]
50          if kind == "":pass
51          elif kind=='Air':
52              air = list(Table['All'][(Table.Drop_Number == 'aircon')
                    | (Table.Drop_Number == 'aircon_ph') ])
53              result = result.loc[result.kind.isin(air)]
54          elif kind=='Light':
```

```
55              light = list(Table['All'][(Table.Drop_Number == 'light')
                    ])
56              result = result.loc[result.kind.isin(light)]
57          elif kind=='Outlet':
58              outlet = list(Table['All'][(Table.Drop_Number == 'outlet
                    ')])
59              result = result.loc[result.kind.isin(outlet)]
60          elif kind=='Sensor':
61              sensor = list(Table['All'][(Table.Drop_Number == 'sensor
                    ')])
62              result = result.loc[result.kind.isin(sensor)]
63          else:
64              print 'Please define variable *kind only in [Air,Light,
                    Outlet,Sensor].'
65              result = result.loc[(result.kind==kind)]
66          if value == "":pass
67          elif value == 'energy':
68              energy = ['energy_r', 'energy_s', 'energy_t', 'energy']
69              result = result.loc[result.value.isin(energy)]
70          elif value == 'pir':
71              pir = ['PIR', 'pir',]
72              result = result.loc[result.value.isin(pir)]
73          elif value in [ 'humidity','illuminance','temperature','
                battery']:
74              result = result.loc[(result.value==value)]
75          else:
76              print 'Please define variable *value only in [energy,pir
                    ,humidity,illuminance,temperature,battery].'
77              result = result.loc[(result.value==value)]
78          Pointname = result[coll].apply(lambda x: '/'.join(x), axis
                =1)
79          [Point.append(Pointname[result.index[x]]) for x in range(len
                (result.index))]
80          ind = Pointname.index.tolist()
81          Pointid["Point_Name"] = Point
82          Pointid["IND"] = ind
83      return Pointid    #type dict
```

ก.3 ชุดคำสั่งสำหรับค้นหาชื่อห้อง

สำหรับค้นหาชื่อห้องในแต่ละชั้นที่มีตัวแปรขาเข้าประกอบไปด้วย ชั้น(fl)

```
1  def Check_Room_name(fl):
2      Have_Air = set(All_Point_ID.loc[(All_Point_ID['fl']==fl) & ((
           All_Point_ID['kind']=='aircon1') | (All_Point_ID['kind']=='
           aircon2')|(All_Point_ID['kind']=='aircon_3ph1')
```

```
3       | (All_Point_ID['kind']=='aircon_3ph2')|(All_Point_ID['kind']=='
            Outlet')|(All_Point_ID['kind']=='Light') )]['room'].unique().
            tolist())
4       return list(Have_Air)
```

ก.4 ชุดคำสั่งสำหรับค้นหาชื่อห้องที่ถูกแบ่งออกเป็นกลุ่ม

สำหรับค้นหาชื่อห้องที่มีตัวแปรขาเข้าประกอบไปด้วย Scale_building ('Eng4','Floor13','Floor12', 'Instructors Room_All', 'Instructors Room_13', 'Instructors Room_12')

```
1  def Defind_room(Scale_building):
2      if Scale_building in ['Eng4','Floor13','Floor12', 'Instructors
           Room_All', 'Instructors Room_13', 'Instructors Room_12']:
3          if Scale_building=='Eng4':
4              rooms = (Check_Room_name('fl12') +Check_Room_name('fl13'
                   ))
5          elif Scale_building=='Floor13' :
6              rooms = Check_Room_name('fl13')
7          elif Scale_building=='Floor12' :
8              rooms= Check_Room_name('fl12')
9          elif Scale_building == 'Instructors Room_All':
10             rooms=[]
11             for x in Check_Room_name('fl12') + Check_Room_name('fl13
                   '):
12                 if 'room' in x.split("_"):
13                     if 'server' in  x.split("_") or 'pantry' in  x.
                           split("_"):pass
14                     else:
15                         rooms.append(x)
16         elif Scale_building == 'Instructors Room_13':
17             rooms=[]
18             for x in Check_Room_name('fl13'):
19                 if 'room' in x.split("_"):
20                     if 'server' in  x.split("_") or 'pantry' in  x.
                           split("_"):pass
21                     else:
22                         rooms.append(x)
23         elif Scale_building == 'Instructors Room_12':
24             rooms=[]
25             for x in Check_Room_name('fl12'):
26                 if 'room' in x.split("_"):
27                     if 'server' in  x.split("_") or 'pantry' in  x.
                           split("_"):pass
28                     else:
29                         rooms.append(x)
```

```
30          return rooms
31      else:
32          print 'Wrong Comnand please defind only Eng4,Floor13,Floor12
                ,Instructors Room_All or 12,13'
```

ข ชุดคำสั่งสำหรับติดต่อกับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

ข.1 ชุดคำสั่งเตรียมคลัง (โปรแกรม) ที่จำเป็นสำหรับติดต่อกับหน่วยเก็บข้อมูลมาตรฐาน IEEE1888

สำหรับประกาศเรียกใช้คลัง(โปรแกรม) ที่จำเป็น และประกาศที่อยู่ WSDL ของหน่วยเก็บข้อมูลโครงการ CUBEMS

```
1  import pandas as pd
2  import suds
3  from suds.client import Client
4  from suds.plugin import MessagePlugin
5  from suds.sax.date import UTC
6  CLASS BugfixMessagePlugin(MessagePlugin):
7      def marshalled(self, context):
8          tp = context.envelope.childAtPath('Body/queryRQ/transport')
9          tp.setPrefix(tp.findPrefix('http://gutp.jp/fiap/2009/11/'))
10
11 myplugins = (
12     BugfixMessagePlugin(),)
13 WSDL_URL = 'http://161.200.90.122/axis2/services/FIAPStorage?wsdl'
14 client = Client(WSDL_URL, cache=None, plugins=myplugins)
```

ข.2 ชุดคำสั่งสำหรับสร้าง XML เพื่อร้องขอข้อมูล

```
1  def Create_TP_FETCH(Point_ID_Name,qt,lt):
2      key = client.factory.create('{http://gutp.jp/fiap/2009/11/}key')
3      key._id = Point_ID_Name
4      key._attrName = "time"
5      key._lteq = lt
6      key._gteq = qt
7      query = client.factory.create('{http://gutp.jp/fiap/2009/11/}
           query')
8      query._id = uuid.uuid4()
9      query._type = "storage"
10     query._acceptableSize = "5000"
11     query.key = key
12     header = client.factory.create('{http://gutp.jp/fiap/2009/11/}
           header')
13     header.error = None
14     header.query = query
15     tp = client.factory.create('{http://gutp.jp/fiap/2009/11/}
           transport')
16     tp.header = header
17     return tp
```

ข.3 ชุดคำสั่งสำหรับสร้าง XML เพื่อร้องขอข้อมูลแบบเฉพาะ

```
def Create_TP_FETCH_Check_Max_Min_time(Point_ID_Name,typedd):
    key = client.factory.create('{http://gutp.jp/fiap/2009/11/}key')
    key._id = Point_ID_Name
    key._attrName = "time"
    key._select = typedd
    query = client.factory.create('{http://gutp.jp/fiap/2009/11/}
        query')
    query._id = uuid.uuid4()
    query._type = "storage"
    query._acceptableSize = "5000"
    query.key = key
    header = client.factory.create('{http://gutp.jp/fiap/2009/11/}
        header')
    header.error = None
    header.query = query
    tp = client.factory.create('{http://gutp.jp/fiap/2009/11/}
        transport')
    tp.header = header
    return tp
```

ข.4 ชุดคำสั่งสำหรับสร้าง XML เพื่อเขียนข้อมูล

```
def Create_TP_WRITE(Point_ID_Name,value,time):
    val = client.factory.create('{http://gutp.jp/fiap/2009/11/}value
        ')
    val.value = value
    val._time = time
    point = client.factory.create('{http://gutp.jp/fiap/2009/11/}
        point')
    point._id = Point_ID_Name
    point.value.append(val)
    body = client.factory.create('{http://gutp.jp/fiap/2009/11/}body
        ')
    body.point.append(point)
    tp = client.factory.create('{http://gutp.jp/fiap/2009/11/}
        transport')
    tp.body = body
    return tp
```

ข.5 ชุดคำสั่งสำหรับส่งคำสั่งร้องขอข้อมูล

```
def Fetch_His(Point_ID_Name,lt,qt,typedd):
    Point_Value,Point_Time,data = Set_Value()
    tp = Create_TP_FETCH(Point_ID_Name,qt,lt)
    print '******-----FETCH Data from IEEE1888 Storage----******'
```

```
    try:
        result = client.service.query(tp)
    except Exception as e:
        print 'Can not connect to IEEE1888 Storage'
        print e.message
    try:
        for j in range(len(result.body.point[0].value)):
            if '_time' and 'value' in result.body.point[0].value[j]:
                if typedd !='PIR':
                    try:
                        Point_Value.append(float(result.body.point
                            [0].value[j].value))
                        Point_Time.append(result.body.point[0].value
                            [j]._time.strftime("%Y-%m-%d %H:%M:%S"))
                    except:pass
                else:
                    Point_Value.append(result.body.point[0].value[j
                        ].value)
                    Point_Time.append(result.body.point[0].value[j].
                        _time.strftime("%Y-%m-%d %H:%M:%S"))
            else:
                print 'No data respond from IEEEE1888 Storage'
                pass
    except:
        pass
    try:
        while result.header.query._cursor != "" : #check cursor
            tp.header.query._cursor = "%s"%result.header.query.
                _cursor
            try:
                result = client.service.query(tp)
            except Exception as e:
                print 'Can not connect to IEEE1888 Storage'
                print e.message
            for j in range(len(result.body.point[0].value)):
                    if '_time' and 'value' in result.body.point[0].
                        value[j]:
                        if typedd !='PIR':
                            try:
                                Point_Value.append(float(result.body
                                    .point[0].value[j].value))
                                Point_Time.append(result.body.point
                                    [0].value[j]._time.strftime("%Y-%
                                    m-%d %H:%M:%S"))
                            except:pass
                        else:
```

```
42                              Point_Value.append(result.body.point[0].
                                    value[j].value)
43                              Point_Time.append(result.body.point[0].
                                    value[j]._time.strftime("%Y-%m-%d %H
                                    :%M:%S"))
44                      else:
45                          pass
46      except:
47          pass
48      data = {'VALUE':Point_Value,'TIME':Point_Time}
49      Table = pd.DataFrame(data)
50      return Table
```

ข.6 ชุดคำสั่งสำหรับส่งคำสั่งเขียนข้อมูล

```
1  def Write_IEEE1888(Point_ID_Name,Write_value,Write_time):
2      class BugfixMessagePlugin(MessagePlugin):
3          def marshalled(self, context):
4              tp = context.envelope.childAtPath('Body/dataRQ/transport
                   ')
5              tp.setPrefix(tp.findPrefix('http://gutp.jp/fiap/2009/11/
                   '))
6      myplugins = (
7          BugfixMessagePlugin(),)
8      client = Client(WSDL_URL, cache=None, plugins=myplugins)
9      tp = Create_TP_WRITE(Point_ID_Name,Write_value,Write_time)
10     try:
11         result = client.service.data(tp)
12         print 'Write Success'
13     except Exception as e:
14         print e.message
```

ข.7 ชุดคำสั่งสำหรับตรวจเช็คค่าเวลาแรกสุดหรือล่าสุด

```
1  def Check_Max_Min_time(Point_ID_Name,typedd):
2      if typedd in ['maximum','minimum']:
3          tp = Create_TP_FETCH_Check_Max_Min_time(Point_ID_Name,typedd
               )
4          print '******-----FETCH Data from IEEE1888 Stirage
               -----******'
5          try:
6              result = client.service.query(tp)
7          except Exception as e:
8              print e.message
9          Time = result.body.point[0].value[0]._time.strftime("%Y-%m-%
               d %H:%M:%S")
10         Value = result.body.point[0].value[0].value
```

```
11          return Time,Value
12      else:
13          print 'Wrong input please defind Max or Min ....'
```

ค ชุดคำสั่งการเตรียมข้อมูลของโปรแกรมประยุกต์

ค.1 ชุดคำสั่งสำหรับตรวจเช็คเวลา

```
def Check_time():
    time_now = datetime.now()
    return time_now
```

ค.2 ชุดคำสั่งสำหรับจัดการเฟรมข้อมูลให้แถวลำดับอยู่ในรูปแบบวันที่และเวลาใด ๆ

```
def Set_Table(Table):
    Table = Table.set_index("TIME")
    Table.index = pd.to_datetime(Table.index)
    Table.index.names = ['TIME']
    Table = Table.sort_index(axis=0)
    return Table
```

ค.3 ชุดคำสั่งสำหรับตรวจเช็คการอยู่ในช่วงเพื่อระบุรูปแบบการร้องขอ

```
def Check_time_in_range(start, end, x):
    if start <= end:
        return start <= x <= end
    else:
        return False
```

ค.4 ชุดคำสั่งสำหรับตรวจเช็คการมีอยู่ของไฟล์ PointID

```
def Check_File(qt,lt,idx):
    try:
        df = pd.read_csv('Data/%s.csv'%idx)
        df = df.dropna()
        df = df.set_index('TIME')
        df.index = pd.to_datetime(df.index)
        df.index.names = ['TIME']
        qt1,lt1,qt2,lt2,case,how = Check_case(df,qt,lt)
        if case == 'Case1':
            return 'Have Point ID Data',qt1,lt1,qt2,lt2,df,case,how
        else:
            return 'NO Point ID data',qt1,lt1,qt2,lt2,df,case,how
    except :
        return 'No Point ID file',qt,lt,"","","",'Case0','No Point
           ID file'
```

ค.5 ชุดคำสั่งสำหรับตรวจเหตุการณ์ในการเตรียมข้อมูลของส่วนต่อประสาน

```
def Check_case (df,qt,lt):
    result1 = Check_time_in_range(df.index[0],df.index[-1],qt)
```

```
3       result2 = Check_time_in_range(df.index[0],df.index[-1],lt)
4       if result1 == True and result2 == True:
5           return qt,lt,"","','Case1','Retrieve all data from system
                storage'
6       elif result1 == False and result2 == True :
7           lt = df.index[0] - timedelta(minutes = 1)
8           return qt,lt,"","','Case2','Fetch data form IEEE1888 storage
                during upper system period and retrieve some data from
                system storage'
9       elif result1 == True and result2 == False :
10          qt = df.index[-1] + timedelta(minutes = 1)
11          return qt,lt,"","','Case3','Fetch data form IEEE1888 storage
                during under system period and retrieve some data from
                system storage'
12      elif result1 == False and result2 == False and lt  > df.index
            [-1] and qt  > df.index[-1] :
13          qt = df.index[-1] + timedelta(minutes = 1)
14          return qt,lt,"","','Case4','Fetch all data form IEEE1888
                storage during under system period'
15      elif result1 == False and result2 == False and lt <df.index[0]
            and qt <df.index[0]:
16          lt = df.index[0] - timedelta(minutes = 1)
17          return qt,lt,"","','Case5','Fetch all data form IEEE1888
                storage during upper system period'
18      elif result1 == False and result2 == False :
19          qt1 = qt
20          lt1 = df.index[0] - timedelta(minutes = 1)
21          qt2 = df.index[-1] + timedelta(minutes = 1)
22          lt2 = lt
23          return qt1,lt1,qt2,lt2,'Case6','Fetch all data form IEEE1888
                storage during under and upper system period'
```

ค.6 ชุดคำสั่งสำหรับเตรียมข้อมูลให้อยู่ในรูปแบบเฟรมข้อมูล

```
1  def Call_Dataframe (Point_ID_Name,idx,qt,lt,typedd,Check_Case='Yes')
       :
2       Start_Time = datetime.now()
3       qt = datetime.strptime(qt,"%Y-%m-%d %H:%M:%S")
4       lt = datetime.strptime(lt,"%Y-%m-%d %H:%M:%S")
5       Count_Fetch,Count_Storage = 0,0
6       if Check_Case == 'Yes':
7           result,qt1,lt1,qt2,lt2,df,case,how = Check_File(qt,lt,idx)
8       elif Check_Case == 'No':
9           result,qt1,lt1,qt2,lt2,df,case,how = 'Only Fetch from
                IEEE1888 Storage',qt,lt,'','','','Case-1','Not Checked'
10      print result
```

```
    if result == 'Have Point ID Data':
        Table = df.loc[qt:lt]
        Count_Storage = len(Table)
    elif result == 'Only Fetch from IEEE1888 Storage':
        Table = Fetch_His(Point_ID_Name,lt1,qt1,typedd)
        Table = Set_Table(Table)
        Count_Fetch = len(Table)
    elif result == 'No Point ID file':
        Table = Fetch_His(Point_ID_Name,lt1,qt1,typedd)
        Table = Set_Table(Table)
        Table.to_csv("Data/%s.csv"%idx)
        Count_Fetch = len(Table)
    elif qt2 != "" and lt2 != "":
        Table = Fetch_His(Point_ID_Name,lt1,qt1,typedd)
        Table2 = Fetch_His(Point_ID_Name,lt2,qt2,typedd)
        Table = Table.append(Table2)
        Table = Set_Table(Table)
        Count_Fetch = len(Table)
        df = df.append(Table)
        df = df.sort_index(axis=0)
        df.to_csv("Data/%s.csv"%idx)
        Table = df.loc[qt:lt]
        Count_Storage = len(Table) - Count_Fetch
    else :
        Table = Fetch_His(Point_ID_Name,lt1,qt1,typedd)
        Table = Set_Table(Table)
        Count_Fetch = len(Table)
        df = df.append(Table)
        df = df.sort_index(axis=0)
        df.to_csv("Data/%s.csv"%idx)
        Table = df.loc[qt:lt]
        Count_Storage = len(Table)- Count_Fetch
    if typedd != 'PIR': Table = Table.apply(pd.to_numeric,args=('
       coerce',))
    End_Time = datetime.now() - Start_Time
    return Table,Count_Fetch,Count_Storage,End_Time.total_seconds()
```

ค.7 ชุดคำสั่งสำหรับเตรียมข้อมูลประเภทต่าง ๆ

```
def Prepare_Data (Data_type,Point_ID_Name,Point_ID_Idx,qt,lt,
   Check_Cased='Yes'):
    Count_Fetch,Count_Storage,End_Time,Tablek_ = [],[],[],{}
    if Data_type =='AIR':
        if len(Point_ID_Name) == 1 :
            print '--------------------------FETCH_EN_AIR1ph
               ---------------------------'
```

```
6               Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
7               Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'AIR'
                    ,Check_Case=Check_Cased)
8           elif len(Point_ID_Name) != 1 :
9               Table3_ = {}
10              print '--------------------------FETCH_EN_AIR3ph
                    --------------------------'
11              Table3_[0],Count_FetchR,Count_StorageR,End_TimeR =
                    Call_Dataframe(Point_ID_Name[0],Point_ID_Idx[0],qt,lt
                    ,'AIR',Check_Case=Check_Cased) #dsdsd
12              Table3_[0].columns = ['R']
13              Table3_[1],Count_FetchS,Count_StorageS,End_TimeS =
                    Call_Dataframe(Point_ID_Name[1],Point_ID_Idx[1],qt,lt
                    ,'AIR',Check_Case=Check_Cased)
14              Table3_[1].columns = ['S']
15              Table3_[2],Count_FetchT,Count_StorageT,End_TimeT =
                    Call_Dataframe(Point_ID_Name[2],Point_ID_Idx[2],qt,lt
                    ,'AIR',Check_Case=Check_Cased)
16              Table3_[2].columns = ['T']
17              Count_Fetch = Count_FetchR+Count_FetchS+Count_FetchT
18              Count_Storage = Count_StorageR+Count_StorageS+
                    Count_StorageT
19              End_Time = End_TimeR+End_TimeS+End_TimeT
20              for i in range(1,3):
21                  Table3_[0] = pd.merge(Table3_[0],Table3_[i],
                        left_index=True, right_index=True, how='outer')
22              Table3_[0] = Table3_[0].apply(pd.to_numeric)
23              Table = pd.DataFrame(Table3_[0].sum(1))
24              Table.columns = ['VALUE']
25      elif Data_type == 'LIGHT':
26          print '--------------------------FETCH_LIGHT
                --------------------------'
27          if len(Point_ID_Name) == 1 :
28              Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
29              Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'
                    LIGHT',Check_Case=Check_Cased)
30          elif len(Point_ID_Name) != 1 :
31              for k in range(len(Point_ID_Name)):
32                  Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
                        qt,lt,'LIGHT',Check_Case=Check_Cased)
33                  Tablek_[k].columns = ['ValueLIGHTNO%i'%(k+1)]
```

```
34                  Count_Fetch.append(Count_FetchS)
35                  Count_Storage.append(Count_StorageS)
36                  End_Time.append(End_TimeS)
37              for i in range(1,len(Point_ID_Name)):
38                  Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
39              Table = pd.DataFrame(Tablek_[0].sum(1))
40              Table.columns = ['VALUE']
41              Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
42      elif Data_type == 'OUTLET':
43          print '---------------------------FETCH_OUTLET
                ---------------------------'
44          if len(Point_ID_Name) == 1 :
45              Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
46              Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'
                    OUTLET',Check_Case=Check_Cased)
47          elif len(Point_ID_Name) != 1 :
48              for k in range(len(Point_ID_Name)):
49                  Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
                        qt,lt,'OUTLET',Check_Case=Check_Cased)
50                  Count_Fetch.append(Count_FetchS)
51                  Count_Storage.append(Count_StorageS)
52                  End_Time.append(End_TimeS)
53              for i in range(1,len(Point_ID_Name)):
54                  Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
55              Table = pd.DataFrame(Tablek_[0].sum(1))
56              Table.columns = ['VALUE']
57              Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
58      elif Data_type == 'PIR':
59          print '---------------------------FETCH_PIR
                ---------------------------'
60          if len(Point_ID_Name) == 1 :
61              Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
62              Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'PIR'
                    ,Check_Case=Check_Cased) #Output[Table] not a
                    nuerical type
63              Table['VALUE'] = Table['VALUE'].map({'ON': 1, 'OFF': 0})
                     #Mapping data
```

```
64              Table = Table.resample('1T', label='right', closed='
                    right').sum()
65              Table = Table.fillna(0)
66          elif len(Point_ID_Name) != 1 :
67              for k in range(len(Point_ID_Name)):
68                  Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
                        qt,lt,'PIR',Check_Case=Check_Cased)
69                  Count_Fetch.append(Count_FetchS)
70                  Count_Storage.append(Count_StorageS)
71                  End_Time.append(End_TimeS)
72                  Tablek_[k].columns = ['ValuePIRNO%i'%(k+1)]
73                  Tablek_[k]['ValuePIRNO%i'%(k+1)] = Tablek_[k]['
                        ValuePIRNO%i'%(k+1)].map({'ON': 1, 'OFF': 0}) #
                        Mapping data
74                  Tablek_[k].apply(pd.to_numeric)
75                  Tablek_[k] = Tablek_[k].resample('1T', label='right'
                        , closed='right').sum()
76                  Tablek_[k] = Tablek_[k].fillna(0)
77                  Tablek_[k].loc[Tablek_[k]['ValuePIRNO%i'%(k+1)] !=
                        0, 'ValuePIRNO%i'%(k+1)] = 1 #Mapping data
78              for i in range(1,len(Point_ID_Name)):
79                  Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
80              Table = pd.DataFrame(Tablek_[0].sum(1))
81              Table.columns = ['VALUE']
82              Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
83          Table.loc[Table['VALUE'] != 0, 'VALUE'] = 1 #Mapping data
84          Table.loc[Table['VALUE'] == 0, 'VALUE'] = -1 #Mapping data
85      elif Data_type == 'TEMP':
86          print '--------------------------FETCH_TEMP
                ---------------------------'
87          if len(Point_ID_Name) == 1 :
88              Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
89              Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'TEMP
                    ',Check_Case=Check_Cased)
90              Table = Table.resample('1T', label='right', closed='
                    right').mean()
91              Table = Table.fillna(method='pad')
92          elif len(Point_ID_Name) != 1 :
93              for k in range(len(Point_ID_Name)):
94                  Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
```

```
                    qt,lt,'TEMP',Check_Case=Check_Cased)
95                  Tablek_[k] = Tablek_[k].resample('1T', label='right'
                        , closed='right').mean()
96                  Tablek_[k] = Tablek_[k].fillna(method='pad')
97                  Count_Fetch.append(Count_FetchS)
98                  Count_Storage.append(Count_StorageS)
99                  End_Time.append(End_TimeS)
100             for i in range(1,len(Point_ID_Name)):
101                 Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
102             Table = pd.DataFrame(Tablek_[0].mean(1))
103             Table.columns = ['VALUE']
104             Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
105     elif Data_type == 'HUMIDITY':
106         print '--------------------------FETCH_HUMIDITY
                ---------------------------'
107         if len(Point_ID_Name) == 1 :
108             Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
109             Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'
                    HUMIDITY',Check_Case=Check_Cased)
110             Table = Table.resample('1T', label='right', closed='
                    right').mean()
111             Table = Table.fillna(method='pad')
112         elif len(Point_ID_Name) != 1 :
113             for k in range(len(Point_ID_Name)):
114                 Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
                        qt,lt,'HUMIDITY',Check_Case=Check_Cased)
115                 Tablek_[k] = Tablek_[k].resample('1T', label='right'
                        , closed='right').mean()
116                 Tablek_[k] = Tablek_[k].fillna(method='pad')
117                 Count_Fetch.append(Count_FetchS)
118                 Count_Storage.append(Count_StorageS)
119                 End_Time.append(End_TimeS)
120                 Tablek_[k].columns = ['ValueTEMPNO%i'%(k+1)]
121             for i in range(1,len(Point_ID_Name)):
122                 Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
123             Table = pd.DataFrame(Tablek_[0].mean(1))
124             Table.columns = ['VALUE']
125             Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
126     elif Data_type == 'ILLUMINANCE':
```

```
127         print '--------------------------FETCH_ILLUMINANCE
                --------------------------'
128         if len(Point_ID_Name) == 1 :
129             Point_ID_Name,Point_ID_Idx = Point_ID_Name[0],
                    Point_ID_Idx[0]
130             Table,Count_Fetch,Count_Storage,End_Time =
                    Call_Dataframe(Point_ID_Name,Point_ID_Idx,qt,lt,'
                    ILLUMINANCE',Check_Case=Check_Cased)
131             Table = Table.resample('1T', label='right', closed='
                    right').mean()
132             Table = Table.fillna(method='pad')
133         elif len(Point_ID_Name) != 1 :
134             for k in range(len(Point_ID_Name)):
135                 Tablek_[k],Count_FetchS,Count_StorageS,End_TimeS =
                        Call_Dataframe(Point_ID_Name[k],Point_ID_Idx[k],
                        qt,lt,'ILLUMINANCE',Check_Case=Check_Cased)
136                 Tablek_[k] = Tablek_[k].resample('1T', label='right'
                        , closed='right').mean()
137                 Tablek_[k] = Tablek_[k].fillna(method='pad')
138                 Count_Fetch.append(Count_FetchS)
139                 Count_Storage.append(Count_StorageS)
140                 End_Time.append(End_TimeS)
141                 Tablek_[k].columns = ['ValueTEMPNO%i'%(k+1)]
142             for i in range(1,len(Point_ID_Name)):
143                 Tablek_[0] = pd.merge(Tablek_[0],Tablek_[i],
                        left_index=True, right_index=True, how='outer')
144             Table = pd.DataFrame(Tablek_[0].mean(1))
145             Table.columns = ['VALUE']
146             Count_Fetch,Count_Storage,End_Time = sum(Count_Fetch),
                    sum(Count_Storage),sum(End_Time)
147     return Table,Count_Fetch,Count_Storage,End_Time
```

ง ชุดคำสั่งสำหรับปรับปรุงข้อมูล

ง.1 ชุดคำสั่งสำหรับแปลงแถวลำดับของเฟรมข้อมูลให้อยู่ในลักษณะเวลาตามแถวตั้ง และวันตามแถวนอน

```
def Covert_Dataframe_date_columns_time_row(df):
    col = []
    Table = pd.DataFrame()
    for group in df.groupby(df.index.date):
        col.append(str(group[1].index[0].date()))
        group[1].index = group[1].index.strftime("%H:%M:%S")
        Table = pd.concat([Table,group[1]], axis=1)
    Table.columns = col
    Table.index = pd.to_datetime(Table.index).time
    return Table
```

ง.2 ชุดคำสั่งสำหรับสร้างเฟรมข้อมูลจากข้อมูลประเภท Dictionary

```
def Create_Dataframe_from_dict(df1):
    df =df1.copy()
    Table = pd.DataFrame()
    col =[]
    for x in df.keys():
        col.append(x)
        if len(df[x].columns) != 1:
            df[x] =pd.DataFrame(df[x].sum(1))
        Table = pd.concat([Table,df[x]], axis=1)
    Table.columns = col
    return Table
```

ง.3 ชุดคำสั่งสำหรับเลือก key จากข้อมูลประเภท Dictionary

```
def Pick_some_element_in_dict(Old_dict,New_keys):
    dict_you_want = { key: Old_dict[key] for key in New_keys }
    return dict_you_want
```

จ ชุดคำสั่งฟังก์ชันวิเคราะห์การสูญเปล่าของพลังงานในระบบปรับอากาศ

จ.1 ชุดคำสั่งตัวกรองการเดินผ่าน

```
def Filter_PIR_Movepass (df,Threshold=1):
    df1 = df[df['%s'%df.columns[0]]==1]
    Table = df.copy()
    for x in df1.index:
        if (Table.loc[x- timedelta(minutes = Threshold):x+ timedelta
            (minutes = Threshold)].drop(x)==-1).all().VALUE == True :
            Table.set_value(x, 'VALUE', -1)
    return Table
```

จ.2 ชุดคำสั่งตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้จะไม่ขยับตัว

```
def Filter_PIR_Minimum_Movement(df,Threshold):
    Table = df.copy()
    df1 = df[df['%s'%df.columns[0]]==1]
    for x in df1.index:
        for y in range(1,Threshold+1):
            try:
                if Table.loc[x+ timedelta(minutes = y)].VALUE==-1:
                    Table.set_value(x+ timedelta(minutes = y), '
                        VALUE', 1)
            except: pass
    return Table
```

จ.3 ชุดคำสั่งตัวกรองระยะเวลาขั้นต่ำที่ผู้ใช้หายไปจากพื้นที่

```
def Filter_PIR_Minimum_time_User_disappear (df,Threshold):
    Table = df.copy()
    df1 = df[df['%s'%df.columns[0]]==-1]
    Time_fix = df1.index[0]
    for x in df1.index[1:]:
        if df.loc[x- timedelta(minutes = 1)].VALUE == 1:
            Table.set_value(x, 'VALUE', 0)
            Time_fix = x
        elif df.loc[x- timedelta(minutes = 1)].VALUE == -1:
            if len(df.loc[Time_fix:x])<=Threshold :
                if x in df.index[0:Threshold]:
                    pass
                else:
                    Table.set_value(x, 'VALUE', 0)
            else:pass
    return Table
```

จ.4 ชุดคำสั่งการคำนวณการสูญเปล่าของพลังงานในระบบปรับอากาศ

```
def Main_Wasted_Energy_Calculation_ALL (Scale,qt,lt,thres=15,
    Temp_thres=24,Write_Storage='Yes',Check_Value='Yes',Room="",
    Check_Case='Yes'):
    time_now = Check_time()
    Write_Time = qt.split()[0]+' 01:00:00'
    Write_Time = datetime.strptime(Write_Time, '%Y-%m-%d %H:%M:%S')
    count_PIR,count_AIR = 0,0
    Count_Fetch,Count_Storage,End_Time = [],[],[]
    Count_Fetch_Room,Count_Storage_Room,End_Time_Room  = [],[],[]
    Count_Fetch_Floor,Count_Storage_Floor,End_Time_Floor  = [],[],[]
    if Scale == 'All':
        pass
    elif Scale == 'Build4':
        Floor = All_Point_ID.loc[(All_Point_ID['place']=='eng4')]['
            fl'].unique().tolist() #Only building4 can calculate
            Wasted energy
        Table_AIR_En_Floor,Table_PIR_Floor,Table_TEMP_Floor=
            {},{},{}
        Total_wasted_energy_floor,Total_Energy_floor,
            Ratio_Waste_floor= {},{},{}
        Total_wasted_energy_by_PIR_floor,
            Total_wasted_energy_by_TEMP_floor,Table_PIR_U_Floor =
            {},{},{}
        for z in Floor:
            Table_AIR_En_Floor[z],Total_wasted_energy_by_PIR_floor[z
                ],Total_wasted_energy_by_TEMP_floor[z],Count_Fetch,
                Count_Storage,End_Time,Total_wasted_energy_floor[z],
                Total_Energy_floor[z],Ratio_Waste_floor[z],
                Table_PIR_Floor[z],Table_PIR_U_Floor[z],
                Table_TEMP_Floor[z] =
                Main_Wasted_Energy_Calculation_ALL (z,qt,lt,thres,
                Temp_thres,Write_Storage,Check_Value,Check_Case=
                Check_Case)
            Count_Fetch_Floor.append(Count_Fetch)
            Count_Storage_Floor.append(Count_Storage)
            End_Time_Floor.append(End_Time)
        return Table_AIR_En_Floor,Total_wasted_energy_by_PIR_floor,
            Total_wasted_energy_by_TEMP_floor,sum(Count_Fetch_Floor),
            sum(Count_Storage_Floor),sum(End_Time_Floor),
            Total_wasted_energy_floor,Total_Energy_floor,
            Ratio_Waste_floor,Table_PIR_Floor,Table_PIR_U_Floor,
            Table_TEMP_Floor
    elif Scale == 'fl13':
        Have_sensor = set(All_Point_ID.loc[(All_Point_ID['fl']=='
```

```
            fl13') & ((All_Point_ID['kind']=='sensor1')|(All_Point_ID
            ['kind']=='sensor2') )]['room'].unique().tolist())
24          Have_Air = set(All_Point_ID.loc[(All_Point_ID['fl']=='fl13')
                & ((All_Point_ID['kind']=='aircon1') | (All_Point_ID['
            kind']=='aircon2')|(All_Point_ID['kind']=='aircon_3ph1')
25          | (All_Point_ID['kind']=='aircon_3ph2') )]['room'].unique().
            tolist())
26          Room = list(Have_sensor.intersection(Have_Air))
27          Table_AIR_En_Room,Total_wasted_energy_by_PIR_Room,
            Total_wasted_energy_by_TEMP_Room= {},{},{}
28          Total_wasted_energy_Room,Total_Energy_Room,Ratio_Waste_Room,
            Table_PIR_Room,Table_PIR_U_Room,Table_TEMP_Room =
            {},{},{},{},{},{}
29          for y in Room:
30              Table_AIR_En_Room[y],Total_wasted_energy_by_PIR_Room[y],
                Total_wasted_energy_by_TEMP_Room[y],Count_Fetch,
                Count_Storage,End_Time,Total_wasted_energy_Room[y],
                Total_Energy_Room[y],Ratio_Waste_Room[y],
                Table_PIR_Room[y],Table_PIR_U_Room[y],Table_TEMP_Room
                [y] = Main_Wasted_Energy_Calculation_ALL ('room',qt,
                lt,thres,Temp_thres,Write_Storage,Check_Value,Room=y,
                Check_Case=Check_Case)
31              Count_Fetch_Room.append(Count_Fetch)
32              Count_Storage_Room.append(Count_Storage)
33              End_Time_Room.append(End_Time)
34          return Table_AIR_En_Room,Total_wasted_energy_by_PIR_Room,
            Total_wasted_energy_by_TEMP_Room,sum(Count_Fetch_Room),
            sum(Count_Storage_Room),sum(End_Time_Room),
            Total_wasted_energy_Room,Total_Energy_Room,
            Ratio_Waste_Room,Table_PIR_Room,Table_PIR_U_Room,
            Table_TEMP_Room
35      elif Scale == 'fl12':
36          Have_sensor = set(All_Point_ID.loc[(All_Point_ID['fl']=='
            fl12') & ((All_Point_ID['kind']=='sensor1')|(All_Point_ID
            ['kind']=='sensor2') )]['room'].unique().tolist())
37          Have_Air = set(All_Point_ID.loc[(All_Point_ID['fl']=='fl12')
                & ((All_Point_ID['kind']=='aircon1') | (All_Point_ID['
            kind']=='aircon2')|(All_Point_ID['kind']=='aircon_3ph1')
38          | (All_Point_ID['kind']=='aircon_3ph2') )]['room'].unique().
            tolist())
39          Room = list(Have_sensor.intersection(Have_Air))
40          Table_AIR_En_Room,Total_wasted_energy_by_PIR_Room,
            Total_wasted_energy_by_TEMP_Room= {},{},{}
41          Total_wasted_energy_Room,Total_Energy_Room,Ratio_Waste_Room,
            Table_PIR_Room,Table_PIR_U_Room,Table_TEMP_Room =
            {},{},{},{},{},{}
```

```
42          for y in Room:
43              #print Y-
44              Table_AIR_En_Room[y],Total_wasted_energy_by_PIR_Room[y],
                    Total_wasted_energy_by_TEMP_Room[y],Count_Fetch,
                    Count_Storage,End_Time,Total_wasted_energy_Room[y],
                    Total_Energy_Room[y],Ratio_Waste_Room[y],
                    Table_PIR_Room[y],Table_PIR_U_Room[y],Table_TEMP_Room
                    [y] = Main_Wasted_Energy_Calculation_ALL ('room',qt,
                    lt,thres,Temp_thres,Write_Storage,Check_Value,Room=y,
                    Check_Case=Check_Case)
45              Count_Fetch_Room.append(Count_Fetch)
46              Count_Storage_Room.append(Count_Storage)
47              End_Time_Room.append(End_Time)
48          return Table_AIR_En_Room,Total_wasted_energy_by_PIR_Room,
                Total_wasted_energy_by_TEMP_Room,sum(Count_Fetch_Room),
                sum(Count_Storage_Room),sum(End_Time_Room),
                Total_wasted_energy_Room,Total_Energy_Room,
                Ratio_Waste_Room,Table_PIR_Room,Table_PIR_U_Room,
                Table_TEMP_Room
49      elif Scale == 'room':
50          PIR_Movepass  = 1
51          Minimun_Movement = 3
52          Table_AIR_En,Table_PIR,Table_PIR_U,Table_TEMP= {},{},{},{}
53          Total_wasted_energy,Total_Energy,Ratio_Waste,
                Total_wasted_energy_by_PIR,Total_wasted_energy_by_TEMP =
                {},{},{},{},{}
54          Zone = All_Point_ID.loc[(All_Point_ID['room']=='%s'%Room)]['
                zone'].unique().tolist()
55          print '****************************************',Room,'
                ****************************************'
56          print '******************',Room,' have ',len(Zone),'
                ******************'
57          print 'Start time =' ,qt
58          print 'End time =' ,lt
59          for x in Zone:
60              floors =  All_Point_ID['fl'].loc[(All_Point_ID['room']==
                    Room) ].unique()[0]
61              building = All_Point_ID['place'].loc[(All_Point_ID['room
                    ']==Room) ].unique()[0]
62              direction =  All_Point_ID['direction'].loc[(All_Point_ID
                    ['room']==Room) ].unique()[0]
63              #1 value per day Point_ID_Wasted_Ratio
                    Point_ID_Wasted_from_PIR Point_ID_Wasted_from_TEMP
                    Point_ID_Wasted
64              print thres , Temp_thres
65              if thres == 15 and Temp_thres == 24 :
```

```
66                  Point_ID_Wasted_Ratio = 'http://khetnon/%s/%s/%s/%s
                        /%s/analytic/wasted_energy/per_day'%(building,
                        floors,direction,Room,x)
67                  Point_ID_Wasted_from_PIR = 'http://khetnon/%s/%s/%s
                        /%s/%s/analytic/wasted_energy/
                        per_day_from_User_motion'%(building,floors,
                        direction,Room,x)
68                  Point_ID_Wasted_from_TEMP = 'http://khetnon/%s/%s/%s
                        /%s/%s/analytic/wasted_energy/
                        per_day_from_Area_temperature'%(building,floors,
                        direction,Room,x)
69                  #every 15 minute
70                  Point_ID_Wasted = 'http://khetnon/%s/%s/%s/%s/%s/
                        analytic/wasted_energy/per_day_value'%(building,
                        floors,direction,Room,x)
71              else:
72                  print 'New Point_ID'
73                  Point_ID_Wasted_Ratio = 'http://khetnon/%s/%s/%s/%s
                        /%s/analytic/wasted_energy/per_day/Userdis_%
                        s_Temp_%s'%(building,floors,direction,Room,x,
                        thres,Temp_thres)
74                  Point_ID_Wasted_from_PIR = 'http://khetnon/%s/%s/%s
                        /%s/%s/analytic/wasted_energy/
                        per_day_from_User_motion/Userdis_%s_Temp_%s'%(
                        building,floors,direction,Room,x,thres,Temp_thres
                        )
75                  Point_ID_Wasted_from_TEMP = 'http://khetnon/%s/%s/%s
                        /%s/%s/analytic/wasted_energy/
                        per_day_from_Area_temperature/Userdis_%s_Temp_%s'
                        %(building,floors,direction,Room,x,thres,
                        Temp_thres)
76                  #every 15 minute
77                  Point_ID_Wasted = 'http://khetnon/%s/%s/%s/%s/%s/
                        analytic/wasted_energy/per_day_value/Userdis_%
                        s_Temp_%s'%(building,floors,direction,Room,x,
                        thres,Temp_thres)
78              print '
                    ----------------------------------------------------------------
                    '
79              print '
                    ----------------------------------------------------------------
                    '
80              print 'Point_ID_Wasted_Ratio   :',Point_ID_Wasted_Ratio
81              print 'Point_ID_Wasted   :',Point_ID_Wasted
82              print 'Point_ID_Wasted_from_PIR   :',
                    Point_ID_Wasted_from_PIR
```

```
83              print 'Point_ID_Wasted_from_TEMP   :',
                    Point_ID_Wasted_from_TEMP
84              print '----',Room,'  ',x,'---'
85              if Check_Value == 'Yes':
86                  Wasted_Value_Ratio = Fetch_His(Point_ID_Wasted_Ratio
                        ,lt,qt,'Waste')
87                  Wasted_Value = Fetch_His(Point_ID_Wasted,lt,qt,'
                        Waste')
88                  Wasted_Value_PIR = Fetch_His(
                        Point_ID_Wasted_from_PIR,lt,qt,'Waste')
89                  Wasted_Value_TEMP = Fetch_His(
                        Point_ID_Wasted_from_TEMP,lt,qt,'Waste')
90              elif Check_Value == 'No':
91                  Wasted_Value_Ratio = pd.DataFrame()
92                  Wasted_Value = pd.DataFrame()
93                  Wasted_Value_PIR = pd.DataFrame()
94                  Wasted_Value_TEMP = pd.DataFrame()
95              if Wasted_Value_Ratio.empty == True or Wasted_Value.
                    empty == True or Wasted_Value_TEMP.empty == True or
                    Wasted_Value_PIR.empty == True:
96              #try:
97                  Command_AIR_EN = Find_PointID(room=Room,zone=x,kind=
                        'Air')
98                  print '--------------------------AIR
                        -------------------------------'
99                  if len(Command_AIR_EN['Point_Name']) != 1 :
100                     Point_ID_AIR = Command_AIR_EN['Point_Name']
101                     idx_AIR = Command_AIR_EN['IND']
102                 else:
103                     Point_ID_AIR = [Command_AIR_EN['Point_Name'][0]]
104                     idx_AIR = [Command_AIR_EN['IND'][0]]
105                 print Point_ID_AIR,idx_AIR
106                 print '--------------------------PIR
                        -------------------------------'
107                 Command_PIR = Find_PointID(room=Room,zone=x,value='
                        pir')
108                 if len(Command_PIR['Point_Name']) != 1 :
109                     Point_ID_PIR = Command_PIR['Point_Name']
110                     idx_PIR = Command_PIR['IND']
111                 else:
112                     Point_ID_PIR = [Command_PIR['Point_Name'][0]]
113                     idx_PIR = [Command_PIR['IND'][0]]
114                 print Point_ID_PIR,idx_PIR
115                 print '--------------------------TEMP
                        -------------------------------'
116                 Command_AIR_TEMP = Find_PointID(room=Room,zone=x,
```

```
                    value='temperature')
117                 if len(Command_AIR_TEMP['Point_Name']) != 1 :
118                     Point_ID_TEMP = Command_AIR_TEMP['Point_Name']
119                     idx_TEMP = Command_AIR_TEMP['IND']
120                 else:
121                     Point_ID_TEMP = [Command_AIR_TEMP['Point_Name'
                            ][0]]
122                     idx_TEMP = [Command_AIR_TEMP['IND'][0]]
123                 print Point_ID_TEMP,idx_TEMP
124
125                 print 'Aircon = 1','  PIR = ',len(Command_AIR_TEMP['
                    Point_Name']),'  Temp = ',len(Command_AIR_TEMP['
                    Point_Name'])
126                 count_PIR = count_PIR + len(Command_AIR_TEMP['
                    Point_Name'])
127                 count_AIR = count_AIR + 1
128                 Count_FetchS_AIR,Count_StorageS_AIR,End_TimeS_AIR =
                    0,0,0
129                 Count_FetchS_PIR,Count_StorageS_PIR,End_TimeS_PIR =
                    0,0,0
130                 Count_FetchS_TEMP,Count_StorageS_TEMP,End_TimeS_TEMP
                     = 0,0,0
131                 if Point_ID_AIR != [] and idx_AIR != []:
132                     Table_AIR_En[x],Count_FetchS_AIR,
                            Count_StorageS_AIR,End_TimeS_AIR =
                            Prepare_Data('AIR',Point_ID_AIR,idx_AIR,qt,lt
                            ,Check_Cased=Check_Case)
133                     print 'Filter AIR data'
134                     Table_AIR_En[x] = Table_AIR_En[x][Table_AIR_En[x
                            ]<300] #fill error
135                     Table_AIR_En[x] = Table_AIR_En[x][Table_AIR_En[x
                            ]>1]
136                     Table_AIR_En[x] = Table_AIR_En[x].fillna(0)
137                     print 'Count_FetchS',Count_FetchS_AIR,'
                            Count_StorageS',Count_StorageS_AIR,'
                            End_TimeS',End_TimeS_AIR
138                 if Point_ID_PIR != [] and idx_PIR != []:
139                     Table_PIR[x],Count_FetchS_PIR,Count_StorageS_PIR
                            ,End_TimeS_PIR = Prepare_Data('PIR',
                            Point_ID_PIR,idx_PIR,qt,lt,Check_Cased=
                            Check_Case)
140                     print 'Count_FetchS',Count_FetchS_PIR,'
                            Count_StorageS',Count_StorageS_PIR,'
                            End_TimeS',End_TimeS_PIR
141                 if Point_ID_TEMP != [] and idx_TEMP != []:
142                     Table_TEMP[x],Count_FetchS_TEMP,
```

```
                    Count_StorageS_TEMP,End_TimeS_TEMP =
                    Prepare_Data('TEMP',Point_ID_TEMP,idx_TEMP,qt
                    ,lt,Check_Cased=Check_Case)
143                 print 'Filter Temperature data'
144
145                 print 'Count_FetchS',Count_FetchS_TEMP,'
                    Count_StorageS',Count_StorageS_TEMP,'
                    End_TimeS',End_TimeS_TEMP
146             Count_Fetch.append(Count_FetchS_AIR+Count_FetchS_PIR
                +Count_FetchS_TEMP)
147             Count_Storage.append(Count_StorageS_AIR+
                Count_StorageS_PIR+Count_StorageS_TEMP)
148             End_Time.append(End_TimeS_AIR+End_TimeS_PIR+
                End_TimeS_TEMP)
149             ###### Process prepare data end
150             ####################################
151             if Table_AIR_En[x].empty != True and Table_PIR[x].
                empty != True and Table_TEMP[x].empty != True:
152                 try:
153                     print '>>Filter PIR Data>>'
154                     Table_PIR[x] = Filter_PIR_Movepass(Table_PIR
                        [x],PIR_Movepass)
155                     Table_PIR[x] = Filter_PIR_Minimum_Movement(
                        Table_PIR[x],Minimun_Movement)
156                     Table_PIR_U[x] =
                        Filter_PIR_Minimum_time_User_disappear(
                        Table_PIR[x],thres)
157                     print '>>>>>>Cal>>>>>>>>>'
158                     Table = pd.merge(Table_AIR_En[x],Table_PIR_U
                        [x],left_index=True, right_index=True,
                        how='outer')
159                     Table = pd.merge(Table,Table_TEMP[x],
                        left_index=True, right_index=True, how='
                        outer')
160                     Table.columns = ['EN_AIR','PIR','TEMP']
161                     Total_wasted_energy_by_PIR[x] = Table['
                        EN_AIR'].loc[(Table['EN_AIR']>0)&(Table['
                        PIR']==-1)].sum()
162                     Total_wasted_energy_by_TEMP[x] = Table['
                        EN_AIR'].loc[(Table['EN_AIR']>0)&(Table['
                        PIR']==1)&(Table['TEMP']<24)].sum()
163                     Total_wasted_energy[x] = Table['EN_AIR'].loc
                        [((Table['EN_AIR']>0)&(Table['PIR']==-1))
                        |((Table['EN_AIR']>0)&(Table['PIR']==1)&(
                        Table['TEMP']<24))]
164                     Total_wasted_energy[x] = Total_wasted_energy
```

```
                [x].resample('15T', label='right').sum().
                fillna(0)
165             Total_wasted_energy[x]  = Create_time_slot(
                Total_wasted_energy[x],Table_AIR_En[x])
166             Total_Energy[x] = Table['EN_AIR'].sum()
167             #print Total_Energy[x]
168             if Total_Energy[x]!=0:
169                 Ratio_Waste[x] = Total_wasted_energy[x][
                    'EN_AIR'].sum()*100/Total_Energy[x]
170             else:
171                 Ratio_Waste[x] = float('nan')
172             if Write_Storage == 'Yes':
173                 #Write every 15 min
174                 #if Wasted_Value.empty == True:
175                 print 'WRITE Wasted_Value to Storage
                    .............'
176                 for Write_time_minute in
                    Total_wasted_energy[x].index:
177                     print Write_time_minute,'  ',
                        Total_wasted_energy[x].loc[
                        Write_time_minute]['EN_AIR']
178                     print Point_ID_Wasted
179                     Write_IEEE1888(Point_ID_Wasted,
                        Total_wasted_energy[x].loc[
                        Write_time_minute]['EN_AIR'],
                        Write_time_minute)
180                 #Write 1 time per day
181                 if Wasted_Value_Ratio.empty == True :
182                     print 'WRITE Ratio of Wasted Value
                        to Storage .............'
183                     Write_IEEE1888(Point_ID_Wasted_Ratio
                        ,Ratio_Waste[x],Write_Time)
184                 if Wasted_Value_TEMP.empty == True :
185                     print 'WRITE Wasted Value from Area
                        temperature to Storage
                        .............'
186                     Write_IEEE1888(
                        Point_ID_Wasted_from_TEMP,
                        Total_wasted_energy_by_TEMP[x],
                        Write_Time)
187                 if Wasted_Value_PIR.empty == True :
188                     print 'WRITE Wasted_Value from User
                        motion to Storage .............'
189                     Write_IEEE1888(
                        Point_ID_Wasted_from_PIR,
                        Total_wasted_energy_by_PIR[x],
```

```
                                    Write_Time)
190                         elif Write_Storage == 'ReWrite':
191                             print 'WRITE Wasted_Value to Storage
                                    .............'
192                             #Write every 15 min
193                             for Write_time_minute in
                                    Total_wasted_energy[x].index:
194                                 print Write_time_minute,'  ',
                                        Total_wasted_energy[x].loc[
                                        Write_time_minute]['EN_AIR']
195                                 print Point_ID_Wasted
196                                 Write_IEEE1888(Point_ID_Wasted,
                                        Total_wasted_energy[x].loc[
                                        Write_time_minute]['EN_AIR'],
                                        Write_time_minute)
197                             #Write 1 time per day
198                             print 'WRITE Ratio of Wasted Value to
                                    Storage .............'
199                             Write_IEEE1888(Point_ID_Wasted_Ratio,
                                    Ratio_Waste[x],Write_Time)
200                             print 'WRITE Wasted Value from Area
                                    temperature to Storage .............'
201                             Write_IEEE1888(Point_ID_Wasted_from_TEMP
                                    ,Total_wasted_energy_by_TEMP[x],
                                    Write_Time)
202                             print 'WRITE Wasted_Value from User
                                    motion to Storage .............'
203                             Write_IEEE1888(Point_ID_Wasted_from_PIR,
                                    Total_wasted_energy_by_PIR[x],
                                    Write_Time)
204                         elif Write_Storage == 'No':
205                             print 'NO write to IEEE1888 storage'
206                         else: print 'Worng input not Write'
207                     except:
208                         print 'Error to filter PIR data'
209                         pass
210                 else:
211                     print 'Not enough data ....'
212             #except:
213                 #print 'Can not caluculated .....'
214             else:
215                 print 'Already calculated'
216         return Table_AIR_En,Total_wasted_energy_by_PIR,
                Total_wasted_energy_by_TEMP,sum(Count_Fetch),sum(
                Count_Storage),sum(End_Time),Total_wasted_energy,
                Total_Energy,Ratio_Waste,Table_PIR,Table_PIR_U,Table_TEMP
```

```
217     else:
218         print "Wrong Command"
```

ฉ ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม

ฉ.1 ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม 1

```
def Check_sensor(Building='eng4',Mode='Test',Report_day='Monday'):
    Time_now = Check_time()
    if Mode == 'Weakly report':
        if datetime.strftime(Time_now,"%A") == Report_day:
            Main_Check_sensor(Building='eng4',Time=Time_now )
        else : print 'Not in %s'%Report_day
    elif Mode == 'Test':
        Main_Check_sensor(Building='eng4',Time=Time_now)
    elif Mode == 'Web analytic':
        return Main_Check_sensor(Building='eng4',Time=Time_now,
            Return_message='Yes')
    else : print 'Wrong command'
```

ฉ.2 ชุดคำสั่งสำหรับตรวจสอบสถานะการทำงานของตัวรับรู้สภาพแวดล้อม 2

```
def Main_Check_sensor(Time,Building='eng4',Return_message='No'):
    Time_now =Time.date()
    result,result_time = [],[]
    for Point in Find_PointID(place=Building)['Point_Name']:
        print Point
        try:Time_Last_value = pd.to_datetime(Check_Max_Min_time(
            Point,'maximum')[0]).date()
        except: pass
        if Time_now !=Time_Last_value:
            result.append(Point)
            result_time.append(Time_Last_value)

    Result = pd.Series(result)
    Result = Result.str.rsplit('/', expand=True,n=8)
    coll = ['http', 'place','fl','direction','room','zone','kind','
        duty','value']
    Result.columns = coll
    noitimessage = []
    for Bld in Result['place'].unique().tolist():
        for floor in Result[Result['place']==Bld]['fl'].unique().
            tolist():
            for room in Result[(Result['place']==Bld)&(Result['fl'
                ]==floor)]['room'].unique().tolist():
                for zone in Result[(Result['place']==Bld)&(Result['
                    fl']==floor)&(Result['room']==room)]['zone'].
                    unique().tolist():
                    messageinzone = []
```

```
22                      for kind in Result[(Result['place']==Bld)&(
                            Result['fl']==floor)&(Result['room']==room)&(
                            Result['zone']==zone)]['kind'].unique().
                            tolist():
23                           messageinzone.append(kind)
24                      noitimessage.append(room+' in '+ zone+' | ' +  '
                             and '.join(messageinzone) + '\n')
25      mess_noti =  '[Weakly report in  %s Equipment maintainace] :
            Please Check \n-%s'% (datetime.strftime(datetime.now(),"%A %Y
            -%m-%d") ,'-'.join(noitimessage))
26      if Return_message == 'Yes':
27          Data_sensor = {'Point Name':result,'Last value':result_time}
28          return Data_sensor
29      else:
30          req = urllib2.Request('https://notify-api.line.me/api/notify
                ')
31          req.add_header('Content-type', 'application/x-www-form-
                urlencoded')
32          req.add_header('Authorization', 'Bearer
                ndX9ALbZKUhnndGHYOrl4Da1tuZlwzWYLQfGQgNe7LW')
33          urllib2.urlopen(req,"message=%s"%mess_noti)
```

ช ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของพลังงานในระบบปรับอากาศ

ช.1 ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของพลังงานในระบบปรับอากาศ 1

```
def Report_Wasted(Scale_building='eng4',room='',Scale_time=1,
    Pir_thes=15,Temp_thes=24,Mode='Test',Report_day='Monday',
    Rank_reports=5):
     Time_now =datetime.now().date()
     if Scale_building=='eng4': rooms = (Check_Room_name('fl12') +
        Check_Room_name('fl13'))
     elif Scale_building=='floor13' : rooms = Check_Room_name('fl13')
     elif Scale_building=='floor12' : rooms= Check_Room_name('fl12')
     elif Scale_building=='room':rooms=room
     elif Scale_building == 'Instructors Room':
         rooms=[]
         for x in Check_Room_name('fl12') +Check_Room_name('fl13'):
             if 'room' in x.split("_"):
                 if 'server' in  x.split("_") or 'pantry' in  x.split
                     ("_"):pass
                 else:rooms.append(x)
     else: print 'Wrong command'
     if Mode == 'Weekly report':
         if datetime.strftime(Time_now,"%A") == Report_day:
             Main_Report_Wasted(room=rooms,Modet='Week',time=
                 Scale_time,Pirthes=Pir_thes,Tempthes=Temp_thes,
                 Rank_report=Rank_reports)
         else : print 'Not in %s'%Report_day
     elif Mode == 'Test':
         Main_Report_Wasted(room=rooms,Modet='Week',time=Scale_time,
             Pirthes=Pir_thes,Tempthes=Temp_thes,Rank_report=
             Rank_reports)
     else : print 'Wrong command'
```

ช.2 ชุดคำสั่งการแจ้งเตือนผ่านโปรแกรมประยุกต์ไลน์ของฟังก์ชันการสูญเปล่าของพลังงานในระบบปรับอากาศ 2

```
def Main_Report_Wasted(room,Rank_report,Modet='Week',time=1,Pirthes
    =15,Tempthes=24,tariff_rate=4.18):
     if type(room) == str :
         Modes = 'Unique'
     elif type(room) == list:
         Modes = 'Compare'
     result = Compare_Wasted_Energy(Mode=Modes,Mode_time=Modet,Time=
```

```
        time,Room=room,Pir_thes=Pirthes,Temp_thes=Tempthes,
        Seperate_zone='Off',Cal_Mode='Sum',Plot='Off',Return='Yes',
        Check_data='No')
    if type(result)==dict:
        Room_name,Room_value=[],[]
        if len(result.keys())<= Rank_report:
            Rank_report = int(len(result.keys())/2)
        for x in result.keys():
            Room_value.append(result[x].sum()[0])
            Room_name.append(x)
        result_ = pd.DataFrame(data=Room_value,index=Room_name,
            columns=['Value'])
        noitimessage = []
        result = result_.sort_values('Value',ascending =False).iloc
            [0:Rank_report]
        for x in result.index:
            noitimessage.append(' '+x+' total wasted energy : '+"
                {0:.2f}".format(result.loc[x].Value/1000)+' kWh'+'
                >>> Expenses : '+str(int(result.loc[x].Value/1000*
                tariff_rate))+' Bath' + '\n')
        mess_noti =  '[Weakly Wasted energy in AIR conditioner
            system report in  %s ] :\n Top %s Wasted energy areas
            last %s days \n-%s'% (datetime.strftime(datetime.now(),"%
            A %Y-%m-%d"),Rank_report ,time*7,'-'.join(noitimessage))
    elif type(result) == type(pd.DataFrame()):
        result_ = "{0:.2f}".format(result.sum()[0]/1000)
        mess_noti =  '[Weakly Wasted energy in AIR conditioner
            system report in  %s ] Total wasted energy last %s days
            in %s : %s kWh  >>> Expenses : %s Bath  '%(datetime.
            strftime(datetime.now(),"%A %Y-%m-%d")
        ,time*7,room,result_,int(result.sum()[0]/1000*tariff_rate))
    req = urllib2.Request('https://notify-api.line.me/api/notify')
    req.add_header('Content-type', 'application/x-www-form-
        urlencoded')
    req.add_header('Authorization', 'Bearer
        ndX9ALbZKUhnndGHYOrl4Da1tuZlwzWYLQfGQgNe7LW')
    urllib2.urlopen(req,"message=%s"%mess_noti)
```

ประวัติผู้เขียนวิทยานิพนธ์

นายเขตต์นนท์ ชูพุทธิพงศ์ เกิดเมื่อวันอาทิตย์ที่ 10 พฤศจิกายน พ.ศ. 2534 จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิตจากจุฬาลงกรณ์มหาวิยาลัย คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า ปีการศึกษา 2556 และได้ศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิยาลัย คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า กลุ่มวิจัยโครง-ข่ายไฟฟ้าอัจฉริยะและพลังงานหมุนเวียน ปีการศึกษา 2557

บทความทางวิชาการจากวิทยานิพนธ์

[1] Chooputtipong, K., and Aswakul, C. Development of data analytic program for building enegy management system with wasted energy analysis using motion sensor. The 39th Electrical Engineering Conference (EECON-39), Phetchaburi City, Thailand, 2016.