

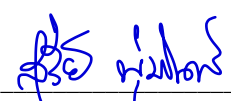
รายงานโครงงานวิศวกรรมไฟฟ้า วิชา 2102499

การวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่อง
บนอุปกรณ์ประมวลผลข้อมูล

Human movement analysis using machine learning
on edge device

นายธีรวัฒน์ เลิศอัมพรวิทย์ เลขประจำตัว 6430183721
อาจารย์ที่ปรึกษา ผศ.ดร.สุริย์ พุ่มรินทร์

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2567

ลงชื่ออาจารย์ที่ปรึกษาหลัก  (ผศ.ดร.สุริย์ พุ่มรินทร์) วันที่ 25 เม.ย. 2568	ลงชื่ออาจารย์ที่ปรึกษาร่วม (ถ้ามี) _____ (_____) วันที่ _____	ลงชื่อตัวแทนบริษัท (เฉพาะนิสิตใน โปรแกรมความเชื่อมโยงอุตสาหกรรม) _____ (_____) วันที่ _____
--	--	---

บทคัดย่อ

โครงการนี้นำเสนอการวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล โดยใช้ตัวรับรู้ความเร่งและตัวรับรู้โรสโคป จากไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART แล้วแสดงผลข้อมูลผ่านทาง Web Server แล้วใช้แพลตฟอร์ม Edge Impulse ในการวิเคราะห์ข้อมูลและสร้างแบบจำลองการเรียนรู้ของเครื่อง โดยใช้แบบจำลองการวิเคราะห์สเปกตรัม โดยการวิเคราะห์ด้วย FFT เพื่อแปลงข้อมูลจากโดเมนเวลาไปเป็นโดเมนความถี่ และการจำแนกประเภทด้วยเครือข่ายประสาท ซึ่งสามารถจำแนกการเคลื่อนไหวออกเป็น 3 กลุ่ม ได้แก่ กลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสี่ยงอันตราย เมื่อนำแบบจำลองมาใช้งานกับบอร์ดจริง พบว่าสามารถทำนายผลการเคลื่อนไหวของมนุษย์ผ่านทาง Web Server ได้ในแต่ละช่วงเวลา

คำสำคัญ: การเรียนรู้ของเครื่อง, การวิเคราะห์การเคลื่อนไหว, Arduino Nano, Edge Impulse, อุปกรณ์ประมวลผลข้อมูล

Abstract

This project presents human motion analysis using machine learning on data processing devices using accelerometer and gyroscope sensors from Arduino Nano 33 BLE Sense and Arduino Nano 33 IoT microcontrollers. The data is communicated between the boards via UART and displayed via Web Server. The Edge Impulse platform is used to analyze the data and create a machine learning model using a spectrum analysis model. FFT analysis is used to convert data from the time domain to the frequency domain and neural network classification, which can classify the movement into 3 groups: idle, move and risk. When the model is used with the actual board, it can predict the human movement results via Web Server at each period.

Keywords: Machine Learning, Human Movement Analysis, Arduino Nano, Edge Impulse, Edge Device

สารบัญ

บทคัดย่อ	ก
สารบัญ	ข
1. บทนำ	1
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ผลลัพธ์ที่คาดหวังจากโครงการ	2
1.5 องค์ความรู้ทางวิศวกรรมที่นำมาประยุกต์ใช้	2
2. หลักการและทฤษฎีที่เกี่ยวข้อง	3
2.1 การเรียนรู้ของเครื่อง (Machine Learning)	3
2.2 การเรียนรู้เชิงลึก (Deep Learning)	4
2.3 การเรียนรู้ของเครื่องขนาดเล็ก (Tiny Machine Learning)	9
2.4 การประเมินแบบจำลองการเรียนรู้ของเครื่อง	9
2.5 แพลตฟอร์ม Edge Impulse	12
2.6 ไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense	15
2.7 ไมโครคอนโทรลเลอร์ Arduino Nano 33 IoT	17
2.8 หน่วยวัดแรงเฉื่อย (IMU)	18
3. แนวทางการดำเนินงาน	20
3.1 การติดตั้งอุปกรณ์สำหรับรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์	20
3.2 การรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ (Training Phase)	21
3.3 การวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse	22

3.4 การทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ (Testing Phase)	23
4. ผลลัพธ์จากการดำเนินการเบื้องต้น	25
4.1 ผลลัพธ์จากการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์	25
4.2 ผลลัพธ์จากการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse	26
4.3 ผลลัพธ์จากการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์	31
5. บทสรุป	33
5.1 บทสรุปการทำโครงการจนถึงปัจจุบัน	33
5.2 แผนการดำเนินงาน	33
5.3 ปัญหา อุปสรรค และแนวทางแก้ไข	34
6. กิตติกรรมประกาศ	35
7. เอกสารอ้างอิง	35
8. ภาคผนวก	39
8.1 Readme File สำหรับอธิบายระบบ	39
8.2 Source Code สำหรับการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse	40

1. บทนำ

1.1 ที่มาและความสำคัญของโครงการ

การเคลื่อนไหวของมนุษย์สามารถพบได้ในชีวิตประจำวัน เนื่องจากมนุษย์ต้องทำกิจกรรมต่างๆที่เกิดขึ้นในชีวิตประจำวัน เช่น การนั่ง การนอน การเดิน การวิ่ง เป็นต้น [1] นอกจากการเคลื่อนไหวในรูปแบบที่กล่าวมานี้ ยังพบการเคลื่อนไหวในรูปแบบที่อาจเกิดอันตรายได้ เช่น การล้ม การลื่น การสะดุด เป็นต้น [2] ซึ่งปัจจัยเสี่ยงที่ทำให้เกิดการเคลื่อนไหวในรูปแบบดังกล่าวเกิดจากการทรงตัวที่ไม่ดีหรือสูญเสียการทรงตัว [3] จึงควรจะมีการแจ้งเตือนเกี่ยวกับการเคลื่อนไหวของมนุษย์ไปยังอุปกรณ์สื่อสารข้อมูลต่างๆ เพื่อให้สามารถช่วยเหลือคนที่เป็นกลุ่มเสี่ยงที่อาจเกิดอันตรายได้

โครงการนี้เป็นการวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่อง (Machine Learning) เพื่อติดตามสถานะการเคลื่อนไหวของมนุษย์ โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT ในการเก็บค่าการเคลื่อนไหว 6 แกน จากตัวรับรู้ความเร่ง (Accelerometer) และตัวรับรู้ไจโรสโคป (Gyroscope) ที่มีอยู่บนบอร์ด แล้วประมวลผลด้วยการเรียนรู้ของเครื่อง เพื่อจำแนกการเคลื่อนไหวของมนุษย์ แล้วแสดงผลในอุปกรณ์สื่อสารข้อมูลต่อไป

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการทำงานและเก็บค่าการเคลื่อนไหว จากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป ในไมโครคอนโทรลเลอร์
2. เพื่อศึกษาและสร้างแบบจำลองการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล โดยใช้แบบจำลองการวิเคราะห์สเปกตรัมและการจัดประเภท
3. เพื่อจำแนกรูปแบบการเคลื่อนไหวของมนุษย์ จากการวัดค่าโดยใช้ตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป ในไมโครคอนโทรลเลอร์ โดยใช้การเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล

1.3 ขอบเขตของโครงการ

1. ใช้ไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT สำหรับเก็บค่าการเคลื่อนไหว 6 แกน จากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป
2. ใช้โปรแกรมสำหรับวิเคราะห์ข้อมูลด้วยการเรียนรู้ของเครื่อง ชื่อว่า Edge Impulse โดยใช้แบบจำลองการวิเคราะห์สเปกตรัมและการจัดประเภท

3. การจำแนกการเคลื่อนไหวของมนุษย์จะแบ่งเป็น 3 กลุ่ม ได้แก่
 - กลุ่มอยู่นิ่ง (Idle) เป็นกลุ่มที่ไม่มีการเคลื่อนไหวของร่างกาย
 - กลุ่มเคลื่อนไหว (Move) เป็นกลุ่มที่มีการเคลื่อนไหวของร่างกายจากที่หนึ่งไปยังอีกที่หนึ่ง
 - กลุ่มเสี่ยงอันตราย (Risk) เป็นกลุ่มที่มีการเคลื่อนไหวของร่างกายจากที่หนึ่งไปยังอีกที่หนึ่ง โดยที่ไม่มี ความมั่นคงและความสมมาตรของการเดิน ซึ่งเป็นปัจจัยเสี่ยงที่ทำให้เกิดการลื่นล้มได้ [4,5]

1.4 ผลลัพธ์ที่คาดหวังจากโครงการ

1. สามารถใช้ไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT ในการ รวบรวมและวิเคราะห์ข้อมูลการเคลื่อนไหว จากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป แล้วแสดงผล ข้อมูลไปยังเครื่องคอมพิวเตอร์
2. สามารถจำแนกรูปแบบการเคลื่อนไหวของมนุษย์ได้โดยการใช้การเรียนรู้ของเครื่องบนอุปกรณ์ ประมวลผลข้อมูลแล้วออกเป็น 3 กลุ่ม ได้แก่ กลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสี่ยงอันตราย

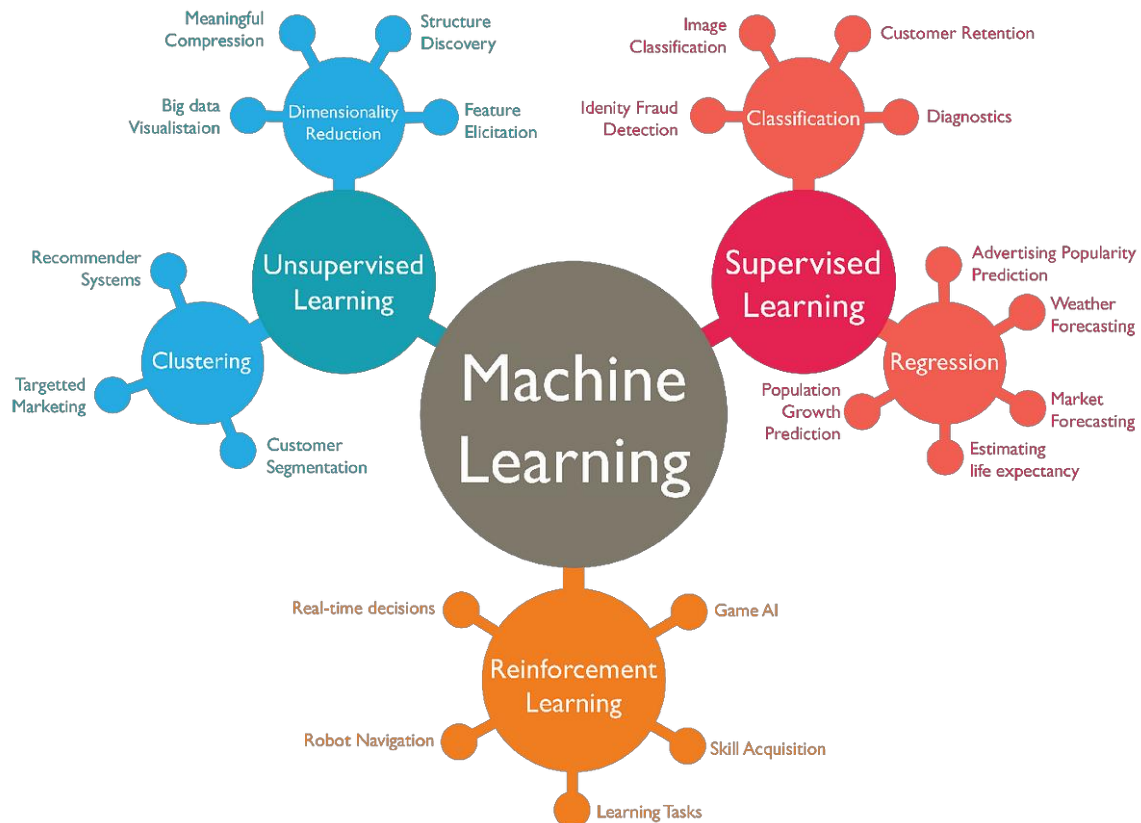
1.5 องค์ความรู้ทางวิศวกรรมที่นำมาประยุกต์ใช้

1. ความรู้ทางระบบฝังตัวเบื้องต้น (Embedded System) และตัวรับรู้ (Sensor)
2. ความรู้เกี่ยวกับการเรียนรู้ของเครื่อง และการเรียนรู้เชิงลึก (Deep Learning)
3. ทักษะการเขียนโปรแกรมคอมพิวเตอร์ ได้แก่ ภาษา Python และภาษา C++

2. หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 การเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง คือ การที่ทำให้คอมพิวเตอร์สามารถเรียนรู้ได้ด้วยตนเองจากข้อมูลและสภาพแวดล้อมที่ได้รับมา โดยที่ไม่มีมนุษย์มาเขียนโปรแกรมหรือคำสั่งเพิ่มเติม [3,4,5] โดยการเรียนรู้ของเครื่องสามารถจำแนกได้เป็น 3 รูปแบบ ดังที่แสดงในรูปที่ 1 ดังนี้



รูปที่ 1 ประเภทของการเรียนรู้ของเครื่อง [3]

2.1.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning)

เป็นการเรียนรู้ที่ต้องมีการสอน โดยให้คอมพิวเตอร์เรียนรู้จากข้อมูลที่เข้าไปสอน ซึ่งข้อมูลที่นำเข้าไปสอน เรียกว่า ชุดข้อมูลฝึก (Training Data) ประกอบด้วย 2 ส่วน คือ ลักษณะของข้อมูล (Feature) และชื่อของข้อมูล (Label) [6,7,8] การเรียนรู้แบบมีผู้สอนสามารถแบ่งได้ 2 กลุ่ม ดังนี้

1. การถดถอย (Regression) คือ การทำนายค่าเป็นตัวเลข เช่น ทำนายการเปลี่ยนแปลงของราคาหุ้น [7,9]
2. การจำแนกประเภท (Classification) คือ การจำแนกหมวดหมู่ เช่น การจำแนกสัตว์ว่าเป็นสุนัขหรือแมว [7,9]

2.1.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning)

เป็นการเรียนรู้ที่ไม่ต้องมีการสอน คอมพิวเตอร์สามารถเรียนรู้ได้ด้วยตัวเอง โดยไม่ต้องมีคำตอบของข้อมูล เป็นการอธิบายถึงลักษณะของข้อมูล แต่ไม่ได้บอกชื่อของข้อมูล [6,7,8] การเรียนรู้แบบไม่มีผู้สอนสามารถแบ่งได้ 2 กลุ่ม ดังนี้

1. การจับกลุ่ม (Clustering) คือ การจัดกลุ่มข้อมูลที่มีความสัมพันธ์กัน [7]
2. การลดมิติ (Dimensionality Reduction) คือ การลดมิติหรือความซับซ้อนของข้อมูล เพื่อวิเคราะห์ข้อมูลได้ง่ายขึ้น [7]

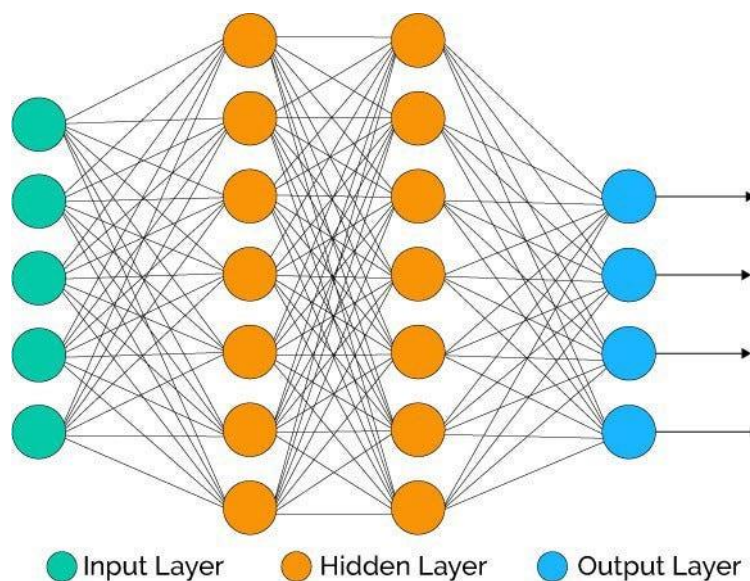
2.1.3 การเรียนรู้แบบเสริมแรง (Reinforcement Learning)

เป็นการเรียนรู้จากสภาพแวดล้อม โดยอาศัยการป้อนกลับแล้วให้ระบบเรียนรู้และปรับปรุงตัวเอง เช่น ระบบการทรงตัวของหุ่นยนต์ ระบบจะทำการป้อนกลับแล้วนำค่าข้อมูลในการทรงตัวมาปรับปรุง ทำให้สามารถทรงตัวได้อย่างมีประสิทธิภาพ [7,8,9]

2.2 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก คือ การใช้เครือข่ายประสาทเทียม (Artificial Neural Network) ที่มีความลึกหลายชั้น เลียนแบบเซลล์เครือข่ายสมองของมนุษย์ ในการเรียนรู้จากข้อมูลจำนวนมากมหาศาล [10]

2.2.1 โครงสร้างของการเรียนรู้เชิงลึก



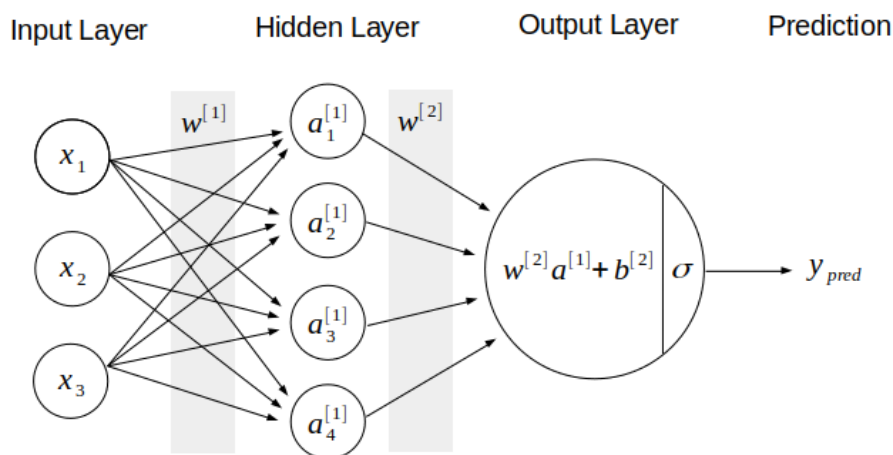
รูปที่ 2 โครงสร้างของการเรียนรู้เชิงลึก [13]

การเรียนรู้เชิงลึกจะใช้เครือข่ายประสาทเทียม โดยจะมีเซลล์ประสาท (Neurons) ซึ่งแสดงด้วยสัญลักษณ์วงกลม ดังรูปที่ 2 โดยเซลล์ประสาทจะเชื่อมต่อถึงกัน [11] องค์ประกอบของเซลล์ประสาท แบ่งออกเป็น 3 กลุ่ม ดังนี้

1. ชั้นข้อมูลนำเข้า (Input Layer) ทำหน้าที่รับข้อมูลนำเข้า (Input) จากภายนอก แล้วส่งข้อมูลไปยังชั้นระหว่างกลาง (Hidden Layer) ชั้นแรก [11,12]
2. ชั้นระหว่างกลาง ทำหน้าที่ประมวลผลข้อมูล ซึ่งสามารถมีได้มากกว่า 1 ชั้น โดยทำการคำนวณข้อมูลที่รับจากชั้นก่อนหน้า แล้วส่งไปยังชั้นถัดไป [11,12]
3. ชั้นข้อมูลส่งออก (Output Layer) ทำหน้าที่ส่งผลลัพธ์ (Output) หรือผลจากการทำนายข้อมูลออกไป [11,12]

2.2.2 การทำงานของการเรียนรู้เชิงลึก

การเรียนรู้เชิงลึกมีการทำงาน 4 ส่วน คือ ชั้นข้อมูลนำเข้า, ชั้นระหว่างกลาง, ชั้นข้อมูลส่งออก และการทำนาย (Prediction) ดังแสดงในรูปที่ 3 ซึ่งจะเรียกกระบวนการทั้งหมดนี้ว่า การแผ่กระจายเดินทาง (Forward propagation) ซึ่งทำงานร่วมกับการแผ่กระจายย้อนกลับ (Backward propagation) [14]



รูปที่ 3 การทำงานของการเรียนรู้เชิงลึก [14]

- การแผ่กระจายเดินทาง (Forward propagation) คือ การนำข้อมูล x เข้ามาประมวลผลร่วมกับค่าน้ำหนัก (Weight) และค่าความเอนเอียง (Bias) เป็นชั้นๆ จนได้คำตอบ [14]
- การแผ่กระจายย้อนกลับ (Backward propagation) คือ การปรับค่าน้ำหนัก และค่าความเอนเอียง ในแต่ละเส้นของโครงประสาทเทียม โดยดูจากค่าความผิดพลาด (Error) ที่เกิดขึ้นในแต่ละเซลล์ประสาท [13]

ในการทำงานของเซลล์ประสาท จะประกอบด้วย 2 ขั้นตอน คือ Weighted Sum และ Activation Function [15]

2.2.2.1 การทำงานของ Weighted Sum

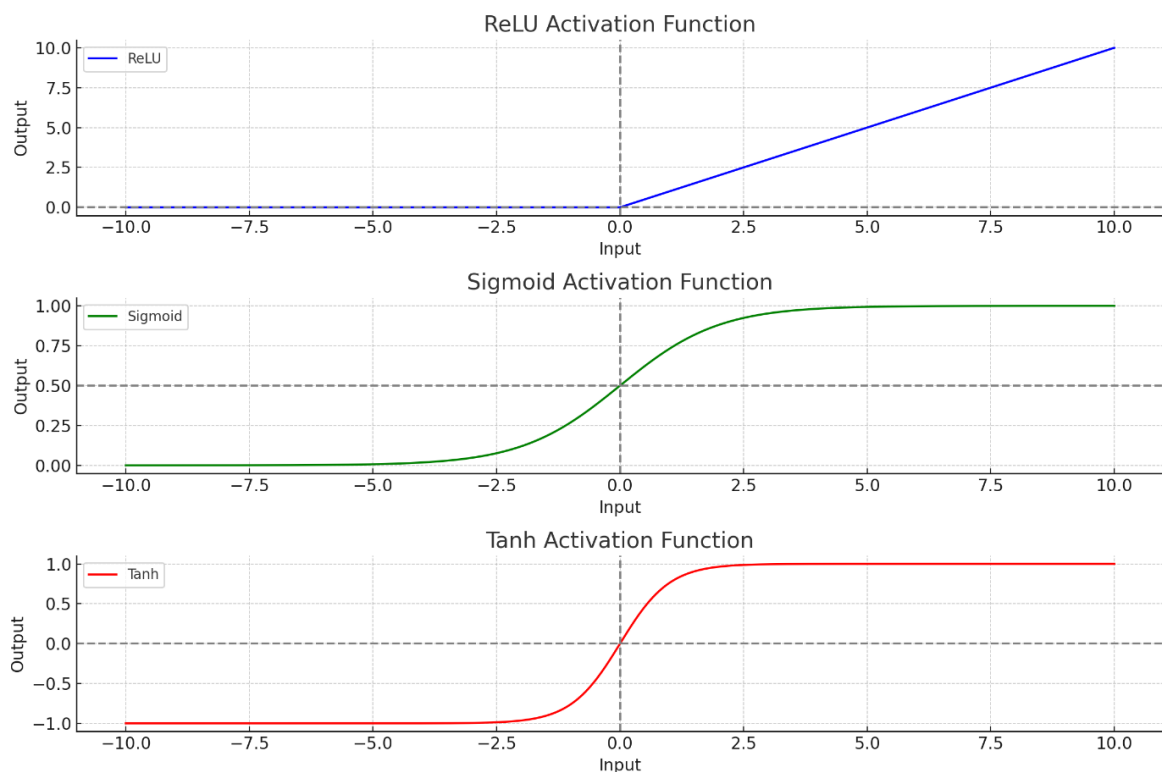
การรวมผล Weighted Sum ทำได้โดยการนำค่าข้อมูลนำเข้า (x_i) คูณกับน้ำหนัก (w_i) ของแต่ละตัว แล้วรวมค่าความเอนเอียง (w_0) เมื่อได้ผลรวมแล้วจะส่งไปตัดสินใจในขั้นตอนต่อไป [10] ซึ่งสามารถเขียนในรูปสมการได้ดังนี้

$$\text{Weighted Sum} = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + \dots + w_n \cdot x_n \quad (1)$$

2.2.2.2 การทำงานของ Activation Function

Activation Function จะทำการกำหนดค่าผลลัพธ์ให้อยู่ในช่วงค่าที่ต้องการ [15] โดย Activation Function ที่นิยมใช้ในเครือข่ายประสาทเทียม จะแสดงในรูปที่ 4 ได้แก่

1. ReLU (Rectified Linear Unit Function) มีช่วงค่าผลลัพธ์ $[0, \infty)$
2. Sigmoid (Logistic Function) มีช่วงค่าผลลัพธ์ $(0, 1)$
3. Tanh (Hyperbolic Tangent Function) มีช่วงค่าผลลัพธ์ $(-1, 1)$



รูปที่ 4 Activation Function ที่นิยมใช้ในเครือข่ายประสาทเทียม

2.2.3 Gradient Descent

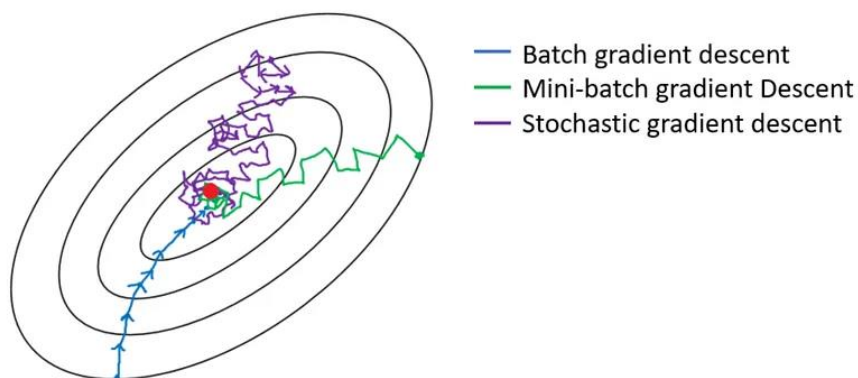
Gradient Descent โดยทั่วไป หมายถึง การเคลื่อนตัวลงมาตามความลาดชัน ในทางการเรียนรู้ของเครื่อง หมายถึง การหาค่าน้ำหนัก ที่ดีที่สุด ที่ทำให้ค่าฟังก์ชันความสูญเสีย (Loss Function) น้อยที่สุด [10] โดยแบ่งตามประเภทต่าง ๆ ได้ดังต่อไปนี้

2.2.3.1 ขั้นตอนการทำงานของ Gradient Descent [10]

1. ตั้งค่าพารามิเตอร์ โดยกำหนดค่าน้ำหนักด้วยค่าที่สุ่มขึ้นมา
2. คำนวณค่าการเปลี่ยนแปลง (Gradient change) คือ การคำนวณหาค่าความสูญเสีย
3. ทำการปรับค่าน้ำหนัก ใหม่ แล้วทำการคำนวณหาค่าความสูญเสียใหม่
4. ทำซ้ำขั้นตอนที่ 3 จนกว่าจะได้ค่าความสูญเสียที่น้อยที่สุด

2.2.3.2 รูปแบบของ Gradient Descent

Gradient Descent แบ่งออกเป็น 3 รูปแบบ ดังที่แสดงในรูปที่ 5



รูปที่ 5 รูปแบบของ Gradient Descent [16]

1. **Batch Gradient Descent** เป็นการนำเอาข้อมูล Dataset ทั้งหมดเข้าไปประมวลผลแต่ละรอบ (Epochs) จากนั้นทำการปรับค่าน้ำหนัก แล้วคำนวณใหม่ ถ้าหากมีข้อมูลจำนวนมาก อาจเกิดปัญหาหน่วยความจำไม่เพียงพอในการประมวลผล [10]
2. **Stochastic Gradient Descent** เป็นการสุ่มเอาข้อมูล Dataset ทีละ 1 ตัวอย่าง เข้าไปประมวลผลแต่ละรอบ จากนั้นทำการปรับค่าน้ำหนัก แล้วคำนวณใหม่ ซึ่งจะทำให้ค่าความสูญเสียมีการแกว่ง (Oscillation) มากขึ้น [10]
3. **Mini-batch Gradient Descent** เป็นการผสมกันระหว่าง 2 แบบแรก คือ เป็นการสุ่มเอาข้อมูล Dataset หลายตัวอย่าง เข้าไปประมวลผลแต่ละรอบ ซึ่งจะประมวลผลได้เร็วและลดการแกว่ง [10]

2.2.4 การหาค่าฟังก์ชันความสูญเสีย (Loss function)

ฟังก์ชันความสูญเสีย คือ ฟังก์ชันที่รับค่ามาจากคุณลักษณะต่าง ๆ แล้วคำนวณออกมาเป็นจำนวนจริงที่แสดงถึงสิ่งที่ต้องสูญเสียในงานนั้น ๆ [17] ซึ่งประกอบด้วย

2.2.4.1 การหาค่าฟังก์ชันความสูญเสียสำหรับงานด้านการถดถอย

1. **Mean Squared Error (MSE)** เป็นค่าเฉลี่ยของผลต่างกำลังสอง ระหว่างค่าที่เป็นผลเฉลย (y) กับค่าที่เกิดจากการทำนายของแบบจำลอง (\hat{y}) [18] โดยมีสมการ คือ

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N (y - \hat{y}_i)^2 \quad (2)$$

2. **Mean Squared Logarithmic Error (MSLE)** เป็นค่าที่แสดงถึงความแตกต่างสัมพัทธ์ ระหว่างค่าที่เป็นผลเฉลยกับค่าที่เกิดจากการทำนายของแบบจำลอง [18] โดยมีสมการ คือ

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2 \quad (3)$$

3. **Mean Absolute Error (MAE)** เป็นค่าเฉลี่ยของผลต่างสัมบูรณ์ ระหว่างค่าที่เป็นผลเฉลยกับค่าที่เกิดจากการทำนายของแบบจำลอง [18] โดยมีสมการ คือ

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N |y - \hat{y}_i| \quad (4)$$

2.2.4.2 การหาค่าฟังก์ชันความสูญเสียสำหรับงานด้านการจำแนกประเภท

1. **Binary Cross-entropy Loss** เป็นฟังก์ชันที่ใช้คำนวณหาค่าความสูญเสีย โดยการใช้การแจกแจงความน่าจะเป็น ซึ่งจำแนกเป็น 2 กลุ่ม (Class) เช่น 0 หรือ 1 [10] โดยมีสมการ คือ

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=0}^N y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \quad (5)$$

2. **Categorical Cross-entropy Loss** เป็นฟังก์ชันที่ใช้คำนวณหาค่าความสูญเสีย ที่มีจำนวนกลุ่มมากกว่า 2 กลุ่ม ซึ่งจะมีการเข้ารหัสผลเฉลยแบบ One-hot encoding [10] โดยมีสมการ คือ

$$L(y, \hat{y}) = -\sum_{i=0}^N y_i \log \hat{y}_i \quad (6)$$

3. **Sparse Categorical Cross-entropy Loss** เป็นฟังก์ชันที่ใช้คำนวณหาค่าความสูญเสีย ที่มีจำนวนกลุ่มมากกว่า 2 กลุ่ม โดยไม่ใช้การเข้ารหัสผลเฉลยแบบ One-hot encoding [10]

2.3 การเรียนรู้ของเครื่องขนาดเล็ก (Tiny Machine Learning)

การเรียนรู้ของเครื่องขนาดเล็ก คือ การนำกระบวนการเรียนรู้ของเครื่องไปประยุกต์ใช้งานบนอุปกรณ์ที่มีพลังงานต่ำ เช่น ตัวรับรู้และไมโครคอนโทรลเลอร์ โดยสามารถวิเคราะห์ข้อมูลโดยใช้พลังงานในระดับต่ำ [19,20]

2.3.1 คุณสมบัติของการเรียนรู้ของเครื่องขนาดเล็ก [19]

1. แบบจำลองทำงานบนอุปกรณ์ประมวลผลข้อมูล (Edge Device) โดยไม่จำเป็นต้องถ่ายโอนข้อมูลไปยังเครื่องบริการ (Server)
2. ใช้พลังงานเพียงเล็กน้อยในการประมวลผลข้อมูล และไม่จำเป็นต้องมีโครงสร้างพื้นฐานเครื่องบริการขนาดใหญ่
3. ใช้ตัวรับรู้บนอุปกรณ์ทำการรวบรวมและประมวลผลข้อมูล โดยไม่จำเป็นต้องเชื่อมต่อกับอินเทอร์เน็ต
4. มีความปลอดภัยและความเป็นส่วนตัวของข้อมูล เนื่องจากข้อมูลไม่ได้ส่งไปยังเครื่องบริการโดยตรง

2.3.2 การประยุกต์ใช้การเรียนรู้ของเครื่องขนาดเล็ก [20]

1. ด้านเกษตรกรรม สามารถวิเคราะห์สภาพแวดล้อมจากการปลูกพืช เช่น สภาพดิน สภาพน้ำ เพื่อทำการดูแลรักษาพืชอย่างเหมาะสม
2. ด้านอุตสาหกรรม สามารถตรวจสอบระบบอัตโนมัติในอุตสาหกรรม เช่น ตรวจสอบความสมบูรณ์และปัญหาที่อาจเกิดขึ้นของอุปกรณ์ในอุตสาหกรรม
3. ด้านสุขภาพ สามารถตรวจวัดและวิเคราะห์สุขภาพ เช่น ระดับน้ำตาลในเลือด อัตราการเต้นของหัวใจ
4. ด้านสิ่งแวดล้อม สามารถตรวจสอบคุณภาพของอากาศ ระดับน้ำ และปัจจัยทางสิ่งแวดล้อมอื่นๆ
5. ด้านยานพาหนะ สามารถจัดการจราจรและความปลอดภัยของยานพาหนะ โดยวิเคราะห์ข้อมูลจากตัวรับรู้แบบเวลาจริง (Real Time)

2.4 การประเมินแบบจำลองการเรียนรู้ของเครื่อง

2.4.1 การแบ่งชุดข้อมูลสำหรับทดสอบ [21]

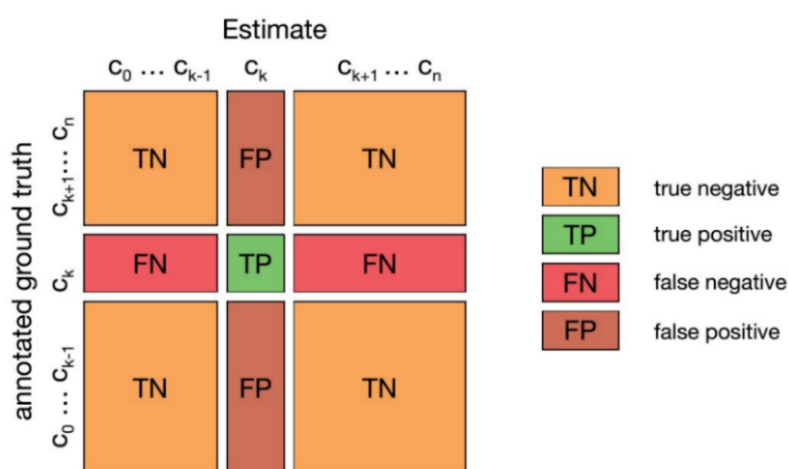
ชุดข้อมูลสำหรับนำมาประเมินแบบจำลองการเรียนรู้ของเครื่องจะแบ่งออกเป็น 3 ส่วน ได้แก่

1. ชุดข้อมูลการฝึก (Training Set) คือ ชุดข้อมูลที่ใช้สำหรับการเรียนรู้ (Train) เพื่อให้เหมาะสมกับพารามิเตอร์กับแบบจำลองการเรียนรู้ของเครื่อง

- ชุดข้อมูลตรวจสอบ (Validation Set) คือ ชุดข้อมูลที่ใช้สำหรับประเมินแบบจำลอง ระหว่างการเรียนรู้ เพื่อปรับให้แบบจำลองทำนายได้ดีขึ้น
- ชุดข้อมูลทดสอบ (Testing Set) คือ ชุดข้อมูลที่ใช้สำหรับทดสอบแบบจำลองของการเรียนรู้ของเครื่องที่ใช้ทดสอบก่อนเอาไปใช้งานจริง

2.4.2 ตารางเมทริกซ์ความสับสน (Confusion Matrix)

ตารางเมทริกซ์ความสับสน เป็นตารางที่ใช้ประเมินผลลัพธ์การทำนายเมื่อเปรียบเทียบกับค่าจริง โดยบอกถึงจำนวนผลลัพธ์ของแต่ละกลุ่มที่ได้จากการทำนายและค่าของจริงในแต่ละช่องของตาราง [22] ดังที่แสดงไว้ในรูปที่ 6 ดังนี้



รูปที่ 6 ตาราง Confusion Matrix [22]

- True Positive (TP) คือ สิ่งที่ทำนายตรงกับค่าของจริง เมื่อค่าที่ทำนายเป็น Positive
- True Negative (TN) คือ สิ่งที่ทำนายตรงกับค่าของจริง เมื่อค่าที่ทำนายเป็น Negative
- False Positive (FP) คือ สิ่งที่ทำนายไม่ตรงกับค่าของจริง เมื่อค่าที่ทำนายเป็น Positive
- False Negative (FN) คือ สิ่งที่ทำนายไม่ตรงกับค่าของจริง เมื่อค่าที่ทำนายเป็น Negative

2.4.3 ค่าความแม่นยำ (Accuracy)

ค่าความแม่นยำ คือ สัดส่วนเปอร์เซ็นต์ความถูกต้อง เป็นสัดส่วนของจำนวนที่ทำนายถูกกับจำนวนทั้งหมด [9] โดยมีสมการ คือ

$$\text{ค่าความแม่นยำ} = \frac{TP+TN}{\text{จำนวนทั้งหมด}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

2.4.4 ค่าความเที่ยง (Precision)

ค่าความเที่ยง คือ ความแม่นยำเมื่อเทียบกับผลทำนาย คำนวณเป็นสัดส่วนเปอร์เซ็นต์ที่เป็น Positive หรือ Negative [9] โดยมีสมการ คือ

$$\text{ค่าความเที่ยง (Positive)} = \frac{TP}{\text{จำนวนที่ทำนายว่าเป็น positive ทั้งหมด}} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{ค่าความเที่ยง (Negative)} = \frac{TN}{\text{จำนวนที่ทำนายว่าเป็น negative ทั้งหมด}} = \frac{TN}{TN+FN} \quad (9)$$

2.4.5 ค่าการเรียกคืน (Recall)

ค่าการเรียกคืน คือ ความแม่นยำเมื่อเทียบกับผลลัพธ์ที่เป็นของจริง [9] โดยมีสมการ คือ

$$\text{ค่าการเรียกคืน (Positive)} = \frac{TP}{\text{จำนวนของจริงที่เป็น positive ทั้งหมด}} = \frac{TP}{TP+FN} \quad (10)$$

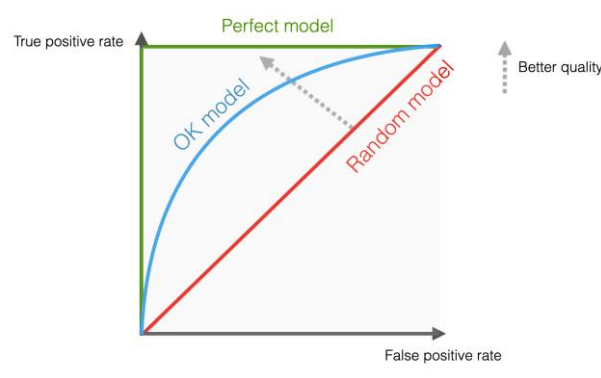
$$\text{ค่าการเรียกคืน (Negative)} = \frac{TN}{\text{จำนวนของจริงที่เป็น negative ทั้งหมด}} = \frac{TN}{TN+FP} \quad (11)$$

2.4.6 ค่า F1 score

ค่า F1 score เป็นค่าที่แสดงประสิทธิภาพ โดยการนำค่าความเที่ยงกับค่าการเรียกคืนมาคำนวณหาค่าเฉลี่ย หากมีค่ามากแสดงว่าแบบจำลองมีประสิทธิภาพดี [9] โดยมีสมการ คือ

$$\text{ค่า F1 score} = \frac{2 \times \text{ค่าความเที่ยง} \times \text{ค่าการเรียกคืน}}{\text{ค่าความเที่ยง} + \text{ค่าการเรียกคืน}} \quad (12)$$

2.4.7 ค่าของ ROC curve และ AUC



รูปที่ 7 ค่าของ ROC curve และ AUC [23]

จากรูปที่ 7 อธิบายได้ว่า ROC curve ย่อมาจาก Receiver Operating Characteristic curve เป็นเส้นที่วัดถึงประสิทธิภาพของแบบจำลองแบบการจำแนกประเภท เป็นการแสดงความสัมพันธ์ระหว่างค่า False Positive Rate (FPR) กับค่า True Positive Rate (TPR) และ AUC ย่อมาจาก Area Under the

Curve คือ ค่าพื้นที่ใต้เส้น ROC Curve ซึ่งจะมีค่าตั้งแต่ 0 – 1 โดยที่แบบจำลองที่มีประสิทธิภาพสูงที่สุดมีค่า AUC เท่ากับ 1 [23]

- ค่าความไว (Sensitivity) คือ ค่าแสดงความสามารถในการบอกได้ว่าเป็น True Positive มีค่าเท่ากับค่าความไวที่เป็น Positive [24]
- ค่าความจำเพาะ (Specificity) คือ ค่าแสดงความสามารถในการบอกได้ว่าเป็น True Negative มีค่าเท่ากับค่าความไวที่เป็น Negative [24]

2.5 แพลตฟอร์ม Edge Impulse

Edge Impulse เป็นแพลตฟอร์มสำหรับการฝึกแบบจำลองการเรียนรู้ของเครื่องสำหรับอุปกรณ์ประมวลผลข้อมูล โดยใช้ร่วมกับตัวรับรู้ต่างๆ เช่น หน่วยวัดแรงเฉื่อย, กล้อง, และไมโครโฟน

2.5.1 กระบวนการทำงานของแพลตฟอร์ม Edge Impulse [25]

แพลตฟอร์ม Edge Impulse จะทำงานร่วมกับอุปกรณ์ประมวลผลข้อมูล ซึ่งสามารถแสดงกระบวนการทำงานได้ดังรูปที่ 8 โดยมีกระบวนการทำงานดังนี้



รูปที่ 8 กระบวนการทำงานของแพลตฟอร์ม Edge Impulse [25]

1. **Edge Devices** ใช้อุปกรณ์ประมวลผลข้อมูลรวบรวมข้อมูลจากตัวรับรู้ เช่น หน่วยวัดแรงเฉื่อย, กล้อง, ไมโครโฟน
2. **Data Collection** อัปโหลดข้อมูลที่ได้จากตัวรับรู้ โดยใช้อุปกรณ์ประมวลผลข้อมูลโดยตรงหรือจัดเก็บในรูปแบบไฟล์ ลงในแพลตฟอร์ม Edge Impulse

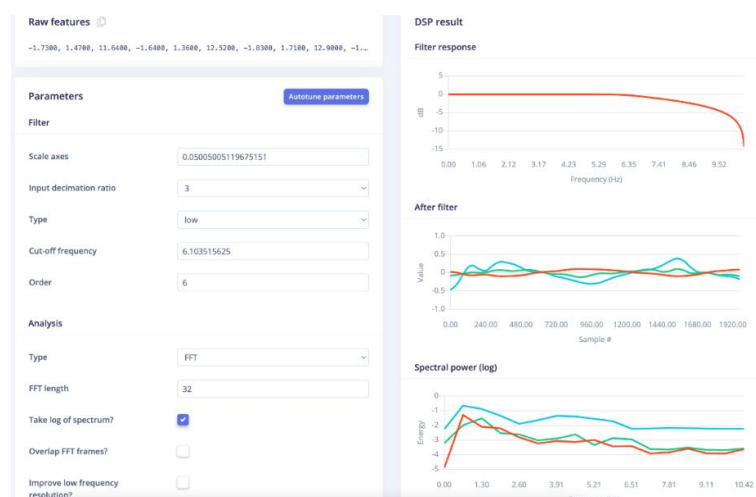
3. **Build** สร้างแบบจำลองที่ใช้สำหรับการเรียนรู้ของเครื่อง เช่น การจำแนกประเภท การถดถอย
4. **Train** ใช้ข้อมูลที่เตรียมไว้ทำการฝึกแบบจำลองการเรียนรู้ของเครื่อง โดยสามารถปรับค่าพารามิเตอร์ได้ตามความเหมาะสม
5. **Optimize** ใช้หลักการลดขนาดและปรับแต่งแบบจำลอง เพื่อให้แบบจำลองทำงานได้ในอุปกรณ์ประมวลผลข้อมูล
6. **Monitor** เป็นการตรวจสอบและประเมินผลแบบจำลองที่ใช้ เพื่อนำไปปรับปรุงและพัฒนาแบบจำลองต่อไป

2.5.2 การวิเคราะห์สเปกตรัม (Spectral Analysis)

การวิเคราะห์สเปกตรัมเป็นกระบวนการที่ใช้แปลงข้อมูลทางเวลา (Time Series) เช่น สัญญาณเสียง ข้อมูลการเคลื่อนที่ โดยแปลงจากโดเมนเวลา (Time Domain) ไปเป็นโดเมนความถี่ (Frequency Domain) ทำให้เห็นถึงองค์ประกอบความถี่ต่าง ๆ ในข้อมูลที่เปลี่ยนแปลงตามช่วงเวลา ทำให้สามารถวิเคราะห์รูปแบบซ้ำ ๆ และองค์ประกอบที่มีความถี่สูงหรือต่ำที่ปรากฏอยู่ในข้อมูล ที่อาจปรากฏไม่ชัดเจนในโดเมนเวลา

2.5.2.1 การวิเคราะห์ด้วย FFT (Fast Fourier Transform)

การวิเคราะห์ด้วย FFT ใช้ในการแยกและตรวจจับความถี่ที่ซับซ้อนจากตัวอย่างข้อมูล ให้ข้อมูลความถี่ของสัญญาณ เหมาะกับการวิเคราะห์รูปแบบสัญญาณที่เกิดซ้ำ สัญญาณมีลักษณะที่ไม่เปลี่ยนแปลงมาก จึงนิยมใช้ในการประมวลผลสัญญาณดิจิทัล การบีบอัดข้อมูล และการวิเคราะห์ความถี่ [26] ซึ่งมีหน้าต่างแสดงผลการทำงานดังรูปที่ 9 โดยมีองค์ประกอบดังนี้



รูปที่ 9 การวิเคราะห์ด้วย FFT ของแพลตฟอร์ม Edge Impulse [26]

1. **Filter Response** แสดงการตอบสนองของตัวกรอง (Filter) หากมีการเปิดใช้งานตัวกรองและกำหนดลำดับของตัวกรองไม่เป็นศูนย์ จะแสดงให้เห็นถึงการลดทอนของสัญญาณในแต่ละช่วงความถี่

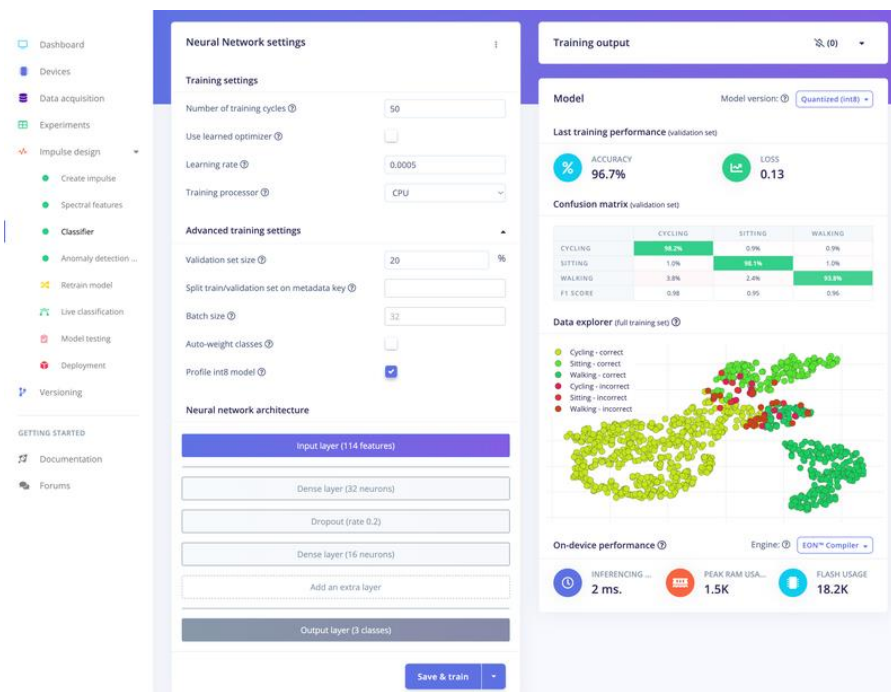
2. **After Filter** แสดงหน้าต่างปัจจุบันของสัญญาณหลังจากที่ตัวกรองถูกนำไปใช้ โดยแสดงในรูปแบบโดเมนเวลา
3. **Spectral Power** แสดงพลังงานของสัญญาณในแต่ละช่วงความถี่ตามที่คำนวณจากขนาด FFT ที่เลือก สามารถแสดงพลังงานของสัญญาณเป็นแบบเชิงเส้นหรือลอการิทึมได้ตามการตั้งค่า

2.5.2.2 การวิเคราะห์ด้วย Wavelet

การวิเคราะห์ด้วยวิธีการ Wavelet ใช้การแปลงสัญญาณแบบแยกย่อยหลายระดับ ให้ข้อมูลความถี่และเวลาที่เกิดของสัญญาณ เหมาะกับสัญญาณที่มีการเปลี่ยนแปลงอย่างฉับพลันและไม่สม่ำเสมอ จึงนิยมใช้ในการประมวลผลภาพ การบีบอัดข้อมูล และการกรองสัญญาณรบกวน [26]

2.5.3 การจำแนกประเภทด้วยเครือข่ายประสาท (Neural Network)

การจำแนกประเภทเป็นกระบวนการที่ทำให้แบบจำลองการเรียนรู้ของเครื่องสามารถจำแนกลักษณะของข้อมูลที่ได้รับเข้ามาออกเป็นหมวดหมู่ตามที่ต้องการ ซึ่งใช้เครือข่ายประสาทในการรับข้อมูลและบอกความน่าจะเป็นว่าข้อมูลจัดอยู่ในหมวดหมู่ใด โดยในแต่ละชั้นของเครือข่ายประสาทจะประกอบด้วยเซลล์ประสาทจำนวนหนึ่ง ซึ่งเซลล์ประสาทในแต่ละชั้นจะเชื่อมต่อกันระหว่างชั้นแรกถึงชั้นต่อไป ซึ่งมีหน้าต่างแสดงผลการทำงานดังรูปที่ 10



รูปที่ 10 การจำแนกประเภทด้วยเครือข่ายประสาทของแพลตฟอร์ม Edge Impulse [27]

2.5.3.1 การตั้งค่าเครือข่ายประสาทของแพลตฟอร์ม Edge Impulse [28]

ในการสร้างแบบจำลองเครือข่ายประสาท จากแพลตฟอร์ม Edge Impulse จะสามารถปรับค่าพารามิเตอร์ได้ดังนี้

1. **Number of training cycles** คือ จำนวนรอบของการฝึกฝนที่ใช้ข้อมูลทั้งหมดในชุดข้อมูลฝึก
2. **Learning rate** คือ การควบคุมการปรับพารามิเตอร์ของแบบจำลองในแต่ละรอบการฝึก หากแบบจำลองเกิดการฝึกที่มากเกินไป ควรจะทำการลดอัตราการเรียนรู้ลง
3. **Validation set size** คือ เปอร์เซนต์ของชุดข้อมูลฝึกที่กั้นไว้สำหรับการตรวจสอบผลการฝึก
4. **Batch size** คือ จำนวนตัวอย่างข้อมูลที่ใช้ต่อรอบการฝึก หากจำนวนตัวอย่างข้อมูลมากเกินไป การฝึกแบบจำลองอาจล้มเหลว

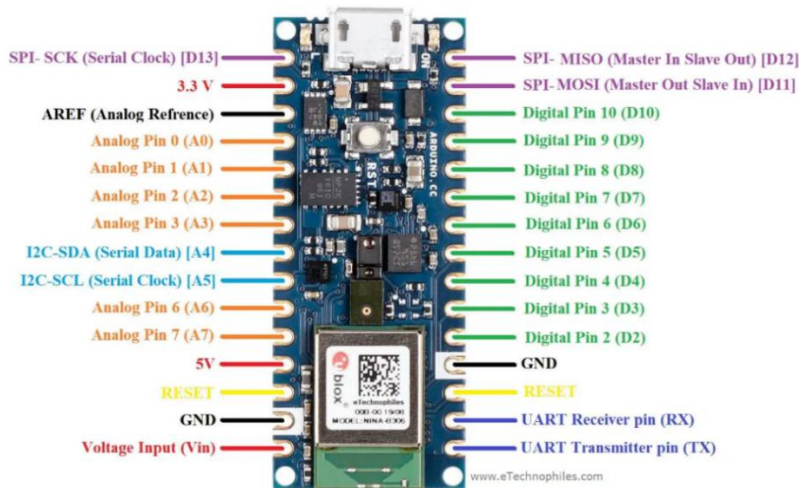
2.5.3.2 สถาปัตยกรรมเครือข่ายประสาทของแพลตฟอร์ม Edge Impulse [28]

ในการสร้างแบบจำลองเครือข่ายประสาท จากแพลตฟอร์ม Edge Impulse จะสามารถกำหนดประเภทของสถาปัตยกรรมเครือข่ายประสาทได้ดังนี้

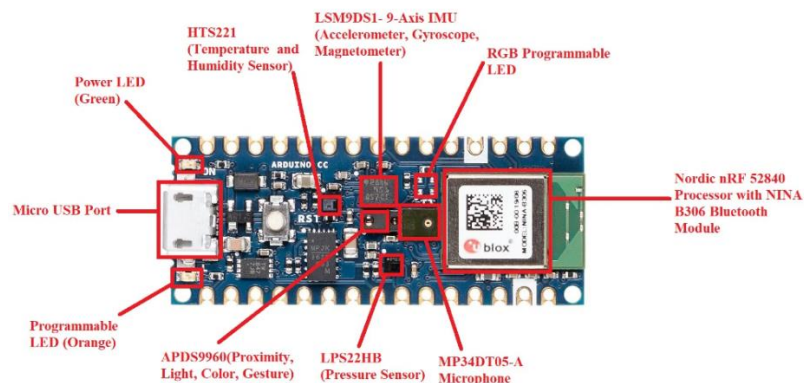
1. **Dense** เป็นชั้นที่ทุกเซลล์ประสาทเชื่อมต่อกับเซลล์ประสาทในชั้นถัดไป ใช้สำหรับข้อมูลที่ผ่านการประมวลผลแล้ว เช่น ผลลัพธ์จากบล็อกวิเคราะห์สัญญาณเสียง
2. **1D Convolution / Pooling** ใช้กับข้อมูลที่มีมิติเดียว เช่น สัญญาณเวลา
3. **2D Convolution / Pooling** ใช้กับข้อมูลที่มีสองมิติ เช่น รูปภาพ ข้อมูลที่มีลักษณะเชิงพื้นที่
4. **Reshape** เป็นการแปลงข้อมูลจากมิติเดียวเป็นหลายมิติ สำหรับการเตรียมข้อมูลสำหรับ Convolutional Layer
5. **Flatten** เป็นการแปลงข้อมูลหลายมิติให้เป็นมิติเดียว สำหรับการส่งข้อมูลต่อหลังจากผ่าน Convolutional Layer
6. **Dropout** ช่วยลดการเกิด Overfitting โดยทำการสุ่มตัดการเชื่อมต่อของบางส่วนภายในเครือข่ายประสาทเทียมระหว่างการฝึกแบบจำลอง

2.6 ไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense

Arduino Nano 33 BLE Sense เป็นไมโครคอนโทรลเลอร์ที่เชื่อมต่อด้วยเทคโนโลยี Bluetooth Low Energy (BLE) โดยใช้ชิป nRF52840 ของ Nordic Semiconductor (ARM Cortex-M4 32 บิต ที่ 64 MHz) [30] สามารถประมวลผลปัญญาประดิษฐ์โดยใช้การเรียนรู้ของเครื่องขนาดเล็ก และ TensorFlow Lite [29] ซึ่งสามารถแสดงแผนผังของบอร์ด ดังรูปที่ 11 และม็อดูลประกอบต่าง ๆ ดังรูปที่ 12 ดังนี้



รูปที่ 11 แผนผังของไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense [30]



รูปที่ 12 องค์ประกอบของไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense [30]

1. หน่วยวัดแรงเฉื่อย (LSM9DS1) ประกอบด้วย

- ตัวรับรู้ความเร่ง สำหรับวัดความเร่งตามแกน X, Y และ Z
- ตัวรับรู้ไจโรสโคป สำหรับตรวจจับการหมุนในสามมิติ
- ตัวรับรู้แมกนีโทมิเตอร์ สำหรับวัดสนามแม่เหล็กโลก

2. ไมโครโฟน (MP34DT05) ใช้ในการบันทึกเสียงและรองรับการประมวลผลสัญญาณเสียง

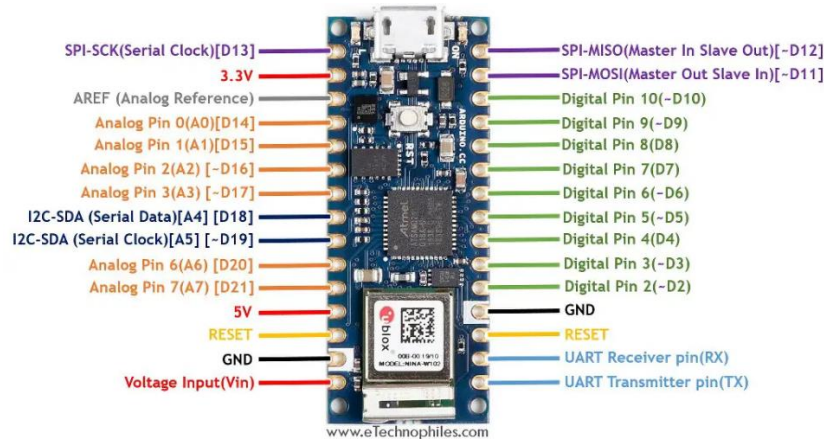
3. บารอมิเตอร์ (LPS22HB) ใช้สำหรับวัดความดันอากาศ เหมาะกับการคาดคะเนระดับความสูง

4. ตัวรับรู้อุณหภูมิและความชื้น (HTS221) ใช้ในการตรวจวัดอุณหภูมิและความชื้นในสภาพแวดล้อม

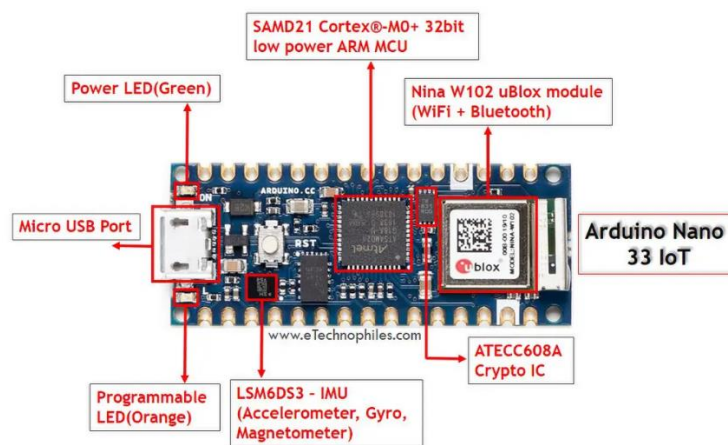
5. ตัวรับรู้แสง (APDS9960) ใช้สำหรับวัดความเข้มของแสง, การรับรู้สี RGB, การตรวจจับการเคลื่อนไหว (Gesture Detection)

2.7 ไมโครคอนโทรลเลอร์ Arduino Nano 33 IoT

Arduino Nano 33 IoT เป็นไมโครคอนโทรลเลอร์ที่สามารถทำงานร่วมกับอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things: IoT) โดยใช้ชิป SAMD21 ของ Nordic Semiconductor (ARM Cortex-M0 32 บิต ที่ 48 MHz) [32] และสามารถเชื่อมต่อได้กับแพลตฟอร์ม Arduino Cloud [31] ซึ่งสามารถแสดงแผนผังของบอร์ด ดังรูปที่ 13 และมีองค์ประกอบต่าง ๆ ดังรูปที่ 14 ดังนี้



รูปที่ 13 แผนผังของไมโครคอนโทรลเลอร์ Arduino Nano 33 IoT [32]



รูปที่ 14 องค์ประกอบของไมโครคอนโทรลเลอร์ Arduino Nano 33 IoT [32]

1. หน่วยวัดแรงเฉื่อย (LSM6DS3) ประกอบด้วย

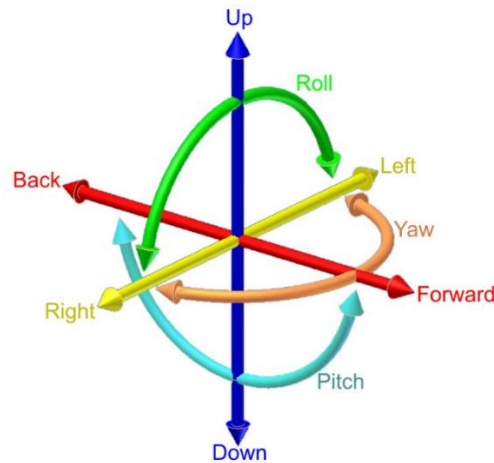
- ตัวรับรู้ความเร่ง สำหรับวัดความเร่งตามแกน X, Y และ Z
- ตัวรับรู้ไจโรสโคป สำหรับตรวจจับการหมุนในสามมิติ

2. Wi-Fi module (NINA-W102) สามารถให้การเชื่อมต่อ Wi-Fi และบลูทูธกับไมโครคอนโทรลเลอร์

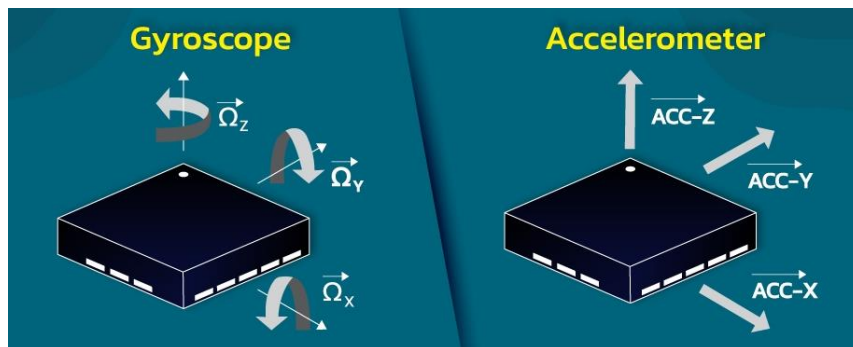
3. Cryptographic IC (ATECC608A) ใช้งานสำหรับเพิ่มความปลอดภัยให้กับไมโครคอนโทรลเลอร์

2.8 หน่วยวัดแรงเฉื่อย (IMU)

หน่วยวัดแรงเฉื่อย (Inertial Measurement Unit: IMU) หรือหน่วยอ้างอิงแรงเฉื่อย (Inertial Reference Unit: IRU) หรือหน่วยอ้างอิงการเคลื่อนที่ (Motion Reference Unit: MRU) ประกอบด้วยตัวรับรู้ความเร่ง 3 แกน, ตัวรับรู้ไจโรสโคป 3 แกน และตัวรับรู้แมกนีโทมิเตอร์ 3 แกน [33] สำหรับการวัดทิศทางการเคลื่อนที่และการหมุนของวัตถุ จะใช้ตัวรับรู้ความเร่ง และตัวรับรู้ไจโรสโคป ดังที่แสดงในรูปที่ 15 และ 16 ดังนี้



รูปที่ 15 ความสัมพันธ์ของแกนที่ได้จากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป [34]



รูปที่ 16 ทิศทางการวัดค่าจากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป [35]

2.8.1 ตัวรับรู้ความเร่ง

ตัวรับรู้ความเร่ง เป็นอุปกรณ์ที่ใช้วัดการเปลี่ยนแปลงของความเร่งหรือแรงดึงดูด (Gravity) ในสามมิติ (x, y, z) ซึ่งบ่งบอกถึงการเคลื่อนที่หรือการเปลี่ยนแปลงตำแหน่งของวัตถุ

- แกน X วัดความเร่งจากแกนด้านหน้า ในทิศทางหน้า-หลัง
- แกน Y วัดความเร่งจากแกนด้านข้าง ในทิศทางซ้าย-ขวา
- แกน Z วัดความเร่งจากแกนแนวตั้ง ในทิศทางขึ้น-ลง

สำหรับไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT จะมีช่วงการวัดอยู่ที่ $[-4, +4]g$ (gravity) [29,31] เมื่อแทนค่า $g = 9.81 \text{ m/s}^2$ ลงไป จะได้ช่วงในการวัดเป็น $[-39.24, +39.24] \text{ m/s}^2$ และมีค่าความละเอียดอยู่ที่ $\pm 0.122 \text{ mg}$ (milli gravity) [29,31] เมื่อแทนค่า $g = 9.81 \text{ m/s}^2$ ลงไป จะได้ค่าความละเอียดอยู่ที่ $\pm 1.196 \text{ mm/s}^2$

2.8.2 ตัวรับรู้ไจโรสโคป (Gyroscope)

ตัวรับรู้ไจโรสโคป เป็นอุปกรณ์ที่ใช้วัดความเร็วในการหมุนหรือการเปลี่ยนแปลงของมุมในสามมิติ (Roll, Pitch, Yaw) ซึ่งสามารถตรวจจับการหมุนหรือการเอียงของวัตถุ

- ค่า Roll วัดการหมุนจากแกนด้านหน้า หรือหมุนรอบแกน X
- ค่า Pitch วัดการหมุนจากแกนด้านข้าง หรือหมุนรอบแกน Y
- ค่า Yaw วัดการหมุนจากแกนแนวตั้ง หรือหมุนรอบแกน Z

สำหรับไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT จะมีช่วงการวัดอยู่ที่ $[-2000, +2000] \text{ dps}$ (degree per second) [29,31] และมีค่าความละเอียดอยู่ที่ $\pm 70 \text{ mdps}$ (milli degree per second) [29,31]

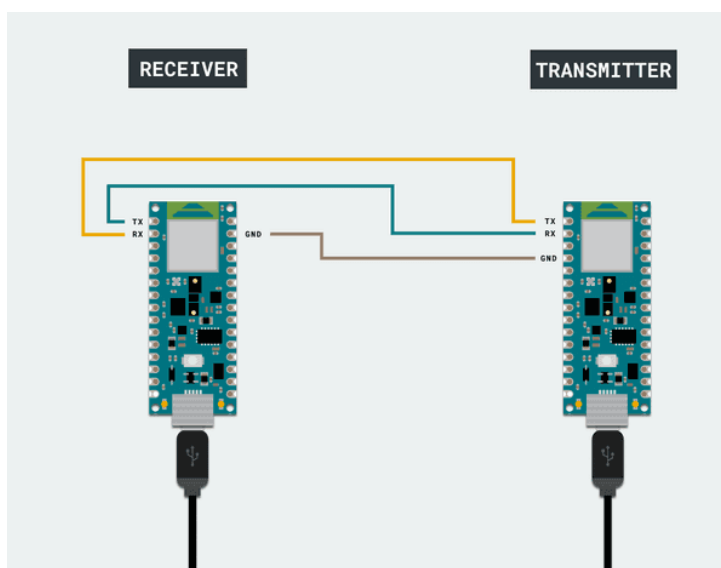
3. แนวทางการดำเนินงาน

การดำเนินงานการวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูลจะใช้ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT ในการรวบรวมและทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ โดยใช้ข้อมูลจากหน่วยวัดแรงเฉื่อย ซึ่งประกอบด้วยตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป ในการวิเคราะห์การเคลื่อนไหวของมนุษย์

3.1 การติดตั้งอุปกรณ์สำหรับรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์

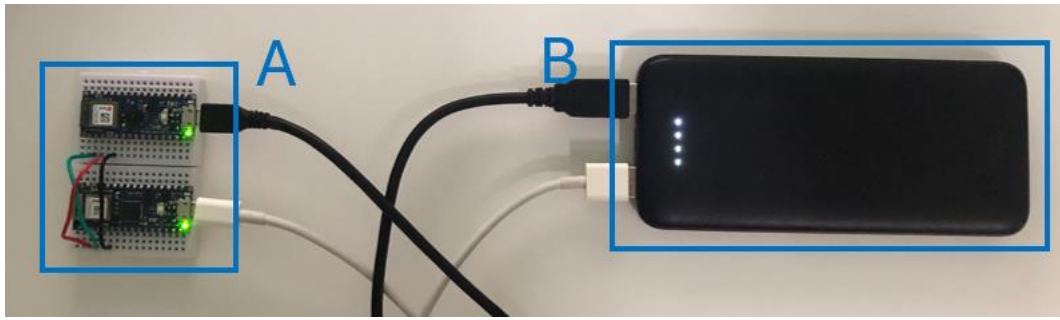
ในการวิเคราะห์ข้อมูลการเคลื่อนไหวของมนุษย์ จะใช้ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT ต่อเข้ากับแบตเตอรี่สำรอง (Power Bank) โดยไม่ต้องต่อสายเข้ากับคอมพิวเตอร์โดยตรง

สำหรับการสื่อสารข้อมูลระหว่างบอร์ด Arduino Nano 33 BLE Sense กับ Arduino Nano 33 IoT จะเป็นการสื่อสารแบบ UART (Universal Asynchronous Receiver and Transmitter) ซึ่งแสดงแผนภาพการต่อวงจรได้ดังรูปที่ 17

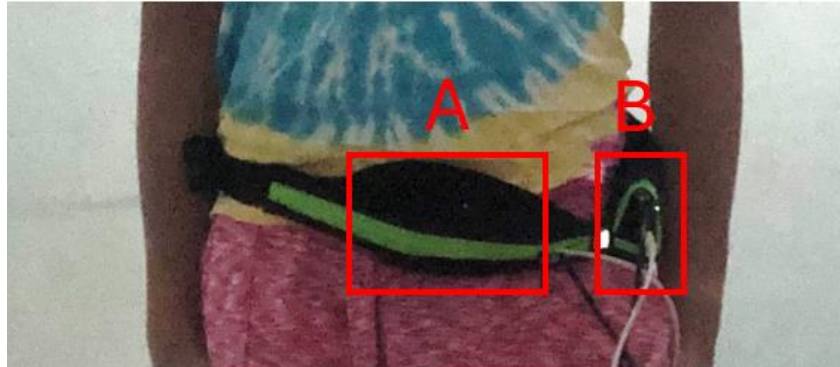


รูปที่ 17 แผนภาพแสดงการเชื่อมต่อบอร์ด Arduino Nano ผ่านการสื่อสารแบบ UART [36]

ในการติดตั้งอุปกรณ์สำหรับรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ จะใช้การเชื่อมต่อบอร์ด Arduino Nano 33 BLE Sense กับ Arduino Nano 33 IoT แล้วต่อเข้ากับแบตเตอรี่สำรอง โดยที่อุปกรณ์ทั้งหมดจะถูกติดตั้งไว้ในกระเป๋าคาดเอวระหว่างการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ ซึ่งสามารถแสดงการต่อวงจรจริงได้ดังรูปที่ 18 และแสดงการติดตั้งอุปกรณ์ได้ดังรูปที่ 19 โดยในส่วน A จะแสดงถึงการเชื่อมต่อบอร์ด Arduino Nano 33 BLE Sense ที่อยู่ด้านบนกับ Arduino Nano 33 IoT ที่อยู่ด้านล่าง ด้วยสายจัมเปอร์ (Jumper Wire) 3 เส้น และในส่วน B จะแสดงถึงแบตเตอรี่สำรอง



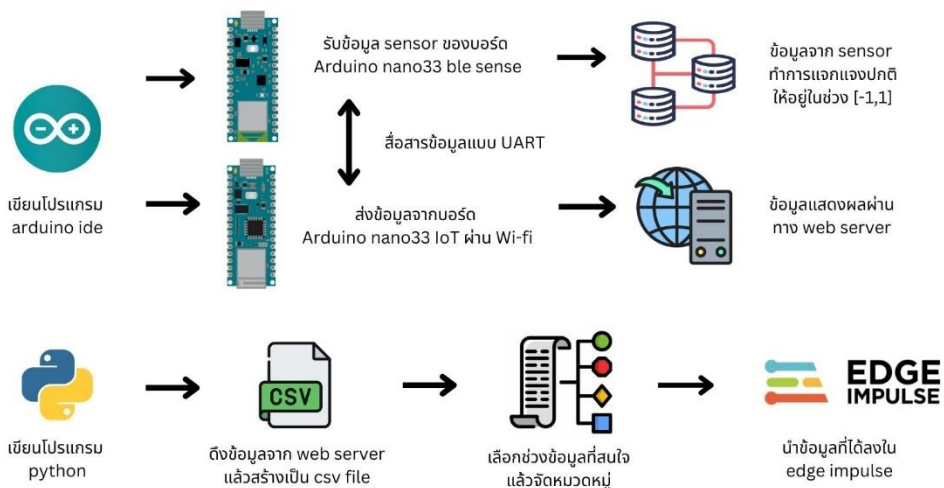
รูปที่ 18 การต่อวงจรจริงระหว่างบอร์ด Arduino Nano 33 BLE Sense กับ Arduino Nano 33 IoT



รูปที่ 19 การติดตั้งอุปกรณ์สำหรับการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์

3.2 การรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ (Training Phase)

รูปแบบการสื่อสารข้อมูลผ่าน wifi : Training Phase

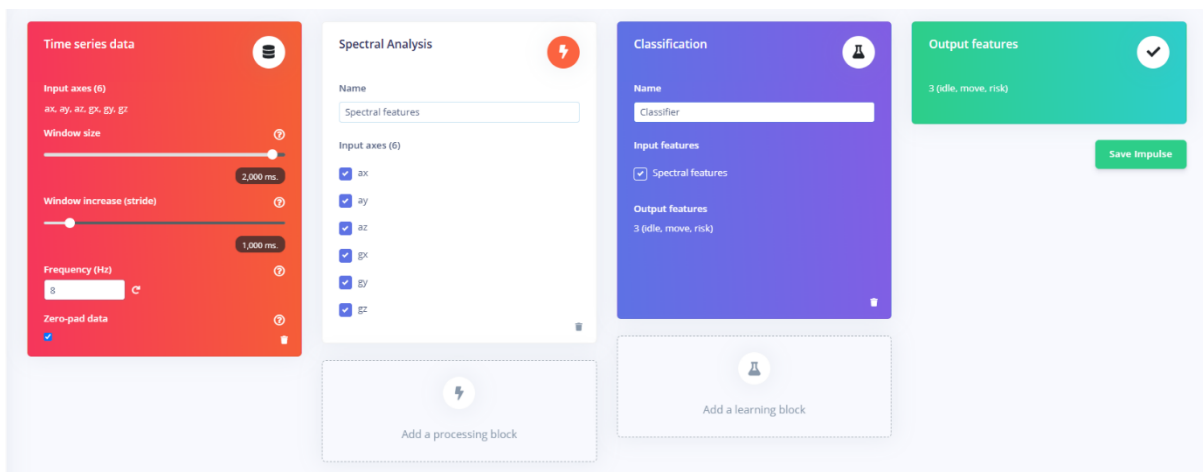


รูปที่ 20 แผนภาพแสดงขั้นตอนการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์โดยสื่อสารข้อมูลผ่าน Wi-Fi

จากรูปที่ 20 จะเป็นการแสดงขั้นตอนการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์โดยสื่อสารข้อมูลผ่าน Wi-Fi โดยอธิบายรายละเอียดในแต่ละขั้นตอนดังนี้

1. เริ่มจากการเขียนโปรแกรมภาษา C++ ในโปรแกรม Arduino IDE เพื่อรับข้อมูลจากตัวรับรู้ความเร่งและตัวรับรู้ไจโรสโคป ในไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT
 - 1.1 สำหรับบอร์ด Arduino Nano 33 BLE Sense จะเป็นการรับข้อมูล 6 ค่า ได้แก่ ค่าจากตัวรับรู้ความเร่งในแกน x, y, z (ax, ay, az) และค่าจากตัวรับรู้ไจโรสโคปในแกน x, y, z (gx, gy, gz) โดยจะรวมข้อมูลรอบละ 0.125 วินาที และทำการการแจกแจงปกติ คือ ทำให้ข้อมูลจากตัวรับรู้อยู่ในช่วง -1 ถึง 1 แล้วส่งข้อมูลไปยังบอร์ด Arduino Nano 33 IoT
 - 1.2 สำหรับบอร์ด Arduino Nano 33 IoT จะเป็นการรับค่าข้อมูลจากตัวรับรู้ที่ได้จากบอร์ด Arduino Nano 33 BLE Sense แล้วนำไปแสดงผลผ่าน Web Server
2. เขียนโปรแกรมเพื่อรับข้อมูลจาก Web Server โดยใช้ภาษา Python ในโปรแกรม Visual Studio Code เพื่อใช้ในการวิเคราะห์ข้อมูลต่อไป
 - 2.1 เขียนโปรแกรมเพื่อรับข้อมูลจาก Web Server ทุก 0.125 วินาที แล้วสร้างเป็นไฟล์ในรูปแบบ csv ที่มีข้อมูล 6 คอลัมน์ คือ ข้อมูลจากตัวรับรู้ 6 แกน ได้แก่ ค่า ax, ay, az, gx, gy, gz
 - 2.2 เลือกช่วงข้อมูลที่สนใจจากไฟล์ข้อมูล csv ที่รวบรวมได้ โดยลบค่าข้อมูลที่ผิดปกติ (Outlier) ออกจากชุดข้อมูล แล้วจัดเป็นหมวดหมู่ตามข้อมูลการเคลื่อนไหวที่บันทึกไว้ ได้แก่ กลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสี่ยงอันตราย

3.3 การวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse



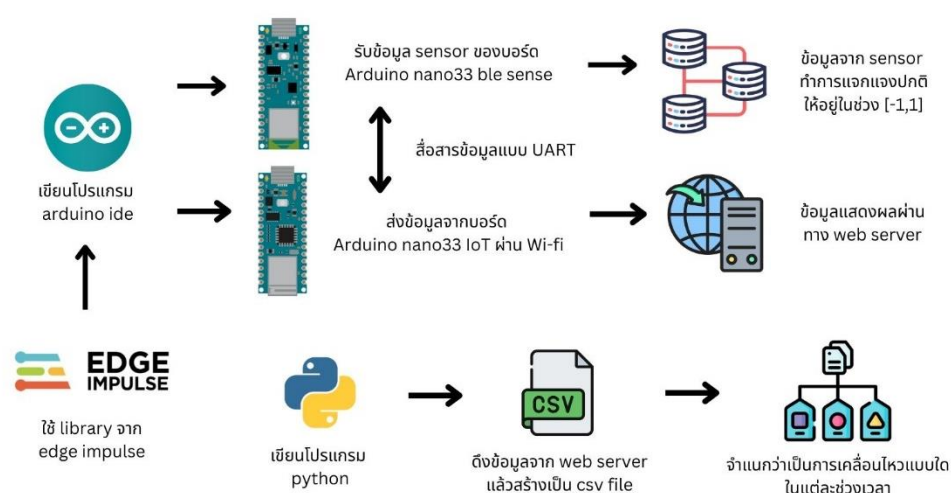
รูปที่ 21 หน้าต่างแสดงขั้นตอนการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse

จากรูปที่ 21 จะเป็นการแสดงขั้นตอนการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse โดยอธิบายรายละเอียดในแต่ละขั้นตอนดังนี้

1. นำไฟล์ csv ที่ได้จากขั้นตอนการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ ซึ่งเป็นข้อมูลทางเวลาอัปโหลดลงในส่วนของ Data Acquisition โดยแต่ละไฟล์ csv จะมีการระบุหมวดหมู่ตามข้อมูลการเคลื่อนไหวที่บันทึกไว้ ได้แก่ กลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสียงอันตราย
2. ข้อมูลที่ได้จะเป็นข้อมูลอนุกรมเวลา จึงทำการแบ่งส่วนข้อมูลจากข้อมูลที่รวบรวมมาได้ โดยกำหนดขนาดของหน้าต่าง (Window Size) เป็น 2 วินาที หรือ 16 ชุดข้อมูล และเลื่อนหน้าต่าง (Stride) ทุก 1 วินาที หรือ 8 ชุดข้อมูล เพื่อให้ข้อมูลมีขนาดเท่ากันก่อนนำไปประมวลผล
3. ใช้แบบจำลองการวิเคราะห์สเปกตรัม โดยใช้การวิเคราะห์ด้วย FFT เพื่อแปลงข้อมูลจากโดเมนเวลาไปเป็นโดเมนความถี่ โดยกำหนดค่าพารามิเตอร์ FFT length เท่ากับ 32 แล้วสร้างลักษณะข้อมูล (Generate Feature) ทำให้เห็นถึงลักษณะข้อมูลที่มีความแตกต่างกันของข้อมูล 3 กลุ่ม
4. ใช้แบบจำลองการจำแนกประเภท โดยใช้เครือข่ายประสาท โดยรับข้อมูลเป็นลักษณะข้อมูล 126 ลักษณะ ที่ได้จากขั้นตอนการวิเคราะห์สเปกตรัม แล้วผลลัพธ์ที่ได้เป็นการจำแนกข้อมูลออกเป็น 3 ประเภท
5. นำแบบจำลองไปใช้งานกับอุปกรณ์ประมวลผลข้อมูล โดยการดาวน์โหลดเป็นคลังโปรแกรม (Library) ของ Arduino ในส่วนของ Deployment

3.4 การทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ (Testing Phase)

รูปแบบการสื่อสารข้อมูลผ่าน wifi : Testing Phase



รูปที่ 22 แผนภาพแสดงขั้นตอนการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์โดยสื่อสารข้อมูลผ่าน Wi-Fi

จากรูปที่ 22 จะเป็นการแสดงขั้นตอนการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์โดยสื่อสารข้อมูลผ่าน Wi-Fi โดยอธิบายรายละเอียดในแต่ละขั้นตอนดังนี้

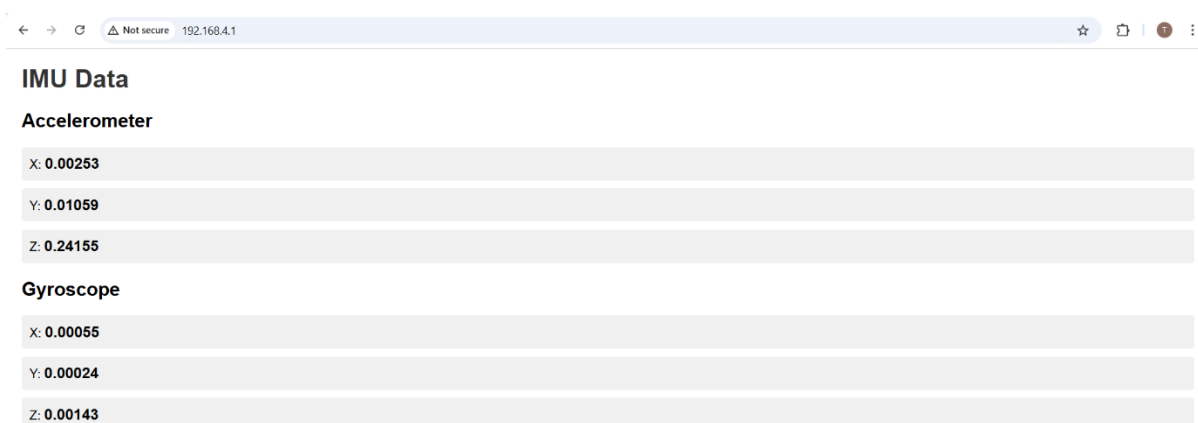
1. ดาวน์โหลดคลังโปรแกรม จากแพลตฟอร์ม Edge Impulse สำหรับการวิเคราะห์ข้อมูลการเคลื่อนไหวของมนุษย์ แล้วทำการเขียนโปรแกรมภาษา C++ ในโปรแกรม Arduino IDE เพื่อรับข้อมูลจากตัวรับรู้ความเร่ง และตัวรับรู้ไจโรสโคป ในไมโครคอนโทรลเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT
 - 1.1 สำหรับบอร์ด Arduino Nano 33 BLE Sense จะเป็นการรับข้อมูล 6 ค่า ได้แก่ ค่าจากตัวรับรู้ความเร่งในแกน x, y, z (ax, ay, az) และค่าจากตัวรับรู้ไจโรสโคปในแกน x, y, z (gx, gy, gz) โดยจะรวมข้อมูลรอบละ 0.125 วินาที และทำการการแจกแจงปกติ คือ ทำให้ข้อมูลจากตัวรับรู้อยู่ในช่วง -1 ถึง 1
 - 1.2 นำข้อมูลที่ได้ทุก 2 วินาที หรือ 16 ชุดข้อมูล ประมวลผลด้วยคลังโปรแกรมจากแพลตฟอร์ม Edge Impulse ซึ่งจะได้ผลลัพธ์เป็นค่าความน่าจะเป็นของข้อมูลการเคลื่อนไหวที่แบบจำลองทำนายได้ ได้แก่ กลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสียงอันตราย แล้วส่งข้อมูลไปยังบอร์ด Arduino Nano 33 IoT
 - 1.3 สำหรับบอร์ด Arduino Nano 33 IoT จะเป็นการรับค่าข้อมูลค่าความน่าจะเป็นที่ได้จากบอร์ด Arduino Nano 33 BLE Sense แล้วนำไปแสดงผลผ่าน Web Server
2. เขียนโปรแกรมเพื่อรับข้อมูลจาก Web Server ทุก 2 วินาที โดยใช้ภาษา Python ในโปรแกรม Visual Studio Code แล้วสร้างเป็นไฟล์ในรูปแบบ csv ที่แสดงข้อมูลค่าความน่าจะเป็นที่แบบจำลองจะทำนายข้อมูลการเคลื่อนไหวของมนุษย์ว่าเป็นกลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว และกลุ่มเสียงอันตรายในแต่ละช่วงเวลา

4. ผลลัพธ์จากการดำเนินการ

การดำเนินการวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล จะใช้รูปแบบการสื่อสารข้อมูลผ่าน Wi-fi โดยใช้ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART ในการรวบรวมและทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ โดยใช้ข้อมูลจากหน่วยวัดแรงเฉื่อย โดยมีผลลัพธ์จากการดำเนินการดังนี้

4.1 ผลลัพธ์จากการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์

1. ผลลัพธ์จากการรวบรวมข้อมูลการเคลื่อนไหว สามารถทำได้โดยเขียนโปรแกรมภาษา C++ ในโปรแกรม Arduino IDE ซึ่งใช้ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART แล้วแสดงผลข้อมูลผ่านทาง Web Server โดยแสดงผลข้อมูลทุก 0.125 วินาที โดยค่าที่แสดงผล ซึ่งสามารถแสดงได้ดังรูปที่ 23



The screenshot shows a web browser window with the address bar displaying '192.168.4.1'. The page content is titled 'IMU Data' and contains two sections: 'Accelerometer' and 'Gyroscope'. Each section has three rows of data for X, Y, and Z axes.

IMU Data	
Accelerometer	
X:	0.00253
Y:	0.01059
Z:	0.24155
Gyroscope	
X:	0.00055
Y:	0.00024
Z:	0.00143

รูปที่ 23 ข้อมูลการรวบรวมข้อมูลการเคลื่อนไหวของมนุษย์ที่แสดงผลผ่านทาง Web Server

2. ผลลัพธ์จากการบันทึกข้อมูล สามารถทำได้โดยเขียนโปรแกรมภาษา Python ในโปรแกรม Visual Studio Code โดยบันทึกไฟล์ในรูปแบบ csv ซึ่งใน 1 ไฟล์จะประกอบด้วยข้อมูล 6 คอลัมน์ คือ ค่า ax, ay, az, gx, gy, gz โดยบันทึกไฟล์ทุก 0.125 วินาที ซึ่งตัวอย่างการบันทึกข้อมูลสามารถแสดงได้ดังรูปที่ 24, 25, 26

ax	ay	az	gx	gy	gz
-0.04953	0.23972	0.0018	0.0007	0.00012	0.00043
-0.04492	0.23969	0.00266	0.00073	0.00018	0.00046
-0.04547	0.24008	0.00232	0.00052	-9.00E-05	0.00046
-0.04706	0.23953	0.00128	0.00061	0.00034	0.00067
-0.04568	0.2402	0.00189	0.00079	0.00024	0.0004
-0.04605	0.2395	0.00232	0.00046	0.00018	0.0004
-0.04626	0.23996	0.00113	0.00107	0.00027	0.00064
-0.04654	0.23975	0.00204	0.00061	0.00018	0.00049
-0.04623	0.24152	0.00034	0.00085	0.00037	0.00034
-0.04541	0.24042	0.00262	0.0007	0.00015	0.00034

รูปที่ 24 ตัวอย่างไฟล์ข้อมูลในรูปแบบ csv กลุ่มอยู่นิ่ง

ax	ay	az	gx	gy	gz
0.05914	0.28079	0.08618	-0.00208	0.01837	-0.00714
-0.00473	0.18784	-0.09598	0.03204	-0.00848	0.01175
0.0184	0.18686	-0.01865	-0.00543	-0.03	0.02029
-0.02835	0.15024	-0.00125	-0.00204	-0.01782	0.00201
-0.03198	0.26425	-0.0112	-0.00134	-0.00146	-0.00348
0.01352	0.47818	-0.11456	-0.01773	0.01172	0.02505
-0.07605	0.2952	-0.07318	-0.0072	0.02527	-0.01675
0.05298	0.1756	0.00906	0.00476	0.03186	-0.00665
0.04248	0.2388	0.01227	0.00674	0.01794	0.00928
0.09128	0.37531	-0.048	0.01849	-0.01245	-0.02856

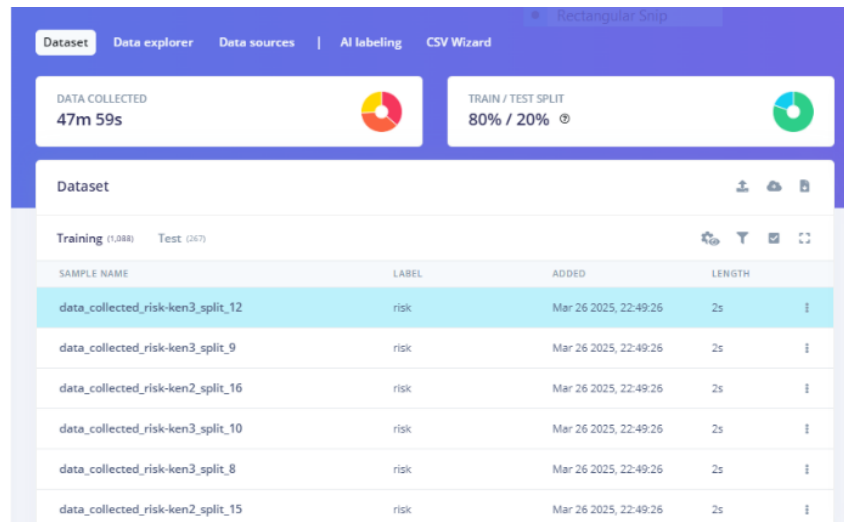
รูปที่ 25 ตัวอย่างไฟล์ข้อมูลในรูปแบบ csv กลุ่มเคลื่อนไหว

ax	ay	az	gx	gy	gz
0.20813	0.40234	-0.07846	0.02112	-0.01132	-0.00888
0.04956	0.22525	-0.05463	-0.01102	-0.04822	0.02908
-0.00284	0.17862	-0.03775	-0.00208	-0.04892	0.00122
-0.04486	0.19775	-0.00409	-0.00104	-0.02686	-0.00247
0.10272	0.40219	-0.0954	-0.02155	-0.04483	0.01834
0.08411	0.35458	-0.02588	0.01862	0.0473	-0.02008
0.05212	0.19742	-0.01498	0.01962	0.05347	-0.0123
0.04611	0.16174	0.01886	0.01114	0.03094	0.00586
0.16559	0.15683	-0.12753	0.01361	0.017	0.02481
0.13785	0.31464	-0.05176	0.00235	-0.0206	0.00787

รูปที่ 26 ตัวอย่างไฟล์ข้อมูลในรูปแบบ csv กลุ่มเสียงอันตราย

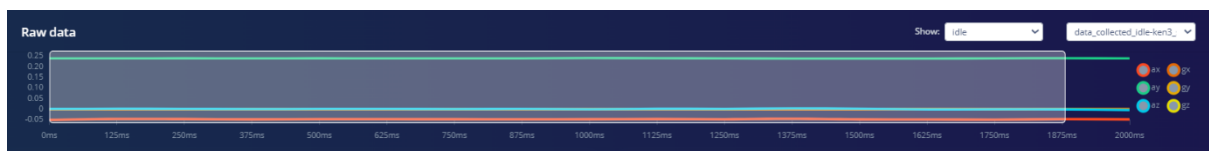
4.2 ผลลัพธ์จากการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse

1. สามารถนำไฟล์ csv ที่ได้จากการบันทึกข้อมูล อัปโหลดลงในส่วนของ Data Acquisition โดยแต่ละไฟล์ csv จะมีการระบุหมวดหมู่ ตามข้อมูลการเคลื่อนไหวที่บันทึกไว้ ประกอบด้วย ชุดข้อมูลการฝึกจำนวน 1,088 ชุดข้อมูล และชุดข้อมูลทดสอบจำนวน 267 ชุดข้อมูล ดังรูปที่ 27



รูปที่ 27 รายการข้อมูลที่รวบรวมไว้ในส่วนของ Data Acquisition

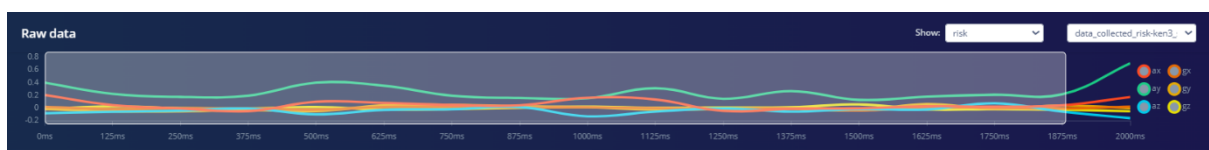
- สามารถแบ่งส่วนข้อมูลจากข้อมูลที่รวบรวมมาได้ โดยกำหนดขนาดของหน้าต่าง เป็น 2 วินาที หรือ 16 ชุดข้อมูล และเลื่อนหน้าต่างต่างทุก 1 วินาที หรือ 8 ชุดข้อมูล ซึ่งแสดงได้ดังรูปที่ 28, 29, 30



รูปที่ 28 การแบ่งส่วนข้อมูลกลุ่มอยู่นิ่ง



รูปที่ 29 การแบ่งส่วนข้อมูลกลุ่มเคลื่อนไหว



รูปที่ 30 การแบ่งส่วนข้อมูลกลุ่มเสียงอันตราย

- สามารถใช้แบบจำลองการวิเคราะห์สเปกตรัม โดยใช้การวิเคราะห์ด้วย FFT เพื่อแปลงข้อมูลจาก โดเมนเวลา ไปเป็นโดเมนความถี่ โดยปรับค่าพารามิเตอร์ FFT length มีค่าเท่ากับ 32 ดังรูปที่ 31 และสามารถแสดงผลจากการวิเคราะห์ด้วย FFT เป็น Spectral power ซึ่งแสดงความสัมพันธ์ระหว่างความถี่ (Frequency) และพลังงาน (Energy) ได้ดังรูปที่ 32, 33, 34

Parameters Autotune parameters

Filter

Scale axes ⓘ

1

Input decimation ratio ⓘ

1

Type ⓘ

none

Analysis

Type ⓘ

FFT

FFT length ⓘ

32

Take log of spectrum? ⓘ

☒

Overlap FFT frames? ⓘ

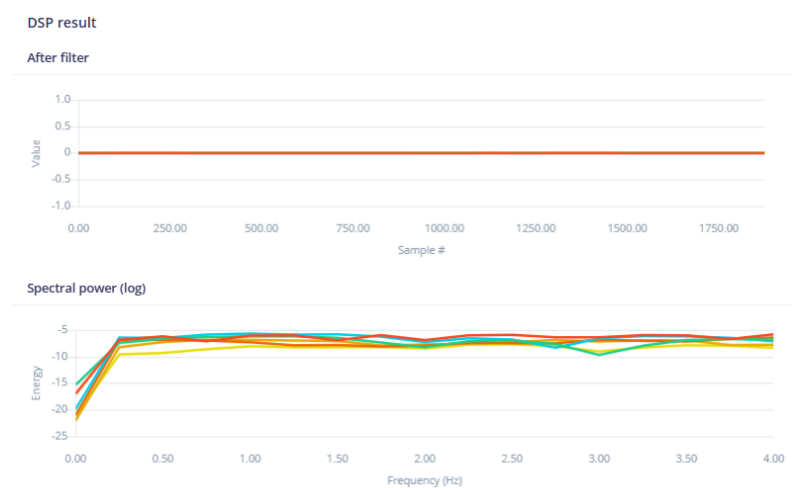
☒

Improve low frequency resolution? ⓘ

☐

Save parameters

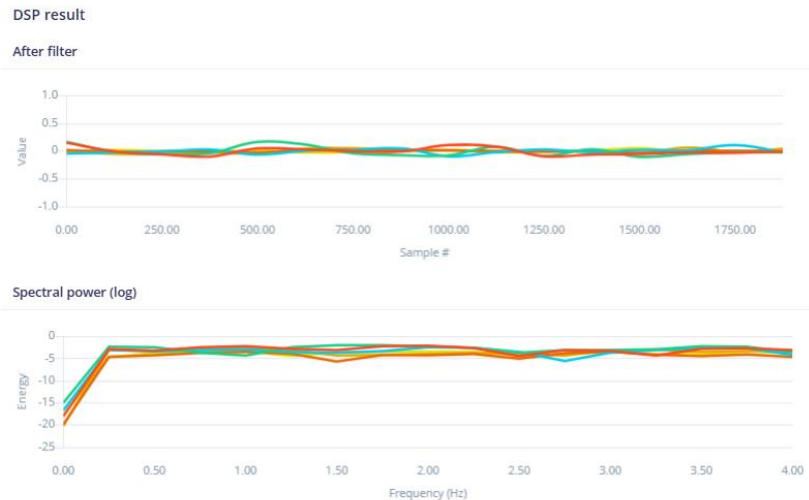
รูปที่ 31 การปรับค่าพารามิเตอร์สำหรับการวิเคราะห์ด้วย FFT



รูปที่ 32 ผลลัพธ์จากการแปลง FFT ของข้อมูลกลุ่มอยู่นิ่ง

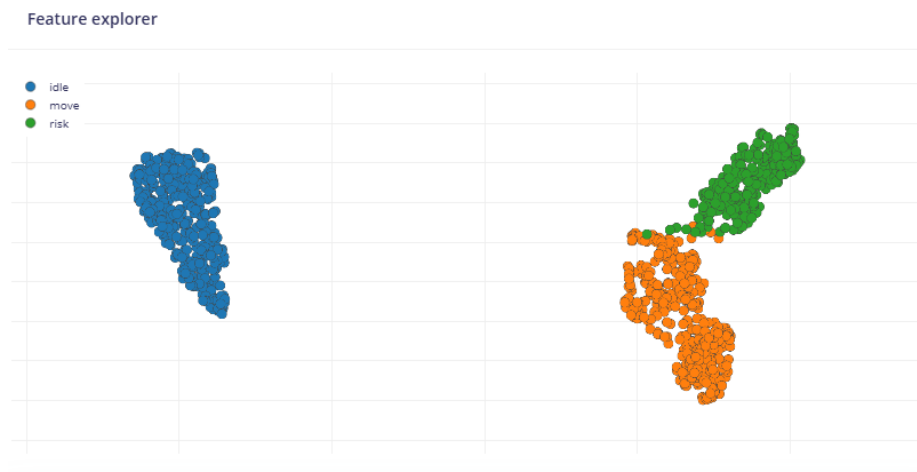


รูปที่ 33 ผลลัพธ์จากการแปลง FFT ของข้อมูลกลุ่มเคลื่อนไหว



รูปที่ 34 ผลลัพธ์จากการแปลง FFT ของข้อมูลกลุ่มเสียงอันตราย

4. หลังจากใช้แบบจำลองการวิเคราะห์สเปกตรัม จะสามารถสร้างลักษณะข้อมูล ซึ่งแสดงถึงลักษณะข้อมูลที่มีความแตกต่างกันของข้อมูล 3 กลุ่ม ดังรูปที่ 35



รูปที่ 35 แผนภาพแสดงลักษณะข้อมูลของข้อมูล 3 กลุ่ม

5. สามารถใช้แบบจำลองการจำแนกประเภท โดยใช้เครือข่ายประสาท โดยชั้นข้อมูลนำเข้า คือ ลักษณะข้อมูล 126 ลักษณะ ที่ได้จากการวิเคราะห์สเปกตรัม และชั้นข้อมูลส่งออกคือ ข้อมูล 3 ประเภท ที่แบบจำลองจำแนกได้

ซึ่งจากการทดลองปรับค่าพารามิเตอร์ โดยมีการกำหนดค่า Number of training cycles = 40 และ Learning rate = 0.001 นอกจากนี้มีการกำหนดจำนวนเซลล์ประสาท ในโครงข่ายประสาทชั้นระหว่างกลาง จำนวน 3 ชั้น โดยมีจำนวน 40, 20, 10 เซลล์ประสาท ตามลำดับ พบว่าแบบจำลองที่ให้ผลลัพธ์การทำนายแม่นยำและมีความซับซ้อนที่ไม่มาก สามารถกำหนดค่าได้ตามรูปที่ 36

Neural Network settings

Training settings

Number of training cycles ⑦

Use learned optimizer ⑦ ☐

Learning rate ⑦

Training processor ⑦

Advanced training settings ▼

Neural network architecture

Input layer (126 features)

Dense layer (40 neurons)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

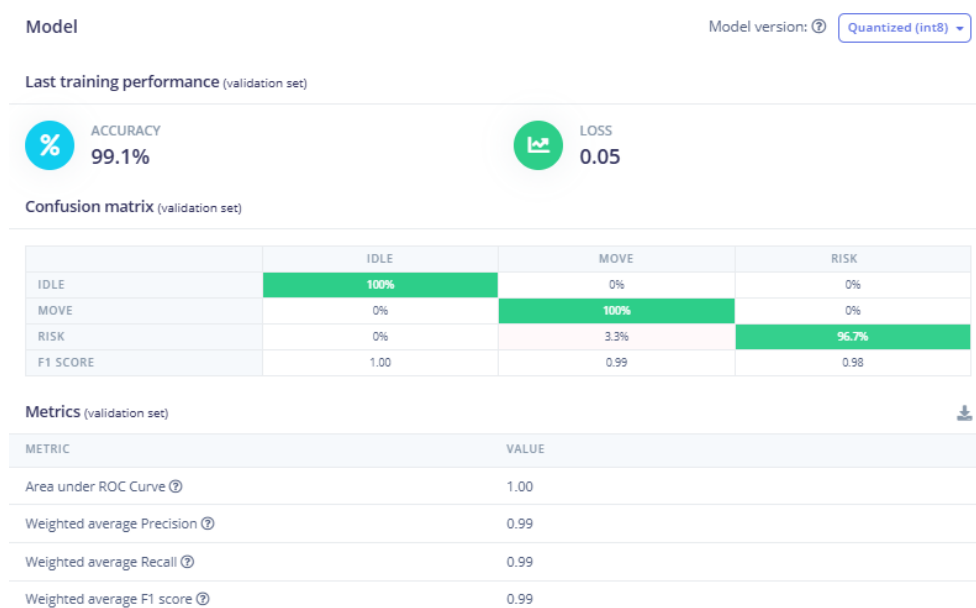
Output layer (3 classes)

Save & train ▼

รูปที่ 36 การปรับค่าพารามิเตอร์ของเครือข่ายประสาทในแบบจำลองการจำแนกประเภท

6. สามารถแสดงผลการประเมินแบบจำลองโดยใช้ชุดข้อมูลตรวจสอบ ดังรูปที่ 37 พบว่าข้อมูลมีความแม่นยำ 99.1% โดยมีความแม่นยำในกลุ่มอยู่นิ่ง 100% กลุ่มเคลื่อนไหว 100% และกลุ่มเสี่ยงอันตราย 96.7% และมีค่า Loss = 0.05

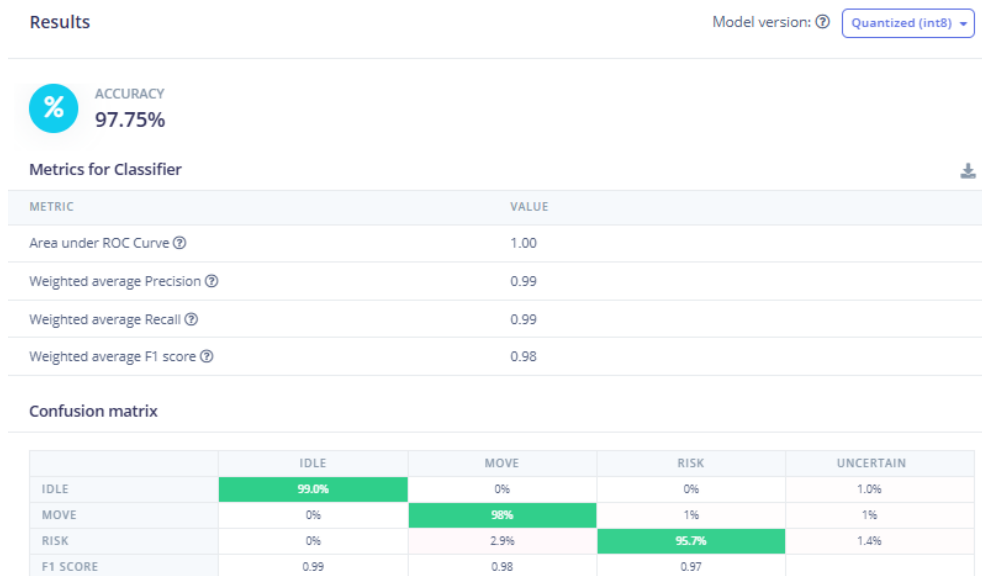
นอกจากนี้ข้อมูลมีค่าพื้นที่ใต้เส้น ROC Curve เท่ากับ 1.00 ค่าความเที่ยงเท่ากับ 0.99 ค่าการเรียกคืนเท่ากับ 0.99 และค่า F1 score เท่ากับ 0.99



รูปที่ 37 การประเมินแบบจำลองโดยใช้ชุดข้อมูลตรวจสอบ

7. สามารถแสดงผลการทดสอบแบบจำลองโดยใช้ชุดข้อมูลทดสอบ ดังรูปที่ 38 พบว่าข้อมูลมีความแม่นยำ 97.75% โดยมีความแม่นยำในกลุ่มอยู่นิ่ง 99% กลุ่มเคลื่อนไหว 98% และกลุ่มเสี่ยงอันตราย 95.7%

นอกจากนี้ข้อมูลมีค่าพื้นที่ใต้เส้น ROC Curve เท่ากับ 1.00 ค่าความเที่ยงเท่ากับ 0.99 ค่าการเรียกคืนเท่ากับ 0.99 และค่า F1 score เท่ากับ 0.98



รูปที่ 38 การประเมินแบบจำลองโดยใช้ชุดข้อมูลทดสอบ

4.3 ผลลัพธ์จากการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์

1. ผลลัพธ์จากการทดสอบข้อมูลการเคลื่อนไหว สามารถทำได้โดยดาวน์โหลดคลังโปรแกรมจากแพลตฟอร์ม Edge Impulse แล้วทำการเขียนโปรแกรมภาษา C++ ในโปรแกรม Arduino IDE สำหรับการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ซึ่งใช้ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense และ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART แล้วแสดงผลข้อมูลผ่านทาง Web Server โดยแสดงผลข้อมูลทุก 2 วินาที ซึ่งสามารถแสดงได้ดังรูปที่ 39



รูปที่ 39 ข้อมูลการทดสอบข้อมูลการเคลื่อนไหวของมนุษย์ที่แสดงผลผ่านทาง Web Server

2. ผลลัพธ์จากการบันทึกข้อมูล สามารถทำได้โดยเขียนโปรแกรมภาษา Python ในโปรแกรม Visual Studio Code โดยบันทึกไฟล์ในรูปแบบ csv โดยใน 1 ไฟล์จะประกอบด้วยข้อมูล 4 คอลัมน์ คือ ค่าความน่าจะเป็นที่แบบจำลองจะทำนายว่าเป็นกลุ่มอยู่นิ่ง กลุ่มเคลื่อนไหว กลุ่มเสี่ยงอันตราย และชื่อคอลัมน์ที่มีค่าความน่าจะเป็นมากที่สุด หรือผลลัพธ์ที่แบบจำลองทำนายได้ โดยบันทึกไฟล์ทุก 2 ซึ่งตัวอย่างการบันทึกข้อมูลสามารถแสดงได้ดังรูปที่ 40

idle	move	risk	class
0.99609	0.00391		0 idle
0.99609	0		0 idle
0.99609	0		0 idle
0.77344	0.22656		0 idle
0.01562	0.98438		0 move
0.01172	0.98828		0 move
0.05078	0.94922		0 move
0.99609	0		0 idle
0.62109	0.37891		0 idle

รูปที่ 40 ผลลัพธ์ข้อมูลที่แบบจำลองทำนายได้ ในรูปแบบ csv

5. บทสรุป

5.1 บทสรุปการทำงาน

จากการดำเนินงานที่ผ่านมา พบว่าสามารถวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล โดยใช้รูปแบบการสื่อสารข้อมูลผ่าน Wi-fi ซึ่งใช้ข้อมูลจากหน่วยวัดแรงเฉื่อย จากไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense แล้วส่งข้อมูลไปยังไมโครคอนโทรเลอร์ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART แล้วแสดงผลข้อมูลผ่านทาง Web Server จากนั้นบันทึกไฟล์ในรูปแบบ csv แล้วส่งไปยังแพลตฟอร์ม Edge Impulse ทำการประมวลผลการเคลื่อนไหวออกเป็น 3 กลุ่ม ได้แก่

- กลุ่มอยู่นิ่ง เป็นกลุ่มที่ไม่มีการเคลื่อนไหวของร่างกาย
- กลุ่มเคลื่อนไหว เป็นกลุ่มที่มีการเคลื่อนไหวของร่างกายจากที่หนึ่งไปยังอีกที่หนึ่ง
- กลุ่มเสี่ยงอันตราย เป็นกลุ่มที่มีการเคลื่อนไหวของร่างกายจากที่หนึ่งไปยังอีกที่หนึ่ง โดยที่ไม่มีความมั่นคงและความสมมาตรของการเดิน ซึ่งเป็นปัจจัยเสี่ยงที่ทำให้เกิดการล้มได้ [4,5]

โดยใช้แบบจำลองการวิเคราะห์สเปกตรัม โดยใช้การวิเคราะห์ด้วย FFT และใช้แบบจำลองการจำแนกประเภทโดยใช้เครือข่ายประสาท ซึ่งมีการปรับค่าพารามิเตอร์ของเครือข่ายประสาท ประกอบด้วย

- ชั้นข้อมูลนำเข้า มีจำนวน 126 เซลล์ประสาท
- ชั้นระหว่างกลาง จำนวน 3 ชั้น มีจำนวน 40, 20, 10 เซลล์ประสาท ตามลำดับ
- ชั้นข้อมูลส่งออก มีจำนวน 3 เซลล์ประสาท

เมื่อทำการปรับค่าพารามิเตอร์ ประเมินแบบจำลอง และทดสอบแบบจำลองแล้ว จะสามารถนำคลังโปรแกรม จากแพลตฟอร์ม Edge Impulse มาใช้กับไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense แล้วส่งข้อมูลไปยังไมโครคอนโทรเลอร์ Arduino Nano 33 IoT โดยสื่อสารข้อมูลระหว่างบอร์ดแบบ UART แล้วแสดงผลข้อมูลผ่านทาง Web Server เพื่อให้สามารถทำนายผลการเคลื่อนไหวของมนุษย์ในแต่ละช่วงเวลา

5.2 แผนการดำเนินงาน

ตารางที่ 1 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	เดือน								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษา Tiny Machine Learning และ Platform ที่ใช้									

2. รวบรวมข้อมูลการเคลื่อนไหวของมนุษย์โดยใช้อุปกรณ์แบบมีสาย									
3. วิเคราะห์ข้อมูลการเคลื่อนไหวของมนุษย์โดยใช้อุปกรณ์แบบมีสาย									
4. รวบรวมข้อมูลการเคลื่อนไหวของมนุษย์โดยใช้อุปกรณ์แบบไร้สาย									
5. วิเคราะห์ข้อมูลการเคลื่อนไหวของมนุษย์โดยใช้อุปกรณ์แบบไร้สาย									
6. ทดสอบและปรับปรุงประสิทธิภาพการทำงานของอุปกรณ์ในสถานการณ์จริง									
7. จัดทำรายงานฉบับสมบูรณ์									

หมายเหตุ สีเทา คือ ความก้าวหน้าที่วางแผนไว้
สีดำ คือ ความก้าวหน้าปัจจุบัน

5.3 ปัญหา อุปสรรค และแนวทางแก้ไข

5.3.1 ปัญหา และอุปสรรค

1. ปัญหาการเชื่อมต่อฮาร์ดแวร์ การต่อสายไฟและสาย UART ทำให้ค่าที่วัดได้จากตัวรับรู้มีค่าที่ผิดพลาด เช่น มีข้อมูลที่เกินขอบเขตที่กำหนด
2. ปัญหาการเขียนโปรแกรม ทำให้ข้อมูลไม่สามารถสื่อสารระหว่างบอร์ดแบบ UART ได้ และข้อมูลไม่สามารถแสดงผลได้ตามที่ต้องการ
3. ปัญหาด้านแบบจำลองการเคลื่อนไหวของมนุษย์ เมื่อทดสอบผ่านแพลตฟอร์ม Edge Impulse พบว่ามีความแม่นยำในระดับหนึ่ง แต่เมื่อทดสอบในบอร์ดจริงข้อมูลที่แบบจำลองทำนายได้ไม่มีความแม่นยำเท่าที่ควร

5.3.2 แนวทางแก้ไข

1. แก้ไขการเชื่อมต่อฮาร์ดแวร์ การต่อสายไฟและสาย UART และการเขียนโปรแกรม ในส่วนของ Arduino IDE และการรับข้อมูลจาก Web Server เพื่อให้ข้อมูลที่แสดงผลได้อย่างปกติ
2. ปรับปรุงแบบจำลองการเคลื่อนไหวของมนุษย์ โดยกำหนดระยะเวลาในการรวบรวมข้อมูลรอบละ 0.125 วินาที และนำข้อมูลไปประมวลผลรอบละ 2 วินาที หรือ 16 ชุดข้อมูล

6. กิตติกรรมประกาศ

โครงการวิศวกรรมไฟฟ้า เรื่อง การวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูล สำเร็จลุล่วงได้ด้วยดี จากการให้ความร่วมมือช่วยเหลือทางด้านโครงการจาก ผศ. ดร.สุรีย พุมรินทร์ ที่ได้สละเวลาอันมีค่ามาให้ความรู้ ซึ่งคุณครูได้ให้แนวทางในการทำโครงการ คอยช่วยเหลือด้านการให้คำปรึกษาในการจัดทำโครงการ และให้คำแนะนำในเรื่องการทำโครงการเป็นอย่างดี

ขอขอบคุณอาจารย์ทุกท่าน บิดา มารดา และขอบคุณเพื่อน ๆ ที่มีส่วนร่วมในการแลกเปลี่ยนความคิดเห็นและช่วยในการทดลองต่าง ๆ รวมถึงการสนับสนุนทางด้านเทคนิค และข้อมูลที่เป็นวิธีการต่าง ๆ ในการทำงานให้เสร็จสมบูรณ์ และส่งเสริมในการทำโครงการจนเกิดความสำเร็จลุล่วงไปได้ด้วยดี

7. เอกสารอ้างอิง

- [1] สถาบันปรับโครงสร้างร่างกาย อริยะ, "อิริยาบถ ยืน เดิน นั่ง นอน", [Online]. Available: <https://www.ariyawellness.com/knowledge/posture/bodily-movement-all/>. [Accessed 5 March 2025]
- [2] สมาคมอาชีวอนามัยและความปลอดภัยในการทำงาน, "การป้องกันอันตรายจากการ ป้องกันการลื่น สะดุด หรือล้ม", [Online]. Available: <https://www.ohswa.or.th/17835288/hse-morning-talk-by-safety-ihs-buu-ep11>. [Accessed 5 March 2025]
- [3] โรงพยาบาลเฉพาะทางกระดูกและข้อ ข้อดีมีสุข จำกัด, " "การเดินให้ช้าลง" ช่วยป้องกันการหกล้มของผู้สูงอายุ", [Online]. Available: <https://kdmshospital.com/article/walk-slower-better-for-adults/>. [Accessed 5 March 2025]
- [4] S. Jiang, B. Zhang, and D. Wei, "The Elderly Fall Risk Assessment and Prediction Based on Gait Analysis," in *2011 11th IEEE International Conference on Computer and Information Technology*, 2011, pp. 176-179. doi: 10.1109/CIT.2011.82.
- [5] J. Howcroft, J. Kofman, and E. D. Lemaire, "Review of fall risk assessment in geriatric populations using inertial sensors," *J. NeuroEng. Rehabil.*, vol. 10, no. 91, pp. 1-12, Aug. 2013. doi: 10.1186/1743-0003-10-91.
- [6] Vithan Minaphinant, "Machine Learning คืออะไร?", [Online]. Available: <https://medium.com/investic/machine-learning-คืออะไร-fa8bf6663c07>. [Accessed 7 March 2025]
- [7] Yok Thanawan, "Machine learning (ML) คืออะไร? เทคโนโลยีที่องค์กรต้องรู้", [Online]. Available: <https://www.disruptignite.com/blog/machine-learning>. [Accessed 7 March 2025]

- [8] Cyber Elite, "Machine Learning เทคโนโลยีประโยชน์ครอบคลุมจักรวาล", [Online]. Available: <https://www.cyberelite.co.th/blog/machine-learning/>. [Accessed 7 March 2025]
- [9] กอบเกียรติ สระอุบล, เรียนรู้ Data Science และ AI: Machine Learning ด้วย Python, กรุงเทพมหานคร: สำนักพิมพ์ อินเทอร์เน็ตมีเดีย, 2563.
- [10] กอบเกียรติ สระอุบล, เรียนรู้ AI: Deep Learning ด้วย Python, กรุงเทพมหานคร: สำนักพิมพ์ อินเทอร์เน็ตมีเดีย, 2565.
- [11] Pakpoom Thaweesitthichat, "Deep Learning มันทำงานอย่างไรกันนะ?", [Online]. Available: <https://medium.com/@pakpoom.thawee/deep-learning-มันทำงานอย่างไรกันนะ-8eef57561d19>. [Accessed 7 March 2025]
- [12] Tangerine, "สรุป Deep Learning คืออะไร ทำงานอย่างไร? ฉบับเข้าใจง่าย", [Online]. Available: <https://www.tangerine.co.th/blogs/data-analytics-artificial-intelligence/what-is-deep-learning/>. [Accessed 7 March 2025]
- [13] Mr.P L, "Deep Learning แบบฉบับคนสามัญชน EP 1 : Neural Network History", [Online]. Available: <https://medium.com/mmp-li/deep-learning-แบบฉบับคนสามัญชน-ep-1-neural-network-history-f7789236a9a3>. [Accessed 7 March 2025]
- [14] ชิตพงษ์ กิตตินราดร. "Neural Network Algorithm", [Online]. Available: <https://guopai.github.io/ml-blog14.html>. [Accessed 7 March 2025]
- [15] Kasidis Satangmongkol. "เข้าใจการทำงานพื้นฐานของ Neurons ใน Neural Networks" [Online]. Available: <https://datarockie.com/blog/how-neurons-work/#output-layer>. [Accessed 7 March 2025]
- [16] Mr.P L, "Deep Learning แบบฉบับสามัญชน EP 2 Optimization & Activation Function เรียนกันสบายๆสไตล์ชิลๆ", [Online]. Available: <https://medium.com/mmp-li/deep-learning-แบบฉบับสามัญชน-ep-2-optimization-activation-function-เรียนกันสบายๆสไตล์ชิลๆ-9feb5a87e3b2>. [Accessed 7 March 2025]
- [17] Subbrain, "Loss Function คืออะไร?", [Online]. Available: <https://www.sub-brain.com/datait/loss-function/>. [Accessed 7 March 2025]
- [18] Pimprom Khunpet, "สรุปการเลือกใช้ Loss Function ในการพัฒนา Deep Learning Model", [Online]. Available: <https://pimprom-k.medium.com/สรุปการเลือกใช้-loss-function-ในการพัฒนา-deep-learning-model-a547c2fb77fd>. [Accessed 7 March 2025]
- [19] Kurtis Pykes, "What is TinyML? An Introduction to Tiny Machine Learning," [Online]. Available: <https://www.datacamp.com/blog/what-is-tinyml-tiny-machine-learning>

[Accessed 10 March 2025]

- [20] อองตวน ทาร์ติฟ, "TinyML: อนาคตของแมชชีนเลิร์นนิงในระดับจิ๋ว," [Online]. Available: <https://www.unite.ai/th/tinyml-the-future-of-machine-learning-on-a-minuscule-scale/>. [Accessed 10 March 2025]
- [21] Mindphp, "หลักการการแบ่ง Dataset ในการพัฒนา Machine Learning หรือ Deep Learning" [Online]. Available: <https://www.mindphp.com/บทเรียนออนไลน์/python-tensorflow/8576-how-to-split-dataset.html>. [Accessed 10 March 2025]
- [22] Devopedia, "Confusion Matrix." [Online]. Available: <https://devopedia.org/confusion-matrix>. [Accessed 10 March 2025]
- [23] Evidently AI Team, "How to explain the ROC curve and ROC AUC score?" [Online]. Available: <https://www.evidentlyai.com/classification-metrics/explain-roc-curve>. [Accessed 10 March 2025]
- [24] Pasith Thanapatpisarn, "วัดประสิทธิภาพของแบบจำลองด้วย Evaluation Metrics Part 2", [Online]. Available: <https://datascihaeng.medium.com/evaluation-matrix-part2-a7d83aea6537>. [Accessed 10 March 2025]
- [25] Edge Impulse, "AI สำหรับอุปกรณ์ Edge ทุกชนิด", [Online]. Available: <https://edgeimpulse.com/>. [Accessed 10 March 2025]
- [26] Edge Impulse, "Spectral features", [Online]. Available: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/spectral-features>. [Accessed 10 March 2025]
- [27] Edge Impulse, "Classification (Keras)", [Online]. Available: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/classification>. [Accessed 10 March 2025]
- [28] Edge Impulse, "Learning blocks", [Online]. Available: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks#neural-network-architecture>. [Accessed 10 March 2025]
- [29] Arduino, "Nano 33 BLE Sense", [Online]. Available: <https://docs.arduino.cc/hardware/nano-33-ble-sense/>. [Accessed 10 March 2025]
- [30] Ankit Negi, "Arduino Nano 33 BLE Sense Pinout, Introduction & Specifications", [Online]. Available: <https://www.etechnophiles.com/arduino-nano-33-ble-sense-pinout-introduction-specifications/>. [Accessed 10 March 2025]

- [31] Arduino, " Nano 33 IoT", [Online]. Available: <https://docs.arduino.cc/hardware/nano-33-iot/>. [Accessed 10 March 2025]
- [32] Ankit Negi, " Arduino Nano 33 IoT Pinout & Specs Guide", [Online]. Available: <https://www.etechnophiles.com/arduino-nano-33-iot-pinout-spec-board-layout/>. [Accessed 10 March 2025]
- [33] Jouav Unmanned Aircraft System, "คู่มือฉบับสมบูรณ์เกี่ยวกับหน่วยวัดแรงเฉื่อย (IMU)", [Online]. Available: <https://www.jouav.com/blog/inertial-measurement-unit.html>. [Accessed 10 March 2025]
- [34] Automatic Addison, "Yaw, Pitch, and Roll Diagrams Using 2D Coordinate Systems", [Online]. Available: <https://automaticaddison.com/yaw-pitch-and-roll-diagrams-using-2d-coordinate-systems/>. [Accessed 10 March 2025]
- [35] เคนชิน, " Gyroscope กับ Accelerometer เซนเซอร์คืออะไร ? และแตกต่างกันอย่างไร ?", [Online]. Available: <https://tips.thaiware.com/427.html>. [Accessed 10 March 2025]
- [36] Arduino, " เชื่อมต่อสองบอร์ด Nano Every ผ่าน UART", [Online]. Available: <https://docs.arduino.cc/tutorials/nano-every/uart/>. [Accessed 10 March 2025]

8. ภาคผนวก

8.1 Readme File สำหรับอธิบายระบบ

ระบบการวิเคราะห์การเคลื่อนไหวของมนุษย์ด้วยการเรียนรู้ของเครื่องบนอุปกรณ์ประมวลผลข้อมูลใช้อุปกรณ์ ได้แก่ ไมโครคอนโทรเลอร์ Arduino Nano 33 BLE Sense, ไมโครคอนโทรเลอร์ Arduino Nano 33 IoT, แบตเตอรี่สำรองที่มีพอร์ตจ่ายไฟ USB อย่างน้อย 2 ช่อง, สายต่อจัมเปอร์ 3 เส้น, breadboard 170 holes 2 อัน, สายชาร์จ USB Micro B 2 เส้น และสามารถดาวน์โหลดไฟล์โปรแกรมได้จาก

<https://drive.google.com/drive/folders/1Ec1rohDuirw-QXnumup4d0dW9OuCtK40?usp=sharing>

8.1.1 ขั้นตอนการติดตั้งคลังโปรแกรม

1. ติดตั้งคลังโปรแกรม Arduino ได้แก่ Arduino_LSM9DS1, Wi-FiNINA, SPI
โดยเลือกที่ Sketch > Include Library > Manage Libraries
2. ติดตั้งคลังโปรแกรม Arduino .zip ที่ชื่อว่า ei-project_final_3-arduino-1.0.3.zip
โดยเลือกที่ Sketch > Include Library > Add .ZIP Library
3. ติดตั้งคลังโปรแกรม requests ในโปรแกรม Visual Studio Code
โดยใช้คำสั่ง pip install requests ใน Terminal

8.1.2 ขั้นตอนการใช้งานโปรแกรม

1. ส่วนของการรวบรวมข้อมูล (Train)
 - นำไฟล์ชื่อว่า uart_ble_sense.ino ลงบอร์ด Arduino Nano 33 BLE Sense
 - นำไฟล์ชื่อว่า uart_iot.ino ลงบอร์ด Arduino Nano 33 IoT
 - แก้ไขไฟล์ arduino_secrets.h โดยกำหนด ssid และ password แล้วเชื่อมต่อ Network ที่กำหนดเข้ากับคอมพิวเตอร์
 - เปิดไฟล์ชื่อว่า csv_save_wifi_long_1.25.ipynb ใน Visual Studio Code เมื่อต้องการรวบรวมข้อมูลให้เลือกคำสั่ง Run All
 - หากต้องการข้อมูลแบบ Real Time สามารถเข้าถึงได้จาก <http://192.168.4.1>
2. ส่วนของการทดสอบข้อมูล (Test)
 - นำไฟล์ชื่อว่า uart_ble_sense_test.ino ลงบอร์ด Arduino Nano 33 BLE Sense
 - นำไฟล์ชื่อว่า uart_iot_test.ino ลงบอร์ด Arduino Nano 33 IoT
 - แก้ไขไฟล์ arduino_secrets.h โดยกำหนด ssid และ password แล้วเชื่อมต่อ Network ที่กำหนดเข้ากับคอมพิวเตอร์

- เปิดไฟล์ชื่อว่า csv_save_wifi_long_test_1.25.ipynb ใน Visual Studio Code เมื่อต้องการทดสอบข้อมูลให้เลือกคำสั่ง Run All
- หากต้องการข้อมูลแบบ Real Time สามารถเข้าถึงได้จาก <http://192.168.4.1>

8.2 Source Code สำหรับการวิเคราะห์ข้อมูลโดยใช้แพลตฟอร์ม Edge Impulse

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten,
Reshape, MaxPooling1D, MaxPooling2D, AveragePooling2D, BatchNormalization, Permute,
ReLU, Softmax
from tensorflow.keras.optimizers.legacy import Adam

EPOCHS = args.epochs or 40
LEARNING_RATE = args.learning_rate or 0.001
# If True, non-deterministic functions (e.g. shuffling batches) are not used.
# This is False by default.
ENSURE_DETERMINISM = args.ensure_determinism
# this controls the batch size, or you can manipulate the tf.data.Dataset objects yourself
BATCH_SIZE = args.batch_size or 32
if not ENSURE_DETERMINISM:
    train_dataset = train_dataset.shuffle(buffer_size=BATCH_SIZE*4)
train_dataset=train_dataset.batch(BATCH_SIZE, drop_remainder=False)
validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)

# model architecture
model = Sequential()
model.add(Dense(40, activation='relu',
    activity_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(Dense(20, activation='relu',
    activity_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(Dense(10, activation='relu',
```

```

activity_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(Dense(classes, name='y_pred', activation='softmax'))

# this controls the learning rate
opt = Adam(learning_rate=LEARNING_RATE, beta_1=0.9, beta_2=0.999)
callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count, epochs=EPOCHS,
ensure_determinism=ENSURE_DETERMINISM))

# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
model.fit(train_dataset, epochs=EPOCHS, validation_data=validation_dataset, verbose=2 ,
callbacks=callbacks)

# Use this flag to disable per-channel quantization for a model.
# This can reduce RAM usage for convolutional models, but may have
# an impact on accuracy.
disable_per_channel_quantization = False

```