

การวิเคราะห์และคาดการณ์กิจกรรมการเคลื่อนไหวด้วยระบบ IoT

บุรินทร์ วิจารณ์รักษ์

TNI

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

บัณฑิตวิทยาลัย สาขาเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2564

PREDICTIVE MODEL BASED ON ACTIVITY BY USING IOT

Burin Wiranurak

TNI

A Term Paper Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology

Graduate School
Thai-Nichi Institute of Technology

Academic Year 2021

หัวข้อสารนิพนธ์

การวิเคราะห์และคาดการณ์กิจกรรมการเคลื่อนไหวด้วยระบบ IoT

โดย

บุรินทร์ วิจารณ์รักษ์

สาขาวิชา

เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาสารนิพนธ์

รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้รับสารนิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทบริหารบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร.พิชิต สุขเจริญพงษ์)
วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบสารนิพนธ์

.....ประธานกรรมการ
(ดร.สรมย์พร เจริญพิทย์)

.....กรรมการ
(ดร.ณปภัช วิชัยดิษฐ์)

.....อาจารย์ที่ปรึกษาสารนิพนธ์
(รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล)

บุรินทร์ วีราณูรักษ์ : การวิเคราะห์และคาดการณ์กิจกรรมการเคลื่อนไหวด้วยระบบ IoT.
 อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.อรรณพ หมั่นสกุล, 90 หน้า.

ในปัจจุบัน ได้มีบุคลากรจำนวนมากได้มีการเจ็บป่วยอันเนื่องมาจากโรคออฟฟิศซินโดรม ซึ่งเป็นผลมาจากการทำงานที่ในท่าทางที่ผิดท่า หรือการทำงานในลักษณะใดลักษณะหนึ่งเป็นเวลานาน และไม่มีการเปลี่ยนท่าทางไม่ว่าจะเป็นการนั่งหรือการยืน เป็นต้น ในงานวิจัยชิ้นจึงเล็งเห็นว่าควรทำการสร้างโมเดลที่ช่วยในการแยกแยะอิริยาบถหรือกิจกรรมต่างๆ ออกมา ผ่านทางอุปกรณ์ IoT ในการเก็บข้อมูลจากผู้ใช้งาน โดยนำโมเดลเหล่านี้ไปใช้งานในอุปกรณ์เทคโนโลยีในอนาคตได้

โดยการทำงานวิจัยชิ้นนี้ จะเริ่มทำการเก็บข้อมูลผ่านอุปกรณ์ IoT ที่พัฒนาขึ้น โดยทำการเก็บข้อมูลอิริยาบถหรือกิจกรรมเป็นช่วงเวลาหนึ่งเข้าสู่ระบบดาต้าเบส โดยที่กำหนดท่าทางกิจกรรมออกมาเป็น 4 รูปแบบ ประกอบไปด้วย การนั่ง การยืน การเดิน และนอนราบ มาทำแปลงค่าเพื่อใช้ในการเอาเข้าสู่กระบวนการพัฒนาโมเดล เพื่อใช้ในการแยกแยะ โดยโมเดลที่ใช้ในการพัฒนานั้นจะเป็น Feed Forward Neural Network ในการแยกแยะลักษณะท่าทางรูปแบบในก่อนหน้านี้ที่กล่าวไป

ภายหลังการดำเนินงานวิจัยจนเสร็จสิ้นพบว่า โมเดลนั้นมีความแม่นยำในระดับที่ดีเยี่ยม โดยที่สามารถปรับปรุงในส่วนอุปกรณ์ IoT ได้ในอนาคต เพื่อใช้เป็นข้อมูลในส่วนหนึ่งในการเอาเข้าสู่ระบบโมเดลในการแยกแยะและยังพบว่าปัจจัยภายนอกบางอย่างนั้นอาจส่งผลกระทบต่อการใช้ข้อมูลเพื่อใช้ในการทำงานได้ เช่น อุณหภูมิ งานวิจัยชิ้นนี้จึงเป็นแนวทางให้ผู้ทำงานวิจัยท่านอื่นๆ ใช้ออนาคตเพื่อใช้เป็นแนวทางได้

TNI

บัณฑิตวิทยาลัย

สาขาวิชา เทคโนโลยีสารสนเทศ

ปีการศึกษา 2564

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

BURIN WIRANURAK : PREDICTIVE MODEL BASED ON ACTIVITY BY USING IOT
ADVISOR : ASSOC.PROF.DR.ANNOP MONSAKUL, 90 PP.

In this time, there are million patients got sick from “Office Syndrome” which resulted from wrong gestures during working alike sit, stand, walk etc.

In this research, we will utilize IoT technology in order to create the model to identify activities from user. Hence, the model from this research will help to develop technology of IoT (Internet of Thing) in the future.

The methodology is that we will collect the four activities data which are sit, stand, walk and recumbent from the IoT equipment then save into the database. After that we input the data into a model development which will identify the activity pattern based on Feed Forward Neural Network.

We found the result is highly strong accuracy for activity prediction. Besides, it has more opportunities to improve IoT equipment in future for collecting more data that would help to develop predictive model by using this research as a guidance. Last but not least, we also found that the environment like temperature during collecting data directly affects to the output.

TNI

Graduate School

Field of Study Information Technology

Academic Year 2021

Student's Signature.....

Advisor's Signature.....

กิตติกรรมประกาศ

สารนิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี โดยได้รับความช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อรรณพ หมั่นสกุล เป็นอาจารย์ที่ปรึกษาสารนิพนธ์ ผู้ให้คำแนะนำ ข้อคิดเห็น และอนุমติ แก่ผู้วิจัยด้วยความเมตตา

ขอกราบขอบพระคุณ ดร.สรมย์พร เจริญพิทย์ และดร.ณปภัช วิชัยดิษฐ์ เป็นคณะกรรมการสอบสารนิพนธ์

ขอกราบขอบพระคุณ ดร.ศรายุทธ นนท์ศิริ ที่ให้การช่วยเหลือ และสนับสนุนในการให้คำปรึกษาในส่วนของงานวิจัยในด้านการทำโมเดลการคิดวิเคราะห์

ขอกราบขอบพระคุณ ดร.สพรั่งสิทธิ์ มฤตสาธ และดร.ณัฐกิตติ จิตรเอื้อตระกูล ที่ให้การช่วยเหลือในส่วนของการคำแนะนำในส่วนของหัวข้องานวิจัย

ขอกราบขอบพระคุณ คณาจารย์ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่ประสิทธิ์ประสาทความรู้ และอบรมสั่งสอนผู้วิจัยด้วย ความรักและเอาใจใส่เสมอมา ตลอดระยะเวลาที่ศึกษา ณ สถาบันแห่งนี้เสมอมาจนสำเร็จการศึกษา

บุรินทร์ วิจารณ์รักษ์

TNI

THAI - JAPAN INSTITUTE OF TECHNOLOGY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ความสำคัญของการวิจัย.....	2
1.4 ขอบเขตของการวิจัย.....	3
2 วรรณกรรมที่เกี่ยวข้อง.....	4
2.1 วิธีการคัดกรองแยกแยะข้อมูลในการคาดการณ์ถึงลักษณะท่าทางของกิจกรรม.....	4
2.2 องค์ประกอบความสัมพันธ์ระหว่างข้อมูลและอุปกรณ์.....	10
2.3 อุปกรณ์เครื่องมือในการใช้ตรวจจับและส่งข้อมูล.....	12
3 วิธีดำเนินการวิจัย.....	16
3.1 เครื่องมือในการเก็บข้อมูลและแสดงผลการทดลองในงานวิจัย.....	16
3.2 วิธีการในการทำงานวิจัย.....	20
3.3 ลำดับขั้นตอนการดำเนินงาน.....	23
3.4 ข้อมูลที่นำเข้ามาใช้เข้ากับโมเดลและทำการทดลอง.....	25

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการวิเคราะห์ข้อมูล.....	29
4.1 การวิเคราะห์ในส่วนของข้อมูล.....	29
4.2 การวิเคราะห์ในส่วนของโมเดล.....	43
5 สรุปผลการวิจัย.....	50
5.1 ข้อเสนอของงานวิจัย.....	50
5.2 การพัฒนางานวิจัยในอนาคต.....	53
บรรณานุกรม.....	56
ภาคผนวก.....	60
ภาคผนวก ก. รูปภาพประกอบงานวิจัยอ้างอิงจากเว็บไซต์.....	61
ภาคผนวก ข. โค้ดส่วนของอุปกรณ์ IoT.....	67
ภาคผนวก ค. โค้ดส่วน Model Python.....	74
ภาคผนวก ง. โค้ดส่วนการแปลงค่าข้อมูล.....	83
ภาคผนวก จ. วงจรการเชื่อมต่ออุปกรณ์ IoT.....	88
ประวัติผู้เขียนสารนิพนธ์.....	90

สารบัญตาราง

ตาราง	หน้า
2.1 ผลลัพธ์งานวิจัยที่ทำการปรับเปลี่ยนรูปแบบของ SVM.....	6
2.2 อุณหภูมิรายจากงานวิจัย.....	12
3.1 ตัวอย่างข้อมูลหลังทำการคำนวณค่าเฉลี่ย.....	23
4.1 Diagram การเก็บข้อมูลภายในงานวิจัย.....	29
4.2 ตัวอย่างเซตข้อมูลใน 1 ชุดก่อนนำมาแปลงค่า	30
4.3 การบ่งบอกค่าของประสิทธิภาพของ AUC.....	44
4.4 ค่าการทำงานของ Accuracy, Precision, Recall, F1-Score และ AUC ในแต่ละช่วง การทำงานของ Epoch (แสดงค่าอยู่ ทศนิยม 3 ตำแหน่ง).....	46
4.5 Classification Report การคาดคะเนอริยาบทจากข้อมูลทดสอบผ่านค่า Precision, Recall และ F1-Score	47
4.6 การเปรียบเทียบค่า F1-Score ในแต่ละโมเดล	49
5.1 Classification Report การคาดคะเนอริยาบทจากข้อมูลทดสอบโดยใช้การเรียนรู้ที่ 20000 epochs.....	52
5.2 Classification Report การคาดคะเนอริยาบทจากข้อมูลทดสอบโดยใช้การเรียนรู้ที่ 50000 epochs.....	52

TNI

TAHITI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญรูป

รูป	หน้า
2.1	รูปแบบการทำงานของ Accelerator Sensor..... 4
2.2	การแบ่งกลุ่มในรูปแบบ Support Vector Machine..... 5
2.3	รูปแบบการทำงานของ Gyroscope Sensor 5
2.4	รูปแบบกราฟกิจกรรมทั้ง 5 รูปแบบ..... 6
2.5	Decision Tree ของหาค่าจาก Naïve Bayes Algorithm..... 7
2.6	รูปแบบการทำงานของงานวิจัยโดยใช้ CNN..... 8
2.7	รูปแบบเลเยอร์ของ FFNN ในงานวิจัย 9
2.8	รูปแบบการทำงานเก็บและส่งข้อมูลจากมือถือไปทำการสร้างโมเดล..... 9
2.9	รูปแบบการสร้างโมเดลในการในการทำนาย..... 10
2.10	ผลการทดลองในการวางเซ็นเซอร์ในบริเวณต่างตามร่างกาย..... 11
2.11	เซ็นเซอร์ต่างๆ ที่ใช้ใน IoT 13
2.12	Arduino (ซ้าย) Heartbeat Sensor (กลาง) Temperature Sensor (ขวา)..... 13
2.13	รูปแบบตัวอย่างข้อมูลดิบจากท่าทางการเดินเป็นหนึ่งในตัวอย่างทดลอง..... 14
2.14	อุปกรณ์ Arduino (ซ้าย) 1 ใน Diagram การทำงานการวัดค่า Quality (ขวา) 15
3.1	Accelerator & Gyroscope Sensor รุ่น MPU6050 17
3.2	Heartrate Sensor รุ่น Max30102 18
3.3	ESP32-WROVER-B Development Board..... 18
3.4	การเชื่อมต่อเชื่อมต่ออุปกรณ์ระหว่าง ESP-32 กับเซ็นเซอร์ 19
3.5	อุปกรณ์บอร์ด ESP32 และเซ็นเซอร์ที่ทำการเชื่อมต่อพร้อมใช้งาน 19
3.6	ตัวอย่างการติดตั้งอุปกรณ์ เซ็นเซอร์การจับการความเร่งและองศา (สีดำ) เซ็นเซอร์ วัดหัวใจและอุณหภูมิ (สีแดง) 20
3.7	การส่งข้อมูลจากอุปกรณ์ IoT มายัง Server ผ่าน Restful API..... 21
3.8	ตัวอย่างข้อมูลดิบก่อนทำการคำนวณค่าเฉลี่ย 22
3.9	รูปแบบของ FFNN บอลสีฟ้า (Input Layer) บอลสีเขียว (Hidden Layer) บอลสีแดง (Output Layer) 22
3.10	ข้อมูลการทำงานของโมเดลในแต่ละชั้น 23
3.11	Diagram การทำงานของโมเดล 24

สารบัญรูป (ต่อ)

รูป	หน้า
3.12 รูปแบบการทำงานของ Sigmoid และ ReLU.....	25
3.13 Diagram การเก็บข้อมูลภายในงานวิจัย.....	26
3.14 การติดตั้งอุปกรณ์บนร่างกายผู้ใช้.....	27
3.15 รูปแบบตัวอย่างการเก็บข้อมูลท่าเดิน.....	27
3.16 รูปแบบตัวอย่างการเก็บข้อมูลทำนั่ง.....	27
3.17 รูปแบบตัวอย่างการเก็บข้อมูลทำยืน.....	28
3.18 รูปแบบตัวอย่างการเก็บข้อมูลทำนอน.....	28
4.1 ค่าความเร่งองศาในตำแหน่งของแกน X ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	32
4.2 ค่าความเร่งองศาในตำแหน่งของแกน Y ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	33
4.3 ค่าความเร่งองศาในตำแหน่งของแกน Z ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	34
4.4 ค่าองศาในตำแหน่งของแกน X ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	36
4.5 ค่าองศาในตำแหน่งของแกน Y ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	37
4.6 ค่าองศาในตำแหน่งของแกน Z ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	38
4.7 ค่าอัตราการเต้นของหัวใจ (ในรูปแบบ IR) ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	40
4.8 ค่าอุณหภูมิของร่างกาย (องศาเซลเซียส) ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน.....	42
4.9 ค่าการทำงานของ Accuracy, Precision, Recall, F1-Score, และ AUC ในแต่ละช่วงการทำงานของ Epoch.....	45
4.10 Classification Report การคาดคะเนอริบาบทจากข้อมูลทดสอบผ่าน SVM บน WEKA.....	48
4.11 Classification Report การคาดคะเนอริบาบทจากข้อมูลทดสอบผ่าน NaiveBayes บน WEKA.....	48

สารบัญรูป (ต่อ)

รูป	หน้า
5.1 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบผ่าน Random Forest บน WEKA	50
5.2 รูปแบบการทำงานบางส่วนของ Random Forest.....	51
5.3 ESP 32 GPS NEO-6M SMA LORA 32 18650 ที่มีใส่เซ็นเซอร์ GPS เข้ามาใน ตัวบอร์ด	53
5.4 Electrocardiogram (ECG) Heart Rate Monitor Kit AD8232	54
ก.1 รูปแบบการทำงานของ Accelerator Sensor.....	62
ก.2 การแบ่งกลุ่มในรูปแบบ Support Vector Machine.....	62
ก.3 รูปแบบการทำงานของ Gyroscope Sensor	63
ก.4 เซ็นเซอร์ต่างๆ ที่ใช้บน IoTs	63
ก.5 Accelerator & Gyroscope Sensor รุ่น MPU6050	64
ก.6 Heartrate Sensor รุ่น Max30102	64
ก.7 ESP32-WROVER-B Development Board.....	64
ก.8 รูปแบบการทำงานของ Sigmoid และ ReLU.....	65
ก.9 ESP 32 GPS NEO-6M SMA LORA 32 18650 ที่มีใส่เซ็นเซอร์ GPS เข้ามาในตัวบอร์ด.....	65
ก.10 Electrocardiogram (ECG) Heart Rate Monitor Kit AD8232	66

TNI

TAHTI - NICHI INSTITUTE OF TECHNOLOGY

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อันเนื่องจากสังคมไทยในปัจจุบันนี้ผู้คนที่เป็นพนักงานออฟฟิศส่วนมากนั้นจะทำงานติดตลอดเวลา รวมไปถึงพฤติกรรมที่ทำงานในลักษณะท่าทางที่เป็นการทำกิจกรรมแบบซ้ำๆ ทำให้พนักงานออฟฟิศส่วนมากนั้นจะพบเจอกับอาการที่เรียกว่า ออฟฟิศซินโดรม (Office Syndrome)

ออฟฟิศซินโดรม (Office Syndrome) เป็นอาการการปวดกล้ามเนื้ออันเนื่องมาจากการใช้กล้ามเนื้อมัดเดิมซ้ำเป็นเวลานานต่อเนื่องหรือเป็นเวลานานๆ โดยที่ไม่ได้ขยับเขยื้อนหรือปรับเปลี่ยนอิริยาบถทำให้เกิดอาการปวดเรื้อรังจากการเส้นประสาทส่วนปลายนั้นถูกกดทับอย่างต่อเนื่องเป็นเวลานานๆ โดยอาการการปวดเหล่านี้สามารถเกิดได้เกือบทั่วร่างกาย นอกจากอาการปวดตามร่างกายแล้ว บางครั้งอาจมีผลกระทบไปถึงอวัยวะบางส่วนอย่างเช่น ดวงตา หรือ หู เป็นต้น



รูปที่ 1.1 อาการปวดหลังจากออฟฟิศซินโดรม

โดยที่ออฟฟิศซินโดรมนั้น ส่วนใหญ่จะเกิดจากพฤติกรรมของคนทำงานเป็นส่วนใหญ่ โดยที่พฤติกรรมเหล่านั้นอาจเกิดมาจากความเคยชินในการทำงาน การตั้งใจในการทำงานอย่างต่อเนื่อง จนบางครั้งทำให้ละเลยหรือไม่ได้สนใจในการเปลี่ยนอิริยาบถในการทำงาน จึงทำให้เห็นว่าในสังคมคนไทยนั้น มักจะมีคนที่ป่วยหรือได้รับผลกระทบในออฟฟิศซินโดรมเป็นจำนวนมากแล้วมีแนวโน้มที่จะเพิ่มขึ้นในทุกๆ ปี

แม้ว่าในปัจจุบันยังไม่มีอุปกรณ์ติดตามตัวที่สามารถบ่งบอกลักษณะหรือกิจกรรมของผู้สวมใส่ได้ แต่ยังมีอุปกรณ์อื่นๆ ที่ใช้ในการตรวจจับลักษณะท่าทางของผู้สวมใส่อยู่ ตัวอย่างเช่น นาฬิกาอัจฉริยะ หรือ สมาร์ทวอตช์ (Smart Watch) ที่สามารถตรวจจับการออกกำลังกาย รวมไปถึงการนับจำนวนก้าวเดินหรือการเดินขึ้นบันไดได้ เป็นต้น แต่ก็ยังมีความแม่นยำหรือความครอบคลุมที่ไม่ครบถ้วนเสมอไป ถึงกระนั้นก็ทำให้เห็นว่า ภายในตัวอุปกรณ์ที่ประกอบไปด้วย ตัวเซ็นเซอร์ (Sensor) ภายในนั้น มีความสามารถที่จะนำมาประยุกต์ใช้ ให้เข้ากับอุปกรณ์ประเภทพกพาหรือติดตามตัวได้

โดยที่การรับมือกับบ็อพฟิสิกซ์ไดรมนั้น จะต้องเริ่มจากการปรับเปลี่ยนพฤติกรรมการใช้ชีวิต ในช่วงของเวลาการทำงาน อย่างที่กล่าวไปในข้างต้นในส่วนของนาฬิกาอัจฉริยะนั้น ยังไม่สามารถตรวจจับอิริยาบถในการใช้ชีวิตประจำวันของผู้สวมใส่ได้อย่างครอบคลุมมากนัก แต่เนื่องจากเทคโนโลยีทุกวันนี้ได้มีการพัฒนามาตลอดจนมีสิ่งที่เรียกว่า IoT (Internet of Thing) หรือ อินเทอร์เน็ตของสรรพสิ่ง ที่ทำให้สามารถพัฒนาอุปกรณ์ได้เพิ่มเติมมากมาย โดยที่สามารถดัดแปลง แก๊ซ หรือ พัฒนาอุปกรณ์ขึ้นมาใหม่ได้โดยที่สามารถใช้เพื่อแก๊ซ ในปัญหาต่างๆ ผ่านตัวเซ็นเซอร์มากมายและเครือข่ายอินเทอร์เน็ต โดยที่เทคโนโลยีเซ็นเซอร์ใน IoT นั้นยังมีไปใช้งานอยู่ในอุปกรณ์มากมาย อาทิเช่น แท็บเล็ต (Tablet) หรือสมาร์ทโฟน (Smartphone) เป็นต้น

1.2 วัตถุประสงค์ของการวิจัย

จุดประสงค์ในงานวิจัยชิ้นนี้นั้น จะทำการพัฒนาซอฟต์แวร์ (Software) และอุปกรณ์ให้ทำงานร่วมกัน และทำการใช้ข้อมูลที่ได้รับมาจากอุปกรณ์นั้นเข้าทำการประมวลผล สร้างโมเดล (Model)

1.2.1 ในส่วนของโมเดลนั้น จะพัฒนาโมเดลเพื่อใช้สำหรับทำการคาดการณ์รูปแบบกิจกรรมของผู้สวมใส่อุปกรณ์ที่กำลังทำอยู่ โดยที่โมเดลนั้นต้องมีความแม่นยำมากกว่า 80%

1.2.2 โดยในส่วนของอุปกรณ์ IoT นั้น จะพัฒนาเพื่อใช้ในการเก็บข้อมูลแทนการใช้อุปกรณ์พกพาอย่างมือถือ เพื่อใช้ในการพัฒนาโมเดลเสร็จโดยสามารถที่สามารถปรับเปลี่ยนการใช้งานของอุปกรณ์ได้อย่างอิสระ

1.2.3 รูปแบบกิจกรรมที่จะใช้ในการวิเคราะห์และคาดการณ์นั้นจะประกอบไปด้วย ทำยืน ทำนั่ง ทำเดิน และ ทำนอน โดยจะอิงท่าทางจากการใช้ชีวิตประจำ

1.3 ความสำคัญของการวิจัย

เนื่องจากการพัฒนารูปแบบโมเดล ที่สามารถระบุท่าทางหรือกิจกรรมของผู้สวมใส่ที่มีความแม่นยำและถูกต้องออกมาได้นั้น จะเป็นแนวทางเพื่อสามารถต่อยอดหรือพัฒนาอุปกรณ์ต่างๆ ได้อีกมากมายซึ่งสามารถ ที่นำเข้ากับอุปกรณ์มากมายในอนาคต ไม่ใช่เพียงแค่ สมาร์ทวอตช์ หรือ อุปกรณ์ติดตาม แต่อาจนำไปเข้าไปประยุกต์เข้ากับอุปกรณ์อีกมากมาย เช่น ขาเทียม หรือ แขนเทียม เป็นต้น ที่

ยกตัวอย่างอุปกรณ์ดังกล่าวขึ้นมานั้น จะเห็นได้ว่าอุปกรณ์สองชนิดนั้นจะให้ผลการทำงานที่แตกต่างจากขาหรือแขนมนุษย์ ดังนั้นการพัฒนาโมเดลของอุปกรณ์ก็จะแตกต่างกันไป แต่โมเดลรูปแบบจากงานวิจัยขึ้นยังคงสามารถที่เป็นแนวทางในการปรับปรุงหรือแก้ไขให้สามารถที่จะทำการพัฒนาให้เข้าอุปกรณ์ดังกล่าวได้ในอนาคต เป็นต้น

1.4 ขอบเขตของการวิจัย

ในงานวิจัยจะกำหนดขอบเขตรูปแบบกิจกรรมที่จะใช้ในการคาดคะเนโดยมีรูปแบบกิจกรรมที่เป็นการนั่ง, ยืน, เดิน, นอนราบ จากนั้นจะทำการวิเคราะห์ข้อมูลที่รับเข้ามาโดยประกอบไปด้วย อัตราการเต้นหัวใจ, อุณหภูมิร่างกาย, การเคลื่อนไหว และรวมไปถึง ความดันโลหิต มาใช้ในการคาดคะเนกิจกรรมในช่วงต้นดังกล่าวเพื่อหาผลลัพธ์ความแม่นยำให้มีความถูกต้องมากที่สุด โดยในส่วนนี้จะกำหนดขอบเขตอายุของผู้ทดลองในงานวิจัยขึ้นนี้อยู่ในช่วง 18-40 ปี โดยที่ขณะที่ผู้ทำการสวมใส่หรือพกพาอุปกรณ์นั้นต้องอยู่ในบริเวณอาณาเขตที่สามารถเข้าใช้งานอินเทอร์เน็ตในการติดต่อได้

บทที่ 2

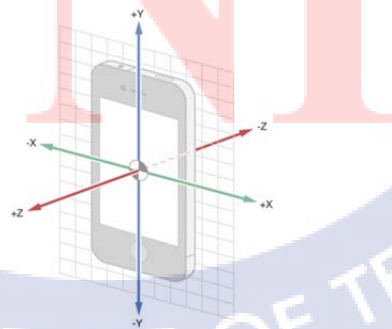
วรรณกรรมที่เกี่ยวข้อง

จากหัวข้องานวิจัยนี้ จะทำการแยกหัวข้องานวิจัยที่เค้าโครงหรือส่วนที่เกี่ยวข้องออกมาเป็นสามกลุ่มหลักๆ โดยแยกออกมาเป็นในส่วนของ

- 2.1 วิธีการคัดกรองแยกแยะข้อมูลในการคาดการณ์ถึงลักษณะท่าทางของกิจกรรม
- 2.2 องค์ประกอบความสัมพันธ์ระหว่างข้อมูลและอุปกรณ์
- 2.3 อุปกรณ์เครื่องมือในการใช้ตรวจจับและส่งข้อมูล

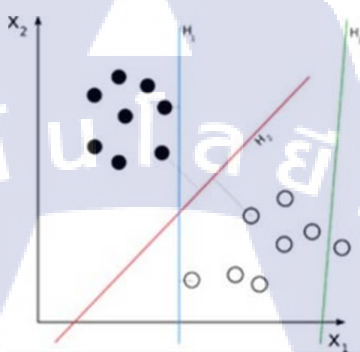
2.1 วิธีการคัดกรองแยกแยะข้อมูลในการคาดการณ์ถึงลักษณะท่าทางของกิจกรรม

จากงานวิจัยหลายชิ้นที่ทำการศึกษามานั้น พบว่ามีการใช้ข้อมูลหลากหลายเข้ามาทำการประมวลผลผ่านการคำนวณและการเรียนรู้ (Training) โดยเทคนิคหลายหลาย หนึ่งในวิธีการนั้นคือการใช้เทคนิคที่เรียกว่า โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) โดยมีงานวิจัยหนึ่ง ได้ใช้เทคนิคดังกล่าวในรูปแบบของ Fully Convolutional Networks (FCN) [1] โดยที่ทำการวิเคราะห์ข้อมูลจากอุปกรณ์มือถืออพกพา จากนั้นจึงทำการแปลงข้อมูลที่ได้รับเข้ามานั้น แปลงออกมาเป็นข้อมูลเป็นในรูปแบบของเมทริกซ์ข้อมูลรูปภาพที่เก็บข้อมูลของสีภายในพิกเซล (Pixel) ออกมาในรูปแบบ 2 Dimension (2D) จากนั้นจึงนำข้อมูลดังกล่าวไปทำการประมวลผลเพื่อทำการแยกแยะลักษณะท่าทางของผู้ที่ทำการพกพาอุปกรณ์ใน 6 ลักษณะ โดยลักษณะท่าทางดังกล่าวนั้นประกอบไปด้วย การเดิน, นั่ง, ยืน, วิ่ง, และการเดินขึ้น-ลง จากนั้นเมื่อทำการรับข้อมูลเข้ามานั้นจึงทำการหาผลลัพธ์ โดยข้อมูลดังกล่าวที่นำมาใช้นั้น จะค่าที่รับมาจาก Accelerator Sensor โดยจะเป็นเซ็นเซอร์ที่มีไว้สำหรับตรวจจับลักษณะการเคลื่อนไหวซึ่งอยู่ภายในตัวอุปกรณ์สื่อสาร โดยการทำงานของเซ็นเซอร์นั้นจะให้ค่าออกเป็นค่าในมุมของแกน



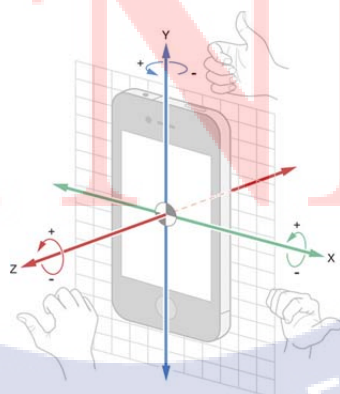
รูปที่ 2.1 รูปแบบการทำงานของ Accelerator Sensor

ประกอบไปด้วย แกนแนวนอน (X), แกนแนวตั้ง (Y), และแกนแนวลึก (Z) โดยค่าเหล่านี้จะแสดงถึงรูปแบบของทิศทางอุปกรณ์ได้อย่างแม่นยำ แต่จากงานวิจัยนี้จะใช้เพียง 2 ค่าที่มาจากเซ็นเซอร์ประกอบไปด้วย แกนแนวนอน (X) และ แกนแนวตั้ง (Y) ในการระบุรูปแบบกิจกรรมที่ผู้พกพาผ่านทางเทคนิค FCN หรือจะเป็นการใช้เทคนิค Support Vector Machine (SVM) [2] ซึ่งเป็นเทคนิคในการตัดแยก จำแนก หรือการแบ่งกลุ่มข้อมูลโดยใช้เส้นในการแบ่งกลุ่มที่มีค่าในการแบ่งกลุ่มที่ให้ผลลัพธ์ที่ดีและแม่นยำที่สุด (Hyperplane)



รูปที่ 2.2 การแบ่งกลุ่มในรูปแบบ Support Vector Machine

โดยการแบ่งกลุ่มจากงานวิจัยชิ้นนี้จะใช้ข้อมูลที่น่าเข้ามาในการคิดคำนวณมาจากอุปกรณ์พกพาเมื่อถือเช่นเดียวกับงานวิจัยในช่วงต้นที่น่าเสนอไป โดยมี Accelerator Sensor และ Gyroscope Sensor เพิ่มเติมขึ้นมา โดยการทำงานของ Gyroscope Sensor หรือ การตรวจจับทิศทางการหมุนของแกนแต่ละอันและจะทำการตรวจจับทิศทางของแต่ละแกนว่ามีทิศทางไปในทางไหน เพื่อทำการตรวจจับกิริยาบถของผู้พกพาไปพร้อมกับเซ็นเซอร์ตรวจจับทิศทาง



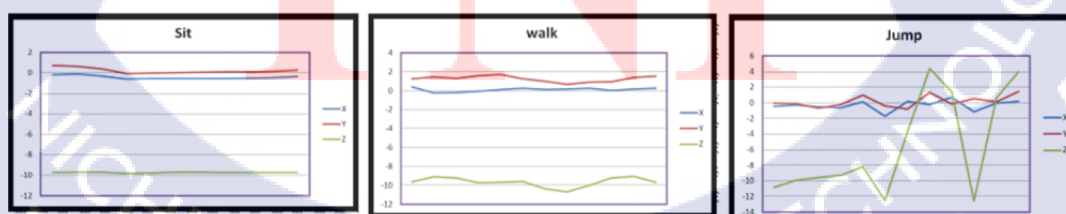
รูปที่ 2.3 รูปแบบการทำงานของ Gyroscope Sensor

จากการใช้งานในอุปกรณ์รวมไปถึงข้อมูลที่ทำกรเก็บมานั้น จะนำเข้าสู่กระบวนการคำนวณในการแบ่งกลุ่มโดยใช้วิธีการ Bagging Algorithm หรือ Bootstrap Aggregation โดยทำการสุ่มข้อมูลจากกลุ่มตัวอย่างที่ทำกรเก็บมาจากข้างต้น แล้วทำการศึกษาหาโมเดล จากนั้นจึงการสุ่มข้อมูลตัวอย่างจากเดิมอีกเรื่อยๆ จนกว่าจะถึงลิมิตของข้อมูลที่จะไม่ทำการสุ่มอีก โดยจากงานวิจัยนี้จะทำการสุ่มไปในแต่ละครั้งอยู่ที่ 100 ครั้งต่อชุดข้อมูล โดยการทดลองของงานวิจัยนี้จะแบ่งข้อมูลออกมาเป็นทั้งหมด 5 รูปแบบ โดยทำการแบ่งเปอร์เซ็นต์ของข้อมูลออกมาเป็นอัตราส่วน (ข้อมูลใช้สำหรับการเรียนรู้:ข้อมูลใช้ทดลอง) 50:50, 60:40, 70:30, 80:20, และ 90:10 โดยที่จะทำการวัดผลลัพธ์ผ่าน 3 องค์ประกอบสำคัญคือ ความแม่นยำ (Accuracy), ความไวในการแยกแยะ (Sensitive), และการแยกแยะหรือการแบ่งกลุ่มข้อมูล (Specificity)

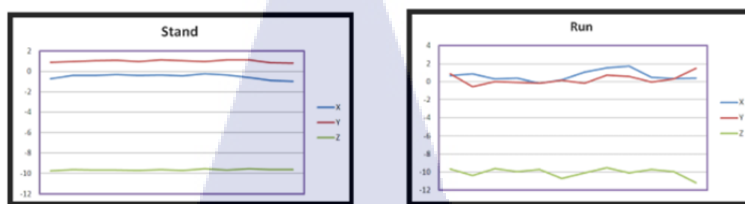
ตารางที่ 2.1 ผลลัพธ์งานวิจัยที่ทำการปรับเปลี่ยนรูปแบบของ SVM

Comparison of Training Data and Test Data	Performance Measurement Ensemble SVM		
	Accuracy	Sensitivity	Specificity
50 : 50	99.4	99.2	99.6
60 : 40	99	98.9	99
70 : 30	99.2	99.6	98.7
80 : 20	99.6	100	99.2
90 : 10	100	100	100

แต่การใช้งานวิจัย [3] ถัดมานั้น ก็ยังมีการใช้วิธีการจัดกลุ่มในอีกรูปแบบเพื่อทำการแยกแยะรูปแบบของกิจกรรมโดยที่ใช้ค่าจาก Accelerator Sensor บนอุปกรณ์มือถือพกพา โดยที่จะใช้การแบ่งกลุ่มในรูปแบบของ Naïve Bayes Algorithm โดยที่จะมีการจัดค่าแล้วรูปแบบค่าของกิจกรรมต่างๆ ที่ทำการเก็บออกมา แล้วทำการแบ่งรูปแบบออกมา

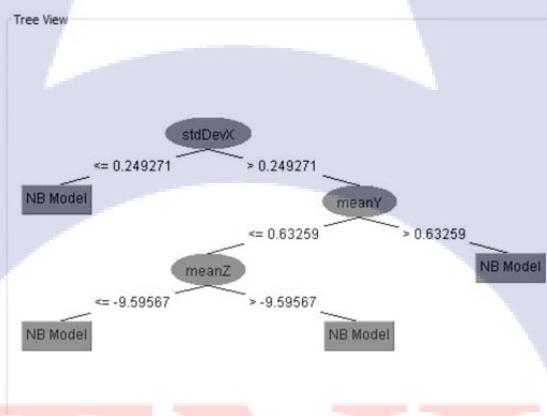


รูปที่ 2.4 รูปแบบกราฟกิจกรรมทั้ง 5 รูปแบบ



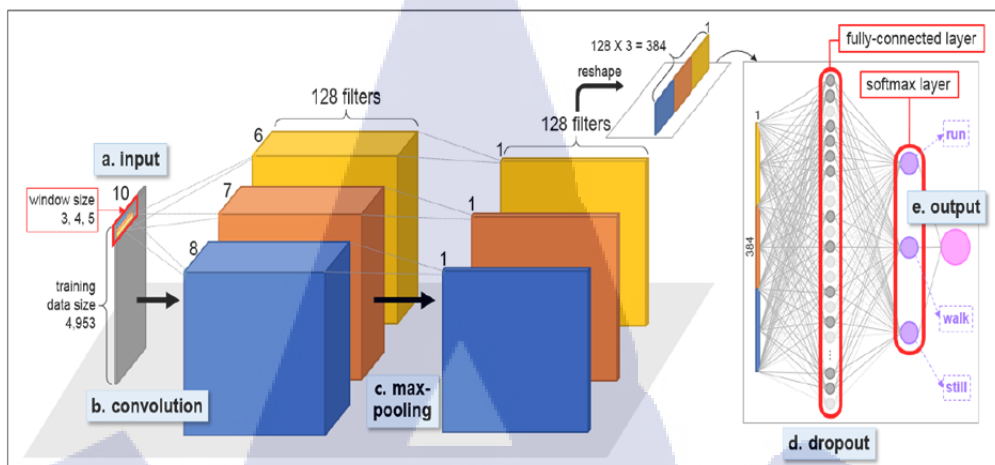
รูปที่ 2.4 รูปแบบกราฟกิจกรรมทั้ง 5 รูปแบบ (ต่อ)

จากนั้นจึงเอาข้อมูลเหล่านี้สู่ระบบในการแบ่งกลุ่มโดยการจัดทำข้อมูลดังกล่าวให้เป็นข้อมูลใช้สำหรับการเรียนรู้ จากนั้นเมื่อมีการส่งค่าที่มาจากเซ็นเซอร์ที่มาจากค่าแกนแนวนอน (X), แนวนอน (Y), และแนวลึก (Z) มา ระบบจะทำการคำนวณหาค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) ออกมา แล้วจากนั้นจึงนำเข้าสู่ Naïve Bayes Algorithm เพื่อใช้ในการหารูปแบบของกิจกรรมต่างๆ ออกมา ในรูปแบบที่คล้ายกับการตัดสินใจความน่าจะเป็นตามค่าที่ทำการคำนวณออกมา หรือเรียกว่า Decision Tree



รูปที่ 2.5 Decision Tree ของหาค่าจาก Naïve Bayes Algorithm

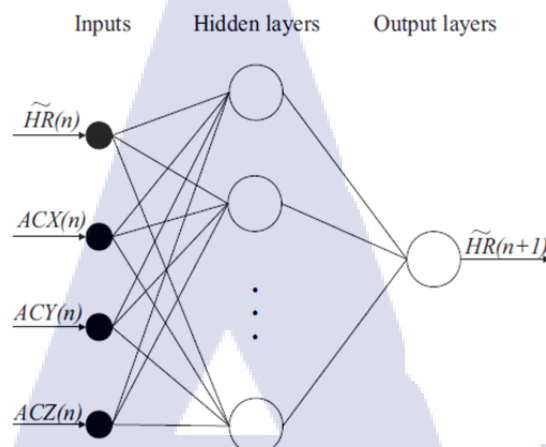
แต่จากงานวิจัยในขั้นถัดไปได้มีการปรับเปลี่ยนวิธีการใช้งาน Neural Network โดยทำการเปลี่ยนรูปแบบของ CNN เป็นในรูปแบบของการเพิ่ม Dimension ในการฟิลเตอร์ก่อนนำเข้าสู่ระบบหรือเรียกวิธีการในงานวิจัย [4] ดังกล่าวว่า Deep Convolution Neural Network (DCNN)



รูปที่ 2.6 รูปแบบการทำงานของงานวิจัยโดยใช้ CNN

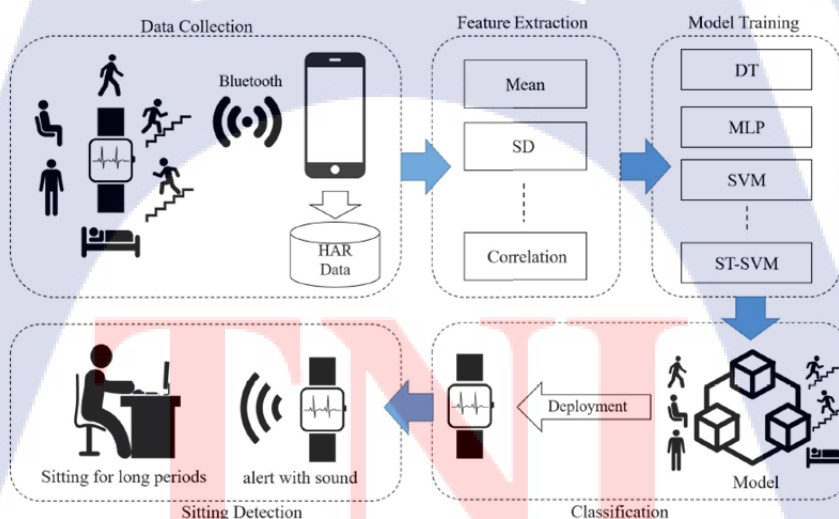
โดยที่ปกติแล้วการทำงานของ CNN ทั่วไปจะเป็นการนำเอาข้อมูลจากส่วน Input (a) เข้าไปตรงๆ ไปยังส่วนของ Dropout (d) แต่การขั้นตอนข้างต้นนั้นจะทำการแยกเอาข้อมูลออกมาก่อนเพื่อนำข้อมูลส่วนที่จำเป็นที่สุดเข้ามาทำการใช้แยกแยะ โดยจะทำการแยกแยะข้อมูลจาก Convolution (b) จากนั้นจึงทำการแปลงข้อมูลที่จำเป็นผ่าน Max-Pooling (c) จะได้ออกมาเป็นข้อมูลที่เอาเข้าสู่ระบบของการแยกแยะอีกที โดยที่ข้อมูลที่ได้รับมานั้นจะเป็นข้อมูลในส่วนของ Accelerator Sensor ที่รับมาจากอุปกรณ์สมาร์ตโฟน (Smart Phone) อีกที

จากงานวิจัยดังกล่าวข้างต้นที่กล่าวมานั้นจะเห็นว่าเป็นการใช้เซ็นเซอร์ต่างๆ เข้ามาช่วยในการคาดคะเนกิจกรรมที่ผู้สวมใส่หรือพกพานั้นใช้อยู่ ซึ่งงานวิจัยในขั้นถัดไปนั้นก็มีสิ่งบ่งชี้อื่นอีกว่าในการคาดคะเน โดยงานวิจัยนี้ [5] ได้ทำการใช้ Heart Rate (อัตราการเต้นของหัวใจ) โดยพัฒนามาจากงานวิจัยในก่อนหน้านี้ควบคู่ไปกับการใช้ Accelerator Sensor ในการเป็นข้อมูลที่ใช้ในการวิเคราะห์เข้าไปในระบบของ Feedforward Neural Network (FFNN) โดยที่ใช้ในการส่งเข้าไปโมเดลนั้นจะใช้ค่า อัตราการเต้นหัวใจ และค่าแกนแนวนอน แนวตั้งและความลึก โดยทำการเก็บข้อมูลเป็นเวลา 90 นาที แล้วทำการแบ่งข้อมูลออกมาเป็นสองชุดเพื่อใช้ในการเก็บข้อมูลสำหรับการเรียนของระบบและใช้สำหรับทดสอบในส่วนหนึ่งของโมเดล



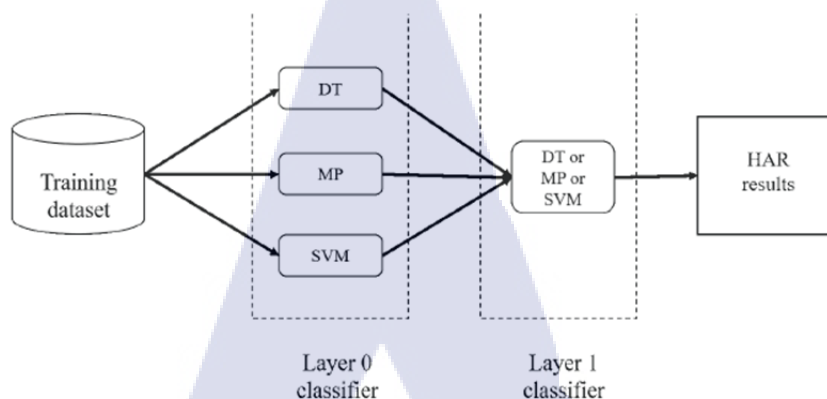
รูปที่ 2.7 รูปแบบเลเยอร์ของ FFNN ในงานวิจัย

ในงานวิจัยถัดมา [6] จะเป็นเรื่องของการใช้เซ็นเซอร์ภายใน Smartwatch ซึ่งคือ Accelerator Sensor และ Gyroscope Sensor ในการตรวจจับการเคลื่อนไหวของมือ โดยที่นำเอาค่าทั้งสองอันนั้นมาทำการเข้าสู่ระบบการเทรนนิ่งก่อน จากนั้นถึงสู่ระบบการคำนวณ



รูปที่ 2.8 รูปแบบการทำงานเก็บและส่งข้อมูลจากมือถือไปทำการสร้างโมเดล

โดยที่วิธีการที่นำเข้าใช้ในการคำนวณนั้นจะมีทั้งหมด 7 แบบ ประกอบไปด้วย Decision Tree, Neural Network, SVM, Majority Voting, Stacking (J48, MLP, SVM) บนเครื่องมือ WEKA และ จะทำการเลือกเอาค่าที่มีผลลัพธ์ดีที่สุดมาเป็นค่าสุดท้าย



รูปที่ 2.9 รูปแบบการสร้างโมเดลในการในการทำนาย

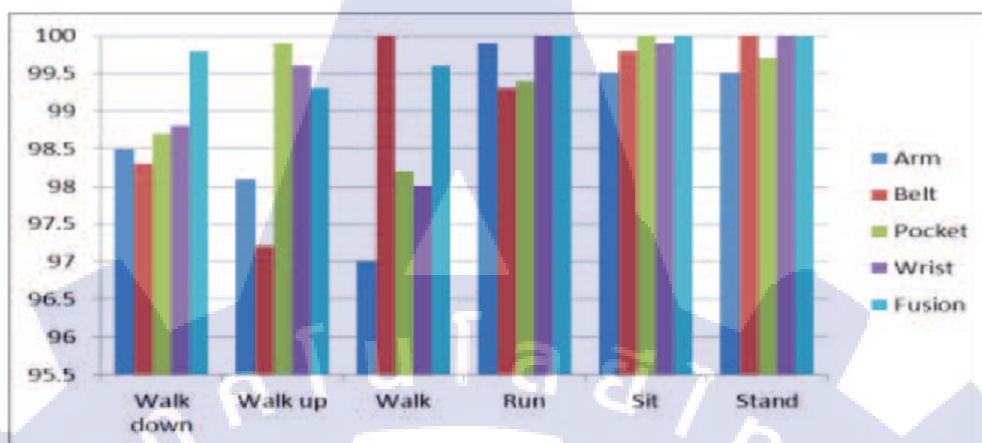
2.2 องค์ประกอบความสัมพันธ์ระหว่างข้อมูลและอุปกรณ์

จากงานวิจัยส่วนใหญ่ นั้น จะเห็นได้ว่าจะใช้เครื่องมือหรืออุปกรณ์ที่มีเซ็นเซอร์อย่างเช่น Accelerator Sensor และ Gyroscope Sensor ที่ประกอบอยู่ในงานวิจัยหลายชิ้น เนื่องมาจากการเคลื่อนไหวร่างกายของมนุษย์นั้นสามารถบ่งบอกท่าทางหรือกิจกรรมที่ทำอยู่ในขณะนั้นได้ โดยทำการจับท่าทางในมุมมองต่างๆ ที่ทำการเคลื่อนไหวอยู่ เป็นต้น

เหตุผลหลักๆ ที่จะเห็นได้ว่า Accelerator Sensor นั้นได้รับความนิยมที่นำไปใช้ในงานวิจัย เป็นเนื่องมาจากที่ตัวเซ็นเซอร์นั้นสามารถที่จะบ่งบอกจังหวะการเดินหรือการวิ่งได้เพื่อใช้ในการนับก้าวเดินผ่านการตรวจจับการสั่นไหวเมื่อผู้สวมใส่อุปกรณ์ รวมไปถึงการตรวจจับหารูปแบบในรูปแบบกิจกรรมต่างๆ ได้ โดยข้อมูลที่ได้รับมานั้นจะเป็นข้อมูลที่แม่นยำและได้รับค่ามาได้อย่างรวดเร็ว แต่ถึงอย่างไรก็ตามการประเมินด้วยเพียง Accelerator Sensor นั้นกลับยังไม่เพียงพอหรือคาดคะเนที่จะทำการหารูปแบบกิจกรรมแบบหนักได้ ตัวอย่างเช่น การปั่นจักรยานในที่ร่ม หรือการเวทเทรนนิ่ง เป็นต้น ทำให้การใช้งานควบคู่ไปกับเซ็นเซอร์ประเภทอื่นๆ นั้น จะช่วยเพิ่มศักยภาพในการช่วยคาดคะเนเพิ่มขึ้นมาได้

ตัวอย่างเช่นงานวิจัยในชั้นถัดไปนั้นได้มีการเซ็นเซอร์มากกว่าหนึ่งตัวในการทำการคาดคะเนกิจกรรมของผู้ใช้งาน โดยงานวิจัยดังกล่าว [7] ได้ทำการใช้ Accelerator Sensor มากกว่า 1 เซ็นเซอร์ และทำการติดตั้งตำแหน่งของเซ็นเซอร์ในจุดที่แตกต่างกันบนตามร่างกายของผู้ใช้งาน โดยจุดที่ทำการติดตั้งเซ็นเซอร์นั้นจะประกอบไปด้วย ข้อมือ, ต้นแขน, เข็มขัด (เอว), และกระเป๋ากางเกง (ต้นขา) เนื่องจากงานวิจัยดังกล่าวมีทฤษฎีที่มีแนวความคิดว่าการวางตำแหน่งของเซ็นเซอร์นั้นมีผลต่อการหาค่าเพื่อมาทำการวิเคราะห์ในการหารูปแบบของกิจกรรมที่ผู้สวมใส่อุปกรณ์นั้นได้ปฏิบัติอยู่ ณ ขณะนั้น โดยได้ใช้ค่าที่อ่านจากอุปกรณ์มือถือพกพา 4 เครื่องในตำแหน่งต่างๆ เข้าด้วยกัน จากนั้นจึงทำการส่งเข้าสู่ระบบแล้วทำการใช้เทคนิคที่เรียกว่า Butterworth Low Pass Filter ในการคัดกรองและจัดการกับ

สัญญาณที่ได้รับเข้ามา เพื่อทำการจัดการข้อมูลก่อนนำเข้ากระบวนการแยกแยะรูปแบบของกิจกรรม ในขณะนั้นอยู่ออกมา



รูปที่ 2.10 ผลการทดลองในการวางเซ็นเซอร์ในบริเวณต่างตามร่างกาย

ในงานวิจัยขึ้นถัดมาจะเป็นบ่งบอกในเรื่องของความสัมพันธ์ในส่วน of ตัวข้อมูลระกวางกัน เนื่องจากในงานวิจัยขึ้น [8] จะช่วยบ่งบอกถึงอุณหภูมิที่มีความเกี่ยวข้องไปยังการเต้นของหัวใจ โดยที่ได้มีการกล่าวถึงในส่วนของการทำกิจกรรมต่างๆ ของชีวิตประจำวัน โดยเฉพาะอย่างยิ่งกิจกรรมที่ต้องใช้กำลังหรือการออกกำลังกายนั้น จะทำให้อุณหภูมิร่างกายนั้นเปลี่ยนแปลงและยังส่งผลให้การทำงานหรือการเต้นของหัวใจนั้นทำงานเร็วขึ้นเพื่อทำการสูดเลือดหรือออกซิเจนด้วย รวมไปถึงขั้นตอนการปรับสภาวะของร่างกายด้วย เนื่องจากมีค้นพบว่าบางครั้งแล้วร่างกายมักมีอุณหภูมิร่างกายสูงขึ้นในช่วงเช้าและมีแนวโน้มที่จะลดลงเมื่อเป็นเวลาช่วงเย็นแม้ว่าอาจจะไม่มีการทำกิจกรรมใดๆ ก็ตาม เป็นต้น คล้ายกับงานวิจัยอีกชิ้น [9] ที่ทำการวัดอุณหภูมิจากส่วนจากส่วนต่างๆ ของร่างกายระหว่างทำการปั่นจักรยาน โดยทำการตรวจจับข้อมูลการเต้นหัวใจและอุณหภูมิ โดยที่ส่วนค่าของอุณหภูมิที่ทำการวัดได้มานั้นจะทำการคำนวณจากค่าเฉลี่ยมาอีกทีก่อนนำข้อมูลมาใช้ โดยที่จะการเก็บข้อมูลเป็นระยะของการทำการทำกิจกรรมทำให้เห็นถึงความแตกต่างในการวัดอุณหภูมิตามแต่ละส่วนที่มีความแตกต่างกันได้อย่างชัดเจน

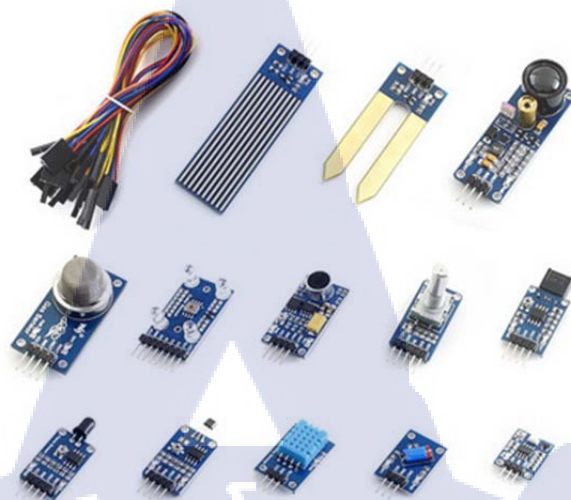
ตารางที่ 2.2 อุณหภูมิรายจากงานวิจัย

Site	Temperature (C)
Forehead	34.6
Oral	35
Upper Arm	33.5
Forearm	34
Hand	32.8
Chest	34.3
Abdomen	34
Upper Back	34
Lower Back	33.4
Colonic	36.8
Thigh	33.5
Calf	33
Foot	31.9

2.3 อุปกรณ์เครื่องมือในการใช้ตรวจจับและส่งข้อมูล

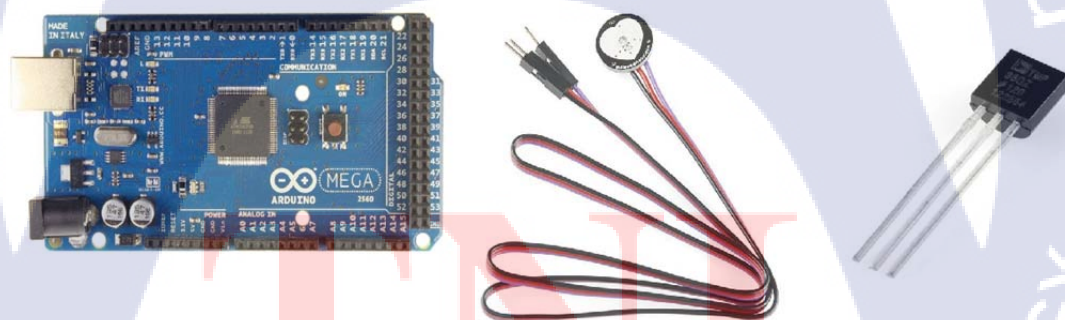
เนื่องด้วยเทคโนโลยีในปัจจุบันที่พัฒนาได้อย่างรวดเร็ว ทำให้การเก็บข้อมูลต่างๆ นั้น สามารถที่เข้าถึงได้เพียงแค่สามารถเชื่อมต่ออินเทอร์เน็ต จึงทำให้เกิดอุปกรณ์มากมายโดยพัฒนามาจากเซ็นเซอร์ที่มีอยู่แล้วในปัจจุบันหรือเป็นที่รู้จักในชื่อของ Internet of Things (IoT) โดยการพัฒนาเทคโนโลยีในส่วนนี้จะทำการพัฒนาอุปกรณ์ต่างๆ โดยที่ทำการเชื่อมต่ออินเทอร์เน็ต แล้วทำการเชื่อมต่อไปยังอุปกรณ์หรือฐานข้อมูลเพื่อทำการเก็บข้อมูลที่สนใจ เพื่อใช้ในการประมวลผลหรือทำการวิเคราะห์ในส่วนถัดไป เป็นต้น

ด้วยเทคโนโลยีด้าน IoT นั้น ทำให้การเก็บข้อมูลนั้นสามารถทำได้หลากหลายวิธีการโดยขึ้นกับข้อมูลที่จะทำการเลือกใช้ควบคู่ไปกับเซ็นเซอร์ที่ได้ใช้ เช่น การเก็บค่าความชื้นในดิน, การเก็บค่าความชื้นในอากาศ หรือการเก็บค่าความห่างระหว่างวัตถุระหว่างสองชิ้น เป็นต้น โดยที่เซ็นเซอร์บางตัวนั้น ยังมีการใช้เข้าไปยังอุปกรณ์เทคโนโลยีในปัจจุบันด้วย ตัวอย่างที่เห็นได้ชัดที่สุดนั้นก็คือ อุปกรณ์มือถือถึงพกพาหรือสมาร์ตโฟน ที่มีการใช้ Accelerator และ Gyroscope Sensor รวมไปถึง GPS Sensor ที่ใช้ในการระบุตำแหน่งใน ณ ขณะนั้นด้วย



รูปที่ 2.11 เซ็นเซอร์ต่างๆ ที่ใช้ใน IoT

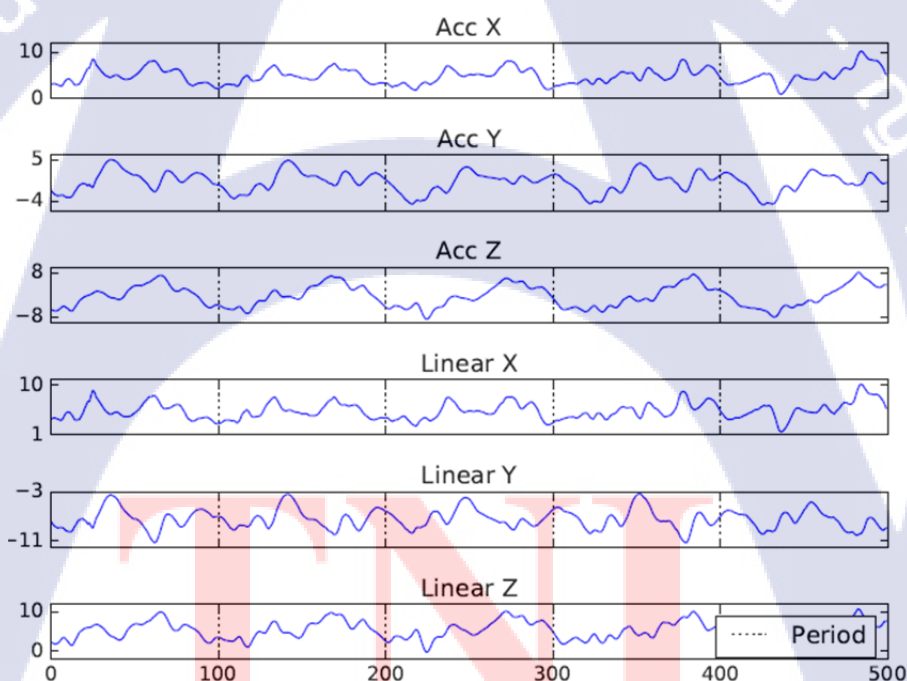
โดยงานวิจัยในขั้นถัดไปนั้น ได้ทำการสร้างอุปกรณ์ขึ้นมาโดยทำการสื่อสารส่งข้อมูลผ่านระบบไร้สายบลูทูธ (Bluetooth) ไปยังระบบปฏิบัติการ Android โดยในงานวิจัย [10] ได้ทำการใช้อุปกรณ์ Arduino ประกอบเข้ากับเซ็นเซอร์สองชนิด นั่นคือ Heartbeat Sensor และ Temperature Sensor ในการเก็บข้อมูลของร่างกายคน



รูปที่ 2.12 Arduino (ซ้าย) Heartbeat Sensor (กลาง) Temperature Sensor (ขวา)

โดยค่าที่อ่านได้จากเซ็นเซอร์อย่างเช่น Temperature Sensor นั้นจะผ่านการคำนวณเพื่อทำการเปลี่ยนค่าจากอุณหภูมิฟาเรนไฮต์ ($^{\circ}\text{F}$) ไปสู่ค่าอุณหภูมิเซลเซียส ($^{\circ}\text{C}$) โดยที่งานวิจัยชิ้นนี้จะทำการนำเอาค่าทั้งสองอันจากที่เก็บได้มานั้นไปทำการวิเคราะห์เพื่อหาค่าความสัมพันธ์ โดยใช้การตีความความเป็นไปได้ถึงสุขภาพและความเสี่ยงในส่วนของอัตราการเต้นของหัวใจของผู้ถูกตรวจสอบ คล้ายกับอุปกรณ์ทางการแพทย์ที่ใช้ในการวัดค่าการเต้นของหัวใจ

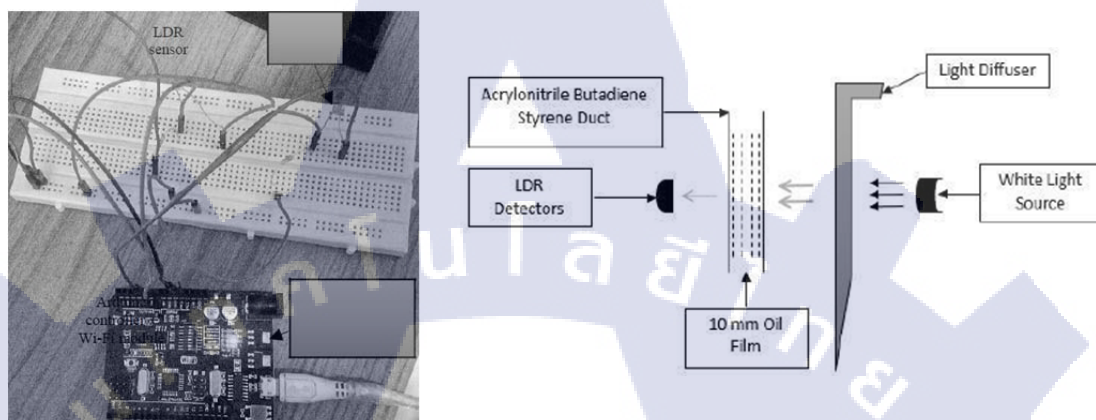
อีกหนึ่งงานวิจัยอีกชิ้นที่ได้ทำการใช้อุปกรณ์ IoT เข้ามาทำการเก็บข้อมูลผู้ใช้งานเพื่อทำการระบุถึงตัวตน [11] โดยที่งานวิจัยดังกล่าวนี้ได้ทำการใช้เซ็นเซอร์ Accelerator Sensor ที่มีอยู่ในอุปกรณ์ประเภทของสมาร์ทโฟน เพื่อใช้ทำการระบุตัวตนจากรูปแบบการเดินของผู้ใช้งานของแต่ละคน ในส่วนของงานวิจัยนี้จะไม่ได้ทำการสร้างอุปกรณ์ขึ้นมา แต่ด้วยเหตุผลที่ว่าอุปกรณ์ IoT นั้นส่วนมากจะถูกนำเอาเข้าไปใช้กับอุปกรณ์อื่นๆ มากมาย ทำให้งานวิจัยชิ้นนี้สามารถที่นำเอาความสามารถในการส่งข้อมูลรูปแบบการเดินของผู้ใช้งานแต่ละคนออกมาเพื่อทำการระบุตัวตนได้ โดยงานวิจัยนี้จะเก็บเอาข้อมูลที่ได้จาก Accelerator Sensor โดยแยกออกมาเป็น Accelerometer และ Linear Acceleration Data โดยที่ข้อมูลเหล่านี้จะถูกดึงออกมาจากเซ็นเซอร์ผ่านทางแอปพลิเคชัน (Application) ที่ผู้ค้นคว้างานวิจัยนี้ทำการพัฒนาขึ้นมา จากนั้นจึงทำการส่งข้อมูลที่เก็บได้มานั้นส่งไปหาเซิร์ฟเวอร์ผ่านระบบอินเทอร์เน็ตไร้สายบนมือถือ แล้วจึงใช้กระบวนการเรียนรู้ (Machine Learning) ในการประมวลผลในการระบุตัวตนในขั้นตอนต่อไป



รูปที่ 2.13 รูปแบบตัวอย่างข้อมูลดิบจากท่าทางการเดินเป็นหนึ่งในตัวอย่างทดลอง

หรือจะเป็นอีกงานวิจัยที่ใช้อุปกรณ์ IoT เข้ามาทำการตรวจเช็คในส่วนของคุณภาพของตัวน้ำมันเครื่องยนต์ [12] เพื่อใช้ประกอบการตัดสินใจในการเปลี่ยนถ่ายน้ำมันเครื่อง โดยพัฒนาอุปกรณ์ขึ้นมาโดยอ้างอิงจาก Spectrophotometer หรือก็อุปกรณ์ที่วัดของเหลวโดยอ้างอิงจากความสว่าง

ของแสงที่ผ่านเข้าแล้วออกมาจากตัวของเหลว โดยเริ่มแรกจะทำการเก็บตัวอย่างน้ำออกมาก่อนเพื่อทำการบ่งคุณภาพของน้ำมันเครื่อง จากนั้นจึงเริ่มการเอาข้อมูลเหล่านั้นสู่เซิร์ฟเวอร์เพื่อใช้ในการประมวลผล เมื่อดำเนินขั้นตอนดังกล่าวเสร็จเรียบร้อยแล้ว จึงจะทำการตรวจเช็คคุณภาพน้ำมันเครื่องผ่านอุปกรณ์ IoT โดยทำการปล่อยแสงผ่านของแล้วทำการตรวจจับการความสว่างของแสงมากระทบที่ LDR Detector



รูปที่ 2.14 อุปกรณ์ Arduino (ซ้าย) 1 ใน Diagram การทำงานการวัดค่า Quality (ขวา)

บทที่ 3

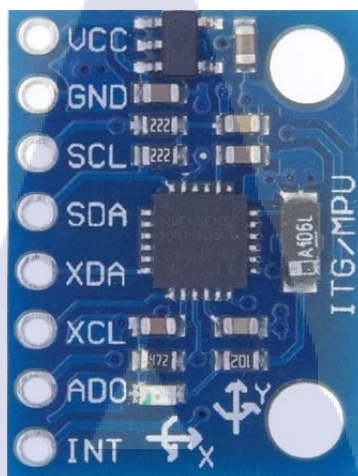
วิธีดำเนินการวิจัย

จากงานวิจัยนี้ ผู้ทำการงานวิจัยจะมุ่งเน้นไปยังการสร้างโมเดลที่ใช้ในการวิเคราะห์และคาดการณ์ความน่าจะเป็นของรูปแบบกิจกรรม ผ่านการเก็บข้อมูลการทดลองผ่านทางเทคโนโลยี IoT และแสดงผลลัพท์ออกทางหน้าจอไปยังผู้ดูแลสังเกตการณ์ โดยกล่าวถึงหัวข้อดังต่อไปนี้

- 3.1 เครื่องมือในการเก็บข้อมูลและแสดงผลการทดลองในงานวิจัย
- 3.2 วิธีการในการทำงานวิจัย
- 3.3 ลำดับขั้นตอนการดำเนินงาน
- 3.4 ข้อมูลที่นำเข้ามาใช้เข้ากับโมเดลและทำการทดลอง

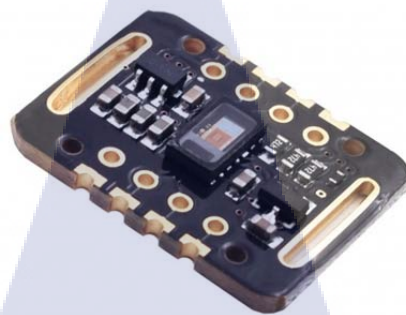
3.1 เครื่องมือในการเก็บข้อมูลและแสดงผลการทดลองในงานวิจัย

จากที่กล่าวไปข้างต้น เนื่องจากปัจจุบันนี้เทคโนโลยีในด้านของ IoT นั้นเติบโตและถูกพัฒนาอย่างต่อเนื่อง จนถูกนำไปประกอบเข้ากับเทคโนโลยีต่างๆ มากมาย โดยเฉพาะอย่างยิ่งอุปกรณ์ประเภทพกพาอย่างเช่น อุปกรณ์มือถือสมาร์ทโฟน เป็นต้น จากงานวิจัยที่ทำการค้นคว้ามาจากหัวข้อ วรรณกรรมที่เกี่ยวข้อง จะเห็นได้ว่าการใช้อุปกรณ์สมาร์ทโฟนอยู่เป็นจำนวนมาก ในการทำการรับและส่งข้อมูลออกมา เนื่องจากสมาร์โฟนนั้นมีความสามารถในการเชื่อมต่ออินเทอร์เน็ตอยู่ตลอดเวลา อีกทั้งยังมีเซ็นเซอร์ ที่สามารถตรวจจับข้อมูลได้อีก จากส่วนนี้ทำให้รับรู้ได้ว่าข้อมูลที่ใช้ในการตรวจที่จำเป็น คือ เรื่องของการเคลื่อนไหวอิริยาบถต่างๆ (Movement) จนไปถึงการเปลี่ยนแปลงอิริยาบถต่างๆ ของร่างกาย ซึ่งจากจุดนี้จะเห็นได้ว่าเซ็นเซอร์ที่จะใช้ในการตรวจจับนั้นจะเป็น Accelerator และ Gyroscope Sensor เป็นหลัก ทำให้ในงานวิจัยนี้จะทำการเลือกใช้เซ็นเซอร์สองตัวนี้เข้ามาทำงานร่วมกัน โดยเซ็นเซอร์ในส่วนของ Accelerator และ Gyroscope Sensor ที่ใช้นั้นเป็นเซ็นเซอร์รุ่น MPU6050 หรือ GY-521 โดยที่เซ็นเซอร์ชนิดนี้ถูกออกแบบมาเพื่อใช้เข้ากับอุปกรณ์พกพาต่างๆ ใช้พลังงานต่ำ ซึ่งข้อดีคืออีกอย่างของอุปกรณ์ชิ้นนี้คือการที่เซ็นเซอร์ทั้งสองแบบนี้ ถูกนำเข้ารวมกันเป็นเซ็นเซอร์ชิ้นเดียวกัน ทำให้การตรวจจับค่าในแกนแต่ละอันนั้น สามารถที่จะตรวจจับและส่งค่าไปพร้อมกันในทีเดียว ซึ่งทำให้ความแม่นยำที่ได้รับมานั้นมีประสิทธิภาพมากขึ้น โดยที่ค่าความเร่งและองศาในแต่ละแกนนั้นจะเป็นข้อมูลที่นำไปใช้เข้ากับงานวิจัยชิ้นนี้



รูปที่ 3.1 Accelerator & Gyroscope Sensor รุ่น MPU6050

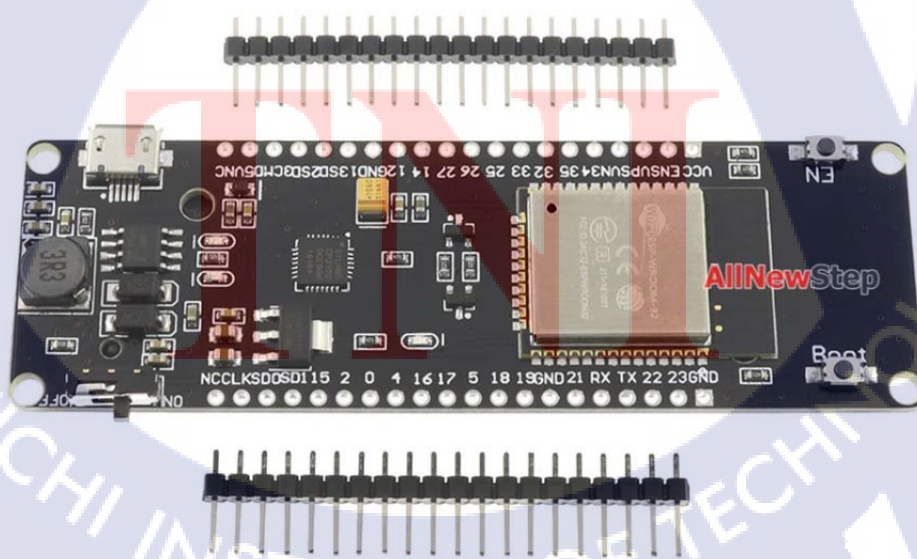
ถัดมาจะเป็นอุปกรณ์ในส่วนของ เซ็นเซอร์วัดอัตราการเต้นของหัวใจ เนื่องจากกิจกรรมทุกอย่างที่มนุษย์ได้ลงมือหรือพึงกระทำอยู่นั้นจะมีผลไปยัง ระบบการเต้นของหัวใจ และรวมไปถึงความดันโลหิต โดยที่มาตรฐานแล้วอัตราการเต้นของหัวใจนั้นจะอยู่ที่ราวๆ 60-100 ครั้งต่อนาที แต่เมื่อมีการทำกิจกรรมที่ต้องแรง จะมีการเปลี่ยนแปลงไปเป็น 130-150 ครั้งต่อนาที จึงทำให้เห็นว่ายิ่งออกกำลังกายมากเท่าไร จะยิ่งทำให้การเต้นของหัวใจนั้นทำงานหนักหรือยิ่งเต้นเร็วขึ้น เพื่อทำการสูบน้ำโลหิตในการส่งออกซิเจนไปยังส่วนต่างๆ ในร่างกายมากขึ้น ดังนั้นข้อมูลการทำงานของระบบการเต้นของหัวใจนั้น จึงเป็นอีกปัจจัยหนึ่งที่จะใช้ทำการคาดคะเนและวิเคราะห์ถึงกิจกรรมที่กระทำอยู่ ณ ขณะนั้นได้ โดยที่งานวิจัยนี้ได้ทำการเลือกใช้เซ็นเซอร์ในการวัดการเต้นของหัวใจรุ่น รุ่น MAX30102 โดยที่มีคุณสมบัตินอกจากการวัดการเต้นของหัวใจนั้นคือการวัดปริมาณระดับออกซิเจนในเลือดเพิ่มเติมเข้ามา โดยที่จุดเด่นของเซ็นเซอร์ตัวนี้คือการส่งผลลัพธ์จากการตรวจเช็คออกมาได้อย่างรวดเร็ว โดยที่การทำงานของเซ็นเซอร์นั้นจะใช้หลักการที่เรียกว่า Light-Dissolving Method โดยที่ขั้นตอนการทำงานนั้นจะเป็นการยิงแสง LED เข้าไปผ่านชั้นผิวหนังจากนั้นเซ็นเซอร์จะทำการแปลงค่าสัญญาณของของแสงที่ส่งผ่านและสะท้อนกลับมาจากเส้นเลือดนั้นออกมาเป็นค่าอินฟราเรด (IR) ออกมา ซึ่งสามารถแปลงค่าส่วนนี้ไปเป็นค่าอัตราการเต้นหัวใจได้



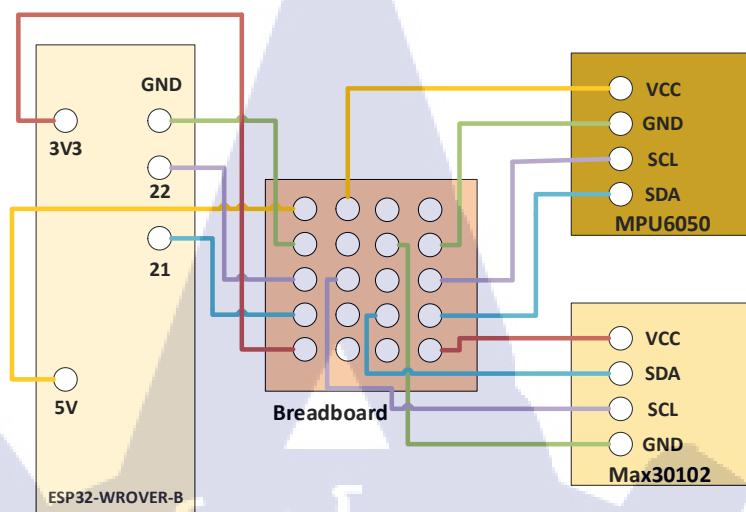
รูปที่ 3.2 Heartrate Sensor รุ่น Max30102

ต่อไปจะเป็นในส่วนของอุปกรณ์ที่ใช้เก็บข้อมูลในด้านของอุณหภูมิของร่างกาย โดยที่การทำงานของร่างกายนั้นเมื่อได้ทำกิจกรรมใดๆ เป็นเวลานานๆหรือจนไปถึงระดับช่วงหนึ่ง ร่างกายจะมีกระบวนการทำงานเปลี่ยนไปเพื่อปรับสมดุลต่างๆ ให้เข้ากับสภาพแวดล้อมภายนอก คล้ายกับการที่เราออกกำลังกายแล้วร่างกายต้องการปริมาณออกซิเจนที่มากขึ้นเป็นต้น ซึ่งเซ็นเซอร์รุ่น MAX30102 นั้นมีความสามารถในการตรวจจับอุณหภูมิซึ่งจะทำงานไปพร้อมกับการตรวจจับหัวใจไปด้วย

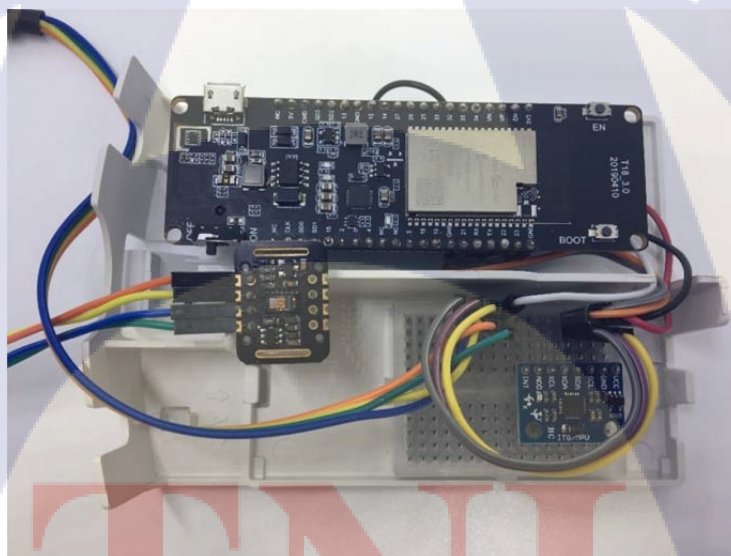
ในส่วนสุดท้ายจะเป็น Board หรือแผงวงจรหลัก ที่ใช้ในเชื่อมต่ออุปกรณ์เซ็นเซอร์ทั้งสามเข้าด้วยกันและทำการส่งข้อมูลไปยังเซิร์ฟเวอร์ (Server) เพื่อใช้ทำการเก็บข้อมูล โดยได้ทำการเลือกใช้ ESP32-WROVER-B Development Board โดยที่บอร์ดชนิดนี้มีความสามารถในการเชื่อมต่อเข้าหาอินเทอร์เน็ตผ่านในส่วนของไวไฟ (Wi-Fi) ได้ รวมไปถึงที่ความสามารถในการทำอุปกรณ์พกพาโดยภายในนั้นได้มีการเพิ่มในส่วนของท่าน Build-in เข้ามาพร้อมทั้งความสามารถในการชาร์จถ่าน



รูปที่ 3.3 ESP32-WROVER-B Development Board



รูปที่ 3.4 การเชื่อมต่อเชื่อมต่ออุปกรณ์ระหว่าง ESP-32 กับเซ็นเซอร์



รูปที่ 3.5 อุปกรณ์บอร์ด ESP32 และเซ็นเซอร์ที่ทำการเชื่อมต่อพร้อมใช้งาน

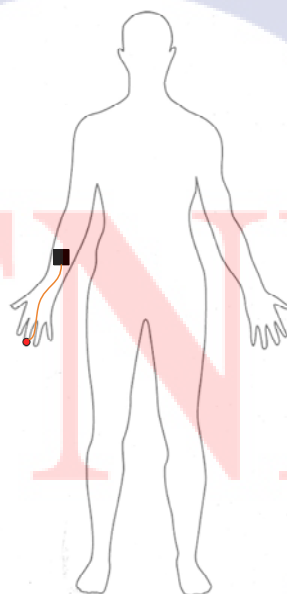
ถัดมาในส่วนของภาษาโปรแกรม (Programming Language) ที่ใช้เขียน รวมไปถึง Database ที่ใช้ในการพัฒนานั้นจะประกอบไปด้วย

- JAVA Version 8
- Python Version 3.8
- C++
- PostgreSQL Database

โดยที่จะใช้การทำงานของ Spring Boot framework (JAVA) ในการจัดการทำในส่วนของ เซิร์ฟเวอร์ (Server) โดยที่จะทำการรับส่งข้อมูลระหว่างอุปกรณ์ (C++) และทำการและทำการส่งข้อมูล และทำการบันทึกข้อมูลที่เซ็นเซอร์ส่งมานั้นเข้าสู่ดาต้าเบส (Database) และจึงนำข้อมูลเหล่านั้นไปทำการประมวลผลแล้วทำการสร้างโมเดลในการคาดคะเนข้อมูลที่น่าจะเป็นไป (Python)

3.2 วิธีการในการทำงานวิจัย

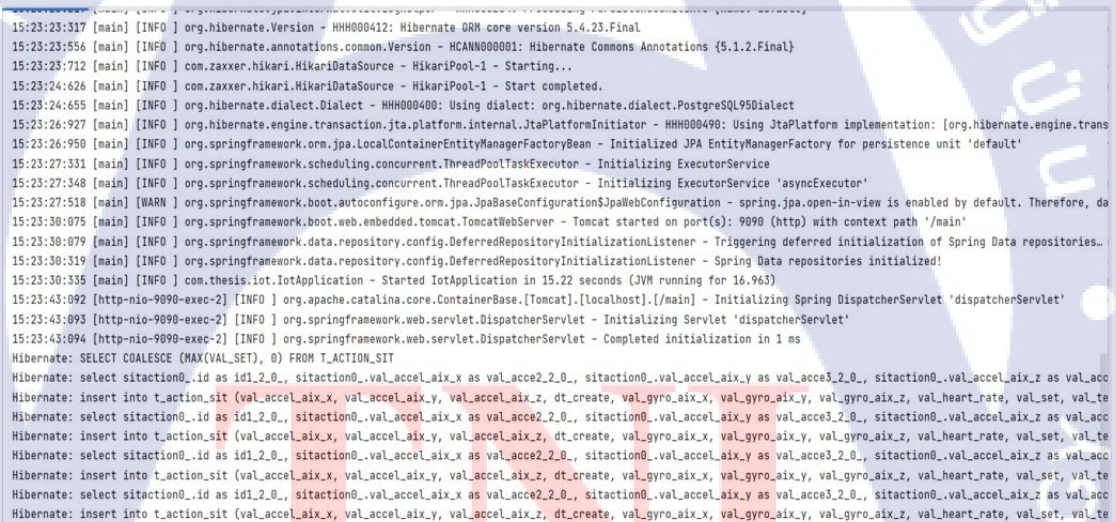
ในการทดลองวิจัยนี้ จะทำการดำเนินการโดยทำการประดิษฐ์อุปกรณ์ขึ้นมาเซ็นเซอร์ที่กล่าวไปก่อนหน้านี้โดยจะทำการติดตั้งอุปกรณ์อยู่ตามบริเวณร่างกายสองจุดคือ นิ้วมือและบริเวณข้อมือ โดยที่เซ็นเซอร์ที่ติดตั้งตรงนิ้วมือจะเป็นในส่วนของเซ็นเซอร์วัดการเต้นของหัวใจและอุณหภูมิ และในส่วนของบริเวณข้อมือนั้นจะเป็นเซ็นเซอร์ในส่วนของการวัดความแรงในการเคลื่อนที่ในความเร่งและองศาในแกน X,Y,Z เนื่องจากการวิจัยในก่อนหน้านี้ [7] ได้แสดงให้เห็นถึงตำแหน่งและการติดตั้งเซ็นเซอร์นั้น มีผลต่อการเก็บข้อมูลรวมไปถึงการงานวิจัย [13] แสดงให้เห็นว่าตำแหน่งการติดตั้งเซ็นเซอร์การวัดค่าการเต้นของหัวใจบริเวณข้อมือที่ข้อมือนั้น อาจทำให้การทำงานของเซ็นเซอร์นั้นให้ผลลัพธ์ที่คลาดเคลื่อนไปได้ ดังนั้นงานวิจัยชิ้นนี้จึงเลือกที่ใช้ตำแหน่งนิ้วมือในการเก็บข้อมูล โดยอ้างอิงจากการทำงานของอุปกรณ์การวัดระดับออกซิเจนและการเต้นของหัวใจของทางการแพทย์ ดังนั้นที่ตัวอุปกรณ์ที่ใช้ในการทำงานวิจัยนั้นจะทำการติดตั้งที่นิ้วมือและบริเวณข้อมือ



รูปที่ 3.6 ตัวอย่างการติดตั้งอุปกรณ์ เซ็นเซอร์การจับการความเร่งและองศา (สีดำ) เซ็นเซอร์วัดหัวใจและอุณหภูมิ (สีแดง)

จากนั้นจะทำการเก็บข้อมูลเหล่านั้นเข้าสู่ดาต้าเบส (Database) แล้วจึงนำข้อมูลเหล่านั้นแบ่งออกเป็นอัตราส่วน 70:30 โดยข้อมูล 70% นั้นจะเป็นข้อมูลที่ใช้ในการให้ระบบทำการเรียนรู้หรือเทรนนิ่ง และ อีก 30% นั้นจะเป็นข้อมูลสำหรับในการทดลองหรือทดสอบระบบของโมเดล โดยรูปแบบการทดลองนั้นจะเป็นการใช้ Neural Network ในรูปแบบของ Feed Forward Neural Network (FFNN) โดยใช้ข้อมูลที่ได้รับมาจากเซ็นเซอร์ซึ่งจะประกอบไปด้วยไปด้วยข้อมูลดังนี้

- ค่าความองศาเร่งในตำแหน่งของแกน X (รูปที่ 2.1)
- ค่าความองศาเร่งในตำแหน่งของแกน Y (รูปที่ 2.1)
- ค่าความองศาเร่งในตำแหน่งของแกน Z (รูปที่ 2.1)
- ค่าองศาในตำแหน่งของแกน X (รูปที่ 2.3)
- ค่าองศาในตำแหน่งของแกน Y (รูปที่ 2.3)
- ค่าองศาในตำแหน่งของแกน Z (รูปที่ 2.3)
- ค่าอัตราการเต้นของหัวใจ (ในรูปแบบ IR)
- ค่าอุณหภูมิของร่างกาย (องศาเซลเซียส)



```

15:23:23:317 [main] [INFO ] org.hibernate.Version - HHH000412: Hibernate ORM core version 5.4.23.Final
15:23:23:556 [main] [INFO ] org.hibernate.annotations.common.Version - HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
15:23:23:712 [main] [INFO ] com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
15:23:24:626 [main] [INFO ] com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
15:23:24:655 [main] [INFO ] org.hibernate.dialect.Dialect - HHH000400: Using dialect: org.hibernate.dialect.PostgreSQL95Dialect
15:23:26:927 [main] [INFO ] org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator - HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.trans
15:23:26:950 [main] [INFO ] org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean - Initialized JPA EntityManagerFactory for persistence unit 'default'
15:23:27:331 [main] [INFO ] org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor - Initializing ExecutorService
15:23:27:348 [main] [INFO ] org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor - Initializing ExecutorService 'asyncExecutor'
15:23:27:518 [main] [WARN ] org.springframework.boot.autoconfigure.orm.jpa.JpaBaseConfiguration$JpaWebConfiguration - spring.jpa.open-in-view is enabled by default. Therefore, da
15:23:30:075 [main] [INFO ] org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat started on port(s): 9090 (http) with context path '/main'
15:23:30:079 [main] [INFO ] org.springframework.data.repository.config.DeferredRepositoryInitializationListener - Triggering deferred initialization of Spring Data repositories...
15:23:30:319 [main] [INFO ] org.springframework.data.repository.config.DeferredRepositoryInitializationListener - Spring Data repositories initialized!
15:23:30:335 [main] [INFO ] com.thesis.iot.IotApplication - Started IotApplication in 15.22 seconds (JVM running for 16.963)
15:23:43:092 [http-nio-9090-exec-2] [INFO ] org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/main] - Initializing Spring DispatcherServlet 'dispatcherServlet'
15:23:43:093 [http-nio-9090-exec-2] [INFO ] org.springframework.web.servlet.DispatcherServlet - Initializing Servlet 'dispatcherServlet'
15:23:43:094 [http-nio-9090-exec-2] [INFO ] org.springframework.web.servlet.DispatcherServlet - Completed initialization in 1 ms
Hibernate: SELECT COALESCE (MAX(VAL_SET), 0) FROM T_ACTION_SIT
Hibernate: select sitaction0_.id as id1_2_0_, sitaction0_.val_accel_aix_x as val_acce2_2_0_, sitaction0_.val_accel_aix_y as val_acce3_2_0_, sitaction0_.val_accel_aix_z as val_acc
Hibernate: insert into t_action_sit (val_accel_aix_x, val_accel_aix_y, val_accel_aix_z, dt_create, val_gyro_aix_x, val_gyro_aix_y, val_gyro_aix_z, val_heart_rate, val_set, val_te
Hibernate: select sitaction0_.id as id1_2_0_, sitaction0_.val_accel_aix_x as val_acce2_2_0_, sitaction0_.val_accel_aix_y as val_acce3_2_0_, sitaction0_.val_accel_aix_z as val_acc
Hibernate: insert into t_action_sit (val_accel_aix_x, val_accel_aix_y, val_accel_aix_z, dt_create, val_gyro_aix_x, val_gyro_aix_y, val_gyro_aix_z, val_heart_rate, val_set, val_te
Hibernate: select sitaction0_.id as id1_2_0_, sitaction0_.val_accel_aix_x as val_acce2_2_0_, sitaction0_.val_accel_aix_y as val_acce3_2_0_, sitaction0_.val_accel_aix_z as val_acc
Hibernate: insert into t_action_sit (val_accel_aix_x, val_accel_aix_y, val_accel_aix_z, dt_create, val_gyro_aix_x, val_gyro_aix_y, val_gyro_aix_z, val_heart_rate, val_set, val_te
Hibernate: select sitaction0_.id as id1_2_0_, sitaction0_.val_accel_aix_x as val_acce2_2_0_, sitaction0_.val_accel_aix_y as val_acce3_2_0_, sitaction0_.val_accel_aix_z as val_acc
Hibernate: insert into t_action_sit (val_accel_aix_x, val_accel_aix_y, val_accel_aix_z, dt_create, val_gyro_aix_x, val_gyro_aix_y, val_gyro_aix_z, val_heart_rate, val_set, val_te

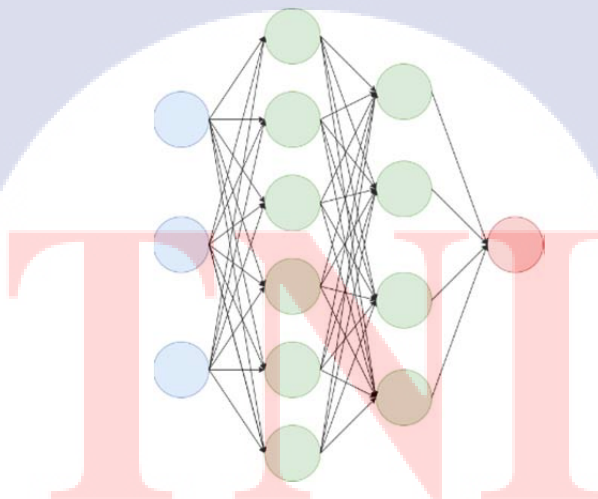
```

รูปที่ 3.7 การส่งข้อมูลจากอุปกรณ์ IoT มายัง Server ผ่าน Restful API

acc_val_accel_alix_x	acc_val_accel_alix_y	acc_val_accel_alix_z	acc_val_gyro_alix_x	acc_val_gyro_alix_y
83.86879292908378777885878892928	62.46661119994583584359475883116	78.86858473797990787084858084778	358.33032425320811911759171813413	32.161313014613714150.165
81.92871039999917461667578828289	49.102681251351281032535335137364	80.7947180778382939379818184838	5.329308290313.13036256252620334	8.92137312516.189413928.1
85.89889191817277748589889497786	67.864710298513594829658475135163	78.81888477789076817981848687697	340.3382952221646995135141.1340325	18.103215515359981371282
48.283523583483131932935713357351	34.352634932232423423273431322304	58.37344185194253273242183.1413483	295.3123262802481908425103.1431391	18.97511461498635119.155
81.72688189809784828479807683899	58.15123987946867126422612954881	76.84858381778277887985.738281798	152.148175213234242178137137186199	161.152143157144118.14314
69.70889796958586818082788179908	28.178012111611045655122653939332	78.83827777585817985.74808395778	61.231732231532732815514815410145	30.3920353058154149.127
77.788592919291781717477858788918	48.4547101110.10697416.1953616575.1	76.78847984828079878473.828584858	156.253285313304215.14814528155167	146.98542025148150.167161
352.35323383513413552716.135935527	33.39515160565261495461698981808	349.351134335534735615.14035935835	266.2481835561619635226204209127	93.104164563613692109.137
53.92430236714712331213352722.1113	48.32155345314350466352495842454	50.13592323108121112.101123292210	301.345196291255249201.13032226220	29.3114254109.114142.12630
19.555754536356675861.758785.79817	45.5856596067617062552042.7346497	19.4145393739393940.51848875.79798	140.125163.186193.298.318.339.321.106.123	151.151.142.163.140.84.7.15.37
86.78777375838589899187817677778	71.28242244545488869471.465524349	80.8484827680838187758180.8874778	134.13057145.163312.314327280201.117.1	128.16132.153.1175527.126.1
354.353356354.11.25464752576262.6467	359.8102434393930.393446384249535	261.324341.348.16.29.51.61.57.65.61.67.66.63	185.170.138.117.96.112.123.76.63.61.36.59.57	135.116.117.124.107.145.160.51
77.76828885818996918276667481807	55.267664645984110.10033612094957	72.83878680768274818566808780748	175.216212265251257201.18351.163.147.1	151.150.167.107.137.126.177.177
81.86918487899180.737877838987888	36.6610135770869633.7433647846780	83.8180908284818387798183.8183.827	253.18359346264.19612787.178.192.222.25	92.47.142.96.157.139.85.146.14
66.8182859592756978.7566887488847	7.584449.122.1993393404237.12.23.66.86	86.77828580909597797884805664648	357.318340317348355331.195.181.202.188	33.39.19.58.17.10.57.159.172.169
78.8178828584888495.10085967763.85	6.5436444534481.58.125.125.35.105.40.357	88.77818285917880817586687991.788	153.175.177.229.292.233.236.209.189.350.169	142.166.150.132.42.159.155.168
357.35530.2349.31.42.80918887788583.81	66.61.73.74.77.64.46.75.96808060.60.15.849	358.3579.6.14.16.40.5675.81.76.70.81.88.838	66.10282669370.355167.151.47.154.159.32	55.118.69.41.96.69.66.162.163.3
85.81799090908878.8177.77828489929	77.48.19.909995.79.36.24.30.42.47.46.84.110	71.798677868281818682788184.82.84.8	114.22306332296203.138.137.28.159.18.338	107.31.46.14.38.153.140.158.22.1
85.878986849389928478.72.708085909	40.5286663.111.89.101.5143.20.17.34.23.90	86.877882898085778378838388828	251.258237254181.192.188.176.159.146.142	128.122.140.113.72.168.168.168
69.7780848684889180.7676758185898	10.41.42.5869.15.76.10453.33246.3257857	86.78818081888383778981.758482808	173.245.199.329.348.311.241.177.125.48.172.1	156.95.109.58.13.29.127.163.142
84.88727383981059289848688838779	56.863584260.134.140.109865828822782	81.7090.74.78807783778188798671.768	150.433452322308351.25.28.162.200.192.1	167.18.14.74.2231.10.14.32.52.11
96.96849594828478.848786908979.797	113.11232117.137.53.74.35.42.77669485.30	75.7586808680708184.798285.7983.758	99.10.14.189.326.181.827.320.319.324.259.16	102.9.52.129.36.167.37.45.35.32
84.90908987726807884878791.88868	45.90968471.323575145537367.108.786	84.8084858378907778280848683827	308.328304227.138.125.78.164.28.324.331.3	34.13.29.133.148.150.72.161.586
84.9387938372808177838490949293.7	62.112359.106.56.13.165747.525494.123.22	80.8290.79808587.76.7680828183.92817	335.340.285.221.108.119.144.164.26.1.336.33	23.9.55.136.116.140.136.152.42.2
81.778387858887788078.72858987918	61.33556728796549.1949.114.89687.103	73.81818387828577867686858183.838	174.195.299.339.319.268.330.151.28.92.42.23	155.104.64.15.21.91.82.162.1794
79.79849187908780807482828489879	60.424798559474.3023.266462.5183333	71.80848186818284858274.768384917	128.346318335314.277.142.133.38.169.59.1	104.38.30.15.22.81.144.158.30.1
81.80818788888682787573.818788898	61.3439745381596035439253573.776	74.83828187828377917781838183.868	108.0302340260.220.175.114.45.164.53.34.3	94.35.56.11.108.141.159.128.38.1
70.70808790848887907267788089969	15.3433529235856897.29243555783.12	84.81838682867183837978907481.798	101.241.270.281.274.232.163.166.166.139.322	98.111.88.67.81.130.158.149.158
81.82818280858789858897989397.757	38.344866341858885870.120.127.112.84	83.84807383888584828477788179789	128.67.171.90.267.221.229.256.244.198.192.1	141.70.163.90.90.155.154.120.14

รูปที่ 3.8 ตัวอย่างข้อมูลดิบก่อนทำการคำนวณค่าเฉลี่ย

โดยที่จะนำเอาค่าดังกล่าวเหล่านี้เข้าทำการเก็บเป็นชุดเซตข้อมูลอย่างละ 60 แล้วทำการหาค่าเฉลี่ยในแต่ละชุดข้อมูลเพื่อให้ชุดข้อมูลที่เก็บมาจากตัวอุปกรณ์ ทำการคิดผลลัพธ์ทั้งหมดจากกิจกรรมในช่วงเวลาหนึ่ง จากนั้นจึงทำการนำเข้าสู่ระบบ FFNN เพื่อทำการวิเคราะห์ในระบบภายในจากนั้นจึงทำการหาผลลัพธ์มา โดยที่ข้อมูลส่วนนี้จะได้ออกมาเป็น Model ในการหาค่ากิจกรรมออกมา



รูปที่ 3.9 รูปแบบของ FFNN บอลสีฟ้า (Input Layer) บอลสีเขียว (Hidden Layer) บอลสีแดง (Output Layer)

โดยค่าเหล่านี้จะทำการยิงส่งไปยังเซิร์ฟเวอร์ ผ่านโปรโทคอล (Protocol) ผ่าน HTTP Request เข้ามาแล้วทำการนำค่าบันทึกไว้ในดาต้าเบส แล้วนำไปสู่กระบวนการสร้างโมเดล

3.3 ลำดับขั้นตอนการดำเนินงาน

ในงานวิจัยนี้ จะทำการทดลองโดยการให้ผู้ทดลองนั้นทำการสวมใส่อุปกรณ์ไว้ โดยที่อุปกรณ์นั้นจะทำการต่ออินเทอร์เน็ตส่งข้อมูลเข้าหาเซิร์ฟเวอร์ โดยที่การเก็บข้อมูลของแต่ละกิจกรรมนั้นจะใช้เวลาในการจับข้อมูลอยู่ที่ 10 วินาที (Second) โดยอ้างอิงจากงานวิจัย [2] ในวรรณกรรมที่เกี่ยวข้องที่มีการกำหนดเวลาในการเก็บข้อมูล เนื่องจากการกิจกรรมนั้นเป็นกิจกรรมที่บางครั้งมีความต่อเนื่องซึ่งค่าต่างๆ ที่ใช้ในการเก็บข้อมูลนั้น อาจมีการเปลี่ยนแปลงไปด้วยตัวอย่างเช่น การวิ่งการเดิน เนื่องจากการกิจกรรมประเภทออกกำลังกายส่วนใหญ่นั้น จะมีการเปลี่ยนแปลงในด้านของลักษณะท่าทางที่เปลี่ยนแปลงตลอดเวลา อัตราการเต้นของหัวใจที่แปรผันเปลี่ยนไปตามเวลาที่ใช้ รวมไปถึงอุณหภูมิของร่างกาย เป็นต้น จากนั้นจึงนำเอาข้อมูลการทดลองทั้งหมดที่ได้มานั้น นำออกมาแยกข้อมูลออกมาเป็น 70:30 (4000 ข้อมูล) แล้วทำการสร้างโมเดลโดยใช้ข้อมูล 70% ในการเรียนรู้ (Training) แล้วอีก 30% (1200 ข้อมูล) จะใช้การทดสอบ (Testing) การความแม่นยำของโมเดล และมีการเก็บข้อมูลที่ใช้ตรวจสอบ (Validation) แยกออกมา โดยโมเดลที่จะทำการใช้นั้นจะอยู่ในรูปที่รับค่าเข้าสู่ระบบหรือ Feature เข้ามาภายหลังจากการหาค่าเฉลี่ยในแต่ละชุดเป็นที่เรียบร้อยแล้ว

ตารางที่ 3.1 ตัวอย่างข้อมูลหลังทำการคำนวณค่าเฉลี่ย

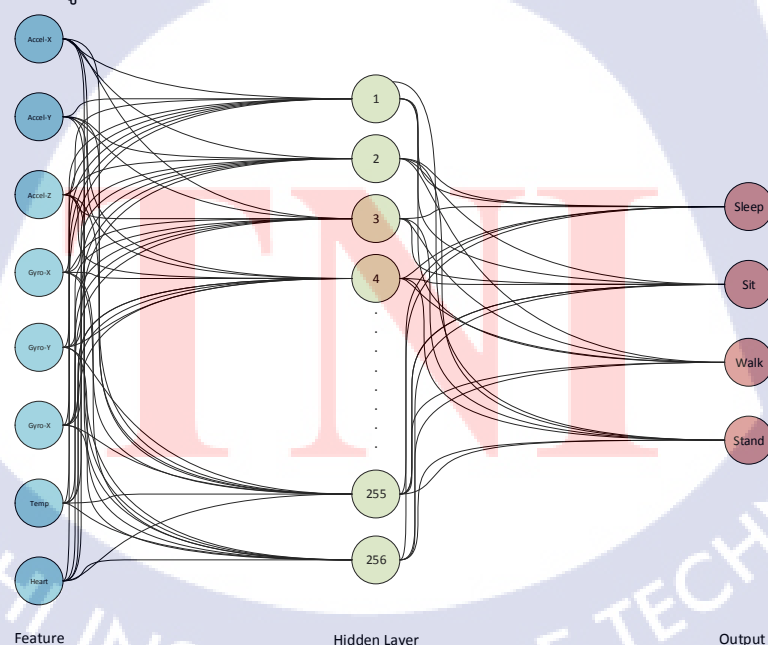
accelX	accelY	accelZ	gyroX	gyroY	gyroZ	heart	temperatu	result_star	result_slee	result_sit	result_wal
245.17	75.58	222.73	192.08	88.2	217.22	111024.4	32.34	0	0	1	0
260.57	42.15	247.4	238.62	84.63	274.72	111403.2	32.43	0	0	1	0
114.13	32.82	134.75	239.43	89.68	278.02	111023.3	32.49	0	0	1	0
349.05	14.02	323.88	272.83	87.62	326.17	111858.7	32.5	0	0	1	0
354.55	13.02	339.92	254.18	88.7	299.82	110563	32.55	0	0	1	0
353.83	11.5	335.15	244.63	87.48	297.38	110799.9	32.56	0	0	1	0
353.43	10.78	330.73	205.87	87.98	231.48	111259.4	32.58	0	0	1	0
335.23	10.57	311.93	236.73	90.08	287.88	110730.6	32.64	0	0	1	0
355.15	17.25	346.02	265.98	88.12	330.93	110963.6	32.68	0	0	1	0
336.7	15.07	325.42	253.58	83.95	298.62	110724.3	32.77	0	0	1	0
338.25	17.9	326.33	242.02	87.78	290.97	110838.5	32.88	0	0	1	0
338.97	15.55	333.58	262.53	87.78	326.4	110625.2	32.96	0	0	1	0
356.7	18.17	351.3	271.25	84.52	326.75	110155.2	33.09	0	0	1	0
357.47	16.73	352.87	274.48	85.68	330.88	109975.7	33.16	0	0	1	0
357.23	16.52	352.37	279.45	86.62	343.15	110373.7	33.29	0	0	1	0
357.63	15.7	353.1	292.78	85.35	349.77	110577.1	33.4	0	0	1	0
320.5	18.62	308.92	235.4	86.03	279.47	110384.2	33.47	0	0	1	0
279.2	11.77	270.93	245.62	85.27	277.62	110310.9	33.54	0	0	1	0
309.23	14.87	304.42	259.27	86.52	309.33	110635.4	33.61	0	0	1	0
227.03	19.62	226.17	263.77	86.15	318.6	110858.8	33.67	0	0	1	0
101.72	14.42	102.13	289	86.18	350.07	110685.1	33.73	0	0	1	0
244.88	15.8	243.68	254.4	85.73	302.33	111009.6	33.77	0	0	1	0
358.9	15.07	357.12	287.63	86.25	350.08	111175	33.8	0	0	1	0
48.12	15.08	49.58	261.8	86.43	320.27	111331.8	33.83	0	0	1	0
89.78	14.63	90.55	282.75	85.33	337.88	111240.1	33.85	0	0	1	0
161.55	14.77	161.9	288.83	85.78	344.43	111223.1	33.86	0	0	1	0
71.88	14.55	73.12	291.23	85.4	350.48	111425	33.87	0	0	1	0
304.8	16.12	303.43	266.8	87.12	325.45	111613	33.88	0	0	1	0

โดยค่า Feature เปร่านี้จะนำเข้าสู่ FFNN โดยที่ตัวโมเดลนั้นโดยใช้ค่าในการนำข้อมูลเข้าระบบหรือ BatchSize อยู่ที่ขนาด 256 จะมีคุณลักษณะการทำงานในแต่ละชั้น (Layer) ของโมเดลทั้งหมดเป็น 3 Layer ดังนี้

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 8)	72
dense_1 (Dense)	(None, 256)	2304
dense_2 (Dense)	(None, 4)	1028
Total params: 3,404		
Trainable params: 3,404		
Non-trainable params: 0		

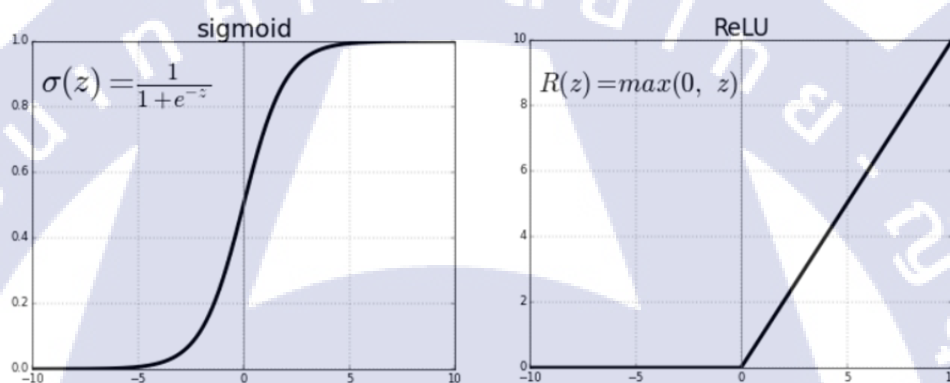
รูปที่ 3.10 ข้อมูลการทำงานของโมเดลในแต่ละชั้น

โดยที่ชั้นแรกนั้นจะเป็นชั้นที่ใช้ในการรับค่า Feature เข้ามาสู่ระบบโดยที่จะรับค่ามาทั้งหมด 8 อย่างจากที่กล่าวไปในข้อ (3.2.2) จากนั้นจึงทำการนำข้อมูลดังกล่าวเข้าสู่ชั้นที่สอง หรือเรียกว่า Hidden Layer โดยการทำงานของชั้นนี้นั้นจะเป็นการเรียนรู้ผ่านระบบภายในเพื่อทำการหาค่าเฉพาะเจาะจง หรือรูปแบบของข้อมูลทำการส่งเข้ามาโดยที่ในงานวิจัยนี้จะกำหนดโหนดการเรียนรู้หรือ Dense ในชั้นที่สองอยู่ที่ 256 โหนด และภายหลังจากทำการเรียนรู้และวิเคราะห์เรียบร้อยแล้ว จะทำการแยกแยะรูปแบบกิจกรรมออกมาในชั้นที่สามหรือ Output โดยค่าที่ได้ออกมานั้นจะเป็นตัวเลขที่บ่งบอกว่าความน่าจะเป็นของรูปแบบกิจกรรมออกมาเป็นค่าของ 1 ใน 4 กิจกรรมที่มีความเป็นไปได้ที่สุด



รูปที่ 3.11 Diagram การทำงานของโมเดล

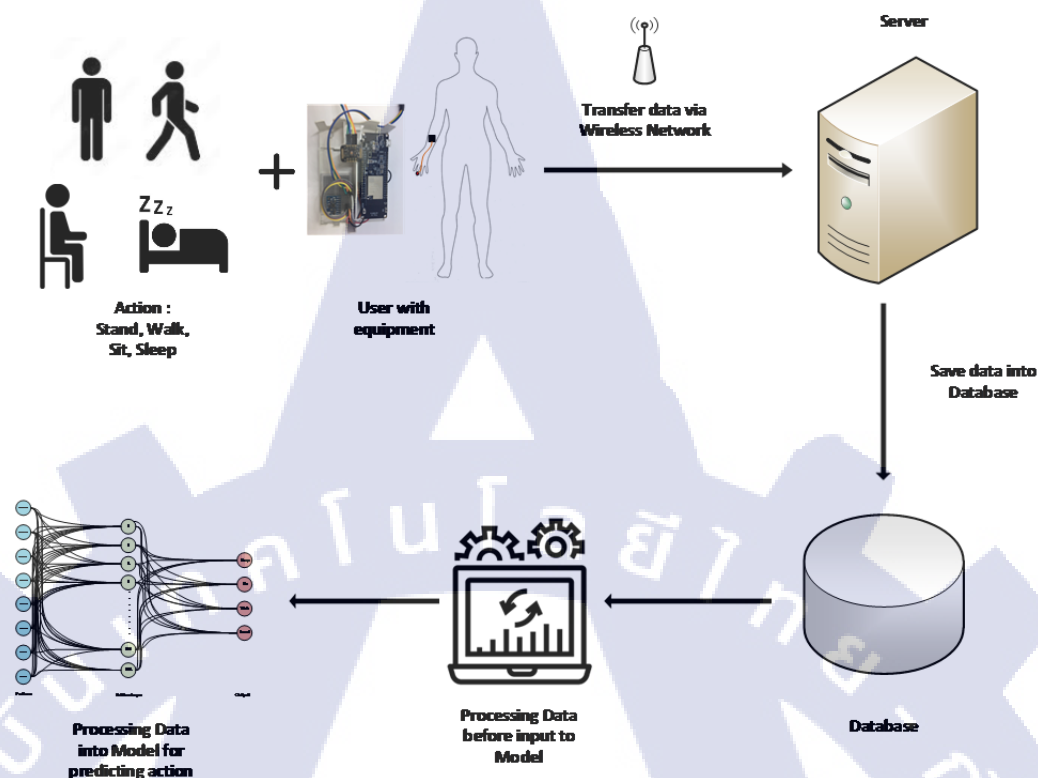
โดยที่ในส่วนของการตั้งค่าของโมเดลนั้น ในงานวิจัยนี้จะทำการใช้เซตค่าในโหนดตั้งแต่การรับค่าของ Feature และ Hidden Layer ให้เป็นในรูปแบบของ ReLU Function (Rectified Linear Unit) เนื่องจากการทำงานในส่วนของ ReLU นั้นมีความสามารถในการคำนวณค่าในการทำงานประเภท Deep Learning ได้อย่างรวดเร็ว รวมไปถึงการคำนวณเมื่อนำโมเดลไปใช้ทำงานต่อ ถึงแม้ว่าจะมีข้อเสียในด้านของการปรับแต่ง (Tunning) แต่โดยรวมแล้วเมื่อนำไปใช้งานเข้ากับการทำงานของ Optimizer ที่เป็น Adam (Adaptive Moment Estimation) และ Binary Crossentropy เนื่องจากชั้น Layer Output สุดท้ายจะบอกค่าออกมาเป็นค่าช่วงระหว่าง 0-1 อันเนื่องมาจากชั้นนี้นั้นใช้รูปแบบการทำงานแบบของ Sigmoid โดยที่ออกมานั้นจะเป็นค่าช่วงของความน่าจะเป็นในแต่ละกิจกรรม โดยที่ค่าไหนเข้าใกล้ 1 มากที่สุดคือมีความเป็นไปได้ที่จะเป็นกิจกรรมในรูปแบบนั้น



รูปที่ 3.12 รูปแบบการทำงานของ Sigmoid และ ReLU

3.4 ข้อมูลที่นำเข้ามาใช้เข้ากับโมเดลและทำการทดลอง

ในการทดลองงานวิจัยชิ้นนี้ จะทำการเก็บข้อมูลในรูปแบบ 4 อิริยาบถ ซึ่งประกอบไปด้วย ท่าทางการนั่ง ท่าทางการยืน ท่าทางการนอนราบ ท่าทางการ โดยจะทำการเก็บข้อมูลท่าทางลักษณะ อิริยาบถเหล่านี้ผ่านทางอุปกรณ์จากที่กล่าวมาในหัวข้อ 3.1 (เครื่องมือในการเก็บข้อมูลและแสดงผล การทดลองในงานวิจัย)



รูปที่ 3.13 Diagram การเก็บข้อมูลภายในงานวิจัย

ซึ่งในส่วนของอุปกรณ์นั้นจะทำการส่งข้อมูลมาทุกๆครั้งที่ข้อมูลเก็บครบ 60 ชุดในแต่ละครั้ง ซึ่งการส่งข้อมูล 60 ชุดดังกล่าวนี้จะประกอบไปด้วยข้อมูลจากเซ็นเซอร์จากที่กล่าวมาในหัวข้อ 3.2 (วิธีการในการทำงานวิจัย) อย่างละ 60 จากนั้นจึงทำการเก็บข้อมูลเหล่านี้ลงดาต้าเบส โดยทำการเก็บข้อมูลในแต่ไอริยาบทนั้นจะเป็นการเก็บข้อมูลโดยให้มีความเป็นธรรมชาติให้มากที่สุดเพื่อทำการเก็บข้อมูลในรูปแบบที่มีความเป็นไปได้อย่างมากที่สุดเท่าที่จะสามารถเก็บได้ เนื่องจากไอริยาบทบางอันนั้นสามารถที่จะทำกิจกรรมอื่นไปพร้อมกันได้

3.4.1 ท่าทางการนั่ง ยืน เดิน นอนราบ

ในการเก็บข้อมูลในไอริยาบททั้งหมดนี้ จะทำการเก็บข้อมูลตั้งแต่จากท่ากิจกรรมอื่นๆมาสู่เป็นการนั่ง ยืน เดิน นอนราบ ตัวอย่างเช่น การเก็บข้อมูลในรูปแบบนั้นนั้น จะทำการเก็บตั้งแต่เดินแล้วเปลี่ยนมานั่ง เป็นต้น โดยที่ระหว่างที่ทำการเก็บข้อมูลอยู่นั้น ผู้ที่สวมใส่อุปกรณ์นั้นสามารถที่จะทำกิจกรรมอื่นๆควบคู่ไปได้

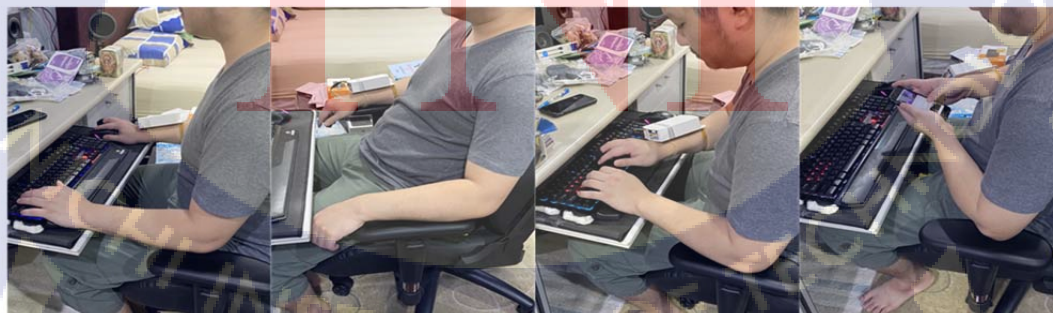


รูปที่ 3.14 การติดตั้งอุปกรณ์บนร่างกายผู้ใช้

โดยที่ท่าทางนั้นต้องยังอยู่ในลักษณะท่าทางของอิริยาบถนั้นอยู่และไม่มีการเคลื่อนที่จนทำให้เปลี่ยนอิริยาบถดังกล่าวไปไหนจนมากเกินไป เช่น ระหว่างที่นั่งอยู่นั้นสามารถที่จะทำการเขียนหรือจดบันทึก หรือ ระหว่างทำการเดินอาจจะสามารถแกว่งแขนออกกำลังกายหรือใช้งานมือถือไปพร้อมกันได้ หรือ การยืนอยู่กับที่แต่ยังคงสามารถสามารถขยับเปลี่ยนท่าทางยืนหรือแม้กระทั่งการหมุนตัวไปยังทิศต่างๆ เป็นต้น



รูปที่ 3.15 รูปแบบตัวอย่างการเก็บข้อมูลท่าเดิน



รูปที่ 3.16 รูปแบบตัวอย่างการเก็บข้อมูลท่านั่ง



รูปที่ 3.17 รูปแบบตัวอย่างการเก็บข้อมูลทำยืน



รูปที่ 3.18 รูปแบบตัวอย่างการเก็บข้อมูลทำนอน

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

บทที่ 4

ผลการวิเคราะห์ข้อมูล

จากการทดลองและดำเนินงานวิจัยมานั้น ทำให้สามารถทำการวิเคราะห์ข้อมูลความสัมพันธ์ในส่วนของการทำงานในโมเดลได้โดย วิเคราะห์จากข้อมูลที่ผ่านการคำนวณและเข้าโมเดลแล้ว โดยแยกออกมาเป็นหัวข้อดังต่อไปนี้

4.1 การวิเคราะห์ในส่วนของข้อมูล

4.2 การวิเคราะห์ในส่วนของโมเดล

4.1 การวิเคราะห์ในส่วนของข้อมูล

ในเนื้อหาส่วนนี้จะทำการวิเคราะห์ความสัมพันธ์ระหว่างข้อมูลทั้งแปดหลังจากที่ทำการเก็บข้อมูลออกมาโดยในการทดลองวิจัยในครั้งนี้ได้ใช้ข้อมูลในการเทรนนิ่งหรือเรียนรู้เป็นจำนวน 4,000 ข้อมูลจาก 4 รูปแบบกิจกรรม และมีข้อมูลสำหรับตรวจสอบและทดสอบนอกเหนือจากนี้ข้อมูลตั้งต้น

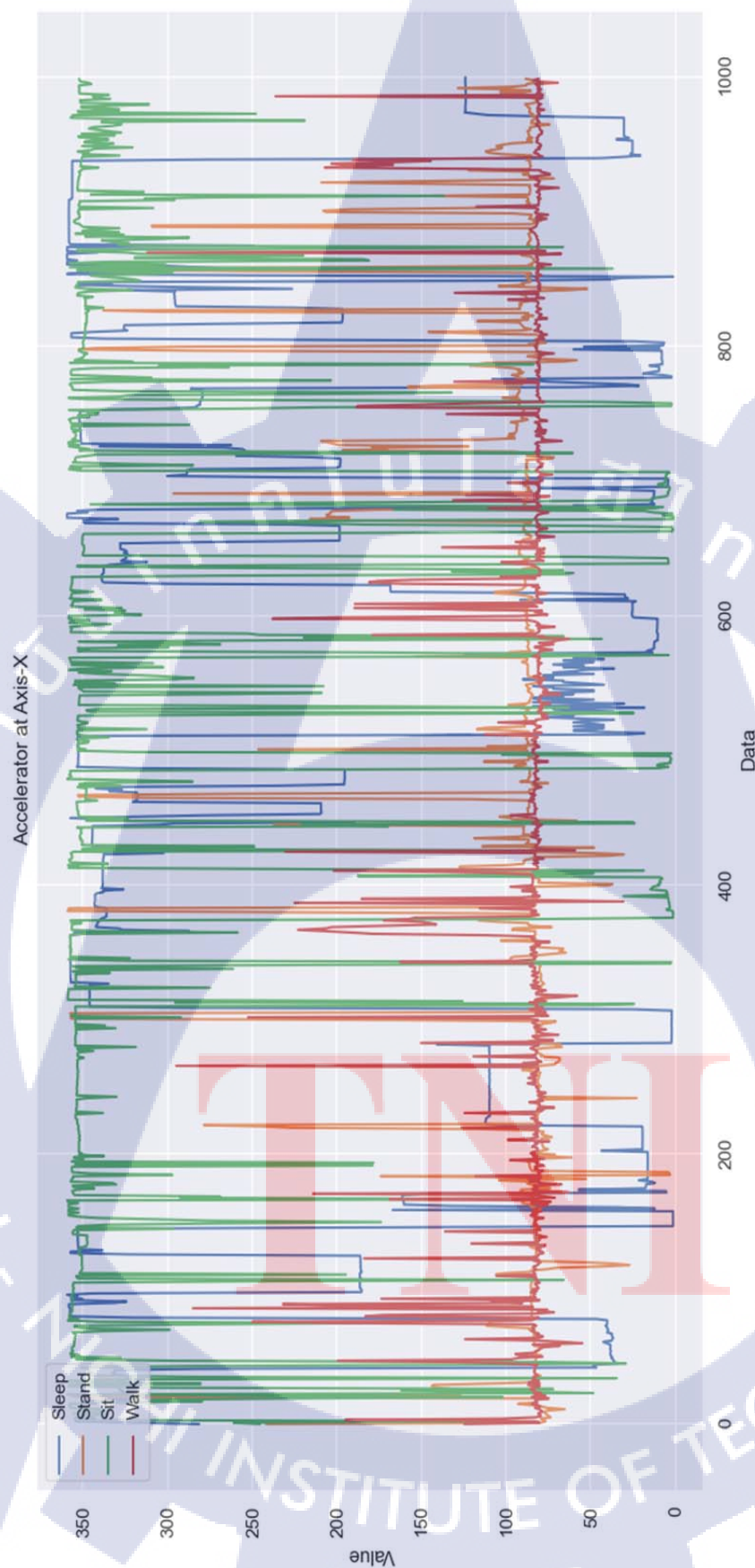
ตารางที่ 4.1 Diagram การเก็บข้อมูลภายในงานวิจัย

	จำนวนข้อมูล
นอนราบ	1,000
นั่ง	1,000
ยืน	1,000
เดิน	1,000
รวม	4,000

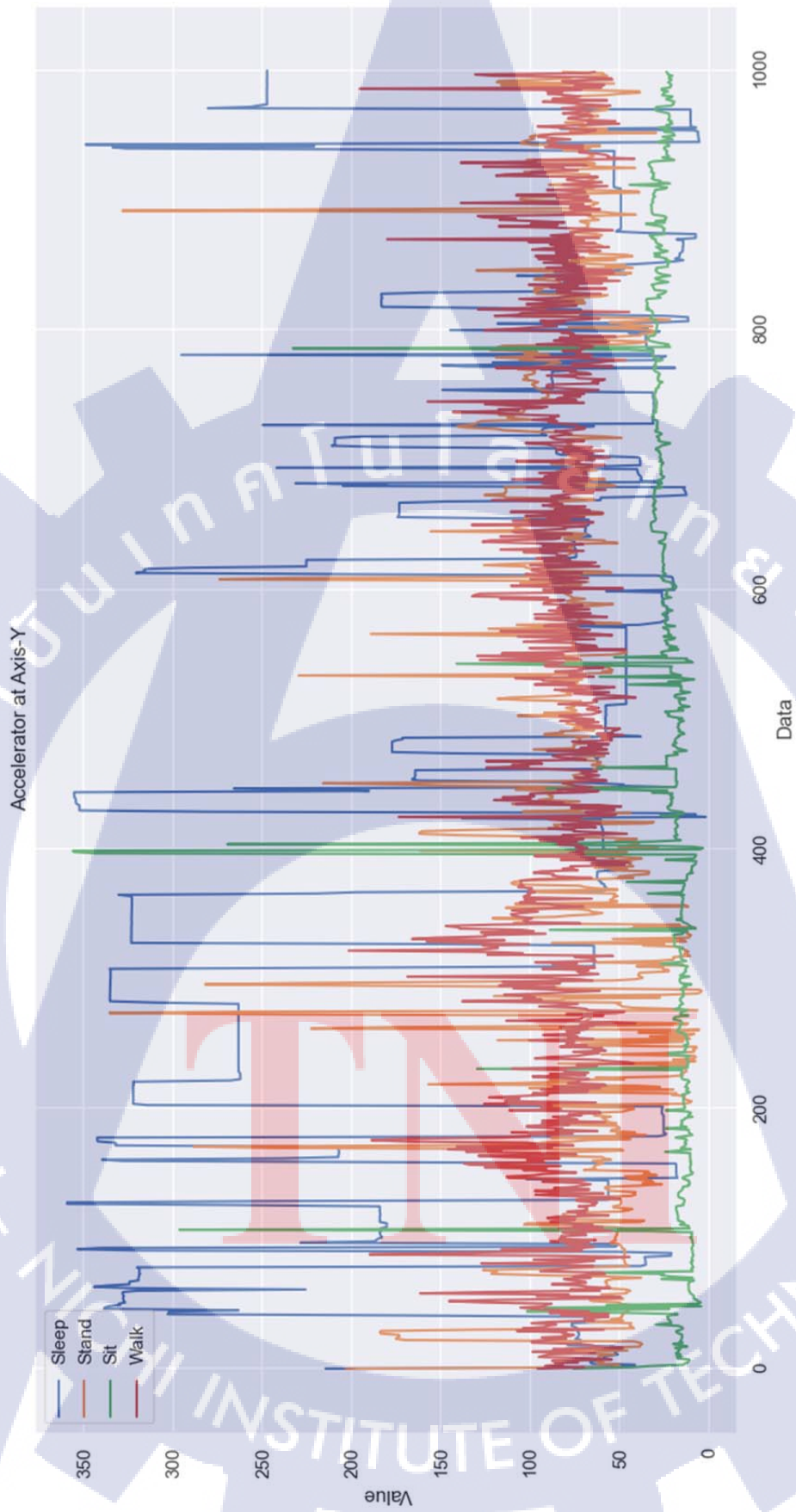
อย่างที่กล่าวไปในบทที่ 3 ว่าด้วยข้อมูลในแต่ละชุดนั้นจะประกอบไปด้วยค่าทั้งหมดแปดค่า ซึ่งข้อมูลเหล่านี้จะถูกเก็บผ่านตัวอุปกรณ์ IoT และในแต่ละค่านั้นจะมีการเก็บข้อมูลในรูปแบบของการทำกิจกรรมอย่างต่อเนื่องเป็นจำนวน 60 ชุด หรือประมาณเป็นเวลาต่อ 10 วินาทีต่อ 1 ชุดข้อมูล

4.1.1 รูปแบบค่าความเร่งองศาในตำแหน่งของแกน X, Y และ Z

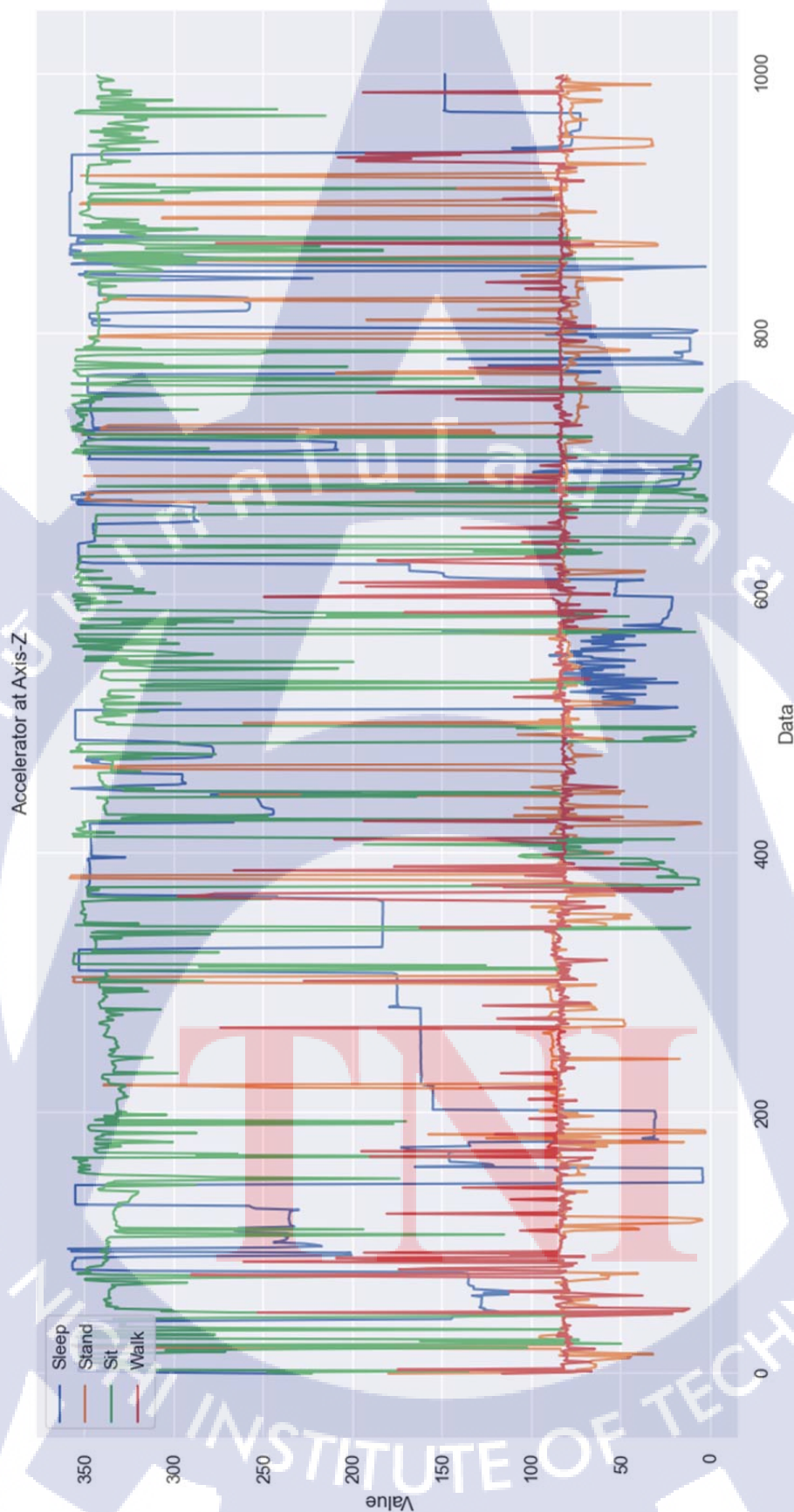
เมื่อทำการวิเคราะห์จากกราฟที่เก็บข้อมูลมาใน รูปที่ 4.1, 4.2 และ 4.3 จะพบว่า ข้อมูลในแต่ละกิจกรรมนั้นจะมีการเกาะกลุ่มเป็น อย่างเช่นข้อมูลในส่วนของแกน X ที่จะมีการแบ่งแยก ออกมาเป็นสองกลุ่มหลักๆ อย่างเช่น การนอนราบและการนั่งจะถูกจัดกลุ่มอยู่ในช่วงของกราฟตอนบน และล่าง ในขณะที่การยืนและการเดินนั้นจะอยู่ในช่วงของกราฟตอนกลาง ถัดมาข้อมูลในส่วนของแกน Y นั้น การนอนราบนั้นจะมีข้อมูลจะมีความผันผวนตามกิจกรรมโดยที่การนอนราบมีความผันผวนมา ที่สุด ทำให้เส้นกราฟเปลี่ยนไปมาอันเนื่องมาจากการนอนราบนั้น ข้อมูลที่ทำการเป็นมาได้มีการเปลี่ยน ท่าทางมาก ในขณะที่กิจกรรมอื่นๆ นั้น จะเก็บในรูปแบบปกติหรือรูปแบบในการทำงานที่เปลี่ยนลักษณะ ท่าทางของส่วนที่ติดตั้งอุปกรณ์นั้นไม่มีความเคลื่อนไหวไปในรูปแบบอื่นๆ มากมายนัก โดยที่การนั่งนั้น ยังคงอยู่ที่แนวช่วงล่างของกราฟ ในส่วนของการยืนและเดินนั้นยังคงอยู่ที่แนวกลางของกราฟ สุดท้าย ในส่วนของแกน Z นั้น ยังคงมีความผันผวนแต่ไม่มากเท่าแกน Y และยังคงแบ่งแยกกลุ่มออกมาได้โดย ที่ข้อมูลประเภทการนอนราบและนั่งนั้นจะเกาะกลุ่มอยู่ที่ช่วงบนของกราฟ ในขณะที่การเดินและยืนนั้น ยังคงอยู่ที่ช่วงกลางของกราฟ



รูปที่ 4.1 ค่าความเร่งองศาในตำแหน่งของแกน X ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน



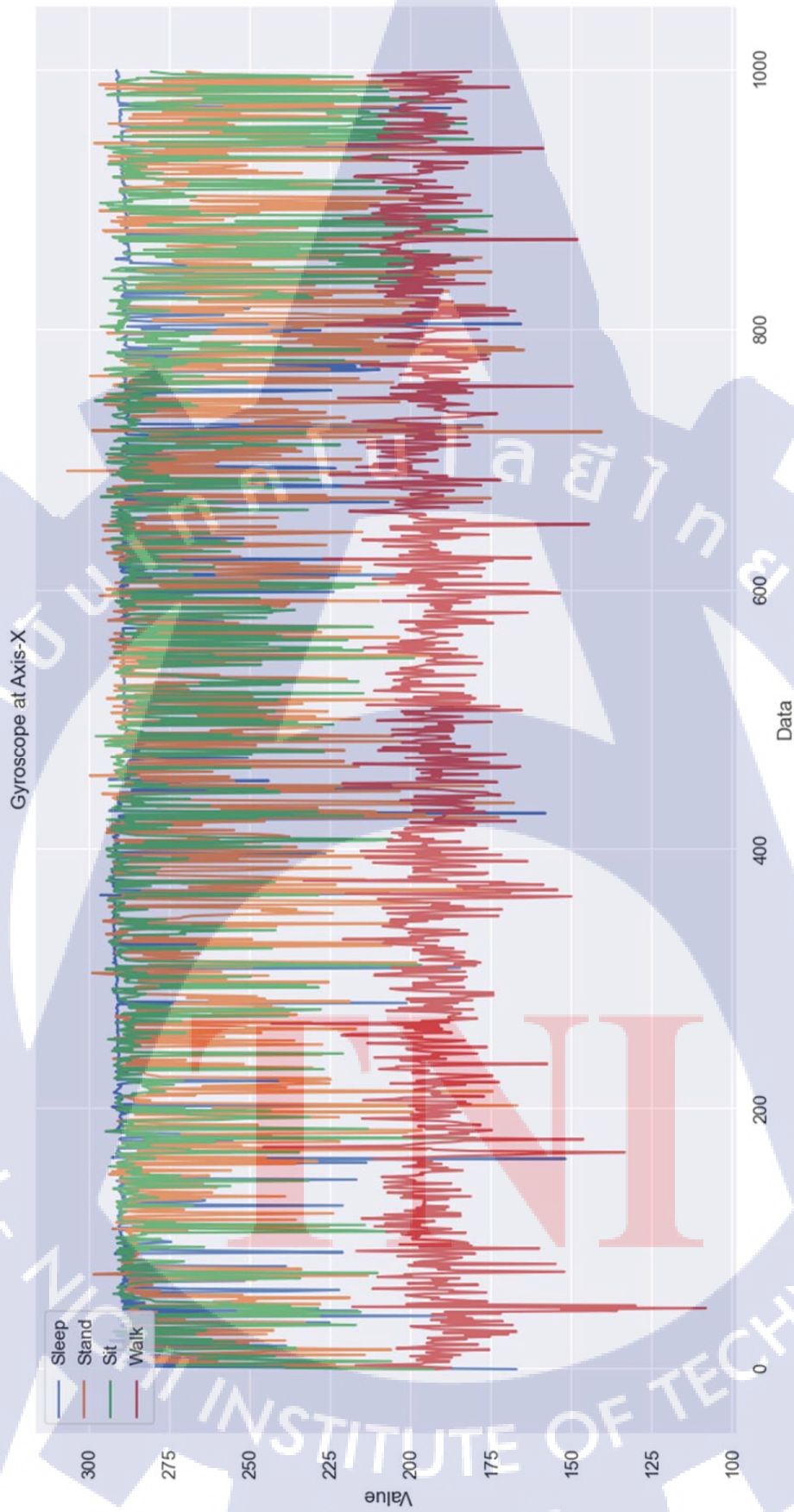
รูปที่ 4.2 ค่าความเร่งองค์ในตำแหน่งของแกน Y ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน



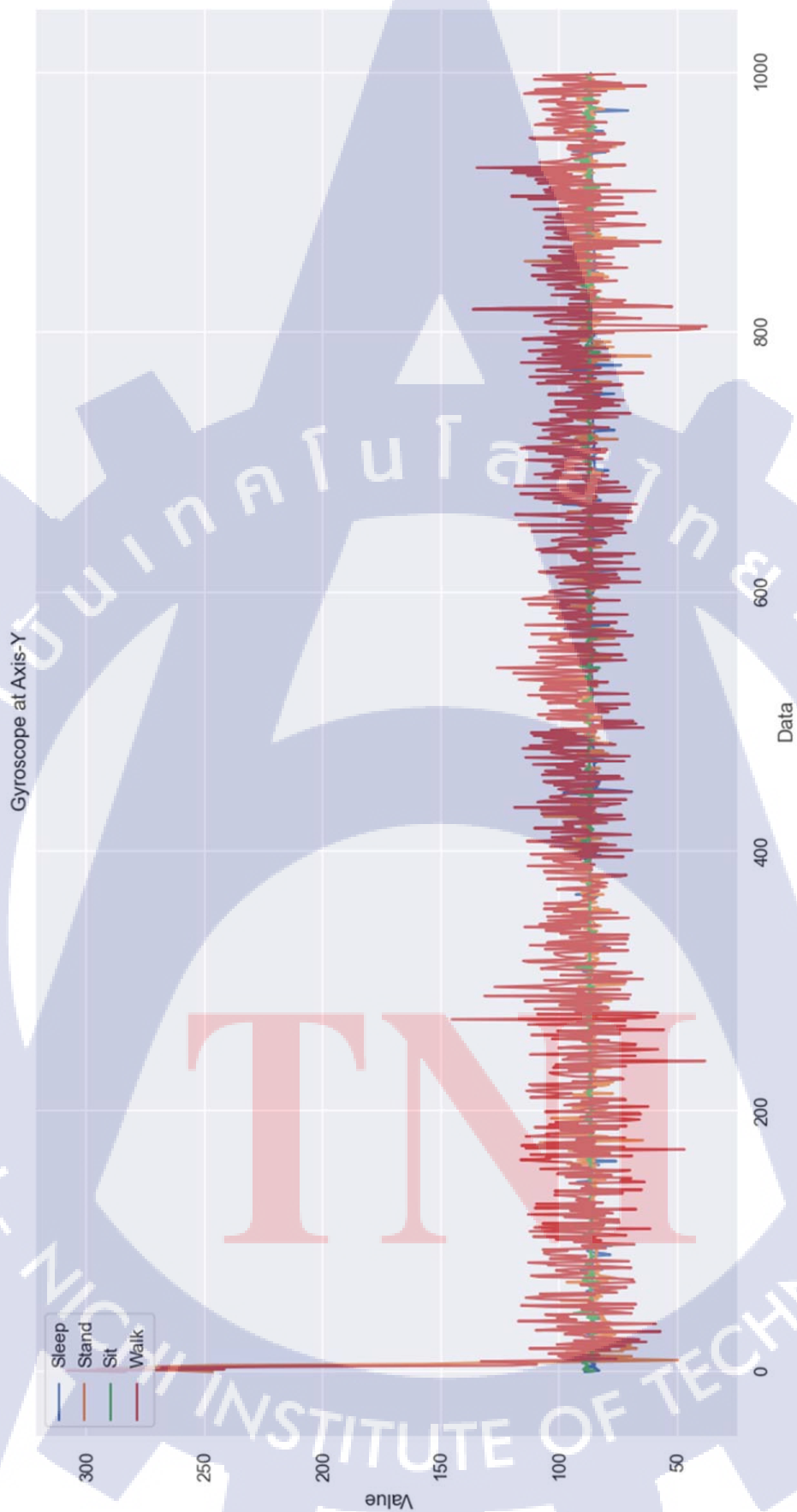
รูปที่ 4.3 ค่าความเร่งองศาในตำแหน่งของแกน Z ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน

4.1.2 รูปแบบค่าองศาในตำแหน่งของแกน X, Y และ Z

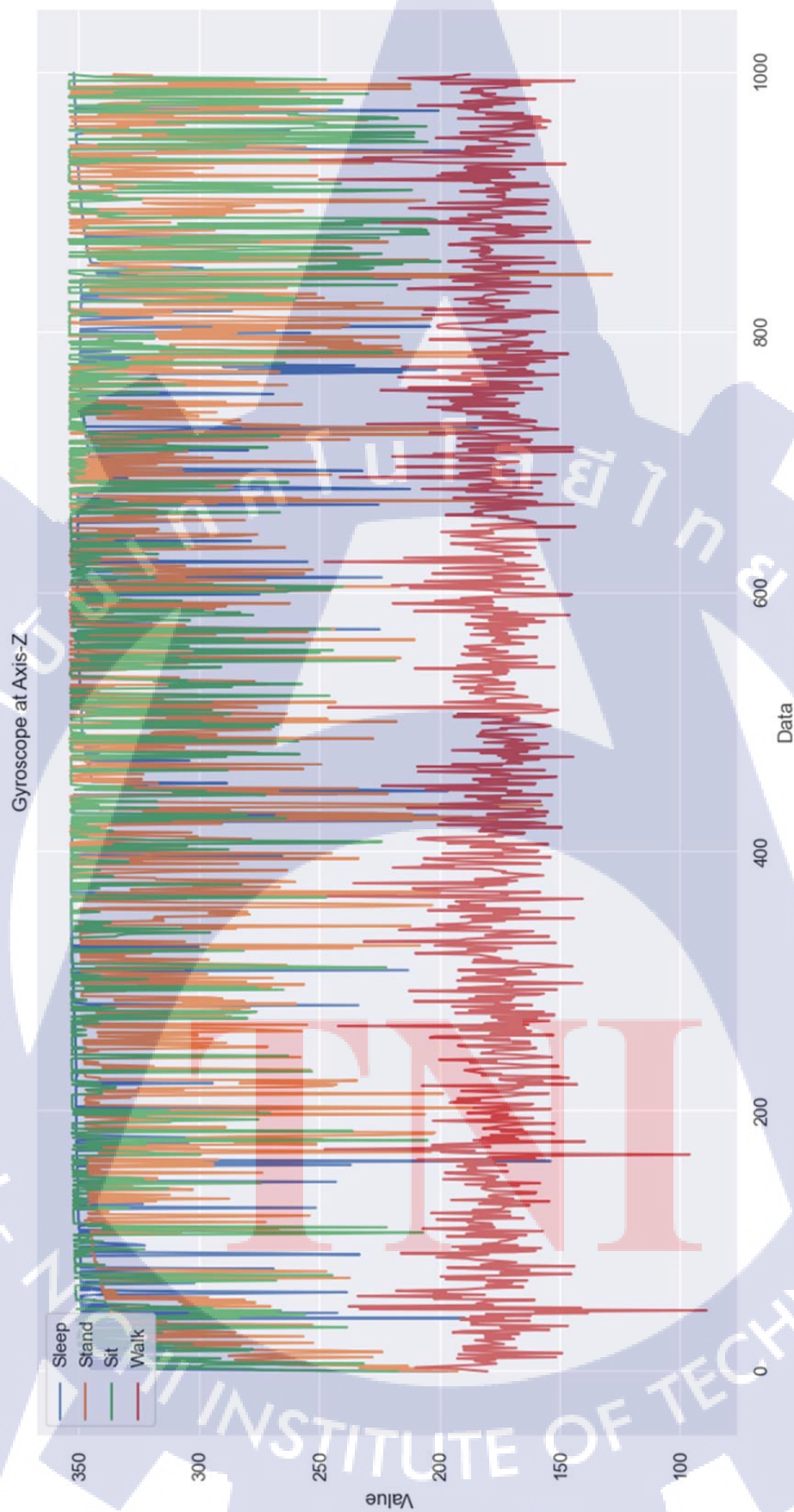
จากรูปที่ 4.4, 4.5 และ 4.6 จะสามารถแบ่งรูปแบบของกิจกรรมออกมาได้สองรูปแบบจากสามกราฟ โดยที่ข้อมูลที่ได้ออกมานั้นจะเห็นว่าเป็นรูปแบบที่เกาะกลุ่มกัน โดยที่ค่าที่วัดได้จากแกน X และ Y นั้น รูปแบบการเดินจะอยู่ในกลุ่มของช่วงกลางกราฟเนื่องจากการหมุนในรูปนี้กิจกรรมนี้จะมีการเปลี่ยนจากการแกว่งมือหรือใช้มือถือ ในขณะที่รูปแบบกิจกรรมอื่นๆ นั้น จะไม่ค่อยมีการเปลี่ยนหมุนมากนัก ทำให้ค่าที่วัดมาได้นั้นจากรูปแบบการเดินนั้นค่าจะอยู่ที่ช่วงกลางกราฟ ในขณะที่รูปแบบกิจกรรมอื่นๆ จะเกาะกลุ่มกันอยู่ที่ช่วงบน โดยที่ค่าในแกน Z ก็ไม่มีความแตกต่างกันมากในเรื่องของการเกาะกลุ่มของข้อมูล แต่ค่าในแกนนี้ยังสามารถแบ่งรูปแบบออกมาได้สามแบบ นั่นคือรูปแบบกิจกรรมการเดินที่นั้นจะมีค่าความผันผวนในระดับมากที่สุด ถัดมาจะเป็นรูปแบบกิจกรรมในการยืน เนื่องจากการยืนยังมีการทำกิจอื่นรวมอยู่ด้วย จึงทำให้เห็นว่าค่าที่อ่านมานั้นจะมีการเปลี่ยนแปลงแบบขึ้นและลงบ้างในบ้างช่วง และกลุ่มสุดท้ายจะเป็นรูปแบบกิจกรรมแบบนอนราบและการนั่งที่จะไม่ค่อยมีการเปลี่ยนแปลงมาเท่าไรในมุมมองของแกน Z



รูปที่ 4.4 ค่าองค์การในตำแหน่งของแกน X ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน



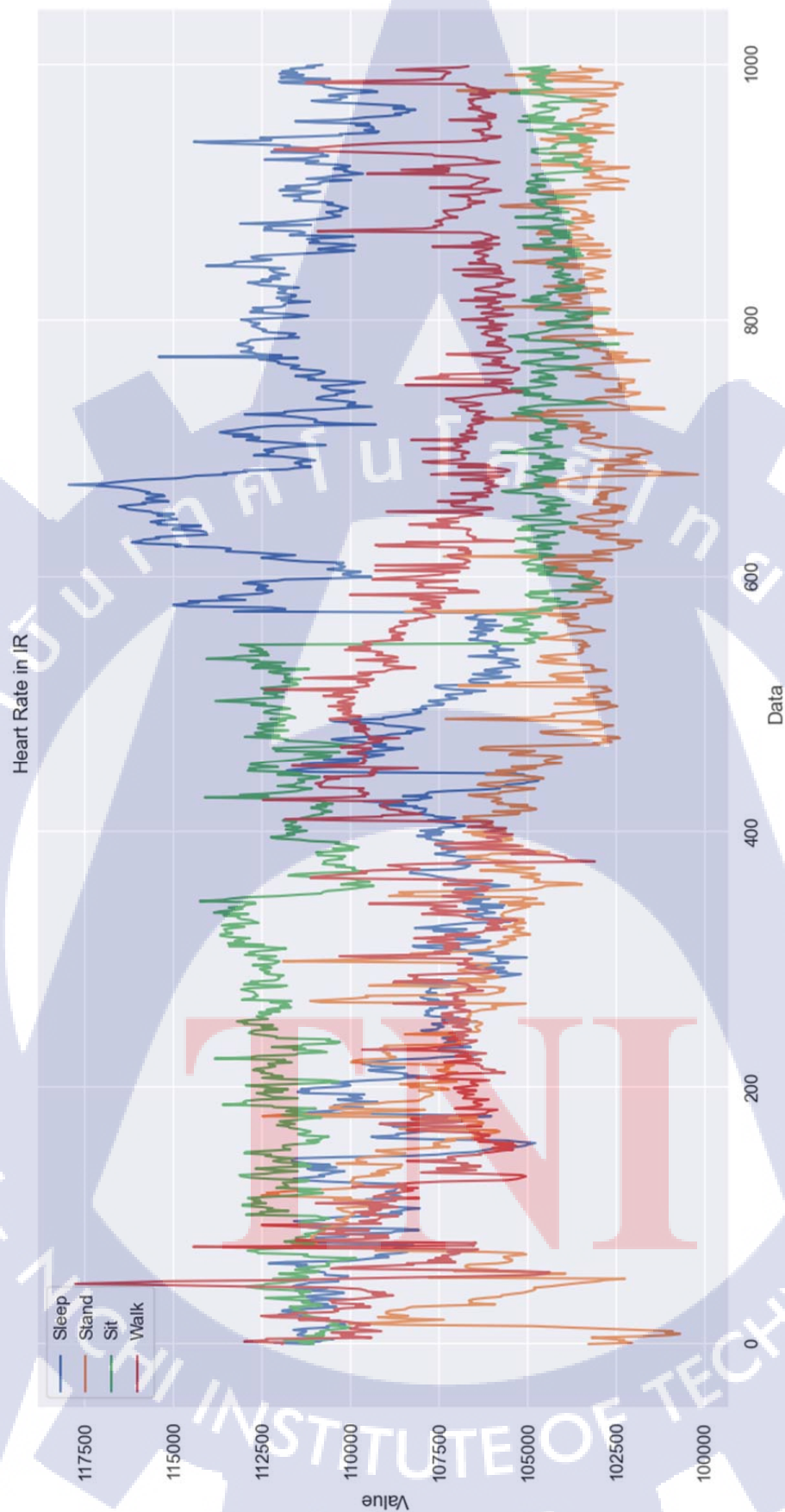
รูปที่ 4.5 ค่าองศาในตำแหน่งของแกน Y ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน



รูปที่ 4.6 ค่าองศาในตำแหน่งของแกน Z ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน

4.1.3 รูปแบบค่าอัตราการเต้นของหัวใจ (ในรูปแบบ IR)

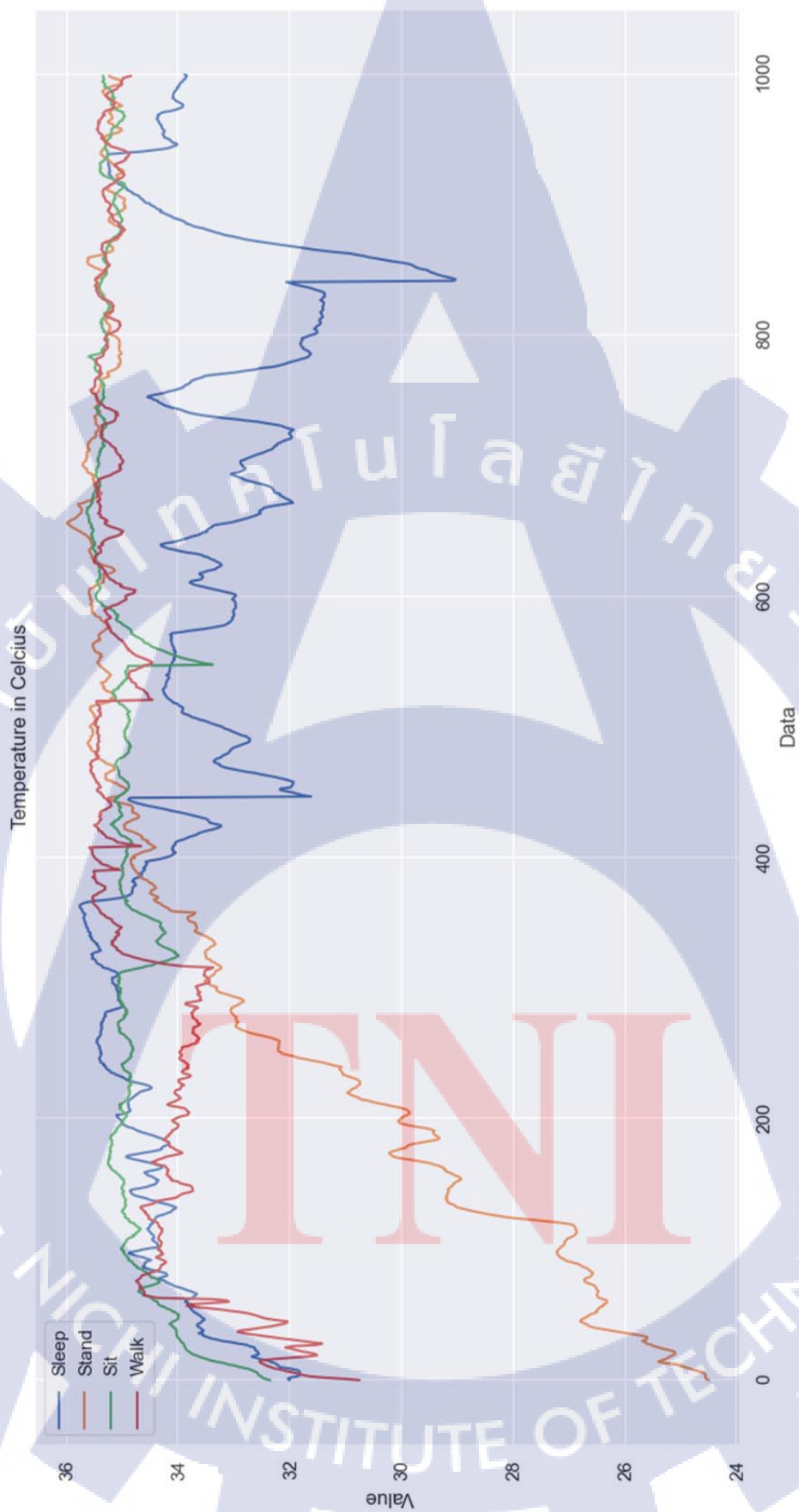
จากรูปที่ 4.7 นั้นจะเป็นข้อมูลที่ในรูปแบบของ IR หรือ Infrared ที่ใช้แสงยิงผ่านเข้าไปยังพื้นที่และให้แสงสะท้อนกลับมา โดยการทำงานในส่วนนี้คือการวัดค่าอัตราการเต้นของหัวใจผ่านทางปลายนิ้ว โดยค่าที่อ่านได้จะเอาค่าที่อ่านได้นั้น มาใช้เข้าคำนวณหาค่าเฉลี่ยโดยตรงโดยไม่ผ่านการเปลี่ยนแปลงค่าใดๆ โดยจากกราฟนั้นจะสามารถแบ่งรูปแบบของข้อมูลได้ออกเป็นสามรูปแบบ โดยที่รูปแบบแรกนั้นจะเป็นในรูปแบบกิจกรรมที่เป็นแบบนั่งและยืน โดยที่รูปแบบค่าที่อ่านออกมาได้จะไปในทิศทางกราฟช่วงบนและล่างลากยาวไปอย่างต่อเนื่อง ในขณะที่รูปแบบกิจกรรมการนอนราบนั้นค่าที่วัดได้จะอยู่ในช่วงกลางและบนของตาราง เนื่องจากค่านั้นจะอยู่ในรูปแบบการนอนราบในรูปแบบที่ตื่นและหลับอยู่ ทำให้เห็นว่าค่าที่วัดได้จากการหลับและตื่นนั้นจะวัดได้ค่าที่ต่างกันอยู่ในระดับหนึ่ง และค่าในกลุ่มสุดท้ายคือรูปแบบกิจกรรมในการเดินที่จะมีค่าอยู่ในช่วงกึ่งกลางของกราฟ แต่จะมีค่าผันผวนในระดับหนึ่ง อันเนื่องจากการเก็บข้อมูลนั้นมีการเดินในรูปแบบต่างๆ โดยประกอบไปด้วยการเดินทางราบ ทางขึ้นขึ้นและทางขึ้นลง (การขึ้น-ลงบันได) รวมไปถึงการเดินถือของ ทำให้เห็นว่าเมื่อทำกิจกรรมที่มีการใช้แรงเข้ามาเกี่ยวข้องแม้เป็นกิจกรรมแบบเดินแต่ก็ทำให้เห็นถึงรูปแบบของอัตราการเต้นของหัวใจที่เปลี่ยนไปในรูปแบบที่ขึ้น โดยบางกิจกรรมนั้นจะเห็นว่าเมื่อมีการทำกิจกรรมอย่างต่อเนื่อง ก็มีผลต่ออัตราการเต้นของหัวใจอยู่อย่างเห็นได้ชัดเจน



รูปที่ 4.7 ค่าอัตราการเต้นของหัวใจ (ในรูปแบบ IR) ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน

4.1.4 รูปแบบค่าอุณหภูมิของร่างกาย (องศาเซลเซียส)

จากรูปที่ 4.8 จะสามารถวิเคราะห์ได้ว่าปัจจัยของอุณหภูมินั้น เมื่อทำการเก็บข้อมูลไปได้ระยะหนึ่งกราฟของทุกกิจกรรมนั้นจะไปในเส้นแนวทางเดียวกันหรือ 34°C ไปจนถึง 36°C แม้ว่าการเก็บค่าเริ่มต้น ณ ขณะนั้นไม่เท่ากันหรือการเก็บข้อมูลเียบ้างช่วงอาจไม่มีความต่อเนื่องแต่อ้างอิงจากงานวิจัย [9] จะสามารถรับรู้ได้ว่า ร่างกายของมนุษย์นั้นจะค่อยทำการปรับหรือควบคุมสมดุลอุณหภูมิของร่างกายให้อยู่ในทิศทางที่เหมาะสมไปตามลำดับ และอีกตัวแปรคือขึ้นอยู่กับปัจจัยของสภาพแวดล้อมภายนอกด้วย เพราะการระหว่างทำการเก็บข้อมูลนั้น ไม่ได้มีการจำกัดสภาพแวดล้อมไว้ ทำให้การเก็บข้อมูลนั้นอาจมีความคลาดเคลื่อนในช่วงต้นหรือระหว่างทำการเก็บข้อมูลได้ โดยข้อมูลที่มีความผันผวนมากที่สุดจะเป็นข้อมูลในส่วนของกิจกรรมที่เป็นในรูปแบบนอนราบ ซึ่งทำให้เซ็นเซอร์ที่ทำการอ่านค่าจากส่วนบริเวณที่อุปกรณ์ทำการติดตั้งนั้นเปลี่ยนแปลงไปในส่วนของช่วงท้ายๆ แต่ยังคงสามารถนำมาใช้เพื่อทำการอ้างอิงในกรณีที่มีปัจจัยภายนอกเข้ามาคาดคะเนในส่วนของกิจกรรมที่สามารถเป็นไปได้ แต่สามารถเห็นได้อีกส่วนคือข้อมูลการเดินทาง เมื่อทำการเดินไประยะเวลาประมาณหนึ่ง อุณหภูมิจะมีการเพิ่มขึ้นจากตอนเริ่มต้น ซึ่งให้เห็นว่าจากกิจกรรมประเภทนั่งและเปลี่ยนเป็นการเดินนั้น จะได้เห็นรูปแบบการเปลี่ยนแปลงไปอย่างต่อเนื่อง ซึ่งจะแตกต่างจากท่าทางการนั่งหรือยืน ที่จะมีรูปแบบและทิศทางที่ไปในทางที่คงที่อยู่เสมอตลอดในเส้นกราฟ



รูปที่ 4.8 ค่าอุณหภูมิของร่างกาย (องศาเซลเซียส) ในกิจกรรมรูปแบบ นอนราบ, ยืน, นั่ง, และเดิน

4.2 การวิเคราะห์ในส่วนของโมเดล

การวิเคราะห์ในส่วนของโมเดลนั้น จะใช้ค่าตรวจสอบจากค่า Accuracy, Precision, Recall, F1-Score, และค่า AOC (Area Under <ROC> Curve) ในการเป็นค่าดัชนีการชี้วัดถึงค่าความน่าเชื่อถือและความถูกต้องของโมเดล

Accuracy คือ ค่าความแม่นยำของโมเดลที่ใช้ในการคาดเดาหรือทำนายผลลัพธ์ออกมา โดยคำนวณจากค่าความถูกต้องทั้งหมดกับค่าความเป็นจริงที่ควรจะเป็น

$$\text{Accuracy} = \frac{T_p + T_n}{n}$$

T_p = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือจริง)

T_n = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้นไม่จริง (ทำนายว่าไม่จริง สิ่งที่เกิดขึ้นคือ ไม่จริง)

n = จำนวนข้อมูลความเป็นจริงที่นำใช้ทั้งหมด

Precision คือ ค่าความถูกต้องของโมเดลที่ใช้ในการทำนายผลลัพธ์ออกมา โดยคำนวณจากค่าความถูกต้องในส่วนของแต่ละชนิดข้อมูลกับค่าความเป็นจริงที่ควรจะเป็น

$$\text{Precision} = \frac{T_p}{T_p + F_p}$$

T_p = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือจริง)

F_p = สิ่งที่ทำนายตรงไม่ตรงกับค่าความเป็นจริงที่เกิดขึ้น (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือไม่จริง)

Recall คือ แม่นยำของโมเดลที่ใช้ในการทำนายผลลัพธ์ที่ถูกต้องและไม่ถูกต้องออกมา โดยคำนวณจากค่าความไม่ถูกต้องกับอัตราส่วนที่ถูกต้องของข้อมูลกับค่าความเป็นจริงที่ควรจะเป็น

$$\text{Recall} = \frac{T_p}{T_p + F_p}$$

Tp = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือจริง)

Fn = สิ่งที่ทำนายไม่ตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าไม่จริง สิ่งที่เกิดขึ้นคือจริง)

F1 Score คือ ค่าดัชนีชี้วัดความน่าเชื่อถือที่ดูจากค่าเฉลี่ยจาก Precision และ Recall เพื่อวัดความสามารถของโมเดล เนื่องด้วยบางครั้งไม่สามารถชี้วัดเพียงจากค่า Accuracy เพียงอย่างเดียว

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

AOC หรือ Area Under <ROC> Curve เป็นค่าอีกชี้วัดอีกแบบหนึ่งที่จะทำการวิเคราะห์การจำแนกเพื่อกำหนดว่าโมเดลใดที่ใช้ทำนายผลลัพธ์ออกมาได้ดีมากแค่ไหน โดยมีค่าละ 0-1 โดยที่ถ้าค่าเข้าใกล้ 1 หมายถึง มีค่าโมเดลที่ดี โดยจะเน้นดูที่ค่า Tpr และค่า Fpr ซึ่งเป็นค่าที่ทำนายถูกและผิดจากค่าความเป็นจริง

$$Tpr = \frac{Tp}{Tp + Fn}$$

$$Fpr = \frac{Fp}{Fp + Tn}$$

Tp = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือจริง)

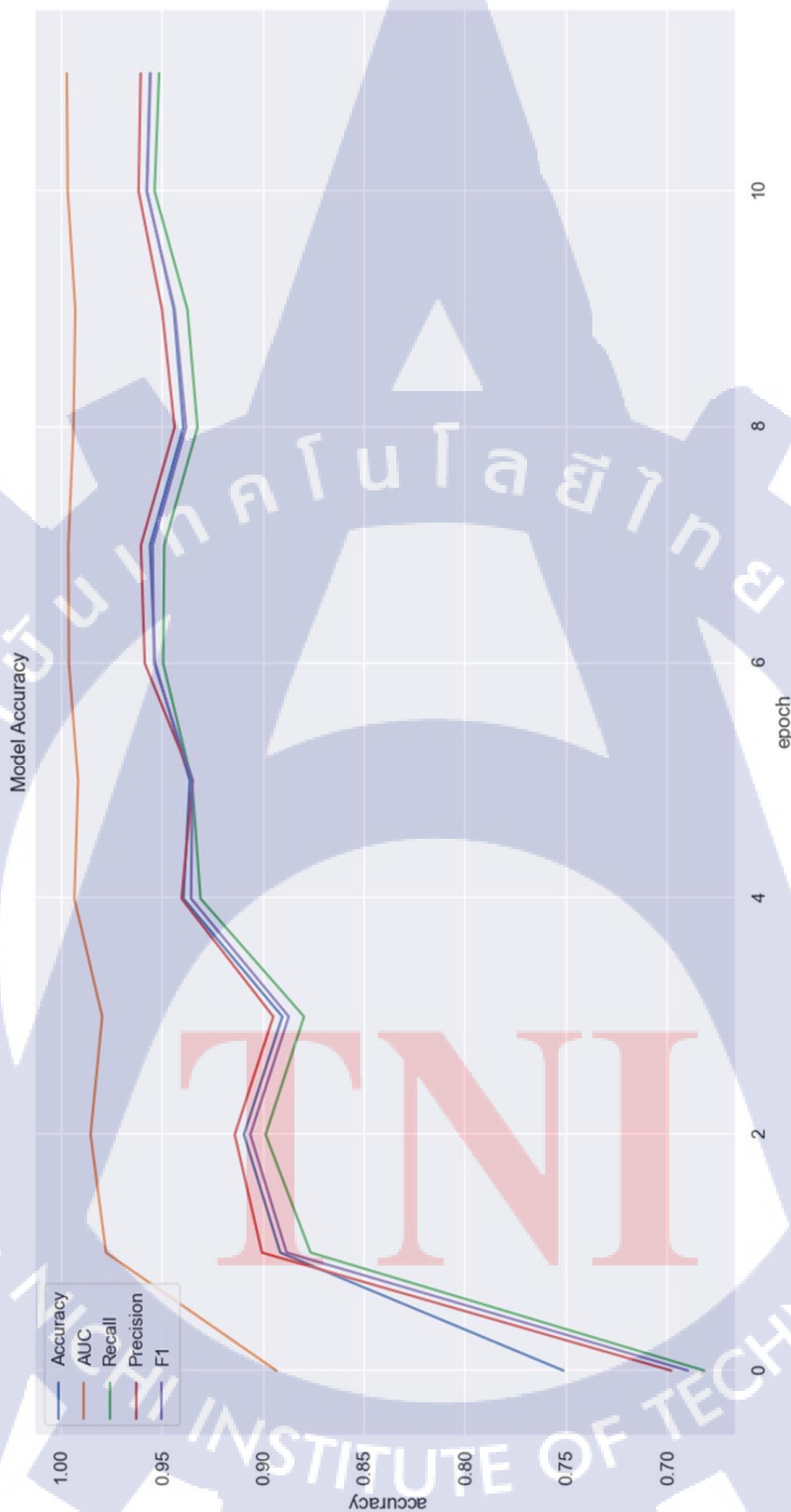
Tn = สิ่งที่ทำนายตรงกับค่าความเป็นจริงที่เกิดขึ้น (ทำนายว่าไม่จริง สิ่งที่เกิดขึ้นคือไม่จริง)

Fp = สิ่งที่ทำนายตรงไม่ตรงกับค่าความเป็นจริงที่เกิดขึ้น (ทำนายว่าจริง สิ่งที่เกิดขึ้นคือไม่จริง)

Fn = สิ่งที่ทำนายไม่ตรงกับค่าความเป็นจริงที่เกิดขึ้นจริง (ทำนายว่าไม่จริง สิ่งที่เกิดขึ้นคือจริง)

ตารางที่ 4.3 การบ่งบอกค่าของประสิทธิภาพของ AUC

	Result
= 0.5	โมเดลทำงานแบบเดาสุ่มเลย
> 0.7	มาตรฐานสำหรับโมเดลส่วนใหญ่
> 0.8	โมเดลทำงานได้ดี
> 0.9	โมเดลทำงานได้ดีมาก



รูปที่ 4.9 ค่าการทำงานของ Accuracy, Precision, Recall, F1-Score, และ AUC ในแต่ละช่วงการทำงานของ Epoch

ตารางที่ 4.4 ค่าการทำงานของ Accuracy, Precision, Recall, F1-Score และ AUC ในแต่ละช่วงการทำงานของ Epoch (แสดงค่าอยู่ ทศนิยม 3 ตำแหน่ง)

Epochs												
	100	500	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000
Accuracy	0.751	0.891	0.909	0.89	0.939	0.936	0.953	0.956	0.939	0.944	0.957	0.956
Precision	0.698	0.9	0.914	0.894	0.931	0.935	0.958	0.96	0.944	0.95	0.961	0.96
Recall	0.681	0.876	0.899	0.88	0.931	0.935	0.949	0.948	0.932	0.937	0.954	0.951
F1	0.689	0.888	0.906	0.887	0.935	0.935	0.954	0.954	0.938	0.944	0.957	0.956
AUC	0.893	0.978	0.985	0.979	0.993	0.991	0.996	0.996	0.994	0.993	0.997	0.997

จากการทดลองงานวิจัย เพื่อทำการวิเคราะห์ถึงความน่าเชื่อถือของโมเดลว่ามีความแม่นยำ และถูกต้อง โดยสามารถเทียบจากรูปที่ 4.9 และตารางที่ 4.4 โดยจากตารางทั้งนั้นจะเป็นการแปลงค่าดัชนีการชี้วัดความน่าเชื่อถือออกมา โดยที่จะแสดงค่าที่ใช้การทำการตรวจสอบโมเดลเข้าไปและแสดงให้เห็นถึงความแตกต่างในแต่ละช่วงที่ใช้ในการเรียนรู้ (Epoch) โดยสามารถเห็นได้ว่าโมเดลนี้ไม่สามารถทำการเรียนรู้ได้อย่างตัวอย่างงานวิจัย [4] ที่ใช้การเรียนรู้เพียงแค่ 200 ครั้ง เนื่องจากค่าดัชนีการชี้วัดแยกแยะของโมเดลนี้นั้นยังคงมีค่าที่ต่ำมากจนทำให้โมเดลขาดความเชื่อถือไป แต่เมื่อทำการเรียนไปเรื่อยๆจนขนาดที่มากขึ้นจะพบเห็นได้ว่า ค่าดัชนีการชี้วัดความแม่นยำของโมเดลนั้น มีค่าเพิ่มขึ้นตามลำดับ โดยค่าจากตารางที่ 4.4 จะแสดงให้เห็นว่า ค่า Accuracy, Precision, Recall, และ AUC นั้นมีค่าสูงสุด ณ ครั้งแรกอยู่ที่ 0.957, 0.961, 0.954 และ 0.997 ตามลำดับ ในช่วงการเรียนรู้ครั้งที่ 9,000 ในขณะที่ค่า และในช่วงการเรียนรู้ครั้งที่ 10,000 และยังแสดงให้เห็นว่าเมื่อมีการเรียนรู้เพิ่มเติมเข้าไปต่อ นั้น ตัวเลขมีนัยยะที่การตกลงไปหน่อยหนึ่ง ถัดมาในส่วนของค่า Precision และ Recall นั้น มีค่าอัตราที่ค่อนข้างสูงทั้งสองค่า ทำให้การเลือกกว่าค่าไหนมีความแม่นยำนั้นจะทำการดูจากค่า F1-Score เนื่องจากค่านี้จะชี้วัดจากการเฉลี่ยกันระหว่างทั้งสองค่าซึ่งอยู่ที่ 0.957 โดยที่ค่าใช้ในการวัดผลมาจากข้อมูลที่ใช้ในการทดสอบ (Testing) ความถูกต้องเป็นจำนวน 1,200 ข้อมูล โดยประกอบไปด้วย การข้อมูลท่าทางอิริยาบถ นอนราบ, ยืน, เดิน, และนั่ง อย่าง 300 ข้อมูลในทดสอบ โดยที่ข้อมูลที่ใช้ในการตรวจสอบ (Validation) ระหว่างทำการเรียนรู้และทดสอบนั้นจะเป็นข้อมูลคนละส่วนกับข้อมูลที่ใช้ในการเรียนรู้ของโมเดล

ตารางที่ 4.5 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบผ่านค่า Precision, Recall และ F1-Score

	Precision	Recall	F1-Score	Support
Stand	0.86	0.94	0.9	300
Sleep	0.91	0.79	0.84	300
Sit	0.84	0.9	0.87	300
Walk	0.94	0.91	0.93	300
Accuracy			0.89	1,200
Macro Avg.	0.89	0.89	0.88	1,200
Weighted Avg.	0.89	0.89	0.88	1,200

จากตารางที่ 4.5 จะแสดงให้เห็นถึงหลังจากทำการตรวจสอบโมเดลด้วยข้อมูลทดสอบจะเห็นว่าค่า Precision และ Recall ในแต่ละท่าทางและอิริยาบถนั้น มีค่าที่สูงหรือมีค่าที่ใกล้เคียงกันมากเหมือนตอนที่โมเดลทำการเรียนรู้ ดังนั้นการวัดความถูกต้องในแต่ละท่าทางและอิริยาบถนั้น จะดูที่ค่า F1-Score แทนในแต่อิริยาบถ

```

=== Classifier model (full training set) ===

LibSVM wrapper, original code by Yasser EL-Manzalawy (= WLSVM)

Time taken to build model: 110.59 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.04 seconds

=== Summary ===

Correctly Classified Instances      1027      85.5833 %
Incorrectly Classified Instances    173       14.4167 %
Kappa statistic                    0.8075
Mean absolute error                0.1253
Root mean squared error            0.2321
Relative absolute error            33.3991 %
Root relative squared error        53.5944 %
Total Number of Instances         1200

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.929	0.105	0.753	0.929	0.832	0.773	0.977	0.937	Sit
	0.691	0.021	0.914	0.691	0.787	0.742	0.963	0.920	Sleep
	0.819	0.050	0.844	0.819	0.831	0.777	0.967	0.931	Stand
	0.977	0.017	0.952	0.977	0.964	0.952	0.997	0.982	Walk
Weighted Avg.	0.856	0.049	0.865	0.856	0.854	0.812	0.976	0.943	

รูปที่ 4.10 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบผ่าน SVM บน WEKA

```

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      1062      88.5 %
Incorrectly Classified Instances    138       11.5 %
Kappa statistic                    0.8466
Mean absolute error                0.0685
Root mean squared error            0.2188
Relative absolute error            18.2528 %
Root relative squared error        50.5307 %
Total Number of Instances         1200

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.926	0.063	0.836	0.926	0.879	0.836	0.974	0.904	Sit
	0.790	0.037	0.871	0.790	0.829	0.779	0.971	0.927	Sleep
	0.862	0.039	0.880	0.862	0.871	0.829	0.962	0.908	Stand
	0.957	0.014	0.957	0.957	0.957	0.942	0.998	0.994	Walk
Weighted Avg.	0.885	0.039	0.886	0.885	0.884	0.847	0.976	0.933	

รูปที่ 4.11 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบผ่าน NaiveBayes บน WEKA

เมื่อทำการเอาข้อมูลที่ใช้งานในโมเดลรูปการทำงานอื่นๆ ผ่านโปรแกรม WEKA โดยใช้โมเดลในรูปแบบของ SVM (Support Vector Machine) และ NaiveBayers จากการทดลองนั้นพบว่าค่าความแม่นยำนั้นอยู่ในระดับที่สูง แต่เมื่อเทียบจาก F-Measure หรือคือค่า F1-Score จะพบว่า โมเดลจากงานวิจัยตารางที่ 4.6 นั้นมีค่าที่สูงหรือเทียบเท่ากันอยู่ เมื่อเทียบกับรูปที่ 4.10 และรูปที่ 4.11

ตารางที่ 4.6 การเปรียบเทียบค่า F1-Score ในแต่ละโมเดล

	FFNN	SVM	Naïve Bayers
F1-Score	0.88	0.85	0.88
Weighted Avg			

บทที่ 5

สรุปผลการวิจัย

5.1 ข้อสรุปของงานวิจัย

จากผลการทดลองในบทที่ 4 จะพบว่าเมื่อทำการเปรียบเทียบกับโมเดลอื่นๆ ในข้อมูลที่ใช้การเรียนรู้และทดสอบ มีความแม่นยำในระดับมีความใกล้เคียงกันมาก ทำให้เห็นว่าข้อมูลที่นำมาใช้ในโมเดลต่าง ๆ นั้นมีประสิทธิภาพดีมากแค่ไหน จึงได้มีการใช้โมเดลอีกรูปแบบเข้าเปรียบเทียบเพิ่มเติม นั่นคือ Random Forest ซึ่งเป็นรูปแบบหนึ่งในรูปแบบของ Decision Tree โดยผลการทำงานของรูปแบบนี้มีค่าที่สูงมากเมื่อทำการทดลองผ่านโปรแกรม WEKA โดยมีค่า F1-Score อยู่ที่ 0.972

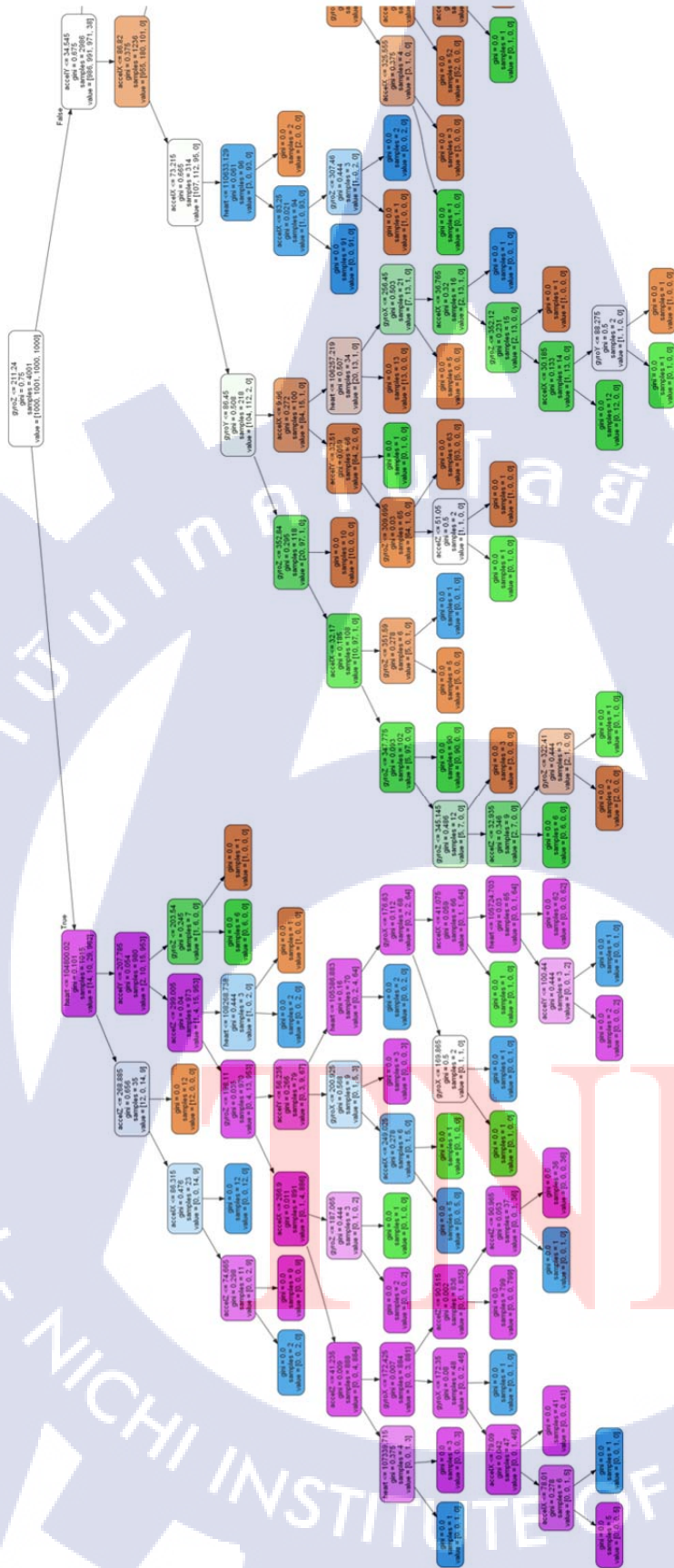
```
=== Classifier model (full training set) ===
RandomForest
Bagging with 100 iterations and base learner
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities
Time taken to build model: 0.99 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0.04 seconds
=== Summary ===
Correctly Classified Instances      1166      97.1667 %
Incorrectly Classified Instances    34        2.8333 %
Kappa statistic                    0.9622
Mean absolute error                 0.0245
Root mean squared error             0.0972
Relative absolute error             6.5219 %
Root relative squared error        22.4389 %
Total Number of Instances          1200

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.961	0.007	0.980	0.961	0.971	0.961	0.999	0.997	Sit
	0.990	0.012	0.963	0.990	0.976	0.969	0.999	0.998	Sleep
	0.956	0.012	0.963	0.956	0.960	0.946	0.998	0.996	Stand
	0.980	0.007	0.980	0.980	0.980	0.973	1.000	0.999	Walk
Weighted Avg.	0.972	0.009	0.972	0.972	0.972	0.962	0.999	0.997	

รูปที่ 5.1 Classification Report การคาดคะเนเริยียบจากข้อมูลทดสอบผ่าน Random Forest บน WEKA

แต่เมื่อเปรียบเทียบจากการทำงานแล้วจะพบว่าข้อมูลข้างอย่างนั้น กลับไม่ถูกนำไปใช้ในโมเดลแม้ว่าวิธีการใส่ค่าเข้าไปแล้ว จากรูปที่ 5.1 จะเป็นรูปแบบการทำงานของ Random Forest โดยที่ จะแสดงให้เห็นว่าค่า Temperature นั้นไม่ถูกนำไปใช้งานในการแบ่งแยกข้อมูลหรือระบุถึงท่าทาง กิจกรรม ทำให้แสดงให้เห็นว่าการทำงานในโมเดลอื่นๆ นั้น มีการทำงานที่มีความเป็นไปได้ที่อาจจะไม่ เลือกใช้ข้อมูลทั้งหมดในการที่เป็น Feature ในการเข้ากระบวนการเรียนรู้ซึ่งแตกต่างจาก FFNN ที่ใช้ใน งานวิจัยที่ใช้ Feature ทั้งหมดหมดในการเรียนรู้และเข้าถึงโหนดทั้งหมดใน Hidden Layer เนื่องจาก งานวิจัยชิ้นนี้ไม่ได้มีการระบุไปยังในส่วนในการสร้างหรือกำหนดค่าของ Dropout เข้าไปยังโมเดล



รูปที่ 5.2 รูปแบบการทำงานบางส่วนของ Random Forest

แต่จากการทดลองงานวิจัยจาก กลับแสดงให้เห็นว่าความแม่นยำในการเรียนรู้ของโมเดลนั้น ยังคงมีการเปลี่ยนแปลงในระดับที่ยังไม่นิ่งมากนัก ทำให้แสดงให้เห็นว่าโมเดลนี้ยังสามารถเรียนรู้ได้ต่อ จึงได้มีการเรียนรู้เพิ่มเติมจากเดิมที่ทำการเรียนรู้ให้โมเดลที่ 10000 epochs เป็น 20000 และ 50000 epochs

ตารางที่ 5.1 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบโดยใช้การเรียนรู้ที่ 20000 epochs

	Precision	Recall	F1-Score	Support
Stand	0.94	0.93	0.93	300
Sleep	0.91	0.83	0.87	300
Sit	0.83	0.95	0.89	300
Walk	0.96	0.9	0.93	300
Accuracy			0.9	1,200
Macro Avg.	0.91	0.9	0.9	1,200
Weighted Avg.	0.91	0.9	0.9	1,200

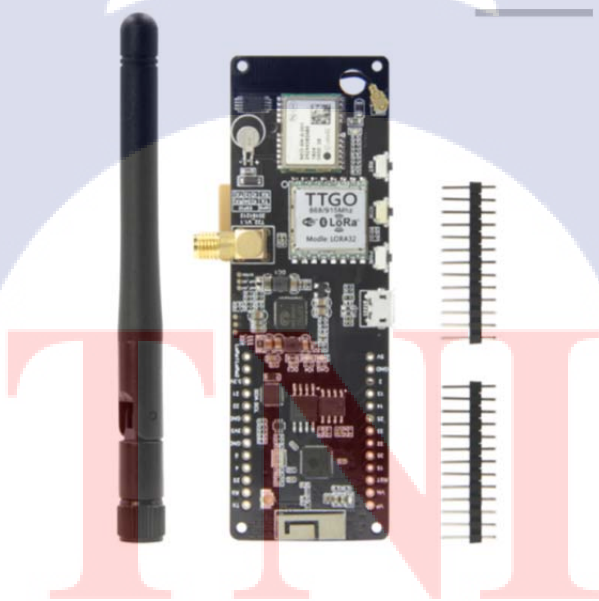
ตารางที่ 5.2 Classification Report การคาดคะเนอิริยาบถจากข้อมูลทดสอบโดยใช้การเรียนรู้ที่ 50000 epochs

	Precision	Recall	F1-Score	Support
Stand	0.89	0.93	0.91	300
Sleep	0.97	0.93	0.95	300
Sit	0.93	0.95	0.94	300
Walk	0.94	0.92	0.93	300
Accuracy			0.93	1,200
Macro Avg.	0.94	0.93	0.93	1,200
Weighted Avg.	0.94	0.93	0.93	1,200

โดยหลังจากที่ทำการเรียนรู้เพิ่มเติมขึ้นมานั้น พบว่าความแม่นยำในการทำนายผลจากข้อมูลเรียนรู้และทดลองชุดเดิมนั้น มีค่าที่เพิ่มขึ้นตามลำดับการเรียนรู้ จากผลการทดลองในตาราง 5.1 และตารางที่ 5.2 ทำให้เห็นว่าการทำงานของโมเดลถ้ามีการเรียนรู้มากขึ้น ยังมีความสามารถที่จะทำให้ทำนายค่าทางและกิจกรรมได้แม่นยำมากขึ้น ทั้งนี้ถ้ามีการ Tunning โมเดลโดยทำการเพิ่ม Hidden Layer หรือการเปลี่ยนค่า BatchSize อาจทำให้โมเดลมีความแม่นยำหรือการใช้เวลาในระยรอบเวลาในการเรียนรู้ที่ลดลงมากกว่าในการทดลองในขั้นนี้

5.2 การพัฒนางานวิจัยในอนาคต

จากงานวิจัยขั้นนี้ทำให้รู้ได้ถึงข้อมูลที่เอาเข้าสู่ระบบนั้นมีผลต่อการเรียนรู้และการทำนายกิจกรรมของผู้สวมใส่เป็นอย่างมากในการแยกแยะข้อมูลต่างๆ โดยในอนาคตสามารถพัฒนาในส่วน of Feature ที่เอาเข้าสู่ระบบสู่โมเดลโดยการการเพิ่มข้อมูลในส่วน of ระยะทางในการเคลื่อนที่เพิ่มเติมเข้ามา เนื่องจากอุปกรณ์ของ IoT นั้นมีเซ็นเซอร์ในการตรวจสอบหรือส่งข้อมูลในส่วน of พิกัดตำแหน่งได้ ทำให้สามารถนำระยะทางการเคลื่อนไหวจากจุดหนึ่งไปยังอีกจุดหนึ่งเข้ามาเป็นอีกข้อมูลหนึ่งในการแยกแยะกิจกรรมประเภทหยุดนิ่งหรือเคลื่อนไหวได้มากยิ่งขึ้น



รูปที่ 5.3 ESP 32 GPS NEO-6M SMA LORA 32 18650 ที่มีใส่เซ็นเซอร์ GPS เข้ามาในตัวบอร์ด

และต้องมีการเพิ่มข้อจำกัดในส่วน of อุนหภูมิของภายนอกเข้ามาเพิ่มเติมเข้ามาด้วย เนื่องจากระหว่างการเก็บข้อมูลในงานวิจัยนั้นพบว่า จุดอุปกรณ์ที่ติดตั้งวัดเซ็นเซอร์วัดอุณหภูมิในบริเวณนิ้วมือนั้น มีการเปลี่ยนแปลงไปควบคู่กับอุณหภูมิภายนอก ทำให้การเก็บข้อมูลในบ้างส่วนอาจมีความคลาด

เคลื่อนและไม่เป็นไปตามอุณหภูมิของร่างกายตามที่ควรจะเป็น เมื่อนำเข้าสู่โมเดลในการคาดคะเนอิริยาบถ และท่าทางในการเรียนรู้หรือทำนายนั่นอาจทำให้ได้ค่าที่คาดคะเนผิดเป็นได้ ซึ่งในส่วนนี้สามารถใช้ อุปกรณ์บางอย่างมาทดแทนการใช้วัดอัตราการเต้นหัวใจแทนที่นิ้วโดยการใช้ ECG Electrode Pad หรือ แผ่นติดหน้าอกเพื่อวัดคลื่นหัวใจได้แทน ซึ่งมีความเป็นไปได้สูงที่อาจเก็บค่าได้มีความแม่นยำมากกว่า



รูปที่ 5.4 Electrocardiogram (ECG) Heart Rate Monitor Kit AD8232

โดยที่การทำการเก็บข้อมูลนั้นควรกำหนดระยะเวลาหรือช่วงเวลาให้มีความชัดเจน ตัวอย่างเช่น การเก็บข้อมูลหลังตื่นนอนหรือก่อนนอน หลังการออกกำลังกาย หรือหลังการทำงานหรือกิจกรรมบางอย่าง เนื่องจากการเก็บข้อมูลที่มีความละเอียดอย่างอัตราการเต้นหัวใจนั้นมีการเปลี่ยนแปลงที่ไปทิศทางที่ละเอียดมาก ทำให้การกำหนดกิจกรรมก่อนที่ทำการเก็บข้อมูลนั้น มีความสำคัญตามมา รวมไปถึงช่วงอายุของตัวบุคคลในการเก็บข้อมูลด้วย เพราะการทำงานของร่างกายคนนั้นมีการเปลี่ยนแปลงไปตามอายุ อัตราการเต้นหัวใจและอุณหภูมิในการทำกิจกรรมต่างๆ ก็แปรผันตาม จึงเห็นว่าภายภาคหน้าในการทำงานทดลองนั้นต้องกำหนดช่วงเวลา กิจกรรม และอายุของผู้ทำการทดลองนั้นให้สอดคล้องและสัมพันธ์กัน รวมไปถึงการพัฒนาอุปกรณ์และระบบที่ทำการนำอุปกรณ์ติดตั้งเพิ่มไปยังบริเวณต่างๆ ตามร่างกายเพิ่มเติม ในการใช้เป็นข้อมูล Feature อีกส่วนหนึ่งในการเอาเข้าสู่โมเดล โดยอ้างอิงจากงานวิจัย [7] ที่พบว่าเมื่อทำการเอาข้อมูลจากจุดที่ติดตั้งอุปกรณ์ตามร่างกายเข้ามาช่วยทำการวิเคราะห์นั้น เพิ่มโอกาสในการทำนายรูปแบบกิจกรรมและอิริยาบถได้มากยิ่งขึ้น ซึ่งมีความเป็นไปได้ที่สามารถจัดปัญหา

ข้อมูลบางส่วนที่มีความคาดเคลื่อนออกไปได้ เนื่องจากงานวิจัยชิ้นนี้ได้พบเจอปัญหาอุณหภูมิที่ปลายนิ้ว นั้นมีการเปลี่ยนแปลงไปกับสภาพแวดล้อมภายนอก แต่ถ้าทำการใช้อุณหภูมิตามจุดต่างๆ ของร่างกาย เข้ามาเป็นข้อมูลเพิ่มเติมนั้น จะทำให้โมเดลสามารถวิเคราะห์ข้อมูลอุณหภูมิจากตำแหน่งต่างๆ หลายจุด ของร่างกายมาช่วยในการประมวลผลให้ให้โมเดลดียิ่งขึ้น



บรรณานุกรม

TNI

บรรณานุกรม

- [1] M. Kim et al., "Activity recognition using fully convolutional network from smartphone accelerometer," *International Conference on Information and Communication Technology Convergence*, ICIT 2018, Jeju, Korea (South), October 17-19, 2018, pp. 1,482-1,484.
- [2] N. Hardiyanti et al., "Classification of human activity based on sensor accelerometer and gyroscope using ensemble SVM method," *2nd East Indonesia Conference on Computer and Information Technology*, EIConCIT 2018, Makassar, Indonesia, November 6-7, 2018, pp. 304-307.
- [3] M. A. Ayu et al., S.S.O. Basamh, "Recognizing user activity based on accelerometer data from a mobile phone," *IEEE Symposium on Computers & Informatics*, ISCI 2011, Kuala Lumpur, Malaysia, March 20-23, 2011, pp. 617-621.
- [4] S. M. Lee et al., "Human activity recognition from accelerometer data using convolutional neural network," *IEEE International Conference on Big Data and Smart Computing*, BigComp 2017, Jeju, Korea (South), February 13-16, 2017, pp. 131-134.
- [5] F. Xiao et al., "Heart rate prediction model based on physical activities using evolutionary neural network," *Fourth International Conference on Genetic and Evolutionary Computing*, ICGEC 2010, Shenzhen, China, December 13-15, 2010, pp. 198-201.
- [6] S. Mekruksavanich et al., "Smartwatch-based sitting detection with human activity recognition for office workers syndrome," *International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering*, ECTI-NCON 2018, Chiang Rai, Thailand, February 25-28, 2018, pp. 160-164.
- [7] M. N. S. Zainudin et al., "Monitoring daily fitness activity using accelerometer sensor fusion," *IEEE International Symposium on Consumer Electronics*, ISCE 2017, Kuala Lumpur, Malaysia, November 14-15, 2017, pp. 35-36.

- [8] S. Sim et al., “Estimation of body temperature rhythm based on heart activity parameters in daily life,” *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, EMBC 2014, Chicago, IL, USA, August 26-30, 2014, pp. 2,245-2,248.
- [9] T. J. Malkinson, “Skin temperature response during cycle ergometry,” *Canadian Conference on Electrical and Computer Engineering*, CCECE 2002, Winnipeg, MB, Canada, May 12-15, 2002, pp. 1,123-1,128.
- [10] S. S. Thomas et al., “Sensing heart beat and body temperature digitally using Arduino,” *International Conference on Signal Processing, Communication, Power and Embedded System*, SCOPES 2016, Paralakhemundi, India, October 3-5, 2016, pp. 1,721-1,724.
- [11] P. Musale et al., “Lightweight gait based authentication technique for IoT using subconscious level activities,” *IEEE 4th World Forum on Internet of Things*, WF-IoT 2018, Singapore, February 5-8, 2018, pp. 564-567.
- [12] S. Sharma et al., “IoT based dipstick type engine oil level and impurities monitoring system : A portable online spectrophotometer,” *4th International Conference on Internet of Things : Smart Innovation and Usages*, IoT-SIU 2019, Ghaziabad, India, April 18-19, 2019, no page.
- [13] C.D. Ther, “Accuracy of Commercially Available Heart Rate Monitors in Athletes : A Prospective Study,” [Online]. Available from : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6732081/>. [Accessed : April 24, 2022].
- [14] Thaimobilecenter.com, “รู้จักเหล่าเซ็นเซอร์อัจฉริยะบนสมาร์ตโฟน มีอะไรบ้าง แต่ละชนิดทำงานอย่างไร?,” [Online]. Available : <https://www.thaimobilecenter.com/article-2557/understanding-sensors-on-mobile-phone-and-smartphone.asp>. [Accessed : 30 กรกฎาคม 2564].
- [15] จารุวิทย์ ประทานชีวินทร์ และอภิวัฒน์ จารุศักดิ์เดช, “เรียนรู้และทำความเข้าใจเรื่อง Support Vector Machine (SVM) คืออะไร,” [Online]. Available : <https://www.glurgeek.com/education/support-vector-machine/>. [Accessed : 24 เมษายน 2565].
- [16] Thaimobilecenter.com, “รู้จักเหล่าเซ็นเซอร์อัจฉริยะบนสมาร์ตโฟน มีอะไรบ้าง แต่ละชนิดทำงานอย่างไร?,” [Online]. Available : <https://www.thaimobilecenter.com/article-2557/understanding-sensors-on-mobile-phone-and-smartphone.asp>. [Accessed : 19 กรกฎาคม 2564].

- [17] Industrial Plug & Play Solutions, “*Raspberry PI 4 Model B 4GB Sensor Kit With 13X Popular Sensors*,” [Online]. Available from : <https://www.iot-store.com.au/products/raspberry-pi-4-model-b-4gb-sensor-kit>. [Accessed : July 19, 2021].
- [18] Shenzhen Qinda Electronics Co.,Ltd., “*GY-521 MPU-6050 MPU 6050 โมดูลเซ็นเซอร์ไจโรแวนะลือก 3 แกน + โมดูล Accelerometer 3 แกน*,” [Online]. Available : <https://thai.alibaba.com/product-detail/GY-521-MPU-6050-MPU6050-Module-62475468526.html>. [Accessed : 24 เมษายน 2565].
- [19] Smart-Prototyping, “*Pulse Oximeter and Heart Rate Sensor (MAX30102) (102090)*,” [Online]. Available from : <https://www.smart-prototyping.com/Pulse-Oximeter-and-Heart-Rate-Sensor-MAX30102>. [Accessed: April 24, 2022].
- [20] AllNewStep, “*WeMos ESP32 ESP-WROOM-32 + 18650 Battery Socket บอร์ด ESP32 รุ่น ESP-WROOM-32 พร้อมที่ใส่แบตเตอรี่รี 18650 และวงจรชาร์จในตัว*,” [Online]. Available : <https://www.allnewstep.com/product/1779/wemos-esp32-esp-wroom-32-18650-battery-socket-บอร์ด-esp-wroom-32-พร้อมที่ใส่แบตเตอรี่รี-1>. [Accessed : 24 เมษายน 2565].
- [21] S. Sharma, “*Activation Functions in Neural Networks*,” [Online]. Available from : <https://towardsdatascience.com/activation-functions-neural-networks-1-cbd9f8d91d6>. [Accessed : April 24, 2022].
- [22] ModuleMore, “*TTGO T-Beam V1.1 ESP32 WiFi Wireless Bluetooth Module ESP 32 GPS NEO LORA 32*,” [Online]. Available from : <https://www.modulemore.com/product/1118/ttgo-t-beam-v1-1-esp32-wifi-wireless-bluetooth-module-esp-32-gps-neo-lora-32>. [Accessed : April 24, 2022].
- [23] Arduitrionics, “*Electrocardiogram (ECG) Heart Rate Monitor Kit AD8232*,” [Online]. Available from : <https://www.arduitronics.com/product/1326/electrocardiogram-ecg-heart-rate-monitor-kit-ad8232>. [Accessed : April 24, 2022].

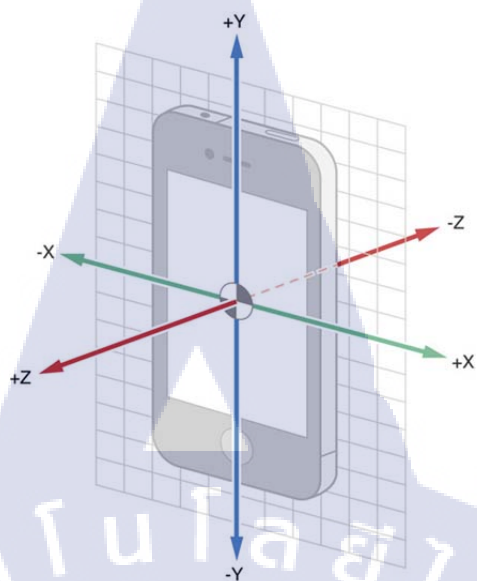
ภาคผนวก

TNI

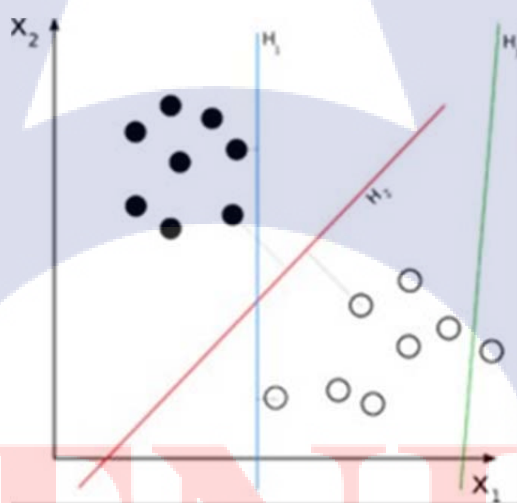
ภาคผนวก ก.

รูปภาพประกอบงานวิจัยอ้างอิงจากเว็บไซต์

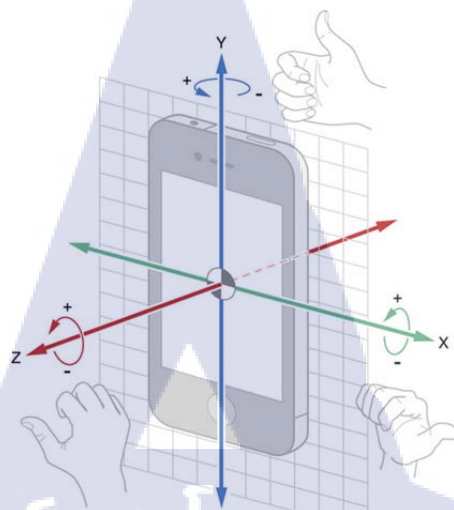
TNI



รูปที่ ก.1 รูปแบบการทำงานของ Accelerator Sensor [14]



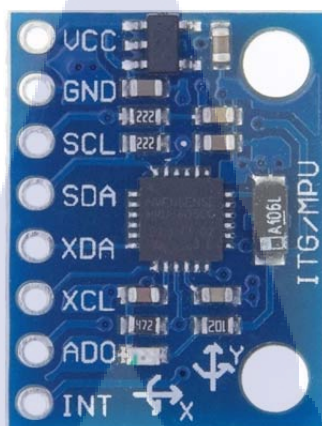
รูปที่ ก.2 การแบ่งกลุ่มในรูปแบบ Support Vector Machine [15]



รูปที่ ก.3 รูปแบบการทำงานของ Gyroscope Sensor [16]



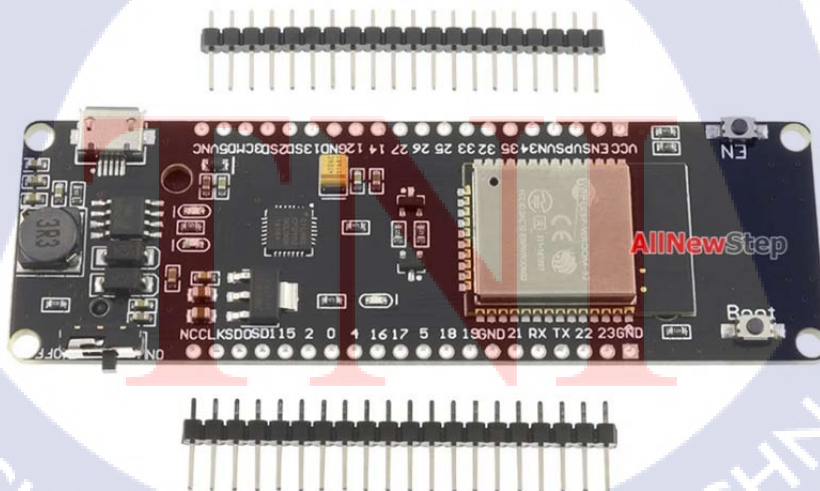
รูปที่ ก.4 เซ็นเซอร์ต่างๆ ที่ใช้บน IoTs [17]



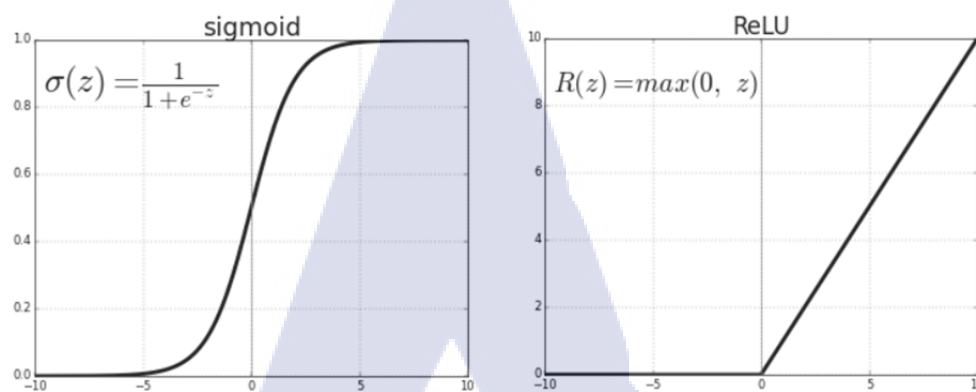
รูปที่ ก.5 Accelerator & Gyroscope Sensor รุ่น MPU6050 [18]



รูปที่ ก.6 Heartrate Sensor รุ่น Max30102 [19]



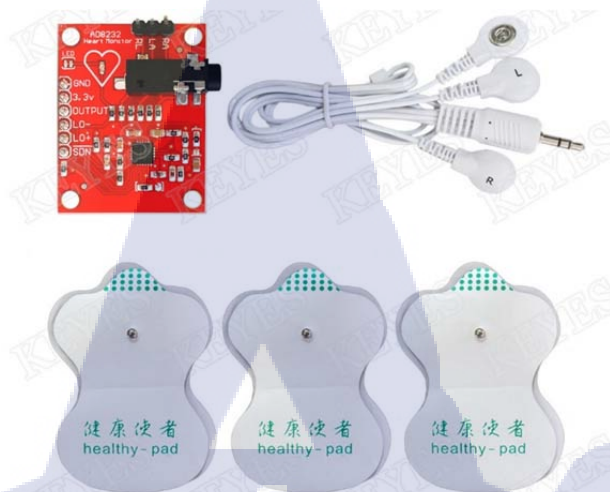
รูปที่ ก.7 ESP32-WROVER-B Development Board [20]



รูปที่ ก.8 รูปแบบการทำงานของ Sigmoid และ ReLU [21]



รูปที่ ก.9 ESP 32 GPS NEO-6M SMA LORA 32 18650 ที่มีใส่เซ็นเซอร์ GPS เข้ามาในตัวบอร์ด [22]



รูปที่ ก.10 Electrocardiogram (ECG) Heart Rate Monitor Kit AD8232 [23]

ภาคผนวก ข.
โค้ดส่วนของอุปกรณ์ IoT

TNI


```
#include <Wire.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include "MAX30105.h"
#include "heartRate.h"
#define WIFI_STA_NAME "home_2g_001"
#define WIFI_STA_PASS "023816542"
//For Home Wifi
const char* serverStand = "http://192.168.1.113:9090/main/action/stand";
const char* serverSleep = "http://192.168.1.113:9090/main/action/sleep";
const char* serverSit = "http://192.168.1.113:9090/main/action/sit";
const char* serverSitUp = "http://192.168.1.113:9090/main/action/situp";
const char* serverRun = "http://192.168.1.113:9090/main/action/run";
const char* serverWalk = "http://192.168.1.113:9090/main/action/walk";
const char* serverPushUp = "http://192.168.1.113:9090/main/action/pushup";
// START MPU
const uint8_t MPU_addr=0x68;
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
int minVal=265;
int maxVal=402;
double ac_x;
double ac_y;
double ac_z;
double gy_x;
double gy_y;
double gy_z;
// END MPU
HTTPClient http;
// HEART SESNSOR
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
```

```

byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
MAX30105 particleSensor;
String heartRate = "";
String tempRate = "";
String accelX = "";
String accelY = "";
String accelZ = "";
String gyroX = "";
String gyroY = "";
String gyroZ = "";
int timeTriggerCount = 0;
void setup() {
  // WIRE SETUP
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
  // HEART SETUP
  Serial.println("Start Heart Sensor");
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz
    speed
    {
      Serial.println("MAX30105 was not found. Please check wiring/power. ");
      while (1);
    }
  particleSensor.setup(); //Configure sensor with default settings

```

```

    particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate
sensor is running

    particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED

    Serial.println("End Heart Sensor");
// WIFI SETUP

    Serial.println("Start Wifi Setting");
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_STA_NAME, WIFI_STA_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    Serial.println("End Wifi Setting");
}

void loop() {
    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
    AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
    AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
    GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
    GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
    GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
    int xAng = map(AcX,minVal,maxVal,-90,90);
    int yAng = map(AcY,minVal,maxVal,-90,90);
    int zAng = map(AcZ,minVal,maxVal,-90,90);

```

```

int gxAng = map(GyX,minVal,maxVal,-90,90);
int gyAng = map(GyY,minVal,maxVal,-90,90);
int gzAng = map(GyZ,minVal,maxVal,-90,90);
ac_x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);
ac_y= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);
ac_z= RAD_TO_DEG * (atan2(-yAng, -xAng)+PI);
gy_x= RAD_TO_DEG * (atan2(-gyAng, -gzAng)+PI);
gy_y= RAD_TO_DEG * (atan2(-gxAng, -gzAng)+PI);
gy_z= RAD_TO_DEG * (atan2(-gyAng, -gxAng)+PI);
//===== HEART RATE in IR Value
long rawValueIr = particleSensor.getIR();
float temperature = particleSensor.readTemperature();
if(timeTriggerCount >=60){
    String httpRequestData = "{\"heartRate\":\\"" + heartRate + "\",\"temperature\":\\"" +
tempRate
                                + "\",\"accelX\":\\"" + accelX + "\",\"accelY\":\\"" + accelY
                                + "\",\"accelZ\":\\"" + accelZ + "\",\"gyroX\":\\"" + gyroX
                                + "\",\"gyroY\":\\"" + gyroY + "\",\"gyroZ\":\\"" + gyroZ + "\"}";
    Serial.println(httpRequestData);
    http.begin(serverSit);
    http.addHeader("Content-Type", "application/json");
    int httpResponseCode = http.POST(httpRequestData);
    resetValue();
}else{
    accelX = appendSpeedString(timeTriggerCount,accelX,ac_x);
    accelY = appendSpeedString(timeTriggerCount,accelY,ac_y);
    accelZ = appendSpeedString(timeTriggerCount,accelZ,ac_z);
    gyroX = appendSpeedString(timeTriggerCount,gyroX,gy_x);
    gyroY = appendSpeedString(timeTriggerCount,gyroY,gy_y);
    gyroZ = appendSpeedString(timeTriggerCount,gyroZ,gy_z);
    heartRate = appendHeartString(timeTriggerCount,heartRate,rawValueIr);
}

```

```

tempRate = appendTemperatureString(timeTriggerCount,tempRate,temperature);
timeTriggerCount++;
}
}
String appendSpeedString(int counter , String originalVal, int16_t appendVal){
    if(counter == 0){
        originalVal = originalVal + appendVal;
    }else{
        originalVal = originalVal + "," + appendVal;
    }
    return originalVal;
}
String appendHeartString(int counter , String originalVal, long appendVal){
    if(counter == 0){
        originalVal = originalVal + appendVal;
    }else{
        originalVal = originalVal + "," + appendVal;
    }
    return originalVal;
}
String appendTemperatureString(int counter , String originalVal, float appendVal){
    if(counter == 0){
        originalVal = originalVal + appendVal;
    }else{
        originalVal = originalVal + "," + appendVal;
    }
    return originalVal;
}
void resetValue(){
    heartRate = "";
    tempRate = "";
}

```



```
accelX = "";  
accelY = "";  
accelZ = "";  
gyroX = "";  
gyroY = "";  
gyroZ = "";  
timeTriggerCount = 0;  
}
```



ภาคผนวก ค.

โค้ดส่วน Model Python

TNI

```

import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report

def calculateF1(precision,recall):
    multiply = precision*recall
    plus = precision+recall
    return 2 * (multiply/plus)

def findMaxIndex(listData):
    maxVal=0
    index=0
    for i in range(len(listData)):
        if maxVal<=listData[i]:
            maxVal=listData[i]
            index=i
    return index

np.set_printoptions(formatter={"float" : '{:0.2f}'.format})
train_data = np.loadtxt("dataset/thesis.csv",delimiter=";",skiprows=1)
validate_data = np.loadtxt("dataset/thesis_validate.csv",delimiter=";",skiprows=1)
test_data = np.loadtxt("dataset/thesis_test.csv",delimiter=";",skiprows=1)
trainX = train_data[:,8]
trainY = train_data[:,4:]
validateX = validate_data[:,8]
validateY = validate_data[:,4:]
testX = test_data[:,8]
testY = test_data[:,4:]

```

```

model = Sequential()
model.add(Dense(8,input_dim=len(trainX[0,:]),activation="relu"))
model.add(Dense(256,activation="relu"))
model.add(Dense(4,activation="sigmoid"))
print(model.summary())
model.compile(optimizer='adam',
              loss=tf.keras.losses.BinaryCrossentropy(),
              metrics=['accuracy',
                      tf.keras.metrics.AUC(),
                      tf.keras.metrics.Recall(),
                      tf.keras.metrics.Precision()])

batch_size_data = 256
epochs_data = 10000
history =
model.fit(trainX,trainY,batch_size=batch_size_data,epochs=epochs_data,validation_data=(validateX, validateY),verbose=2)
model.save('thesis_model.h5')
prediction = model.predict(testX)
print("PREDICT")
print("STAND\tSLEEP\tSIT\tWALK")
# print(prediction[1:20])
print(prediction[1])
print("=====")
print("SHOULD BE")
print("STAND\tSLEEP\tSIT\tWALK")
print(testY[1])
# print(testY[1:20])
# print("INDEX :",np.argmax(prediction[1]))
print("=====")
print("PRINT RESULT")
# arrayToList = prediction[1].tolist()

```

```

# index = findMaxIndex(arrayToList)
labelInfo = ['Stand','Sleep','Sit','Walk']
index = np.argmax(prediction[1])
print("Result :",labelInfo[index])
accuracy_epochs_Oneh = history.history['accuracy'][99]
accuracy_epochs_Fiveh = history.history['accuracy'][499]
accuracy_epochs_Onek = history.history['accuracy'][999]
accuracy_epochs_Twok = history.history['accuracy'][1999]
accuracy_epochs_Threek = history.history['accuracy'][2999]
accuracy_epochs_Fourk = history.history['accuracy'][3999]
accuracy_epochs_Fivek = history.history['accuracy'][4999]
accuracy_epochs_Sixk = history.history['accuracy'][5999]
accuracy_epochs_Sevenk = history.history['accuracy'][6999]
accuracy_epochs_Eightk = history.history['accuracy'][7999]
accuracy_epochs_Ninek = history.history['accuracy'][8999]
accuracy_epochs_Tenk = history.history['accuracy'][9999]
print("===== accuracy =====");
print("100 : ",accuracy_epochs_Oneh)
print("500 : ",accuracy_epochs_Fiveh)
print("1000 : ",accuracy_epochs_Onek)
print("2000 : ",accuracy_epochs_Twok)
print("3000 : ",accuracy_epochs_Threek)
print("4000 : ",accuracy_epochs_Fourk)
print("5000 : ",accuracy_epochs_Fivek)
print("6000 : ",accuracy_epochs_Sixk)
print("7000 : ",accuracy_epochs_Sevenk)
print("8000 : ",accuracy_epochs_Eightk)
print("9000 : ",accuracy_epochs_Ninek)
print("10000 : ",accuracy_epochs_Tenk)

```

```

listAccuracy =
[accuracy_epochs_Oneh,accuracy_epochs_Fiveh,accuracy_epochs_Onek,
    accuracy_epochs_Twok,accuracy_epochs_Threek,accuracy_epochs_Fourk,
    accuracy_epochs_Fivek,accuracy_epochs_Sixk,accuracy_epochs_Sevenk,
    accuracy_epochs_Eightk,accuracy_epochs_Ninek,accuracy_epochs_Tenk]

auc_epochs_Oneh = history.history['auc'][99]
auc_epochs_Fiveh = history.history['auc'][499]
auc_epochs_Onek = history.history['auc'][999]
auc_epochs_Twok = history.history['auc'][1999]
auc_epochs_Threek = history.history['auc'][2999]
auc_epochs_Fourk = history.history['auc'][3999]
auc_epochs_Fivek = history.history['auc'][4999]
auc_epochs_Sixk = history.history['auc'][5999]
auc_epochs_Sevenk = history.history['auc'][6999]
auc_epochs_Eightk = history.history['auc'][7999]
auc_epochs_Ninek = history.history['auc'][8999]
auc_epochs_Tenk = history.history['auc'][9999]
print("===== AUC =====");
print("100 : ",auc_epochs_Oneh)
print("500 : ",auc_epochs_Fiveh)
print("1000 : ",auc_epochs_Onek)
print("2000 : ",auc_epochs_Twok)
print("3000 : ",auc_epochs_Threek)
print("4000 : ",auc_epochs_Fourk)
print("5000 : ",auc_epochs_Fivek)
print("6000 : ",auc_epochs_Sixk)
print("7000 : ",auc_epochs_Sevenk)
print("8000 : ",auc_epochs_Eightk)
print("9000 : ",auc_epochs_Ninek)
print("10000 : ",auc_epochs_Tenk)

```



```

listAuc = [auc_epochs_Oneh,auc_epochs_Fiveh,auc_epochs_Onek,
          auc_epochs_Twok,auc_epochs_Threek,auc_epochs_Fourk,
          auc_epochs_Fivek,auc_epochs_Sixk,auc_epochs_Sevenk,
          auc_epochs_Eightk,auc_epochs_Ninek,auc_epochs_Tenk]

recall_epochs_Oneh  = history.history['recall'][99]
recall_epochs_Fiveh = history.history['recall'][499]
recall_epochs_Onek  = history.history['recall'][999]
recall_epochs_Twok  = history.history['recall'][1999]
recall_epochs_Threek = history.history['recall'][2999]
recall_epochs_Fourk  = history.history['recall'][3999]
recall_epochs_Fivek  = history.history['recall'][4999]
recall_epochs_Sixk   = history.history['recall'][5999]
recall_epochs_Sevenk = history.history['recall'][6999]
recall_epochs_Eightk = history.history['recall'][7999]
recall_epochs_Ninek  = history.history['recall'][8999]
recall_epochs_Tenk   = history.history['recall'][9999]
print("===== recall =====");
print("100 : ",recall_epochs_Oneh)
print("500 : ",recall_epochs_Fiveh)
print("1000 : ",recall_epochs_Onek)
print("2000 : ",recall_epochs_Twok)
print("3000 : ",recall_epochs_Threek)
print("4000 : ",recall_epochs_Fourk)
print("5000 : ",recall_epochs_Fivek)
print("6000 : ",recall_epochs_Sixk)
print("7000 : ",recall_epochs_Sevenk)
print("8000 : ",recall_epochs_Eightk)
print("9000 : ",recall_epochs_Ninek)
print("10000 : ",recall_epochs_Tenk)
listRecall = [recall_epochs_Oneh,recall_epochs_Fiveh,recall_epochs_Onek,
             recall_epochs_Twok,recall_epochs_Threek,recall_epochs_Fourk,

```

```

recall_epochs_Fivek,recall_epochs_Sixk,recall_epochs_Sevenk,
recall_epochs_Eightk,recall_epochs_Ninek,recall_epochs_Tenk]

precision_epochs_Oneh = history.history['precision'][99]
precision_epochs_Fiveh = history.history['precision'][499]
precision_epochs_Onek = history.history['precision'][999]
precision_epochs_Twok = history.history['precision'][1999]
precision_epochs_Threek = history.history['precision'][2999]
precision_epochs_Fourk = history.history['precision'][3999]
precision_epochs_Fivek = history.history['precision'][4999]
precision_epochs_Sixk = history.history['precision'][5999]
precision_epochs_Sevenk = history.history['precision'][6999]
precision_epochs_Eightk = history.history['precision'][7999]
precision_epochs_Ninek = history.history['precision'][8999]
precision_epochs_Tenk = history.history['precision'][9999]
print("===== precision =====");
print("100 : ",precision_epochs_Oneh)
print("500 : ",precision_epochs_Fiveh)
print("1000 : ",precision_epochs_Onek)
print("2000 : ",precision_epochs_Twok)
print("3000 : ",recall_epochs_Threek)
print("4000 : ",precision_epochs_Fourk)
print("5000 : ",precision_epochs_Fivek)
print("6000 : ",precision_epochs_Sixk)
print("7000 : ",precision_epochs_Sevenk)
print("8000 : ",precision_epochs_Eightk)
print("9000 : ",precision_epochs_Ninek)
print("10000 : ",precision_epochs_Tenk)
listPrecision =
[precision_epochs_Oneh,precision_epochs_Fiveh,precision_epochs_Onek,
precision_epochs_Twok,precision_epochs_Threek,precision_epochs_Fourk,
precision_epochs_Fivek,precision_epochs_Sixk,precision_epochs_Sevenk,

```

```

precision_epochs_Eightk,precision_epochs_Ninek,precision_epochs_Tenk]
f1_epochs_Oneh = calculateF1(precision_epochs_Oneh ,recall_epochs_Oneh )
f1_epochs_Fiveh = calculateF1(precision_epochs_Fiveh ,recall_epochs_Fiveh )
f1_epochs_Onek = calculateF1(precision_epochs_Onek ,recall_epochs_Onek )
f1_epochs_Twok = calculateF1(precision_epochs_Twok ,recall_epochs_Twok )
f1_epochs_Threek = calculateF1(precision_epochs_Threek,recall_epochs_Threek)
f1_epochs_Fourk = calculateF1(precision_epochs_Fourk ,recall_epochs_Fourk )
f1_epochs_Fivek = calculateF1(precision_epochs_Fivek ,recall_epochs_Fivek )
f1_epochs_Sixk = calculateF1(precision_epochs_Sixk ,recall_epochs_Sixk )
f1_epochs_Sevenk = calculateF1(precision_epochs_Sevenk,recall_epochs_Sevenk)
f1_epochs_Eightk = calculateF1(precision_epochs_Eightk,recall_epochs_Eightk)
f1_epochs_Ninek = calculateF1(precision_epochs_Ninek ,recall_epochs_Ninek )
f1_epochs_Tenk = calculateF1(precision_epochs_Tenk ,recall_epochs_Tenk )
print("===== F1 =====");
print("100 : ",f1_epochs_Oneh)
print("500 : ",f1_epochs_Fiveh)
print("1000 : ",f1_epochs_Onek)
print("2000 : ",f1_epochs_Twok)
print("3000 : ",f1_epochs_Threek)
print("4000 : ",f1_epochs_Fourk)
print("5000 : ",f1_epochs_Fivek)
print("6000 : ",f1_epochs_Sixk)
print("7000 : ",f1_epochs_Sevenk)
print("8000 : ",f1_epochs_Eightk)
print("9000 : ",f1_epochs_Ninek)
print("10000 : ",f1_epochs_Tenk)
listFOne = [f1_epochs_Oneh,f1_epochs_Fiveh,f1_epochs_Onek,
            f1_epochs_Twok,f1_epochs_Threek,f1_epochs_Fourk,
            f1_epochs_Fivek,f1_epochs_Sixk,f1_epochs_Sevenk,
            f1_epochs_Eightk,f1_epochs_Ninek,f1_epochs_Tenk]

```

```
sns.set_theme()
plt.plot(listAccuracy)
plt.plot(listAuc)
plt.plot(listRecall)
plt.plot(listPrecision)
plt.plot(listFOne)
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Accuracy', 'AUC', 'Recall', 'Precision', 'F1'], loc='upper left')
plt.show()
prediction=np.argmax(prediction, axis=1)
testY=np.argmax(testY, axis=1)
print(classification_report(testY,prediction,target_names= ['Stand','Sleep','Sit','Walk']))
```

ภาคผนวก ง.
โค้ดส่วนการแปลงค่าข้อมูล

TNI

```

import pandas as pd
import psycopg2
def convertMean(dataValue):
    data = dataValue.split(',')
    result = 0
    for i in range(len(data)):
        result += float(data[i])
    result = result / 60
    return round(result,2)

conn = psycopg2.connect("dbname=postgres user=postgres password=password")
cur = conn.cursor()
cur.execute("select
val_accel_aix_x,val_accel_aix_y,val_accel_aix_z,val_gyro_aix_x,val_gyro_aix_y,val_gyro
_aix_z,val_heart_rate,val_temp, Result_Data from ( select *, 'Stand' as Result_Data
from thesis.t_action_stand where val_set <= 1000 union select *, 'Sit' as Result_Data
from thesis.t_action_sit where val_set <= 1000 union select *, 'Sleep' as Result_Data
from thesis.t_action_sleep where val_set <= 1005 union select *, 'Walk' as
Result_Data from thesis.t_action_walk where val_set <= 1000) temp_table order by
temp_table.Result_Data,temp_table.val_set")
cur.execute("select
val_accel_aix_x,val_accel_aix_y,val_accel_aix_z,val_gyro_aix_x,val_gyro_aix_y,val_gyro
_aix_z,val_heart_rate,val_temp, Result_Data from ( select *, 'Stand' as Result_Data
from thesis.t_action_stand where val_set > 1300 union select *, 'Sit' as Result_Data
from thesis.t_action_sit where val_set > 1300 union select *, 'Sleep' as Result_Data
from thesis.t_action_sleep where val_set > 1300 union select *, 'Walk' as Result_Data
from thesis.t_action_walk where val_set > 1300) temp_table order by
temp_table.Result_Data,temp_table.val_set")
cur.execute("select
val_accel_aix_x,val_accel_aix_y,val_accel_aix_z,val_gyro_aix_x,val_gyro_aix_y,val_gyro
_aix_z,val_heart_rate,val_temp, Result_Data from ( select *, 'Stand' as Result_Data
from thesis.t_action_stand where val_set > 1000 and val_set <=1300 union select

```



```

*, 'Sit' as Result_Data from thesis.t_action_sit where val_set > 1000 and val_set
<=1300 union select *, 'Sleep' as Result_Data from thesis.t_action_sleep where
val_set > 1000 and val_set <=1300 union select *, 'Walk' as Result_Data from
thesis.t_action_walk where val_set > 1000 and val_set <=1300) temp_table order by
temp_table.Result_Data,temp_table.val_set")
records = cur.fetchall()
data = []
data_AccX_sl,data_AccY_sl,data_AccZ_sl = [],[],[]
data_AccX_st,data_AccY_st,data_AccZ_st = [],[],[]
data_AccX_si,data_AccY_si,data_AccZ_si = [],[],[]
data_AccX_wa,data_AccY_wa,data_AccZ_wa = [],[],[]
data_GyrX_sl,data_GyrY_sl,data_GyrZ_sl = [],[],[]
data_GyrX_st,data_GyrY_st,data_GyrZ_st = [],[],[]
data_GyrX_si,data_GyrY_si,data_GyrZ_si = [],[],[]
data_GyrX_wa,data_GyrY_wa,data_GyrZ_wa = [],[],[]
data_Temp_sl = []
data_Temp_si = []
data_Temp_st = []
data_Temp_wa = []
data_Heart_sl = []
data_Heart_si = []
data_Heart_st = []
data_Heart_wa = []
for record in records:
    accelX = convertMean(record[0])
    accelY = convertMean(record[1])
    accelZ = convertMean(record[2])
    gyroX = convertMean(record[3])
    gyroY = convertMean(record[4])
    gyroZ = convertMean(record[5])
    heart = convertMean(record[6])

```

```

temperature = convertMean(record[7])
result = record[8]
result_stand = 1 if record[8] == "Stand" else 0
result_sleep = 1 if record[8] == "Sleep" else 0
result_sit = 1 if record[8] == "Sit" else 0
result_walk = 1 if record[8] == "Walk" else 0
if record[8] == "Stand":
    data_AccX_st.append(accelX)
    data_AccY_st.append(accelY)
    data_AccZ_st.append(accelZ)
    data_GyrX_st.append(gyroX)
    data_GyrY_st.append(gyroY)
    data_GyrZ_st.append(gyroZ)
    data_Temp_st.append(temperature)
    data_Heart_st.append(heart)
elif record[8] == "Sleep":
    data_AccX_sl.append(accelX)
    data_AccY_sl.append(accelY)
    data_AccZ_sl.append(accelZ)
    data_GyrX_sl.append(gyroX)
    data_GyrY_sl.append(gyroY)
    data_GyrZ_sl.append(gyroZ)
    data_Temp_sl.append(temperature)
    data_Heart_sl.append(heart)
elif record[8] == "Sit":
    data_AccX_si.append(accelX)
    data_AccY_si.append(accelY)
    data_AccZ_si.append(accelZ)
    data_GyrX_si.append(gyroX)
    data_GyrY_si.append(gyroY)
    data_GyrZ_si.append(gyroZ)

```

```

data_Temp_si.append(temperature)
data_Heart_si.append(heart)
elif record[8] == "Walk":
    data_AccX_wa.append(accelX)
    data_AccY_wa.append(accelY)
    data_AccZ_wa.append(accelZ)
    data_GyrX_wa.append(gyroX)
    data_GyrY_wa.append(gyroY)
    data_GyrZ_wa.append(gyroZ)
    data_Temp_wa.append(temperature)
    data_Heart_wa.append(heart)
subData = {"accelX" : accelX, "accelY" : accelY, "accelZ" : accelZ
            , "gyroX" : gyroX , "gyroY" : gyroY , "gyroZ" : gyroZ
            , "heart" : heart , "temperature" : temperature , "result" : result,
            "result_stand" : result_stand,"result_sleep" : result_sleep,"result_sit" :
result_sit,"result_walk" : result_walk}
    data.append(subData)
df = pd.DataFrame(data)
df=df[["accelX","accelY","accelZ","gyroX","gyroY","gyroZ","heart","temperature",
      "result_stand","result_sleep","result_sit","result_walk"]]
df.to_csv(r'.\dataset\thesis.csv',header=True, index=False)
df.to_csv(r'.\dataset\thesis_validate.csv',header=True, index=False)
df.to_csv(r'.\dataset\thesis_test.csv',header=True, index=False)

```

ภาคผนวก จ.
วงจรการเชื่อมต่ออุปกรณ์ IoT

TNI

