

**แพลตฟอร์มสำหรับตรวจจับท่าทางมือ**  
**SMART PLATFORM FOR GESTURE RECOGNITION**



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2561

แพลตฟอร์มสำหรับตรวจจับท่าทางมือ  
SMART PLATFORM FOR GESTURE RECOGNITION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง แพลตฟอร์มสำหรับตรวจจับท่าทางมือ

SMART PLATFORM FOR GESTURE RECOGNITION

ผู้จัดทำ

1. นายพัทธวีร์ ชุมภูวร รหัสนักศึกษา 58010866
2. นายพิทวัส คุณกะมุต รหัสนักศึกษา 58010886



  
อาจารย์ที่ปรึกษา  
(ผศ. ชนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# แพลตฟอร์มสำหรับตรวจจับท่าทางมือ

นายพัทธวีร์ ชุมภูว	58010866
นายพิทวัส คุณกะมุต	58010886
ผศ. ธนา หงษ์สุวรรณ	อาจารย์ที่ปรึกษา
ปีการศึกษา 2561	

## บทคัดย่อ

ถุงมือสำหรับตรวจจับท่าทางมือในปัจจุบันมักได้รับการพัฒนามาเพื่อวัตถุประสงค์ใดอย่างหนึ่ง เช่น เพื่อใช้งานร่วมกับเทคโนโลยี Virtual Reality หรือเพื่อใช้แปลภาษามือเป็นคำหรือตัวสะกดภาษาอังกฤษ แต่สำหรับท่าทางที่แสดงถึงภาษาไทยนั้นจะมีท่าที่แตกต่างออกไป โดยเฉพาะตัวสะกดไทยที่ไม่สามารถนำถุงมือที่แปลภาษามือที่เป็นภาษาอังกฤษมาใช้ได้โดยตรง เพราะเกิดข้อจำกัดเรื่องท่าทางที่สามารถตรวจจับได้ในด้านฮาร์ดแวร์หรือซอฟต์แวร์

โครงการนี้จึงมีเป้าหมายเพื่อพัฒนาแพลตฟอร์มที่ประกอบไปด้วยถุงมือที่ใช้เทคโนโลยี Deep Learning ที่มีความนิยมในปัจจุบันเพื่อตรวจจับท่าทางมือ และเว็บแอปพลิเคชันให้ผู้ใช้สามารถตั้งค่าชุดของท่าทางที่จะใช้เป็นอินพุตและการกระทำที่ต้องการจากท่าทางที่ตั้งค่าไว้ได้ โดยเบื้องต้นจะสามารถตรวจจับท่าทางที่แสดงถึงตัวสะกดไทย และท่าพื้นฐานอื่น ๆ ทั้งท่านิ่งและเคลื่อนไหว และสามารถตั้งค่าการกระทำได้ทั้งการแสดงผลบนแอปพลิเคชัน แสดงข้อความบนจอแสดงผลบนถุงมือ และส่งคำขอ HTTP ไปยังอุปกรณ์อื่นในเครือข่ายเป็นต้น เนื่องจากผู้ใช้สามารถตั้งค่าได้อย่างอิสระ แพลตฟอร์มนี้จึงรองรับการใช้งานที่หลากหลายวัตถุประสงค์มากขึ้น

ถุงมือที่ใช้ในการพัฒนาประกอบด้วยหน่วยไมโครคอนโทรลเลอร์ NanoPI-DUO เพื่อใช้ประมวลผล และเซนเซอร์หน่วยวัดความเฉื่อย Polulu MinIMU9-v5 6 ตัวเพื่อวัดการเคลื่อนที่ของหลังมือและนิ้วทั้ง 5 นำมาประมวลผลเป็นท่าทางด้วยโมเดลโครงข่ายประสาทเทียมแบบ LSTM ขอบเขตของถุงมือคือสามารถตรวจจับท่าทางที่เป็นภาษามือตัวสะกดไทยได้อย่างน้อย 30 ตัว และสามารถตั้งค่าท่าทางดังกล่าวให้มีการกระทำแบบอื่นได้ผ่านเว็บแอปพลิเคชัน โดยแพลตฟอร์มทั้งหมดจะอยู่ภายในถุงมือและไม่จำเป็นต้องใช้อุปกรณ์อื่นเพิ่มเติม

# Smart Platform for Gesture Recognition

Pathawee Chumpuvorn 58010866

Pitawat Koonkamoot 58010886

Asst. Prof. Thana Hongsuwan Advisor

Academic Year 2561

## ABSTRACT

Hand gesture recognition technologies recently are likely developed for one purpose, for example, Virtual Reality and English sign language translation. But for the Thai language, because of more and different postures, Existing technologies cannot be used to recognize new posture with the limitation of hardware or software.

This project aims to develop a platform consists of a deep-learning assisted gesture recognition glove and a web application which allows users to set up Gestures and Actions. For Gestures, the glove can recognize most Thai spelling and some other basic postures, Still and moving postures. For Actions, users can set whether to display something on the web application, or display something on OLED screen module, or send HTTP requests to other devices in the glove's network, etc. Because of the wild range of configuration, this enables the platform to do not just detecting Thai spelling gestures and display results but can be set up for a variety of purposes.

The glove is powered by NanoPI-DUO microcontroller for processing and six Polulu MinIMU9-v5 inertial measurement unit for gathering fingers and back of the hand movement data. The data is passed and processed through LSTM neural network model. The scope of this project is to recognize at least 30 Thai spelling gestures and can be set to do different actions with the web application. All parts of the platform are inside the glove and do not require additional equipment.

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความช่วยเหลือของ ผศ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำ แนวทาง ให้ข้อคิดเห็น และสนับสนุนอุปกรณ์ต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการทำโครงการงาน

ขอขอบคุณ รศ. ดร.อรฉัตร จิตต์โสภักตร์ อาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ให้แนวทางในการพัฒนาระบบในส่วนปัญญาประดิษฐ์ให้แม่นยำยิ่งขึ้น

ขอขอบคุณ นายจิรภัทร ศุภกุลนิธิไพศาล และเพื่อนสมาชิกในห้องปฏิบัติการ PEARL ภาควิชาวิศวกรรมไฟฟ้า ที่ได้ช่วยให้คำแนะนำ และช่วยแก้ไขจุดที่มีงานสามารถทำงานได้อย่างไร้ปัญหา

ขอขอบคุณ นายศุภวิชญ์ วาตะฐานะวโร และนายสุชัชจัตต์ ตั้งจิตนบ เพื่อนภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้คำแนะนำในการเริ่มต้นการสร้างโมเดลปัญญาประดิษฐ์

นอกจากนี้ต้องขอขอบคุณเพื่อนในห้องปฏิบัติการ ที่ได้ช่วยเป็นทดสอบ และเป็นกำลังใจให้ตลอดในการดำเนินโครงการงาน

พัทธวีร์ ชุมภูว

พิทวัส คุณกะมุต

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
กิตติกรรมประกาศ.....	II
สารบัญ .....	III
สารบัญตาราง .....	IV
สารบัญรูป .....	V
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.4 ขอบเขตของโครงการ .....	3
1.5 ข้อจำกัดของโครงการ .....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	4
2.1 เทคโนโลยีด้านฮาร์ดแวร์.....	4
2.2 เทคโนโลยีด้านการประมวลผลทำทางมือ .....	8
2.3 เทคโนโลยีด้านฐานข้อมูล .....	13
2.4 เทคโนโลยีด้านแอปพลิเคชัน.....	13
2.5 ทฤษฎีเกี่ยวกับการคำนวณ.....	15
2.6 งานวิจัยที่เกี่ยวข้อง.....	18
บทที่ 3 การออกแบบและพัฒนา.....	22
3.1 การออกแบบระบบโดยรวม .....	22
3.2 การออกแบบลงมือ .....	24
3.3 การออกแบบเว็บแอปพลิเคชัน .....	27
3.4 การออกแบบฐานข้อมูล.....	34
3.5 การออกแบบชุดข้อมูล.....	34
3.6 การออกแบบโมเดล Machine Learning .....	36
บทที่ 4 การทดลอง .....	40

4.1 การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ .....	43
4.2 การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ.....	44
4.3 การทดลองเกี่ยวกับเว็บแอปพลิเคชัน .....	46
4.4 การทดลองเกี่ยวกับ Machine Learning .....	49
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	59
5.1 บทสรุป .....	59
5.2 ปัญหาและอุปสรรค.....	60
5.3 แนวทางแก้ไขปัญหา.....	61
5.4 แนวทางการพัฒนา .....	62



# สารบัญตาราง

	หน้า
2.1 ข้อมูลจำเพาะ Nanopi-DUO .....	4
2.2 ข้อมูลจำเพาะเซนเซอร์ Polulu MinIMU9-v5 .....	6
4.1 ผลการทดสอบความแม่นยำโมเดลแบบที่ 1.....	50
4.2 Confusion Matrix ของโมเดลแบบที่ 1 .....	50
4.3 ผลการทดสอบความแม่นยำโมเดลแบบที่ 2.....	51
4.4 Confusion Matrix ของโมเดลแบบที่ 2 .....	51
4.5 ผลการทดสอบความแม่นยำโมเดลแบบที่ 3.....	53
4.6 Confusion Matrix ของโมเดลแบบที่ 3 .....	53
4.7 ผลการทดสอบความแม่นยำโมเดลแบบที่ 4.....	55
4.8 Confusion Matrix ของโมเดลแบบที่ 4 .....	55
4.9 ผลการทดสอบความแม่นยำโมเดลแบบที่ 4 โดยเพิ่มท่าทาง.....	57
4.10 Confusion Matrix ของโมเดลแบบที่ 4 โดยเพิ่มท่าทาง .....	58

# สารบัญรูป

รูป	หน้า
1.1 ถุงมือตรวจจับท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality .....	1
1.2 ถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือ .....	1
2.1 แผนภาพโครงสร้าง, ขาเชื่อมต่อของ NanoPI-DUO .....	5
2.2 Polulu MinIMU9-v5 .....	6
2.3 SSD1306 .....	7
2.4 ตัวอย่างเครือข่ายการเชื่อมต่อของบัส I2C .....	7
2.5 TCA9548A .....	8
2.6 สัญลักษณ์ TensorFlow .....	8
2.7 สถาปัตยกรรมโครงข่ายประสาทเทียม .....	9
2.8 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ RNN .....	10
2.9 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ LSTM .....	11
2.10 สัญลักษณ์ MongoDB .....	13
2.11 สัญลักษณ์ VueJS .....	13
2.12 สัญลักษณ์ Flask .....	14
2.13 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูลระหว่าง REST กับ WebSocket .....	14
2.14 ระบบของตัวกรองร่วมสองแบบ .....	16
2.15 ระบบของตัวกรองคาถมาน .....	17
2.16 ถุงมือของงานวิจัย A Gesture Detection Glove For Human-computer Interaction .....	18
2.17 Rotation Matrix .....	19
2.18 แผนภูมิเปรียบเทียบความแม่นยำและเวลาที่ใช้ระหว่างโมเดล Machine Learning แต่ละตัวของงานวิจัย A Gesture Detection Glove For Human-computer Interaction .....	19
2.19 แผนภูมิเวลาในการวิเคราะห์ท่าของงานวิจัย A real-time continuous gesture recognition system for sign language .....	20
2.20 ถุงมือจากงานวิจัย 3-D hand motion tracking and gesture recognition using a data glove .....	21
3.1 การเชื่อมต่อระหว่างฮาร์ดแวร์ .....	22
3.2 โครงสร้างระบบ .....	23
3.3 ขั้นตอนการทำงานแบ่งตามส่วนของการตรวจจับท่าทาง .....	23

3.4	หน้าที่การทำงานของแต่ละส่วนของระบบ .....	24
3.5	โครงสร้างถุงมือแบบที่ 1 .....	25
3.6	โครงสร้างถุงมือแบบที่ 2 .....	26
3.7	แผนผังลำดับการทำงานของกรตรวจจับท่าทาง .....	27
3.8	แถบเมนูด้านซ้ายของเว็บแอปพลิเคชัน .....	28
3.9	หน้าเว็บแอปพลิเคชันส่วนของผู้ใช้ .....	28
3.10	แผนผังลำดับการทำงานของกรเรียกดู/เพิ่ม/แก้ไข/ลบ ท่าทางของผู้ใช้ .....	31
3.11	หน้าเว็บแอปพลิเคชันส่วนของนักพัฒนา .....	32
3.12	แผนผังลำดับการทำงานของกรบันทึกและส่งออกชุดข้อมูลของนักพัฒนา .....	33
3.13	แผนภาพฐานข้อมูล .....	34
3.14	สถาปัตยกรรมโครงข่ายประสาทเทียม โมเดลที่ 1 .....	36
3.15	สถาปัตยกรรมโครงข่ายประสาทเทียม โมเดลที่ 2 .....	37
3.16	สถาปัตยกรรมโครงข่ายประสาทเทียม โมเดลที่ 3 .....	38
3.17	สถาปัตยกรรมโครงข่ายประสาทเทียม โมเดลที่ 4 .....	39
4.1	ทำนองที่ใช้ในการทดสอบ .....	40
4.2	ท่าเคลื่อนไหวที่ใช้ในการทดสอบ - พยัญชนะ ข .....	41
4.3	ท่าเคลื่อนไหว - พยัญชนะ ค .....	41
4.4	ท่าเคลื่อนไหว - พยัญชนะ ง .....	41
4.5	ท่าชุดสุดท้าย .....	42
4.6	แผนภาพชุดข้อมูลของท่าแบมมือ (ถุงมือแบบที่ 1) .....	43
4.7	แผนภาพชุดข้อมูลเฉพาะนิ้วชี้ของท่าแบมมือ (ถุงมือแบบที่ 2) .....	45
4.8	การแสดงผลข้อมูลเป็นแผนภาพเส้นที่เปลี่ยนแปลงตามเวลา .....	47
4.9	การจำลองการแสดงผลการตรวจจับท่าทาง .....	48

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาของปัญหา

ในปัจจุบันมีการพัฒนาอุปกรณ์ต่าง ๆ โดยมีเป้าหมายเพื่อให้ผู้ใช้สามารถใช้งานอุปกรณ์ทางอิเล็กทรอนิกส์โดยใช้ร่างกายของตนเองเป็นอินพุตได้ และได้รับความรู้สึกสมจริง หรือ สะดวกสบายมากขึ้นหากเทียบกับการใช้งานอินพุตปกติอย่างเช่นคีย์บอร์ด เมาส์ หรือจอสัมผัส

หากกล่าวถึงอุปกรณ์ที่พยายามตรวจจับท่าทางของมือ ในปัจจุบันก็มีการพัฒนาขึ้นมา หลากหลายประเภทเพื่อวัตถุประสงค์ได้อย่างหนึ่งอย่างชัดเจน ตัวอย่างเช่น ถุงมือตรวจจับท่าทางมือ ที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality เพื่อใช้ในการเล่นเกมเพื่อความบันเทิง หรือฝึกฝน ทักษะปฏิบัติเสมือนจริง และถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือเป็นคำหรือตัวสะกด ภาษาอังกฤษ เป็นต้น



รูป 1.1 ถุงมือตรวจจับท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality



รูป 1.2 ถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือ

หากกล่าวถึงภาษามือที่เป็นภาษาไทย เนื่องจากถุงมือตรวจจับท่าทางมือเพื่อแปลงเป็นตัวสะกด ภาษาอังกฤษตามตัวอย่างที่กล่าวมาข้างต้นอาจมีท่าทางต่าง ๆ ที่ใช้เป็นอินพุตรวมถึงเอาต์พุตอยู่ใน ขอบเขตที่ได้กำหนดไว้เพียงเท่านั้น ไม่สามารถนำมาใช้กับภาษาไทยที่มีท่าทางแตกต่างออกไปจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษามืออื่น ๆ ได้ โดยเฉพาะตัวสะกดที่มีมากกว่าอย่างเห็นได้ชัด หรืออาจมีท่าที่อยู่นอกเหนือขอบเขตที่ถนัดมือจะสามารถตรวจจับได้ในระดับฮาร์ดแวร์ หรือหากใช้ถนัดมือตรวจจับท่าทางสำหรับนักพัฒนา ก็อาจต้องมีการพัฒนาต่อโดยใช้ชุดพัฒนาซอฟต์แวร์ที่รองรับจากผู้พัฒนา ร่วมกับความรู้เฉพาะด้าน อีกทั้งอาจมีราคาสูง ไม่คุ้มค่าที่จะนำมาพัฒนาเพื่อวัตถุประสงค์เดียว ซึ่งเป็นเหตุผลหนึ่งที่ทำให้ควรมีการพัฒนาถนัดมือตรวจจับท่าทางสำหรับภาษามือที่เป็นภาษาไทยขึ้นมาเอง หรือถนัดมือตรวจจับท่าทางที่สามารถใช้งานในวัตถุประสงค์อื่นได้นอกจากภาษามือ

ทางผู้พัฒนาจึงสนใจที่จะสร้างแพลตฟอร์มเพื่อตรวจจับท่าทางมือ ที่ประกอบไปด้วย 3 ส่วน คือถนัดมือที่ติดตั้งไมโครคอนโทรลเลอร์และเซนเซอร์ไว้ ส่วนตรวจจับท่าทาง และส่วนแอปพลิเคชันที่เป็นส่วนติดต่อผู้ใช้ โดยทุกส่วนนั้นอยู่บนถนัดมือ และไม่จำเป็นต้องใช้อุปกรณ์ภายนอกในการประมวลผลเพิ่มเติม ส่วนและมีการใช้เทคโนโลยี Machine Learning ที่ได้รับการพัฒนาประสิทธิภาพขึ้นและมีความนิยมในปัจจุบันเข้ามาใช้เพื่อตรวจจับท่าทางมือ โดยเริ่มจากท่าทางที่แสดงถึงตัวสะกดไทย เพื่อลดข้อจำกัดในด้านท่าทางที่กล่าวมาข้างต้นลง จะมีแอปพลิเคชันให้ผู้ใช้สามารถตั้งค่าชุดของท่าทางที่จะใช้เป็นอินพุตได้ นอกจากนั้นแอปพลิเคชันจะสามารถตั้งค่าการกระทำที่ต้องการจากท่าทางที่ตั้งค่าไว้ได้ ตัวอย่างเช่น การแสดงผลรูปบนแอปพลิเคชัน แสดงข้อความบนจอแสดงผลบนถนัดมือ และใช้ประโยชน์จากเครือข่ายเพื่อส่งค่าขอ HTTP ไปยังอุปกรณ์อื่นเป็นต้น ดังนั้นถนัดมือจะไม่เพียงแต่สามารถตรวจจับท่าที่แสดงถึงตัวสะกดไทยเท่านั้น แต่จะสามารถตรวจจับท่าทางที่เป็นการเคลื่อนไหวได้หลากหลายแบบ หลากหลายท่าติดต่อกันตามที่ต้องการ และสามารถทำได้มากกว่าเพียงแค่แสดงผล เพื่อการใช้งานที่หลากหลายวัตถุประสงค์มากขึ้น

## 1.2 วัตถุประสงค์ของโครงการ

- 1) พัฒนาถนัดมือเพื่อการตรวจจับท่าทางมือรวมถึงระบบที่เกี่ยวข้อง ให้อยู่ภายในไมโครคอนโทรลเลอร์
- 2) ศึกษาการใช้งานไมโครคอนโทรลเลอร์และระบบปฏิบัติการ Linux ภายในไมโครคอนโทรลเลอร์
- 3) ศึกษาและพัฒนาแอปพลิเคชันและเซิร์ฟเวอร์ ที่มีส่วนที่ต้องติดต่อกับ Hardware เช่น การอ่านค่าจากเซนเซอร์
- 4) ศึกษาการประมวลผลข้อมูลของเซนเซอร์ผ่านกระบวนการทาง Machine Learning

### 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ถุงมือสามารถอำนวยความสะดวกให้กับผู้ใช้งานได้
- 2) ถุงมือสามารถให้ความบันเทิงกับผู้ใช้งานได้
- 3) ถุงมือสามารถช่วยเป็นสื่อกลางในการสื่อสาร โดยภาษามือได้บางส่วน
- 4) ผู้ใช้สามารถเข้าถึงและตั้งค่าการใช้งานถุงมือได้ง่าย
- 5) ผู้ใช้สามารถประยุกต์ใช้ถุงมือไปในวัตถุประสงค์ที่ต้องการได้

### 1.4 ขอบเขตของโครงการ

ถุงมือสามารถตรวจจับท่าทางในรูปแบบที่เซนเซอร์ 6 ตัวสามารถตรวจจับได้เท่านั้น ประกอบด้วยเซนเซอร์กลางหลังมือ 1 ตัว และบนนิ้วแต่ละนิ้ว 5 ตัว

การตรวจจับท่าทางของถุงมือต้องทำได้อย่างน้อย 40 ท่า รวมทั้งท่านิ่งและเคลื่อนไหว และในท่าที่สามารถตรวจจับได้นั้น จะเป็นท่าทางภาษามือที่เป็นตัวสะกดภาษาไทยอย่างน้อย 30 ตัว

สำหรับการกระทำหลังจากตรวจจับท่าทาง เบื้องต้นจะสามารถแสดงผลข้อความบนเว็บแอปพลิเคชันหรือจอแสดงผลได้ และส่งคำขอ HTTP ไปยังอุปกรณ์อื่นในเครือข่ายเดียวกันกับถุงมือได้

### 1.5 ข้อยกจำกัดของโครงการ

รูปแบบท่าทางของของมือผู้ใช้ที่ไม่เหมือนกันอาจจะส่งผลให้ความแม่นยำแตกต่างกัน เกิดจากค่าที่ได้จากเซนเซอร์ ซึ่งอาจจะมีค่าผิดพลาดหรือแตกต่างกันพอที่จะทำให้ตรวจจับได้ท่าที่ผิดพลาดได้ โดยเฉพาะเซนเซอร์ที่ใช้วัดความงอของนิ้วมือ ซึ่งวัดการงอได้เพียงแค่ทิศทางที่กำมือเท่านั้น สำหรับผู้ใช้ที่สามารถกางและงอนิ้วมือไปในทิศทางตรงข้ามได้มาก หรือท่าที่มีความใกล้เคียงกันมาก อาจจะทำให้เกิดการตรวจจับที่ผิดพลาดได้ แต่ท่าที่อยู่ในกรณีดังกล่าวนี้ ถือว่าเป็นส่วนน้อยของท่าทั้งหมดที่ถุงมือควรตรวจจับได้ และไม่ได้อยู่ในขอบเขตการพัฒนา

## บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

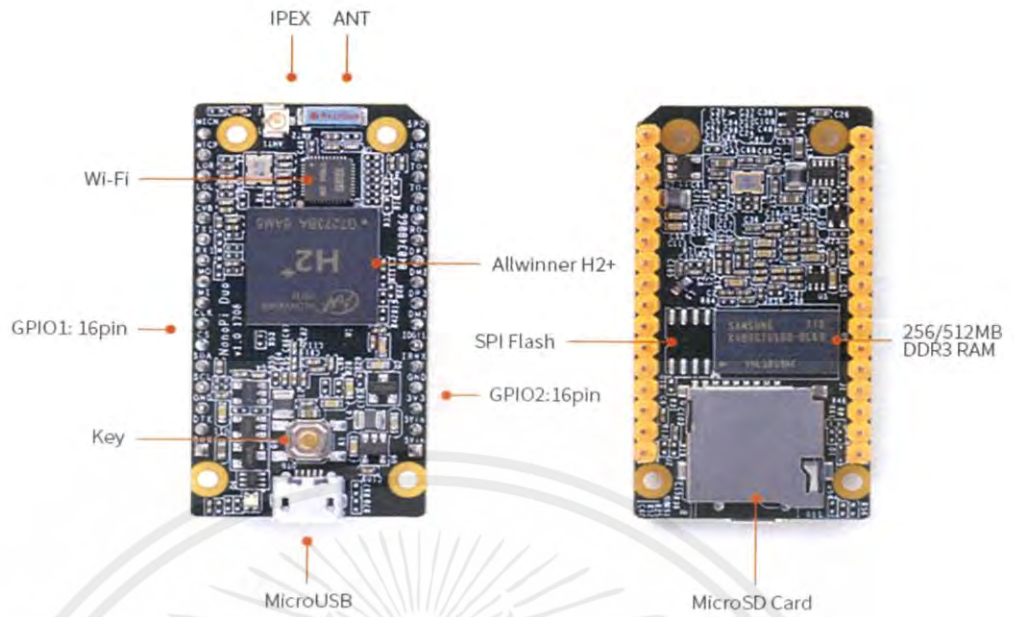
### 2.1 เทคโนโลยีด้านฮาร์ดแวร์

#### 2.1.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

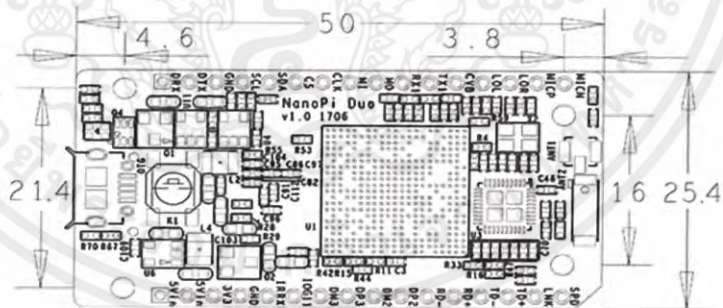
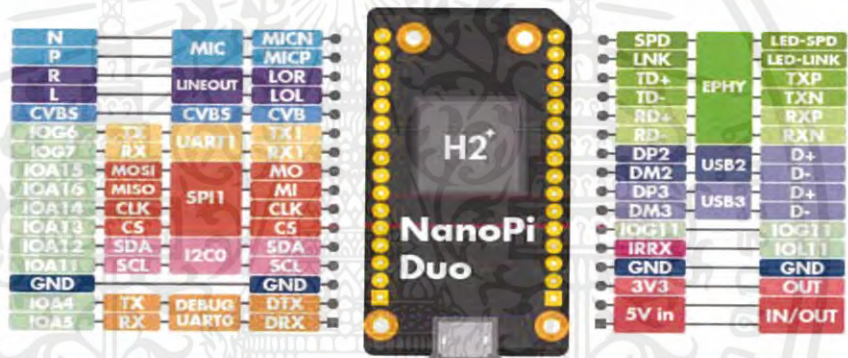
Nanopi-DUO หน่วยประมวลผลขนาดเล็กที่สามารถวางไว้บนข้อมือได้อย่างสะดวก มีคอร์ในการประมวลผลถึง 4 คอร์ จึงสามารถทำงานหลายงานพร้อมกันได้สะดวก เหมาะสมกับโครงการที่จำเป็นจะต้องประมวลผล เปิดเซิร์ฟเวอร์ และติดต่อกับฮาร์ดแวร์ในเวลาเดียวกัน อีกทั้งยังรันในระบบปฏิบัติการ Linux (Ubuntu) ที่ยอดนิยม สามารถศึกษาเพิ่มเติมได้ง่าย และสามารถติดตั้งแพ็คเกจที่จำเป็นต้องใช้ในการดำเนินโครงการได้

ตาราง 2.1 ข้อมูลจำเพาะ Nanopi-DUO [4]

CPU	Allwinner H2+, Quad-core Cortex-A7
RAM	DDR3: 256MB/512MB
การเชื่อมต่อเครือข่าย	10/100M Ethernet, Wifi XR819
อินเทอร์เฟซ USB	ขา 2.54 มม. pin x2, รวมอยู่ในขา 2.54 มม.
ช่องใส่การ์ด MicroSD	x1
MicroUSB	x2: สำหรับ OTG และสำหรับจ่ายไฟ
อินเทอร์เฟซสำหรับ Serial Debugging	รวมอยู่ในขา 2.54 มม.
อินเทอร์เฟซสำหรับเสียง	รวมอยู่ในขา 2.54 มม.
GPIO1	16 ขา 2.54 มม. ประกอบไปด้วย UART, SPI, I2C, Audio ฯลฯ
GPIO2	16 ขา 2.54 มม. ประกอบไปด้วย USB, 10/100M Ethernet, IO ฯลฯ
ขนาดของ PCB	25.4 x 50 มม.
Power Supply	DC 5V/2A
อุณหภูมิที่วัดได้	-40 ถึง 80 องศาเซลเซียส
ระบบปฏิบัติการ/ซอฟต์แวร์	U-boot, Linux Kernel 4.11.2 (mainline) , Ubuntu 16.04.2 LTS (Xenial)



NanoPi Duo pinout diagram

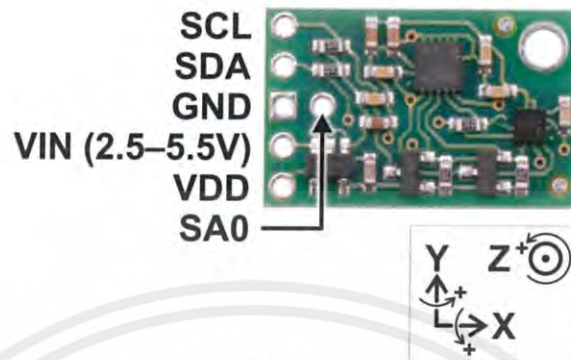


รูป 2.1 แผนภาพโครงสร้าง, ขาเชื่อมต่อของ NanoPI-DUO

2.1.2 เซนเซอร์ (Sensors)

เซนเซอร์ที่ใช้ตรวจจับการเคลื่อนไหวของมือจะประกอบไปด้วยเซนเซอร์วัดค่าความงอ (Flex Sensor) เพื่อวัดความงอของแต่ละนิ้วมือ และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อย (IMU: Inertial Measurement Unit) ซึ่งประกอบด้วยเซนเซอร์วัดความเร่งเชิงเส้น (Accelerometer)

และเซนเซอร์วัดความเร็วเชิงมุม (Gyroscope) เพื่อใช้ตรวจจับการเคลื่อนไหวและวัดมุมปัจจุบันของมือ



รูป 2.2 Polulu MinIMU9-v5

Polulu MinIMU9-v5 เป็นเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยตัวที่ใช้ในโครงการนี้ ซึ่งมีขนาดเล็ก วางบนมือและปลายนิ้วได้ สามารถส่งข้อมูลผ่านบัส I2C ได้ อีกทั้งยังมีเซนเซอร์สนามแม่เหล็ก (Magnetometer) ซึ่งเป็นเซนเซอร์ตรวจจับทิศทางรวมเข้ามาด้วย กลายเป็น 9 แกน ซึ่งสามารถนำไปประยุกต์ใช้ได้ในอนาคตหากจำเป็น

ตาราง 2.2 ข้อมูลจำเพาะเซนเซอร์ Polulu MinIMU9-v5 [5]

ขนาด:	0.8" × 0.5" × 0.1"
น้ำหนัก:	0.7 g
อินเทอร์เฟส:	I <sup>2</sup> C
ศักย์ไฟฟ้าต่ำสุดในการทำงาน (Minimum operating voltage):	2.5 V
ศักย์ไฟฟ้าสูงสุดในการทำงาน (Minimum operating voltage):	5.5 V
แกนที่วัด:	pitch (x), roll (y), yaw (z)
ช่วงการวัด:	±125, ±245, ±500, ±1000, or ±2000°/s ความเร็วเชิงมุม(gyro)
	±2, ±4, ±8, or ±16 g ความเร่งเชิงเส้น(accelerometer)
	±4, ±8, ±12, or ±16 gauss สนามแม่เหล็ก(magnetometer)
กระแสไฟที่จ่ายออก:	5 mA

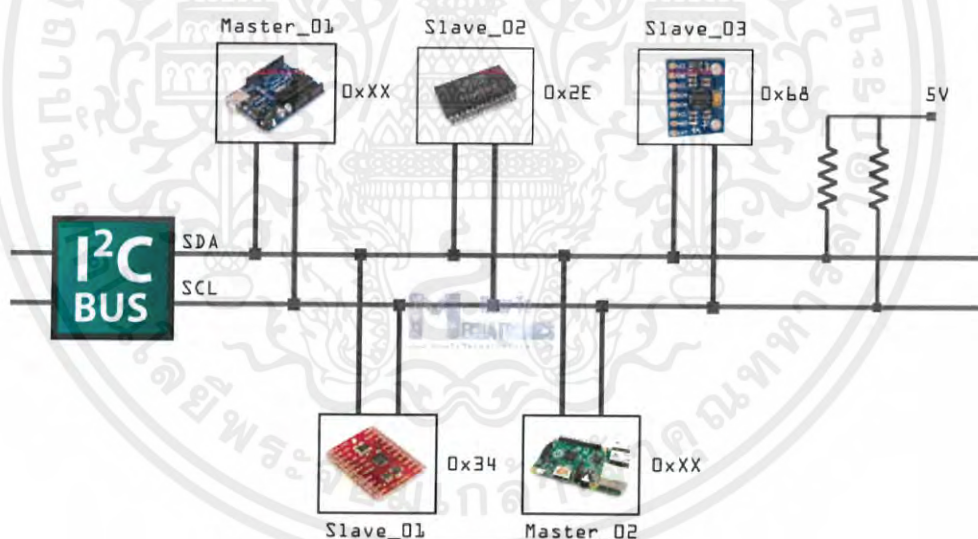
### 2.1.3 จอแสดงผล (Display)



รูป 2.3 SSD1306

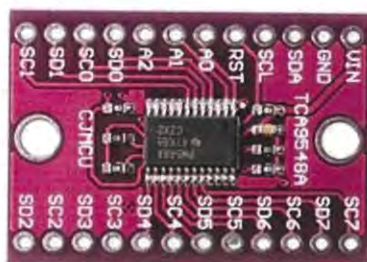
SSD1306 OLED Display เป็นจอแสดงผล ขนาด 128x64 ซึ่งมีขนาดพอดี สามารถวางไว้บนมือได้ นอกจากนั้นยังใช้การสื่อสารด้วย I2C เช่นเดียวกับเซนเซอร์อื่น ๆ

### 2.1.4 การติดต่อระหว่างฮาร์ดแวร์



รูป 2.4 ตัวอย่างเครือข่ายการเชื่อมต่อของบัส I<sup>2</sup>C

Inter Integrate Circuit (I2C) เป็นการสื่อสารอนุกรมแบบเข้าสัญญาณ ที่ทั้งสองฝั่งสามารถเป็นได้ทั้งผู้รับและผู้ส่ง เซนเซอร์จะต่อเข้ากับไมโครคอนโทรลเลอร์ผ่านบัส I2C แล้วไมโครคอนโทรลเลอร์สามารถเขียนและอ่านเซนเซอร์ได้จากการเข้าถึงเลขตำแหน่ง (Address) จำเพาะนั้น ๆ ของเซนเซอร์



รูป 2.5 TCA9548A

โมดูลขยายช่องสัญญาณ I<sup>2</sup>C 8 ช่อง (TCA9548A) เนื่องจากเซนเซอร์หน่วยวัดอาศัยหลักความถี่ที่ใช้ในการดำเนินโครงการอาจมีหลายตัว แต่ทุกตัวที่ใช้มีเลขตำแหน่งเดียวกันเพื่อแก้ไขปัญหาที่เลขตำแหน่งซ้ำ จึงต้องใช้โมดูลตัวนี้ โดยหลักการการทำงานเหมือน Multiplexer กล่าวคือสามารถเขียนช่องที่ต้องการไปยังตำแหน่งของ TCA9548A เพื่ออ่านค่าจากช่องนั้น ๆ ได้ สำหรับการอ่านข้อมูลจาก I<sup>2</sup>C จะใช้ไลบรารี SMBus ของ Python ที่เขียนขึ้นมาเพื่อรับส่งข้อมูลด้วย I<sup>2</sup>C โดยเฉพาะ

## 2.2 เทคโนโลยีด้านการประมวลผลทำทางมือ

การตรวจจับและประมวลผลทำทางของมือ จะใช้ภาษา Python เวอร์ชัน 3.6 เนื่องจากการที่เป็นภาษายอดนิยมในการพัฒนา Machine Learning ซึ่งส่งผลให้มีเฟรมเวิร์คหรือไลบรารีต่าง ๆ ที่สนับสนุนมากมาย

TensorFlow เป็นเฟรมเวิร์คที่ใช้ในการพัฒนาโมเดล Machine Learning เพราะเป็นหนึ่งในเฟรมเวิร์คที่นิยมมากในการพัฒนาโมเดล Machine Learning เนื่องจากมีประสิทธิภาพในเรื่องของความเร็วในการคำนวณอย่างเหมาะสมเมื่อเทียบกับความยากในการเขียน และ TensorFlow ถูกพัฒนาโดยบริษัท Google ทำให้มีการอัปเดตเรื่องของประสิทธิภาพของเฟรมเวิร์คนี้อย่างสม่ำเสมอ

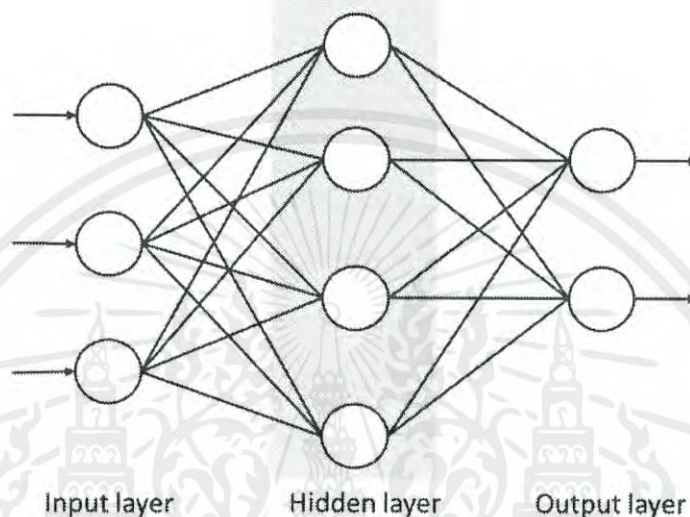


รูป 2.6 สัญลักษณ์ TensorFlow

โมเดลสำหรับการพัฒนา Machine Learning เพื่อการทำนายการเคลื่อนไหวของมือ มีทั้งหมด 2 โมเดล ได้แก่

## 2.2.1 โครงข่ายประสาทเทียม (Artificial Neural Networks - ANN)

เป็น โมเดลทางคณิตศาสตร์ที่จำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ โดยมีวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เช่นเดียวกับความสามารถที่มีในสมองมนุษย์



รูป 2.7 สถาปัตยกรรมโครงข่ายประสาทเทียม

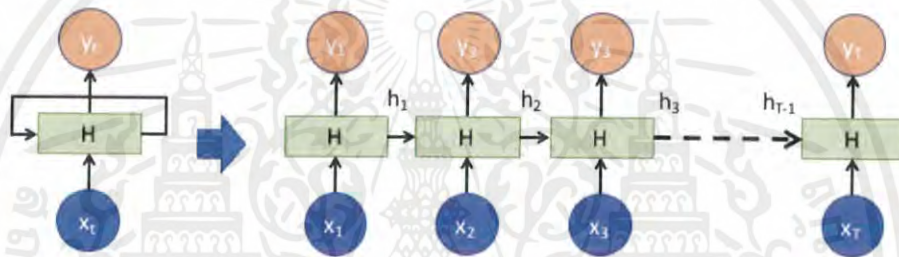
จากรูปที่ 2.7 คือโครงสร้างหลักๆของโครงข่ายประสาทเทียม จะมีสามส่วน ได้แก่ ชั้นขาเข้า (Input layer) ชั้นซ่อน (Hidden layer) และชั้นขาออก (Output layer)

ชั้นขาเข้าจะรับข้อมูลเข้ามาซึ่งอาจมีหลายข้อมูลได้ และชั้นซ่อนจะเอาข้อมูลนั้นมาคูณกับค่าน้ำหนักของแต่ละขาและนำผลลัพธ์ที่ได้จากทุก ๆ ข้อมูลมารวมกันแล้วเอามาเทียบกับค่า Threshold ที่กำหนดไว้ ถ้าผลลัพธ์เกินค่า Threshold ก็จะทำการส่งผลลัพธ์ออกไป แต่ถ้าค่าน้อยกว่า Threshold ก็จะไม่เกิดผลลัพธ์ ซึ่งค่าน้ำหนักจะเป็นค่าที่ไม่แน่นอน แต่สามารถให้คอมพิวเตอร์ปรับค่าได้เองโดยการสอนให้คอมพิวเตอร์รู้จัก Pattern ของสิ่งที่ต้องการให้คอมพิวเตอร์รู้ เรียกว่า "Back propagation" ซึ่งจะมีการใช้อัลกอริทึม Back-propagation เพื่อใช้ในการปรับปรุงน้ำหนัก หลังจากได้รูปแบบข้อมูลสำหรับฝึกให้ในแต่ละครั้งแล้ว ค่าเอาต์พุตที่ได้รับจากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลลัพธ์ที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนักต่อไป เมื่อปรับค่าน้ำหนักแล้วจะทำให้มีความแม่นยำมากขึ้น

## 2.2.2 โครงข่ายประสาทเทียมแบบมีหน่วยความจำระยะสั้นและยาว (Long Short-Term Memory)

Long Short-Term Memory (LSTM) เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลแบบมีลำดับ (Sequence) และเป็นรูปแบบหนึ่งของ Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่เหมาะสมกับการใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) ซึ่งก็คือการเอาผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลขาเข้าอีกครั้ง ดังรูปที่ 13 ซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง เช่น Time series ข้อมูลเสียง ข้อความ เป็นต้น ดังนั้น 2 ส่วนที่สำคัญของ RNN ก็คือ Hidden state ก่อนหน้า และข้อมูลอินพุต ณ ตอนนั้น



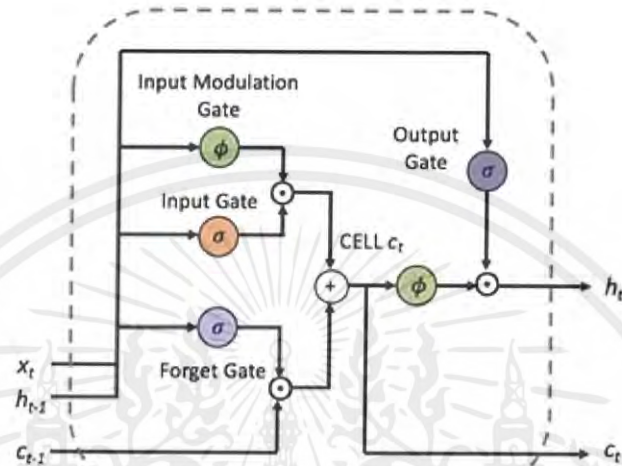
รูป 2.8 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ RNN

- H = Hidden layer
- $y_t$  = เอาต์พุตจาก RNN ที่เวลา t
- $x_t$  = ข้อมูลอินพุตที่เวลา
- $h_t$  = Hidden state ที่เวลา t

ในแต่ละ Node ของ RNN จะมีข้อมูลขาเข้าสองอย่างอันได้แก่อินพุต ที่ Node นั้น ๆ และผลลัพธ์ที่ได้จากการคำนวณใน Node ก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทางคือ ผลลัพธ์ที่ออก ที่ Node นั้น ๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าใน node ถัดไป

ข้อดีของ RNN คือ มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคต นั้นหมายถึง อะไรที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย แต่แม้ RNN จะมีข้อดีในการทำงานของข้อมูลที่มีความต่อเนื่อง หนึ่งในข้อเสียของ RNN คือสามารถดูย้อนกลับได้แค่เพียงในช่วงระยะเวลาสั้น ๆ เท่านั้น

Long Short-Term Memory ได้มีการเปลี่ยนตัวฟังก์ชันด้านในให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหน่วยความจำเข้ามาเกี่ยวข้องที่ไว้คอยจัดการจำสิ่งที่เกิดขึ้นและใช้ในการตัดสินใจในเวลาถัดไป โดยในหน่วยความจำนี้จะต้องถูกปรับไปเรื่อย ๆ การที่มีหน่วยความจำเข้ามาเกี่ยวข้องนั้นทำให้แก้ปัญหของ RNN ได้คือสามารถดูข้อมูลย้อนกลับได้ระยะที่ยาวมากขึ้น



รูป 2.9 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ LSTM

### 2.2.2.1 การทำงานของโครงข่ายประสาทเทียมแบบ LSTM

- 1) Cell state เป็นตัวเก็บ State ของ Memory cell ใน LSTM
- 2) Gate เป็นตัวที่ควบคุมการไหลของข้อมูล ซึ่งก็คือ ค่า Analog ที่คอยควบคุมว่าเมื่อไรควรจะ Read, Write หรือ Forget ซึ่งก็เหมือนกับประตูที่จะดูว่า เมื่อไรควรเปิดให้ข้อมูลไหลเข้า ไหลออก หรือไหลหายไป (Forget)
  - 2.1) Forget gate คือการลบ Cell state ออกไปและเตรียมพื้นที่รับข้อมูลใหม่ โดยมี Forget gate เป็นตัวตัดสินใจ ถ้า Forget gate ให้ค่าเป็น 0 ก็ลบ Cell state ออกไป แต่ถ้า Forget gate ให้ค่าเป็น 1 ก็ยังเก็บ Cell state เดิมต่อไป ซึ่งข้อมูลที่จำเป็นต้องใช้ในการตัดสินใจนั้นมาจาก ข้อมูลอินพุตที่เข้ามาใหม่ ประกอบกับ Hidden state ก่อนหน้า โดยจะใช้ Sigmoid function เป็นตัวตัดสินใจ
  - 2.2) Input gate คือตัวที่ตัดสินใจว่าจะอนุญาตให้อัพเดท Cell state หรือไม่ เมื่อมีข้อมูลอินพุตใหม่เข้ามา การคำนวณนี้ใช้ค่าข้อมูลอินพุตที่เข้ามา กับ Hidden state ก่อนหน้านั้น

- 2.3) Input modulation gate คือตัวที่ตัดสินใจว่าถ้ามีการอัปเดต Cell state จะอัปเดตด้วยค่าใด
- 2.4) การอัปเดต cell state คือการนำข้อมูลจาก Forget Gate , Input Gate และ Input modulation Gate มารวมเข้าด้วยกัน ดังนี้

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

กำหนดให้

$c_t$  คือค่าของ Cell state

$f_t$  คือค่าของ Forget gate

$c_{t-1}$  คือค่าของ Cell state ณ เวลาก่อนหน้า

$i_t$  คือค่า Input gate

$g_t$  คือค่าที่จะอัปเดต

เริ่มจากส่วนแรกของสมการ ถ้า  $f_t$  เป็น 0 ก็จะไม่เอา  $c_{t-1}$  มาพิจารณาในการอัปเดต Cell state ถ้า  $f_t$  เป็น 1 จะยังคงค่า  $c_{t-1}$  เอาไว้ ประกอบการพิจารณาการอัปเดต ส่วนหลังของสมการ ส่วนนี้จะเป็นส่วนของการอัปเดต Cell state จากข้อมูลใหม่ ซึ่งตอนนี้มีค่าที่จะอัปเดตจาก  $g_t$  แล้ว และจะใช้  $i_t$  มาเป็นตัวตัดสินใจในการอัปเดต ถ้าเป็น 1 ก็อัปเดตได้เลย แต่ถ้าเป็น 0 ก็จะไม่มีการอัปเดต จากค่าทั้งหมดทำให้ได้ค่า  $c_t$  ตัวใหม่

- 2.5) Output gate คือสิ่งที่ต้องผลิตออกไป ก็คือ Hidden state ณ เวลาที่  $t$  หรือ  $h_t$  ซึ่งเมื่อตอนที่เวลา  $t+1$  ตัว LSTM จะเอาค่า  $h_t$  นี้ไปคำนวณด้วย ซึ่งการส่งค่า  $h_t$  ออกไปนั้นจะมีตัว Output Gate เป็นตัวตัดสินใจ และจะนำค่าเอาต์พุตไปเป็นค่า  $h_t$  (อินพุต) สำหรับ sequence ถัดไป

จะเห็นได้ว่า LSTM จะเหมาะกับข้อมูลที่เข้ามาเป็นลำดับและสามารถนำข้อมูลเก่ามาคำนวณซึ่งอาจจะมีผลต่อในอนาคตได้ โมเดลนี้จึงเหมาะที่จะนำมาทำนายคุณภาพอากาศในอนาคต เนื่องจากข้อมูลที่เข้ามาเป็นข้อมูลที่เรียงลำดับและมีความต่อเนื่อง [6]

## 2.3 เทคโนโลยีด้านฐานข้อมูล

### 2.3.1 MongoDB



รูป 2.10 สัญลักษณ์ MongoDB

เป็นระบบจัดการฐานข้อมูลแบบไม่เชิงสัมพันธ์ (Non-relational Database) ที่ยืดหยุ่น จุดเด่นคือสามารถเก็บข้อมูลเป็นประเภทวัตถุที่ไม่จำเป็นต้องมีขนาดคงที่ได้ ซึ่งมีประโยชน์อย่างมากในการดำเนินโครงการ เนื่องจากมีข้อมูลบางชนิดที่ไม่ทราบขนาด การใช้ฐานข้อมูลแบบนี้และเก็บข้อมูลเป็นวัตถุแทนจะทำให้ดำเนินการเก็บและเรียกค้นได้ง่ายขึ้น

## 2.4 เทคโนโลยีด้านแอปพลิเคชัน

ในการพัฒนาแอปพลิเคชันโต้ตอบกับผู้ใช้ ผู้พัฒนาเลือกที่จะทำเป็นเว็บแอปพลิเคชัน เนื่องจากเมื่อผู้ใช้เชื่อมต่อกับมือถือ ผู้ใช้ก็จะสามารถเข้าถึงได้ทันทีผ่านอุปกรณ์ทั่วไป เช่น คอมพิวเตอร์, โทรศัพท์ และแท็บเล็ต โดยไม่ต้องดาวน์โหลดแอปพลิเคชันใด ๆ ล่วงหน้า

### 2.4.1 Vue.JS



รูป 2.11 สัญลักษณ์ Vue.JS

เป็นเฟรมเวิร์กที่ใช้ในการพัฒนาเว็บแอปพลิเคชันฝั่งผู้ใช้ เขียนโดยใช้ภาษา JavaScript โดยเป็นเฟรมเวิร์กสามารถพัฒนาเว็บแอปพลิเคชันที่สามารถเปลี่ยนแปลงข้อมูลได้แบบเรียลไทม์ จึงเหมาะสมสำหรับการพัฒนาที่ต้องมีการป้อนและอัปเดตข้อมูลจากเซิร์ฟเวอร์ตลอดเวลา

## 2.4.2 Flask

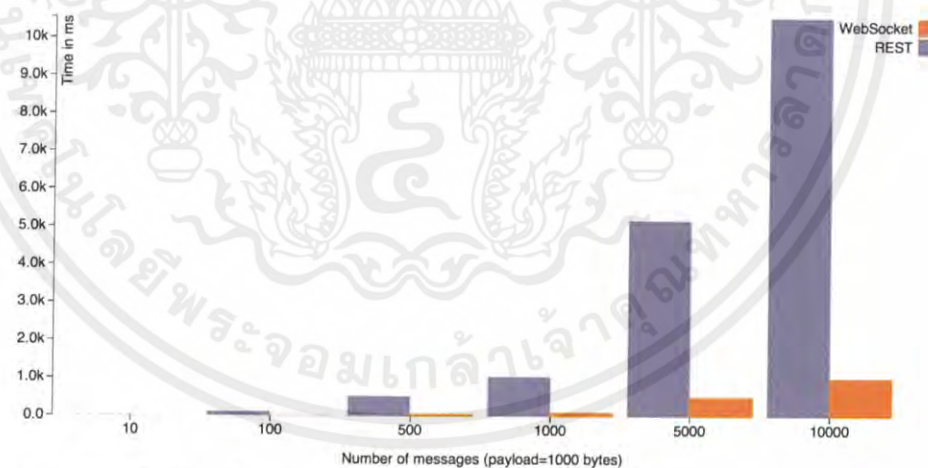


รูป 2.12 สัญลักษณ์ Flask

Flask เป็นเฟรมเวิร์คขนาดเล็กที่ใช้ในการพัฒนาเว็บแอปพลิเคชันฝั่งเซิร์ฟเวอร์ เขียนโดยใช้ภาษา Python สามารถติดต่อฐานข้อมูล MongoDB ได้ง่าย และด้วยเนื่องจากการประมวลผลข้อมูลทำโดยใช้ภาษา Python จึงควรใช้เฟรมเวิร์คสำหรับพัฒนาเซิร์ฟเวอร์เป็นภาษาเดียวกันด้วย ถึงจะสามารถทำงานร่วมกันได้

## 2.4.3 Socket.IO

Socket.IO เป็นไลบรารีที่ทำให้เว็บแอปพลิเคชันฝั่งผู้ใช้สามารถเชื่อมต่อกับเซิร์ฟเวอร์แบบเรียลไทม์ได้ ผ่านเทคโนโลยี WebSocket เมื่อมีการเชื่อมต่อตั้งแต่เริ่มเข้าเว็บแอปพลิเคชัน ทั้งสองฝั่งสามารถส่งข้อมูลหากันได้ตลอดเวลา และใช้เวลาน้อย เพราะไม่จำเป็นจะต้องเปิดการเชื่อมต่อทุกครั้งที่ส่งข้อมูลดังเช่นการส่งข้อมูลผ่าน HTTP อย่าง REST ที่นิยมทั่วไป [7]



รูป 2.13 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูลระหว่าง REST กับ WebSocket

## 2.4.4 Eventlet

เป็นไลบรารีสำหรับช่วยในการทำงานแบบ Multithread ช่วยให้สามารถทำการเปิดเซิร์ฟเวอร์โดยใช้ Flask และรอข้อมูลจาก Socket.IO พร้อมกับการประมวลผลข้อมูลด้วย Machine Learning ไปพร้อม ๆ กันได้

## 2.5 ทฤษฎีเกี่ยวกับการคำนวณ

### 2.5.1 การคำนวณมุมโดยใช้เซนเซอร์วัดความเร่งเชิงเส้น

เซนเซอร์วัดความเร่งเชิงเส้น สามารถนำไปใช้คำนวณมุมได้สองแกน ดังนี้ [8]

$$A_x = \tan^{-1}\left(\frac{X}{\sqrt{Y^2 + Z^2}}\right)$$

$$A_y = \tan^{-1}\left(\frac{Y}{\sqrt{X^2 + Z^2}}\right)$$

กำหนดให้

$A_x$  คือมุมแกน x ที่คำนวณได้

$A_y$  คือมุมจากแกน y ที่คำนวณได้

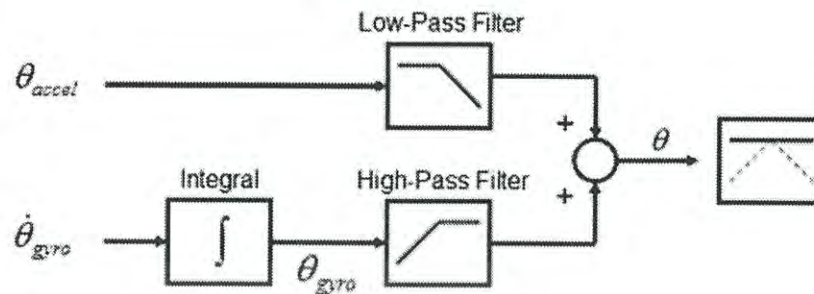
$X$  คือค่าความเร่งแกน x

$Y$  คือค่าความเร่งแกน y

$Z$  คือค่าความเร่งแกน z

แต่เนื่องจากเซนเซอร์วัดความเร่งเชิงเส้นไม่สามารถแยกแยะความเร่งพื้นฐานของวัตถุกับความเร่งที่ภายนอกกระทำกับวัตถุได้ จึงอาจจะทำให้เกิดมุมที่ผิดพลาดได้หากมีแรงภายนอกกระทำ หรือมีการขยับที่เร็วเกิดขึ้น

### 2.5.2 การคำนวณหามุมโดยใช้ตัวกรองร่วมสองแบบ (Complementary filter)



รูป 2.14 ระบบของตัวกรองร่วมสองแบบ

เป็นขั้นตอนการคำนวณหามุมของวัตถุโดยการใช้ค่าที่ได้จากเซนเซอร์วัดความเร่งเชิงเส้นกับเซนเซอร์วัดความเร็วเชิงมุมร่วมกันเพื่อสร้างเป็นตัวกรอง

ค่าที่ได้จากเซนเซอร์วัดความเร่งเชิงเส้นจะถูกนำไปคำนวณกลายเป็นมุมโดยใช้สูตรการคำนวณตามข้อที่ 2.5.1 เรียกว่าตัวกรองช่วงล่าง (Low-pass filter)

ค่าที่ได้จากเซนเซอร์วัดความเร็วเชิงมุมจะถูกนำไปคำนวณมุมที่กวาดไปได้นับตั้งแต่การคำนวณครั้งก่อนหน้า เรียกว่าตัวกรองช่วงบน (High-pass filter)

จากนั้นจึงนำมาคำนวณตามแต่ละแกนโดยใช้สูตรดังกล่าว [9]

$$Angle_{axis} = \alpha * (Angle_{acc}) + (1 - \alpha) * (Angle_{gyr})$$

กำหนดให้

$Angle_{axis}$  คือมุมผลลัพธ์ ที่ได้จากการคำนวณของมุนั้น ๆ

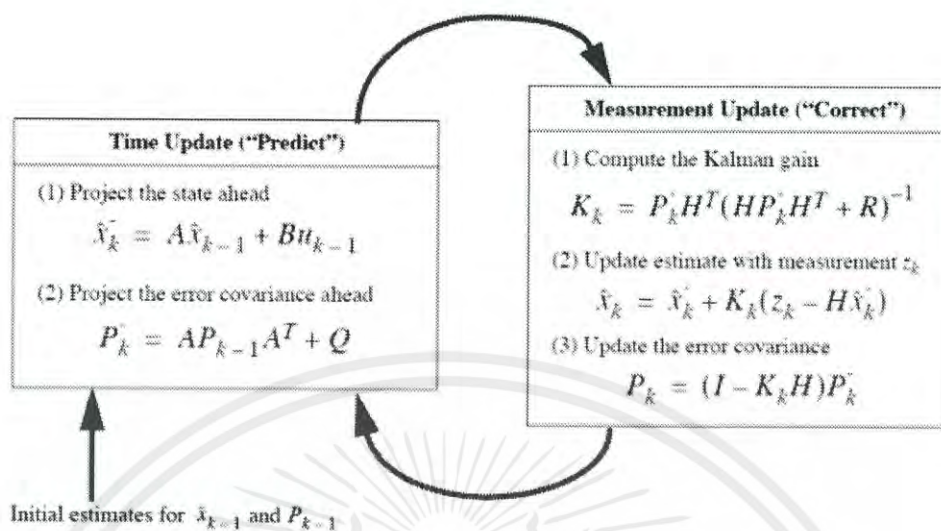
$\alpha$  คือค่าน้ำหนักของ Low-Pass Filter มีค่าระหว่าง 0.0 ถึง 1.0

$Angle_{acc}$  คือค่ามุมที่ได้จากเซนเซอร์วัดความเร่งเชิงเส้น

$Angle_{gyr}$  คือค่ามุมที่กวาดได้จากเซนเซอร์วัดความเร็วเชิงมุม

ข้อดีที่เห็นได้ชัดของขั้นตอนการคำนวณแบบนี้ คือสามารถคำนวณได้เร็ว ใช้ทรัพยากรน้อย ให้ค่าที่แม่นยำ แต่ข้อเสียคือไม่สามารถกำจัดค่ารบกวนได้ทั้งหมด

### 2.5.3 การคำนวณหาผลโดยใช้ตัวกรองคาลมาน (Kalman filter)



รูป 2.15 ระบบของตัวกรองคาลมาน

ความหมายของตัวแปร โดยสังเขป

- $x$  = ค่าสถานะที่คาดเดา (Estimate State)
- $A$  = เมตริกซ์ที่เกี่ยวกับการเปลี่ยนแปลงสถานะ (State transition matrix)
- $B$  = เมตริกซ์ที่เกี่ยวกับตัวแปรควบคุม (Control matrix)
- $u$  = ค่าของตัวแปรควบคุม (Control Signal)
- $P$  = ความแปรปรวนของข้อมูลที่คาดเดา (Error covariance)
- $Q$  = ค่าความผิดพลาดในขั้นตอนการคำนวณ (Estimated error in process)
- $K$  = Kalman gain
- $H$  = เมตริกซ์ที่เกี่ยวกับจำนวนตัวแปรในระบบ (Observation matrix)
- $R$  = ค่าความผิดพลาดของค่าสถานะจริง (Estimated error in measurements)
- $z$  = ค่าสถานะจริงจากการวัด (อินพุตจากเซนเซอร์)

Kalman filter จะมี 5 ขั้นตอนคือ

- 1) การคาดเดาสถานะถัดไปค่าจากตัวแปรควบคุม โดยการคำนวณความเร็วและระยะทางจากเซนเซอร์วัดความเร็วเชิงเส้นและเซนเซอร์ความเร็วเชิงมุมโดยใช้สูตรคำนวณหาความเร็วจากความเร็วและระยะทางจากความเร็ว
- 2) การคำนวณค่าความผิดพลาดของความแปรปรวนของข้อมูล (Error covariance) ที่จะใช้ในขั้นตอนต่อไป ซึ่งจะส่งผลต่อค่า  $K$  (Kalman Gain)
- 3) การคำนวณค่า Kalman Gain จากค่า Error covariance

- 4) การปรับค่าหลังจากได้ค่าที่แท้จริง หลังจากการการคาดเดาในขั้นตอนที่ 1 แล้ว  
ในขั้นตอนนี้คือการรับค่าอินพุตจากด้านนอกเข้ามาคำนวณด้วยเพื่อความแม่นยำ  
มากขึ้น โดยอินพุตดังกล่าวนั้นคือค่าสถานะที่วัดได้จริง
- 5) การปรับค่า Error covariance เพื่อใช้ในการคำนวณในรอบต่อไป (การ  
ประมวลผลอย่างต่อเนื่อง) [10]

## 2.6 งานวิจัยที่เกี่ยวข้อง

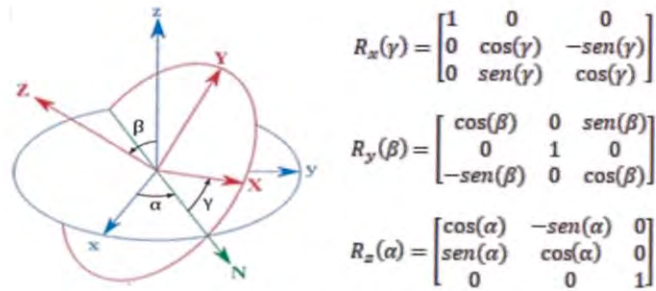
### 2.6.1 A Gesture Detection Glove For Human-computer Interaction [1]

งานวิจัยนี้มีเป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับท่าทางต่าง ๆ ในการควบคุม  
เมาส์และคีย์บอร์ดผ่านการเชื่อมต่อแบบ Bluetooth โดยใช้ MCU เป็น PAMPIUM ที่ประกอบจาก  
FPGA พัฒนาโดยมหาวิทยาลัย Pampa และใช้เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อย ที่กลางมือ  
และปลายนิ้วทั้ง 5 รวม 6 ตัว



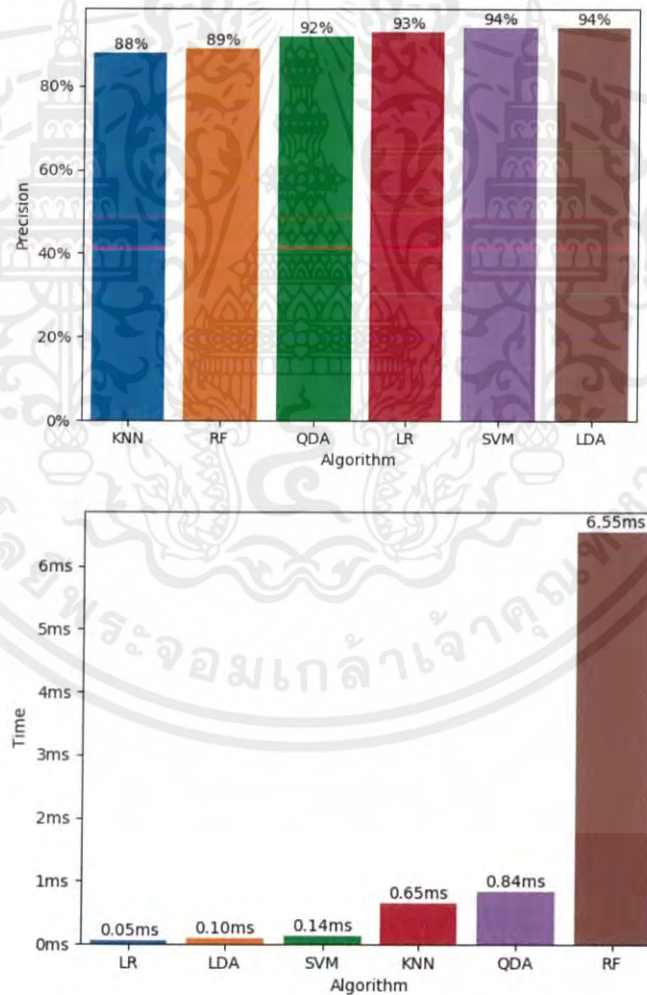
รูป 2.16 ถุงมือของงานวิจัย A Gesture Detection Glove For Human-computer Interaction

ค่าที่ได้จาก เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยเชิงเส้นจะถูกปรับให้อยู่ในค่า  
ระหว่าง -10 ถึง +10 และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยเชิงมุมจะถูกปรับให้อยู่ในค่า  
ระหว่าง  $-\pi$  ถึง  $+\pi$  จากนั้นจะถูกนำมาแปลงเป็นเมตริกซ์การหมุน (Rotation Matrix)



รูป 2.17 Rotation Matrix

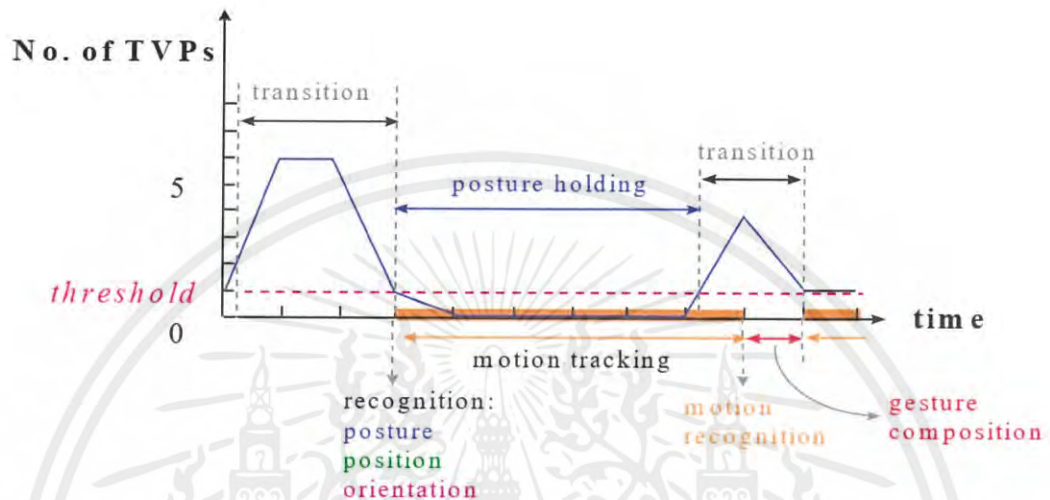
การตรวจจับท่าทางมือจะเป็นการพัฒนาและใช้โมเดล Machine Learning หลากหลายแบบ ได้แก่ KNN, RF, QDA, LR, SVM และ LDA เพื่อนำมาเปรียบเทียบความแม่นยำและเวลาที่ใช้



รูป 2.18 แผนภูมิเปรียบเทียบความแม่นยำและเวลาที่ใช้ระหว่างโมเดล Machine Learning แต่ละตัวของงานวิจัย A Gesture Detection Glove For Human-computer Interaction

### 2.6.2 A real-time continuous gesture recognition system for sign language [2]

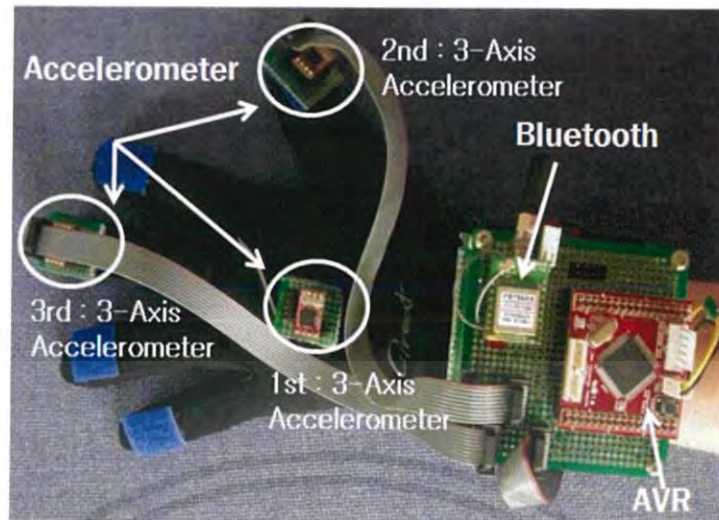
งานวิจัยนี้มีเป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับท่าทางของมือ 40 ท่า เพื่อแปลเป็นศัพท์ตามภาษามืออเมริกัน โดยการหาคุณลักษณะจากชุดข้อมูลเซนเซอร์ตามเวลา นำเข้าไปฝึกในโมเดลโครงข่ายประสาทเทียมแบบ Markov Model



รูป 2.19 แผนภูมิเวลาในการวิเคราะห์ท่าของงานวิจัย A real-time continuous gesture recognition system for sign language

### 2.6.3 3-D hand motion tracking and gesture recognition using a data glove [3]

งานวิจัยนี้มีเป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับการเคลื่อนไหวของมือและท่าทางต่าง ๆ โดยใช้ MCU เป็น ATmega128 (AVR) ใช้เซนเซอร์เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่กลางมือ, ปลายนิ้วโป้ง และปลายนิ้วกลาง รวม 3 ตัว และติดต่อกับคอมพิวเตอร์ผ่านสัญญาณ Bluetooth



รูป 2.20 ถุงมือจากงานวิจัย 3-D hand motion tracking and gesture recognition using a data glove

การตรวจจับการเคลื่อนไหว จะใช้การเทียบค่าจากเซนเซอร์กับมุมต่าง ๆ ของมือ ส่วนการตรวจจับท่าทาง จะใช้การตรวจสอบเงื่อนไขว่าค่าตรงกับเงื่อนไขของท่าใด ก็จะตรวจจับได้ตรงกับท่านั้น ๆ การทดลองของงานวิจัยพบว่าหลังจากการทดสอบท่าละ 50 ครั้ง สามารถแยกท่าเป่ายังจวบ 3 ท่า ได้แก่ ค้อน กระดาก กรรไกร ออกจากกันได้ร้อยละ 100

## บทที่ 3

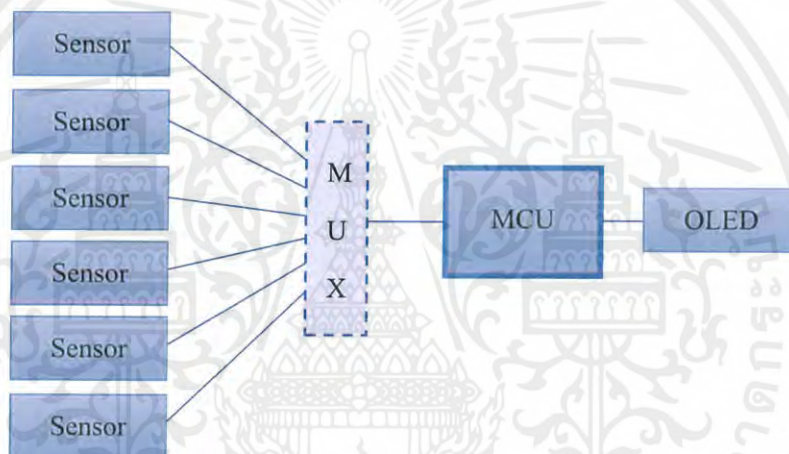
### การออกแบบและพัฒนา

#### 3.1 การออกแบบระบบโดยรวม

เพื่อให้เห็นได้ชัดเจน ระบบสามารถแบ่งออกได้เป็น 3 ส่วน คือฮาร์ดแวร์ เซิร์ฟเวอร์ และเว็บแอปพลิเคชัน

##### 3.1.1 ฮาร์ดแวร์ (Hardware)

คือส่วนของถู่มือทั้งหมด รวมถึงเซนเซอร์และ MCU มีหน้าที่เก็บตัวอย่างข้อมูล (Sample) จากการกระทำทางต่าง ๆ ของผู้ใช้ และแสดงผลข้อมูลต่าง ๆ ตามที่ได้คำสั่งจากเซิร์ฟเวอร์



รูป 3.1 การเชื่อมต่อระหว่างฮาร์ดแวร์

##### 3.1.2 เซิร์ฟเวอร์ (Server)

คือส่วนซอฟต์แวร์ที่รันบน MCU และมีหน้าที่ดังต่อไปนี้

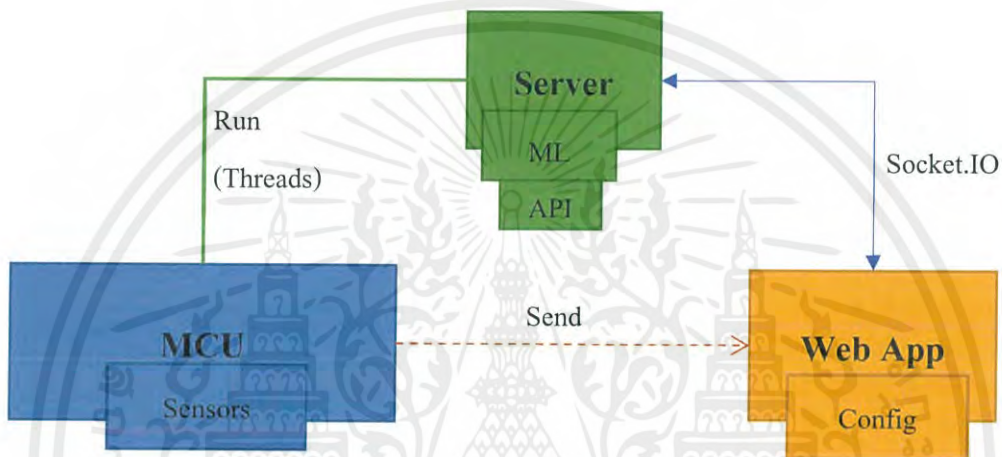
3.1.1.1 มีส่วนที่วนทำงานตลอดเวลา เพื่อควบคุมการทำงานทั้งหมดของถู่มือ ได้แก่ อ่านข้อมูลจากเซนเซอร์ แปลงชุดข้อมูลให้พร้อมสำหรับการตรวจจับท่าทาง ตรวจสอบการจับท่า นำชุดข้อมูลไปตรวจใน โมเดล Machine Learning รอรับคำตอบ และเริ่มการกระทำของท่านั้น ๆ

3.1.1.2 เป็นศูนย์กลางสำหรับให้เว็บแอปพลิเคชันติดต่อ คอยรับและส่งข้อมูลเช่น การตั้งค่า และผลการตรวจจับ ผ่าน Socket.IO

### 3.1.3 เว็บแอปพลิเคชัน (Web Application)

คือส่วนที่ทำงานก็ต่อเมื่อผู้ใช้ส่งคำขอ HTTP (HTTP Request) ให้แก่เซิร์ฟเวอร์ แล้วเซิร์ฟเวอร์ก็จะส่งคำตอบรับ (HTTP Response) เป็นเว็บแอปพลิเคชัน และทำงานอยู่บนบราวเซอร์ของผู้ใช้เอง ส่วนนี้มีหน้าที่เบื้องต้นดังนี้

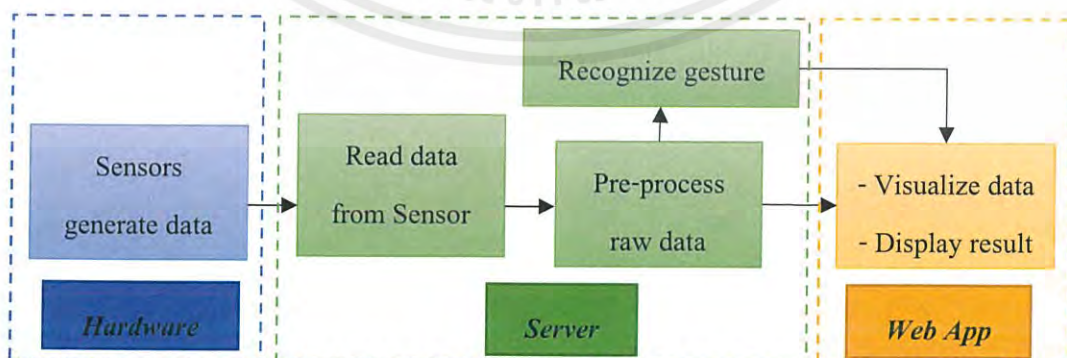
1. เชื่อมต่อ รับ และส่งข้อมูลผ่าน Socket.IO จากเซิร์ฟเวอร์
2. แสดงผลข้อมูล/ผลการตรวจจับท่าทางที่ส่งมาจากเซิร์ฟเวอร์
3. เป็นส่วนที่ติดต่อกับผู้ใช้ (UI) สำหรับการตั้งค่าต่าง ๆ ที่เกี่ยวกับระบบโดยรวม



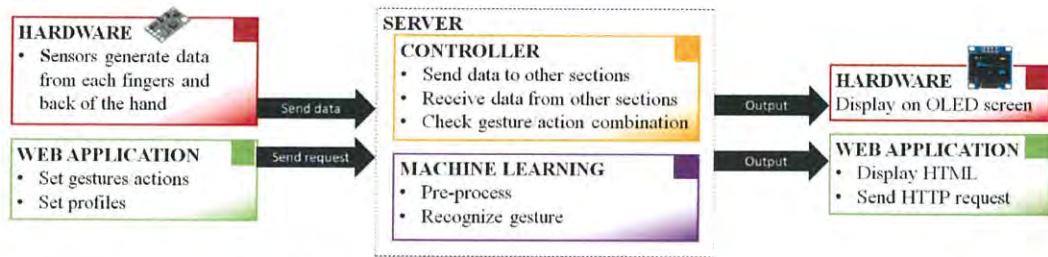
รูป 3.2 โครงสร้างระบบ

การตรวจจับท่าทางจะเริ่มจากส่วนของฮาร์ดแวร์ ที่เซนเซอร์ทุกตัวจะสร้างข้อมูลขึ้นมา จากนั้นเซิร์ฟเวอร์ก็มีหน้าที่อ่านข้อมูลเหล่านั้น นำมาประมวลผล (Pre-process) แล้วจึงส่งต่อให้โมเดล Machine Learning ทำการตรวจจับท่าทาง (Recognize Data)

ในส่วนของเว็บแอปพลิเคชันนั้นจะคอยแสดงข้อมูลจากเซนเซอร์ และผลการตรวจจับว่าเป็นท่าใด



รูป 3.3 ขั้นตอนการทำงานแบ่งตามส่วนของการตรวจจับท่าทาง



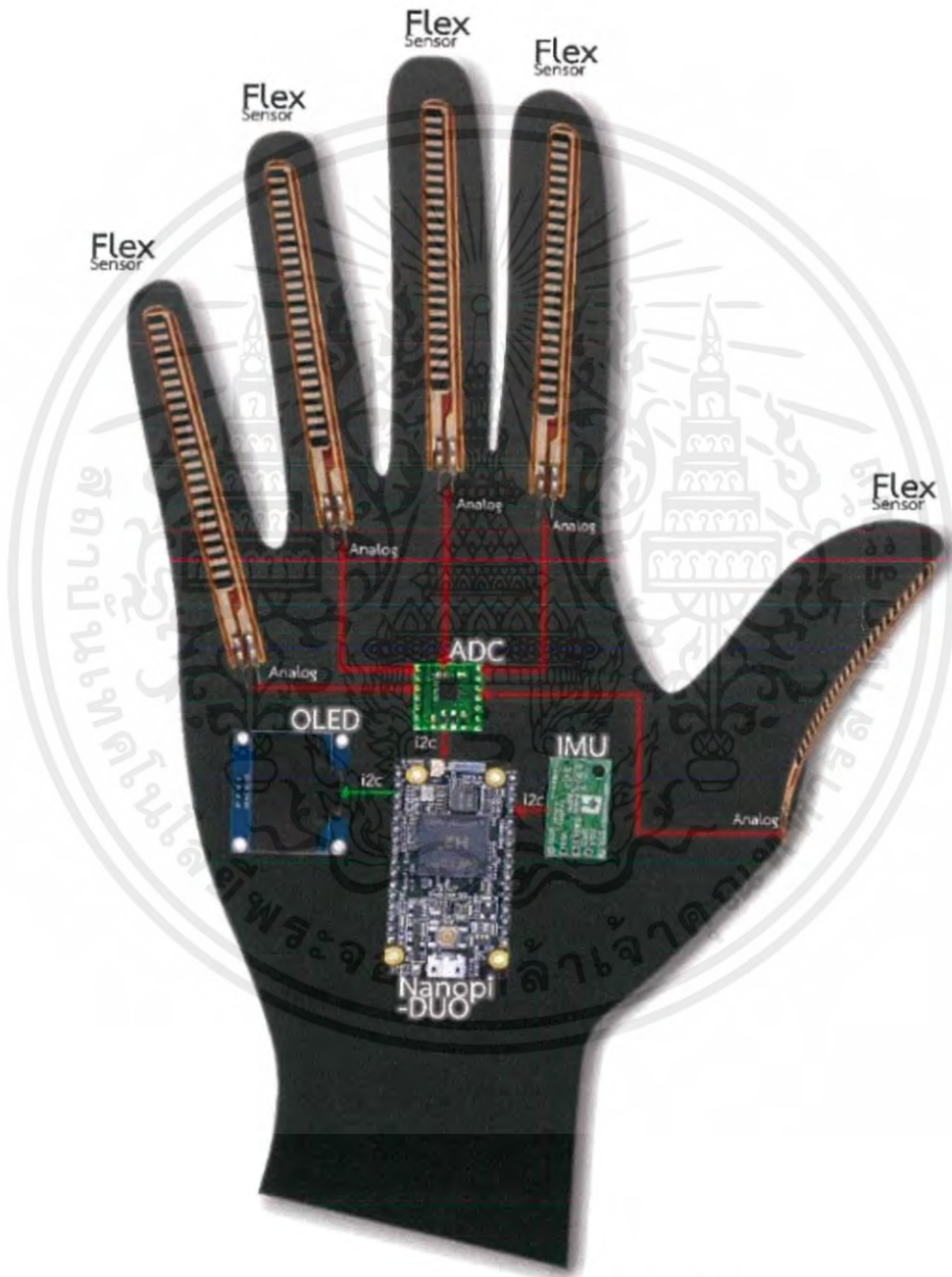
รูป 3.4 หน้าที่การทำงานของแต่ละส่วนของระบบ

### 3.2 การออกแบบถุงมือ

สำหรับถุงมือ ผู้พัฒนาได้ออกแบบไว้ 2 แบบ คือแบบที่ใช้เซนเซอร์วัดความงอสำหรับนิ้ว และเซนเซอร์หน่วยวัดอาศัยหลักความถี่สำหรับวัดมุมของมือ และอีกแบบคือแบบที่ใช้เซนเซอร์หน่วยวัดอาศัยหลักความถี่วัดทั้งความงอและมุมของมือ ถุงมือทั้ง 2 แบบ จะมีรูปแบบข้อมูลที่ได้รับแตกต่างกัน และส่งผลให้การประมวลผลในโมเดล Machine Learning ต่างกันด้วย

### 3.2.1 ถุงมือแบบที่ 1:

เป้าหมายของถุงมือรูปแบบนี้ คือเพื่อความแม่นยำของการงอนิ้ว 1 แขนที่เป็นแกนหลัก นั่นคือแกนในการกำมือ และเพื่อความง่ายในการพัฒนา เซนเซอร์ที่ใช้จะมีทั้งหมด 6 ตัว ประกอบไปด้วยเซนเซอร์วัดค่าความงอของนิ้ว 5 ตัว วางตามนิ้วแต่ละนิ้ว และเซนเซอร์หน่วยวัดอาศัยหลักความถี่ตรวจสอบการเคลื่อนไหวและการวางมือ วางไว้กลางหลังมือ 1 ตัว



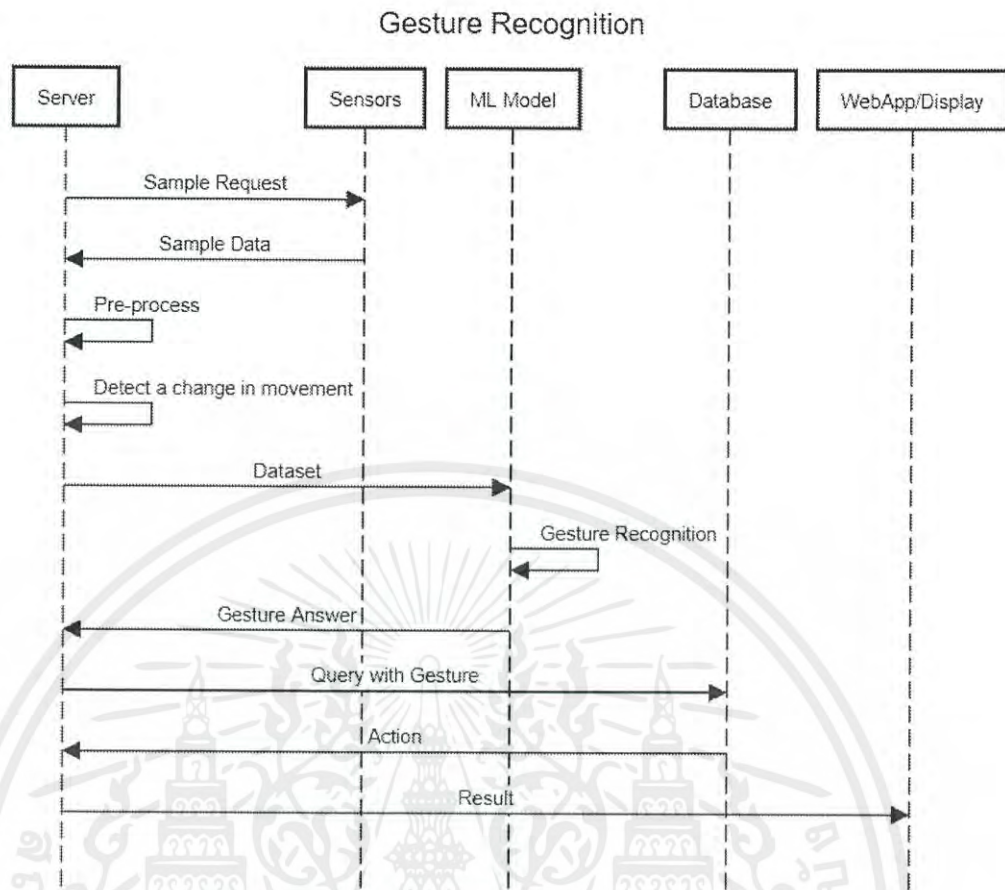
รูป 3.5 โครงสร้างถุงมือแบบที่ 1

### 3.2.2 ถุงมือแบบที่ 2:

เป้าหมายของถุงมือรูปแบบนี้ คือเพื่อความแม่นยำของตำแหน่งนิ้วในทุกแกน เซนเซอร์ที่ใช้จะมีทั้งหมด 6 ตัว ประกอบไปด้วยเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยของนิ้ว 5 ตัว วางตามปลายนิ้วบริเวณเล็บของแต่ละนิ้ว และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยตรวจสอบการเคลื่อนไหวและการวางมือ วางไว้กลางหลังมือ 1 ตัว [1]



รูป 3.6 โครงสร้างถุงมือแบบที่ 2

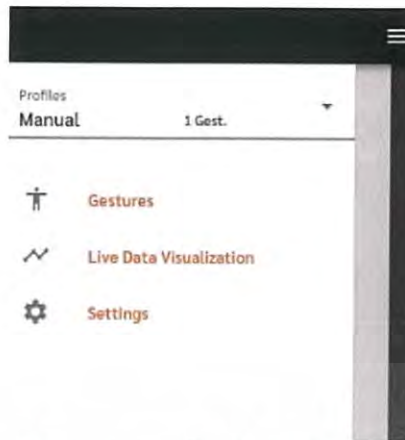


รูป 3.7 แผนผังลำดับการทำงานของ การตรวจจับท่าทาง

### 3.3 การออกแบบเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันออกแบบมาให้ใช้ได้หลากหลายอุปกรณ์ เมื่อผู้ใช้เข้าไปยังหน้าเว็บแอปพลิเคชัน จะมีการเปลี่ยนรูปแบบให้เหมาะสมกับอุปกรณ์นั้น ๆ ตามขนาดหน้าจอ (Responsive)

เว็บแอปพลิเคชันจะแบ่งออกเป็นส่วนหลัก ๆ 2 ส่วน คือ ส่วนของผู้ใช้ และส่วนของนักพัฒนาสามารถเข้าถึงได้ผ่านแถบเมนูด้านซ้าย



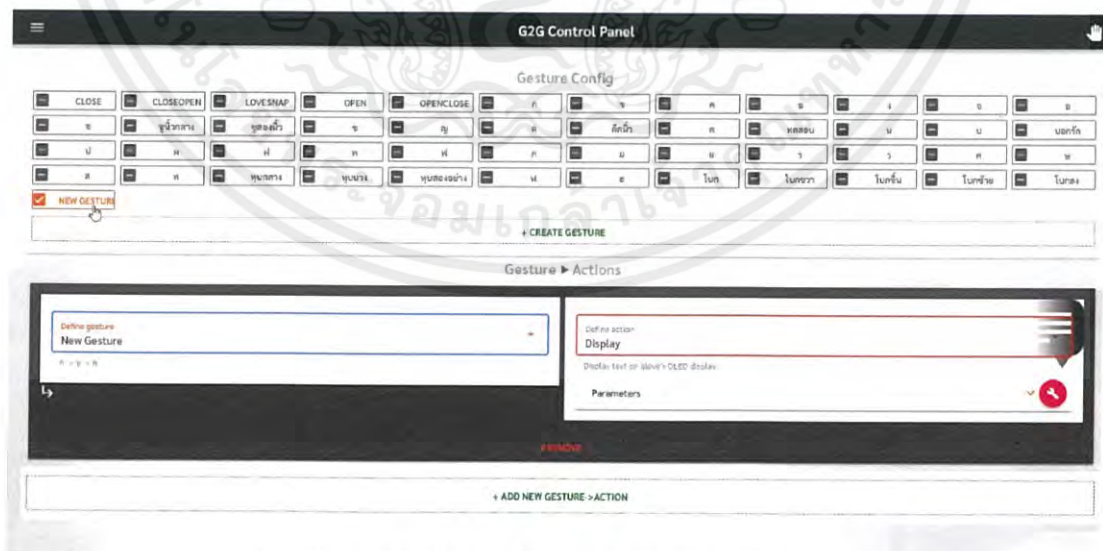
รูป 3.8 แถบเมนูด้านซ้ายของเว็บแอปพลิเคชัน

### 3.3.1 แถบเมนูด้านซ้าย

สามารถแสดงหรือซ่อนได้ไม่ว่าจะอยู่หน้าใดก็ตาม ด้านบนของเมนูจะเป็นกล่องเลือกโปรไฟล์ ผู้ใช้สามารถสร้าง ลบ หรือเปลี่ยนโปรไฟล์ได้ แต่ละโปรไฟล์จะมีท่าทางที่ตั้งค่าไว้เหมือนกัน แต่การตั้งค่าการกระทำจะแตกต่างกันไป ตามที่ผู้ใช้ตั้งไว้ในแต่ละโปรไฟล์ เช่น โปรไฟล์ ก ตั้งค่าท่าโบกซ้ายให้แสดงข้อความ แต่โปรไฟล์ ข ตั้งค่าท่าโบกซ้ายให้แสดงรูป เป็นต้น

### 3.3.2 ส่วนของผู้ใช้

มีจุดหมายเพื่อให้ผู้ใช้งานถนัดมือตั้งค่าท่าทางต่าง ๆ ว่าต้องการทำท่าแบบใดและต้องการให้แสดงผลหรือเกิดสิ่งใดขึ้น โดยจะมีการแสดงรายการท่าทางและขั้นตอนในการทำท่าทางนั้น ๆ เป็นทำย่อเรียงกัน ผู้ใช้สามารถเพิ่ม แก้ไข ลบท่าทางหรือทำย่อต่าง ๆ ได้



รูป 3.9 หน้าเว็บแอปพลิเคชันส่วนของผู้ใช้

ส่วนด้านบน (Gesture Config) ผู้ใช้สามารถคลิกเข้าไปตั้งค่าท่าแต่ละท่าได้ว่าประกอบไปด้วยท่าย่อยใดบ้าง หรือสร้างท่าใหม่ขึ้นมาตามที่ผู้ใช้ต้องการ โดยเบื้องต้นจะมี 40 ท่า ตามที่ถู่มือสามารถตรวจจับได้

ส่วนด้านล่าง (Gesture -> Actions) จะเป็นส่วนที่ตั้งค่าว่าให้ทำนั้น ๆ ทำการกระทำใดบ้าง ด้านซ้ายของส่วนคือท่าที่ต้องการตั้งค่า ด้านขวาของส่วนคือการกระทำ

หนึ่งท่าทางสามารถตั้งค่าให้กระทำหลายอย่างต่อกันได้ มุมล่างขวาของการกระทำจะมีเมนูวงกลม สามารถคลิกเข้าไปเพื่อเพิ่ม/ลบ การกระทำถัดไปได้

สามารถนำไปประยุกต์ใช้ได้หลายวัตถุประสงค์ เช่น แสดงข้อความชุดแรก จากนั้นรอ 5 วินาที แล้วจึงแสดงข้อความอีกชุดหนึ่ง

การกระทำแต่ละแบบจะมีพารามิเตอร์ที่ต้องการ คือส่วนที่ผู้ใช้ตั้งค่าได้และจำเป็นในการกระทำนั้น ๆ เช่น ข้อความที่ต้องการแสดง วินาทีที่ต้องการรอ ซึ่งแตกต่างกันไปตามประเภทการกระทำ

การกระทำมีให้เลือกทั้งหมด 8 แบบ โดยมีข้อมูลดังนี้  
ประเภทแสดงผล ได้แก่

1. Display: แสดงข้อความบนจอ OLED ของถู่มือ

- พารามิเตอร์ที่ต้องการ

- i. Text = ข้อความที่จะแสดงผล

- ค่าที่ส่งต่อ

- i. Text = ข้อความที่ได้แสดงผลไป

- ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ

2. Draw: วาดรูปลงบนจอ OLED ของถู่มือ

- พารามิเตอร์ที่ต้องการ

- i. Pixels = ชุดของตัวเลขฐานสองยาวตามขนาดพิกเซลของจอ OLED แทนสีที่จะวาดลงไปบนจอ โดย 0 แทนสีดำ และ 1 แทนสีขาว

- ค่าที่ส่งต่อ

- i. Text = มีค่าเป็น 'Drawn' หากทำการวาดสำเร็จ

- ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ

3. Screen: แสดงหน้าจอพื้นฐานของถู่มือ

- พารามิเตอร์ที่ต้องการ

- i. Screen to Show = หน้าจอที่จะแสดง มีให้เลือก 2 แบบ คือหน้านาฬิกา และ หน้าจอว่างเปล่า

- ค่าที่ส่งต่อ
  - i. Text = ชื่อหน้าจอบนเว็บที่แสดงไป
  - ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ
- 4. HTML: แสดงกล่องข้อความบนเว็บแอปพลิเคชัน สามารถแสดงตามรูปแบบของภาษา HTML ได้

- พารามิเตอร์ที่ต้องการ

- i. Code = ข้อความในรูปแบบ HTML ที่ต้องการแสดง

- ค่าที่ส่งต่อ

- i. Text = มีค่าเป็น 'HTML Completed' หากแสดงบนเว็บแอปพลิเคชันสำเร็จ
- ii. Code = ข้อความที่ได้แสดงไป
- iii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ

#### ประเภท IoT ได้แก่

5. HTTP: ส่ง HTTP Request ไปยัง URL ที่ต้องการและรอรับผลลัพธ์กลับมา สามารถตั้งค่า Method (GET,POST) Headers และ Body ได้

- พารามิเตอร์ที่ต้องการ

- i. Request Type = ประเภทของ Request มีให้เลือก 2 แบบ คือ GET และ POST
- ii. URL = ตำแหน่งของปลายทาง เขียนได้ทั้ง Domain name และ IP Address
- iii. Headers = ส่วนหัวของ Request
- iv. Request Parameters = ส่วนพารามิเตอร์หรือส่วนตัวของ Request

- ค่าที่ส่งต่อ

- i. Result = ข้อความ/วัตถุตอบกลับของ Request
- ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ

#### ประเภทระบบ ได้แก่

6. Command: สั่งให้ตัวโมโครคอนโทรลเลอร์รันคำสั่ง Linux ตามที่ตั้งค่าไว้

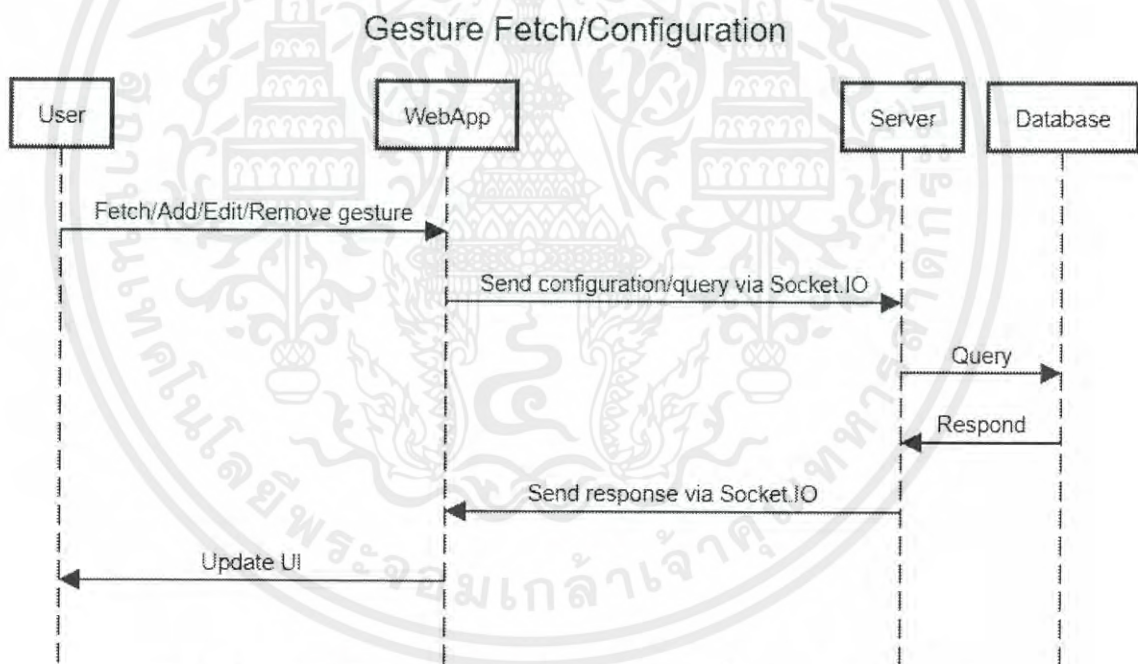
- พารามิเตอร์ที่ต้องการ

- i. Command = คำสั่งที่ต้องการใช้

- ค่าที่ส่งต่อ

- i. Result = ข้อความตอบกลับหลังจากใช้คำสั่ง
- ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ

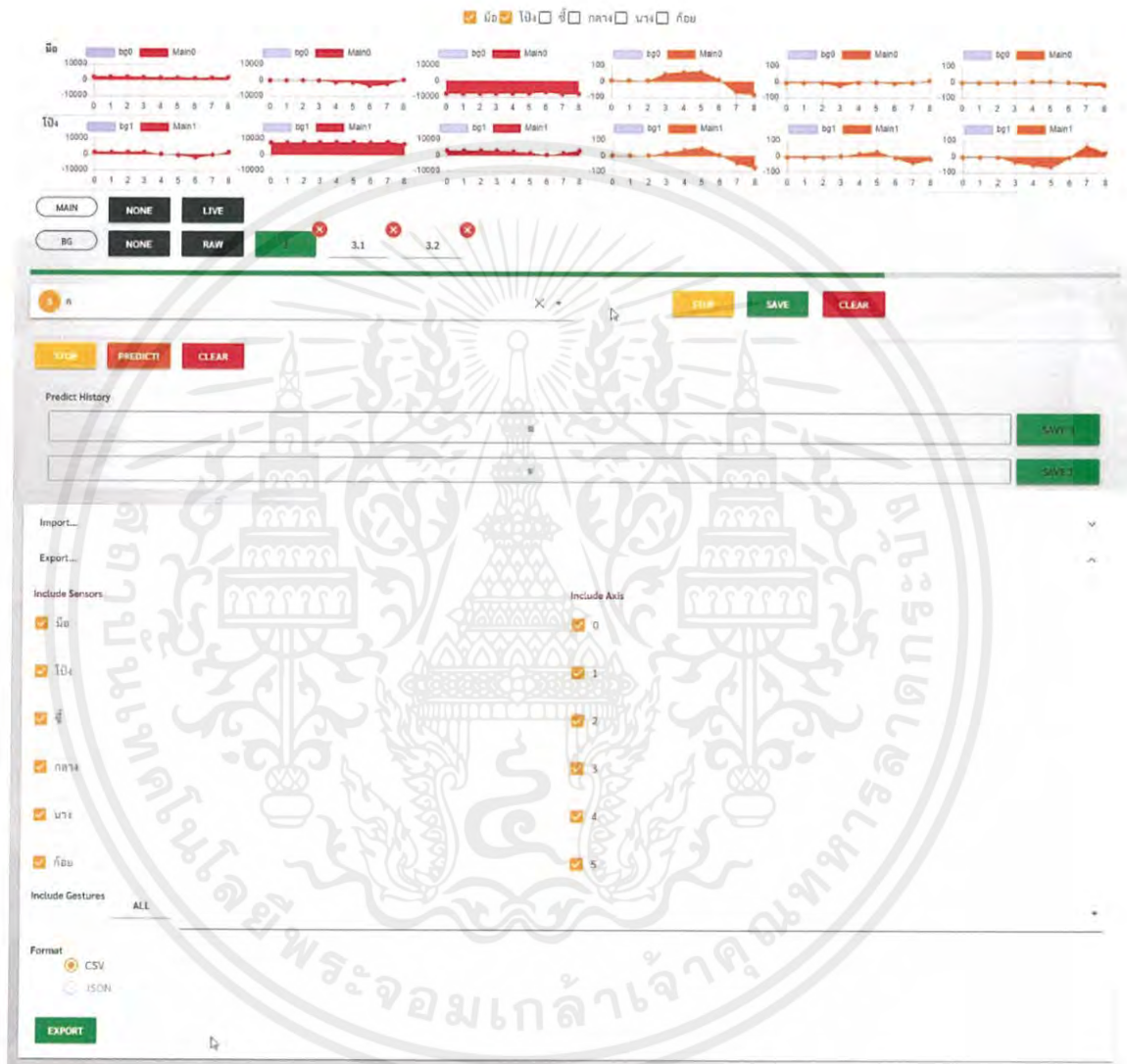
7. Wait: รอตามเวลาที่ตั้งค่าไว้ เมื่อรอจนครบถึงมือถึงจะทำงานต่อ
- พารามิเตอร์ที่ต้องการ
    - i. Time = เวลาที่ต้องการรอในหน่วยวินาที มีทศนิยมได้ 1 ตำแหน่ง
  - ค่าที่ส่งต่อ
    - i. Text = มีค่าเป็น 'Wait Completed' หากรอสำเร็จ
    - ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ
8. Switch Profile: เปลี่ยนโปรไฟล์เป็นอย่างอื่นตามที่ตั้งค่าไว้
- พารามิเตอร์ที่ต้องการ
    - i. Profile Name = ชื่อโปรไฟล์ที่ต้องการสลับไป
  - ค่าที่ส่งต่อ
    - i. Text = มีค่าเป็น 'Switching Completed' หากเปลี่ยนสำเร็จ
    - ii. Error = ปัญหาหากที่เกิดขึ้นระหว่างดำเนินการกระทำ



รูป 3.10 แผนผังลำดับการทำงานของ การเรียกดู/เพิ่ม/แก้ไข/ลบ ท่าทางของผู้ใช้

### 3.3.3 ส่วนของนักพัฒนา

มีจุดหมายเพื่อรวมถึงสิ่งที่มีประโยชน์ต่าง ๆ ที่ใช้ในกระบวนการพัฒนา ได้แก่ แสดงผลข้อมูลจากเซนเซอร์ บันทึก/นำเข้า/ส่งออกชุดข้อมูลสำหรับสร้างโมเดล Machine Learning และคูประวัติการตรวจจับท่าทาง



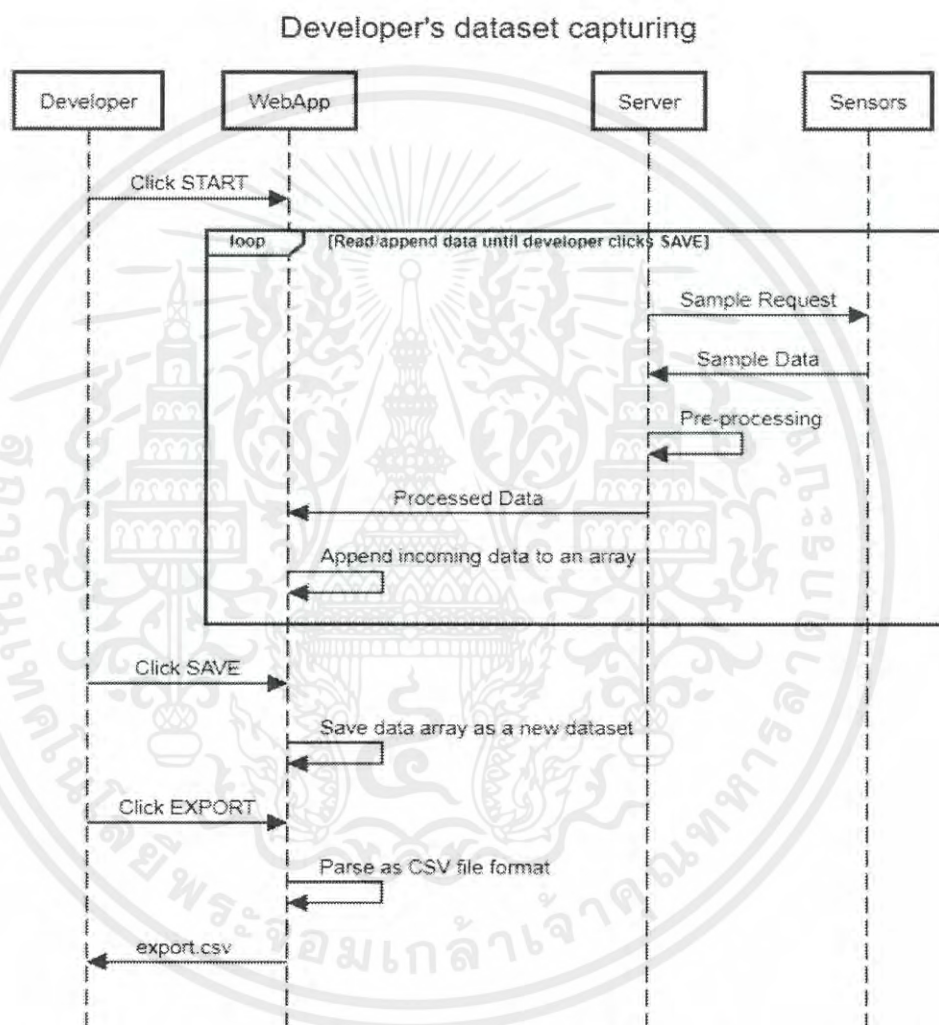
รูป 3.11 หน้าเว็บแอปพลิเคชันส่วนของนักพัฒนา

ส่วนด้านบน จะแสดงข้อมูลจากเซนเซอร์ทั้ง 6 สามารถเลือกเปิด/ปิดบางส่วนได้เพื่อลดหน่วยความจำที่ใช้ กลุ่มของปุ่มทางด้านขวาจะเป็นปุ่มที่เกี่ยวกับการบันทึกข้อมูลเซนเซอร์ สามารถใช้บันทึกข้อมูลเป็นชุดละ 10 (AUTO) หรือบันทึกข้อมูลเดียวจนกว่าจะกดหยุด (START/STOP) และลบที่บันทึกไว้ (CLEAR) ข้อมูลที่บันทึกแล้วจะแสดงเป็นปุ่มเล็ก ๆ ที่มีเครื่องหมายกากบาทอยู่ตรงมุม นักพัฒนาสามารถย้อนดู หรือลบข้อมูลชุดนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนตรงกลาง จะเป็นส่วนแสดงประวัติการตรวจจับท่าทาง กรณีเกิดการตรวจจับท่าที่ผิด แล้วนักพัฒนาต้องการข้อมูลชุดที่ผิดนั้น ไปฝึกโมเดลเพิ่ม สามารถกด SAVE ด้านขวาของรายการประวัติแต่ละรายการ เพื่อบันทึกข้อมูลชุดที่นำไปตรวจจับนั้น ๆ ได้

ส่วนด้านล่าง เป็นส่วนนำเข้า/ส่งออก (Import/Export) นักพัฒนาสามารถส่งออกข้อมูลทั้งหมดที่ได้บันทึกไป เป็นไฟล์ csv ที่พร้อมสำหรับการฝึกโมเดล หรือนักพัฒนาสามารถเลือกไฟล์ csv ที่ตรงรูปแบบเหมือนตอนส่งออกไป เพื่อนำเข้ามาแสดงผลเป็นกราฟได้



รูป 3.12 แผนผังลำดับการทำงานของการทำงานของการบันทึกและส่งออกชุดข้อมูลของนักพัฒนา

### 3.4 การออกแบบฐานข้อมูล

มีเป้าหมายสำหรับเก็บการตั้งค่าของผู้ใช้ จะใช้ Collection ทั้งหมด 5 Collection ประกอบไปด้วย

Pose: ทำหนึ่งหรือทำซ้ำ 1 ทำ เช่น กำมือ

Gesture: ชุดท่าทาง 1 ชุด ประกอบไปด้วย Pose หลายท่าที่ทำต่อเนื่องกัน เช่น แบมือแล้วค่อยกำมือ

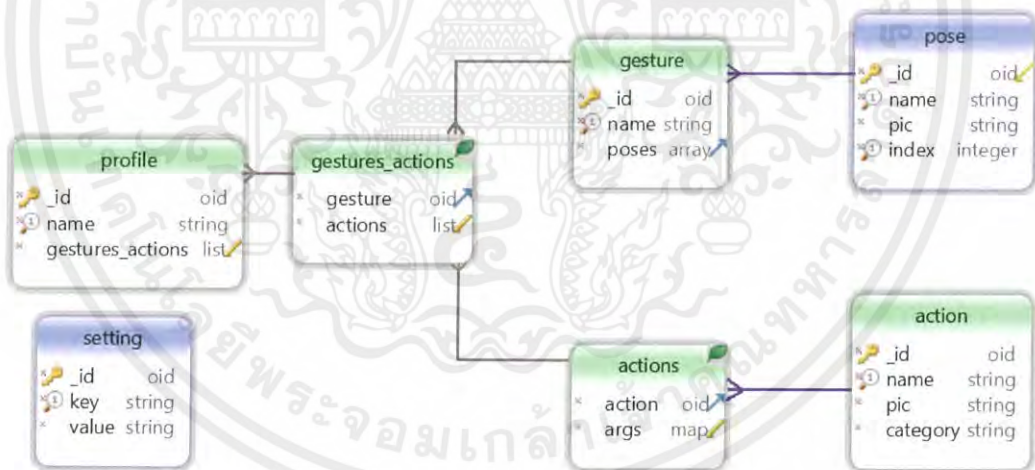
Action: การกระทำหลังจากตรวจจับท่าได้ 1 การกระทำ เช่น แสดงผลข้อความบนจอแสดงผล

Actions: ชุดของการกระทำ ประกอบไปด้วย Action หลายการกระทำที่ทำต่อเนื่องกัน (รวมถึง args ของแต่ละ Action เพื่อเก็บค่าที่จำเป็นต้องใช้เพิ่มเติมในการกระทำ) เช่น รอหนึ่งวินาที แล้วค่อยแสดงผลข้อความบนจอแสดงผล

Gestures\_actions: ประกอบไปด้วย Gesture และ Actions อย่างละชุด

Profile: โพรไฟล์ 1 โพรไฟล์ คือชุดของท่าและการกระทำ (Gestures\_actions) เช่น โพรไฟล์สำหรับตรวจจับและแสดงผลภาษามือไทย และ โพรไฟล์สำหรับใช้ภาษามืออังกฤษ

Setting: การตั้งค่าพื้นฐานที่จดจำไว้ในฐานข้อมูล เพื่อที่เมื่อเริ่มการทำงานของถุงมือใหม่จะได้ไม่ต้องตั้งค่าอีกครั้ง เช่น Profile เริ่มต้น



รูป 3.13 แผนภาพฐานข้อมูล

### 3.5 การออกแบบชุดข้อมูล

ชุดข้อมูลที่จะส่งไปใช้ในการฝึกโมเดลและการตรวจจับท่าทาง จะเป็นชุดของค่าที่จับได้จากเซนเซอร์ติดต่อกัน 10 ถึง 20 ครั้ง (ตามค่า Timestep ที่ใช้ในการทดสอบ) ที่แสดงการเปลี่ยนแปลงของค่าเซนเซอร์ติดต่อกันภายในเวลา 1 วินาที จึงทำให้ชุดข้อมูลหนึ่งจะมี 10 ถึง 20 ชุดย่อย แต่โครงสร้างของชุดย่อยจะแตกต่างกันไปตามรูปแบบของถุงมือทั้งสองแบบดังนี้

### 3.5.1 การออกแบบชุดข้อมูลของถุงมือแบบที่ 1

ชุดย่อยจะประกอบไปด้วยค่ามุมจากเซนเซอร์ 3 แกน และค่าจากเซนเซอร์วัดความเร่งตามแนว 5 แนว รวมเป็น 8 ค่า ชุดย่อยสามารถแสดงเป็นอาร์เรย์ได้ดังนี้

[มือ\_x, มือ\_y, มือ\_z, โป้่ง, ซี้, กลาง, นาง, ก้อย]

### 3.5.2 การออกแบบชุดข้อมูลของถุงมือแบบที่ 2: ข้อมูลมุม

ชุดย่อยจะประกอบไปด้วยค่ามุมจากเซนเซอร์ เบื้องต้นจะใช้ทั้ง 3 แกน รวมเป็น 18 ค่า ชุดย่อยสามารถแสดงเป็นอาร์เรย์ได้ดังนี้ [มือ\_x, มือ\_y, มือ\_z, โป้่ง\_x, โป้่ง\_y, โป้่ง\_z, ซี้\_x, ซี้\_y, ซี้\_z, กลาง\_x, กลาง\_y, กลาง\_z, นาง\_x, นาง\_y, นาง\_z, ก้อย\_x, ก้อย\_y, ก้อย\_z]

ดังนั้นหากนำชุดย่อยมาต่อกันตามจำนวน Timestep จะกลายเป็นชุดข้อมูล 1 ชุด และในการฝึกโมเดล จะใช้ชุดข้อมูลทั้งหมดอย่างน้อย 50 ชุด

### 3.5.3 การออกแบบชุดข้อมูลของถุงมือแบบที่ 2: ข้อมูลดิบ

ชุดย่อยจะประกอบไปด้วยค่าจากเซนเซอร์วัดความเร่งเชิงเส้น 3 แกน และเซนเซอร์วัดความเร็วเชิงมุม 3 แกน รวมเป็น 36 ค่า ชุดย่อยสามารถแสดงเป็นอาร์เรย์ได้ดังนี้

[มือ\_accel\_x, มือ\_accel\_y, มือ\_accel\_z, มือ\_gyro\_x, มือ\_gyro\_y, มือ\_gyro\_z, โป้่ง\_accel\_x, โป้่ง\_accel\_y, โป้่ง\_accel\_z, โป้่ง\_gyro\_x, โป้่ง\_gyro\_y, โป้่ง\_gyro\_z, ซี้\_accel\_x, ซี้\_accel\_y, ซี้\_accel\_z, ซี้\_gyro\_x, ซี้\_gyro\_y, ซี้\_gyro\_z, กลาง\_accel\_x, กลาง\_accel\_y, กลาง\_accel\_z, กลาง\_gyro\_x, กลาง\_gyro\_y, กลาง\_gyro\_z, นาง\_accel\_x, นาง\_accel\_y, นาง\_accel\_z, นาง\_gyro\_x, นาง\_gyro\_y, นาง\_gyro\_z, ก้อย\_accel\_x, ก้อย\_accel\_y, ก้อย\_accel\_z, ก้อย\_gyro\_x, ก้อย\_gyro\_y, ก้อย\_gyro\_z]

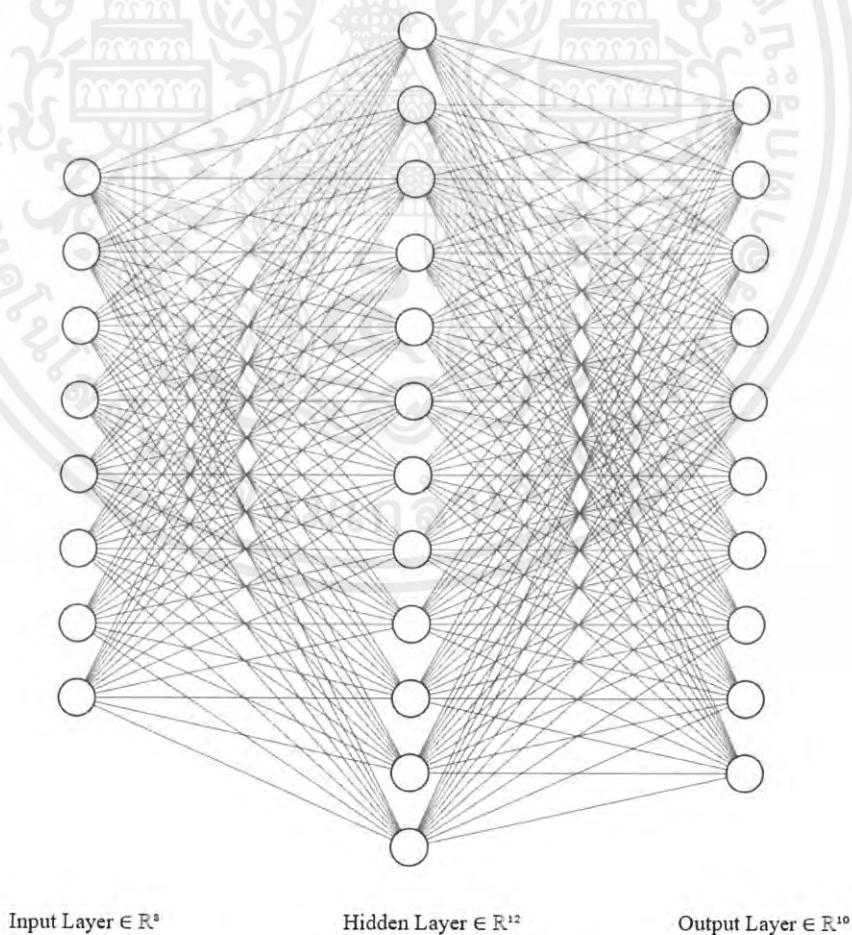
ดังนั้นหากนำชุดย่อยมาต่อกันตามจำนวน Timestep จะกลายเป็นชุดข้อมูล 1 ชุด และในการฝึกโมเดล จะใช้ชุดข้อมูลทั้งหมดอย่างน้อย 50 ชุด

### 3.6 การออกแบบโมเดล Machine Learning

โมเดลที่ใช้ในการฝึกทำทางที่มีข้อมูลของเซนเซอร์ในแต่ละนิ้ว เลือกเป็นโครงข่ายประสาทเทียม (Neural Network) เพราะมีชุดข้อมูลจำนวนมากในการฝึกทำทาง โครงข่ายประสาทเทียมช่วยทำให้ไม่ต้องหาคุณลักษณะ (Feature) ให้กับข้อมูล และใช้การเรียนรู้แบบมีการสอน (Supervised Learning) ก็คือการเรียนที่มีการตรวจคำตอบเพื่อให้โครงข่ายประสาทเทียมปรับตัวเอง เพื่อให้ได้คำตอบที่ดีขึ้น โดยโมเดลสำหรับข้อมูลที่ใช้ในการฝึกจะทดลองทั้งหมด 3 โมเดลดังนี้

#### 3.6.1 การออกแบบโมเดลของถูงมือแบบที่ 1 สำหรับทำนึ่ง

เป็นโมเดลโครงข่ายประสาทเทียมธรรมดา (Neural Network) จะใช้ไลบรารีของ Scikit-learn ในการสร้างโมเดล มีคุณลักษณะ (Feature) ทั้งหมด 8 ลักษณะ ประกอบด้วยเซนเซอร์วัดความงอตามนิ้วทั้งหมด 5 ลักษณะ เซนเซอร์วัดมุมข้อมือ 3 ลักษณะ เพื่อให้ได้ผลลัพธ์ เป็นทำทางของมือรูปแบบหนึ่ง ก่อนที่จะนำคุณลักษณะทั้งหมดไปฝึกจำเป็นต้องแปลง (Transform) ก่อน เพื่อให้การทำนายผลลัพธ์แม่นยำยิ่งขึ้น โดยโครงสร้างของโครงข่ายประสาทเทียมประกอบด้วย ชั้นขาเข้า (Input layer) คือลักษณะทั้งหมด 8 ชั้น ชั้นซ่อน (Hidden layer) ใช้เป็น 12 ชั้น

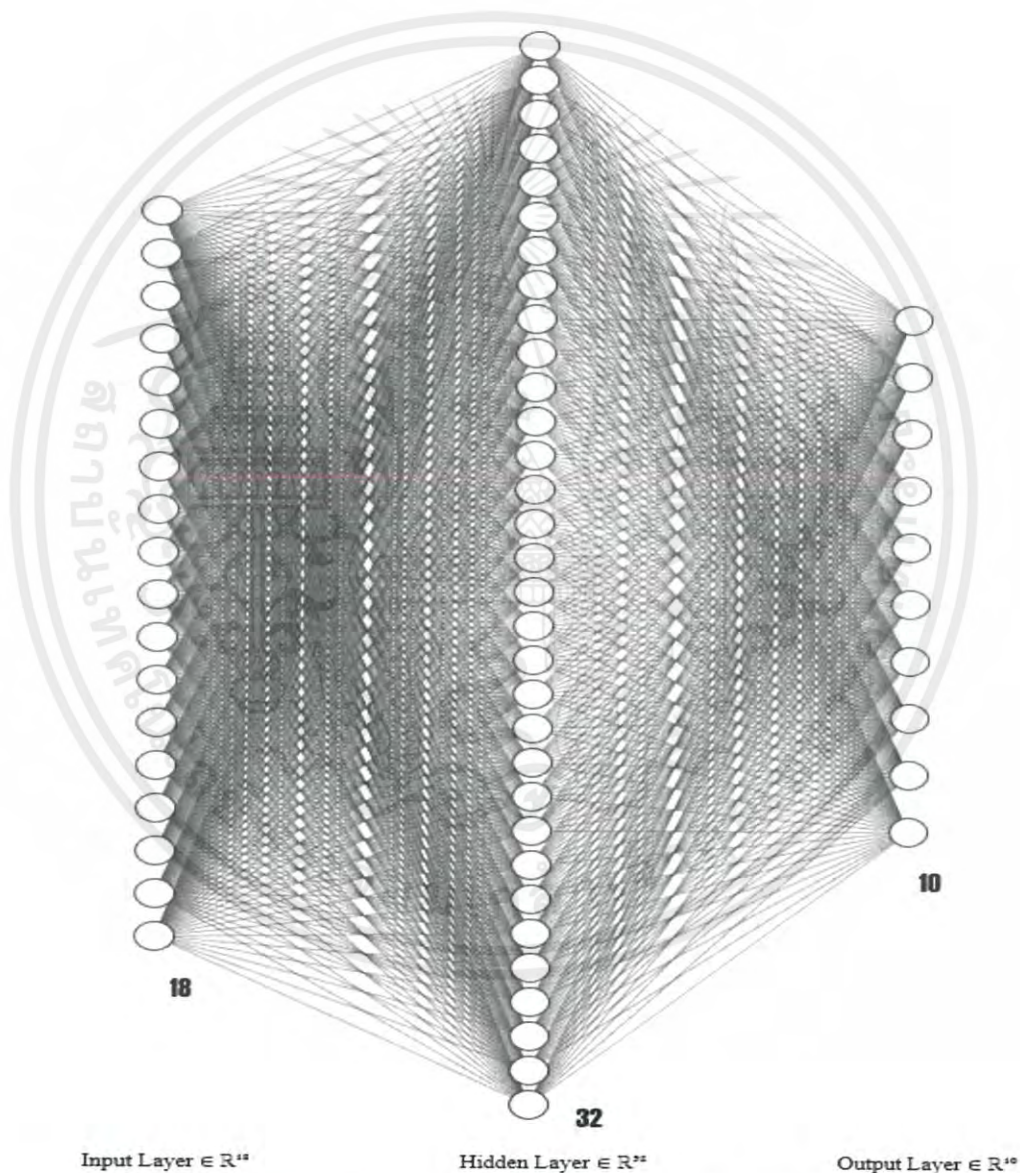


รูป 3.14 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6.2 การออกแบบโมเดลของถุงมือแบบที่ 2 สำหรับทำนึ่ง

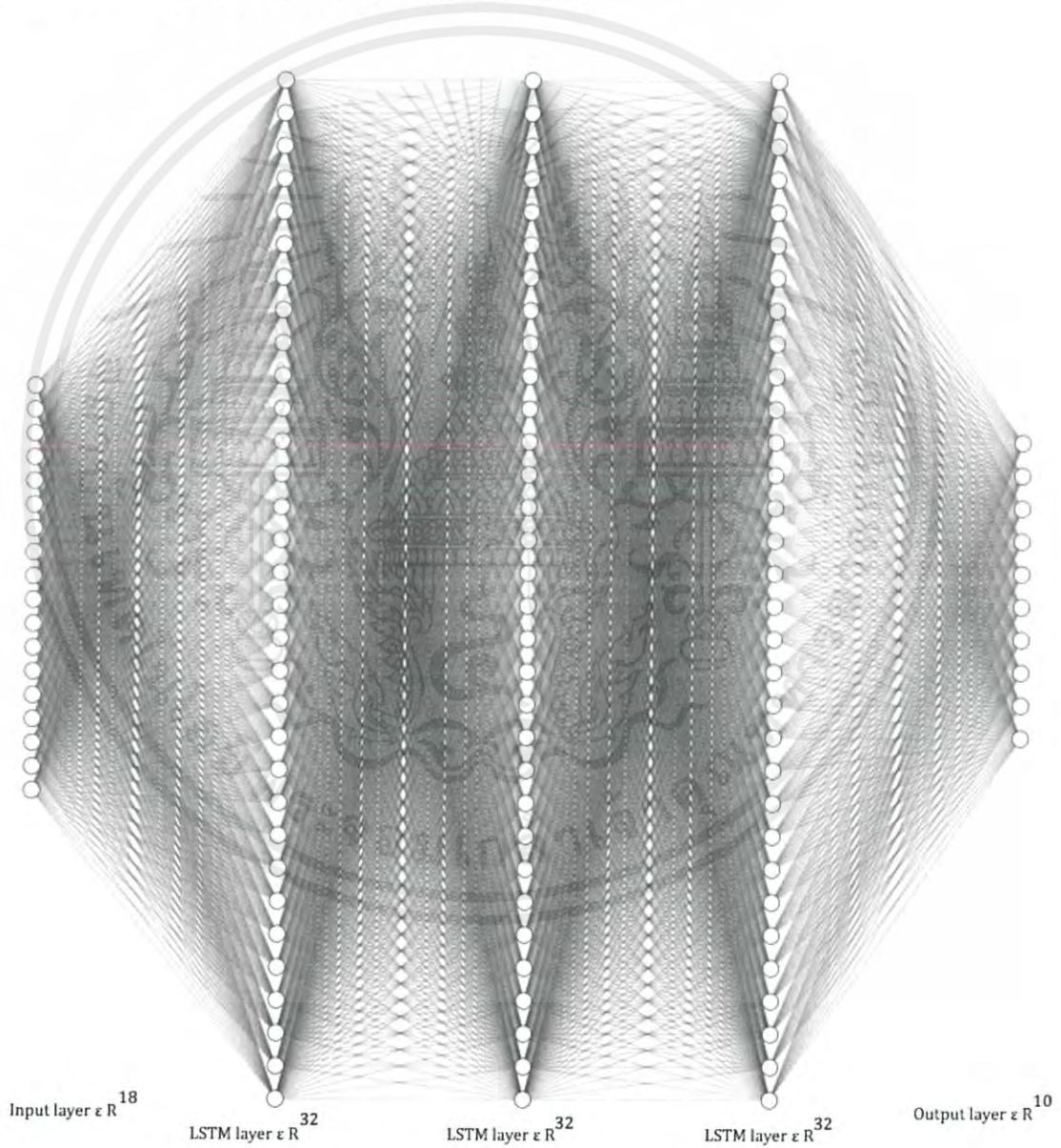
เป็น โมเดลโครงข่ายประสาทเทียมธรรมดา จะใช้ไลบรารีของ Keras ซึ่งมี Tensorflow เป็น Back-end ในการสร้างโมเดล มีคุณลักษณะ (Feature) ทั้งหมด 18 ลักษณะ ประกอบด้วยเซนเซอร์วัดมุมของแต่ละนิ้วทั้งหมด 15 ลักษณะ เซนเซอร์วัดมุมข้อมือ 3 ลักษณะ เพื่อให้ได้ผลลัพธ์เป็นท่าทางของมือรูปแบบหนึ่ง โครงสร้างของประสาทเทียมประกอบด้วย ชั้นขาเข้าคือลักษณะทั้งหมด 18 ชั้น และชั้นซ่อนไว้เป็น 32 ชั้น ส่วนชั้นขาออก (Output layer) ใช้ตามจำนวนของท่าทั้งหมดที่ฝึกเป็น 10 ชั้น



รูป 3.15 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 2

### 3.6.3 การออกแบบโมเดลของถุงมือแบบที่ 2 สำหรับทำนึ่งและเคลื่อนไหวโดยใช้มูม

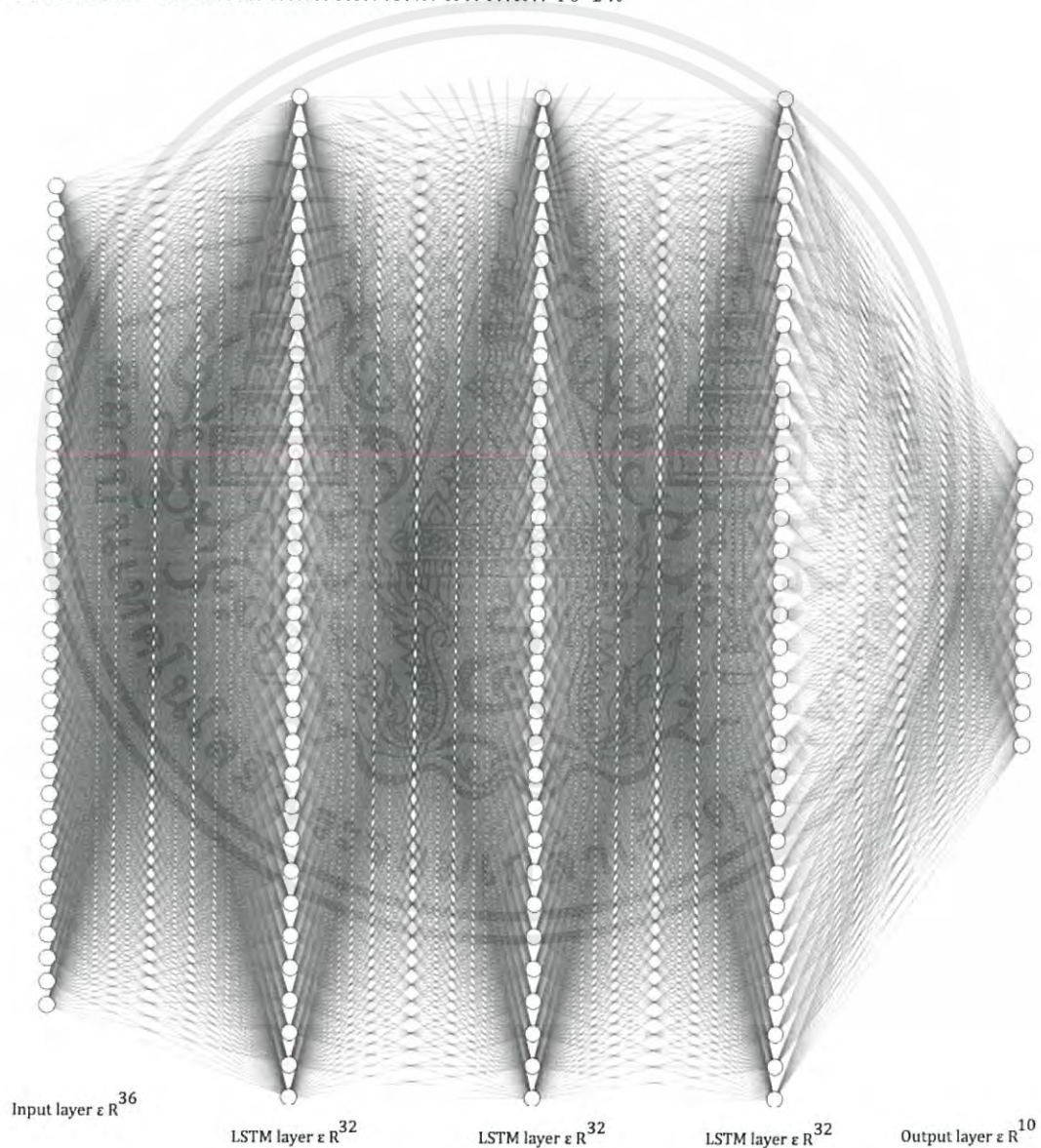
เป็น โมเดลโครงข่ายประสาทเทียมที่ออกแบบมาสำหรับประมวลผลลำดับ (LSTM) จะใช้ไลบรารีของ Keras ในการสร้างโมเดล มีคุณลักษณะทั้งหมด 18 ลักษณะ ประกอบด้วย เซนเซอร์วัดมุมของแต่ละนิ้วทั้งหมด 15 ลักษณะ เซนเซอร์วัดมุมข้อมือ 3 ลักษณะ โดยมีลำดับของท่าทางทั้งหมด 20 ครั้ง (Timestep) เพื่อให้ได้ผลลัพธ์เป็นท่าทางของมือทั้งทำนึ่งและเคลื่อนไหว โครงสร้างของประสาทเทียมประกอบด้วย ชั้นขาเข้าเป็นอาร์เรย์ลักษณะ (20,18) 20 ก็คือจำนวนครั้งของลำดับ ส่วน 18 ก็คือคุณลักษณะทั้งหมด ชั้นซ่อนใช้เป็น 32 ชั้น ทั้งหมด 3 เลเยอร์ และชั้นขาออกใช้เป็นจำนวนท่าทั้งหมดในการฝึก มีทั้งหมด 10 ชั้น



รูป 3.16 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 3

### 3.6.4 การออกแบบโมเดลของถุงมือแบบที่ 2 สำหรับทำนึ่งและเคลื่อนไหวโดยใช้ข้อมูลดิบ

เป็นโมเดลโครงข่ายประสาทเทียมที่ออกแบบมาสำหรับประมวลผลลำดับ (LSTM) จะใช้ไลบรารีของ Keras ในการสร้างโมเดล มีคุณลักษณะทั้งหมด 36 ลักษณะ ประกอบด้วย เซนเซอร์วัดมุมของแต่ละนิ้วและข้อมือ 36 ลักษณะเป็นข้อมูลดิบทั้งหมด โดยมีลำดับของท่าทางทั้งหมด 10 ครั้ง (Timestep) เพื่อให้ได้ผลลัพธ์เป็นท่าทางของมือทั้งทำนึ่งและเคลื่อนไหว โครงสร้างของประสาทเทียมประกอบด้วย ชั้นขาเข้าเป็นอารีย์ลักษณะ (10,36) 10 ก็คือจำนวนครั้งของลำดับ ส่วน 36 ก็คือคุณลักษณะทั้งหมด ชั้นซ่อนใช้เป็น 32 ชั้น ทั้งหมด 3 เลเยอร์ และชั้นขาออกใช้เป็นจำนวนท่าทั้งหมดในการฝึก มีทั้งหมด 10 ชั้น

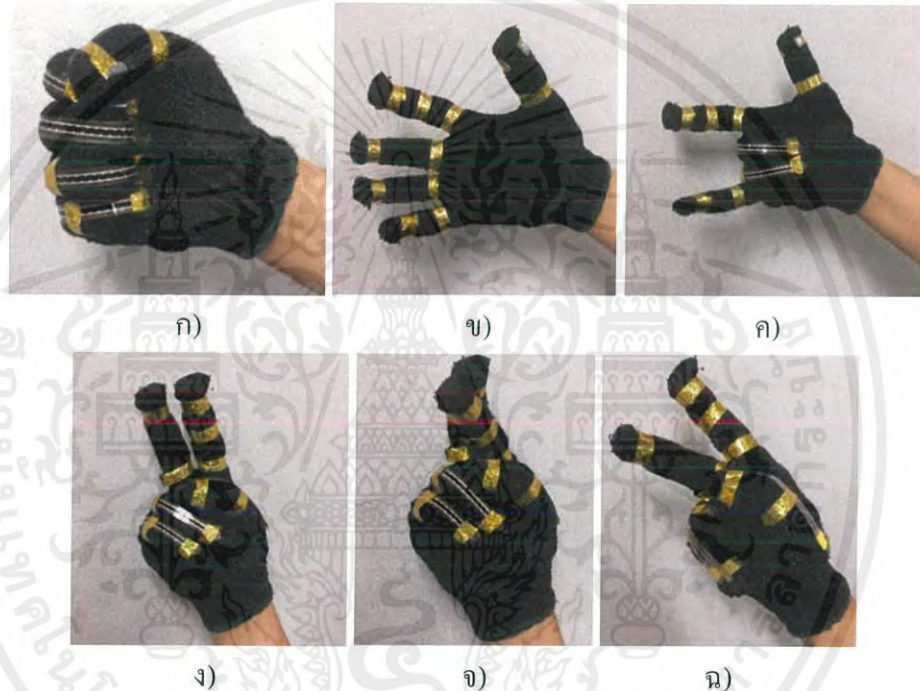


รูป 3.17 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 4

## บทที่ 4

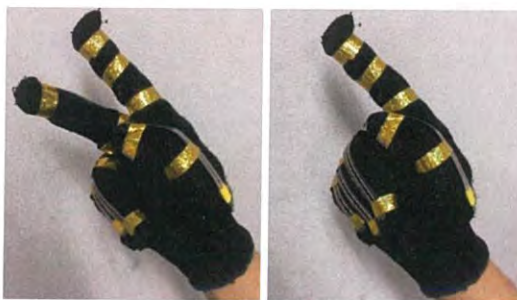
### การทดลอง

การทดลองในเบื้องต้น คือการทดลองเพื่อค้นหาวิธีในการใช้ถุงมือ และวิธีการตรวจจับท่าทางมือให้ได้แม่นยำที่สุด โดยท่าทางที่เลือกมามีทั้งสิ้น 9 ท่าทาง ที่มีลักษณะแตกต่างกัน 9 แบบ ซึ่งมีความสำคัญในการวัดความความแม่นยำในส่วนต่าง ๆ ของถุงมือ ท่าทางที่เลือกมาใช้ในการทดลองมีดังนี้



รูป 4.1 ท่าหนึ่งที่ใช้ในการทดสอบ

- ก) กำมือ
- ข) แแบมือ
- ค) บอกรัก
- ง) ชูสองนิ้ว
- จ) ไขว้นิ้ว
- ฉ) พยัญชนะ ก



ก)

ข)

รูป 4.2 ทำเคลื่อนไหวที่ใช้ในการทดสอบ - พัลซุนะ ข

ก) ชั้นตอนแรก

ข) ชั้นตอนที่สอง



ก)

ข)

ค)

รูป 4.3 ทำเคลื่อนไหว - พัลซุนะ ค

ก) ชั้นตอนแรก

ข) ชั้นตอนที่สอง

ค) ชั้นตอนที่สาม



ก)

ข)

รูป 4.4 ทำเคลื่อนไหว - พัลซุนะ ง

ก) ชั้นตอนแรก

ข) ชั้นตอนที่สอง

ท่ากำมือและแบมือ เป็นท่าพื้นฐานที่ถุงมือตรวจจับท่าทางควรจะสามารถตรวจจับได้ ท่าบอกรัก เป็นท่าที่ใช้ทดสอบกรณีค่าของทุกนิ้วไม่เท่ากัน ท่าชูสองนิ้วและไขว้นิ้ว เป็นท่าที่ทดสอบการแยกแยะท่าในกรณีที่ท่าทั้งสองแตกต่างกันในแกนอื่นที่ไม่ใช่แกนหลักของการอนิ้ว และท่าพยัญชนะเป็นท่าที่ทดสอบการตรวจจับท่าที่มีการเคลื่อนไหว

เมื่อได้โมเดลที่มีความแม่นยำมากกว่า 90% แล้ว จะทำการเพิ่มท่าชุดสุดท้ายเข้าไปจนครบตามขอบเขตโครงการ แล้วจึงทำการทดสอบความแม่นยำอีกครั้ง โดยทำในชุดสุดท้ายมีทั้งหมดดังนี้



รูป 4.5 ท่าชุดสุดท้าย

การทดลองแบ่งได้เป็น 4 ขั้นตอนหลัก ๆ ได้แก่ การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ การทดลองเกี่ยวกับเว็บแอปพลิเคชัน และการทดลองเกี่ยวกับ Machine Learning

## 4.1 การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ

คือการทดลองเกี่ยวกับถุงมือที่ได้ออกแบบมาทั้งสองแบบ ว่าสามารถจับค่าได้แม่นยำและมีปัญหาในการจับค่าหรือไม่

### 4.1.1 การทดลองใช้ถุงมือแบบที่ 1

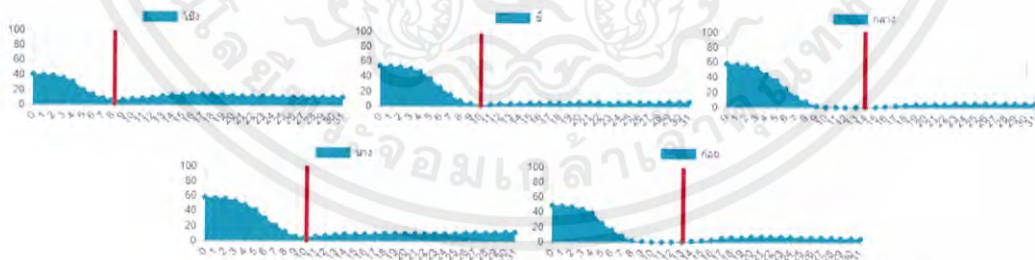
#### 4.1.1.1 วิธีการทดลอง

ประดิษฐ์ถุงมือแบบที่ 1 ขึ้นตามที่ออกแบบไว้ จากนั้นลองอ่านข้อมูลที่ได้จากเซนเซอร์ทุกตัว (เซนเซอร์วัดค่าความงอ 5 ตัว และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อย 1 ตัว) สำหรับเซนเซอร์วัดค่าความงอจะมีค่าตั้งแต่ 0-1023 และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยจะอ่านเพียงค่าเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยเชิงเส้น 3 แกนและเซนเซอร์หน่วยวัดความเร็วเชิงมุม 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 จากนั้นลองทำท่าหนึ่งเบื่องต้น ได้แก่ท่ากำมือ แบนมือ ชูนิ้วแต่ละนิ้ว ชูสองนิ้ว และไขว้ นิ้ว

#### 4.1.1.2 ผลการทดลอง

ถุงมือไม่มีปัญหาด้านการจับค่า สามารถทำการจับค่าได้ตามปกติ ยกเว้นในบางกรณีที่จับค่าไม่ได้หรือเพี้ยน ดังนี้

- 1) กรณีแบนมือจนสุด จะทำให้เซนเซอร์วัดค่าความงอเกิดการงอ ถุงมือจะจับค่าได้เท่ากับการงอนิ้ว จากรูป 4.6 สามารถสังเกตได้ว่า หลังจากเส้นสีแดงค่ามีแนวโน้มเพิ่มขึ้นเล็กน้อย เกิดจากการงอของเซนเซอร์วัดค่าความงอเมื่อแบนมือมากเกินไป ในผู้ใช้บางรายอาจจะแบนมือได้มากกว่านี้ ซึ่งอาจจะทำให้ไม่สามารถแยกท่าบางท่าออกจากกันได้



รูป 4.6 แผนภาพชุดข้อมูลของท่าแบนมือ (ถุงมือแบบที่ 1)

- 2) กรณีที่มีการไขว้ นิ้ว ค่าจะอยู่นอกเกณฑ์เซนเซอร์วัดค่าความงอสามารถจับได้ ตัวอย่างเช่นการไขว้ นิ้วชี้กับนิ้วกลาง ค่าจะได้ใกล้เคียงกับการกางนิ้วทั้งสองปกติ

- 3.) กรณีที่งอนี้วโป้งไปหลายทิศทาง ค่าจะอยู่นอกเกณฑ์เช่นเซอร์วัดค่าความงสามารถจับได้เช่นกัน ตัวอย่างเช่น มุมระหว่างนี้วโป้งกับนี้วชี้ การกางเป็นมุมฉากกับการแนบชนิดนี้วทั้งสองจะได้ค่าเหมือนกัน

#### 4.1.2 การทดลองใช้ถูงมือแบบที่ 2

##### 4.1.2.1 วิธีการทดลอง

ประดิษฐ์ถูงมือแบบที่ 2 ขึ้นตามทีออกแบบไว้ จากนั้นลองอ่านข้อมูลทีได้จากเซนเซอร์ทุกตัว (เช่นเซอร์หน่วยวัดอาศัยหลักความเฉื่อย 6 ตัว) สำหรับเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยทุกตัว จะอ่านเพียงค่าเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยเชิงเส้น 3 แกนและเซนเซอร์หน่วยวัดความเร็วเชิงมุม 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 จากนั้นลองทำทำนึ่งเบื่องต้น ได้แก่ทำกำมือ แบมือ ชูนี้วแต่ละนี้ว ชูสองนี้ว และไขว้นี้ว

##### 4.1.2.2 ผลการทดลอง

ถูงมือไม่มีปัญหาค้านการจับค่า ทำทีเคยมีปัญหในถูงมือแบบที่ 1 มีค่าแตกต่างกันอย่างชัดเจน ทำใหสามารถแยกทำทีเคยมีปัญหออกจากกัน ได้

จากการทดลอง 4.1 ทั้งหมดพบว่า ถูงมือแบบที่ 2 มีโอกาสในการจับค่าผิดพลาดทีน้อยกว่า แต่ยงต้องนำค่าไปปรับปรุงและประยุกต์ใช้ต่อไป

#### 4.2 การทดลองการปรับปรุงและประยุกต์ค่าจากถูงมือ

เนื่องค่าทีได้จากเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยโดยตรงนั้นเป็นค่าทีมีช่วงกว้างและไม่สามารถนำไปใช้ในขั้นตอนการทดลองเกี่ยวกับ Machine Learning ได้โดยตรง การทดลองนี้จึงเป็นการนำค่าโดยตรงจากถูงมือมาปรับปรุงหรือประยุกต์ใช้ในการหาค่าใหม่ทีมีประโยชน์มากขึ้นก่อนนำมาวิเคราะห์ด้วย Machine Learning โดยใช้วิธีต่าง ๆ

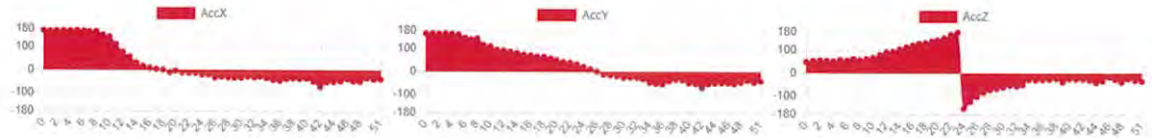
##### 4.2.1 การทดลองใช้สูตรคำนวณหามุมจากเซนเซอร์วัดความเร็ว

###### 4.2.1.1 วิธีการทดลอง

อ่านค่าของเซนเซอร์วัดความเร็วเชิงเส้น 3 แกน แล้วนำมาคำนวณตามสูตรคำนวณหามุมจากเซนเซอร์วัดความเร็ว ให้กลายเป็นค่ามุม 3 ค่า โดยเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยทีอยู่หลังมือจะมีค่ามุมระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยทีอยู่ปลายนี้วจะนำไปหาความแตกต่างจาก เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยทีอยู่หลังมือก่อนกลายเป็นค่าความแตกต่างระหว่างช่วง -180 ถึง 180 จากนั้นลองทำทำนึ่งเบื่องต้น ขยับมือช้า และขยับมือเร็ว

#### 4.2.1.2 ผลการทดลอง

สำหรับทำนึ่งและการขยับอย่างช้า ๆ จะได้ค่ามุม 3 แกนที่แม่นยำ แต่เมื่อมีการขยับมืออย่างรวดเร็ว จะทำให้มีค่ามุมที่เพี้ยนเล็กน้อย เกิดจากธรรมชาติของเซนเซอร์วัดความเร่ง



รูป 4.7 แผนภาพชุดข้อมูลเฉพาะนิ้วชี้ของท่าแบมือ (ถุงมือแบบที่ 2)

### 4.2.2 การทดลองใช้การคำนวณมุมด้วยตัวกรองร่วมสองแบบ

#### 4.2.2.1 วิธีการทดลอง

อ่านค่าของเซนเซอร์วัดความเร่งเชิงเส้น 3 แกน แล้วนำมาคำนวณตามขั้นตอนของตัวกรองร่วมสองแบบ ให้กลายเป็นค่ามุม 3 ค่า โดยเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่หลังมือจะมีค่ามุมระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่ปลายนิ้วจะนำไปหาความแตกต่างจาก เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่หลังมือก่อน กลายเป็นค่าความแตกต่างระหว่างช่วง  $-180$  ถึง  $180$  จากนั้นลองทำทำนึ่งเบื่องต้น ขยับมือช้า ขยับมือเร็ว และลองปรับค่า  $\alpha$  ในสูตรระหว่างช่วง  $(0,1)$  ให้คำนวณค่าที่ผิดพลาดน้อยที่สุด

#### 4.2.2.2 ผลการทดลอง

ที่ค่า  $\alpha=0.6$  จะได้ค่ามุม 3 มุมที่แม่นยำ และไม่มีปัญหาเมื่อทำการขยับมืออย่างรวดเร็ว แต่เมื่อมีการขยับมีเกินค่า 360 องศา หรือต่ำกว่า 0 องศา การคำนวณด้วยตัวกรองร่วมสองแบบจะพยายามทำให้ค่าเปลี่ยนไปอย่างช้า ๆ ซึ่งทำให้ค่าเพี้ยนไปจากความเป็นจริงพอสมควร

### 4.2.3 การทดลองใช้การคำนวณมุมด้วยตัวกรองคาลมาน

#### 4.2.3.1 วิธีการทดลอง

อ่านค่าของเซนเซอร์วัดความเร่งเชิงเส้น 3 แกน แล้วนำมาคำนวณตามขั้นตอนของตัวกรองคาลมาน ให้กลายเป็นค่ามุม 3 ค่า โดยเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่หลังมือจะมีค่ามุมระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่ปลายนิ้วจะนำไปหาความแตกต่างจาก เซนเซอร์หน่วยวัดอาศัยหลักความเฉื่อยที่อยู่หลังมือก่อน กลายเป็นค่าความแตกต่างระหว่างช่วง  $-180$  ถึง  $180$  จากนั้นลองทำทำนึ่งเบื่องต้น ขยับมือช้า และขยับมือเร็ว

#### 4.2.3.2 ผลการทดลอง

ได้ค่ามุม 3 มุมที่แม่นยำ และไม่มีปัญหาเมื่อทำการขยับมืออย่างรวดเร็ว เมื่อมีค่ามุมเกินค่า 360 องศา หรือต่ำกว่า 0 องศา การคำนวณด้วยตัวกรองคาลมานจะยังคงสร้างค่าที่เพี้ยน

ไปจากความเป็นจริงเหมือนกับการใช้ตัวกรองร่วมสองแบบ และการทำงานของไมโครคอนโทรลเลอร์โดยรวมช้าลง และมีแนวโน้มที่จะใช้หน่วยความจำจนหมดได้เร็วขึ้น

#### 4.2.4 การทดลองการตรวจจับการเคลื่อนไหวและหยุดนิ่ง

##### 4.2.5.1 วิธีการทดลอง

กำหนดค่าขั้นต่ำของการเคลื่อนไหว (Threshold) ของเซนเซอร์วัดความเร่งเชิงเส้น (ภายในช่วง 0-20 เมตรต่อวินาที<sup>2</sup>) และเซนเซอร์วัดความเร็วเชิงมุม (ภายในช่วง 0-50 องศาต่อวินาที) ไว้ และให้บันทึกข้อมูลตามค่าจริงในเฉพาะข้อมูลที่มีการเคลื่อนไหวจากจุดต่ำสุดมากกว่าค่าขั้นต่ำเท่านั้น หากน้อยกว่าค่าขั้นต่ำให้เปลี่ยนค่าให้เท่ากับที่แล้ว ทดลองปรับค่าขั้นต่ำในช่วงไปเรื่อย ๆ จนกว่าจะไม่มีค่าที่เกิดจากความผิดพลาดของเซนเซอร์ (Noise) เมื่อหยุดนิ่ง

##### 4.2.5.2 ผลการทดลอง

ค่าขั้นต่ำเซนเซอร์วัดความเร็วเชิงมุมที่ค่า 10 องศาต่อวินาที และค่าขั้นต่ำเซนเซอร์วัดความเร่งเชิงเส้นที่ 0 เมตรต่อวินาที<sup>2</sup> (ไม่ต้องมีค่าขั้นต่ำ) ข้อมูลที่บันทึกเมื่อทำท่าหนึ่งมีค่าที่คงที่ ไม่มีค่าที่เกิดจากความผิดพลาดของเซนเซอร์เมื่อหยุดนิ่ง และสามารถเริ่มตรวจจับการเคลื่อนไหวเพื่อเปลี่ยนท่าได้ดีที่สุด

#### 4.3 การทดลองเกี่ยวกับเว็บแอปพลิเคชัน

คือการทดลองพัฒนาเว็บแอปพลิเคชันในส่วนต่าง ๆ รวมถึงการเชื่อมต่อ แลกเปลี่ยนข้อมูลกับเซิร์ฟเวอร์ เพื่อนำข้อมูลจากเซิร์ฟเวอร์มาประยุกต์ใช้งานเกี่ยวกับ Machine Learning หรือขั้นตอนอื่น ๆ ของโครงการต่อไป

##### 4.3.1 การทดลองการส่งค่าผ่าน Socket.IO

###### 4.3.1.1 วิธีการทดลอง

เชื่อมต่อเว็บแอปพลิเคชันเข้ากับเซิร์ฟเวอร์ผ่านทาง Socket.IO เมื่อเริ่มเข้าเว็บไซต์ ส่งชุดข้อมูลจำนวน 20 ชุดต่อวินาทีจากเซิร์ฟเวอร์ไปยังเว็บแอปพลิเคชัน จากนั้นลองกดปุ่มบนเว็บแอปพลิเคชันเพื่อส่งข้อมูลล่าสุดทั้ง 20 ชุดกลับไปวิเคราะห์ที่เซิร์ฟเวอร์

###### 4.3.1.2 ผลการทดลอง

ทั้งสองฝั่งสามารถส่งข้อมูลหากันได้ทุกครั้งตลอดกระบวนการ

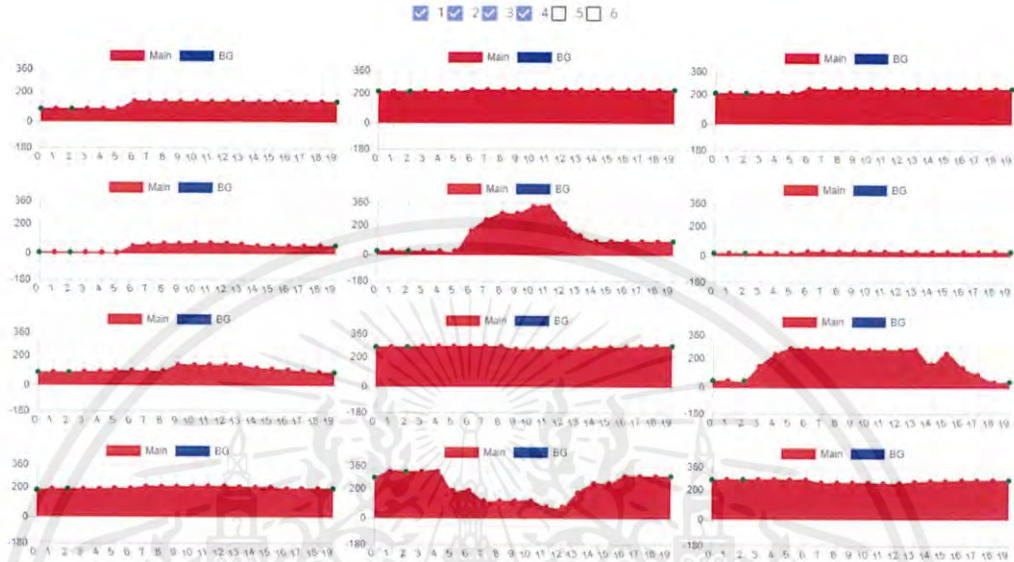
##### 4.3.2 การทดลองการแสดงผลข้อมูลแบบเปลี่ยนแปลงตามเวลา

###### 4.3.2.1 วิธีการทดลอง

เชื่อมต่อเว็บแอปพลิเคชันเข้ากับเซิร์ฟเวอร์ผ่านทาง Socket.IO เซิร์ฟเวอร์ส่งค่าอย่างต่อเนื่องมาให้เว็บแอปพลิเคชัน สร้างแผนภาพเส้นแสดงค่า 20 ค่าล่าสุดจากเซนเซอร์แต่ละค่าที่ส่งมาจากเซิร์ฟเวอร์

#### 4.3.2.2 ผลการทดลอง

สามารถแสดงค่าได้ตรงตามที่อ่านได้จากเซิร์ฟเวอร์และเปลี่ยนแปลงตามเวลา โดยมีความล่าช้าน้อยกว่าหนึ่งวินาที



รูป 4.8 การแสดงผลข้อมูลเป็นแผนภาพเส้นที่เปลี่ยนแปลงตามเวลา

#### 4.3.3 การทดลองบันทึกชุดข้อมูลของท่าต่าง ๆ และส่งออกเป็นไฟล์ .CSV

##### 4.3.3.1 วิธีการทดลอง

ในหน้านักพัฒนา สำหรับการบันทึกครั้ง ตั้งค่าชื่อท่าแล้วกดปุ่ม RECORD เพื่อเริ่มบันทึกค่าที่ส่งมาจากเซิร์ฟเวอร์ ระหว่างบันทึกให้ทำตามที่ต้องการ กดปุ่ม Stop เพื่อหยุดบันทึก จากนั้นกดปุ่ม Save เพื่อบันทึกท่า ทำแบบนี้จนครบ 50 ครั้ง เพื่อให้ได้ชุดข้อมูลจำนวนที่ขึ้นค่าสำหรับหนึ่งท่า

สำหรับการบันทึกต่อเนื่อง กดปุ่ม AUTO เพื่อเริ่มบันทึกค่าที่ส่งมาจากเซิร์ฟเวอร์ ทุก ๆ 20 ค่า ค่าถัดไปจะถูกตัดเป็นชุดข้อมูลใหม่ ระหว่างบันทึกให้ทำตามที่ต้องการ ทำจนครบอย่างน้อย 50 ครั้ง กดปุ่ม Stop เพื่อหยุดบันทึก จากนั้นกดปุ่ม Save เพื่อบันทึกท่า เพื่อให้ได้ชุดข้อมูลจำนวนที่ขึ้นค่าสำหรับหนึ่งท่า

จากนั้นกดปุ่ม Export เพื่อส่งออกชุดข้อมูลทั้งหมดออกเป็นไฟล์ export.csv

##### 4.3.3.2 ผลการทดลอง

ทั้งการบันทึกครั้งและบันทึกต่อเนื่อง สามารถบันทึกค่าลงเป็นไฟล์ csv และนำไปใช้ในกระบวนการฝึกโมเดล Machine Learning ได้อย่างไม่มีปัญหา

#### 4.3.4 การทดลองส่งข้อมูลล่าสุดไปตรวจจับท่าแล้วแสดงผลตอบกลับ

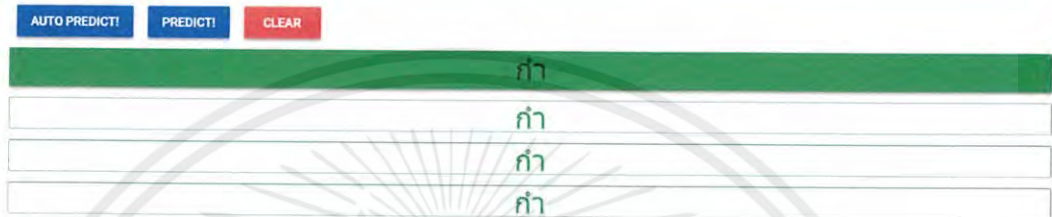
##### 4.3.4.1 วิธีการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้านักพัฒนา กดปุ่ม PREDICT เพื่อส่งข้อมูล 20 ค่าล่าสุดไปยังเซิร์ฟเวอร์ ผ่านทาง Socket.IO ทางฝ่ายเซิร์ฟเวอร์จำลองการส่งคำตอบกลับไปยังเว็บแอปพลิเคชัน แสดงผลตอบกลับที่ได้มา

#### 4.3.4.2 ผลการทดลอง

สามารถจำลองการส่งข้อมูลไปวิเคราะห์และแสดงผลตอบกลับได้ พร้อมทั้งจะใช้งานเพื่อแสดงผลคำตอบจริงในการทดลองถัดไป



รูป 4.9 การจำลองการแสดงผลการตรวจจับทำทาง

## 4.4 การทดลองเกี่ยวกับ Machine Learning

คือการนำข้อมูลที่ผ่านการปรับปรุงแล้วมาทดลองเกี่ยวกับโมเดล Machine Learning ต่าง ๆ ที่ใช้ในการวิเคราะห์ทำทางมือ โดยการทดลองอัลกอริธึมการสร้างโมเดลที่แตกต่างกัน หรือการใช้ค่าที่แตกต่างกัน

### 4.4.1 การทดลองโมเดลแบบที่ 1: การฝึกโมเดล

#### 4.4.1.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการส่วนของเว็บแอปพลิเคชัน แบ่งข้อมูลที่จะนำเข้าไปฝึกโดยใช้ข้อมูลร้อยละ 70 ของทั้งหมดในการฝึก ส่วนที่เหลือใช้ในการประเมินผลผลลัพธ์ของโมเดล (Evaluate) นำข้อมูลที่จะฝึกไปเปลี่ยนรูป (Transform) ให้ข้อมูลอยู่ในช่วงเดียวกัน หลังจากนั้นสร้างโมเดลขึ้นมา โดยใช้จำนวนครั้งในการฝึก (Epochs) 10,000 ครั้ง จึงนำโมเดลไปประเมินด้วยข้อมูลที่ไม่ได้นำไปฝึก

ถุงมือที่ใช้ในโมเดลนี้คือถุงมือแบบที่ 1 ทำที่ใช้ในการฝึกและทดสอบ โมเดลนี้ได้แก่ กำมือ, แบมือ, บอกรัก, ชูสองนิ้ว และไขว้นิ้ว

#### 4.4.1.2 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูงมาก ผลจากการประเมินสูงเกือบถึงร้อยละ 100 แต่ไม่สามารถทำนายท่าทางเคลื่อนไหวได้

### 4.4.2 การทดลองโมเดลแบบที่ 1: ทดสอบความแม่นยำ

#### 4.4.2.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัปเดตลงไปในไมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำท่าที่ต้องการทดสอบ กดปุ่มผ่านหน้าเว็บแอปพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำท่าหนึ่งทั้งหมด 50 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

#### 4.4.2.2 ผลการทดลอง

จากการนำไปทำนายท่าหนึ่งแต่ละท่า ทั้งหมด 50 ครั้ง ถูกต้อง 41 ครั้งหรือค่าความแม่นยำร้อยละ 82 เนื่องจากมีบางท่าที่เป็นข้อจำกัดของถุงมือแบบที่ 1 ทำให้การทำนายเกิดข้อผิดพลาด

ตาราง 4.1 ผลการทดสอบความแม่นยำโมเดลแบบที่ 1

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบมือ	10	10	100%
บอกรัก	10	10	100%
ชูสองนิ้ว	10	10	100%
ไขว้ นิ้ว	10	1	10%
รวม	50	41	82%

ตาราง 4.2 Confusion Matrix ของโมเดลแบบที่ 1

		ทำจริง					Precision
		กำมือ	แบมือ	บอกรัก	ชูสองนิ้ว	ไขว้ นิ้ว	
ท่าที่ตรวจจับได้	กำมือ	10	0	0	0	0	1
	แบมือ	0	10	0	0	0	1
	บอกรัก	0	0	10	0	0	1
	ชูสองนิ้ว	0	0	0	10	9	0.53
	ไขว้ นิ้ว	1	0	0	0	1	1

#### 4.4.3 การทดลองโมเดลแบบที่ 2: การฝึกโมเดล

##### 4.4.3.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการอัปเดตส่วนของเว็บแอปพลิเคชัน แบ่งข้อมูลที่น่าสนใจไปฝึกเป็นร้อยละ 70 นำข้อมูลไปเปลี่ยนรูป หลังจากนั้นสร้างโมเดลขึ้นมาโดยเปลี่ยนชั้นซ่อน (Hidden layer) เรื่อยๆ จนเจอค่าเหมาะสม ที่จะทำให้โมเดลสมบูรณ์มากขึ้น ปรับเปลี่ยนโมเดลไม่ให้ค่าฟิตเกินไปโดยใช้ Dropout ใช้จำนวนครั้งในการฝึกประมาณ 150 ครั้ง จึงนำโมเดลไปประเมิน

ถุมือที่ใช้ในโมเดลนี้คือถุมือแบบที่ 2 ท่าที่ใช้ในการฝึกและทดสอบโมเดลนี้ได้แก่ กำมือ, แบมือ, บอกรัก, ชูสองนิ้ว และไขว้ นิ้ว

##### 4.4.3.2 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูง ผลจากการประเมินสูงเกือบร้อยละ 100 เช่นเดียวกับโมเดลแบบแรก ไม่สามารถทำนายท่าเคลื่อนไหวได้เช่นเดียวกัน

#### 4.4.4 การทดลองโมเดลแบบที่ 2: ทดสอบความแม่นยำ

##### 4.4.4.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัปเดตลงไปยังบนไมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำหน้าที่ที่ต้องการทดสอบ กดปุ่มผ่านหน้าเว็บแอปพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำทำหนึ่งทั้งหมด 50 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

##### 4.4.4.2 ผลการทดลอง

จากการนำไปทำนายทำหนึ่งในแต่ละทำ ทั้งหมด 50 ครั้ง ถูกต้อง 48 ครั้งหรือค่าความแม่นยำร้อยละ 96 เนื่องจากการใช้ถุงมือแบบที่ 2 ลดข้อจำกัดด้านการงอนิ้วลง ทำให้ผลลัพธ์ที่ได้แม่นยำมากขึ้น

ตาราง 4.3 ผลการทดสอบความแม่นยำโมเดลแบบที่ 2

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบมือ	10	10	100%
บอกรัก	10	10	100%
ชูสองนิ้ว	10	10	100%
ไขว้ นิ้ว	10	8	80%
รวม	50	48	96%

ตาราง 4.4 Confusion Matrix ของโมเดลแบบที่ 2

		ทำจริง					Precision
		กำมือ	แบมือ	บอกรัก	ชูสองนิ้ว	ไขว้ นิ้ว	
ที่ทำที่ตรวจจับได้	กำมือ	10	0	0	0	0	1
	แบมือ	0	10	0	0	0	1
	บอกรัก	0	0	10	0	0	1
	ชูสองนิ้ว	0	0	0	10	2	0.83
	ไขว้ นิ้ว	0	0	0	0	8	1

#### 4.4.5 การทดลองโมเดลแบบที่ 3: การฝึกโมเดล

##### 4.4.5.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการอัปเดตส่วนของเว็บแอปพลิเคชัน ทำการแบ่งกลุ่มของข้อมูลเป็นลำดับของเวลา คือ ข้อมูลของมูมทั้งหมดในช่วงเวลาที่ 1 (เริ่มต้นทำ) ไปจนถึงข้อมูลของช่วงเวลาที่ 20 (สิ้นสุดทำ) นำข้อมูลมูมทั้ง 20 ช่วงเวลา ไปฝึกฝนเป็นทำหนึ่งทำ ใช้ Dropout ในการปรับไม่ให้โมเดลฟิตเกินไป จำนวนครั้งในการฝึกประมาณ 100 ครั้ง

คู่มือที่ใช้ในโมเดลนี้คือคู่มือแบบที่ 2 ทำที่ใช้ในการฝึกและทดสอบโมเดลนี้ได้แก่ กำมือ, แบนมือ, ชูสองนิ้ว, ไขว้ นิ้ว, พยัญชนะ ก, ข, ค และ ง

##### 4.4.5.1 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำ แต่เนื่องจากเป็นโมเดลที่ประมวลผลลำดับด้วยทำให้โมเดลมีความซับซ้อน ใช้เวลาในการฝึกค่อนข้างนานและจำเป็นต้องใช้ชุดข้อมูลจำนวนมาก เพื่อให้ได้ผลลัพธ์ในทุกๆกรณีที่จะเกิด

#### 4.4.6 การทดลองโมเดลแบบที่ 3: ทดสอบความแม่นยำ

##### 4.4.6.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัปโหลดลงไปในไมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำทำการที่ต้องการทดสอบ กดปุ่มผ่านหน้าเว็บแอปพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำทำหนึ่งและเคลื่อนไหวทั้งหมด 90 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

##### 4.4.6.2 ผลการทดลอง

จากการนำไปทำนายทำหนึ่งและท่าเคลื่อนไหวในแต่ละท่า ทั้งหมด 90 ครั้ง ถูกต้อง 70 ครั้ง หรือความแม่นยำร้อยละ 77.77 เนื่องจากมีท่าเคลื่อนไหวเพิ่มมา และทำหนึ่งที่ใกล้เคียงท่าเดิมมากขึ้น ทำให้เกิดข้อผิดพลาดมากที่สุด ใน 3 โมเดลที่ผ่านมา โดยเฉพาะกลุ่มท่า ชูสองนิ้ว, ไขว้ นิ้ว และพยัญชนะ ก

ตาราง 4.5 ผลการทดสอบความแม่นยำโมเดลแบบที่ 3

ข้อทำ	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบมือ	10	10	100%
บอกรัก	10	10	100%
ชูสองนิ้ว	10	9	90%
ไขว้ นิ้ว	10	4	40%
ก	10	5	50%
ข	10	8	80%
ค	10	6	60%
ง	10	8	80%
รวม	90	70	77.77%

ตาราง 4.6 Confusion Matrix ของโมเดลแบบที่ 3

		ทำจริง								Precision	
		กำมือ	แบมือ	บอกรัก	ชูสองนิ้ว	ไขว้ นิ้ว	ก	ข	ค		ง
ทำที่ตรวจจับได้	กำมือ	10	0	0	0	0	0	0	0	1	0.91
	แบมือ	0	10	0	0	0	0	0	0	0	1
	บอกรัก	0	0	10	0	0	0	0	0	0	1
	ชูสองนิ้ว	0	0	0	9	3	3	0	3	0	0.50
	ไขว้ นิ้ว	0	0	0	0	4	2	2	1	0	0.44
	ก	0	0	0	1	3	5	0	0	0	0.55
	ข	0	0	0	0	0	0	8	0	0	1
	ค	0	0	0	0	0	0	0	6	1	0.86
	ง	0	0	0	0	0	0	0	0	8	1

#### 4.4.7 การทดลองโมเดลแบบที่ 4: การฝึกโมเดล

##### 4.4.7.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการอัปเดตส่วนของเว็บแอปพลิเคชัน ทำการแบ่งกลุ่มของข้อมูลเป็นลำดับของเวลา คือ ข้อมูลดิบทั้งหมดในช่วงเวลาที่ 1 (เริ่มต้นทำ) ไปจนถึงข้อมูลของช่วงเวลาที่ 10 (สิ้นสุดทำ) นำข้อมูลดิบทั้ง 10 ช่วงเวลา ไปฝึกฝนเป็นทำหนึ่งทำ ใช้ Dropout ในการปรับไม่ให้โมเดลฟิตเกินไป จำนวนครั้งในการฝึกประมาณ 100 ครั้ง

ถุงมือที่ใช้ในโมเดลนี้คือถุงมือแบบที่ 2 ทำที่ใช้ในการฝึกและทดสอบ โมเดลนี้ได้แก่ กำมือ, แขนมือ, ชูสองนิ้ว, ไขว้ นิ้ว, พยัญชนะ ก, ข, ค และ ง เหมือนกับโมเดลแบบที่ 3

##### 4.4.7.1 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูง แต่เนื่องจากเป็นโมเดลที่ประมวลผลลำดับด้วย ทำให้โมเดลมีความซับซ้อน ใช้เวลาในการฝึกนานกว่าโมเดลแบบที่ 3 เพราะใช้ข้อมูลดิบในการประมวลผลซึ่งมีถึง 36 ลักษณะ

#### 4.4.8 การทดลองโมเดลแบบที่ 4: ทดสอบความแม่นยำ

##### 4.4.8.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัปเดตไปบนโมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำทำการที่ต้องการทดสอบ กดปุ่มผ่านหน้าเว็บแอปพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำทำหนึ่งและเคลื่อนไหวทั้งหมด 90 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

##### 4.4.8.2 ผลการทดลอง

จากการนำไปทำนายทำหนึ่งและทำเคลื่อนไหวในแต่ละทำ ทั้งหมด 90 ครั้ง ถูกต้อง 88 ครั้ง หรือความแม่นยำร้อยละ 97.77 ผลการทดสอบออกมาก่อนข้างคิเพราะว่าใช้ข้อมูลดิบในการฝึกฝน ตัวโมเดลสามารถรู้แนวโน้มของข้อมูลได้โดยไม่ต้องไปแปลงข้อมูลเป็นนาม ทำให้การทำนายมีความแม่นยำมากที่สุด

ตาราง 4.7 ผลการทดสอบความแม่นยำโมเดลแบบที่ 4

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบมือ	10	10	100%
บอกรัก	10	8	80%
ชูสองนิ้ว	10	10	100%
ไขว้ นิ้ว	10	10	100%
ก	10	10	100%
ข	10	10	100%
ค	10	10	100%
ง	10	10	100%
รวม	90	88	97.77%

ตาราง 4.8 Confusion Matrix ของโมเดลแบบที่ 4

		ทำจริง								Precision	
		กำมือ	แบมือ	บอกรัก	สองนิ้ว	ไขว้ นิ้ว	ก	ข	ค		ง
ท่าที่ตรวจพบได้	กำมือ	10	0	0	0	0	0	0	0	0	1
	แบมือ	0	10	0	0	0	0	0	0	0	1
	บอกรัก	0	0	8	0	0	0	0	0	0	1
	สองนิ้ว	0	0	0	10	0	0	0	0	0	1
	ไขว้ นิ้ว	0	0	0	0	10	0	0	0	0	1
	ก	0	0	0	0	0	10	0	0	0	1
	ข	0	0	2	0	0	0	10	0	0	0.83
	ค	0	0	0	0	0	0	0	10	0	1
	ง	0	0	0	0	0	0	0	0	10	1

#### 4.4.9 การทดลองโมเดลแบบที่ 4 โดยเพิ่มท่า: การฝึกโมเดล

##### 4.4.9.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการอัปเดตส่วนของเว็บแอปพลิเคชัน ทำการแบ่งกลุ่มของข้อมูลเป็นลำดับของเวลา คือ ข้อมูลดิบทั้งหมดในช่วงเวลาที่ 1 (เริ่มต้นท่า) ไปจนถึงข้อมูลของช่วงเวลาที่ 10 (สิ้นสุดท่า) นำข้อมูลดิบทั้ง 10 ช่วงเวลา ไปฝึกฝนเป็นท่าหนึ่งท่า ใช้ Dropout ในการปรับไม่ให้โมเดลฟิตเกินไป จำนวนครั้งในการฝึกประมาณ 200 ครั้ง โดยจะมีการเพิ่มท่าทางเข้ามาเป็น พญานาคภาษาไทยทั้งหมด 30 ท่า และท่าทั่วไปทั้งหมด 10 ท่า

##### 4.4.9.1 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูง แต่เนื่องจากเป็นโมเดลที่ประมวลผลลำดับด้วย ทำให้โมเดลมีความซับซ้อน ใช้เวลาในการฝึกนานกว่าการทดลองก่อนๆหลายเท่า เพราะจำนวนชุดข้อมูล (Dataset) เพิ่มมากขึ้น

#### 4.4.10 การทดลองโมเดลแบบที่ 4 โดยเพิ่มท่า: ทดสอบความแม่นยำ

##### 4.4.10.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัปเดตลงไปในไมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำท่าที่ต้องการทดสอบ กดปุ่มผ่านหน้าเว็บแอปพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองท่าทำหนึ่งและเคลื่อนไหวทั้งหมด 400 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

##### 4.4.10.2 ผลการทดลอง

จากการนำไปทำนายท่าหนึ่งและท่าเคลื่อนไหวในแต่ละท่า ทั้งหมด 400 ครั้ง ถูกต้อง 388 ครั้ง หรือความแม่นยำร้อยละ 97 ผลการทดสอบออกมาดีมากเนื่องจากการที่เลือกใช้โมเดลที่เหมาะสมที่สุดกับลักษณะข้อมูล ทำให้ได้โมเดลสุดท้ายที่มีท่ามากถึง 40 ท่า และยังคงความแม่นยำไว้

ตาราง 4.9 ผลการทดสอบความแม่นยำโมเดลแบบที่ 4 โดยเพิ่มท่าทาง

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
Open	10	10	100.00%
Close/สี	10	10	100.00%
ก	10	10	100.00%
ข	10	10	100.00%
ค	10	10	100.00%
ฅ	10	10	100.00%
ง	10	10	100.00%
จ	10	10	100.00%
ฉ	10	10	100.00%
ช	10	8	80.00%
ซ	10	10	100.00%
ญ	10	10	100.00%
ถ	10	8	80.00%
ด	10	9	90.00%
น	10	10	100.00%
ม	10	10	100.00%
ป	10	10	100.00%
ผ	10	10	100.00%
ฝ	10	10	100.00%
พ	10	10	100.00%
ฟ	10	10	100.00%
ภ	10	10	100.00%
บ	10	10	100.00%
ย	10	10	100.00%
ร	10	10	100.00%
ว	10	10	100.00%
ศ	10	10	100.00%
ษ	10	9	90.00%
ห	10	10	100.00%
ฬ	10	10	100.00%
อ	10	10	100.00%
โปกซ้าย	10	10	100.00%
โปกขวา	10	9	90.00%
สัดนิ้ว	10	9	90.00%
บอกรัก	10	8	80.00%
หุบนิ้วนาง	10	10	100.00%
หุบนิ้วกลาง	10	9	90.00%
ชูสองนิ้ว	10	10	100.00%
ชูนิ้วกลาง	10	10	100.00%
คมโต๊ะ	10	9	90.00%
รวม	400	388	97.00%

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 บทสรุป

เนื่องจากท่าทางมือที่เป็นตัวอักษรภาษาไทยนั้นมีท่าที่มากกว่าภาษาอังกฤษ จึงอาจจะทำให้การตรวจจับท่าทางเป็นไปได้ยาก และการสร้างถุงมือขึ้นมาเองนั้นอาจจะไม่คุ้มค่านำมาใช้เฉพาะสำหรับวัตถุประสงค์การแปลเพียงอย่างเดียว โครงการนี้จึงมีเป้าหมายเพื่อพัฒนาแพลตฟอร์มสำหรับตรวจจับท่าทางมือโดยใช้ Machine Learning เพื่อตรวจจับท่าทางมือที่เป็นตัวอักษรไทยทั้งทำนิ่งและเคลื่อนไหว รวมถึงสามารถตั้งค่าท่าทางเพิ่มเติมและการกระทำผ่านแอปพลิเคชันบนถุงมือ เพื่อให้ผู้ใช้สามารถนำไปประยุกต์ใช้ได้หลากหลายวัตถุประสงค์มากขึ้น โดยแพลตฟอร์มจะประกอบไปด้วย

##### 5.1.1 ส่วนประกอบของแพลตฟอร์ม

แพลตฟอร์มสำหรับตรวจจับท่าทางมือ สามารถแบ่งส่วนการทำงานออกเป็น 3 ส่วนได้ ดังนี้

###### 5.1.1.1 ส่วนฮาร์ดแวร์

ส่วนของถุงมือที่สวม วงจร เซนเซอร์ (Polulu MinIMU9-v5) จอแสดงผล (OLED) และไมโครคอนโทรลเลอร์ (Nanopi-DUO) การทำงานหลักคือ เซนเซอร์สร้างข้อมูลการเคลื่อนไหวของแต่ละนิ้วรวมถึงหลังมือ เพื่อนำไปประมวลผลในไมโครคอนโทรลเลอร์ ซึ่งเป็นเสมือนศูนย์กลางของส่วนอื่น ๆ บนแพลตฟอร์ม และเป็นส่วนแสดงผล

###### 5.1.1.2 ส่วนเซิร์ฟเวอร์

เป็นส่วนที่ทำงานอยู่ตลอดเวลาภายในระบบปฏิบัติการของไมโครคอนโทรลเลอร์ ประกอบด้วยส่วนคอนโทรลเลอร์สำหรับควบคุมการทำงาน ส่วนอ่านค่าจากเซนเซอร์ ส่วน Machine Learning สำหรับตรวจจับท่าทาง และเว็บเซิร์ฟเวอร์สำหรับบริการเว็บแอปพลิเคชันให้ผู้ใช้ โดยส่วนทั้งหมดทำงานพร้อมกันแบบมัลติเทรด

###### 5.1.1.3 ส่วนเว็บแอปพลิเคชัน

เป็นส่วนติดต่อผู้ใช้งาน สร้างขึ้นจากเว็บเซิร์ฟเวอร์ ผู้ใช้สามารถเข้าถึงได้ผ่านการเชื่อมต่อเข้ากับเครือข่ายของแพลตฟอร์ม แล้วเข้าไปยังไอพีแอดเดรสของแพลตฟอร์มผ่านเบราว์เซอร์ ผู้ใช้งานสามารถดูค่าจากเซนเซอร์แบบเรียลไทม์ ดูผลลัพธ์ ตั้งค่าท่าทาง การกระทำ ตั้งค่าการเชื่อมต่อเครือข่าย และบันทึกข้อมูลดิบของเซนเซอร์สำหรับนำไปฝึกโมเดล Machine Learning ได้ โดยส่วนนี้จะติดต่อกับเว็บเซิร์ฟเวอร์ตลอดเวลาผ่าน Socket.IO

## 5.1.2 การทดลอง

การทดลองของแพลตฟอร์มประกอบไปด้วย 4 หัวข้อ ได้แก่ การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ การทดลองเกี่ยวกับเว็บแอปพลิเคชัน และการทดลองเกี่ยวกับ Machine Learning

### 5.1.2.1 การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ

เป็นการทดลองเกี่ยวกับถุงมือที่ได้ออกแบบมาทั้งสองแบบ ว่าสามารถจับค่าได้แม่นยำและมีปัญหาในการจับค่าหรือไม่ จากผลการทดลองพบว่า ถุงมือแบบที่สองที่ใช้เซนเซอร์วัดอาศัยหลักความเฉื่อย (IMU) สามารถให้ค่าที่หลากหลายและแม่นยำมากกว่าถุงมือแบบที่หนึ่งที่ใช้เซนเซอร์วัดความงอ (Flex) ที่ยังมีปัญหาเรื่องแกนที่ไม่สามารถจับค่าได้ จึงใช้ถุงมือแบบที่สองเป็นหลักในการทำการทดลองถัดไป

### 5.1.2.2 การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ

เป็นการทดลองนำค่าโดยตรงจากถุงมือมาปรับปรุงให้สามารถใช้งานได้จริง หรือประยุกต์ใช้ในการหาค่าใหม่ที่มีประโยชน์มากขึ้นก่อนนำมาวิเคราะห์ด้วย Machine Learning โดยใช้วิธีต่าง ๆ จากการทดลองพบว่า การปรับค่าที่เหมาะสมที่สุดคือการใช้ข้อมูลดิบจาก IMU ที่ประกอบไปด้วยความเร่งเชิงเส้น 3 แกน และความเร็วเชิงมุม 3 แกน และเริ่มมีการตรวจจับการเคลื่อนไหวหรือหยุดนิ่งเมื่อใช้ค่าการเปลี่ยนแปลงมุมขั้นต่ำ 10 องศา/วินาที ในแกนใดแกนหนึ่งของเซนเซอร์วัดความเร็วเชิงมุม

### 5.1.2.3 การทดลองเกี่ยวกับเว็บแอปพลิเคชัน

เป็นการทดลองพัฒนาเว็บแอปพลิเคชันในส่วนต่าง ๆ รวมถึงการเชื่อมต่อแลกเปลี่ยนข้อมูลกับเซิร์ฟเวอร์ เพื่อนำข้อมูลจากเซิร์ฟเวอร์มาประยุกต์ใช้งานเกี่ยวกับ Machine Learning หรือขั้นตอนอื่น ๆ ของโครงการต่อไป จากการทดลอง เว็บแอปพลิเคชันสามารถเชื่อมต่อเซิร์ฟเวอร์ได้ และสามารถใช้งานได้ครบทุกคุณลักษณะ

### 5.1.2.4 การทดลองเกี่ยวกับ Machine Learning

เป็นการนำข้อมูลที่ผ่านการปรับปรุงแล้วมาทดลองเกี่ยวกับโมเดล Machine Learning ต่าง ๆ ที่ใช้ในการวิเคราะห์ท่าทางมือ โดยการทดลองอัลกอริธึมการสร้างโมเดลที่แตกต่างกัน หรือการใช้ค่าที่แตกต่างกัน จากผลลัพธ์การทดลองพบว่า การใช้โมเดล LSTM โดยมีข้อมูลที่ใช้ฝึกเป็นข้อมูลดิบที่มีค่า Timestep = 10 จำนวน 7636 ชุด มีความแม่นยำที่สูงที่สุด เฉลี่ยทุกท่าที่ 97%

## 5.2 ปัญหาและอุปสรรค

ปัญหาที่เกิดขึ้นระหว่างดำเนินการโครงการ แบ่งตามส่วนต่าง ๆ ของแพลตฟอร์มได้ดังนี้

### 5.2.1 ส่วนฮาร์ดแวร์

- อุปกรณ์บางชิ้นจำเป็นต้องสั่งมาจากต่างประเทศ อาจทำให้เกิดความล่าช้าในการทำงาน
- การบัดกรีวงจรที่มีขนาดเล็กและซับซ้อน ทำให้เสียเวลานานในการดำเนินงาน และมีความเสี่ยงที่จะทำให้อุปกรณ์เสียหาย
- เมื่อเชื่อมต่อเข้าเครือข่ายอื่นแล้วไม่ได้รับไอพีแอดเดรส หรือเชื่อมต่อช้า หรือหลุดการเชื่อมต่อบ่อยครั้ง อาจทำให้ตัวจัดการเครือข่ายในไมโครคอนโทรลเลอร์สับสน และไม่ปล่อยสัญญาณให้เชื่อมต่อเข้าไปแก้ไขปัญหา

#### 5.2.2 ส่วนคอนโทรลเลอร์

- เมื่อเริ่มต้นการทำงาน การเรียกใช้ไลบรารี TensorFlow รวมถึงโมเดลนั้นใช้เวลานาน และไม่สามารถทำงานในอีกเซตได้ จึงทำให้การทดสอบกระบวนการทำงานหลังจากการแก้ไขต่าง ๆ นั้นช้า

#### 5.2.3 ส่วน Machine Learning

- เนื่องจากเราเลือกใช้ Deep Learning ในการตรวจจับท่าทาง จำนวนของชุดข้อมูล (Dataset) จึงต้องมีมากพอที่จะทำให้ตัวโมเดลสามารถเรียนรู้ได้ด้วยตัวเอง
- ท่าทางตัวละครคนไทยมีบางท่าที่ออกท่าทางคล้ายกัน ทำให้ตัวโมเดลสับสนอยู่บ่อยครั้ง
- การติดตั้งไลบรารี TensorFlow ที่ใช้เป็นส่วนประมวลผลหลัก ลงบอร์ดนั้นทำได้ยาก และใช้เวลานาน เนื่องจากใช้บอร์ดที่ไม่นิยม

#### 5.2.4 ส่วนเว็บแอปพลิเคชัน

- ส่วนแสดงค่าข้อมูลจากเซนเซอร์ หากเปิดแสดงหลาย ๆ ตัวพร้อมกัน จะทำให้เว็บแอปพลิเคชันทำงานช้าหรือค้าง จึงไม่สามารถตรวจสอบค่าของเซนเซอร์ทุกนิ้วพร้อมกันได้

### 5.3 แนวทางแก้ไขปัญหา

#### 5.3.1 ส่วนฮาร์ดแวร์

- เขียนการทำงานของคอนโทรลเลอร์ขึ้นมาอีกชุดเพื่อทำงานในกรณีที่ขาดอุปกรณ์บางตัว หรือกรณีที่อุปกรณ์บางตัวชำรุดระหว่างการทำงาน
- ออกแบบและกัดแผ่นวงจรแบบ PCB แทนการบัดกรีด้วยมือ และใช้ Socket เพื่อเสียบอุปกรณ์แทนการเชื่อมต่อกับวงจรโดยตรง
- เพิ่ม Socket สำหรับติดต่อไมโครคอนโทรลเลอร์ผ่าน Serial Debugging

#### 5.3.2 ส่วนเซิร์ฟเวอร์

- เขียนการทำงานในโหมด DEBUG เพื่อทดสอบการกระบวนการทำงาน โดยเฉพาะที่ไม่จำเป็นต้องเรียกใช้ Machine Learning

### 5.3.3 ส่วน Machine Learning

- เพิ่มชุดข้อมูล (Dataset) เข้าไปให้เพียงพอสำหรับการเรียนรู้ และไม่มากไปจนทำให้โมเดลเข้าใจผิดเป็นทำอื่น โดยจะต้องเกลี่ยจำนวนข้อมูลของแต่ละทำให้ใกล้เคียงกัน
- นำโครงสร้างของบอร์ดไปเปรียบเทียบกับบอร์ดที่คนนิยมใช้ จึงสามารถหาตัวติดตั้งไลบรารีสำเร็จรูปที่ผ่านการสร้างมาได้

### 5.3.4 ส่วนเว็บแอปพลิเคชัน

- เปิดดูค่าของเซนเซอร์เท่าที่จำเป็น หรือบันทึกค่าไว้แล้วกลับมาดูภายหลัง

## 5.4 แนวทางการพัฒนา

แนวทางการพัฒนาแพลตฟอร์ม สามารถแบ่งไปตามส่วนต่าง ๆ ของแพลตฟอร์มได้ดังนี้

### 5.3.1 ส่วนฮาร์ดแวร์

- สามารถหาอุปกรณ์ที่คุณภาพดีกว่า หรือมีขนาดเล็กกว่ามาใช้แทนได้ เนื่องจากส่วนต่างๆ ของฮาร์ดแวร์ เช่น เซนเซอร์ ไมโครคอนโทรลเลอร์ ได้ออกแบบมาให้สามารถถอดเปลี่ยนได้ง่าย หากมีอุปกรณ์ที่ดีกว่า สามารถถอดเปลี่ยนได้ง่าย
- ตัวถูงมือยังมีขนาดใหญ่ และอาจลำบากในการใช้งานจริง จึงสามารถออกแบบและพัฒนารูปลักษณะภายนอกให้ใช้ได้จริงมากขึ้น
- ถูงมือสามารถพัฒนาเพิ่มส่วนแบตเตอรี่ และจ่ายไฟ รวมถึงทำให้สามารถเปิด-ปิดได้ตลอดเวลา
- เพิ่มลำโพง เพื่อการใช้งานที่หลากหลายมากขึ้น

### 5.3.2 ส่วนคอนโทรลเลอร์

- พัฒนาเพิ่มการกระทำให้มากขึ้น เช่น สามารถควบคุมค่าที่จะพุดผ่านลำโพง หรือตั้งค่าการควบคุมอุปกรณ์ IoT เฉพาะยี่ห้อที่ขอดนิยมได้โดยตรง

### 5.3.3 ส่วน Machine Learning

- เพิ่มคำพุดในภาษามือหรือตัวสะกดภาษาไทยให้มากขึ้น เพียงพอที่จะสามารถผสมคำเป็นประโยคขึ้นได้ เพื่อจะนำไปสื่อสารกับผู้พิการได้จริง

### 5.3.4 ส่วนแอปพลิเคชัน

- สร้างแอปพลิเคชันสำหรับคอมพิวเตอร์หรือโทรศัพท์โดยตรง ทำให้สามารถติดต่อกับส่วนต่าง ๆ ที่ต้องการคำอนุญาตของอุปกรณ์ได้ เพื่อนำมาประยุกต์ใช้งานต่อ เช่น การบังคับเชื่อมต่อกับถูงมืออย่างรวดเร็ว การนำข้อมูลสถานที่จากโทรศัพท์มาแสดงผล เป็นต้น

## บรรณานุกรม

- [1] Felipe, A. Q. Marcelo, R. Rafael, R. D. Enzo, H. W and Alessandro, G. “**A Gesture Detection Glove for Human-Computer Interaction.**” Federal University of Pampa, Alegrete - RS, Brazil
- [2] Rung-Huei, L. Ming, O. “**A real-time continuous gesture recognition system for sign language.**” Dept. of Inf. Manage., Shieh Chen Univ., Taipei, Taiwan
- [3] Ji-Hwan, K. Nguyen, D. T. Tae-Seong, K. “**3-D hand motion tracking and gesture recognition using a data glove.**” Department of Biomedical Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, 446-701 Gyeonggi-do, Korea
- [4] FriendlyARM. 2018. **NanoPI-DUO Hardware spec.** [Online] Available: [http://wiki.friendlyarm.com/wiki/index.php/NanoPi\\_Duo#Hardware\\_Spec](http://wiki.friendlyarm.com/wiki/index.php/NanoPi_Duo#Hardware_Spec)
- [5] Polulu. 2016. **MinIMU-9 v5 Gyro, Accelerometer, and Compass.** [Online] Available: <https://www.pololu.com/product/2738>
- [6] Doratong24. 2017. **LSTM.** [Online] Available: <https://medium.com/@tongkornkitt/a3a55cd37883>
- [7] Arun Gupta. 2017. **REST vs WebSocket Comparison and Benchmarks.** [Online] Available: <http://blog.arungupta.me/rest-vs-websocket-comparison-benchmarks/>
- [8] Hobbytronics. **Measuring Tilt Angle using Three Axis.** [Online] Available: <http://www.hobbytronics.co.uk/accelerometer-info>
- [9] Bayesian Adventures. 2013. **Gyroscopes, Accelerometers and the Complementary Filter.** [Online] Available: <https://bayesianadventures.wordpress.com/2013/10/20/gyroscopes-accelerometers-and-the-complementary-filter/>

[10] GoomGum. 2016. **Kalman filter**. [Online] Available:

<https://goomgum.wordpress.com/2016/07/07/>



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้