

ระบบสั่งการด้วยเสียง  
VOICE CONTROL SYSTEM



โดย  
นายพลชรร  
นายพิชฌุตม์  
นางสาวภาณุมาศ  
ณอมงาม  
สุขสวัสดิ์  
สาทิพย์จันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสั่งการด้วยเสียง  
VOICE CONTROL SYSTEM

โดย

นายพลชรร	ถนอมงาม	61010704
นายพิชณุตม์	สุขสวัสดิ์	61010743
นางสาวภาณุมาศ	สาทิพย์จันทร์	61010807

อาจารย์ที่ปรึกษา

ดร.สมปอง วิเศษพานิชกิจ  
ผศ. ดร.ธเนศ พัฒนธาดาพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสั่งการด้วยเสียง

VOICE CONTROL SYSTEM

ผู้จัดทำ

- |                  |               |          |
|------------------|---------------|----------|
| 1. นายพลชธร      | ณอมงาม        | 61010704 |
| 2. นายพิชญุตม์   | สุขสวัสดิ์    | 61010743 |
| 3. นางสาวภาณุมาศ | สาทิพย์จันทร์ | 61010807 |



อาจารย์ที่ปรึกษา

(ดร.สมปอง วิเศษพานิชกิจ)



อาจารย์ที่ปรึกษาร่วม

(ผศ. ดร.ธเนศ พัฒนธาดาทพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาานิพนธ์เรื่อง “ระบบสังการด้วยเสียง” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความอนุเคราะห์และสนับสนุนอย่างดียิ่งจากอาจารย์ที่ปรึกษาหลัก ดร.สมปอง วิเศษพานิชกิจ อาจารย์ที่ปรึกษาร่วม ผศ. ดร.ธเนศ พัฒนธาดาพงษ์ และผศ. ดร. นภัทร สระเอี่ยม ที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกา ค้นคว้าวิจัยให้ปริญญาานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงฝึกฝนให้ผู้จัดทำมีทักษะทางด้านความคิด การอ่าน การเขียน และการนำเสนอผลงานวิชาการ ขอขอบพระคุณความหวังใย และความปรารถนาดีให้แก่ผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอขอบคุณเพื่อนร่วมงานและรุ่นพี่ทุกท่านในห้องปฏิบัติการทดลองภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่กรุณาให้คำแนะนำ คำปรึกษา และถ่ายทอดวิชาความรู้ให้แก่ผู้จัดทำ

สุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความหวังใย และเป็นกำลังใจที่สำคัญเสมอมาในยามที่เกิดปัญหา จนกระทั่งปริญญาานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดีและที่สำคัญคือสนับสนุนให้โอกาสทางการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำ

นายพลชธร	ธนอมงาม
นายพิชญุตม์	สุขสวัสดิ์
นางสาวภาณุมาศ	สาทิพย์จันทร์
	ผู้จัดทำ

ระบบสั่งการด้วยเสียง  
VOICE CONTROL SYSTEM

โดย	นายพลชจร	ถนอมงาม	61010722
	นายพิชฌุต์	สุขสวัสดิ์	61010743
	นางสาวภาณุมาศ	สาทิพย์จันทร์	61010807

อาจารย์ที่ปรึกษา ดร.สมปอง วิเศษพานิชกิจ  
อาจารย์ที่ปรึกษาร่วม ผศ. ดร.ธเนศ พัฒนธาดาพงษ์

#### บทคัดย่อ

ปัจจุบันเทคโนโลยีนับเป็นส่วนหนึ่งในชีวิตประจำวัน มนุษย์หันมาสนใจเทคโนโลยีที่ช่วยอำนวยความสะดวกสบายมากขึ้น จากผลสำรวจข้อมูลผู้บริโภคของ EIC (Economic Intelligence Center) พบว่าเทคโนโลยีสมาร์ทโฮมกลายเป็นปัจจัยที่สำคัญมากขึ้น พบว่าราว 77% ของผู้ตอบแบบสำรวจอยากให้มระบบเตือนอัจฉริยะภายในบ้าน ในขณะที่ราว 73% ต้องการให้มีระบบช่วยควบคุมอุปกรณ์ไฟฟ้า และจัดการพลังงานภายในบ้านเพื่อช่วยอำนวยความสะดวกสบาย

ปัญญานិพนธ์นี้จึงมีวัตถุประสงค์เพื่อนำเทคโนโลยีการรู้จำเสียงมาประยุกต์ใช้ในการออกแบบระบบสั่งการด้วยเสียง ซึ่งเป็นระบบที่รับอินพุตได้ก็ต่อเมื่อได้รับคำกระตุ้น และเป็นเสียงภาษาไทยที่เป็นคำสั่งในการควบคุมอุปกรณ์ไฟฟ้า โดยจะแบ่งการออกแบบระบบเป็น 2 ส่วน ประกอบด้วยส่วนปัญญาประดิษฐ์ (AI) โดยใช้ข้อมูลเสียงจากผู้จัดทำนำไปเตรียมข้อมูลเปลี่ยนจากข้อมูลเสียงเป็นข้อมูลที่อยู่ในรูปของสัมประสิทธิ์เซปตรัมความถี่เมล (MFCC) และนำข้อมูลไปฝึกฝนโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน (CNN) และโครงข่ายประสาทเทียมแบบวนซ้ำ (RNN) ซึ่งการเลือกโมเดลมีรายละเอียดในการพิจารณา 2 ส่วน ได้แก่ ค่าความผิดพลาด (Loss) และค่าความแม่นยำ (Accuracy) โดยจะทำการเลือกโมเดลที่มีค่าความผิดพลาดต่ำที่สุดและค่าความแม่นยำมากที่สุดในการนำโมเดลไปประยุกต์ใช้งานในส่วนการควบคุมอุปกรณ์ไฟฟ้า จะรับอินพุตเสียงเพื่อนำไปประมวลผลคำสั่งผ่าน raspberry pi และใช้โมดูลอินฟราเรดส่งสัญญาณไปควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSTRACT

Nowadays, Technology is part of our daily life. People are increasingly interested in technologies that make them more comfortable. Economic Intelligence Center (EIC) consumer data survey found that Smart Home technology is becoming an increasingly important factor, finding that about 77% of respondents would like a Smart Home alarm system, while about 73% would like it to be controlled electrical equipment and manage the energy in the home to help comfort.

This thesis aims to apply speech recognition technology in making a voice command system to control devices through voice. The system accepts input only when receiving a trigger word and controls electrical equipment in the Thai language. It's divided into two parts: Artificial Intelligence (AI) part. Use the audio data from the authors to prepare the data to change from the audio data to the data in the form of Mel-Frequency Spectrum Coefficient (MFCC) and train the data using Convolutional Neural Networks (CNN) and Recursive Neural Networks (RNN). When choosing a model, two parts need to be considered, loss and accuracy. The model with the lowest error and the highest accuracy is selected to apply to control electrical equipment. It receives audio inputs to process commands through the raspberry pi and uses the infrared module to send signals to control electrical devices.

## สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	IV
สารบัญตาราง	IV
<b>บทที่ 1</b>	
<b>บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
<b>บทที่ 2</b>	
<b>ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	<b>2</b>
2.1 การรู้จำคำพูด	2
2.2 ปัญญาประดิษฐ์ (Artificial Intelligence: AI)	2
2.3 การเรียนรู้ของเครื่อง (Machine Learning: ML)	3
2.4 หลักการของ Neural Network	3
2.4.1 Feed forward propagation	5
2.4.2 Back propagation	8
2.4.3 ความผิดพลาดในการฝึกฝนโมเดล	10
2.5 กระบวนการสกัดคุณลักษณะเด่น (Feature Extraction)	11
2.6 Convolution Neural Network (CNN)	13
2.6.1 Convolution Layer	13

## สารบัญ (ต่อ)

	หน้า
2.6.2 Pooling Layer	14
2.6.3 Flatten Layer	15
2.6.4 Fully Connected Layer	16
2.7 Long Short-Term Memory (LSTM)	16
2.7.1. Forget Gate Layer	16
2.7.2 Input Gate Layer	17
2.7.3 Output Gate Layer	17
2.8 หลักการในการตรวจจับการเกิดปัญหา Overfit	18
2.9 Library Python ที่เกี่ยวข้องกับงาน	20
2.9.1 ไลบรารี OS	20
2.9.2 ไลบรารี Numpy	20
2.9.3 ไลบรารี Matplotlib	20
2.9.4 ไลบรารี Time	20
2.9.5 Tensorflow	20
2.9.6 Librosa	20
2.9.7 Pyaudio	20
2.9.8 Linux Infrared Remote Control (LIRC)	20
2.10 ภาษาไพธอน (Python)	21
2.11 ระบบปฏิบัติการ Linux	21
2.12 โมดูล Raspberry Pi 4 model B	22
2.13 โมดูลส่งสัญญาณอินฟราเรด KY-005	23
2.14 โมดูลรับสัญญาณอินฟราเรด KY-022	24
2.15 สัญญาณอินฟราเรด	24

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 3</b>	<b>25</b>
<b>การออกแบบและการจัดทำปฏิญานินทร์</b>	
3.1 การออกแบบ	
3.1.1 การออกแบบโปรแกรมการจัดเก็บข้อมูลเสียง	27
3.1.2 การออกแบบการสกัดคุณลักษณะเด่นของเสียง (Feature Extraction)	28
3.1.3 การเตรียมข้อมูล (Prepare Data)	28
3.1.4 การออกแบบกระบวนการฝึกฝน (Training)	30
3.1.5 การออกแบบการทดสอบการใช้งานโมเดล	35
3.1.6 การออกแบบการใช้งานโมเดลในระบบสั่งการด้วยเสียง	37
3.2 เครื่องมือที่ใช้ในการทดลอง	42
3.3 การจัดเก็บผลการทดลอง	46
3.3.1 การจัดเก็บข้อมูลเสียง	46
3.3.2 การจัดเก็บผลที่ได้จากการสกัดคุณลักษณะเด่น	46
3.3.3 การจัดเก็บผลการทดสอบประสิทธิภาพของแบบจำลอง	47
3.3.4 การจัดเก็บผลการทดสอบการทำงานของโมเดล	56
3.3.5 การจัดเก็บผลการทดสอบการใช้งานโมเดลในระบบสั่ง การด้วยเสียง	57

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 4</b>	
<b>ผลการทดลอง</b>	<b>58</b>
4.1 ผลการเปลี่ยนระดับข้อมูลเสียง (Pitch)	58
4.2 ผลการทดสอบประสิทธิภาพแบบจำลอง	60
4.3 ผลการทดสอบประสิทธิภาพแบบจำลอง	66
4.4 ผลการทดสอบประสิทธิภาพของโมเดล	70
4.5 ผลการทดสอบประสิทธิภาพของระบบสั่งการด้วยเสียงหลังตั้งค่า เงื่อนไขการตัดสินใจ (Threshold) ให้กับระบบ	81
4.6 ผลการทดสอบประสิทธิภาพการใช้งานระบบสั่งการด้วยเสียง	84
<b>บทที่ 5</b>	
<b>สรุปผลและข้อเสนอแนะ</b>	<b>88</b>
5.1 สรุปผล	88
5.2 ข้อเสนอแนะ	88

## สารบัญรูป

รูปที่	หน้า
2.1	4
2.2	5
2.3	6
2.4	6
2.5	7
2.6	7
2.7	8
2.8	9
2.9	9
2.10	11
2.11	13
2.12	13
2.13	14
2.14	14
2.15	15
2.16	15
2.17	16
2.18	17
2.19	17

## สารบัญรูป (ต่อ)

รูปที่	หน้า	
2.20	โครงสร้าง Output Gate Layer	18
2.21	การแบ่งข้อมูลอินพุตให้ออกเป็นข้อมูล 2 ชุด	18
2.22	ความสัมพันธ์ระหว่างค่าความผิดพลาดกับรอบที่ทำการฝึก	19
2.23	หลักการ Dropout	19
2.24	รายละเอียดของ Raspberry Pi 4 model B	22
2.25	พอร์ต GPIO ของ Raspberry Pi 4 model B	23
2.26	โมดูลส่งสัญญาณอินฟราเรด KY-005	23
2.27	โมดูลตัวรับอินฟราเรด KY-022	24
3.1	ภาพรวมของโครงงานระบบสั่งการด้วยเสียง	25
3.2	ภาพรวมของโครงงานในส่วนการสร้างโมเดล	26
3.3	หลักการเก็บข้อมูลเสียง	27
3.4	ป้ายกำกับคำสั่งของโมเดลส่วนกระตุ้นการทำงานของระบบป้ายกำกับคำสั่ง	29
3.5	ป้ายกำกับคำสั่งของโมเดลส่วนควบคุมหลอดไฟ	29
3.6	ป้ายกำกับคำสั่งของโมเดลส่วนควบคุมเครื่องปรับอากาศ	29
3.7	ภาพรวมการทำงานของ Neural Network	31
3.8	ผังการทำงานแบบจำลอง CNN	32
3.9	ผังการทำงานแบบจำลอง LSTM	33
3.10	ผังการทำงานแบบจำลอง CNN+LSTM	34
3.11	หลักการทำงานของการทดสอบการใช้โมเดลเปลี่ยนเสียงเป็นข้อความ (Speech to text)	36
3.12	บล็อกไดอะแกรมการตั้งค่าตัดสินใจ	37
3.13	วิธีการหาค่าตัดสินใจ	38

## สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.14	บล็อกไดอะแกรมแสดงระบบการทำงานของระบบสั่งการด้วยเสียง	39
3.15	ตัวอย่างชุดสัญญาณของการสั่งงานปิดไฟ ของหลอดไฟที่ทำการทดลอง ชุดคำสั่งของรีโมทควบคุม แอร์ และ หลอดไฟ	40
3.16	Laptop HP Pavilion x360	41
3.17	Lenovo Ideapad 320S	42
3.18	Desktop	42
3.19	Raspberry Pi 4 model B	43
3.20	Respeaker 4-mic array	43
3.21	IR Transmitter	44
3.22	IR Receiver	44
3.23	หลอดไฟ	44
3.24	เครื่องปรับอากาศ	46
3.25	ลำโพง	45
3.26	Data Frame ของไฟล์ข้อมูล mfcc	45
3.27	บล็อกไดอะแกรมแบบจำลอง Convolution neural network	47
3.28	(CNN) บล็อกไดอะแกรมแสดง Activating function ของแบบจำลอง CNN	49
3.29	บล็อกไดอะแกรมแบบจำลอง Long-short term memory (LSTM)	50
3.30	บล็อกไดอะแกรมแสดง Activating function ของแบบจำลอง LSTM	51
3.31	บล็อกไดอะแกรมแบบจำลอง CNN ประเภทอินพุตมีหลายภาพ	51
3.32	บล็อกไดอะแกรมของแบบจำลอง CNN ที่เป็นอินพุตชั่วคราว	52
3.33		54

## สารบัญรูป (ต่อ)

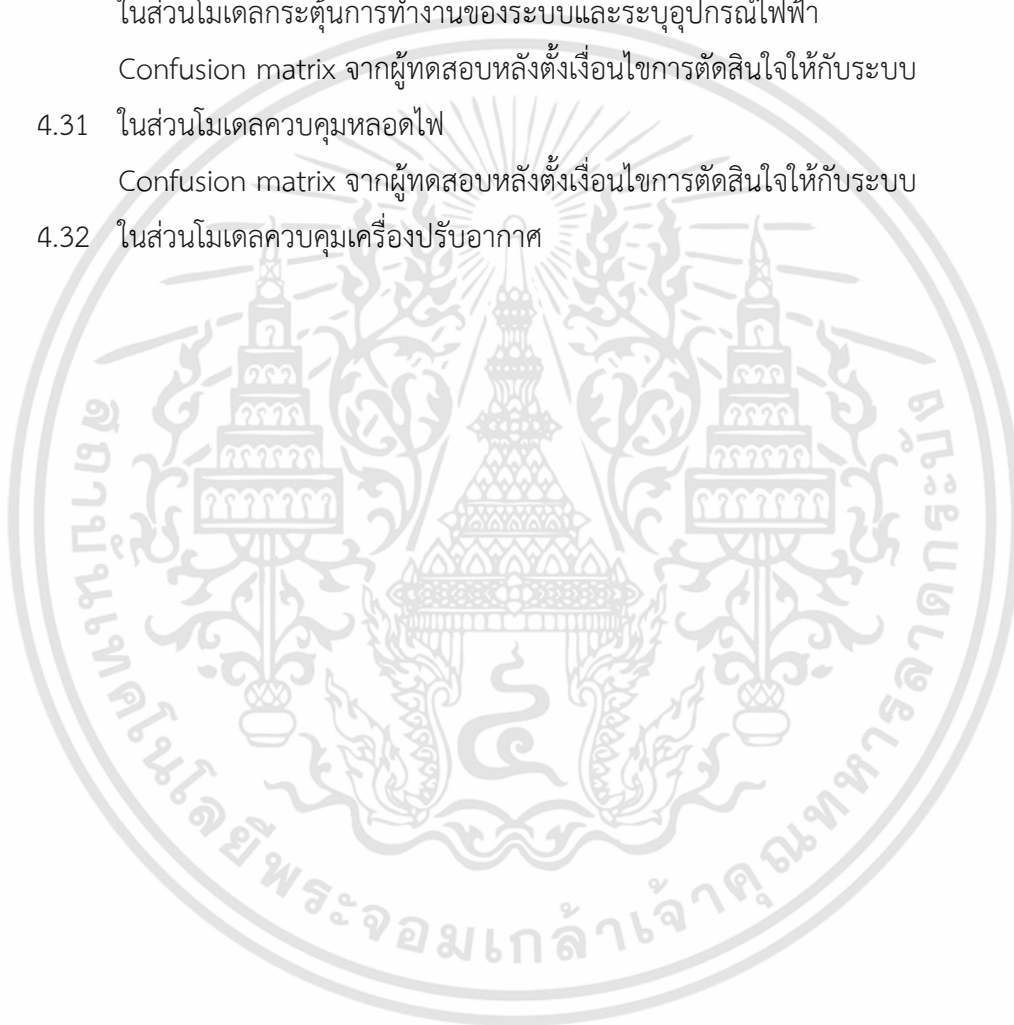
รูปที่	หน้า	
3.34	บล็อกไดอะแกรมของแบบจำลอง CNN+LSTM	55
3.35	บล็อกไดอะแกรมแสดง Activating function ของแบบจำลอง CNN+LSTM	56
4.1	ข้อมูลเสียงที่ได้จากการบันทึกจากโปรแกรม	58
4.2	ตัวอย่างข้อมูลเสียงที่ถูกลดความเร็ว	58
4.3	ข้อมูลเสียงที่ถูกเพิ่มความเร็ว	59
4.4	ข้อมูลเสียงที่ถูกเพิ่มระดับเสียง	59
4.5	ข้อมูลเสียงที่ถูกลดระดับเสียง	60
4.6	แสดงกราฟเปรียบเทียบค่า Loss Function ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของโครงสร้าง CNN	66
4.7	แสดงกราฟเปรียบเทียบค่า Loss Function ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของโครงสร้าง LSTM	66
4.8	แสดงกราฟเปรียบเทียบค่าความผิดพลาดระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง CNN+LSTM	67
4.9	แสดงกราฟเปรียบเทียบค่า Accuracy ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง CNN	67
4.10	แสดงกราฟเปรียบเทียบค่า Accuracy ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง LSTM	68
4.11	แสดงกราฟเปรียบเทียบค่าความแม่นยำระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง CNN+LSTM	68
4.12	Confusion matrix จากชุดข้อมูลทดสอบ ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า	70

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.13 Confusion matrix จากชุดข้อมูลทดสอบ ในส่วนโมเดลควบคุมหลอดไฟ	71
4.14 Confusion matrix จากชุดข้อมูลทดสอบ ในส่วนโมเดลควบคุมเครื่องปรับอากาศ	71
4.15 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า	73
4.16 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ ในส่วนโมเดลควบคุมหลอดไฟ	73
4.17 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ ในส่วนโมเดลควบคุมเครื่องปรับอากาศ	74
4.18 กราฟ Normal distribution ของชุดคำสั่งโทรคม	75
4.19 กราฟ Normal distribution ของชุดคำสั่งแอร์	76
4.20 กราฟ Normal distribution ของชุดคำสั่งหลอดไฟ	76
4.21 กราฟ Normal distribution ของชุดคำสั่งปิดไฟ	76
4.22 กราฟ Normal distribution ของชุดคำสั่งเปิดไฟ	77
4.23 กราฟ Normal distribution ของชุดคำสั่งสว่างมาก	77
4.24 กราฟ Normal distribution ของชุดคำสั่งสว่างน้อย	77
4.25 กราฟ Normal distribution ของชุดคำสั่งปิดแอร์	78
4.26 กราฟ Normal distribution ของชุดคำสั่งเปิดแอร์	78
4.27 กราฟ Normal distribution ของชุดคำสั่งลดอุณหภูมิ	79
4.28 กราฟ Normal distribution ของชุดคำสั่งเพิ่มอุณหภูมิ	79
4.29 กราฟ Normal distribution ของชุดคำสั่งเปลี่ยนโหมด	79

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.30	82
Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า	
Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ	
4.31	82
ในส่วนโมเดลควบคุมหลอดไฟ	
Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ	
4.32	83
ในส่วนโมเดลควบคุมเครื่องปรับอากาศ	



## สารบัญตาราง

ตารางที่	หน้า
4.1 ผลการฝึกฝนข้อมูลของแบบจำลอง Convolution Neural Network (CNN) ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์	61
4.2 ผลการฝึกฝนข้อมูลของแบบจำลอง Long-Shot Term Memory (LSTM) ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์	63
4.3 รูปแบบที่ 1 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM	64
4.4 รูปแบบที่ 2 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM	64
4.5 รูปแบบที่ 2 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM	64
4.6 รูปแบบที่ 3 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM	65
4.7 ผลการฝึกฝนข้อมูลของแบบจำลอง CNN+LSTM ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์	65
4.8 ผลการคำนวณค่าความผิดพลาดของแบบจำลอง	69
4.9 ผลการคำนวณความแม่นยำของแบบจำลอง	69
4.10 ความแม่นยำของโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้าของผู้จัดทำ	85
4.11 ความแม่นยำของโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้าของผู้ทดสอบ	85
4.12 ความแม่นยำของโมเดลในการควบคุมหลอดไฟของผู้จัดทำ	85
4.13 ความแม่นยำของโมเดลในการควบคุมหลอดไฟของผู้จัดทำของผู้ทดสอบ	86
4.14 ความแม่นยำของโมเดลในการควบคุมเครื่องปรับอากาศของผู้จัดทำ	86
4.15 ความแม่นยำของโมเดลในการควบคุมเครื่องปรับอากาศของผู้จัดทำของผู้ทดสอบ	86



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการใช้เทคโนโลยีนับเป็นส่วนหนึ่งในชีวิตประจำวัน ไม่ว่าจะเป็นโทรศัพท์มือถือ คอมพิวเตอร์ และสิ่งอำนวยความสะดวกต่าง ๆ ซึ่งเครื่องมืออำนวยความสะดวกเหล่านี้ต่างมีจุดประสงค์เพื่ออำนวยความสะดวกสบายให้มนุษย์ และยกระดับคุณภาพชีวิตให้ดีขึ้น โดยเทคโนโลยีที่กำลังได้รับความนิยมอย่างมากในปัจจุบัน คือ ระบบสั่งการด้วยเสียง (Voice Control System) เป็นระบบที่ให้ผู้ใช้งานมีการสื่อสารกับระบบผ่านเสียงหรือคำพูด ระบบสั่งการด้วยเสียงที่ได้รับความนิยมอย่างมากในปัจจุบัน ยกตัวอย่างเช่น Siri, Google Assistant และ Amazon Alexa

ผู้จัดทำจึงมีแนวคิดในการพัฒนาระบบสั่งการด้วยเสียง ซึ่งมีวัตถุประสงค์การจัดทำเพื่อนำเทคโนโลยีการรู้จำเสียงมาประยุกต์ใช้เพื่อทำการควบคุมอุปกรณ์อำนวยความสะดวกแก่มนุษย์ ให้สามารถควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าในชีวิตประจำวัน อีกทั้งในอนาคตยังสามารถนำไปประยุกต์พัฒนาเพื่อให้สามารถควบคุมอุปกรณ์ไฟฟ้าอิเล็กทรอนิกส์ที่ซับซ้อนต่าง ๆ ได้อีกด้วย

### 1.2 วัตถุประสงค์

- 1) เพื่อสร้างความตระหนักถึงเทคโนโลยีปัญญาประดิษฐ์ที่ส่งผลต่อชีวิตประจำวัน
- 2) เพื่ออำนวยความสะดวกให้กับผู้ใช้งานโดยสามารถสั่งการควบคุมอุปกรณ์ผ่านเสียง
- 3) เพื่อศึกษากระบวนการสร้างเทคโนโลยีปัญญาประดิษฐ์

### 1.3 ขอบเขตของปริญญาานิพนธ์

- 1) จัดเตรียมข้อมูลเสียงเพื่อสามารถนำไปใช้ในการแปลงเป็นตัวหนังสือ
- 2) ออกแบบโมเดลให้สอดคล้องกับรูปแบบการทำนายผลและข้อมูล
- 3) สร้างการเรียนรู้ให้กับโมเดลจากข้อมูลที่ทำกรจัดเก็บ
- 4) ฝึกฝนโมเดลให้เรียนรู้เฉพาะเสียงภาษาไทยที่เป็นคำสั่งในการควบคุมอุปกรณ์
- 5) ออกแบบระบบให้มีการจดจำคำกระตุ้นก่อนเริ่มการประมวลผล
- 6) นำข้อมูลคำสั่งที่ทำนายโดยปัญญาประดิษฐ์มาประเมินความแม่นยำของโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 การรู้จำคำพูด

เทคโนโลยีรู้จำเสียงพูด (Automatic Speech Recognition: ASR) เป็นสาขาย่อยของวิชาภาษาศาสตร์คอมพิวเตอร์ ที่พัฒนาวิธีการและเทคโนโลยีที่ช่วยให้การรับรู้และการแปลภาษาพูดเป็นข้อความโดยคอมพิวเตอร์ ซอฟต์แวร์รู้จำเสียงพูดขั้นพื้นฐานมีคำศัพท์ที่จำกัด ซึ่งประเภทของระบบรู้จำเสียงพูดสามารถแบ่งได้เป็น 3 ประเภทดังนี้

1. เทคโนโลยีรู้จำเสียงพูดแบบคำโดด (Isolated speech) คือระบบที่รู้จำคำสั้น ๆ เพียงไม่กี่คำสั่ง เพื่อให้ระบบสามารถตอบโต้ได้อย่างรวดเร็ว เช่น Siri ของ Apple หรือ Google assistant
2. เทคโนโลยีรู้จำเสียงพูดแบบต่อเนื่อง (Continuous speech) คือระบบรู้จำคำจากเสียงอย่างต่อเนื่อง แล้วทำการพิจารณาตัดเสียงพูด เช่น ต้องการแยกเสียงของแต่ละคนออกจากเสียงรบกวนรอบข้างจำนวนมากหรือเสียงไวโอลินจากการแสดงดนตรีที่เหลือ
3. เทคโนโลยีรู้จำที่จำเสียงเพียงบางส่วน (Spontaneous speech) คือระบบที่จดจำเสียงที่ตรวจหาคำสำคัญเพียงคำเดียวในประโยคเพื่อหาใจความสำคัญ เช่น การสั่งใช้งานอุปกรณ์ด้วยเสียง หรือ การใช้เสียงประดิษฐ์สำหรับตัวแทนการสนทนา [1]

#### 2.2 ปัญญาประดิษฐ์ (Artificial Intelligence: AI)

Artificial Intelligence (AI) คือ เทคโนโลยีที่การจำลองความฉลาดของมนุษย์ลงในเครื่องจักรที่ถูกตั้งโปรแกรมให้คิด วิเคราะห์ วางแผน และตัดสินใจเลียนแบบมนุษย์ โดยการพัฒนา ระบบเพื่อให้มีความสามารถในการรับรู้ เรียนรู้ ใช้เหตุผล และตัดสินใจเลือกทางเลือกที่ดีที่สุดจากการวิเคราะห์ข้อมูลที่เกี่ยวข้องตามเงื่อนไขที่กำหนด โดยแบ่งออกเป็น 3 ประเภท

1. Artificial Narrow Intelligent (ANI) หรือ Weak AI หรือ Narrow AI คือ ปัญญาประดิษฐ์ที่มีระดับปัญญาและความสามารถในการเรียนรู้เฉพาะด้านใดเรื่องหนึ่ง
2. Artificial General Intelligent (AGI) หรือ Strong AI หรือ Deep AI คือ ปัญญาประดิษฐ์ที่มีระดับปัญญาและความสามารถเทียบเท่ามนุษย์ โดยสามารถเรียนรู้ได้หลายด้านในเวลาเดียวกัน
3. Artificial Super Intelligent (ASI) คือ ปัญญาประดิษฐ์ที่มีระดับปัญญาและความสามารถเทียบเท่ามนุษย์ โดยมีความสามารถที่หลากหลายรวมถึงการแก้ปัญหาและการตัดสินใจที่เหนือกว่ามนุษย์ [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 การเรียนรู้ของเครื่อง (Machine Learning: ML)

Machine Learning (ML) เป็นการประยุกต์คณิตศาสตร์และวิทยาศาสตร์ของคอมพิวเตอร์เพื่อให้คอมพิวเตอร์สามารถเรียนรู้ ตัดสินใจได้ และทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูล ผู้ใช้จำเป็นต้องเตรียมข้อมูลที่มีคุณภาพและมีปริมาณเพียงพอ เพื่อให้ Machine learning เรียนรู้ได้ผลดี Machine learning เหมาะสำหรับการแก้ไขปัญหาที่ค่อนข้างซับซ้อน โดยแบ่งออกเป็น 3 ประเภท ดังนี้

1. Supervised learning เป็นกลุ่มลอการิทึมที่ทำให้ระบบคอมพิวเตอร์เรียนรู้โดยการศึกษาค้นคว้าจากข้อมูลตัวอย่าง ในชีวิตจริงมักถูกนำไปใช้แทนการทำงานแบบมีกฎ (Rule base) หรือรูปแบบการทำงานที่ตายตัว หรือสามารถอธิบายเหตุผลออกมาได้อย่างชัดเจน Supervised learning ประกอบไปด้วย Regression เป็นคำตอบประเภทตัวเลข เช่น พยากรณ์สภาพอากาศ และ Classification เป็นคำตอบที่เป็นหมวดหมู่ เช่น แยกประเภทหมวดหมู่ของดอกไม้ แยกภาพสุนัขออกจากสัตว์ประเภทอื่น ๆ

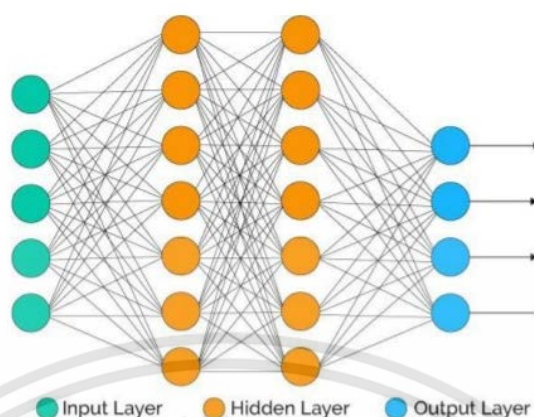
2. Unsupervised learning เป็นกลุ่มลอการิทึมที่ไม่มีป้ายกำกับ หรือการสอนอย่างชัดเจนว่าผลลัพธ์ที่ถูกต้องคืออะไร ประกอบไปด้วย Clustering เป็นการจัดกลุ่มข้อมูล เช่น มีข้อมูลผู้บริจาครายชื่อ ต้องการจัดกลุ่มผู้บริจาครายชื่อเป็น 3 กลุ่ม เพื่อออกแบบกลยุทธ์การสื่อสารกับลูกค้าแต่ละกลุ่มที่ต่างกันออกไป และ Non clustering เป็นปัญหาอื่น ๆ ที่ไม่ใช่การจัดกลุ่ม เช่น ตรวจสอบหาข้อมูลขึ้นที่มีความผิดปกติจากพวก (Anomaly detection)

3. Reinforcement learning เป็นการเรียนรู้ที่เปลี่ยนแปลงไปตามสภาพแวดล้อม เรียนรู้จากการลองผิดลองถูกจากสถานการณ์ในอดีตหรือระบบจำลอง เพื่อพัฒนาระบบการตัดสินใจให้ดียิ่งขึ้น [3]

## 2.4 หลักการของ Neural Network

โครงข่ายประสาทเทียมหรือที่เรียกกันว่า Artificial Neural Network (ANN) เป็นแบบจำลองหนึ่งในแขนงของงานวิจัยทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) โดยมีการจำลองมาจากการทำงานของระบบเซลล์ประสาทในสมองของมนุษย์ โดยโครงสร้างของ Neural Network จะมีลักษณะเหมือนเซลล์ประสาทหลาย ๆ เซลล์เชื่อมต่อกันเป็นโหนด เรียกว่า เพอร์เซพตรอน (Perceptron) ซึ่งจะเชื่อมต่อกับเส้นประสาทและเซลล์อื่นอีกหลาย ๆ เส้น มีโครงสร้างดังรูปที่ 2.1 ซึ่งประกอบด้วยชั้นข้อมูลเข้า (Input layer) ชั้นซ่อน (Hidden layer) และชั้นข้อมูลออก (Output layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ส่วนประกอบของโครงข่ายประสาทเทียม [4]

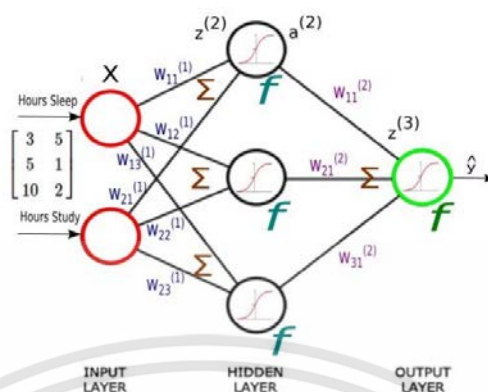
จากรูปที่ 2.1 จะเห็นว่าส่วนประกอบของโครงข่ายประสาทเทียมประกอบด้วย 3 ส่วน ดังนี้

1) ชั้นข้อมูลเข้า (Input Layer) ที่ชั้นนี้จะเป็นข้อมูลเข้า จำนวนของโหนดในชั้นนี้ขึ้นอยู่กับจำนวนของข้อมูลเข้าว่ามีข้อมูลอะไรบ้างที่จะนำเข้ามาคิดในแบบจำลอง เช่น ถ้าข้อมูลของลูกค้าเป็นข้อมูลเข้าที่ประกอบด้วย อายุ เพศ จังหวัด ที่อาศัย ดังนั้นชั้นข้อมูลเข้าก็จะมี 4 โหนด ซึ่งอาจจะเรียกปัจจัยที่นำมาวิเคราะห์เหล่านี้ว่าคุณลักษณะ (Feature)

2) ชั้นซ่อน (Hidden Layer) เป็นชั้นที่อยู่ระหว่างกลาง ซึ่งจะมีผลอย่างมากต่อประสิทธิภาพในการเรียนรู้ของแบบจำลอง ซึ่งในชั้นซ่อนจะสามารถมีกี่ชั้นก็ได้ และแต่ละชั้นจะมีจำนวนของนิวรอนจำนวนเท่าไรก็ได้เช่นกัน ซึ่งการเพิ่มชั้นและจำนวนนิวรอนจะส่งผลต่อการทำงานของโมเดล ในส่วนของชั้นซ่อนมีการทำงานเปรียบเสมือนส่วนที่เรียนรู้ข้อมูลเชิงลึก (deep learning) โดยสิ่งสำคัญในชั้นซ่อนอีกประการหนึ่งคือทุก ๆ โหนด ต้องประกอบด้วยฟังก์ชันกระตุ้นแบบไม่เป็นเชิงเส้น (Non-linear Activating Function)

3) ชั้นข้อมูลออก (Output Layer) ชั้นที่จะนำเอาข้อมูลจากการคำนวณไปใช้ และจำนวนของโหนดในชั้นนี้ขึ้นอยู่กับรูปแบบของข้อมูลออกที่จะเอาไปใช้ [4]

เป้าหมายหลักของการนำโครงข่ายประสาทเทียมมาใช้งาน คือ ต้องการพัฒนาระบบการเรียนรู้ของเครื่องจักรหรือที่เรียกว่า Machine Learning ให้มีประสิทธิภาพสูง การนำเอาโครงข่ายประสาทเทียมมาใช้งานแบ่งเป็น 2 ขั้นตอนหลักคือ ขั้นตอนฝึกฝน (Training, Feed forward propagation) หรือเรียนรู้ (Learning, Back propagation) และขั้นตอนการทดสอบ (Testing) หรือการใช้งานจริง



รูปที่ 2.2 โครงสร้างการทำงานของโครงข่ายประสาทเทียม [5]

#### 2.4.1 Feed forward propagation

จากรูปที่ 2.2 สามารถนำมาเขียนเป็นสมการทางคณิตศาสตร์ได้ดังสมการต่อไปนี้

$$z^{(21)} = x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + \dots + b_1 \quad (2.1)$$

$$z^{(22)} = x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + \dots + b_2 \quad (2.2)$$

$$z^{(23)} = x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + \dots + b_3 \quad (2.3)$$

เมื่อ ค่า  $x_i$  คือ Input ตัวที่  $i$

ค่า  $w_{ij}$  คือ ค่าน้ำหนักของอินพุตของข้อมูลตัวที่  $i$  Hidden Node ตัวที่  $j$

ค่า  $b_j$  คือ ค่าไบอัสของ Hidden Node ที่  $j$

ค่า  $z^{(ij)}$  คือ ผลของการคูณเมทริกซ์ของอินพุตตัวที่  $i$  Hidden Node ตัวที่  $j$

ดังนั้น หากที่  $n$  hidden node จะสามารถแสดงได้ดังสมการที่ 2.4

$$z^{(2n)} = x_1 w_{1n} + x_2 w_{2n} + x_3 w_{3n} + \dots + b_n \quad (2.4)$$

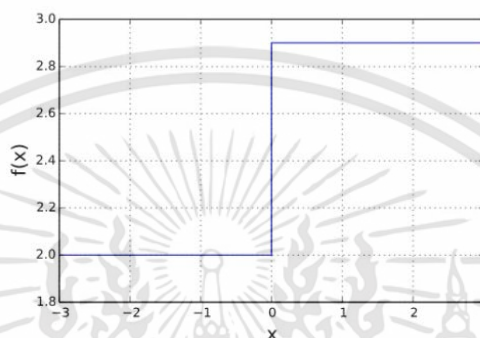
ค่าเอาต์พุตจากสมการที่ 2.4 ต้องนำมาผ่านฟังก์ชันกระตุ้น (Activating Function) เป็นฟังก์ชันในการคำนวณเพื่อทำนายค่าของข้อมูล ฟังก์ชันกระตุ้นจะถูกนำไปใช้ทั้งโหนดซ่อน (Hidden node) และโหนดข้อมูลออก (Output node) ซึ่งทั้งสองโหนดอาจใช้ฟังก์ชันกระตุ้นที่เหมือนหรือต่างกันก็ได้ สามารถแบ่งฟังก์ชันกระตุ้นได้เป็น 3 ประเภทดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.1.1 ฟังก์ชัน BINARY STEP

แสดงสมการของอินพุตและเอาต์พุตดังสมการที่ 2.5 และมีลักษณะเป็นกราฟดังรูปที่ 2.3

$$y = 0 \text{ if } i < 0, y = 1 \text{ if } i > 0 \quad (2.5)$$

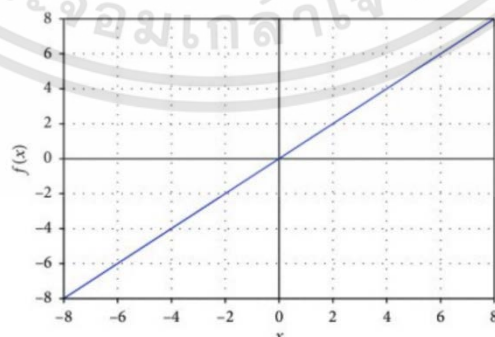


รูปที่ 2.3 ลักษณะของฟังก์ชัน Binary Step [6]

### 2.4.1.2 ฟังก์ชันเชิงเส้น (Linear Function)

ฟังก์ชันเชิงเส้นไม่นิยมนำมาใช้เป็นฟังก์ชันกระตุ้น เนื่องจากในโหนดซ่อนจะมีการคำนวณแบบการรวมเชิงเส้น (Linear combination) ถ้าฟังก์ชันกระตุ้นของโหนดซ่อนจะมีการคำนวณแบบเชิงเส้นอีก จะเป็นการทำงานซ้ำซ้อนกับการคำนวณแบบการรวมเชิงเส้นในชั้นข้อมูลออก และจะทำให้ผลลัพธ์เทียบเท่ากับสมการถดถอยลอจิสติก แสดงสมการของอินพุต และเอาต์พุตดังสมการที่ 2.6 และมีลักษณะเป็นกราฟดังรูปที่ 2.4 [7]

$$y = x \quad (2.6)$$



รูปที่ 2.4 ลักษณะของฟังก์ชันเชิงเส้น (Linear Function) [7]

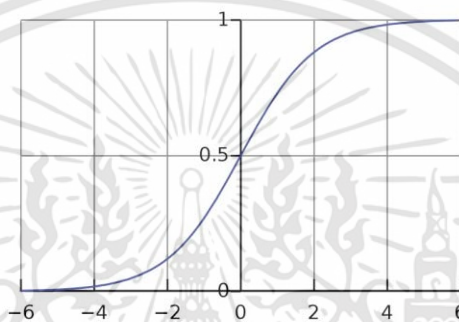
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.1.3 ฟังก์ชันไม่เป็นเชิงเส้น (Non-linear Function)

เป็นรูปแบบที่นิยมนำมาใช้มากที่สุด แบ่งออกเป็น 2 แบบดังต่อไปนี้

1) ฟังก์ชัน Sigmoid แสดงสมการของอินพุต และเอาต์พุตดังสมการที่ 2.7 และมีลักษณะกราฟดังรูปที่ 2.5

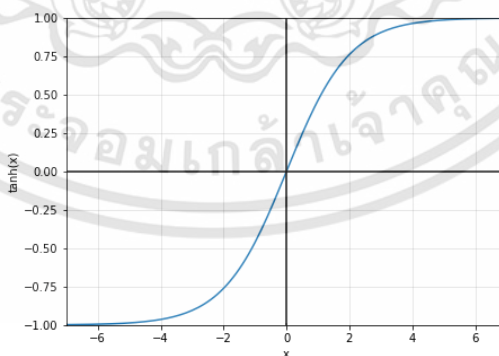
$$y = \frac{1}{1 + e^{-z}} \quad (2.7)$$



รูปที่ 2.5 ลักษณะของฟังก์ชัน Sigmoid [6]

2) ฟังก์ชัน Tanh แสดงสมการของอินพุต และเอาต์พุตดังสมการที่ 2.8 และมีลักษณะกราฟดังรูปที่ 2.6

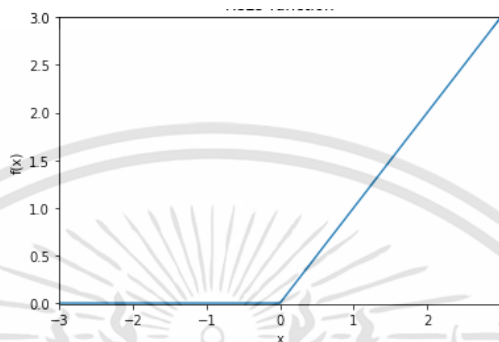
$$y = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.8)$$



รูปที่ 2.6 ลักษณะของฟังก์ชัน Tanh [6]

3) ฟังก์ชัน Relu มีสมการของอินพุต และเอาต์พุตดังสมการที่ 2.9 และมีลักษณะกราฟดังรูปที่ 2.7

$$y = \max(0, z) \quad (2.9)$$



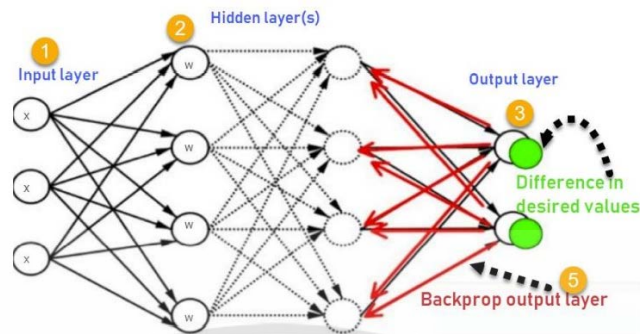
รูปที่ 2.7 ลักษณะของฟังก์ชัน Relu [6]

4) ฟังก์ชัน Softmax หรือ SoftArgMax Function หรือ Normalized Exponential Function คือ ฟังก์ชันที่รับ Input เป็น Vector ของจำนวนจริง แล้ว Normalize ออกมาเป็นความน่าจะเป็น Probability ที่ผลรวมเท่ากับ 1 มักถูกนำไปใช้ Layer สุดท้ายของ Neural Network ถูกนำไปใช้บ่อยในงาน Classification แสดงสมการของอินพุต และเอาต์พุตดังสมการที่ 2.10

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \text{ for } i = 1, 2, \dots, k \text{ and } z = (z_1, z_2, \dots, z_k) \in R^k \quad (2.10)$$

#### 2.4.2 Back propagation

เป็นขั้นตอนที่ต้องทำหลังจากขั้นตอนของ Feed forward propagation โดยเป็นขั้นตอนที่ใช้ในการปรับค่าน้ำหนักของเมทริกซ์ โดยสามารถแสดงขั้นตอนในการ Back propagation ดังรูปที่ 2.8



รูปที่ 2.8 ขั้นตอนในการ Back propagation [6]

กระบวนการในการ feed back สามารถแบ่งออกเป็นอีก 2 ขั้นตอนประกอบด้วย การคำนวณค่า cost ของฟังก์ชัน และกระบวนการในการปรับค่าน้ำหนักโดยใช้วิธีการของ Gradient descent

#### 2.4.2.1 กระบวนการคำนวณค่า cost ของฟังก์ชัน

เป็นกระบวนการคำนวณค่าความผิดพลาดกำลังสองของการฝึกฝนในแต่ละรอบของ Neural Network จะทำการวัดค่า  $y$  ที่ทำนายได้ว่ามีความคลาดเคลื่อนกับค่า  $y$  ที่เป็นข้อมูลจริงอยู่เท่าใด โดยสามารถแสดง cost ได้ดังสมการที่ 2.11

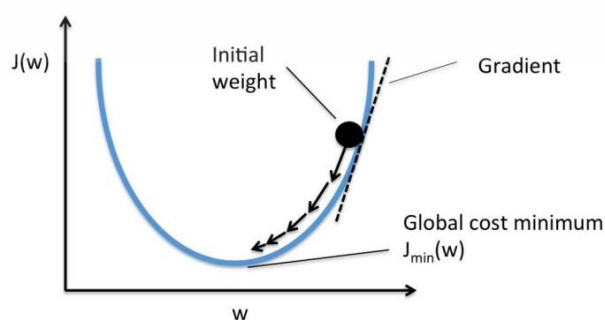
$$J = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 \quad (2.11)$$

เมื่อ ค่า  $m$  คือ จำนวนของข้อมูล

ค่า  $y_i$  คือ ข้อมูลที่ต้องการทำนายตัวที่  $i$

ค่า  $\hat{y}_i$  คือ ข้อมูลที่ทำนายได้โดย Neural Network ตัวที่  $i$

จากสมการที่ 2.11 สามารถนำมาพล็อตกราฟเพื่อดูความสัมพันธ์ระหว่างค่า cost ที่เกิดขึ้นกับค่าน้ำหนักได้ดังรูปที่ 2.9



รูปที่ 2.9 ความสัมพันธ์ระหว่างค่า cost ที่เกิดขึ้นกับค่าน้ำหนักของเมทริกซ์ [6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.2.2 กระบวนการในการปรับค่าน้ำหนักของเมทริกซ์โดยใช้วิธีการของ Gradient descent

กระบวนการ Gradient descent เป็นกระบวนการปรับค่าน้ำหนักของเมทริกซ์โดยการหาอัตราการเปลี่ยนแปลงของค่า cost เทียบกับน้ำหนักของเมทริกซ์เพื่อพยายามปรับค่าน้ำหนักของเมทริกซ์ให้ค่า cost ลดลงสู่จุด 0 หรือมีค่าต่ำที่สุดเท่าที่จะเป็นไปได้ เรียกจุดนี้ว่า Global cost minimum สามารถแสดงได้ดังรูปที่ 2.9 โดยสามารถแสดงสมการการหาอัตราเปลี่ยนแปลงของค่า cost เทียบกับค่าน้ำหนักของเมทริกซ์ได้ดังสมการที่ 2.12

$$\frac{\partial J}{\partial w^{(s)}} \quad (2.12)$$

จากสมการที่ 2.12 สามารถนำมาปรับค่าน้ำหนักของเมทริกซ์ได้ดังสมการที่ 2.13

$$w^{new} = w^{old} - lr * \frac{\partial J}{\partial w} \quad (2.13)$$

เมื่อแทนค่าในสมการที่ 2.12 และ 2.13 สามารถแสดงได้ดังสมการที่ 2.14

$$\frac{\partial J}{\partial w^{(s)}} = a^{(i)T} * (y - \hat{y})f(z^{(i+1)}) \quad (2.14)$$

เมื่อ ค่า  $i$  แสดงถึงชั้นที่  $i$  ของ Hidden Node

ค่า  $y$  คือ ข้อมูลที่ต้องการทำนาย

ค่า  $\hat{y}$  คือ ข้อมูลที่ทำนายได้โดย Neural Network

ค่า  $a$  คือ output ของแต่ละ Hidden Node

ค่า  $z$  คือ ผลคูณระหว่างเมทริกซ์อินพุตกับเมทริกซ์ของน้ำหนักเมทริกซ์

ค่า  $f$  คือ ฟังก์ชันอนุพันธ์ของฟังก์ชัน sigmoid

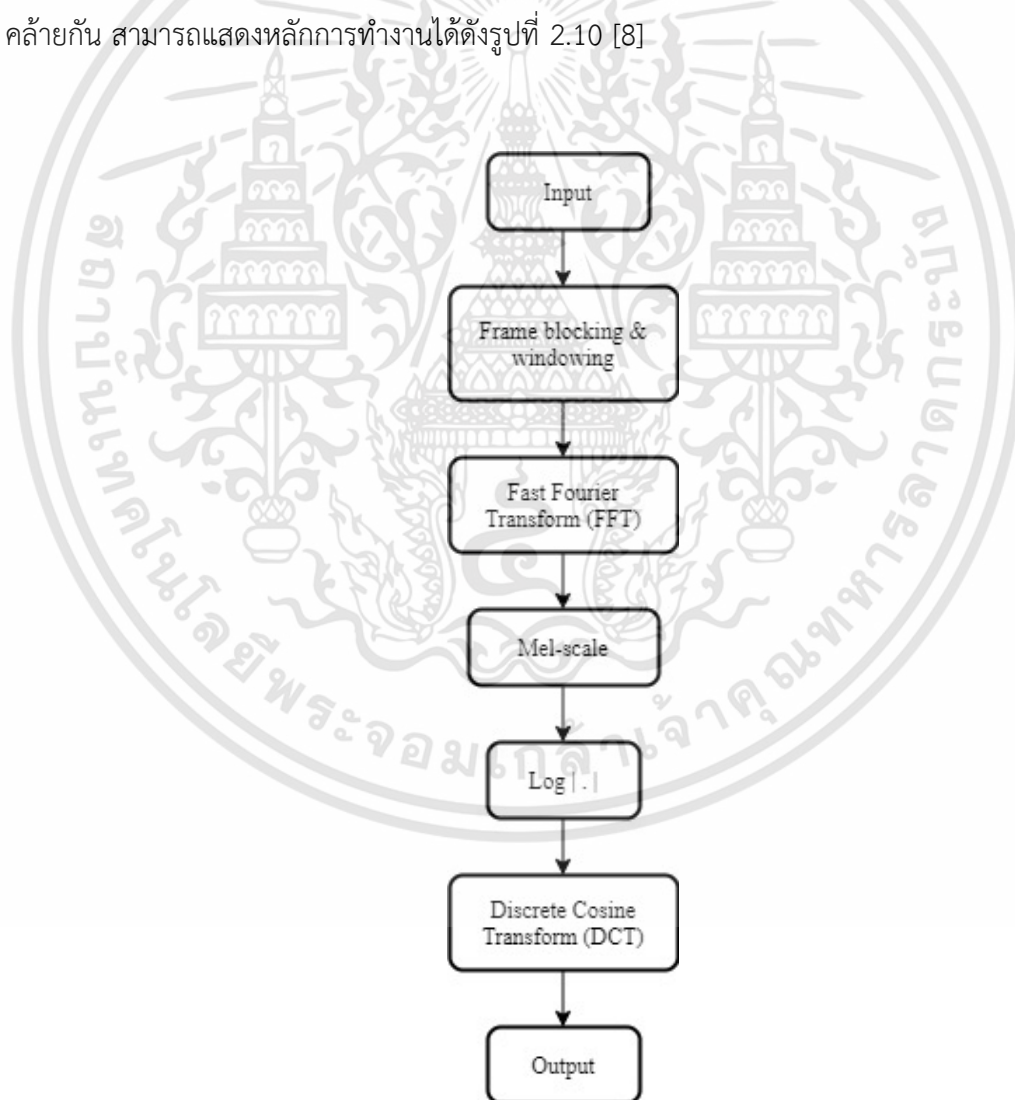
### 2.4.3 ความผิดพลาดในการฝึกฝนโมเดล

ในการคำนวณค่าความผิดพลาดในการฝึกฝนของข้อมูลจะใช้วิธีการหาค่าความผิดพลาดด้วยวิธีการ Sparse Categorical Cross entropy เป็นวิธีที่เหมาะสมสำหรับการสร้างโมเดลแบบจำแนกข้อมูลหลายประเภท โดยสามารถแสดงได้ดังสมการที่ 2.15 [7]

$$Loss = - \sum_{i=1}^{output\ size} y_i \cdot \log \hat{y}_i \quad (2.15)$$

## 2.5 กระบวนการสกัดคุณลักษณะเด่น (Feature Extraction)

Mel Frequency Cepstral Coefficient (MFCC) เป็นกระบวนการสกัดคุณลักษณะเด่น เพื่อดึงคุณลักษณะต่าง ๆ ของสัญญาณเสียง ก่อนนำไปเข้าสู่กระบวนการฝึกฝน Neural Network กระบวนการ MFCC เป็นการหาค่าลักษณะเด่นออกมาในรูปแบบสเปกโตรแกรม (Spectrogram) เนื่องจากสเปกโตรแกรมเป็นการจัดเรียงสามมิติที่แสดงการกระจายความถี่และความเข้มของเสียง เป็นฟังก์ชันของเวลา ช่วยยึดความถี่ที่ต่ำลงเลียนแบบการรับรู้ของมนุษย์ ซึ่งไวต่อการเปลี่ยนแปลงความถี่ สูงน้อยกว่า เหมาะสำหรับการนำมาทำระบบการรู้จำเสียง ซึ่งทำได้โดยการคำนวณหาค่าสัมประสิทธิ์ซีเชปตรัลความถี่เมล ซึ่งข้อมูลเสียงพูดที่มีลักษณะแบบเดียวกันก็จะมีลักษณะเด่นของเสียงที่เหมือนหรือคล้ายกัน สามารถแสดงหลักการทำงานได้ดังรูปที่ 2.10 [8]



รูปที่ 2.10 กระบวนการสกัดคุณลักษณะเด่น  
(Mel Frequency Cepstral Coefficient: MFCC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.10 สัญญาณเสียงอินพุตที่อยู่ในรูปโดเมนเวลาจะถูกแบ่งสัญญาณออกเป็นเฟรมขนาดเล็ก โดยในแต่ละเฟรมจะมีการลดการรบกวนที่จุดเริ่มต้นและจุดสิ้นสุดของแต่ละเฟรม จากนั้นจะนำสัญญาณในแต่ละเฟรมผ่านกระบวนการ Fast Fourier Transform (FFT) เพื่อเปลี่ยนให้สัญญาณอยู่ในรูปโดเมนความถี่ จะได้ค่าสเปกตรัมกำลังของสัญญาณ (Power spectrum) สามารถคำนวณ FFT ได้จากสมการที่ 2.15

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot w_N^{kn} \quad (2.15)$$

เมื่อ ค่า  $w_N^{kn} = e^{\frac{j2\pi}{N}kn}$ ;  $k = 0, 1, 2, 3, \dots$

ค่า  $x(n)$  เป็นสัญญาณขาเข้า

ค่า  $X(k)$  เป็นสัญญาณขาออก

จากนั้นค่าสเปกตรัมกำลังของสัญญาณที่ได้จากกระบวนการ FFT จะถูกส่งเข้าไปในกระบวนการของชุดตัวกรอง (Mel Filterbank) ที่อยู่บนสเกลความถี่แบบเมล (Mel-scale) ดังสมการที่ 2.16 และจะทำการรวมสเปกตรัมในแต่ละเฟรม จะได้ค่าสเปกตรัมกำลังแบบเมล (Mel-power spectrum scale) ออกมา

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (2.16)$$

เมื่อ ค่า  $m$  เป็นค่าเมลสเกล (Mel scale) มีหน่วยเป็น mel

ค่า  $f$  เป็นค่าสเปกตรัมของสัญญาณ มีหน่วยเป็น Hertz (Hz)

จากนั้นทำการหาค่าลอการิทึมของสเปกตรัมทั้งหมด และสุดท้ายนำไปผ่านกระบวนการ Discrete Cosine Transform (DCT) ดังสมการที่ 2.17 เพื่อคำนวณหาค่าสัมประสิทธิ์เซปเตรียมความถี่เมล สำหรับนำไปใช้งานในกระบวนการฝึกฝน Neural Network

$$y(n) = \sqrt{\frac{2}{n}} \sum_{k=0}^{n-1} x(k) \cdot \frac{1}{\sqrt{1+\delta_{k1}}} \cdot \cos\left(\frac{\pi}{2n}(2n-1)(k-1)\right) \quad (2.17)$$

เมื่อ ค่า  $k = 0, 1, 2, \dots$ ,

$x(n)$  เป็นสัญญาณขาเข้า

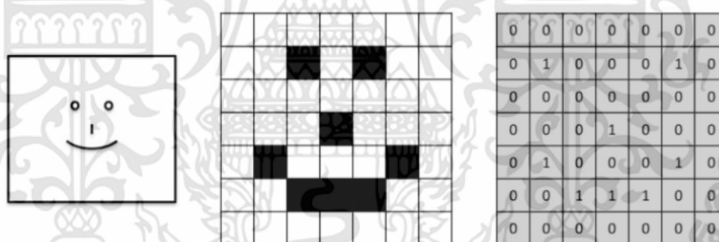
$N$  เป็นความยาวของสัญญาณ

$y(n)$  เป็นสัญญาณขาออก

$\delta_{k1}$  เป็นค่า Kronecker delta

## 2.6 Convolution Neural Network (CNN)

แบบจำลอง Convolution Neural Network (CNN) เป็นแบบจำลองที่มีความสามารถในการจำแนกข้อมูลประเภทรูปภาพ หลักการทำงานของ CNN คือการใช้ชั้นชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาส่วนต่างๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่างๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำ CNN จะทำการแบ่งรูปภาพออกเป็นพื้นที่ย่อย ๆ เป็นพิกเซลแต่ละอันเพื่อทำการวิเคราะห์เมทริกซ์ของรูปภาพ โดยถ้าเป็นรูปภาพสีขาวดำ จะเป็นเมทริกซ์  $2 \times 2$  แต่ถ้าเป็นภาพสีจะเป็นเมทริกซ์  $3 \times 3$  ยกตัวอย่างดังรูปที่ 2.11



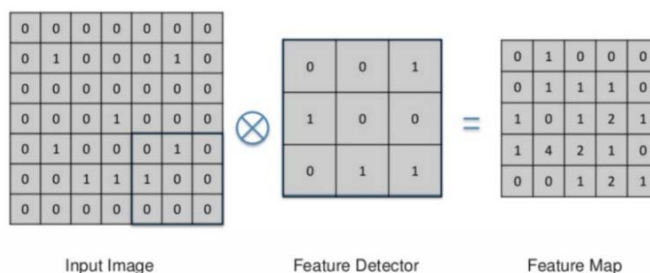
รูปที่ 2.11 การแบ่งภาพออกเป็นเมทริกซ์ [9]

ขั้นตอนในการทำแบบจำลอง Convolution Neural Network แบ่งออกเป็น 4 ขั้นตอนดังต่อไปนี้

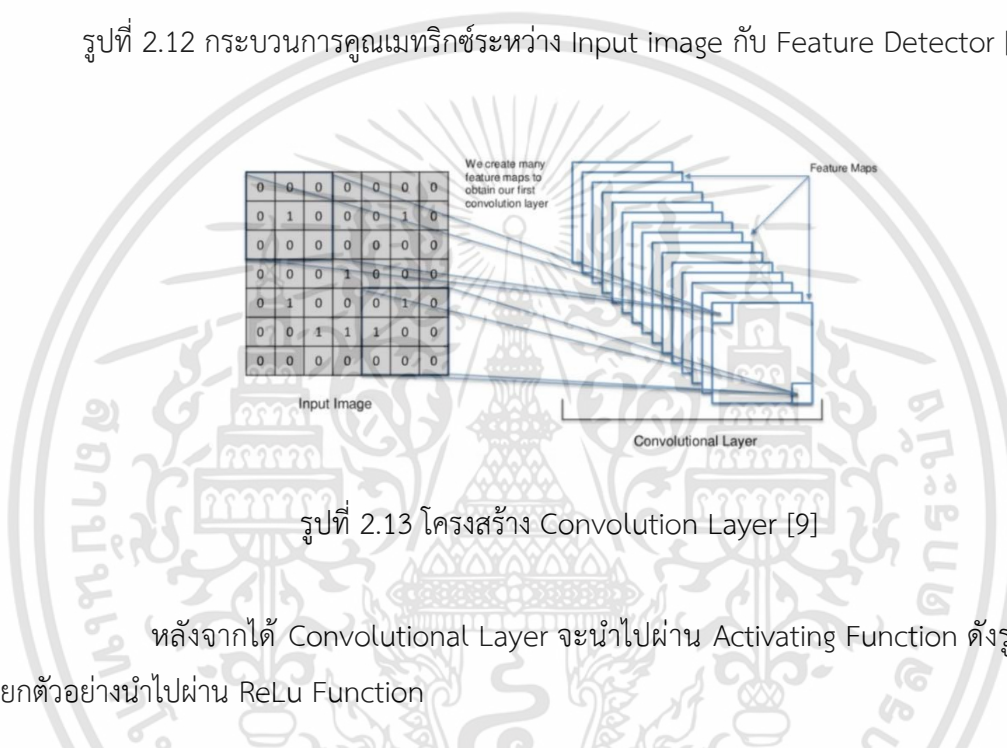
### 2.6.1 Convolution Layer

เป็นขั้นตอนในการคูณเมทริกซ์ระหว่าง Input Image กับ Feature Detector (Filter) ทำให้ได้ Feature Map โดยจะทำการคูณภายใน kernel size ที่กำหนด

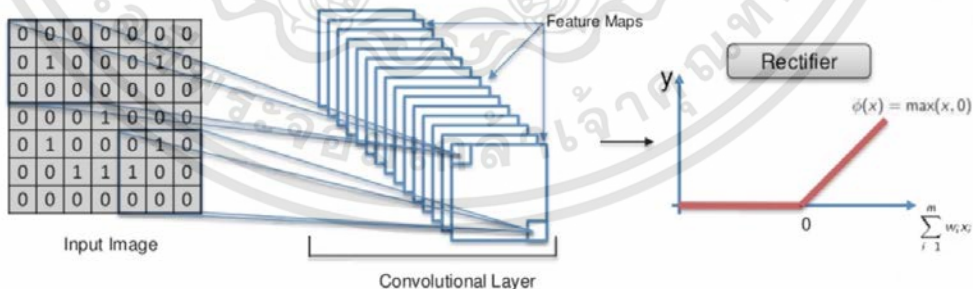
จากรูปที่ 2.12 กำหนดให้ kernel size มีขนาด  $3 \times 3$  คูณกับเมทริกซ์ Feature Detector ขนาด  $3 \times 3$  และกำหนดค่า stride=1 เมื่อทำการคูณเมทริกซ์ทั้งหมดแล้วจะได้ Feature Map จำนวนมากโดยจะเรียกทั้งหมดว่า Convolutional Layer ดังรูปที่ 2.13



รูปที่ 2.12 กระบวนการคูณเมทริกซ์ระหว่าง Input image กับ Feature Detector [9]



หลังจากได้ Convolutional Layer จะนำไปผ่าน Activating Function ดังรูปที่ 2.14 ยกตัวอย่างนำไปผ่าน ReLu Function



## 2.6.2 Pooling Layer

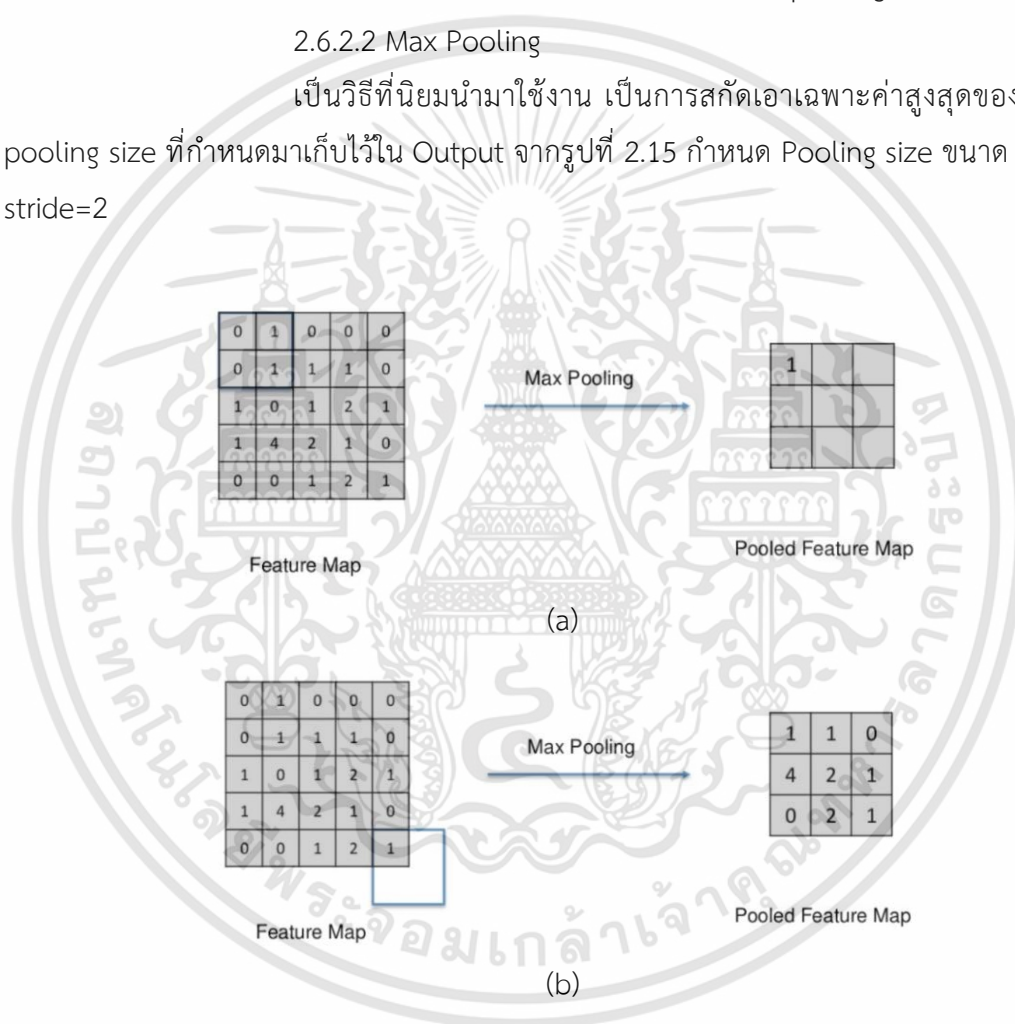
Pooling layer เป็นชั้นที่เชื่อมจาก Convolutional Layer โดยมีเป้าหมายคือทำให้ขนาดของ Feature Map ลดลง มีความสำคัญ คือช่วยสกัดเอาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลให้รวดเร็วยิ่งขึ้น แบ่งออกเป็น 2 ประเภท ดังนี้

### 2.6.2.1 Average Pooling

เป็นการหาค่าเฉลี่ยทั้งหมดในบริเวณเมทริกซ์ pooling size ที่กำหนด

### 2.6.2.2 Max Pooling

เป็นวิธีที่นิยมนำมาใช้งาน เป็นการสกัดเอาเฉพาะค่าสูงสุดของเมทริกซ์ pooling size ที่กำหนดมาเก็บไว้ใน Output จากรูปที่ 2.15 กำหนด Pooling size ขนาด 2x2 มีค่า stride=2

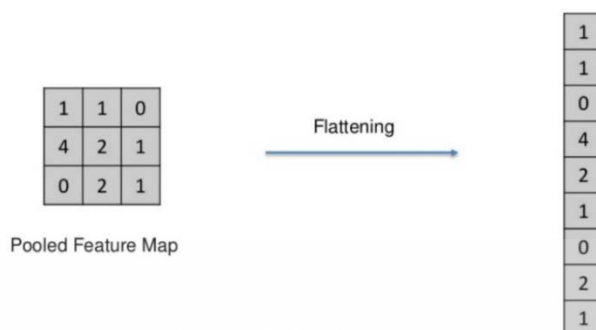


รูปที่ 2.15 (a), (b) กระบวนการ Max Pooling [9]

## 2.6.3 Flatten Layer

เป็นการทำ Pooling Feature Map ที่ได้ทำเป็นคอลัมน์เดียวกัน เพื่อความสะดวกในการนำไปวิเคราะห์ข้อมูล แสดงได้ดังรูปที่ 2.16

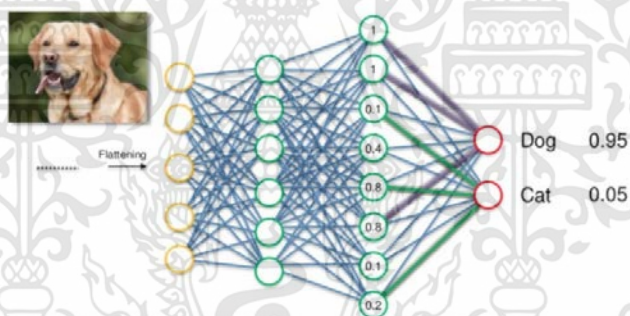
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 กระบวนการ Flatten Layer [9]

### 2.6.4 Fully Connected Layer

คือ การนำข้อมูลที่ได้จากขั้นตอน Flatten มาเข้าสู่โมเดล Deep Learning เพื่อนำผลลัพธ์ของโมเดลที่ได้ไปใช้ในการทำนาย แสดงได้ดังรูปที่ 2.17



รูปที่ 2.17 กระบวนการ Fully Connected Layer [9]

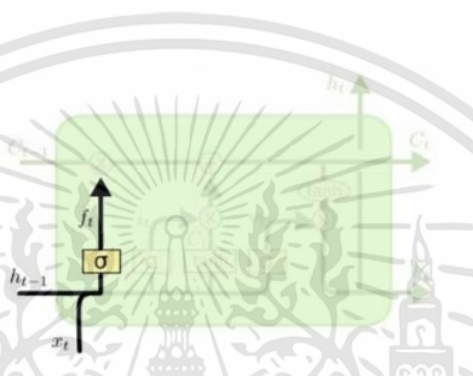
### 2.7 Long Short-Term Memory (LSTM)

แบบจำลอง Long Short-Term Memory (LSTM) เป็นโครงข่ายประเภท Recurrent Neural Network (RNN) รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหลักการการทำงาน คือ สามารถเก็บสถานะหรือข้อมูลของแต่ละโหนดไว้ เพื่อที่เวลาย้อนกลับไปดูจะได้ทราบถึงที่มาของข้อมูลค่าดังกล่าวเดิมเป็นค่าอะไร จุดเด่นของ LSTM คือฟังก์ชันพิเศษที่มีหน้าที่เสมือนประตู (Gate) ที่คอยควบคุมข้อมูลที่จะเข้ามาในแต่ละโหนด ซึ่งประกอบไปด้วย 3 ส่วนดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.7.1. Forget Gate Layer

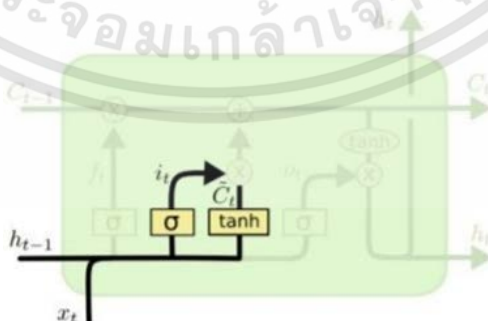
ทำหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน cell state นั้นควรจะถูกลบไว้หรือควรที่จะทิ้งไป ซึ่งข้อมูลที่ถูกตัดสินว่าควรเก็บไว้นั้นจะถูกประเมินจากข้อมูลอินพุตที่เข้ามาในโหนดนั้น ๆ รวมกับผลลัพธ์ที่ได้จากการคำนวณของโหนดก่อนหน้าผ่านฟังก์ชัน sigmoid ผลลัพธ์ที่ได้จาก Forget gate layer จะอยู่ระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้น หมายถึงให้ลบค่า cell state เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า cell state นี้ต่อไป แสดงโครงสร้างได้ดังรูปที่ 2.18



รูปที่ 2.18 โครงสร้าง Forget gate layer [10]

### 2.7.2 Input Gate Layer

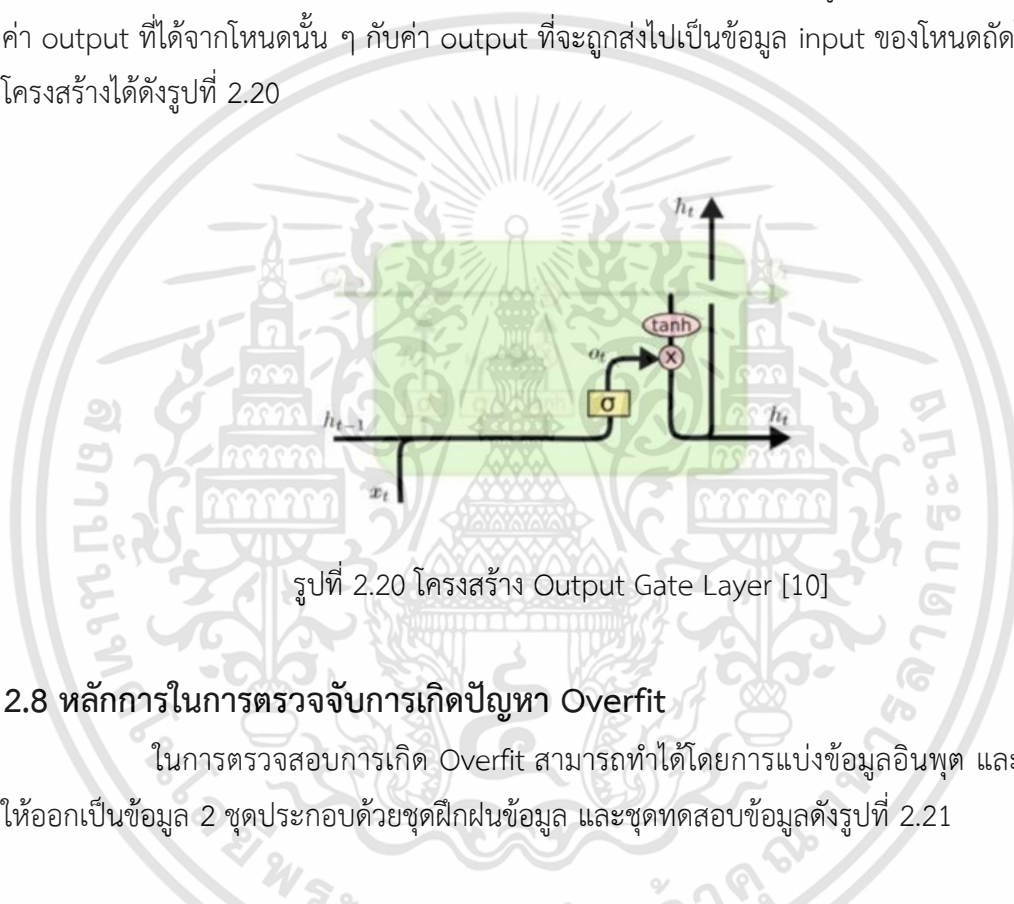
ทำหน้าที่รับข้อมูล input เข้ามาใหม่แล้วทำการบันทึกหรือข้อมูลลงไปในแต่ละโหนด โดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ ถ้าต้องการ update cell state เมื่อทำการรับข้อมูลอินพุตเข้ามาแล้ว ฟังก์ชัน sigmoid ที่เป็นตัวควบคุมจะเรียกใช้ input gate เพื่อเลือกว่าจะให้ update cell state หรือไม่ และในส่วนที่สองถ้า input gate เลือกที่จะทำการ update cell state ฟังก์ชัน tanh ก็จะทำการสร้าง candidate values ขึ้นมาใน state แสดงโครงสร้างได้ดังรูปที่ 2.19



2.19 โครงสร้าง Input Gate Layer [10]

### 2.7.3 Output Gate Layer

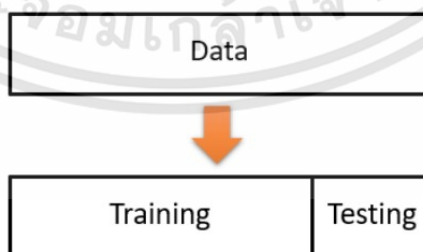
ทำหน้าที่เตรียมการส่งออกข้อมูล (output data) โดยข้อมูลที่จะทำการส่งออกนั้นจะดูจาก cell state ที่ผ่านกระบวนการคำนวณต่าง ๆ แล้ว โดยฟังก์ชัน sigmoid จะเป็นตัวเลือกว่าข้อมูลส่วนไหนใน cell state ที่จะถูกส่งออก จากนั้นก็นำค่า cell state เข้าฟังก์ชัน tanh เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1 แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า output ที่ได้จาก sigmoid gate จากนั้นก็จะได้ค่า output ซึ่งค่า output ที่ได้ออกมานั้นจะถูกแบ่งออกเป็น 2 ส่วน คือค่า output ที่ได้จากโหนดนั้น ๆ กับค่า output ที่จะถูกส่งไปเป็นข้อมูล input ของโหนดถัดไป แสดงโครงสร้างได้ดังรูปที่ 2.20



รูปที่ 2.20 โครงสร้าง Output Gate Layer [10]

## 2.8 หลักการในการตรวจจับการเกิดปัญหา Overfit

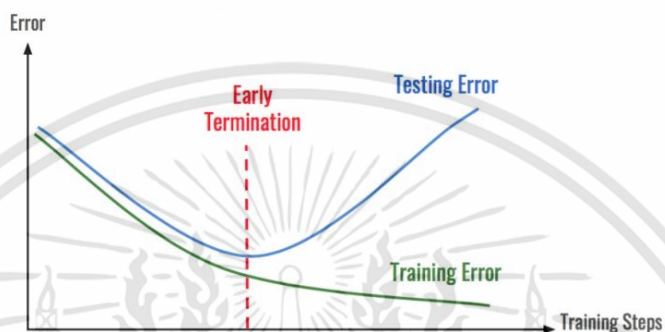
ในการตรวจสอบการเกิด Overfit สามารถทำได้โดยการแบ่งข้อมูลอินพุต และเอาต์พุตให้ออกเป็นข้อมูล 2 ชุดประกอบด้วยชุดฝึกฝนข้อมูล และชุดทดสอบข้อมูลดังรูปที่ 2.21



รูปที่ 2.21 การแบ่งข้อมูลอินพุตให้ออกเป็นข้อมูล 2 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

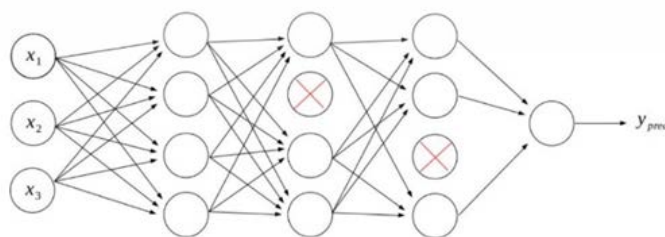
จากรูปที่ 2.21 จะนำข้อมูลฝึกฝนมาใช้ในการฝึกฝนโมเดล Neural Network และนำค่าข้อมูลเอาต์พุตที่ได้จากการฝึกฝนโมเดล Neural Network มาคำนวณหาค่าความผิดพลาด โดยใช้ข้อมูลฝึกฝน และข้อมูลทดสอบในการคำนวณ โดยจะนำมาพล็อตกราฟแสดงความสัมพันธ์ระหว่างค่าความผิดพลาดกับรอบที่ทำการฝึกฝน และข้อมูลทดสอบดังรูปที่ 2.22



รูปที่ 2.22 ความสัมพันธ์ระหว่างค่าความผิดพลาดกับรอบที่ทำการฝึก [11]

จากรูปที่ 2.22 ทำการตรวจสอบการเกิด Overfit ได้โดยการเช็คจุดที่จุด Early Termination หากมีการคำนวณค่าผิดพลาดของข้อมูลทดสอบมีความผิดพลาดมากขึ้นในจำนวนรอบที่สูงขึ้นจะถือว่าเกิดปรากฏการณ์ Overfit

หลักการแก้ปัญหา Overfitting ด้วยวิธีการ Dropout ปัญหา Overfitting เกิดจากการที่โมเดลผ่านการฝึกฝนแล้วมีความเหมาะสมกับชุดฝึกฝนข้อมูล แต่ทำนายออกมาไม่แม่นยำเมื่อนำมาใช้กับชุดทดสอบข้อมูล ซึ่งโมเดล Deep Neural Network มักจะมีปัญหา Overfit เมื่อฝึกฝนกับ Dataset ที่มีขนาดเล็ก หลักการของวิธีการ Dropout คือ จะเข้าไปปิด Neuron บางตัวในแต่ละชั้น การปิด Neuron บางตัวในเครือข่ายแต่ละชั้น เป็นการเลือกปิดแบบสุ่ม ไม่ให้รับข้อมูลใดๆ ดังนั้น Neuron ที่ถูกปิดก็จะไม่ส่งข้อมูลออกไปยัง Neuron ตัวอื่นเช่นเดียวกันดังรูปที่ 2.23 [12]



รูปที่ 2.23 หลักการ Dropout [12]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 Library Python ที่เกี่ยวข้องกับงาน

### 2.9.1 ไลบรารี OS

เป็นไลบรารีที่เรียกใช้ระบบปฏิบัติการโดยขึ้นอยู่กับฟังก์ชันการทำงาน เช่น หากต้องการเปิดไฟล์ หรืออ่านไฟล์ จะใช้ `open()`, หากต้องการจัดการหนทางจะใช้ `os.path`

### 2.9.2 ไลบรารี Numpy

เป็นไลบรารีที่รองรับการดำเนินการทางเมทริกซ์

### 2.9.3 ไลบรารี Matplotlib

เป็นไลบรารีที่ใช้ในการพล็อตกราฟสำหรับภาษาไพธอน และเป็นส่วนขยายของ Numpy ถูกออกแบบมาเพื่อให้ความคล้ายคลึงกับ MATLAB

### 2.9.4 ไลบรารี Time

เป็นไลบรารีที่เกี่ยวข้องกับวันที่ และเวลาต่างๆ สำหรับฟังก์ชันการทำงานที่เกี่ยวข้องกับเวลา เช่น `time.time()` ถูกใช้เพื่อเรียกการเริ่มจับเวลาของโปรแกรม โดยจะจับเวลาเป็นหน่วยวินาที

### 2.9.5 Tensorflow

เป็น open source ใช้สำหรับการพัฒนาโปรแกรมด้วยการเรียนรู้ของเครื่อง โดยเฉพาะการสร้างโครงข่ายประสาทเทียม ใช้งานง่าย และมีประสิทธิภาพการทำงานสูง

### 2.9.6 Librosa

เป็นไลบรารีสำหรับการประมวลผลเสียง มีหลายฟังก์ชันการทำงาน เช่น `librosa.feature.mfcc()` ถูกใช้เพื่อประมวลผลเสียงให้อยู่ในรูปของ MFCC

### 2.9.7 Pyaudio

เป็นไลบรารีสำหรับเล่นและอัดเสียงในภาษาไพธอน ซึ่งรองรับเฉพาะไฟล์เสียง wav. และ wave เท่านั้น

### 2.9.8 Linux Infrared Remote Control (LIRC)

เป็นแพ็คเกจ Open source ที่อนุญาตให้ผู้รับและส่งสัญญาณอินฟราเรดด้วยระบบคอมพิวเตอร์ที่ใช้ Linux และตัวรับสัญญาณ IR ผู้ใช้สามารถควบคุมคอมพิวเตอร์ด้วยรีโมทคอนโทรลอินฟราเรด เช่น รีโมทคอนโทรลของทีวี ผู้ใช้อาจควบคุมทีวีดีหรือเล่นเพลงด้วยรีโมทคอนโทรลได้ [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.10 ภาษาไพธอน (Python)

ภาษาไพธอน กำเนิดขึ้นในปลายปี 1980 และเริ่มต้นพัฒนาในเดือนธันวาคม 1989 โดย Guido van Rossum ที่ประเทศเนเธอร์แลนด์ ภาษาไพธอนเป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่มีการใช้อย่างกว้างขวาง เนื่องจากถูกออกแบบมาให้มีโครงสร้างไม่ซับซ้อนและเรียนรู้ได้ง่าย ภาษาไพธอนเป็นภาษาเขียนโปรแกรมแบบไดนามิก มีระบบจัดการหน่วยความจำอัตโนมัติ และมีตัวแปลคำสั่ง (Interpreter) ให้ใช้ในหลายระบบปฏิบัติการ ได้แก่ Mac, Windows, Linux และ Unix นอกจากนี้ยังเป็นโปรแกรมแบบ Open source จึงเหมาะสำหรับการ พัฒนาโปรแกรมหลายประเภท เช่น เว็บแอปพลิเคชัน เกม รวมถึงโปรแกรมประยุกต์ใช้งานต่างๆอีกด้วย [14]

คุณสมบัติของภาษาไพธอน

- 1) ภาษาไพธอนเป็นภาษาสคริปต์ ซึ่งใช้เวลาในการเขียนชุดคำสั่งและเวลาในการคอมไพล์ไม่นาน เกินไป และสามารถทำงานร่วมกับภาษาอื่นได้ จึงเหมาะกับงานด้านดูแลระบบ (System administration)
- 2) ภาษาไพธอนเขียนง่าย รูปแบบคำสั่งไม่ซับซ้อน ทำให้เกิดความเข้าใจได้ง่าย และมีไลบรารีช่วยให้ การเขียนชุดคำสั่งรวดเร็วยิ่งขึ้น ซึ่งเหมาะกับผู้เริ่มต้นเขียนโปรแกรม รวมถึงนักพัฒนาโปรแกรมอีกด้วย
- 3) ตัวแปลคำสั่งของภาษาไพธอน (Python interpreter) ภาษาไพธอนทำงานโดยการแปลคำสั่งทีละคำสั่ง จะสามารถเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที ทำให้ง่ายต่อการทดสอบดูแล และแก้ไขชุดคำสั่ง ซึ่งแตกต่างจากภาษาอื่นเช่น C หรือ Java ที่ต้องผ่านกระบวนการคอมไพล์เพื่อแปลคำสั่งทั้งหมดในคราวเดียวก่อนจึงจะเรียกใช้งานโปรแกรมได้

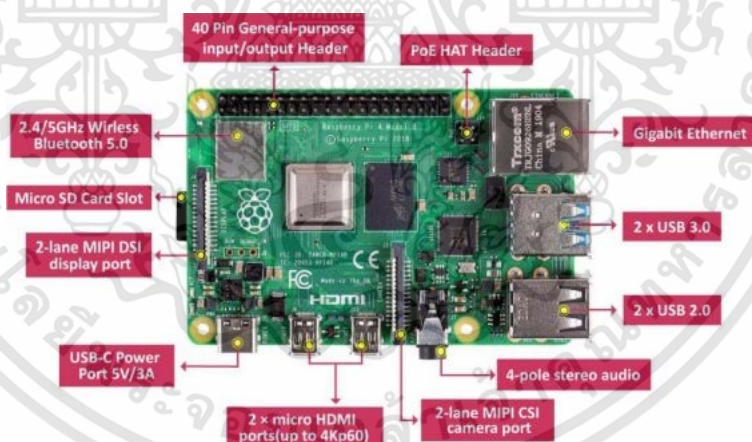
## 2.11 ระบบปฏิบัติการ Linux

ระบบปฏิบัติการ Linux เป็นระบบปฏิบัติการ (Operating System) ชนิดหนึ่งเช่นเดียวกันกับ Windows เป็นระบบปฏิบัติการที่มีความสามารถสูงในการบริหารระบบเครือข่าย อินเทอร์เน็ต มีลักษณะคล้ายการจำลองการทำงานมาจากยูนิกซ์ แต่จะมีความยืดหยุ่นในการทำงานมากกว่า เป็นระบบปฏิบัติการประเภทแจกฟรี (Open Source) ผู้นำไปใช้งานสามารถที่จะพัฒนาและปรับปรุงในส่วนที่เกิดปัญหาระหว่างใช้งานได้ทันที อีกทั้งยังสามารถปรับให้เข้ากับฮาร์ดแวร์ที่ใช้เพื่อให้ได้ประสิทธิภาพของระบบมากที่สุด และยังมีการเพิ่มสมรรถนะ (Update) อยู่ตลอดเวลา [15]

## 2.12 โมดูล Raspberry Pi 4 model B

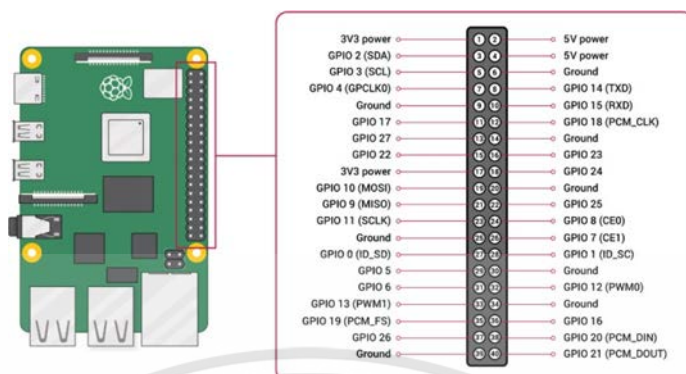
Raspberry Pi เป็นบอร์ดประมวลผลขนาดเล็กที่มีรูปแบบการทำงานคล้ายกับคอมพิวเตอร์เครื่องหนึ่ง สามารถรับข้อมูลจกคีย์บอร์ด จอมอนิเตอร์ และเมาส์ได้ นอกจากนี้ยังมีระบบการเชื่อมต่ออินเทอร์เน็ต ทั้งแบบใช้สาย และไร้สาย มีประสิทธิภาพในการทำงานได้หลากหลาย ไม่ว่าจะเป็นการเล่นเกม การใช้งาน อินเทอร์เน็ตสำหรับการส่งอีเมล หรือการเล่นไฟล์วิดีโอที่มีความละเอียดสูง (High-Definition) รองรับการติดตั้งระบบปฏิบัติการของ Linux (Linux Operating System) ได้หลายระบบ เช่น Arch Linux, Pidora (Fedora), Raspbian (Debian) เป็นต้น ซึ่งติดตั้งระบบบน SD Card [16]

Raspberry Pi 4 model B มีคุณสมบัติคือโปรเซสเซอร์แบบ 4 คอร์ รุ่น 64 บิต ทำงานที่ความเร็ว 1.5 GHz, พร้อมรองรับการแสดงผลแบบสองจอในความคมชัดสูงสุด 4K ที่อัตรา 60 fps, RAM สูงสุด 4 GB, LAN ไร้สายที่รองรับทั้งคลื่น 2.4/5.0 GHz, Bluetooth 5.0/BLE, เครื่องถ่าย True Gigabit Ethernet, USB 3.0 และการใช้งานเทคโนโลยีPoE (ผ่านส่วนเสริม PoE HAT แบบแยก) [17] และมีจุดเชื่อมต่อ GPIO ให้สามารถเชื่อมต่อกับอุปกรณ์อื่นได้ มีรายละเอียดดังรูปที่ 2.24



รูปที่ 2.24 รายละเอียดของ Raspberry Pi 4 model B [17]

จากรูปที่ 2.24 ส่วนของพอร์ต GPIO ของ Raspberry Pi 4 model B มีทั้งหมด 40 pin ดังรูปที่ 2.25



รูปที่ 2.25 พอร์ต GPIO ของ Raspberry Pi 4 model B [18]

### 2.13 โมดูลส่งสัญญาณอินฟราเรด KY-005

โมดูลส่งสัญญาณอินฟราเรด KY-005 ปล่อยแสงอินฟราเรดที่ 38 kHz ระยะทางในการส่งสัญญาณ 20-400 ซม. โมดูลนี้ประกอบด้วย LED, อินฟราเรดขนาด 5 มม. และหมุดส่วนในการเชื่อมต่ออุปกรณ์ 3 ขา สามารถใช้ควบคุมทีวี สเตอริโอ เครื่องปรับอากาศ และอุปกรณ์อื่น ๆ ด้วยตัวรับสัญญาณ IR นอกจากนี้ยังสามารถใช้ร่วมกับโมดูลตัวรับอินฟราเรด KY-022 และสามารถเข้ากันได้กับ Arduino, Raspberry Pi, ESP32 และไมโครคอนโทรลเลอร์ชนิดอื่น ๆ แสดงได้ดังรูปที่ 2.26



รูปที่ 2.26 โมดูลส่งสัญญาณอินฟราเรด KY-005 [19]

## 2.14 โมดูลตัวรับสัญญาณอินฟราเรด KY-022

โมดูลตัวรับอินฟราเรด KY-022 ตอบสนองต่อแสง IR 38 kHz โมดูลนี้ประกอบด้วยตัวรับ IR รหัส 1838 ตัวต้านทาน 1 k $\Omega$ , LED และหมุดส่วนในการเชื่อมต่ออุปกรณ์ 3 ขา ระยะทางในการรับสัญญาณ 1800 ซม. สามารถใช้เพื่อรับคำสั่งจากรีโมตคอนโทรล IR จากทีวี สเตอริโอ และอุปกรณ์อื่น ๆ แสดงได้ดังรูปที่ 2.27



รูปที่ 2.27 โมดูลตัวรับอินฟราเรด KY-022 [20]

## 2.15 สัญญาณอินฟราเรด (Infrared)

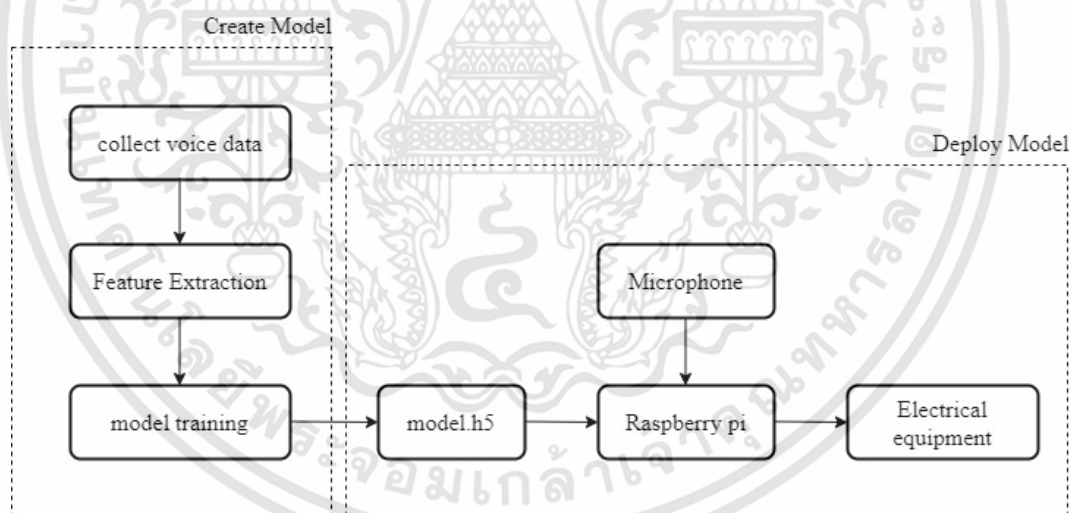
อินฟราเรด (IR) คือ รูปแบบการติดต่อสื่อสารข้อมูลโดยใช้คลื่นแสงอินฟราเรด การสื่อสารด้วยแสงอินฟราเรดจะต้องหันตัวรับและตัวส่งให้ตรงกันและไม่มีสิ่งกีดขวางเส้นสายตา (Line-of-sight) หรือขวางแสงอินฟราเรด การสื่อสารด้วยวิธีนี้ใช้ได้ในระยะทางที่ไม่ไกลนัก ข้อดีของคลื่นอินฟราเรดคือ ใช้พลังงานน้อย แพงวงจรควบคุมราคาต่ำ เรียบง่าย และสามารถเชื่อมต่อกับระบบอื่นได้อย่างรวดเร็ว มีความปลอดภัยในการเรื่องข้อมูลสูง ลักษณะการส่งคลื่น (Directionality of the beam) จะไม่รั่วไปที่เครื่องรับตัวอื่นในขณะที่ส่งสัญญาณ คลื่นแทรกจากเครื่องใช้ไฟฟ้าใกล้เคียงมีน้อย คลื่นแทรกจากเครื่องใช้ไฟฟ้าใกล้เคียงมีน้อย (High noise immunity) ส่วนข้อเสียของอินฟราเรด คือ เครื่องส่ง (Transmitter) และเครื่องรับ (receiver) ต้องอยู่ในแนวเดียวกัน คลื่นจะถูกกั้นโดยวัตถุทั่วไปได้ง่าย เช่นคน กำแพง ต้นไม้ ทำให้สื่อสารไม่ได้ ระยะทางการสื่อสารจะน้อย ประสิทธิภาพจะตกลงถ้าระยะทางมากขึ้น สภาพอากาศ เช่นหมอก แสงอาทิตย์แรง ๆ ฝนและมลภาวะจะมีผลต่อประสิทธิภาพการสื่อสาร และอัตราการส่งข้อมูลจะช้ากว่าแบบใช้สายไฟทั่วไป [21]

## บทที่ 3

### การออกแบบและการจัดทำปฏิญญานิพนธ์

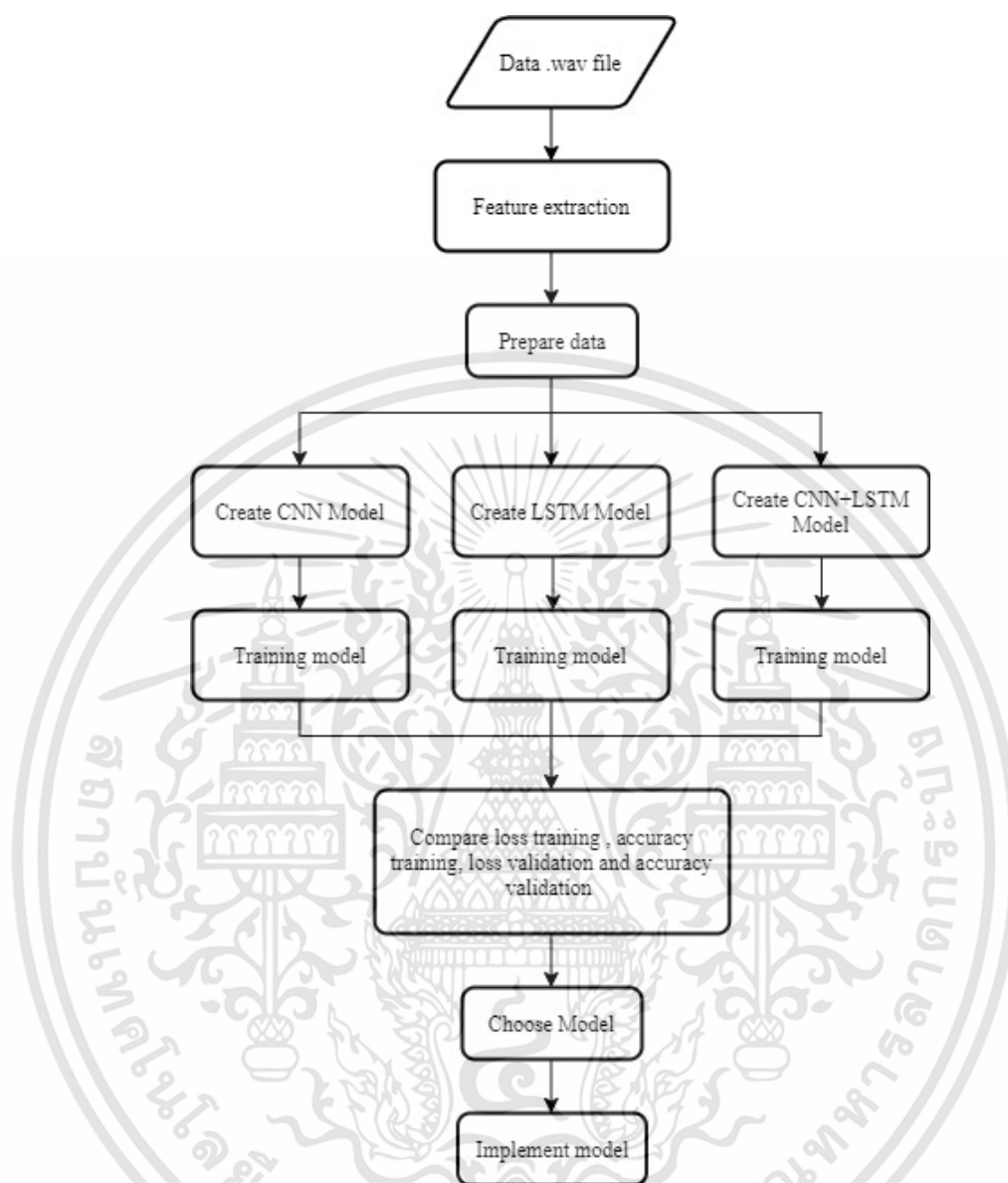
#### 3.1 การออกแบบ

โครงการระบบสั่งการด้วยเสียง (Voice Control System) แบ่งออกเป็น 2 ส่วน ประกอบด้วยส่วนการสร้างโมเดล เป็นส่วนของการนำข้อมูลเสียงมาทำการฝึกฝน (Training) โดยประกอบด้วยขั้นตอนกระบวนการสกัดคุณลักษณะเด่น (Feature Extraction) เพื่อดึงลักษณะเฉพาะของแต่ละสัญญาณเสียงออกมาก่อนเข้าสู่กระบวนการการฝึกฝนให้ได้โมเดลโดยใช้โครงข่ายประสาทเทียม (Neural Network) ในส่วนนี้จะได้โมเดลออกมาเพื่อนำมาใช้งานในส่วนถัดไป ส่วนการประยุกต์ใช้งานโมเดล เป็นส่วนที่นำโมเดลมาประยุกต์ใช้งานในระบบสั่งการอุปกรณ์ไฟฟ้าด้วยเสียง โดยนำมาใช้งานใน Raspberry pi โดยจะรับอินพุตเสียงผ่านไมโครโฟน และนำอินพุตที่ได้ไปป้อนให้กับระบบและทำนายค่าเพื่อนำไปควบคุมอุปกรณ์ไฟฟ้า แสดงดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของโครงการระบบสั่งการด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ภาพรวมของโครงงานในส่วนการสร้างโมเดล

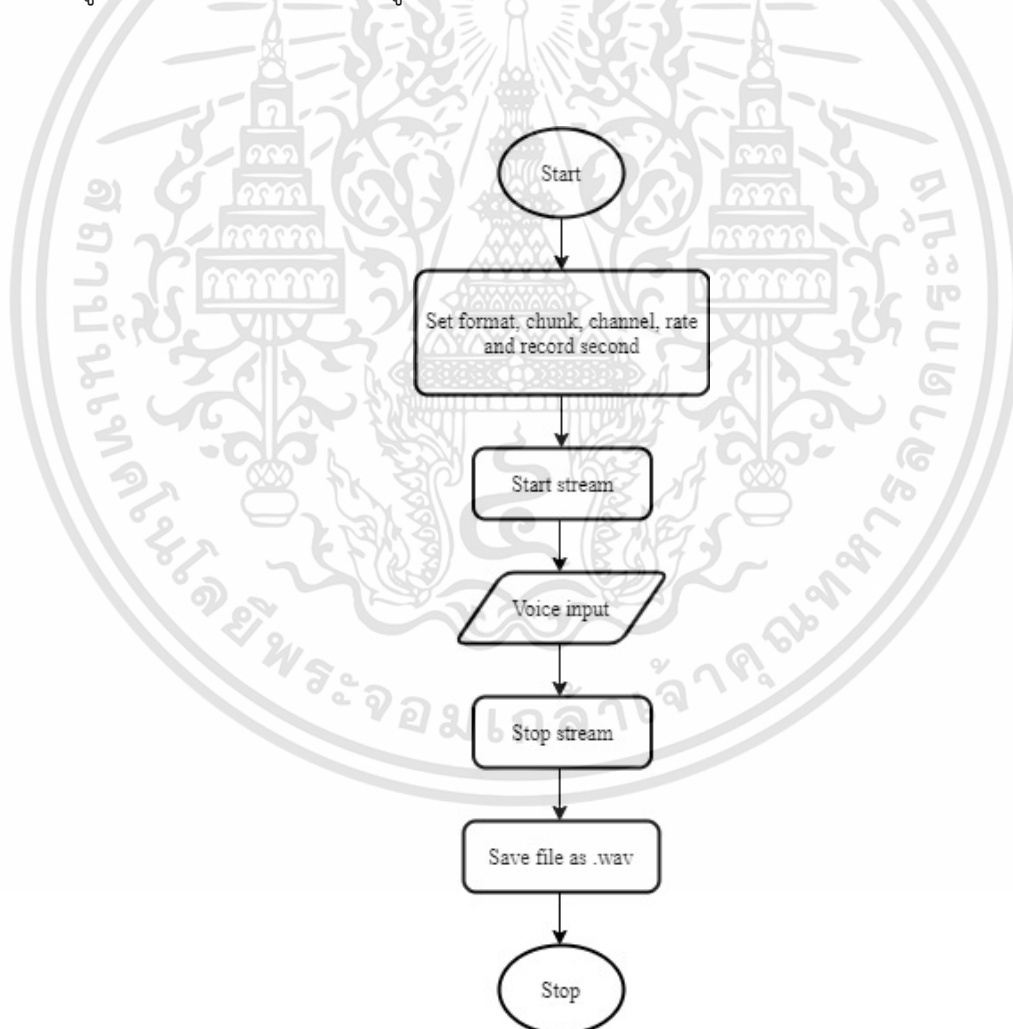
จากรูปที่ 3.2 ในส่วนของการสกัดคุณลักษณะเด่น เป็นขั้นตอนในการแปลงข้อมูลเสียงให้อยู่ในรูปที่สามารถนำไปใช้งานได้กับ Neural Network โดยใช้วิธีการ Mel Frequency Cepstral Coefficient (MFCC) จะทำการสกัดคุณลักษณะเด่นให้อยู่ในรูปของสัมประสิทธิ์เซปตรัมบนความถี่ เมล ก่อนจะนำเข้าไปสู่กระบวนการฝึกฝน

จากรูปที่ 3.2 ในส่วนของกระบวนการฝึกฝน Neural Network จะใช้อินพุตที่ผ่านกระบวนการสกัดคุณลักษณะเด่นมาผ่านกระบวนการรู้จำลักษณะเด่นของเสียงโดยใช้แบบจำลอง Neural Network จะทำการเลือกโมเดลจากแบบจำลองที่มีประสิทธิภาพในการทำนายที่ดีที่สุด ซึ่งจะนำโมเดลนั้นมาประยุกต์ใช้งานในการควบคุมอุปกรณ์ไฟฟ้าด้วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.1 การออกแบบโปรแกรมการจับเก็บข้อมูลเสียง

ในเทคโนโลยีปัญญาประดิษฐ์ ข้อมูลถือเป็นองค์ประกอบสำคัญ ข้อมูลที่มีคุณภาพดีจะทำให้การฝึกฝนเรียนรู้มีประสิทธิภาพที่ดีตามไปด้วย การออกแบบการจับเก็บข้อมูลเสียง (dataset) เริ่มจากการเก็บข้อมูลเสียงจากผู้จัดทำ ซึ่งจะทำให้การเก็บข้อมูลเสียงออกเป็น 4 ส่วน ได้แก่ ส่วนของคำ กระตุ้นระบบให้ระบบเปิดใช้งาน คือคำสั่งเสียงโทรคม ส่วนคำสั่งในการเลือกควบคุมอุปกรณ์ไฟฟ้า ประกอบด้วยจำนวน 2 เสียง ประกอบด้วยคำสั่งเสียงแอร์ และหลอดไฟ ส่วนคำสั่งเสียงควบคุมหลอดไฟ ประกอบด้วยจำนวน 5 คำสั่งเสียง ประกอบไปด้วยคำสั่งเสียงปิดไฟ, เปิดไฟ, สว่างมาก และสว่างน้อย และส่วนคำสั่งเสียงควบคุมเครื่องปรับอากาศ ประกอบด้วยจำนวน 5 คำสั่งเสียง ประกอบด้วยคำสั่งเสียงปิดแอร์, เปิดแอร์, ลดอุณหภูมิ, เพิ่มอุณหภูมิ และเปลี่ยนโหมด หลักการในการเก็บข้อมูลเสียงสามารถอธิบายได้ดังรูปที่ 3.3



รูปที่ 3.3 หลักการเก็บข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 สามารถอธิบายขั้นตอนการทำงานของโปรแกรมได้ดังนี้

1) กำหนดค่าพารามิเตอร์ โดย

ค่า format คือ รูปแบบไฟล์บันทึกข้อมูล

ค่า chunk คือ จำนวนเฟรมในบัฟเฟอร์

ค่า channels คือ จำนวน sample ในแต่ละเฟรม โดยที่ channels = 1 สำหรับ mono และ channels = 2 สำหรับ stereo

ค่า rate คือ จำนวน samples ที่ถูกเก็บใน 1 วินาที

ค่า record\_seconds คือ เวลาในการบันทึกเสียง

2) เมื่อเริ่มการทำงาน โปรแกรมจะเริ่มทำการบันทึกเสียงเป็นเวลาที่กำหนดในพารามิเตอร์ record\_seconds จากนั้นจะทำการหยุดบันทึก ผลลัพธ์ที่ได้จะเป็นไฟล์ข้อมูลเสียง .wav และนำไปเข้าสู่กระบวนการสกัดคุณลักษณะเด่น

### 3.1.2 การออกแบบการสกัดคุณลักษณะเด่นของเสียง (Feature Extraction)

จากรูปที่ 3.2 สามารถอธิบายกระบวนการการสกัดคุณลักษณะเด่นของเสียงได้ในหลักการของบทที่ 2 ในหัวข้อที่ 2.5 กระบวนการสกัดคุณลักษณะเด่นมีจุดประสงค์ในการดึงลักษณะเฉพาะของแต่ละสัญญาณออกมา ซึ่งผลลัพธ์จากการนำไปผ่านการสกัดคุณลักษณะเด่นโดยใช้วิธีการ Mel Frequency Cepstral Coefficient (MFCC) จะได้ค่าสัมประสิทธิ์เซปตริ้มบนความถี่เมล และทำการเขียนโปรแกรมให้จัดเก็บข้อมูลอยู่ในรูปเมทริกซ์โดยใช้ไฟล์ข้อมูล .npy สำหรับนำไปเป็นค่าอินพุตให้ระบบ

### 3.1.3 การเตรียมข้อมูล (Prepare Data)

ในขั้นตอนการเตรียมข้อมูล จะนำข้อมูลที่ได้จากกระบวนการสกัดคุณลักษณะเด่นมาทำการแบ่งข้อมูลออกเป็น 3 ส่วน ประกอบด้วยชุดข้อมูลฝึกฝน (Training dataset) ชุดข้อมูล Validation และชุดข้อมูลทดสอบ (Test dataset) โดยมีอัตราส่วนอยู่ที่ 80 เปอร์เซ็นต์ 10 เปอร์เซ็นต์ และ 10 เปอร์เซ็นต์ ตามลำดับ สำหรับชุดข้อมูลฝึกฝน ใช้สำหรับเป็นอินพุตในการฝึกฝนกระบวนการ Neural Network สำหรับชุดข้อมูล Validation ใช้สำหรับทดสอบหลังจากฝึกฝนโมเดลเสร็จว่าโมเดลมีประสิทธิภาพดีเท่าไร และเพื่อเปรียบเทียบโมเดลว่าครั้งไหนทำงานได้มีประสิทธิภาพมากกว่ากัน ส่วนชุดข้อมูลทดสอบใช้สำหรับทดสอบหลังจากได้โมเดลที่ดีที่สุดมาแล้วว่าโมเดลจะสามารถทำงานได้ดีเท่าไรกับข้อมูลที่ยังไม่เคยเรียนรู้มาก่อน

จากนั้นจะทำการจับคู่ไฟล์เสียงกับป้ายกำกับ (Label) และทำการเก็บไว้ในไฟล์ .csv จากรูปที่ 3.4 แสดงป้ายกำกับของโมเดลส่วนกระตุ้นการทำงานของระบบ จากรูปที่ 3.5 แสดงป้ายกำกับของโมเดลส่วนควบคุมหลอดไฟ และจากรูปที่ 3.6 แสดงป้ายกำกับของโมเดลส่วนควบคุมเครื่องปรับอากาศ

	A	B
1	0	Air
2		
3	1	LoatFai
4		
5	2	ThoRaKhom
6		

รูปที่ 3.4 ป้ายกำกับคำสั่งของโมเดลส่วนกระตุ้นการทำงานของระบบ

	A	B
1	0	PidFai
2		
3	1	PoetFai
4		
5	2	SawangMak
6		
7	3	SawangNoi
8		

รูปที่ 3.5 ป้ายกำกับคำสั่งของโมเดลส่วนควบคุมหลอดไฟ

	A	B
1	0	LotUnhaphum
2		
3	1	PhoemUnhaphum
4		
5	2	PidAir
6		
7	3	PlianMot
8		
9	4	PoetAir
10		

รูปที่ 3.6 ป้ายกำกับคำสั่งของโมเดลส่วนควบคุมเครื่องปรับอากาศ

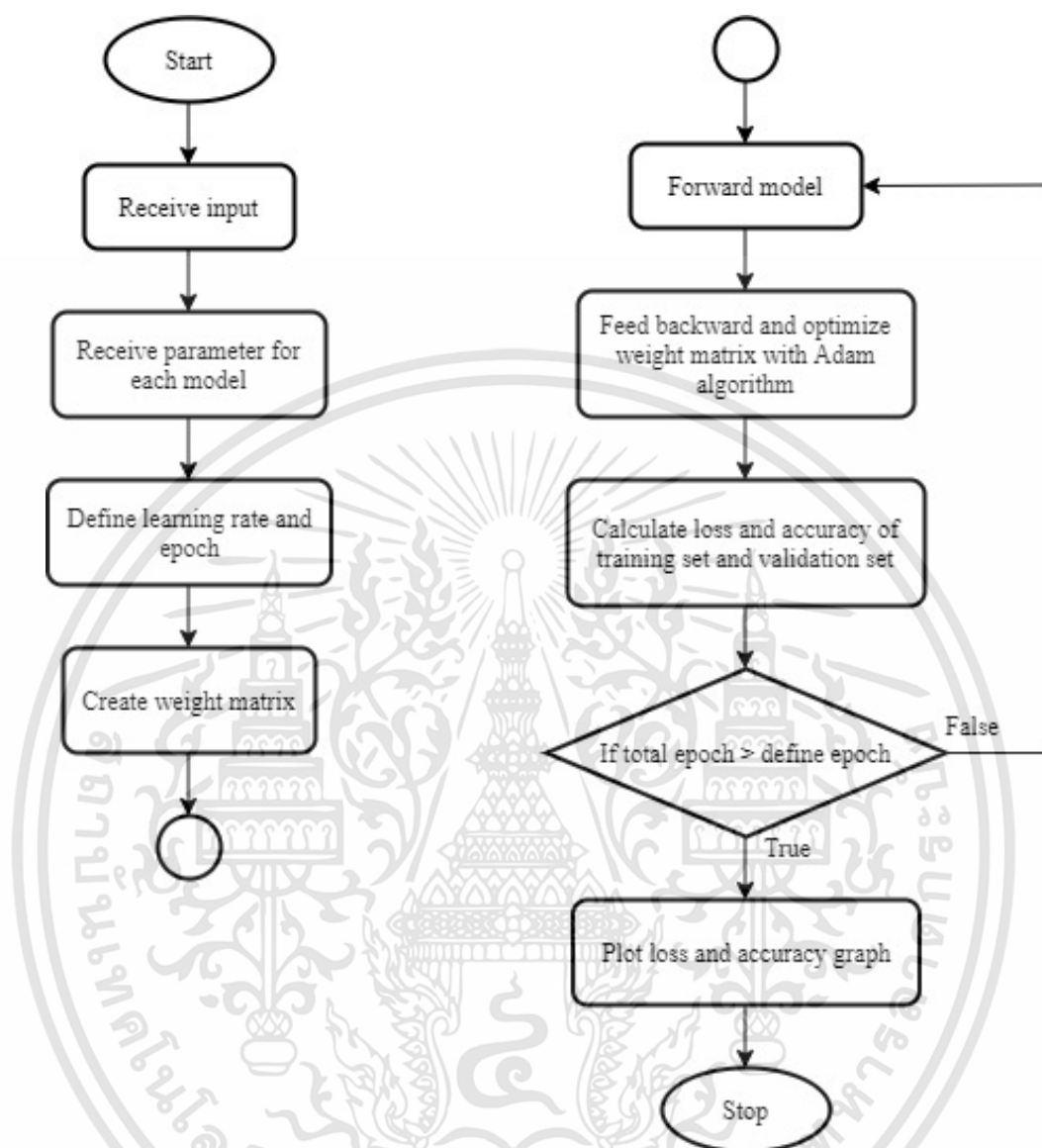
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.4 การออกแบบกระบวนการฝึกฝน (Training)

3.1.4.1 ค่าไฮเปอร์พารามิเตอร์ (Hyperparameter) ที่ใช้ในการฝึกฝนข้อมูล  
ไฮเปอร์พารามิเตอร์ (Hyperparameter) คือพารามิเตอร์ที่มีค่าควบคุมกระบวนการฝึกฝนข้อมูล สามารถกำหนดเองได้ก่อนที่โมเดลจะทำการเรียนรู้ เช่น ขนาด kernel หรือขนาด filter ในแบบจำลอง Convolution layers, ขนาด Pooling หรือจำนวนของ hidden layers ใน Neural network ในส่วนนี้เราจะทำการหารูปแบบของค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมสำหรับโมเดล โดยจะแบ่งกระบวนการฝึกฝนเพื่อเปรียบเทียบค่าไฮเปอร์พารามิเตอร์ออกเป็น 3 แบบจำลอง ได้แก่ แบบจำลอง Convolution Neural Network (CNN) แบบจำลอง Long shot-term memory (LSTM) และแบบจำลอง CNN+LSTM

3.1.4.2 ออกแบบการฝึกฝนข้อมูลในแต่ละแบบจำลองของ Neural Network

จากหัวข้อที่ 3.1.4.1 จะทำการเลือกรูปแบบของค่าไฮเปอร์พารามิเตอร์ที่ทำให้โมเดลมีความผิดพลาดต่ำ และมีค่าความแม่นยำมากที่สุดในแต่ละแบบจำลองมาเปรียบเทียบว่าแบบจำลองใดที่มีประสิทธิภาพและเหมาะสมกับการทำงานของระบบสั่งการด้วยเสียงมากที่สุด ซึ่งในกระบวนการฝึกฝน Neural Network สามารถอธิบายภาพรวมการทำงานของ Neural Network ได้ดังรูปที่ 3.7 ในกระบวนการฝึกฝน Neural Network ของโครงงานจะทำการฝึกฝนแบบจำลอง Convolution Neural Network (CNN) แสดงดังรูปที่ 3.8 แบบจำลอง Long Short-Term Memory (LSTM) แสดงดังรูปที่ 3.9 และแบบจำลอง CNN+LSTM แสดงดังรูปที่ 3.10



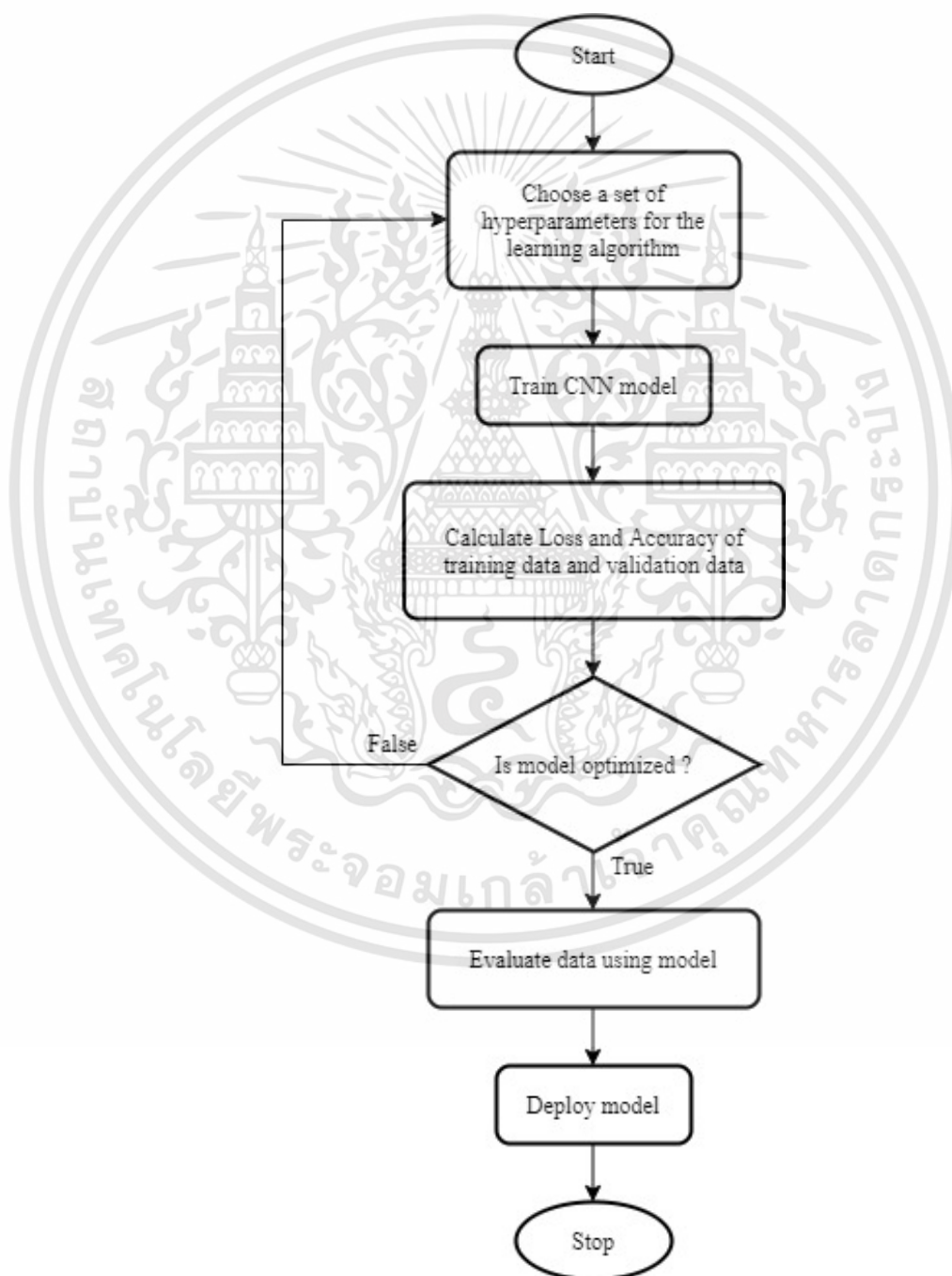
รูปที่ 3.7 ภาพรวมการทำงานของ Neural Network

จากรูปที่ 3.7 แสดงรายละเอียดภาพรวมการทำงานภายในกระบวนการฝึกฝน Neural Network ค่าอินพุตที่ป้อนเข้าสู่ระบบ ประกอบด้วยค่า

1. timesteps เป็นค่าความยาวของลำดับอินพุตที่ป้อนเข้าสู่โครงข่ายประสาทเทียม
2. frame width เป็นค่าความกว้างของเฟรมข้อมูล
3. number of feature เป็นจำนวนค่าของลักษณะคุณสมบัติของข้อมูลที่ต้องการนำไปฝึกฝน ในโครงงานนี้คือค่าสัมประสิทธิ์เซปตรัมบนความถี่เมลจำนวน 40 ค่า
4. colour scale เป็นจำนวนช่องสีของอินพุตเลเยอร์ กำหนดให้มีค่าเท่ากับ 1 คือมีช่องสีเดียวในอินพุตเลเยอร์เป็นสเกลสีเทา

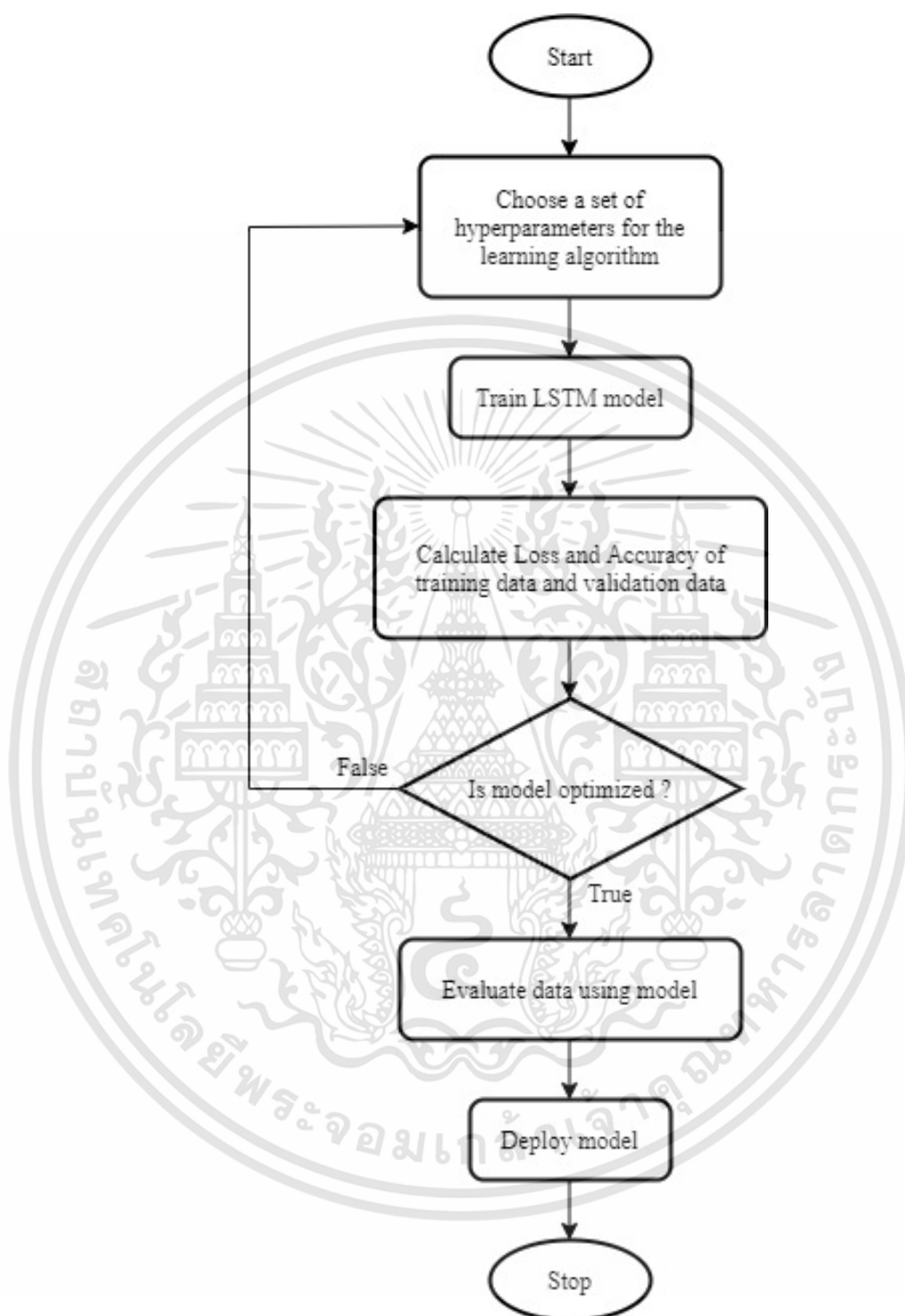
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 ภาพรวมการทำงานของ Neural Network จะรับอินพุตเข้าสู่กระบวนการฝึกฝนโดยกำหนดค่า learning rate มีค่าเป็น 0.001 และกำหนดจำนวนการฝึกฝน (epoch) จำนวน 30 รอบ และทำการคำนวณหาค่าความผิดพลาด (Loss) และความแม่นยำ (Accuracy) ของชุดข้อมูลฝึกฝน และชุดข้อมูล Validation จากนั้นทำการพล็อตกราฟค่าความผิดพลาดและความแม่นยำของโมเดล



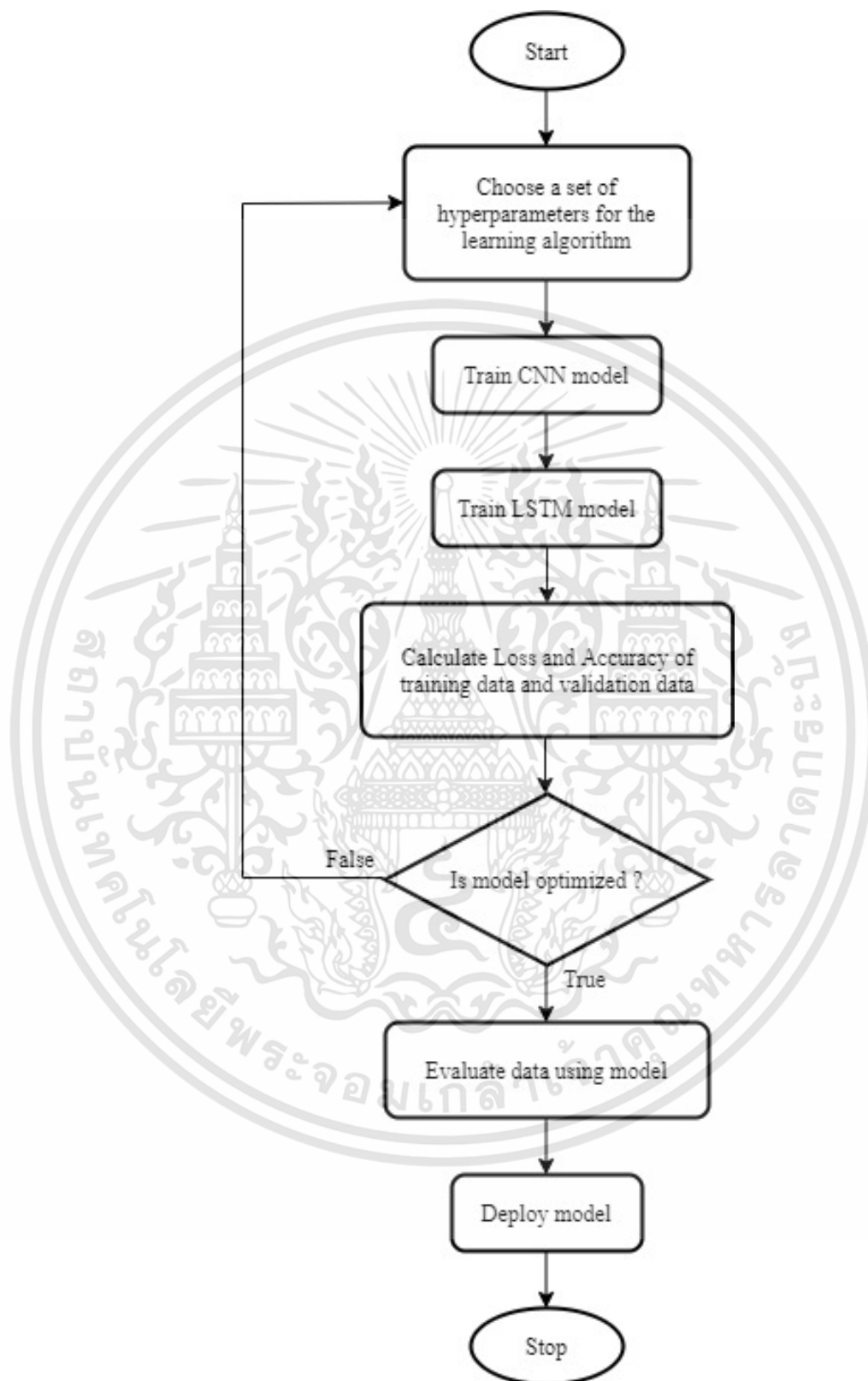
รูปที่ 3.8 ผังการทำงานแบบจำลอง CNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ผังการทำงานแบบจำลอง LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



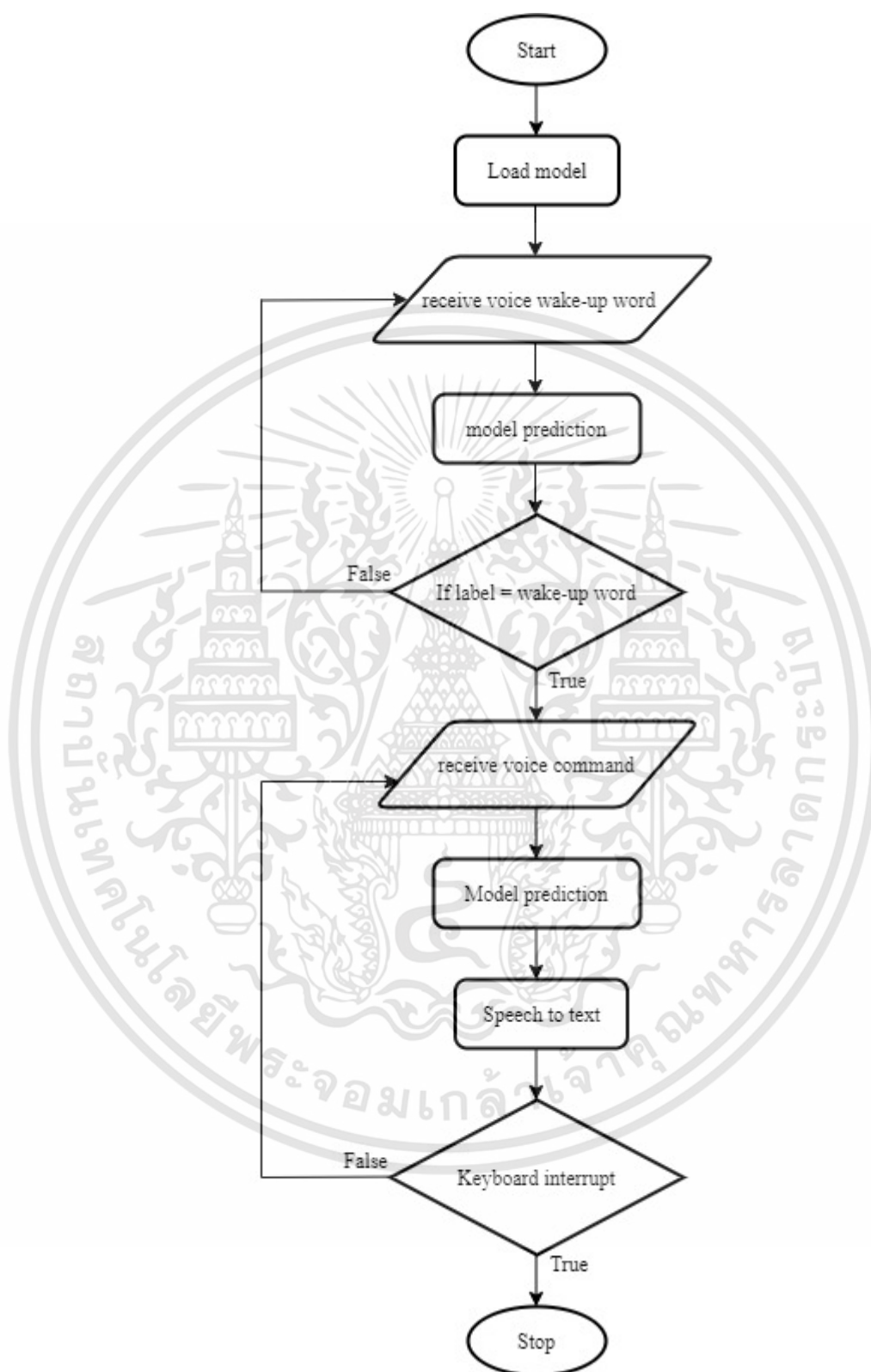
รูปที่ 3.10 ผังการทำงานแบบจำลอง CNN+LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8, 3.9 และ 3.10 แสดงการทำงานของแบบจำลอง Convolution Neural Network (CNN) แบบจำลอง Long short-term memory (LSTM) และแบบจำลอง CNN+LSTM ตามลำดับ โดยผู้จัดทำได้ออกแบบแบบจำลองตามหลักการในบทที่ 2 หัวข้อ 2.6 และ 2.7 ตั้งค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมสำหรับการฝึกฝนโมเดลที่ได้จากหัวข้อ 3.1.4.1 และนำแบบจำลองมาเปรียบเทียบค่าความผิดพลาด (Loss) และความแม่นยำ (Accuracy) เพื่อหาแบบจำลองที่ดีที่สุด และจัดเก็บโมเดลเพื่อนำไปประยุกต์ใช้ในระบบสั่งการด้วยเสียง (Voice control system)

### 3.1.5 การออกแบบการทดสอบการใช้งานโมเดล

ในส่วนการทดสอบการใช้งานโมเดล จะนำโมเดลมาประยุกต์ใช้ในเทคโนโลยีเปลี่ยนจากข้อมูลเสียงเป็นข้อความ โครงสร้างการใช้งานโมเดลสามารถอธิบายการทำงานได้ดังรูปที่ 3.11 ซึ่งในส่วนการทดสอบการใช้โมเดล จะทำการทดสอบโดยจะรับอินพุตเสียงแบบเรียลไทม์ ซึ่งถ้าอินพุตเสียงที่ป้อนตรงกับ Wake-up word คือชุดคำสั่ง “โทรคม” จะทำการรับอินพุตเสียงเพื่อนำไปทำนายค่าต่อไป และเมื่อมีการรบกวนจากคีย์บอร์ดการรับอินพุตเสียงจะหยุดการทำงาน



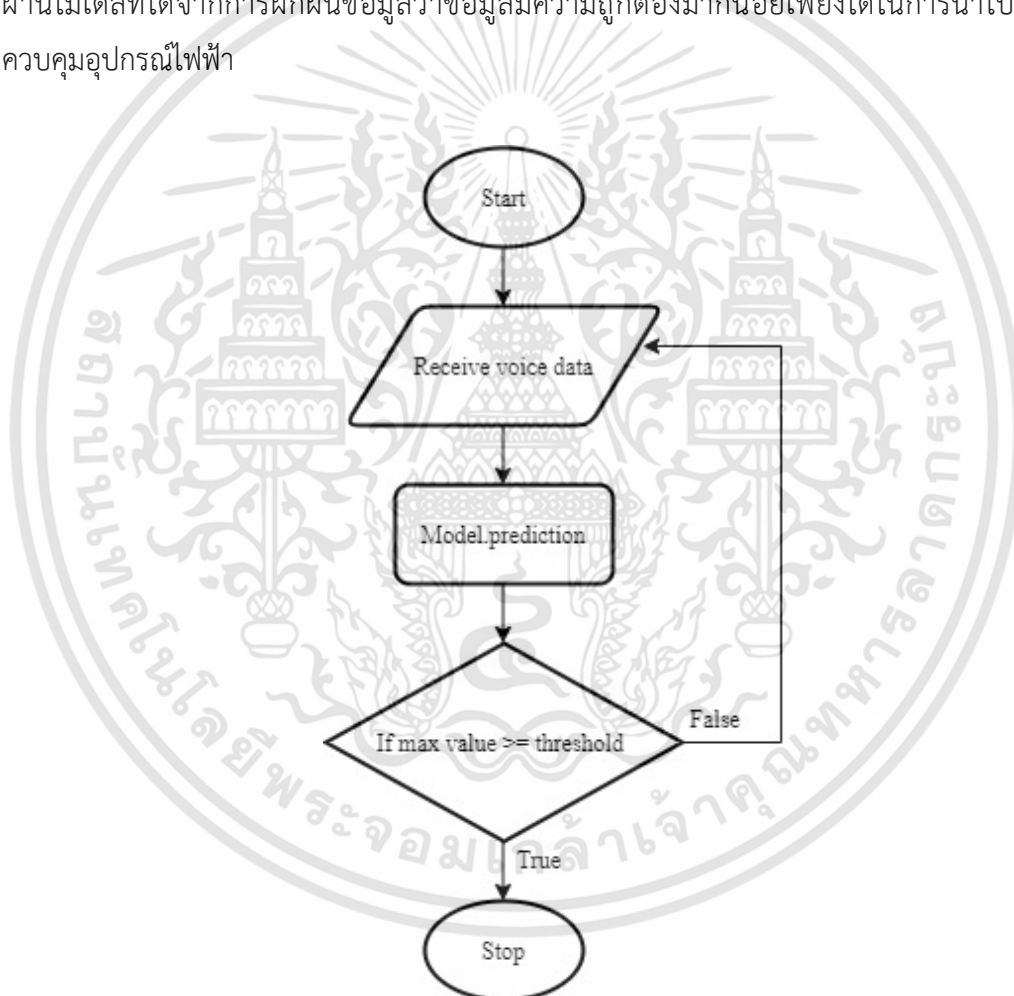
รูปที่ 3.11 หลักการทำงานการใช้งานโมเดลในเทคโนโลยีเปลี่ยนจากเสียงเป็นข้อความ (Speech to text)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.6 การออกแบบการใช้งานโมเดลในระบบสั่งการด้วยเสียง

#### 3.1.6.1 ออกแบบการตั้งค่าตัดสินใจให้กับระบบ

ระบบสั่งการด้วยเสียง (Voice control system) เป็นระบบที่ให้ผู้ใช้งานป้อนคำสั่งเสียงเพื่อนำคำสั่งเสียงไปทำนายค่า และนำไปควบคุมอุปกรณ์ไฟฟ้า ซึ่งหากผู้ใช้งานไม่ได้ป้อนคำสั่งหรือหากคำสั่งที่ถูกป้อนเข้าสู่ระบบไม่ใช่คำสั่งเสียงที่เรากำหนด ระบบจะอย่างไรในการตัดสินใจรับคำสั่งเสียงมาใช้งาน เพื่อให้ระบบสั่งการด้วยเสียงทำงานอย่างมีประสิทธิภาพ ระบบจึงจำเป็นต้องมีค่าตัดสินใจ (Threshold) เพื่อเป็นเงื่อนไขในการประเมินผลลัพธ์ที่ได้จากการทำนายค่าผ่านโมเดลที่ได้จากการฝึกฝนข้อมูลว่าข้อมูลมีความถูกต้องมากน้อยเพียงใดในการนำไปใช้ในการควบคุมอุปกรณ์ไฟฟ้า

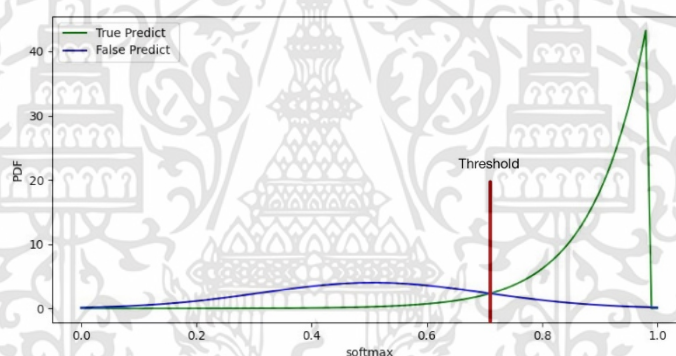


รูปที่ 3.12 บล็อกไดอะแกรมการตั้งค่าตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.12 เมื่อคำสั่งเสียงถูกป้อนเข้าสู่ระบบ ระบบจะนำคำสั่งเสียงมา ทำนายค่าและนำค่าเอาต์พุตที่ได้จากการทำนายมาผ่านเงื่อนไขการตัดสินใจของระบบ หากไม่ผ่านเงื่อนไขการตัดสินใจ ระบบจะไม่ทำงานและกลับไปเริ่มต้นการทำงานใหม่อีกครั้ง ซึ่งจะช่วยให้สามารถตัดสินใจว่าจะนำคำสั่งไปใช้งานต่อหรือตัดคำสั่งนั้นทิ้งไปใน กรณีที่คำสั่งไม่ใช่คำสั่งเสียงที่ผู้ใช้งาน กำหนด หรือคำสั่งเสียงที่ทำนายค่าออกมาผิดพลาด ช่วยให้ระบบสั่งการด้วยเสียงสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น

วิธีการหาค่าตัดสินใจ (Threshold) ให้กับระบบสั่งการด้วยเสียง จะทำการพล็อตกราฟ Normal distribution ของค่าความน่าจะเป็นที่ได้จากฟังก์ชัน softmax จะประกอบด้วยกราฟ False prediction และ กราฟ True prediction ในแต่ละชุดคำสั่งเสียง และจะทำการเลือกค่าตัดสินใจจากจุดตัดระหว่าง 2 กราฟ ดังรูปที่ 3.13



รูปที่ 3.13 วิธีการหาค่าตัดสินใจ

### 3.1.6.2 ออกแบบการใช้งานโมเดลในระบบสั่งการด้วยเสียง

การใช้งานในระบบสั่งการด้วยเสียงจะแบ่งการทำงานของโมเดลออกเป็น 3 ส่วน ประกอบด้วย

- ส่วนที่ 1 โมเดลส่วนการกระตุ้นระบบ และการระบุอุปกรณ์ไฟฟ้า

โมเดลส่วนแรกจะเป็นส่วนในการกระตุ้นการทำงานของระบบ และการระบุอุปกรณ์ไฟฟ้าที่ต้องการสั่งการ ซึ่งประกอบด้วยคำสั่งเสียงโทรคม เป็นคำสั่งเสียงในการกระตุ้นระบบให้เปิดใช้งานรับคำสั่งเสียงต่อไป คือคำสั่งเสียงในการระบุอุปกรณ์ไฟฟ้า ได้แก่คำสั่งเสียงแอร์ เป็นคำสั่งเสียงในการระบุอุปกรณ์ไฟฟ้าเป็นเครื่องปรับอากาศ และคำสั่งเสียงหลอดไฟ เป็นคำสั่งเสียงในการระบุอุปกรณ์ไฟฟ้าเป็นหลอดไฟ ซึ่งระบบจะมีการแจ้งเตือนผู้ใช้งานให้ระบุอุปกรณ์ที่ต้องการ เมื่อผู้ใช้ระบุอุปกรณ์ไฟฟ้าเสร็จเรียบร้อยแล้ว ระบบจะมีการแจ้งเตือนให้ผู้ใช้ระบุคำสั่งที่ต้องการควบคุมอีกครั้ง และระบบจะเข้าสู่การทำงานของโมเดลในอุปกรณ์นั้น ๆ ที่ผู้ใช้ระบุ

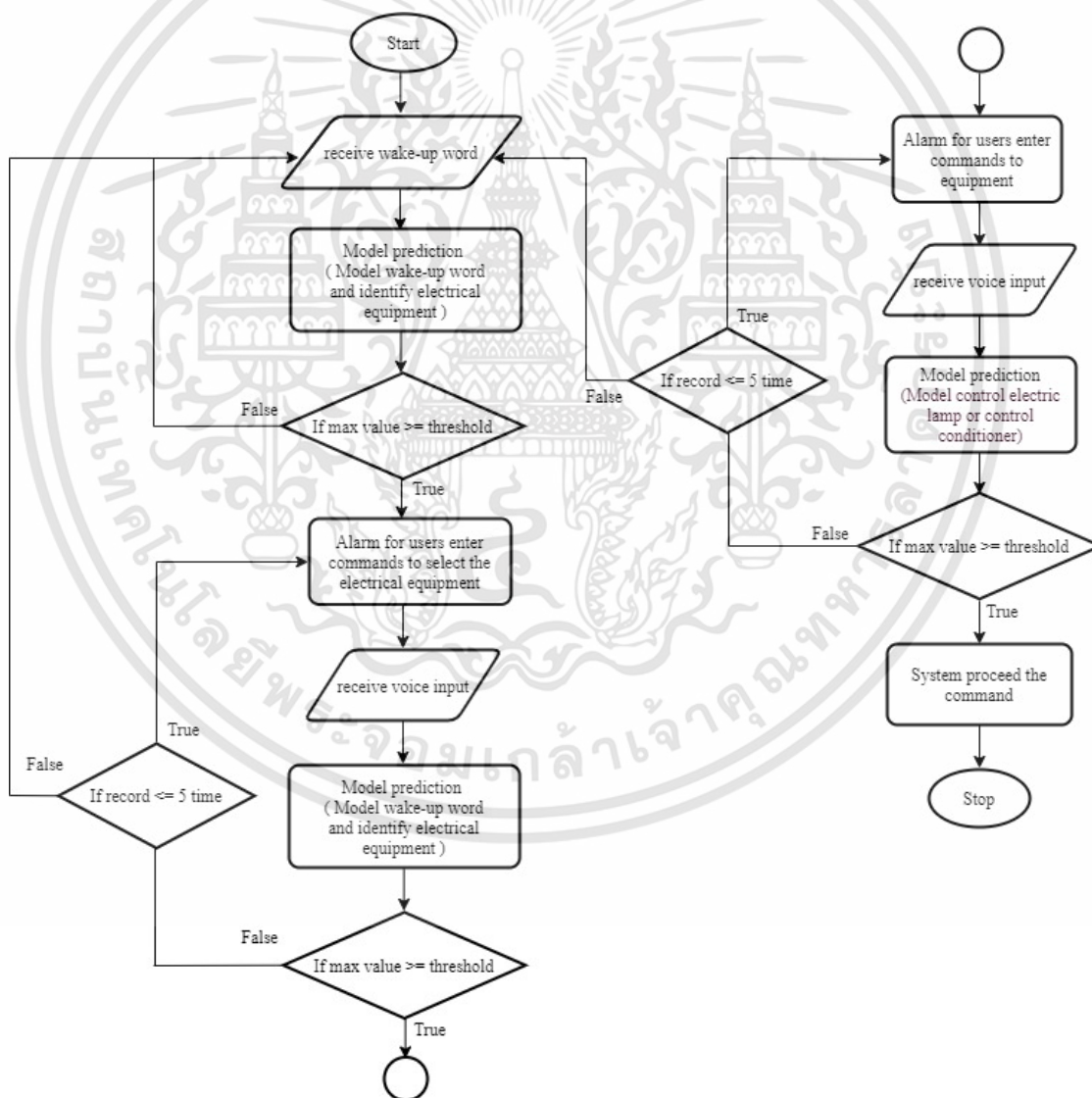
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนที่ 2 โมเดลเครื่องปรับอากาศ

เมื่อผู้ใช้งานระบุอุปกรณ์เป็นเครื่องปรับอากาศ ระบบจะเข้าสู่โมเดลและรับคำสั่งเสียงมาทำนายค่า ซึ่งคำสั่งเสียงในโมเดลเครื่องปรับอากาศ ประกอบด้วย คำสั่งเสียงปิดแอร์, เปิดแอร์, เพิ่มอุณหภูมิ, ลดอุณหภูมิ และเปลี่ยนโหมด

- ส่วนที่ 3 โมเดลหลอดไฟ

เมื่อผู้ใช้งานระบุอุปกรณ์เป็นหลอดไฟ ระบบจะเข้าสู่โมเดลและรับคำสั่งเสียงมาทำนายค่า ซึ่งคำสั่งเสียงในการควบคุมหลอดไฟ ประกอบด้วย ปิดไฟ, เปิดไฟ, สว่างมาก และสว่างน้อย สามารถแสดงการทำงานของระบบสั่งการด้วยเสียงได้ ดังรูปที่ 3.14



รูปที่ 3.14 บล็อกไดอะแกรมแสดงระบบการทำงานของระบบสั่งการด้วยเสียง

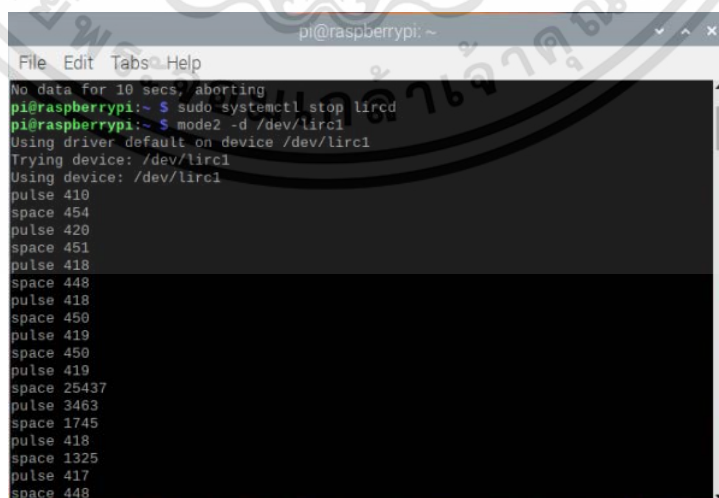
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.14 เมื่อป้อนคำสั่งเสียงเข้าสู่ระบบ ระบบจะนำข้อมูลที่ถูกป้อนไปทำนายค่าผ่านโมเดลในส่วนที่1 เมื่อผลลัพธ์การทำนายค่าเป็นคำสั่งเสียงโทรคม ซึ่งเป็นคำกระตุ้น (Wake-up word) ในการเริ่มการทำงานของระบบ โดยระบบจะทำงานต่อไปเมื่อค่าความน่าจะเป็นที่ได้จากค่าทำนายเท่ากับหรือมากกว่าค่าตัดสินใจ (threshold) หากไม่ผ่านเงื่อนไขจะทำการรับคำสั่งเสียงใหม่ เมื่อผลลัพธ์ที่ได้ผ่านเงื่อนไข ระบบจะทำการแจ้งเตือน (Alarm) ให้ผู้ใช้งานป้อนคำสั่งเสียงถัดไปคือคำสั่งเสียงในการระบุเครื่องใช้ไฟฟ้า ระบบจะนำคำสั่งเสียงที่ผู้ใช้งานป้อนไปทำนายค่า ซึ่งระบบจะมีเงื่อนไขการทำงานเช่นเดียวกันกับการป้อนคำกระตุ้น หากไม่ผ่านเงื่อนไขจะทำการแจ้งเตือนให้ผู้ใช้งานป้อนคำสั่งเสียงในการระบุเครื่องใช้ไฟฟ้าอีกครั้ง ซึ่งหากป้อนคำสั่งเสียงแล้วไม่ผ่านเงื่อนไขจำนวน 5 ครั้ง ระบบจะกลับไปเริ่มต้นการทำงานในการรับคำกระตุ้นใหม่ตั้งแต่แรก

ซึ่งหากการทำนายค่าผ่านเงื่อนไข ระบบจะทำการแจ้งเตือนให้ผู้ใช้งานป้อนคำสั่งเสียงในการควบคุมอุปกรณ์ที่ระบุ ซึ่งระบบจะมีเงื่อนไขการทำงานเช่นเดียวกับในส่วนของการกระตุ้นระบบและระบุอุปกรณ์เครื่องใช้ไฟฟ้า ซึ่งหากป้อนคำสั่งเสียงแล้วไม่ผ่านเงื่อนไขจำนวน 5 ครั้ง ระบบจะกลับไปเริ่มต้นการทำงานในการรับคำกระตุ้นใหม่ตั้งแต่แรกเช่นกัน

### 3.1.6.3 การออกแบบการควบคุมอุปกรณ์โดยใช้งานผ่านโมดูลอินฟราเรด

ในส่วนการควบคุมอุปกรณ์ ได้แก่ แอร์ และ หลอดไฟ จะทำการควบคุมผ่านสัญญาณอินฟราเรด โดยที่แต่ละอุปกรณ์จะมีชุดสัญญาณที่ใช้ในการควบคุมแต่ละคำสั่งที่ต่างกัน จึงจำเป็นต้องมีการเก็บชุดสัญญาณแต่ละชุดของอุปกรณ์แต่ละตัว ด้วยโมดูลรับสัญญาณอินฟราเรด KY-022 และ ไลบรารี Linux Infrared Remote Control (LIRC) สามารถแสดงตัวอย่างชุดสัญญาณได้ดังรูปที่ 3.15



```

pi@raspberrypi: ~
File Edit Tabs Help
No data for 10 secs, aborting
pi@raspberrypi:~$ sudo systemctl stop lircd
pi@raspberrypi:~$ mode2 -d /dev/lirc1
Using driver default on device /dev/lirc1
Trying device: /dev/lirc1
Using device: /dev/lirc1
pulse 410
space 454
pulse 420
space 451
pulse 418
space 448
pulse 418
space 450
pulse 419
space 450
pulse 419
space 25437
pulse 3463
space 1745
pulse 418
space 1325
pulse 417
space 448

```

รูปที่ 3.15 ตัวอย่างชุดสัญญาณของการสั่งงานปิดไฟ ของหลอดไฟที่ทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.15 นำชุดสัญญาณมาบันทึกค่าเป็นรีโมทเพื่อควบคุม แอร์ และ หลอดไฟด้วย irsend บนไลบรารี Linux Infrared Remote Control (LIRC) แสดงดังรูปที่ 3.16

```

pi@raspberrypi:~$ irsend LIST daikin ""
0000000000000001 off
0000000000000002 on
0000000000000003 22C
0000000000000004 23C
0000000000000005 24C
0000000000000006 25C
0000000000000007 26C
0000000000000008 27C
0000000000000009 28C
000000000000000a power1
000000000000000b power2
pi@raspberrypi:~$ irsend LIST light ""
0000000000ffe01f KEY_OPEN
0000000000ff609f KEY_CLOSE
0000000000ffa05f KEY_BRIGHTNESSUP
0000000000ff20df KEY_BRIGHTNESSDOWN
pi@raspberrypi:~$

```

รูปที่ 3.16 ชุดคำสั่งของรีโมทควบคุม แอร์ และ หลอดไฟ

จากรูปที่ 3.16 แสดงชุดคำสั่งรีโมทที่จะนำไปสั่งใช้งานอุปกรณ์ แอร์ และ หลอดไฟ โดยมีชุดคำสั่งดังต่อไปนี้

1) รีโมทควบคุมแอร์ โดยมีคีย์คำสั่งดังต่อไปนี้

- Off เป็นคำสั่งเพื่อใช้ในการปิดแอร์
- On เป็นคำสั่งเพื่อใช้ในการเปิดแอร์
- 22C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 22 องศาเซลเซียส
- 23C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 23 องศาเซลเซียส
- 24C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 24 องศาเซลเซียส
- 25C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 25 องศาเซลเซียส
- 26C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 26 องศาเซลเซียส
- 27C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 27 องศาเซลเซียส
- 28C เป็นคำสั่งเพื่อใช้ในการเปลี่ยนอุณหภูมิเป็น 28 องศาเซลเซียส
- Power1 เป็นคำสั่งเพื่อใช้ในการเปลี่ยนโหมดเป็นโหมดที่1
- Power2 เป็นคำสั่งเพื่อใช้ในการเปลี่ยนโหมดเป็นโหมดที่2

2) รีโมทควบคุมหลอดไฟ โดยมีคีย์คำสั่งดังต่อไปนี้

- KEY\_OPEN เป็นคำสั่งเพื่อใช้ในการเปิดไฟ
- KEY\_CLOSE เป็นคำสั่งเพื่อใช้ในการปิดไฟ
- KEY\_BRIGHTNESSUP เป็นคำสั่งเพื่อใช้ในการเพิ่มความสว่างของหลอดไฟ
- KEY\_BRIGHTNESSDOWN เป็นคำสั่งเพื่อใช้ในการลดความสว่างของหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 เครื่องมือที่ใช้ในการทดลอง

### 3.2.1 Laptop

ใช้สำหรับเปิดใช้งานโปรแกรม Visual studio code เพื่อใช้สำหรับศึกษากระบวนการ  
ออกแบบปัญญาประดิษฐ์ (AI) มีสเปกและคุณลักษณะดังรูปที่ 3.17 ถึง 3.19



รูปที่ 3.17 Laptop HP Pavilion x360

Laptop HP Pavilion x360 มีสเปกดังนี้

- 1) CPU : Intel(R) Core(TM) i7-10510U CPU @ 1.80GHZ
- 2) GPU : NVIDIA Geforce MX250
- 3) RAM : 16.0 GB (15.8 GB usable)

รูปที่ 3.18 Lenovo Ideapad 320S

Laptop Lenovo Ideapad 320S มีสเปกดังนี้

- 1) CPU : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHZ
- 2) GPU : Intel(R) UHD Graphics 620
- 3) RAM : 8.0 GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 Desktop

Desktop มีสเปกดังนี้

- 1) CPU : intel(R) core(TM) i5-9400F CPU @ 2.90GHz
- 2) GPU : Nvidia Geforce GTX 1660
- 3) RAM : 16.0GB

### 3.2.2 Raspberry Pi 4 model B

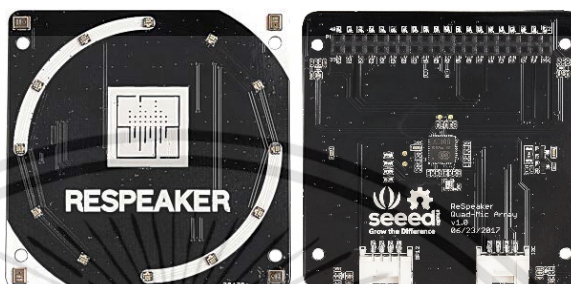
ใช้สำหรับนำโมเดลไปใช้งานในการควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 3.20 Raspberry Pi 4 model B

### 3.2.3 Respeaker 4-mic array

ใช้สำหรับเป็นไมโครโฟนเชื่อมต่อกับ Raspberry Pi รับอินพุตที่ถูกป้อนจากผู้ใช้งานไป ทำนายค่าผ่านโมเดลบน Raspberry Pi



รูปที่ 3.21 Respeaker 4-mic array

### 3.2.4 IR Transmitter รหัส KY-005

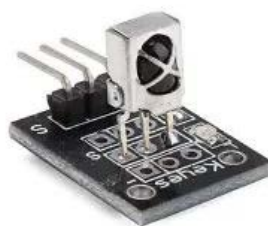
ใช้สำหรับส่งค่า packet อินฟราเรดของคำสั่งควบคุมต่าง ๆ ไปยังอุปกรณ์ไฟฟ้า



รูปที่ 3.22 IR Transmitter

### 3.2.5 IR Receiver รหัส KY-022

ใช้สำหรับการรับค่า packet อินฟราเรดจากรีโมตคอนโทรลอุปกรณ์ที่ส่งสัญญาณผ่านอินฟราเรด



รูปที่ 3.23 IR Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.6 หลอดไฟ

หลอดไฟที่สามารถปรับความสว่างได้ ซึ่งใช้สำหรับสั่งใช้งานด้วยเสียงในการควบคุมหลอดไฟ



รูปที่ 3.24 หลอดไฟ

### 3.2.7 เครื่องปรับอากาศ

ใช้สำหรับสั่งใช้งานด้วยเสียงในการควบคุมเครื่องปรับอากาศ



รูปที่ 3.25 เครื่องปรับอากาศ

### 3.2.8 ลำโพง

ใช้สำหรับแสดงเสียงเตือน (Alarm) ให้ผู้ใช้งานป้อนคำสั่งเสียงที่ต้องการควบคุมอุปกรณ์



รูปที่ 3.26 ลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การจัดเก็บผลการทดลอง

#### 3.3.1 การจัดเก็บข้อมูลเสียง

การจัดเก็บผลการทดลองในส่วนของการจัดเก็บข้อมูลเสียง สำหรับนำไปใช้ในการสกัดคุณลักษณะเด่น และนำไปใช้ในการฝึกฝนแบบจำลอง Neural Network ข้อมูลจะประกอบด้วยข้อมูล 2 ส่วน ประกอบด้วยเสียงที่บันทึกเสียงด้วยโปรแกรมจัดเก็บข้อมูลเสียงของผู้จัดทำ เป็นข้อมูลไฟล์ .wav ข้อมูลเสียงมีระยะเวลา 3 วินาที และข้อมูลเสียงที่นำมาเปลี่ยนระดับเสียงและความเร็วเสียง ประกอบด้วยข้อมูลเสียงที่นำมาเปลี่ยนให้มีระดับเสียงสูงขึ้น ระดับเสียงต่ำลง และข้อมูลเสียงที่นำมาเปลี่ยนให้มีความเร็วมากขึ้น และความเร็วลดลง โดยใช้ไลบรารี librosa ในการเปลี่ยนระดับเสียง โดยใช้คำสั่งดังนี้ คำสั่ง librosa.effects.pitch\_shift เป็นกระบวนการเปลี่ยนระดับเสียงให้สูงขึ้นหรือต่ำลง ซึ่งจะไม่ส่งผลกระทบต่อความเร็วของเสียงหรือมีระยะเวลาของเสียงเท่าเดิม และคำสั่ง librosa.effects.time\_stretch เป็นกระบวนการเปลี่ยนความเร็วหรือระยะเวลาของเสียง

ในการจัดเก็บข้อมูลเสียง จะทำการจัดเก็บออกเป็น 3 ส่วน เนื่องจากต้องการโมเดลที่ได้จากกระบวนการฝึกฝน 3 โมเดล ประกอบด้วย

- 1) ข้อมูลเสียงในส่วนการกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า ประกอบด้วยข้อมูลเสียง “โทรคม” เป็นคำสั่งในการกระตุ้นการทำงานของระบบ ให้รับคำสั่งต่อไป “แอร์” เป็นคำสั่งในการระบุอุปกรณ์ไฟฟ้าที่ต้องการควบคุมคือเครื่องปรับอากาศ และ “หลอดไฟ” เป็นคำสั่งในการระบุอุปกรณ์ไฟฟ้าที่ต้องการควบคุมคือหลอดไฟ
- 2) ข้อมูลเสียงในส่วนคำสั่งในการควบคุมหลอดไฟ ประกอบด้วยข้อมูลเสียง “ปิดไฟ” “เปิดไฟ” เป็นคำสั่งในการควบคุมหลอดไฟให้ปิดและเปิดไฟตามลำดับ และ “สว่างมาก” “สว่างน้อย” เป็นคำสั่งในการควบคุมหลอดไฟให้เพิ่มความสว่างและลดความสว่าง
- 3) ข้อมูลเสียงในส่วนคำสั่งในการควบคุมเครื่องปรับอากาศ ประกอบด้วยข้อมูลเสียง “ปิดแอร์” “เปิดแอร์” เป็นคำสั่งในการควบคุมเครื่องปรับอากาศให้ปิดและเปิดแอร์ตามลำดับ คำสั่ง “เพิ่มอุณหภูมิ” “ลดอุณหภูมิ” เป็นคำสั่งในการควบคุมเครื่องปรับอากาศให้เพิ่มและลดอุณหภูมิ และคำสั่ง “เปลี่ยนโหมด” เป็นคำสั่งในการควบคุมเครื่องปรับอากาศให้เปลี่ยนจากโหมดเย็นเป็นโหมดพัดลม

#### 3.3.2 การจัดเก็บผลที่ได้จากการสกัดคุณลักษณะเด่น

การจัดเก็บผลการทดลองในส่วนของการกระบวนการสกัดคุณลักษณะเด่นสำหรับนำไปใช้ในกระบวนการฝึกฝน หลังจากนำข้อมูลเสียงไปผ่านกระบวนการ MFCC จะทำการจัดเก็บผลลัพธ์ที่ได้จากกระบวนการ MFCC ให้อยู่ในรูปแบบเมทริกซ์โดยใช้ไฟล์ข้อมูล .npy สามารถแสดงได้ดังรูปที่ 3.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	A	B	C	D	E	F	G	H	AI	AJ	AK	AL	AM	AN	AO	AP
1	0	0	1	2	3	4	5	6	33	34	35	36	37	38	39	40
2	0	-17.389	-42.8444	-7.51995	1.007054	-6.2259	-2.27011	-6.15227	-7.4789	-8.02781	-10.3403	-2.06478	19.97682	-7.02034	-4.22407	2
3	1	-17.9245	-37.3333	-9.41808	12.4729	-4.53754	-4.71238	-7.37367	8.545967	-5.43904	2.724194	2.325645	11.81086	-4.0524	0.51603	2
4	2	-19.5276	-20.9531	-6.41356	2.659595	-0.48768	2.522352	-8.75848	-7.84891	-2.36774	3.102836	5.094342	-2.13177	3.145742	-1.56689	2
5	3	-28.0072	-8.41252	13.8931	-2.80988	-15.3654	0.655672	0.34355	-5.12445	-6.26649	-0.81127	-7.11229	3.05329	3.82405	1.124747	2
6	4	-13.5885	9.176819	20.03936	0.157576	-11.2693	-12.426	-0.24866	-2.60753	-2.72686	-0.47546	-0.25728	11.34989	0.71462	-9.27977	2
7	5	-11.3587	10.29259	12.86437	-0.2642	-9.2022	-3.89657	-7.08191	0.52427	-4.57084	-3.95672	-4.17074	10.35104	3.565623	-10.8135	2
8	6	-3.37482	12.92372	29.53249	1.237594	-4.24298	-8.2624	5.389484	-0.07788	-4.66857	-8.13919	-5.14959	-0.14814	4.981077	-7.99628	2
9	7	-4.01208	10.28065	20.9842	11.53875	3.672216	-22.6254	7.042429	7.263489	-2.31108	-4.78116	-6.42132	5.621347	3.788548	-12.277	2
10	8	-5.57959	6.087753	19.1242	4.587935	-1.97997	-23.7739	5.109756	1.219085	-2.18598	-10.2249	-8.37411	10.65205	-0.2447	-4.01912	2
11	9	1.287598	5.00383	13.23782	-9.56069	-17.3025	-38.8859	-1.46736	5.015321	-2.55899	-2.39492	-1.77798	10.99354	2.371982	-1.06973	2
12	10	1.421631	10.38824	24.06983	-0.66581	-12.0232	-26.1253	25.31617	4.356692	0.004399	-6.42303	-10.2233	3.516675	0.111558	-3.78947	2
13	11	17.16235	13.40572	12.05695	0.988628	-10.8252	-19.5264	31.19288	0.842343	-1.97455	-5.43042	-10.651	5.279908	-0.53398	0.412796	2
14	12	22.87979	14.12531	10.93223	4.685919	-4.12149	-17.4999	37.79637	4.417158	0.711265	-8.13597	-11.7698	11.86519	2.995922	-1.85648	2
15	13	25.75076	8.95961	8.01091	-3.6308	-5.91474	-19.1355	31.66155	9.344056	0.235247	-3.64314	-2.36517	8.183237	3.623058	-1.52772	2
16	14	24.26218	11.8604	0.170818	5.414383	-5.77991	-30.9848	25.52861	8.20943	-2.09937	2.618373	-8.07176	-0.46302	5.636628	-9.13678	2
17	15	31.49765	11.97926	1.098553	14.68412	7.746252	-33.7571	26.68892	1.220435	0.339013	3.720087	-7.06786	-6.37324	3.725579	-3.07224	2
18	16	26.04898	11.29582	-13.8182	14.64376	2.365871	-28.5576	28.28952	10.23583	-3.30397	-7.84238	-6.44941	4.828724	5.082641	-10.4277	2
19	17	27.45529	3.425369	-23.3936	11.73403	2.140322	-30.3705	20.61734	3.735495	2.148324	-0.46216	1.326287	0.328498	4.452241	-0.95221	2
20	18	34.70212	8.869308	-25.2632	15.14809	5.190266	-33.786	24.75593	6.863822	-2.33447	-0.22956	-0.68969	2.783651	2.381623	2.125602	2
21	19	29.02327	16.7596	-25.581	5.468597	16.57081	-16.8947	28.28134	-1.04592	-3.03488	-1.06148	-9.93672	-5.37984	0.981557	-4.23045	2

รูปที่ 3.27 Data Frame ของไฟล์ข้อมูล mfcc

จากรูปที่ 3.27 แสดงค่าสัมประสิทธิ์เซปตรัมบนความถี่เมลที่ได้จากกระบวนการ MFCC ซึ่งค่าที่ได้จะประกอบไปด้วยหลักที่ 0 ถึงหลักที่ 39 แสดงค่าสัมประสิทธิ์จำนวน 40 ค่าของแต่ละข้อมูลเสียง และหลักที่ 40 แสดงค่าป้ายกำกับของแต่ละข้อมูลเสียง แถวที่ 0 ถึงแถวที่ 4 จำนวน framewidth ที่เป็นอินพุตเลเยอร์ของกระบวนการฝึกฝน เมื่อนำข้อมูลไปฝึกฝนการรู้จำลักษณะเด่นเมื่อมีสัญญาณที่เข้ามาภายหลัง ระบบรู้จำจะสามารถบอกได้ว่าเสียงที่เข้ามานั้นมีลักษณะเหมือนหรือใกล้เคียงกับเสียงใดมากที่สุด

### 3.3.3 การจัดเก็บผลการทดสอบประสิทธิภาพของแบบจำลอง

#### 3.3.3.1 การจัดเก็บผลการเปรียบเทียบไฮเปอร์ค่าพารามิเตอร์ที่ใช้ในการ

ฝึกฝนข้อมูล

- แบบจำลอง Convolution Neural Network (CNN)

รูปแบบค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN จะประกอบไปด้วยจำนวน CNN layer, ขนาดของ kernel, ขนาด pooling, Feature map filter และ Dense hidden units ซึ่งจะจัดเก็บผลค่าความผิดพลาด (Loss) และความแม่นยำ (Accuracy) ของโมเดลอยู่ในรูปแบบตาราง เพื่อนำมาเปรียบเทียบหารูปแบบค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด ที่ทำให้โมเดลมีความผิดพลาดที่ต่ำ และค่าความแม่นยำในการทำนายที่มากที่สุด

- แบบจำลอง Long-Shot term memory (LSTM)

รูปแบบค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง LSTM จะประกอบไปด้วยจำนวน LSTM layer, จำนวน loopback และจำนวน hidden node ซึ่งจะจัดเก็บผลค่าความผิดพลาด (Loss) และความแม่นยำ (Accuracy) ของโมเดลอยู่ในรูปแบบตาราง เพื่อนำมาเปรียบเทียบหารูปแบบค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด ที่ทำให้โมเดลมีค่าความผิดพลาดที่ต่ำ และค่าความแม่นยำในการทำนายที่มากที่สุด

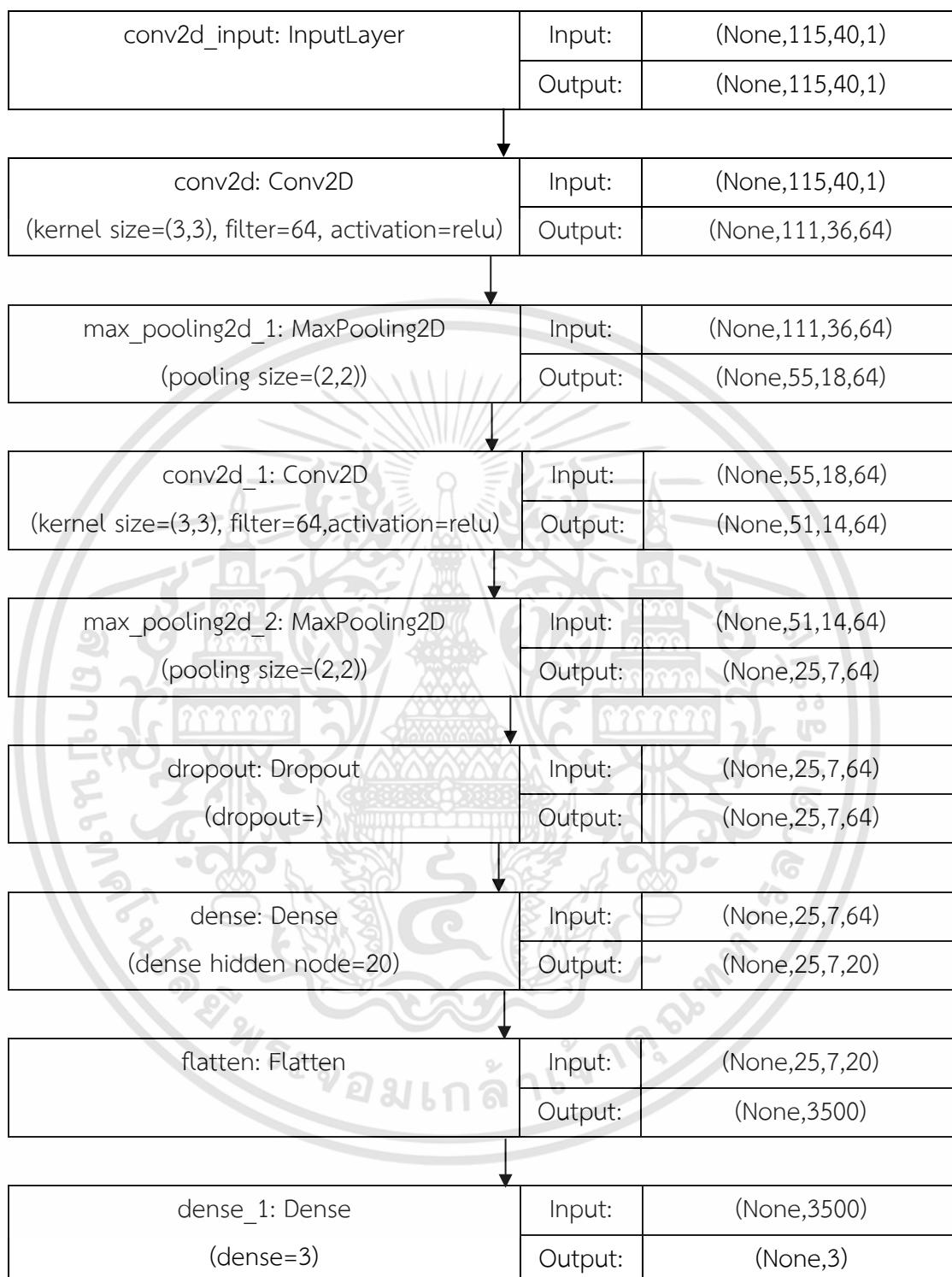
- แบบจำลอง CNN+LSTM

รูปแบบค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM จะทำการนำแบบจำลอง CNN โดยเลือกรูปแบบค่าไฮเปอร์พารามิเตอร์ที่ดีที่สุดของแบบจำลอง CNN ทั้ง 1 ชั้นและ 2 ชั้น นำมาต่อด้วยแบบจำลอง LSTM โดยเลือกรูปแบบไฮเปอร์พารามิเตอร์ที่ดีที่สุดของแบบจำลอง LSTM 1 ชั้นและ 2 ชั้นเช่นกัน ซึ่งจะจัดเก็บผลค่าความผิดพลาด (Loss) และความแม่นยำ (Accuracy) ของโมเดลอยู่ในรูปแบบตาราง เพื่อนำมาเปรียบเทียบหารูปแบบค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด ที่ทำให้โมเดลมีค่าความผิดพลาดที่ต่ำ และค่าความแม่นยำในการทำนายที่มากที่สุด

3.3.3.2 การจัดเก็บผลการทดสอบประสิทธิภาพของแบบจำลอง

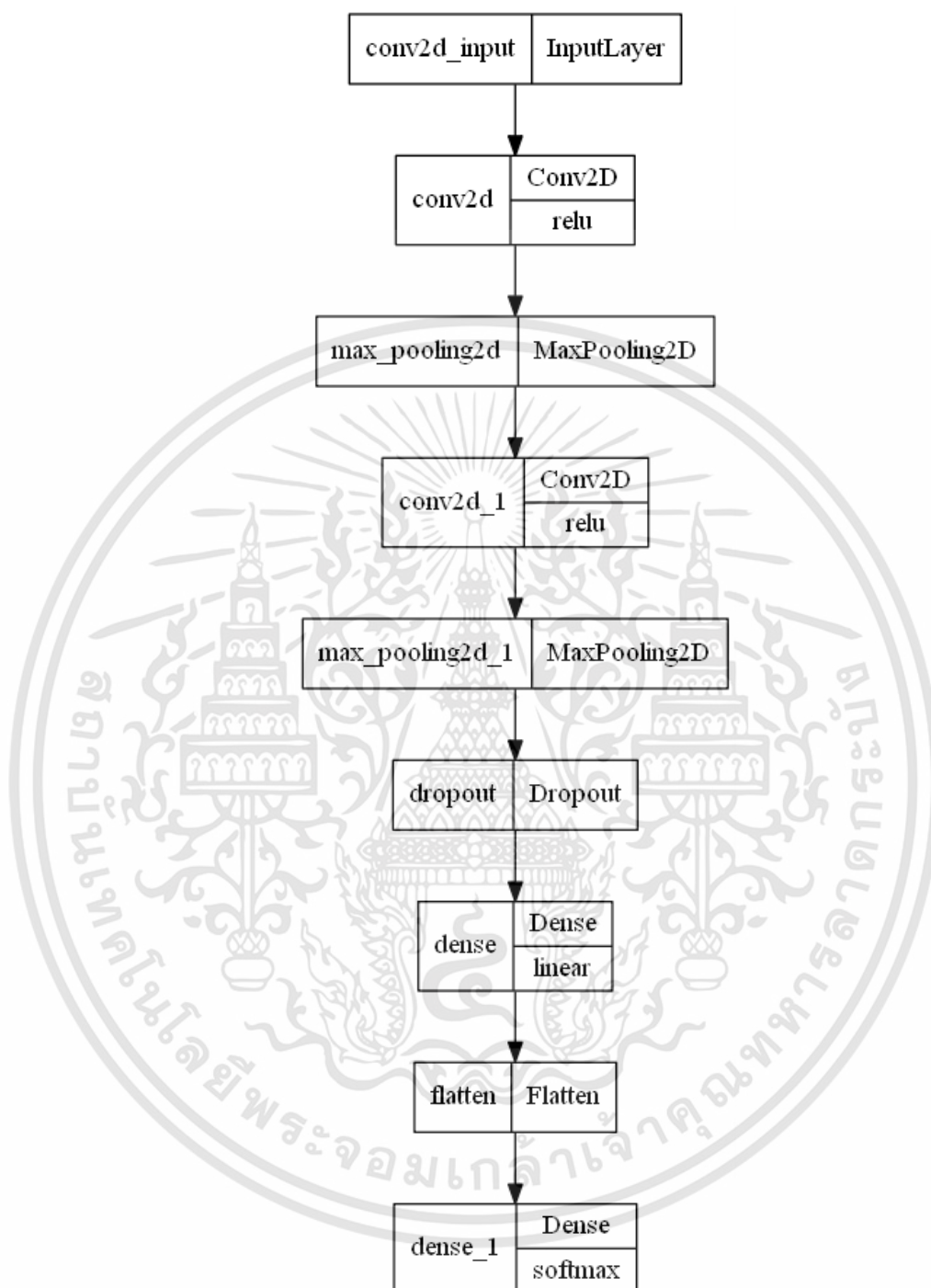
ในการฝึกฝน Neural Network ของโครงงานจะทำการฝึกฝนแบบจำลองจำนวน 3 แบบจำลอง เพื่อทำมาเปรียบเทียบหาแบบจำลองที่มีประสิทธิภาพในการทำนายค่าที่ดีที่สุด ประกอบด้วย แบบจำลอง Convolution Neural Network (CNN) แสดงดังรูปที่ 3.28 แบบจำลอง Long Short-Term Memory (LSTM) แสดงดังรูปที่ 3.30 และแบบจำลอง CNN+LSTM แสดงดังรูปที่ 3.33 และ 3.34

จากรูปที่ 3.28 บล็อกไดอะแกรมโครงสร้างของแบบจำลอง CNN เป็นกระบวนการ Convolution 2 มิติ กำหนดให้สัญญาณเสียงอินพุตมีจำนวน 3 ค่า จากรูปค่าอินพุต คือ (115, 40, 1) ประกอบด้วยค่า 115 เป็นค่าพารามิเตอร์ frame width\*time step ค่าถัดมาเป็นค่า Number of feature ซึ่งเป็นค่าจำนวนสัมประสิทธิ์เซปตริ่มความถี่เมลจำนวน 40 ค่า และค่าสุดท้ายเป็นค่า color of scale แสดงถึงจำนวนช่องสีของอินพุตเลเยอร์ กำหนดให้มีค่าเท่ากับ 1 คือมีช่องสีเดียวในเลเยอร์อินพุต ซึ่งเป็นสเกลสีเทา แสดงว่าเลเยอร์อินพุตเป็นภาพระดับสีเทา เมื่อนำอินพุตเข้าสู่เลเยอร์ Convolution 2 มิติ จะได้ค่าที่มีรูปร่างเปลี่ยนไปตามขนาดของ filter ที่กำหนด และนำเข้าสู่เลเยอร์ max pooling เพื่อสกัดเอาค่าเฉพาะสูงสุดเก็บไว้ในเอาต์พุต จะทำให้ได้ค่ามิติที่มีขนาดเล็กลง ก่อนนำเข้าสู่เลเยอร์ Dense เพื่อทำการแยกเอาต์พุตตามประเภทข้อมูล



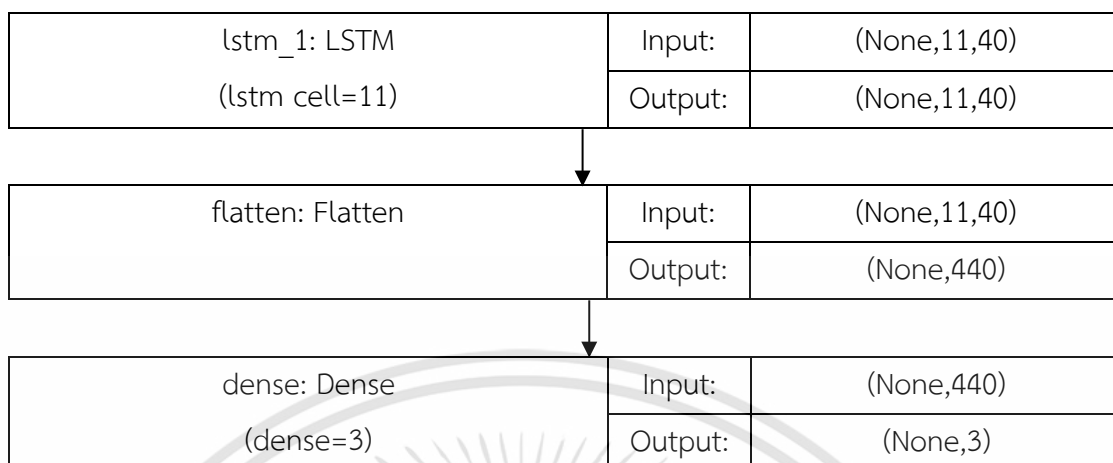
รูปที่ 3.28 บล็อกไดอะแกรมแบบจำลอง Convolution neural network (CNN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

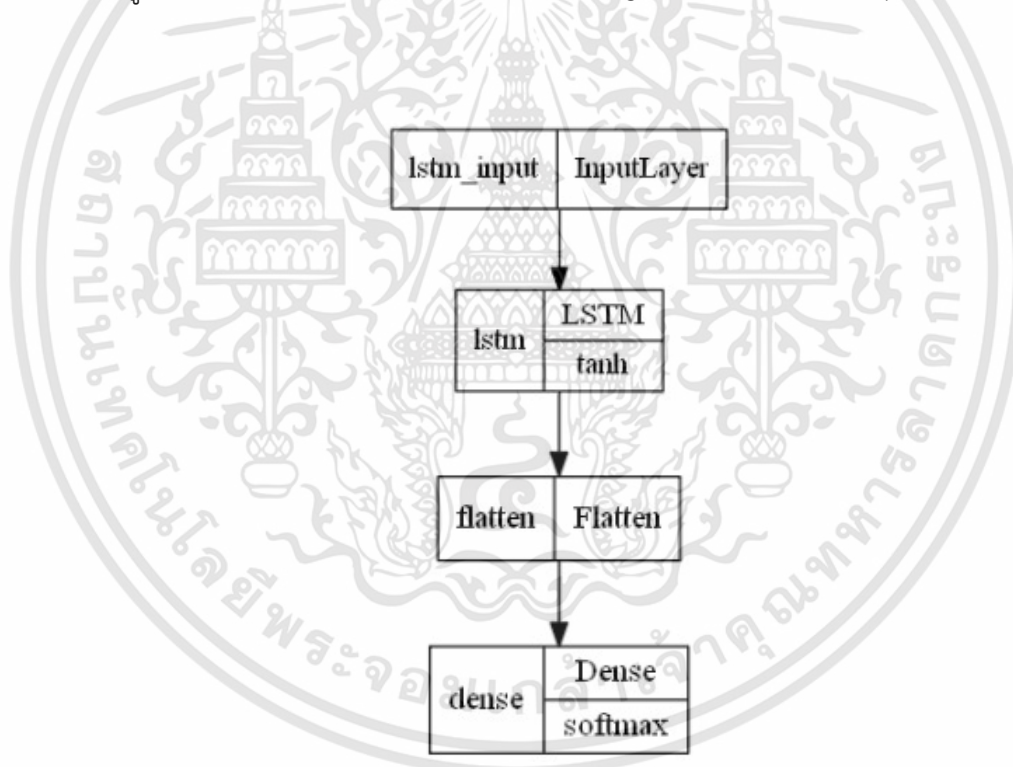


รูปที่ 3.29 บล็อกไดอะแกรมแสดง Activating function ของแบบจำลอง CNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 บล็อกไดอะแกรมแบบจำลอง Long-short term memory (LSTM)

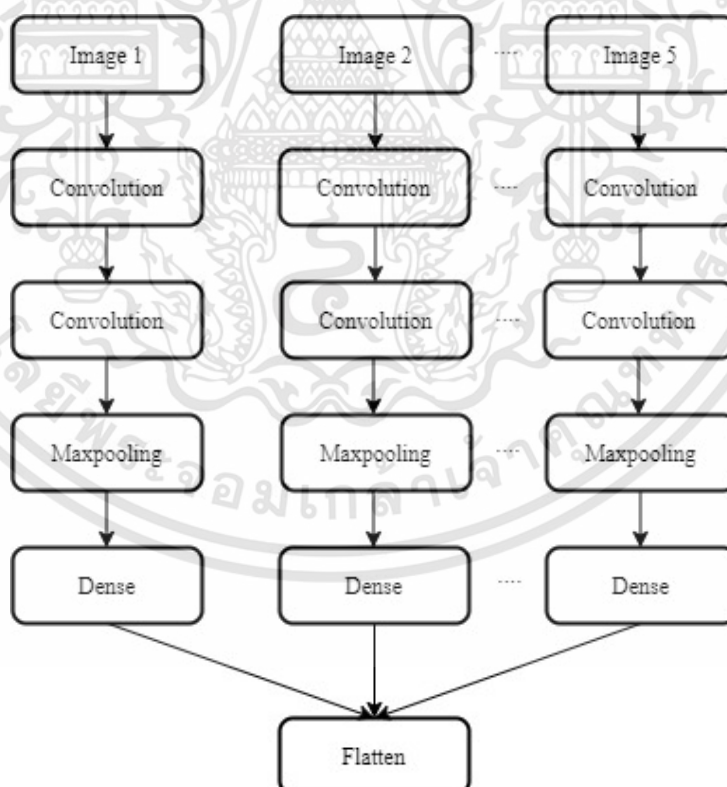


รูปที่ 3.31 บล็อกไดอะแกรม Activating function ของแบบจำลอง LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.30 กำหนดให้สัญญาณอินพุตมีจำนวน 2 ค่าประกอบด้วยค่าพารามิเตอร์ frame width และค่า Number of feature ซึ่งเป็นค่าจำนวนสัมประสิทธิ์เซปตรัมบนความถี่เมลจำนวน 40 ค่า การตัดสินใจจะพิจารณาจากอินพุตปัจจุบันและเอาต์พุตที่ได้จากการเรียนรู้ก่อนหน้า หลังจากนั้นนำมาผ่าน Dense layer เพื่อแยกตามประเภทข้อมูล

ในแบบจำลอง CNN+LSTM โครงสร้างการทำงานจะแบ่งออกเป็น 2 ส่วนประกอบด้วย ส่วนแรกคือแบบจำลอง Convolution Neural Network (CNN) ที่มีความสามารถในการจำแนกข้อมูลประเภทรูปภาพ มีข้อเสียคือสามารถรับอินพุตของภาพได้แค่ภาพเดียว เพื่อนำไปค้นหาคุณสมบัติของภาพ แต่อินพุตของโครงงานนี้เป็นข้อมูลเสียงที่เป็นค่าตามลำดับเวลา จึงทำการแบ่งลำดับข้อมูลเสียงออกเป็น 5 ภาพเรียกว่าค่า timestep ซึ่งมีขนาด 23x40 พิกเซลในโทนสีเทา โดยค่า 23 เป็นค่าความกว้างของเฟรมข้อมูล ค่า 40 เป็นค่าจำนวนสัมประสิทธิ์เซปตรัมบนความถี่เมล ซึ่งใน Convolution 2 มิติไม่สามารถใช้รูปร่าง (5,23,40,1) ได้ ดังนั้นจำเป็นต้องส่งภาพที่ 1 ไปยัง Convolution ที่ 1 และภาพที่ 2 ไปยัง Convolution ที่ 2 จนถึงภาพที่ 5 ไปยัง Convolution ที่ 5 สามารถอธิบายได้ดังรูปที่ 3.32

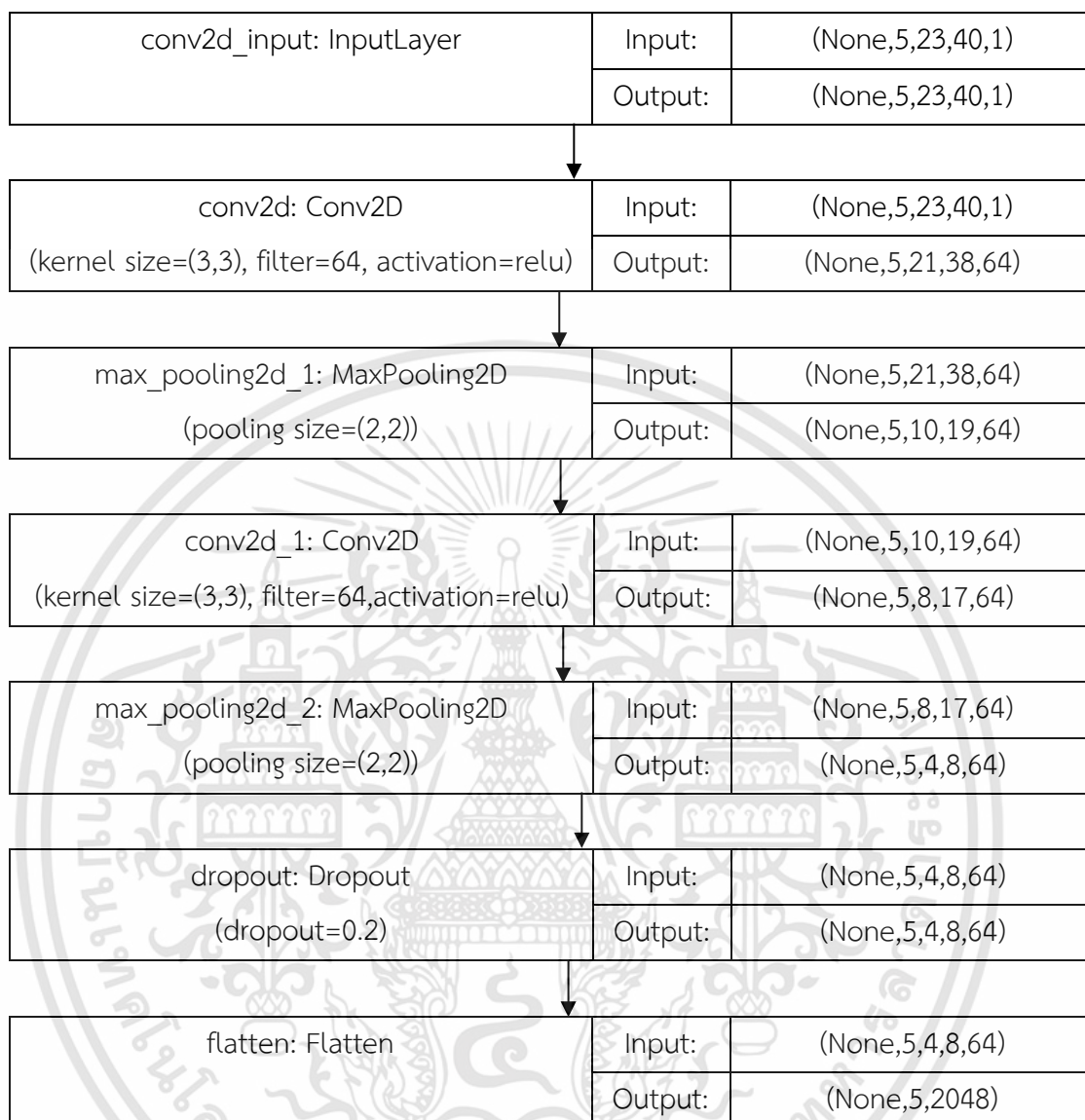


รูปที่ 3.32 บล็อกไดอะแกรมโครงสร้าง CNN ประเภทอินพุตมีหลายภาพ

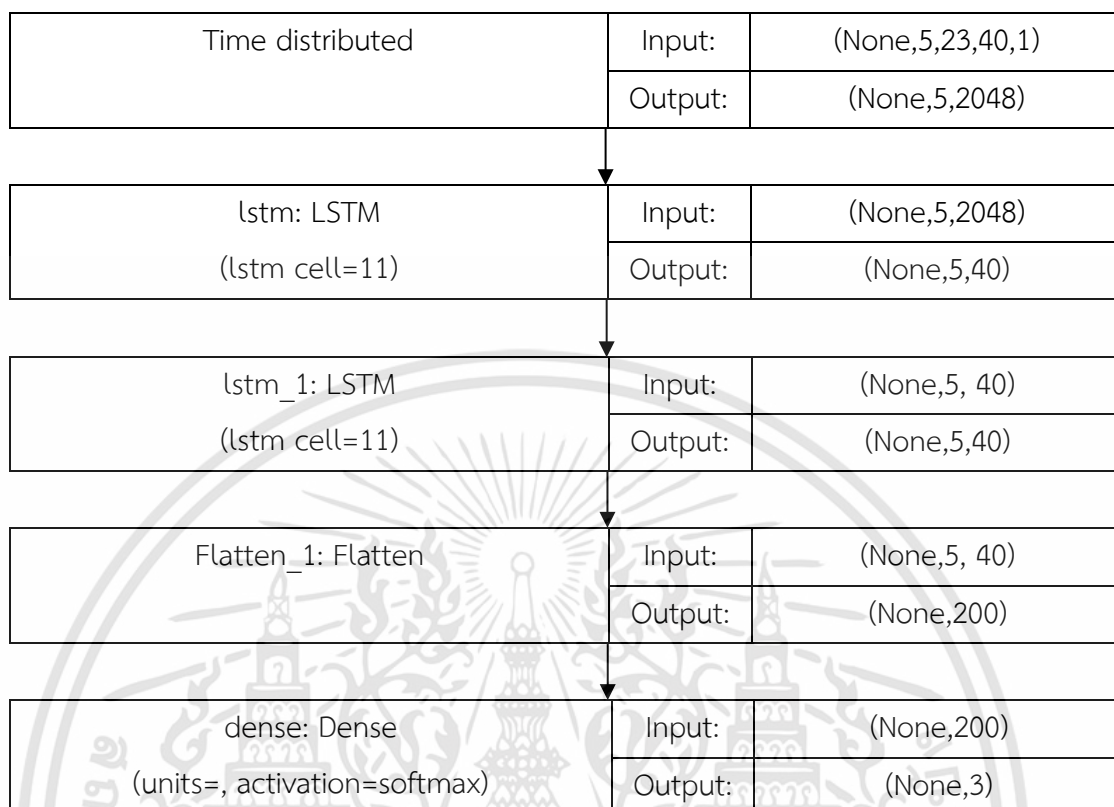
โดยโครงสร้างที่สามารถรับหลายภาพเป็นอินพุต ทำโครงสร้างเดียวกันในทุกภาพ และสามารถนำเลเยอร์มาต่อได้อีก คือ โครงสร้าง Time distributed layer

Time distributed layer จะช่วยให้สามารถใช้เลเยอร์ Convolution 2 มิติได้ในแต่ละภาพ หรือเรียกว่าทุก time step โดยปกติเลเยอร์ Convolution จะรับได้หนึ่งภาพที่มีรูปร่าง 3 มิติ ในที่นี่ต้องกำหนดจำนวนภาพที่ต้องการส่งเข้าไปในเครือข่ายแบบครั้งเดียว ดังนั้นเลเยอร์ Time Distributed คืออินพุตของโมเดล จำเป็นต้องตั้งค่าอีกมิติหนึ่งคือจำนวนภาพที่ต้องการส่ง ดังรูปที่ 3.33 ต้องการส่ง 5 ภาพที่มีรูปร่าง (23,40,1) จึงจำเป็นต้องเพิ่มอีกหนึ่งมิติเพื่อนำเสนอ  $5 \times (23,40,1)$  ดังนั้นรูปร่างอินพุตของของโครงสร้างนี้จะมีรูปร่าง (5,23,40,1) จะเรียกว่าอินพุตชั่วคราว ไปผ่านเลเยอร์ convolution 2 มิติเป็นอันดับแรก และนำมาผ่านเลเยอร์ LSTM ดังรูปที่ 3.34 หลังจากนั้นนำมาผ่าน Dense layer เพื่อแยกเอาที่พุดตามประเภทข้อมูล



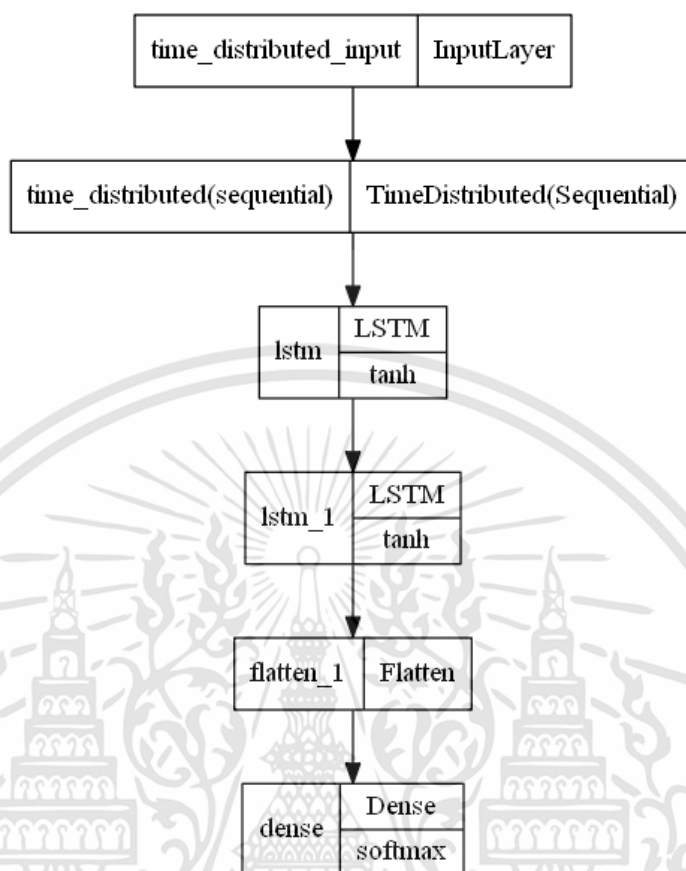


รูปที่ 3.33 บล็อกไดอะแกรมของแบบจำลอง CNN ที่เป็นอินพุตชั่วคราว



รูปที่ 3.34 บล็อกไดอะแกรมของแบบจำลอง CNN+LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.35 บล็อกไดอะแกรมแสดง Activating function ของแบบจำลอง CNN+LSTM

การจัดเก็บผลการทดสอบประสิทธิภาพของแบบจำลองจะทำการพิจารณา 2 ส่วน ประกอบด้วยค่าความผิดพลาด (Loss) และค่าความแม่นยำ (Accuracy) ของชุดข้อมูลการฝึกฝนและชุดข้อมูล Validation โดยจะทำการเลือกโมเดลที่มีค่าความผิดพลาดต่ำที่สุดและค่าความแม่นยำมากที่สุดในการนำโมเดลไปประยุกต์ใช้งาน

### 3.3.4 การจัดเก็บผลการทดสอบประสิทธิภาพของโมเดล

ในการจัดเก็บผลการทดสอบประสิทธิภาพในการทำนายค่าของโมเดลจะแบ่งออกเป็น 3 โมเดล ประกอบด้วย ส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า ส่วนโมเดลควบคุมหลอดไฟ และส่วนโมเดลควบคุมเครื่องปรับอากาศ จะทำการแสดง confusion matrix ในแต่ละโมเดลออกเป็น 3 ส่วน คือผลการทดสอบการทำนายค่าจากตาราง Confusion Matrix ของข้อมูลชุดทดสอบ, ผลการทดสอบการทำนายค่าจากตาราง Confusion Matrix ของเสียงจากผู้จัดทำ และผลการทดสอบการทำนายค่าจากตาราง Confusion Matrix ของเสียงจากผู้ทดสอบ ซึ่งจะบอกความแม่นยำในการทำนายค่าที่ถูกต้องของโมเดลว่ามีประสิทธิภาพดีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5 การจัดเก็บผลการทดสอบการใช้งานโมเดลในระบบสั่งการด้วยเสียง

ในการทดสอบการใช้งานโมเดลบน Raspberry Pi จะทำการทดสอบการใช้งานโดยให้ผู้จัดทำ 3 คนและผู้ทดสอบ 7 คน ทำการป้อนคำสั่งเสียงผ่านไมโครโฟนเสมือนใช้งานจริง โดยทำการป้อนคำสั่งเสียงละ 10 ครั้ง จำนวน 12 คำสั่ง แล้วนับจำนวนครั้งที่โมเดลสามารถทำนายค่าได้อย่างถูกต้องว่ามีค่าเท่าไร และมีประสิทธิภาพในการใช้งานจริงหรือไม่



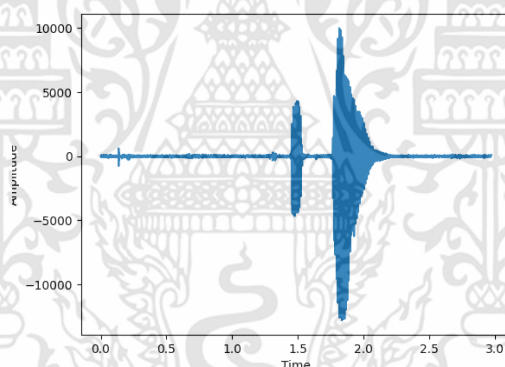
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

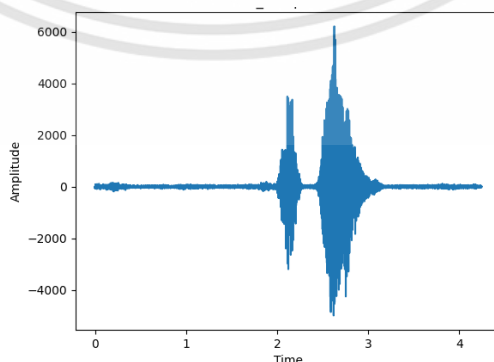
### ผลการทดลอง

#### 4.1 ผลการเปลี่ยนระดับข้อมูลเสียง (Pitch)

ในส่วนของการจัดเก็บข้อมูลเสียงของโครงการ ผู้จัดทำได้ทำการจัดเก็บข้อมูลเสียงออกเป็น 2 ส่วนประกอบ ได้แก่ข้อมูลเสียงที่ทำการบันทึกจากโปรแกรม เป็นข้อมูลเสียงที่มีระยะเวลา 3 วินาที แสดงตัวอย่างเสียงคำว่าปิดไฟได้ดังรูปที่ 4.1 และส่วนข้อมูลเสียงที่ถูกทำการเปลี่ยนระดับเสียง โดยทำการเพิ่มระดับเสียงที่ 1 เซมิโทน (Semitone) และลดระดับเสียงที่ -1 เซมิโทน (Semitone) และเปลี่ยนความเร็วเสียงที่ได้จากไลบรารี librosa ประกอบไปด้วยเพิ่มความเร็วเสียงจาก 1.0 เป็น 1.05, 1.15 และ 1.25 ลดความเร็วเสียงจาก 1.0 เป็น 0.75, 0.85 และ 0.95 แสดงตัวอย่างได้ดังรูปที่ 4.2-4.5

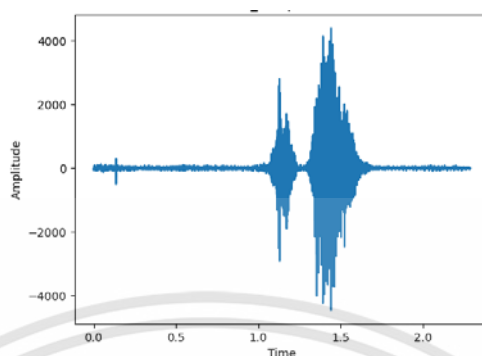


รูปที่ 4.1 ข้อมูลเสียงที่ได้จากการบันทึกจากโปรแกรม



รูปที่ 4.2 ตัวอย่างข้อมูลเสียงที่ถูกลดความเร็ว

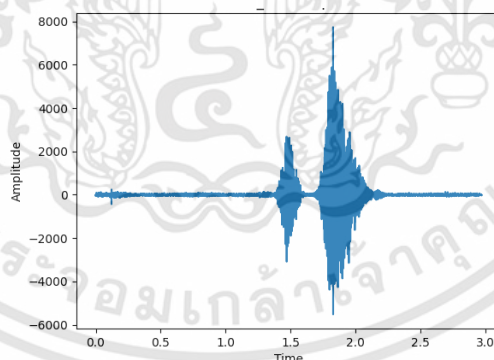
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ข้อมูลเสียงที่ถูกเพิ่มความเร็ว

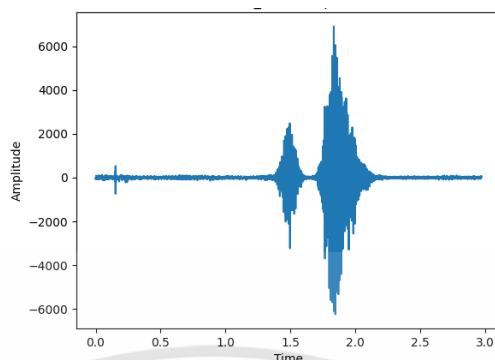
จากรูปที่ 4.2 เป็นตัวอย่างข้อมูลเสียงที่ถูกนำมาลดความเร็วลง ซึ่งพบว่าข้อมูลเสียงที่ถูกนำมาลดความเร็วเมื่อเปรียบเทียบกับข้อมูลเสียงเดิมจากรูปที่ 4.1 พบว่าข้อมูลเสียงที่ถูกลดความเร็วลงจะมีระยะเวลาเพิ่มมากขึ้น

จากรูปที่ 4.3 เป็นตัวอย่างข้อมูลเสียงที่ถูกนำมาเพิ่มความเร็ว ซึ่งพบว่าข้อมูลเสียงที่ถูกนำมาเพิ่มความเร็วเมื่อเปรียบเทียบกับข้อมูลเดิม พบว่าข้อมูลเสียงมีระยะเวลาที่สั้นลง



รูปที่ 4.4 ข้อมูลเสียงที่ถูกเพิ่มระดับเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ข้อมูลเสียงที่ถูกลดระดับเสียง

จากรูปที่ 4.4 เป็นตัวอย่างข้อมูลเสียงที่ถูกเพิ่มระดับเสียง จากรูปจะพบว่าค่าแอมพลิจูดของเสียงมีค่าเพิ่มขึ้น และจากรูปที่ 4.5 เป็นตัวอย่างข้อมูลเสียงที่ถูกลดระดับเสียง พบว่าค่าแอมพลิจูดของเสียงมีค่าลดลง โดยที่ระยะเวลาของเสียงยังมีค่าเท่าเดิม ซึ่งจุดประสงค์ในการเปลี่ยนระดับเสียงและความเร็วเสียง เพื่อให้ข้อมูลมีความหลากหลายมากยิ่งขึ้น เป็นการเพิ่มประสิทธิภาพในการฝึกฝนแบบจำลอง

## 4.2 ผลการทดสอบการเปรียบเทียบรูปแบบของค่าไฮเปอร์พารามิเตอร์ที่ใช้ในการฝึกฝนข้อมูล

### 4.2.1 ผลการเปรียบเทียบรูปแบบของค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง

#### Convolution Neural Network (CNN)

การเปรียบเทียบรูปแบบของค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN แสดงได้ดังตารางที่ 4.1 จะทำการเลือกรูปแบบของค่าไฮเปอร์พารามิเตอร์ที่ทำให้ค่าความผิดพลาด (Loss) ของโมเดลน้อยที่สุด และทำให้ความแม่นยำ (Accuracy) ในการทำนายค่าของโมเดลมีความแม่นยำมากที่สุด เพื่อนำค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมไปใช้ในขั้นตอนการทดสอบประสิทธิภาพของแบบจำลอง

ตารางที่ 4.1 ผลการฝึกฝนข้อมูลของแบบจำลอง Convolution Neural Network (CNN) ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์

จำนวน CNN layer	Kernel size	Pool size	Feature map filter	Dense hidden units	Accuracy training	Loss training	Accuracy validation	Loss validation
1	3x3	2x2	32	10	0.9988	0.0565	0.9652	0.0554
1	3x3	2x2	32	20	0.9994	0.0563	0.9887	0.0496
1	3x3	2x2	64	10	0.9982	0.0559	0.9990	0.0394
1	3x3	2x2	64	20	0.9988	0.0553	0.9996	0.0380
1	3x3	3x3	32	10	0.9976	0.0889	0.9674	0.0875
1	3x3	3x3	32	20	0.9982	0.0633	0.9752	0.0604
1	3x3	3x3	64	10	0.9988	0.0549	0.9878	0.0526
1	3x3	3x3	64	20	0.9990	0.0489	0.9988	0.0429
1	5x5	2x2	32	10	0.9994	0.0369	0.9952	0.041
1	5x5	2x2	32	20	0.9988	0.0552	0.9978	0.0597
1	5x5	2x2	64	10	0.9994	0.028	0.9998	0.0469
1	5x5	2x2	64	20	0.9994	0.0288	0.9994	0.0265
1	5x5	3x3	32	10	0.9968	0.0653	0.9994	0.0629
1	5x5	3x3	32	20	0.9994	0.0462	0.9994	0.0383
1	5x5	3x3	64	10	0.9994	0.034	0.9994	0.0321
1	5x5	3x3	64	20	0.9994	0.0413	0.9994	0.0371
2	3x3	2x2	32	10	0.9889	0.0561	0.9546	0.0554
2	3x3	2x2	32	20	0.9898	0.0369	0.9785	0.0492
2	3x3	2x2	64	10	0.9994	0.0292	0.9824	0.0268
2	3x3	2x2	64	20	0.9998	0.0245	0.9987	0.0283
2	3x3	3x3	32	10	0.9878	0.1091	0.9567	0.1018
2	3x3	3x3	32	20	0.9985	0.0996	0.9978	0.0949
2	3x3	3x3	64	10	0.9987	0.0743	0.9998	0.0714
2	3x3	3x3	64	20	0.9998	0.0953	0.9998	0.0911

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการฝึกฝนข้อมูลของแบบจำลอง Convolution Neural Network (CNN) ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์ (ต่อ)

จำนวน CNN layer	Kernel size	Pool size	Feature map filter	Dense hidden units	Accuracy training	Loss training	Accuracy validation	Loss validation
2	5x5	2x2	32	10	0.9994	0.0369	0.9543	0.0330
2	5x5	2x2	32	20	0.9994	0.0312	0.9870	0.0265
2	5x5	2x2	64	10	0.9996	0.0293	0.9976	0.0286
2	5x5	2x2	64	20	0.9998	0.0251	0.9980	0.0234
2	5x5	3x3	32	10	0.9988	0.1052	0.9987	0.1009
2	5x5	3x3	32	20	0.9994	0.0881	0.9987	0.0830
2	5x5	3x3	64	10	0.9987	0.0728	0.9987	0.0709
2	5x5	3x3	64	20	0.9998	0.1121	0.9987	0.1104

จากตารางที่ 4.1 พบว่ารูปแบบค่าของไฮเปอร์พารามิเตอร์ที่ทำให้โมเดลมีค่าความผิดพลาดในการทำนายที่น้อย และค่าความแม่นยำในการทำนายค่าที่มากที่สุด คือ จำนวน CNN layer เท่ากับ 2 ขนาดของ kernel เท่ากับ 3x3 ขนาด pooling เท่ากับ 2x2 Feature map filter เท่ากับ 64 และ Dense hidden units เท่ากับ 20 ซึ่งทำให้ค่าความผิดพลาดในการฝึกฝนมีค่า 0.0245 ค่าความแม่นยำในการฝึกฝนมีค่า 99.88% ค่าความผิดพลาดในการทดสอบ 0.0214 และค่าความแม่นยำในการทดสอบ 99.87% โดยจะนำรูปแบบของไฮเปอร์พารามิเตอร์นี้ไปใช้ในขั้นตอนการเปรียบเทียบประสิทธิภาพของแบบจำลอง CNN ในหัวข้อ 4.3

#### 4.2.2 ผลการเปรียบเทียบค่าพารามิเตอร์ของแบบจำลอง Long-Shot Term

##### Memory (LSTM)

การเปรียบเทียบรูปแบบของค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง LSTM แสดงได้ดังตารางที่ 4.2 จะทำการเลือกรูปแบบของค่าไฮเปอร์พารามิเตอร์ที่ทำให้ค่าความผิดพลาด (Loss) ของโมเดลน้อยที่สุด และทำให้ความแม่นยำ (Accuracy) ในการทำนายค่าของโมเดลมีความแม่นยำมากที่สุด เพื่อนำค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมไปใช้ในขั้นตอนการทดสอบประสิทธิภาพของแบบจำลอง

ตารางที่ 4.2 ผลการฝึกฝนข้อมูลของแบบจำลอง Long-Shot Term Memory (LSTM)  
ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์

จำนวน LSTM layer	จำนวน Loopback	จำนวน hidden node	Accuracy training	Loss training	Accuracy validation	Loss validation
1	1	10	0.8873	0.4944	0.8845	0.5040
1	1	20	0.9253	0.4110	0.9195	0.4157
1	1	30	0.9340	0.3794	0.9269	0.3907
1	1	40	0.9402	0.3650	0.9337	0.3728
1	11	10	0.9806	0.5210	0.9565	0.2899
1	11	20	0.9953	0.1798	0.9787	0.2031
1	11	30	0.9987	0.1450	0.9894	0.1693
1	11	40	0.9986	0.1282	0.9913	0.1451
1	23	10	0.9717	0.2981	0.9273	0.3689
1	23	20	0.9846	0.2494	0.9475	0.3156
1	23	30	0.9894	0.2031	0.9414	0.2864
1	23	40	0.9934	0.1779	0.9515	0.2747
2	1	10	0.8959	0.4047	0.8889	0.4235
2	1	20	0.9339	0.2804	0.9195	0.3106
2	1	30	0.9515	0.2263	0.9295	0.2719
2	1	40	0.9659	0.2147	0.9389	0.2635
2	11	10	0.9339	0.2804	0.9195	0.3106
2	11	20	0.9515	0.2263	0.9295	0.2719
2	11	30	0.9625	0.1988	0.9325	0.2623
2	11	40	0.9888	0.1328	0.9432	0.2476
2	23	10	0.9760	0.2060	0.9091	0.3619
2	23	20	0.9849	0.1679	0.9253	0.3607
2	23	30	0.9860	0.1510	0.9263	0.3408
2	23	40	0.9882	0.1411	0.9354	0.3054

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 พบว่ารูปแบบค่าของไฮเปอร์พารามิเตอร์ที่ทำให้โมเดลมีความผิดพลาดในการทำนายที่น้อย และค่าความแม่นยำในการทำนายค่าที่มากที่สุด คือ จำนวน LSTM layer เท่ากับ 1 จำนวน Loopback เท่ากับ 11 และจำนวน hidden node เท่ากับ 40 ซึ่งทำให้ค่าความผิดพลาดในการฝึกฝนมีค่า 0.1282 ค่าความแม่นยำในการฝึกฝนมีค่า 98.88% ค่าความผิดพลาดในการทดสอบ 0.1451 และค่าความแม่นยำในการทดสอบ 94.32% โดยจะนำรูปแบบของไฮเปอร์พารามิเตอร์นี้ไปใช้ในขั้นตอนการเปรียบเทียบประสิทธิภาพของแบบจำลอง LSTM ในหัวข้อ 4.3

#### 4.2.3 ผลการเปรียบเทียบค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM

แบบจำลอง CNN+LSTM จะทำการนำแบบจำลอง CNN โดยเลือกรูปแบบค่าไฮเปอร์พารามิเตอร์ที่ดีที่สุดของแบบจำลอง CNN ทั้ง 1 ชั้นและ 2 ชั้น นำมาต่อกับแบบจำลอง LSTM โดยเลือกรูปแบบไฮเปอร์พารามิเตอร์ที่ดีที่สุดของแบบจำลอง LSTM 1 ชั้นและ 2 ชั้นเช่นกัน โดยแบ่งการทดสอบโมเดลออกเป็นตารางดังต่อไปนี้

ตารางที่ 4.3 รูปแบบที่ 1 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM

จำนวน CNN layer	จำนวน LSTM layer	Kernel size	Pool size	Feature map filter	Dense hidden units	จำนวน Loopback	จำนวน hidden node
1	1	3x3	2x2	64	20	11	40

ตารางที่ 4.4 รูปแบบที่ 2 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM

จำนวน CNN layer	จำนวน LSTM layer	Kernel size	Pool size	Feature map filter	Dense hidden units	จำนวน Loopback	จำนวน hidden node
1	2	3x3	2x2	64	20	11	40

ตารางที่ 4.5 รูปแบบที่ 3 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM

จำนวน CNN layer	จำนวน LSTM layer	Kernel size	Pool size	Feature map filter	Dense hidden units	จำนวน Loopback	จำนวน hidden node
2	1	3x3	2x2	64	20	11	40

ตารางที่ 4.6 รูปแบบที่ 3 ของไฮเปอร์พารามิเตอร์ของแบบจำลอง CNN+LSTM

จำนวน CNN layer	จำนวน LSTM layer	Kernel size	Pool size	Feature map filter	Dense hidden units	จำนวน Loopback	จำนวน hidden node
2	2	3x3	2x2	64	20	11	40

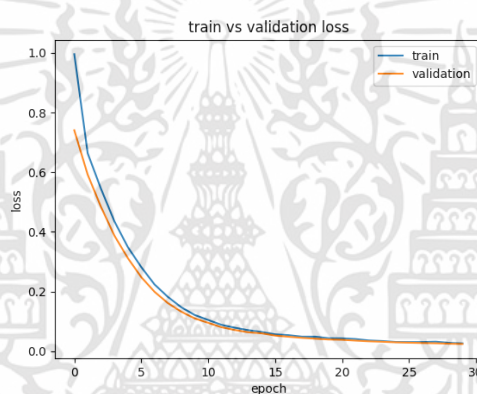
ตารางที่ 4.7 ผลการฝึกฝนข้อมูลของแบบจำลอง CNN+LSTM ในแต่ละรูปแบบของค่าไฮเปอร์พารามิเตอร์

รูปแบบ	Accuracy training	Loss training	Accuracy validation	Loss validation
1	0.9988	0.0779	1.0000	0.0779
2	1.0000	0.0575	1.0000	0.0573
3	1.0000	0.0642	0.9952	0.0670
4	1.0000	0.0528	1.0000	0.0553

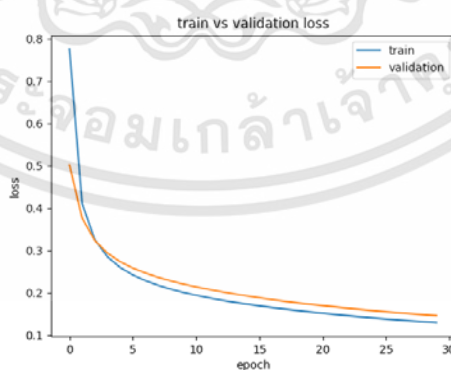
จากตารางที่ 4.7 พบว่ารูปแบบค่าของไฮเปอร์พารามิเตอร์ที่ทำให้โมเดลมีค่าความผิดพลาดในการทำนายที่น้อย และค่าความแม่นยำในการทำนายค่าที่มากที่สุด ได้แก่ รูปแบบที่ 4 คือ จำนวน CNN layer เท่ากับ 2, จำนวน LSTM layer เท่ากับ 2, ขนาด Kernel size เท่ากับ 3x3, ขนาด Pooling เท่ากับ 2x2, Feature map filter เท่ากับ 64, Dense hidden units เท่ากับ 20, จำนวน Loopback เท่ากับ 11 และจำนวน hidden node เท่ากับ 40 ซึ่งทำให้ค่าความผิดพลาดในการฝึกฝนมีค่า 0.0528 ค่าความแม่นยำในการฝึกฝนมีค่า 100% ค่าความผิดพลาดในการทดสอบ 0.0553 และค่าความแม่นยำในการทดสอบ 100% โดยจะนำรูปแบบของไฮเปอร์พารามิเตอร์นี้ไปใช้ในขั้นตอนการเปรียบเทียบประสิทธิภาพของแบบจำลอง CNN+LSTM ในหัวข้อ 4.3

### 4.3 ผลการทดสอบประสิทธิภาพแบบจำลอง

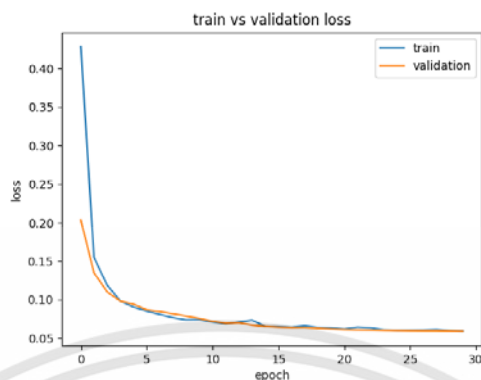
ในการทำการฝึกฝน Neural Network ได้ออกแบบแบบจำลองออกเป็น 3 แบบจำลอง เพื่อนำมาเปรียบเทียบประสิทธิภาพ ได้แก่ CNN, LSTM และ CNN+LSTM นำข้อมูลที่ได้จากการสกัดคุณลักษณะเด่นมาทำการแบ่งข้อมูลออกเป็น 3 ส่วนประกอบด้วยชุดข้อมูลฝึกฝน ชุดข้อมูล validation และชุดข้อมูลทดสอบ โดยมีอัตราส่วน 80 เปอร์เซ็นต์ 10 เปอร์เซ็นต์ และ 10 เปอร์เซ็นต์ ตามลำดับ จากนั้นจะทำการฝึกฝน Neural Network โดยกำหนดอัตราการเรียนรู้ (Learning rate) ของโครงสร้างโมเดลอยู่ที่ 0.001 กำหนดให้การปรับเมทริกซ์ใช้วิธีการ Adaptive momentum (Adam) และจำนวนในการฝึกฝนอยู่ที่ 30 รอบ



รูปที่ 4.6 แสดงกราฟเปรียบเทียบค่า Loss Function ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของโครงสร้าง CNN

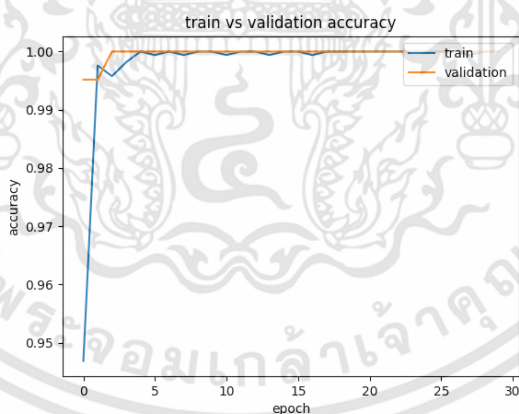


รูปที่ 4.7 แสดงกราฟเปรียบเทียบค่า Loss Function ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของโครงสร้าง LSTM

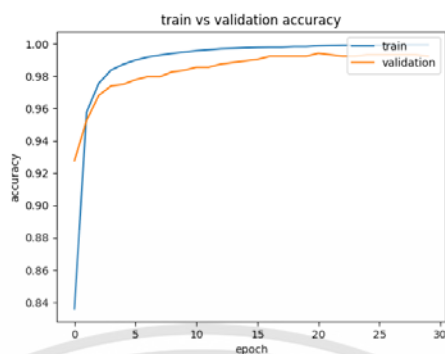


รูปที่ 4.8 แสดงกราฟเปรียบเทียบค่าความผิดพลาดระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง CNN+LSTM

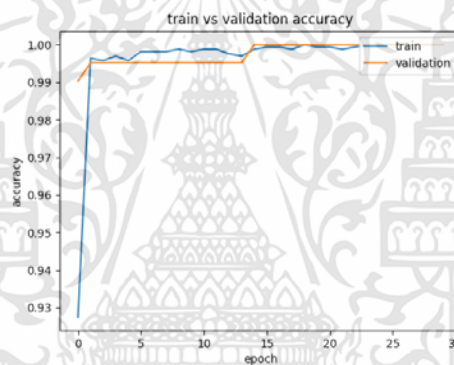
และสามารถแสดงกราฟ Accuracy ของชุดข้อมูลฝึกฝนและชุดข้อมูล validation data ของแบบจำลอง CNN ได้ดังรูปที่ 4.9 กราฟ Accuracy ของชุดข้อมูลฝึกฝนและชุดข้อมูล validation data ของแบบจำลอง LSTM ได้ดังรูปที่ 4.10 และกราฟ Accuracy ของชุดข้อมูลฝึกฝนและชุดข้อมูล validation data ของแบบจำลอง CNN+LSTM ได้ดังรูปที่ 4.11



รูปที่ 4.9 แสดงกราฟเปรียบเทียบค่า Accuracy ระหว่างข้อมูลฝึกฝน และข้อมูล validation data ของแบบจำลอง CNN



รูปที่ 4.10 แสดงกราฟเปรียบเทียบค่า Accuracy ระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง LSTM



รูปที่ 4.11 แสดงกราฟเปรียบเทียบค่าความแม่นยำระหว่างชุดข้อมูลฝึกฝน และชุดข้อมูล validation data ของแบบจำลอง CNN+LSTM

โดยในการเลือกแบบจำลองที่ดีที่สุด จะทำการคำนวณค่าความผิดพลาดและความแม่นยำของชุดข้อมูลฝึกฝน และชุดข้อมูล validation จะเลือกแบบจำลองที่มีค่าความผิดพลาดต่ำที่สุด และค่าความแม่นยำมากที่สุด โดยจะมีผลลัพธ์จากการฝึกฝนของ Neural Network ดังตารางที่ 4.8 ตามลำดับ

ตารางที่ 4.8 ผลการคำนวณค่าความผิดพลาดของแบบจำลอง

แบบจำลอง	Loss จากข้อมูลฝึกฝน	Loss จากข้อมูล validation data
CNN	0.0245	0.0214
LSTM	0.1282	0.1451
CNN+LSTM	0.0528	0.0553

ตารางที่ 4.9 ผลการคำนวณความแม่นยำของแบบจำลอง

แบบจำลอง	Accuracy จากข้อมูลฝึกฝน (%)	Accuracy จากข้อมูล validation data (%)
CNN	99.98	99.87
LSTM	98.88	94.32
CNN+LSTM	99.75	93.94

จากตารางที่ 4.8 และ 4.9 พบว่าการทดสอบประสิทธิภาพของแบบจำลอง Neural Network ที่มีค่าความผิดพลาดต่ำและมีความแม่นยำมากที่สุด คือแบบจำลอง CNN โดยมีค่าความผิดพลาดจากข้อมูลฝึกฝนเท่ากับ 0.0245 ค่าความผิดพลาดจากข้อมูล validation เท่ากับ 0.0214 และค่าความแม่นยำจากข้อมูลฝึกฝนเท่ากับ 99.98 เปอร์เซ็นต์ และมีค่าความแม่นยำจาก validation data เท่ากับ 99.87 เปอร์เซ็นต์ จะนำโมเดลที่ได้จากแบบจำลอง CNN ไปทดสอบประสิทธิภาพด้วยการนำไปประยุกต์ใช้ในเทคโนโลยีระบบสั่งการอุปกรณ์ไฟฟ้าด้วยเสียง

#### 4.4 ผลการทดสอบประสิทธิภาพของโมเดล

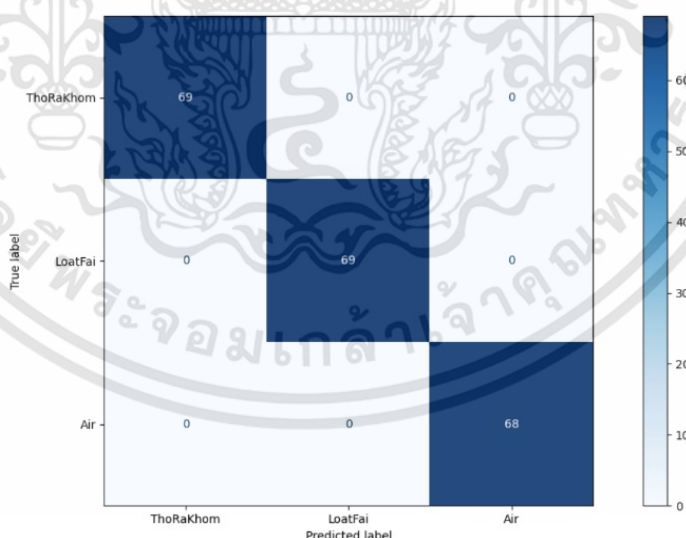
จากผลการทดสอบประสิทธิภาพแบบจำลองในหัวข้อที่ 4.3 พบว่าแบบจำลอง CNN มีประสิทธิภาพในการทำนายค่าได้ดีที่สุด เนื่องจากค่าความผิดพลาดจากการฝึกฝนข้อมูลเท่ากับ 0.0245 และค่าความแม่นยำเท่ากับ 99.98% ซึ่งก่อนนำโมเดลไปใช้งานบน Raspberry pi จะทำการทดสอบประสิทธิภาพของโมเดลโดยทดสอบการทำนายค่าจากตาราง Confusion Matrix แบ่งออกเป็น 2 ส่วน ประกอบด้วย Confusion Matrix ของข้อมูลชุดทดสอบ และ Confusion Matrix ของข้อมูลเสียงจากผู้ทดสอบ โดยจะทดสอบจำนวน 12 คำสั่งคำสั่งเสียงละ 10 ครั้ง ซึ่งจะแบ่งโมเดลออกเป็น 3 ส่วน ได้แก่

1. ส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า
2. ส่วนโมเดลควบคุมหลอดไฟ
3. ส่วนโมเดลควบคุมเครื่องปรับอากาศ

##### 4.4.1 ผลการทดสอบการทำนายค่าจากตาราง Confusion Matrix ของข้อมูลชุด

ทดสอบ

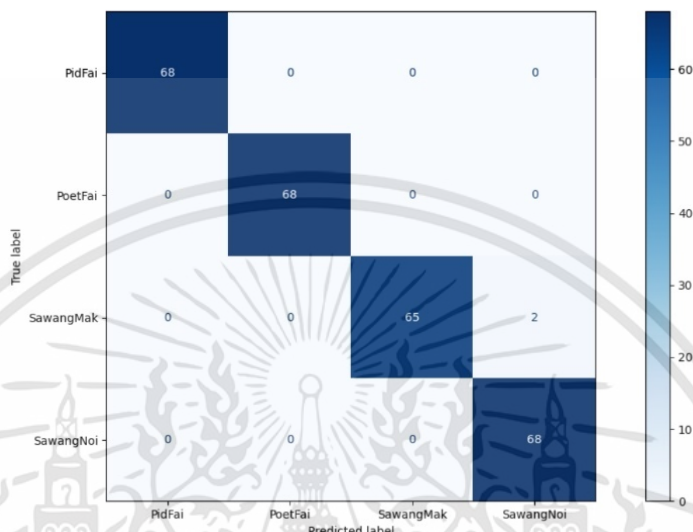
- ส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า สามารถแสดง Confusion matrix จากชุดข้อมูลทดสอบได้จากรูปที่ 4.12



รูปที่ 4.12 Confusion matrix จากชุดข้อมูลทดสอบ

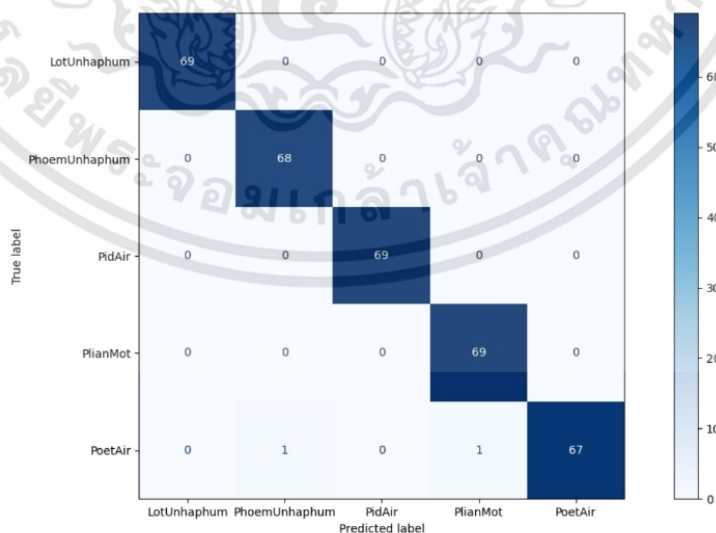
ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า

- ส่วนโมเดลควบคุมหลอดไฟ สามารถแสดง Confusion matrix จากชุดข้อมูลทดสอบได้จากรูปที่ 4.13



รูปที่ 4.13 Confusion matrix จากชุดข้อมูลทดสอบ  
ในส่วนโมเดลควบคุมหลอดไฟ

- ส่วนโมเดลควบคุมเครื่องปรับอากาศ สามารถแสดง Confusion matrix จากชุดข้อมูลทดสอบได้จากรูปที่ 4.14



รูปที่ 4.14 Confusion matrix จากชุดข้อมูลทดสอบ  
ในส่วนโมเดลควบคุมเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบ พบว่าค่าชุดข้อมูล คำสั่งโทรคม (Thorakhom) สามารถทำนายค่าที่ถูกต้อง 69 ครั้ง ชุดคำสั่งหลอดไฟ (LoatFai) สามารถทำนายค่าที่ถูกต้อง 69 ครั้ง ชุดคำสั่งแอร์ (Air) สามารถทำนายค่าที่ถูกต้อง 68 ครั้ง และไม่พบการทำนายค่าที่ผิดพลาดทั้ง 3 คำสั่ง ซึ่งแสดงให้เห็นว่าสามารถทำนายชุดคำสั่งในส่วนกระบวนการทำงานของระบบและระบุอุปกรณ์ไฟฟ้าได้อย่างมีประสิทธิภาพ

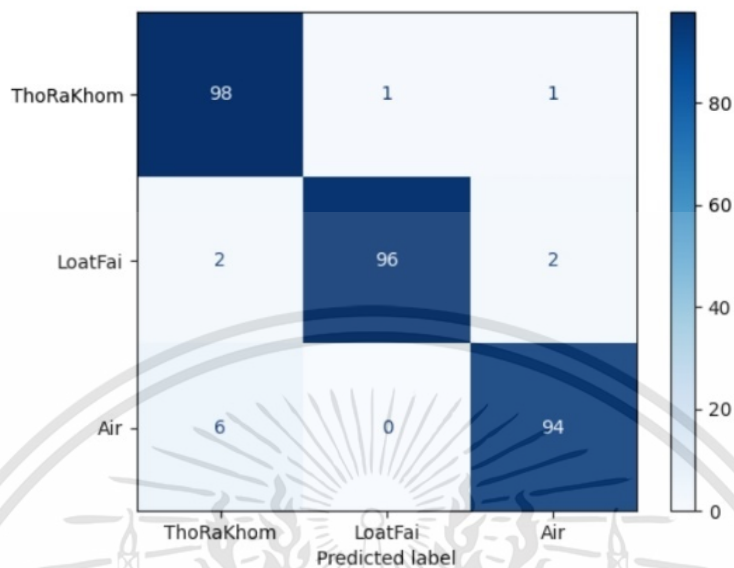
จากรูปที่ 4.13 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบ พบว่าค่าชุดข้อมูล คำสั่งปิดไฟ (Pidfai) สามารถทำนายค่าที่ถูกต้อง 68 ครั้ง ชุดคำสั่งเปิดไฟ (Poetfai) สามารถทำนายค่าที่ถูกต้อง 68 ครั้ง ชุดคำสั่งสว่างมาก (Sawangmak) สามารถทำนายค่าที่ถูกต้อง 65 ครั้ง ทำนายค่าผิดพลาดเป็นคำสั่งสว่างน้อย 2 ครั้ง และชุดคำสั่งสว่างน้อย (Sawangnoi) สามารถทำนายค่าที่ถูกต้อง 68 ครั้ง ซึ่งพบว่ายังมีการทำนายค่าที่ผิดพลาดในคำสั่งสว่างมาก แต่คำสั่งอื่นยังสามารถทำนายค่าออกมาได้อย่างมีประสิทธิภาพ

จากรูปที่ 4.14 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบ พบว่าค่าชุดข้อมูล คำสั่งลดอุณหภูมิ (LotUnhaphum) สามารถทำนายค่าที่ถูกต้อง 69 ครั้ง ชุดคำสั่งเพิ่มอุณหภูมิ (PhoemUnhaphum) สามารถทำนายค่าที่ถูกต้อง 68 ครั้ง ชุดคำสั่งปิดแอร์ (PidAir) สามารถทำนายค่าที่ถูกต้อง 69 ครั้ง ชุดคำสั่งเปลี่ยนโหมด (PlianMot) สามารถทำนายค่าที่ถูกต้อง 69 ครั้ง และชุดคำสั่งเปิดแอร์ (PoetAir) สามารถทำนายค่าที่ถูกต้อง 67 ครั้ง ทำนายค่าออกมาเป็นคำสั่งเพิ่มอุณหภูมิ 1 ครั้ง และคำสั่งเปลี่ยนโหมด 1 ครั้ง ซึ่งแสดงให้เห็นว่าการทำนายชุดคำสั่งในส่วนการควบคุมเครื่องปรับอากาศ ยังมีการทำนายค่าที่ผิดพลาดในคำสั่งเปิดแอร์ แต่คำสั่งอื่นสามารถทำนายค่าออกมาได้อย่างมีประสิทธิภาพ

#### 4.4.2 ผลการทดสอบการทำนายค่าจากตาราง Confusion Matrix ของเสียงจากผู้

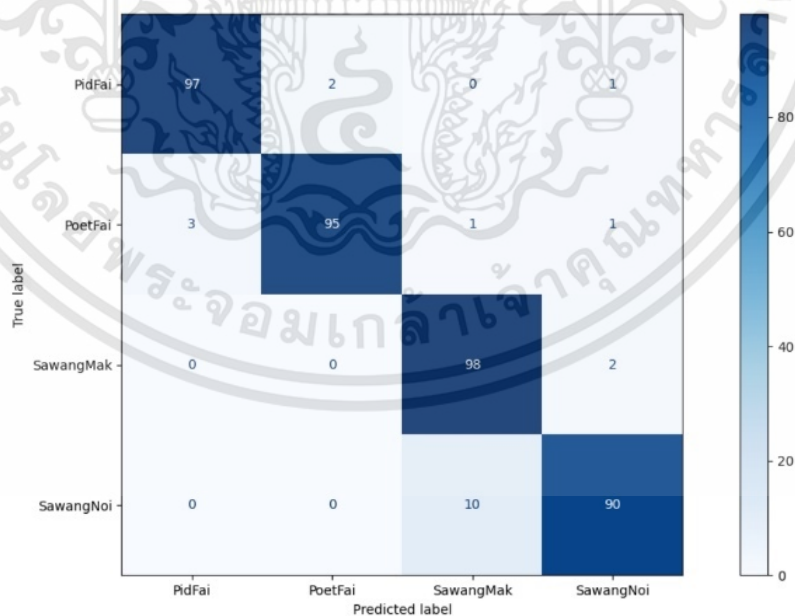
##### ทดสอบ

- ส่วนโมเดลกระบวนการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า สามารถแสดง Confusion matrix จากของเสียงจากผู้ทดสอบได้จากรูปที่ 4.15



รูปที่ 4.15 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ  
ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า

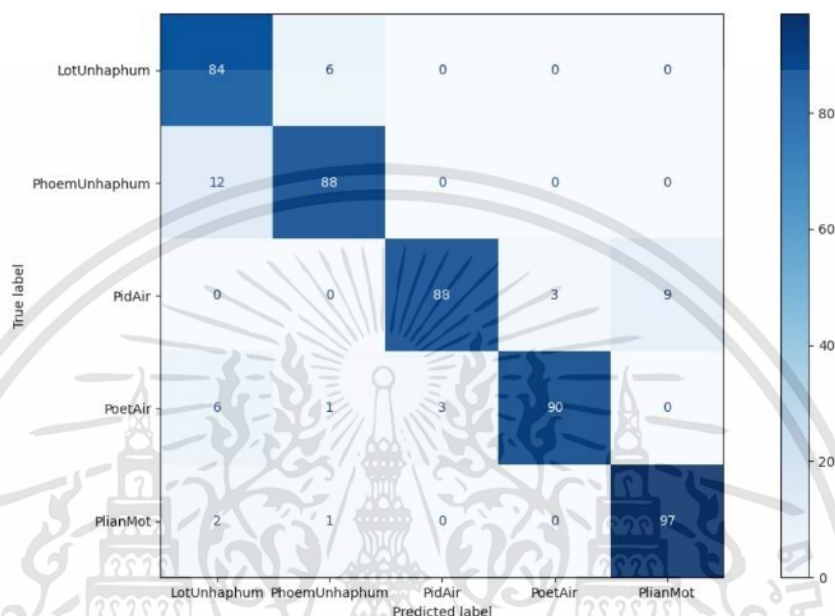
- ส่วนโมเดลควบคุมหลอดไฟ สามารถแสดง Confusion matrix จากของ  
เสียงจากผู้ทดสอบได้จากรูปที่ 4.16



รูปที่ 4.16 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ  
ในส่วนโมเดลควบคุมหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนโมเดลควบคุมเครื่องปรับอากาศ สามารถแสดง Confusion matrix จากของเสียงจากผู้ทดสอบได้จากรูปที่ 4.17



รูปที่ 4.17 Confusion matrix จากชุดข้อมูลเสียงจากผู้ทดสอบ  
ในส่วนโมเดลควบคุมเครื่องปรับอากาศ

จากรูปที่ 4.15 แสดง Confusion matrix ที่ได้จากชุดข้อมูลเสียงจากผู้ทดสอบ พบว่าค่าชุดข้อมูลคำสั่งโทรคม (Thorakhom) สามารถทำนายค่าที่ถูกต้อง 98 ครั้ง ทำนายผิดพลาดออกเป็นคำสั่งหลอดไฟ 1 ครั้งและแอร์ 1 ครั้ง ชุดคำสั่งหลอดไฟ (LoatFai) สามารถทำนายค่าที่ถูกต้อง 96 ครั้ง ทำนายผิดพลาดออกเป็นโทรคม 2 ครั้งและคำสั่งแอร์ 2 ครั้ง ชุดคำสั่งแอร์ (Air) สามารถทำนายค่าที่ถูกต้อง 94 ครั้ง ทำนายผิดพลาดออกเป็นคำสั่งโทรคม 6 ครั้ง ซึ่งแสดงให้เห็นว่าสามารถทำนายชุดคำสั่งในส่วนกระตุ้นการทำงานของระบบได้ทำนายออกมาผิดพลาดค่อนข้างน้อยเมื่อเทียบกับค่าทำนายที่ถูกต้อง

จากรูปที่ 4.16 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบ พบว่าค่าชุดข้อมูลคำสั่งปิดไฟ (Pidfai) สามารถทำนายค่าที่ถูกต้อง 97 ครั้ง ทำนายค่าผิดพลาดออกเป็นคำสั่งเปิดไฟ 2 ครั้งและคำสั่งสว่างน้อย 1 ครั้ง ชุดคำสั่งเปิดไฟ (Poetfai) สามารถทำนายค่าที่ถูกต้อง 95 ครั้ง ทำนายผิดพลาดออกเป็นชุดคำสั่งปิดไฟ 3 ครั้ง คำสั่งสว่างมาก 1 ครั้งและคำสั่งสว่างน้อย 1 ครั้ง ชุดคำสั่งสว่างมาก (Sawangmak) สามารถทำนายค่าที่ถูกต้อง 98 ครั้ง ทำนายค่าผิดพลาดเป็นคำสั่งสว่างน้อย 2 ครั้ง และชุดคำสั่งสว่างน้อย (Sawangnoi) สามารถทำนายค่าที่ถูกต้อง 90 ครั้ง ทำนายผิดพลาดเป็นคำสั่งสว่างมาก 10 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.17 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบ พบว่าค่าชุดข้อมูล คำสั่งลดอุณหภูมิ (LotUnhaphum) สามารถทำนายค่าที่ถูกต้อง 84 ครั้ง ทำนายผิดพลาดเป็นคำสั่งเพิ่มอุณหภูมิ 6 ครั้ง ชุดคำสั่งเพิ่มอุณหภูมิ (PhoemUnhaphum) สามารถทำนายค่าที่ถูกต้อง 88 ครั้ง ทำนายผิดพลาดเป็นคำสั่งเพิ่มอุณหภูมิ 12 ครั้ง ชุดคำสั่งปิดแอร์ (PidAir) สามารถทำนายค่าที่ถูกต้อง 88 ครั้ง ทำนายผิดพลาดเป็นคำสั่งเปิดแอร์ 3 ครั้ง คำสั่งเปลี่ยนโหมด 9 ครั้ง ชุดคำสั่งเปลี่ยนโหมด (PlianMot) สามารถทำนายค่าที่ถูกต้อง 90 ครั้ง ทำนายผิดพลาดเป็นคำสั่งลดอุณหภูมิ 6 ครั้ง คำสั่งเพิ่มอุณหภูมิ 1 ครั้งและคำสั่งปิดแอร์ 3 ครั้ง และชุดคำสั่งเปิดแอร์ (PoetAir) สามารถทำนายค่าที่ถูกต้อง 97 ครั้ง ทำนายค่าออกมาเป็นคำสั่งเพิ่มอุณหภูมิ 1 ครั้ง และคำสั่งลดอุณหภูมิ 1 ครั้ง

จากการทดสอบทั้ง 3 โมเดลพบว่าโมเดลมีประสิทธิภาพในการทำนายค่าได้ถูกต้องเป็นส่วนมาก และยังมีการทำนายที่ผิดพลาดบ้างซึ่งอาจเกิดจากปัญหาเสียงรบกวน หรือการขาดหายของเสียง ต่อไปเราจะทำการหาค่าตัดสินใจให้กับระบบเพื่อแก้ปัญหาในการทำนายค่าที่ผิดพลาด เพื่อให้ระบบการทำงานของระบบสั่งการด้วยเสียงทำงานอย่างมีประสิทธิภาพดียิ่งขึ้น

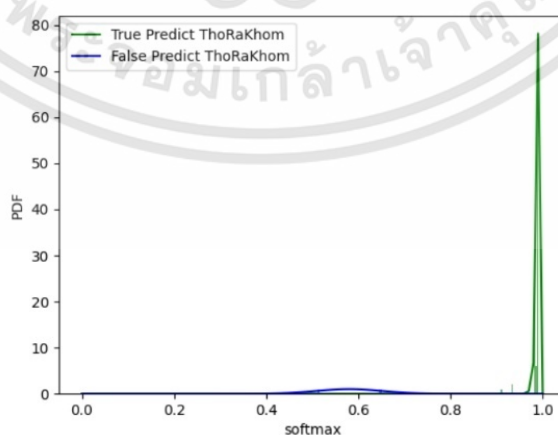
#### 4.4.3 ผลการทดสอบการคำนวณหาค่าตัดสินใจให้กับระบบ

ในการหาค่าตัดสินใจของแต่ละชุดคำสั่งข้อมูล จะนำข้อมูลที่ได้จากทดสอบในแต่ละชุดข้อมูลซึ่งประกอบด้วยข้อมูล True Prediction และ False Prediction จากผู้จัดทำ 3 คน และผู้ทดสอบ 7 คน จำนวนคนละ 10 เสียง มาหาค่าตัดสินใจที่มีความน่าจะเป็นการทำนายค่าที่ถูกต้อง แสดงกราฟ Normal distribution ของแต่ละชุดข้อมูลเสียงได้ดังรูปที่ 4.18-4.29

1) ส่วนโมเดลกระตุ้นการทำงานของระบบและระบบอุปกรณ์ไฟฟ้า

- คำสั่งโทรคม

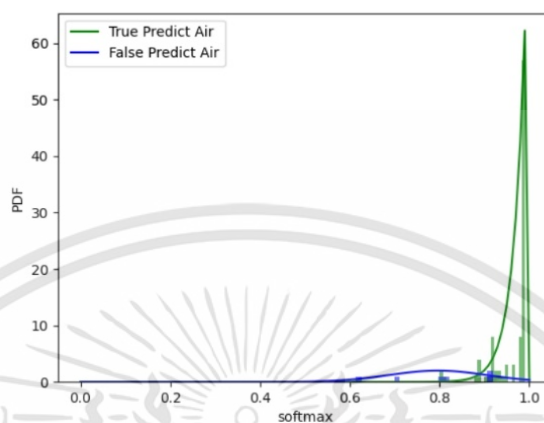
ค่าตัดสินใจของคำสั่งโทรคม เท่ากับ 0.92 ดังรูปที่ 4.18



รูปที่ 4.18 แสดงกราฟ Normal distribution ของชุดคำสั่งโทรคม

- คำสั่งแอร์

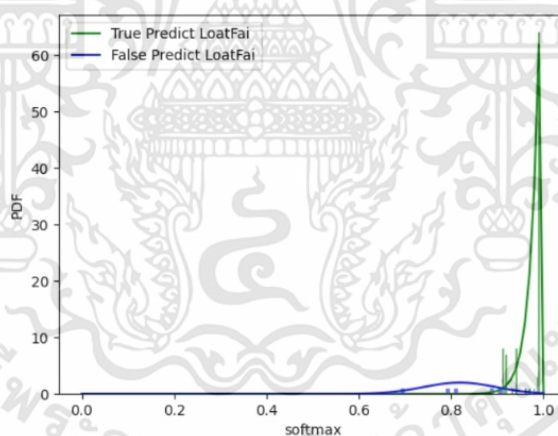
ค่าตัดสินใจของคำสั่งแอร์ เท่ากับ 0.89 ดังรูปที่ 4.19



รูปที่ 4.19 แสดงกราฟ Normal distribution ของชุดคำสั่งแอร์

- คำสั่งหลอดไฟ

ค่าตัดสินใจของคำสั่งหลอดไฟ เท่ากับ 0.89 ดังรูปที่ 4.20



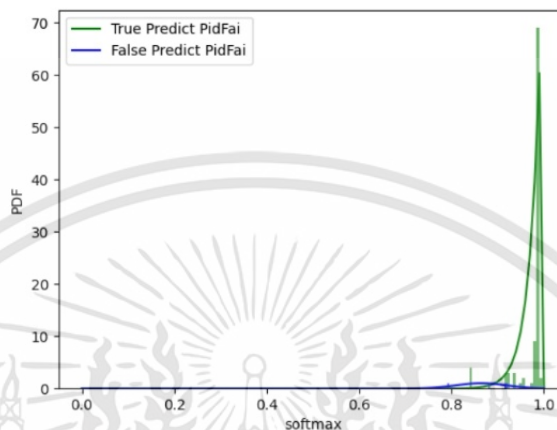
รูปที่ 4.20 แสดงกราฟ Normal distribution ของชุดคำสั่งหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. ส่วนโมเดลควบคุมหลอดไฟ

### - คำสั่งปิดไฟ

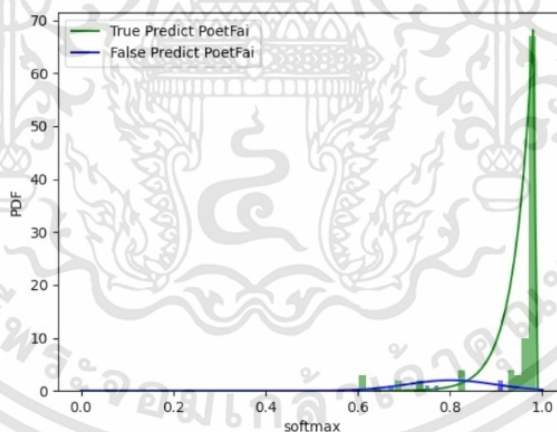
ค่าตัดสินใจของคำสั่งปิดไฟ เท่ากับ 0.89 ดังรูปที่ 4.21



รูปที่ 4.21 แสดงกราฟ Normal distribution ของชุดคำสั่งปิดไฟ

### - คำสั่งเปิดไฟ

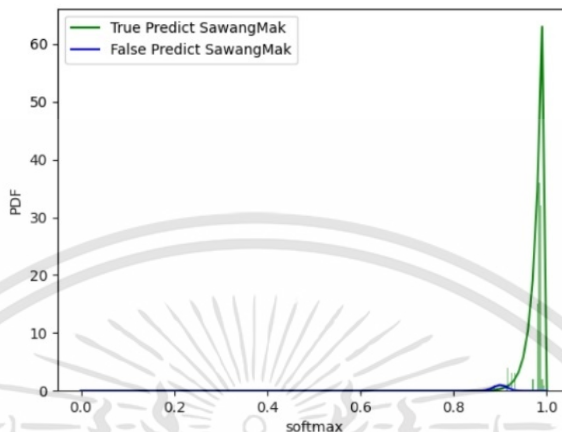
ค่าตัดสินใจของคำสั่งเปิดไฟ เท่ากับ 0.84 ดังรูปที่ 4.22



รูปที่ 4.22 แสดงกราฟ Normal distribution ของชุดคำสั่งเปิดไฟ

- คำสั่งสว่างมาก

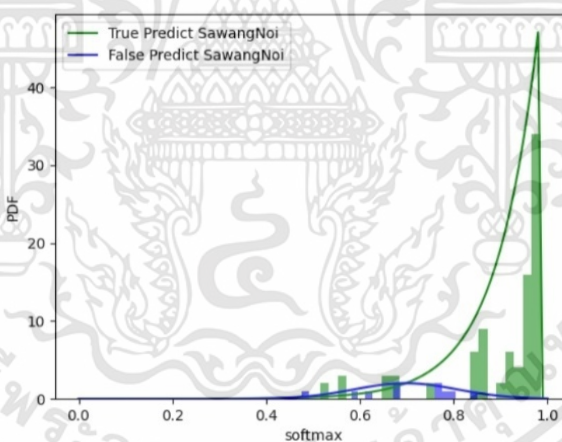
ค่าตัดสินใจของคำสั่งสว่างมาก เท่ากับ 0.86 ดังรูปที่ 4.23



รูปที่ 4.23 แสดงกราฟ Normal distribution ของชุดคำสั่งสว่างมาก

- คำสั่งสว่างน้อย

ค่าตัดสินใจของคำสั่งสว่างน้อย เท่ากับ 0.72 ดังรูปที่ 4.24



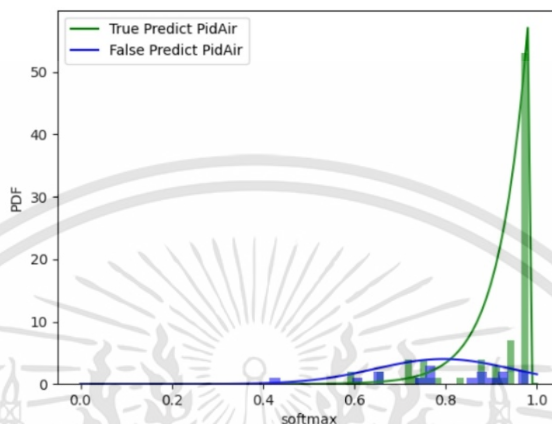
รูปที่ 4.24 แสดงกราฟ Normal distribution ของชุดคำสั่งสว่างน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ส่วนโมเดลควบคุมเครื่องปรับอากาศ

-คำสั่งปิดแอร์

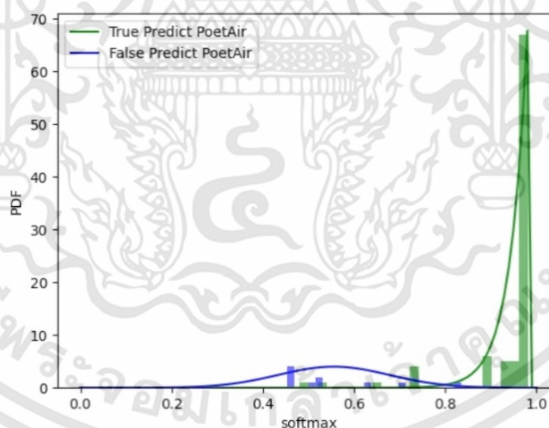
ค่าตัดสินใจของคำสั่งปิดแอร์ เท่ากับ 0.82 ดังรูปที่ 4.25



รูปที่ 4.25 แสดงกราฟ Normal distribution ของชุดคำสั่งปิดแอร์

-คำสั่งเปิดแอร์

ค่าตัดสินใจของคำสั่งเปิดแอร์ เท่ากับ 0.80 ดังรูปที่ 4.26

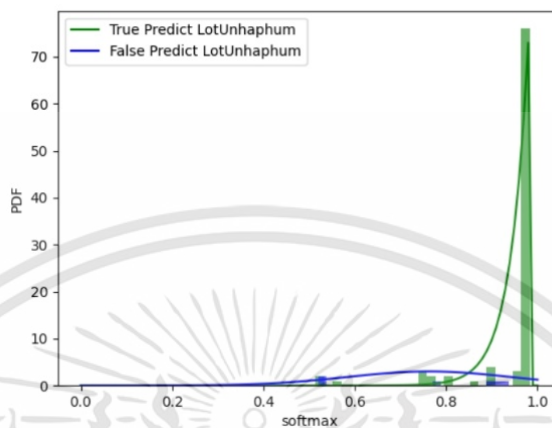


รูปที่ 4.26 แสดงกราฟ Normal distribution ของชุดคำสั่งเปิดแอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-คำสั่งลดอุณหภูมิ

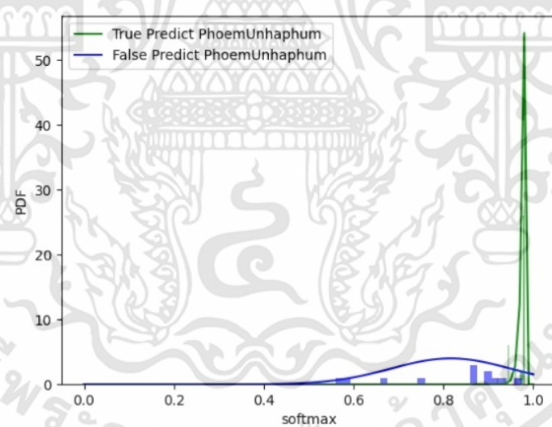
ค่าตัดสินใจของคำสั่งลดอุณหภูมิเท่ากับ 0.85 ดังรูปที่ 4.27



รูปที่ 4.27 แสดงกราฟ Normal distribution ของชุดคำสั่งลดอุณหภูมิ

-คำสั่งเพิ่มอุณหภูมิ

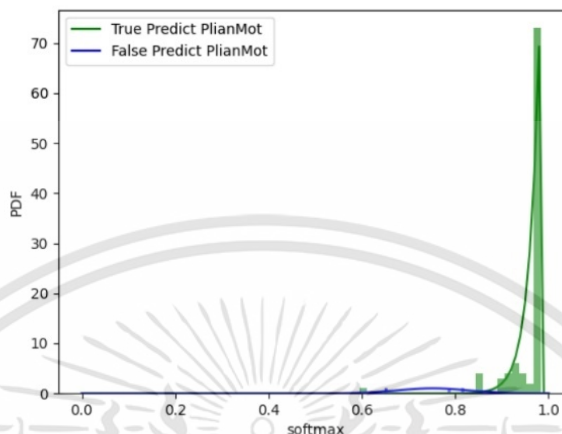
ค่าตัดสินใจของคำสั่งเพิ่มอุณหภูมิ เท่ากับ 0.95 ดังรูปที่ 4.28



รูปที่ 4.28 แสดงกราฟ Normal distribution ของชุดคำสั่งเพิ่มอุณหภูมิ

-คำสั่งเปลี่ยนโหมด

ค่าตัดสินใจของคำสั่งเปลี่ยนโหมด เท่ากับ 0.87 ดังรูปที่ 4.29



รูปที่ 4.29 แสดงกราฟ Normal distribution ของชุดคำสั่งเปลี่ยนโหมด

ในการใช้งานโมเดล เมื่อคำสั่งเสียงถูกป้อนเข้าสู่ระบบ ระบบจะนำคำสั่งเสียงมาทำนายค่าและนำค่าเอาต์พุตที่ได้จากการทำนายมาผ่านเงื่อนไขการตัดสินใจของระบบ หากค่าที่ได้ต่ำกว่าเงื่อนไขการตัดสินใจ ระบบจะไม่ทำงานและกลับไปเริ่มต้นรับอินพุตเสียงใหม่มาทำนายค่าอีกครั้ง จะช่วยให้สามารถตัดสินใจว่าจะนำคำสั่งไปใช้งานต่อหรือตัดคำสั่งนั้นทิ้งไปในกรณีที่คำสั่งไม่ใช่คำสั่งเสียงที่ผู้ใช้งานกำหนด หรือคำสั่งเสียงที่ทำนายค่าออกมาผิดพลาด ช่วยให้ระบบสั่งการด้วยเสียงสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น

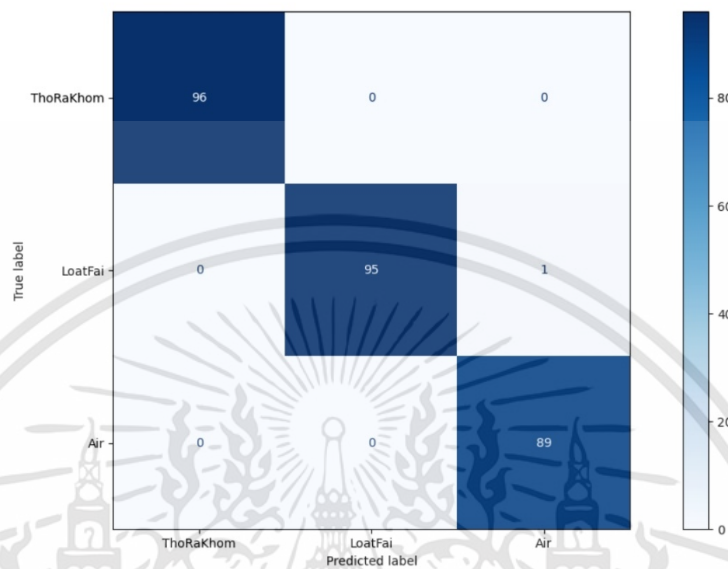
#### 4.5 ผลการทดสอบประสิทธิภาพของระบบสั่งการด้วยเสียงหลังตั้งค่าเงื่อนไขการตัดสินใจ (Threshold) ให้กับระบบ

จากผลการทดสอบหาค่าตัดสินใจให้กับระบบในหัวข้อที่ 4.4.3 ค่าตัดสินใจจะช่วยแก้ปัญหาในการที่ระบบทำนายค่าผิดพลาด ทำให้ระบบเปิดรับคำสั่งเอง ซึ่งจะนำค่าที่ได้จากหัวข้อที่ 4.4.3 ไปตั้งค่าเงื่อนไขการตัดสินใจให้กับระบบ และจะทำการทดสอบวัดประสิทธิภาพของโมเดล โดยทดสอบการทำนายค่าจากตาราง Confusion Matrix ของข้อมูลเสียงจากผู้ทดสอบจำนวน 10 คน โดยจะทดสอบจำนวน 12 คำสั่งคำสั่งเสียงละ 10 ครั้ง ซึ่งจะแบ่งโมเดลออกเป็น 3 ส่วน ได้แก่

1. ส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า
2. ส่วนโมเดลควบคุมหลอดไฟ
3. ส่วนโมเดลควบคุมเครื่องปรับอากาศ

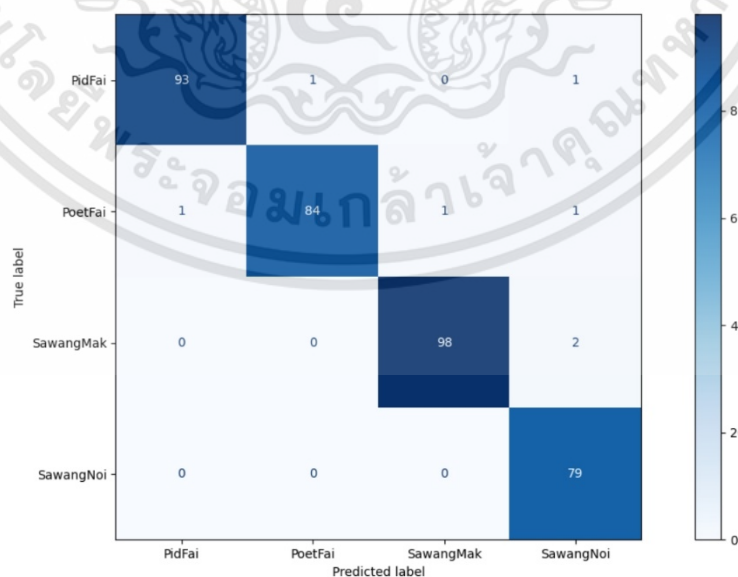
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า สามารถแสดง Confusion matrix จากผู้ทดสอบได้จากรูปที่ 4.30



รูปที่ 4.30 Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ  
ในส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า

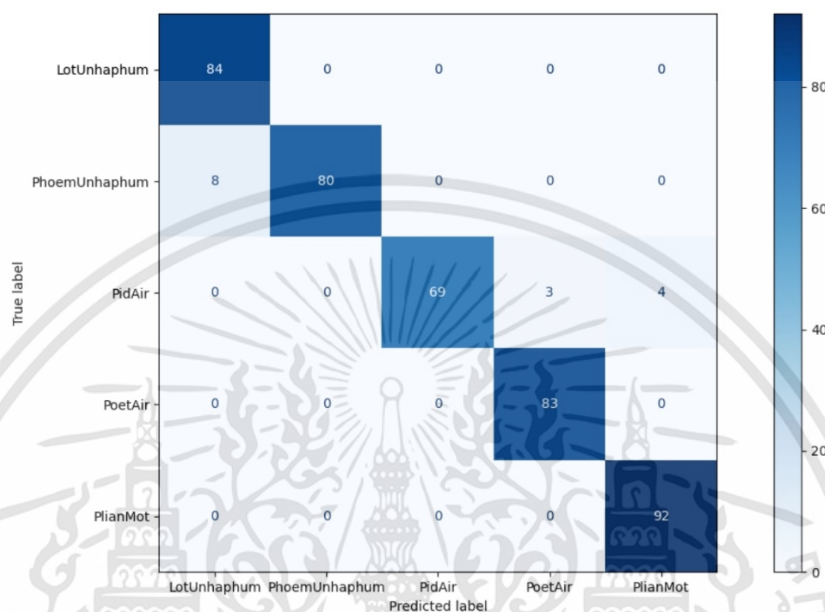
2. ส่วนโมเดลควบคุมหลอดไฟ สามารถแสดง Confusion matrix จากผู้ทดสอบได้จาก  
รูปที่ 4.31



รูปที่ 4.31 Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ  
ในส่วนโมเดลควบคุมหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนโมเดลควบคุมเครื่องปรับอากาศ สามารถแสดง Confusion matrix จากผู้ทดสอบได้จากรูปที่ 4.32



รูปที่ 4.32 Confusion matrix จากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบในส่วนโมเดลควบคุมเครื่องปรับอากาศ

จากรูปที่ 4.30 แสดง Confusion matrix ที่ได้จากชุดข้อมูลเสียงจากผู้ทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ พบว่าค่าชุดข้อมูลคำสั่งโทรคม (Thorakhom) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 96 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 4 ครั้ง ชุดคำสั่งหลอดไฟ (LoatFai) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 95 ครั้ง และทำนายผิดพลาดออกเป็นคำสั่งแอร์ 1 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 4 ครั้ง ชุดคำสั่งแอร์ (Air) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 89 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 11 ครั้ง

จากรูปที่ 4.31 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ พบว่าค่าชุดข้อมูลคำสั่งปิดไฟ (Pidfai) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 93 ครั้ง ทำนายค่าผิดพลาดออกเป็นคำสั่งเปิดไฟ 1 ครั้งและคำสั่งสว่างน้อย 1 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 5 ครั้ง ชุดคำสั่งเปิดไฟ (Poetfai) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 84 ครั้ง ทำนายผิดพลาดออกเป็นชุดคำสั่งปิดไฟ 1 ครั้ง คำสั่งสว่างมาก 1 ครั้งและคำสั่งสว่างน้อย 1 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 13 ครั้ง ชุดคำสั่งสว่างมาก (Sawangmak) สามารถทำนายค่าหลังผ่านเงื่อนไขค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัดสินใจได้ถูกต้อง 98 ครั้ง ทำนายค่าผิดพลาดเป็นคำสั่งสว่างน้อย 2 ครั้ง แสดงว่าสามารถทำนายค่าผ่านเงื่อนไขค่าตัดสินใจได้ทุกครั้ง และชุดคำสั่งสว่างน้อย (Sawangnoi) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 79 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 21 ครั้ง

จากรูปที่ 4.32 แสดง Confusion matrix ที่ได้จากชุดข้อมูลทดสอบหลังตั้งเงื่อนไขการตัดสินใจให้กับระบบ พบว่าค่าชุดข้อมูลคำสั่งลดอุณหภูมิ (LotUnhaphum) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 84 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 16 ครั้ง ชุดคำสั่งเพิ่มอุณหภูมิ (PhoemUnhaphum) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 80 ครั้ง ทำนายผิดพลาดเป็นคำสั่งเพิ่มอุณหภูมิ 8 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 12 ครั้ง ชุดคำสั่งปิดแอร์ (PidAir) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 69 ครั้ง ทำนายผิดพลาดเป็นคำสั่งเปิดแอร์ 3 ครั้ง คำสั่งเปลี่ยนโหมด 4 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 24 ครั้ง ชุดคำสั่งเปลี่ยนโหมด (PlianMot) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 92 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 8 ครั้ง และชุดคำสั่งเปิดแอร์ (PoetAir) สามารถทำนายค่าหลังผ่านเงื่อนไขค่าตัดสินใจได้ถูกต้อง 83 ครั้ง แสดงว่าทำนายค่าออกมาไม่ผ่านเงื่อนไขค่าตัดสินใจจำนวน 17 ครั้ง

จากการทดสอบทั้ง 3 โมเดลหลังจากตั้งค่าการตัดสินใจให้ พบว่าเมื่อนำไปใช้งานในระบบสั่งการด้วยเสียง สามารถช่วยให้ระบบมีประสิทธิภาพในการใช้งานดียิ่งขึ้น การตั้งค่าตัดสินใจให้กับระบบสามารถช่วยลดปัญหาเมื่อระบบทำนายข้อมูลที่ไม่ได้ต้องการสั่งให้ใช้งาน เช่น เสียงคนคุยกัน ช่วยลดปัญหาระบบเปิดใช้งานเอง หรือระบบทำนายค่าออกมาผิดพลาด

#### 4.6 ผลการทดสอบประสิทธิภาพของระบบสั่งการด้วยเสียง

ในส่วนของการทดสอบประสิทธิภาพของระบบสั่งการด้วยเสียง จะนำโมเดลมาใช้งานบน Raspberry pi ซึ่งป้อนอินพุตเป็นคำสั่งเสียงแบบเรียลไทม์ผ่านไมโครโฟน โดยทำการทดสอบการใช้งานควบคุมอุปกรณ์ด้วยเสียงออกเป็น 3 ส่วน ประกอบด้วยส่วนโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้า ส่วนโมเดลควบคุมหลอดไฟ และส่วนโมเดลควบคุมเครื่องปรับอากาศ ทำการทดสอบการป้อนคำสั่งเสียงคำสั่งละ 10 ครั้ง จำนวน 12 คำสั่งผ่านไมโครโฟน ประกอบด้วยการทดสอบจากผู้จัดทำ 3 คนและผู้ทดสอบ 7 คน นำมาหาค่าความแม่นยำในการทำนายค่า ซึ่งหากอินพุตที่ได้ค่าต่ำกว่าค่าตัดสินใจ (Threshold) จะเปรียบเสมือนทำนายค่าผิดพลาด สามารถแสดงค่าความแม่นยำของโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์ไฟฟ้างดตารางที่ 4.10-4.11 ค่าความแม่นยำของโมเดลในส่วนคำสั่งควบคุมหลอดไฟดงตารางที่ 4.12-4.13 และค่าความแม่นยำของโมเดลในส่วนคำสั่งควบคุมเครื่องปรับอากาศดงตารางที่ 4.14-4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5.1 โมเดลในส่วนของคำสั่งในการกระตุ้นการทำงานของระบบและระบุอุปกรณ์

##### ไฟฟ้า

ตารางที่ 4.10 ความแม่นยำของโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์

##### ไฟฟ้าของผู้จัดทำ

คำสั่ง ผู้จัดทำ	โทรคม (ค่ากระตุ้น)	แอร์	หลอดไฟ
ผู้จัดทำคนที่ 1	0.90	1.00	1.00
ผู้จัดทำคนที่ 2	1.00	0.80	0.90
ผู้จัดทำคนที่ 3	1.00	0.09	1.00

ตารางที่ 4.11 ความแม่นยำของโมเดลกระตุ้นการทำงานของระบบและระบุอุปกรณ์

##### ไฟฟ้าของผู้ทดสอบ

คำสั่ง ผู้ทดสอบ	โทรคม (ค่ากระตุ้น)	แอร์	หลอดไฟ
ผู้ทดสอบคนที่ 1	0.80	0.70	0.90
ผู้ทดสอบคนที่ 2	1.00	0.80	0.90
ผู้ทดสอบคนที่ 3	0.70	0.60	1.00
ผู้ทดสอบคนที่ 4	0.90	1.00	0.80
ผู้ทดสอบคนที่ 5	1.00	0.90	0.70
ผู้ทดสอบคนที่ 6	0.70	1.00	0.90
ผู้ทดสอบคนที่ 7	0.80	0.80	1.00

#### 4.5.2 โมเดลในส่วนของคำสั่งในการควบคุมหลอดไฟ

ตารางที่ 4.12 ความแม่นยำของโมเดลในการควบคุมหลอดไฟของผู้จัดทำ

คำสั่ง ผู้จัดทำ	ปิดไฟ	เปิดไฟ	สว่างมาก	สว่างน้อย
ผู้จัดทำคนที่ 1	1.00	0.90	0.80	1.00
ผู้จัดทำคนที่ 2	1.00	1.00	0.90	0.80
ผู้จัดทำคนที่ 3	1.00	0.90	0.90	1.00

ตารางที่ 4.13 ความแม่นยำของโมเดลในการควบคุมหลอดไฟของผู้จัดทำของผู้ทดสอบ

ผู้ทดสอบ	คำสั่ง	ปิดไฟ	เปิดไฟ	สว่างมาก	สว่างน้อย
ผู้ทดสอบคนที่ 1		0.70	0.80	1.00	0.70
ผู้ทดสอบคนที่ 2		1.00	0.90	0.70	0.60
ผู้ทดสอบคนที่ 3		0.90	0.60	0.80	0.70
ผู้ทดสอบคนที่ 4		0.80	1.00	0.90	0.80
ผู้ทดสอบคนที่ 5		0.90	0.90	0.70	1.00
ผู้ทดสอบคนที่ 6		1.00	1.00	0.80	0.70
ผู้ทดสอบคนที่ 7		0.90	0.80	0.70	0.60

#### 4.5.3 โมเดลในส่วนของคำสั่งในการควบคุมเครื่องปรับอากาศ

ตารางที่ 4.14 ความแม่นยำของโมเดลในการควบคุมเครื่องปรับอากาศของผู้จัดทำ

ผู้จัดทำ	คำสั่ง	ปิดแอร์	เปิดแอร์	ลดอุณหภูมิ	เพิ่มอุณหภูมิ	เปลี่ยนโหมด
ผู้จัดทำคนที่ 1		1.00	1.00	0.80	0.90	1.00
ผู้จัดทำคนที่ 2		1.00	1.00	0.90	1.00	1.00
ผู้จัดทำคนที่ 3		1.00	0.90	1.00	0.90	1.00

ตารางที่ 4.15 ความแม่นยำของโมเดลในการควบคุมเครื่องปรับอากาศของผู้จัดทำของผู้

ทดสอบ

ผู้ทดสอบ	คำสั่ง	ปิดแอร์	เปิดแอร์	ลดอุณหภูมิ	เพิ่มอุณหภูมิ	เปลี่ยนโหมด
ผู้ทดสอบคนที่ 1		0.90	0.80	0.60	0.90	0.90
ผู้ทดสอบคนที่ 2		0.80	1.00	0.70	1.00	0.80
ผู้ทดสอบคนที่ 3		0.90	1.00	0.90	0.80	0.90
ผู้ทดสอบคนที่ 4		1.00	0.80	0.90	0.90	0.70
ผู้ทดสอบคนที่ 5		0.80	0.90	0.80	0.60	0.90
ผู้ทดสอบคนที่ 6		0.70	0.70	0.90	0.70	0.80
ผู้ทดสอบคนที่ 7		1.00	0.90	0.80	1.00	0.90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.10-4.15 เมื่อนำโมเดลที่ได้จากแบบจำลอง CNN มาทดสอบความแม่นยำ โดยป้อนอินพุตเสียงแบบเรียลไทม์ผ่านไมโครโฟนบน Raspberry Pi แล้วนำไปควบคุมอุปกรณ์พบว่า ระบบมีประสิทธิภาพในการทำนายค่าค่อนข้างแม่นยำ ในส่วนของการทำนายค่าของเสียงอินพุตจากผู้จัดทำ ระบบสั่งการด้วยเสียงสามารถใช้งานได้ดี ไม่ค่อยมีความผิดพลาดในการทำนายค่า ในส่วนของการทำนายค่าของเสียงอินพุตที่ป้อนจากผู้ทดสอบ ระบบสั่งการด้วยเสียงสามารถใช้งานได้ค่อนข้างมีประสิทธิภาพเช่นกัน แต่จะมีความผิดพลาดที่มากกว่าในส่วนของผู้จัดทำ โดยการทำนายค่าผิดพลาดส่วนมากเกิดขึ้นในคำสั่งที่มีลักษณะเฉพาะแต่ละบุคคลพูด เช่น ลดอุณหภูมิ เพิ่มอุณหภูมิ และเนื่องจากยังมีข้อจำกัดในด้านเสียงรบกวนในการป้อนอินพุตแบบเรียลไทม์ทำให้เกิดความผิดพลาดในการทำนายค่าของโมเดล

ในส่วนของปัญหาการทำนายค่าที่ผิดพลาด ส่งผลให้ระบบเปิดใช้งานเอง ผู้จัดทำได้แก้ปัญหาโดยการตั้งค่าเงื่อนไขการตัดสินใจให้กับระบบ (Threshold) ทำให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น ลดปัญหาในการสั่งการอุปกรณ์ที่ผิดพลาด แต่ส่งผลให้ผู้ใช้งานต้องป้อนคำสั่งจำนวนหลายครั้ง

ในส่วนของการควบคุมอุปกรณ์ผ่านโมดูลอินฟราเรด ในการควบคุมหลอดไฟและเครื่องปรับอากาศสามารถทำงานได้ดีในการส่งสัญญาณ ไม่มีข้อผิดพลาดในการทำงาน แต่เนื่องจากมีข้อจำกัด คือ ต้องวางตัวส่งสัญญาณให้ตรงกับอุปกรณ์เครื่องใช้ไฟฟ้า ซึ่งหากผู้ใช้งานวางตัวส่งสัญญาณไม่ตรงกับทิศทางของอุปกรณ์จะทำให้มีความผิดพลาดในการควบคุมอุปกรณ์

## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

ปริญญานิพนธ์นี้เป็นการออกแบบระบบสั่งการด้วยเสียงที่สามารถควบคุมอุปกรณ์ด้วยการเสียงผ่าน Raspberry Pi เพื่อให้ผู้ใช้งานไม่จำเป็นต้องสัมผัสอุปกรณ์ ช่วยอำนวยความสะดวกให้แก่ผู้ใช้งาน สามารถควบคุมอุปกรณ์ผ่านเสียงได้

โดยตลอดภาคการศึกษาที่ 1 ผู้จัดทำได้ออกแบบและจัดทำระบบสั่งการด้วยเสียงที่ประกอบไปด้วยการเรียนรู้และออกแบบปัญญาประดิษฐ์ในการรู้จำเสียง (Voice Recognition) ทดสอบประสิทธิภาพในการทำนาย นำโมเดลไปประยุกต์ใช้ในการแปลงจากเสียงเป็นข้อความ (Speech to text) และในภาคการเรียนที่ 2 ผู้จัดทำได้เพิ่มอุปกรณ์ในการควบคุม คือ เครื่องปรับอากาศ ออกแบบให้ระบบสามารถทำงานได้อย่างเรียลไทม์และมีค่ากระตุ้นในระบบและสั่งการควบคุมไปยังอุปกรณ์ไฟฟ้า แต่ยังมีส่วนที่ยังสามารถนำไปพัฒนาได้ในการควบคุมอุปกรณ์ไฟฟ้าชนิดอื่น ๆ และมีประสิทธิภาพในการทำนายค่าที่แม่นยำภายใต้เสียงรบกวนจากภายนอก

#### 5.2 ข้อเสนอแนะ

ระบบสั่งการด้วยเสียงยังมีข้อจำกัดในเรื่องเสียงรบกวน (Noise) ส่งผลให้การใช้งานระบบคำสั่งการด้วยเสียงไม่สามารถทำงานได้อย่างมีประสิทธิภาพ และควรมีค่าความผิดพลาดที่ต่ำและความแม่นยำที่สูงกว่านี้ หากมีการนำชิ้นงานไปพัฒนาต่อจะมีแนวโน้มที่จะสามารถเพิ่มประสิทธิภาพการทำนายค่าโดยใช้การปรับปรุงค่าความผิดพลาดให้ต่ำกว่าเดิม และแก้ปัญหาด้านเสียงรบกวน จะส่งผลให้การใช้งานจริงผู้ใช้สามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

### บรรณานุกรม

- [1] Nithdanai Wangpratham. “เทคโนโลยีรู้จำเสียงพูด.”  
<https://nutdnuy.medium.com/เทคโนโลยีรู้จำเสียงพูด-automatic-speech-recognition-asr-9c40a16f2416>
- [2] Sean Allan. “The Three Tiers of AI: Automating tomorrow with AGI, ASI & ANI”  
<https://www.aware.co.th/three-tiers-ai-automating-tomorrow-agi-asi-ani/>
- [3] Pongthep Vijite. “ประเภทของ Machine Learning”  
<https://medium.com/coeffest/ประเภทของ-machine-learning-f3159fee7>
- [4] Zou, Jinming, Yi Han, and Sung-Sau So. 2009. “Overview of Artificial Neural Networks.” Pp. 14–22 in Artificial Neural Networks: Methods and Applications, Methods in Molecular Biology™, edited by D. J. Livingstone. Totowa, NJ: Humana Press.
- [5] R. Athieno, P. T. Jayachandran, and D. R. Themens. “A neural network-based foF2 model for a single station in the polar cap.” Radio Science, Vol. 52, No. 6, pp. 784-796, 2017
- [6] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang. “Deep Learning with Long Short-Term Memory for Time Series Prediction.” IEEE Communications Magazine, Vol. 57, No. 6, pp. 114-119, 2019
- [7] Pragati. “How to choose cross-entropy loss function in Keras”  
<https://androidkt.com/category/keras/>
- [8] จันทนา ปัญญาวรรณ. “ศึกษาความเป็นไปของการวิเคราะห์โรคอัลไซเมอร์จากภาพถ่ายทางการแพทย์.” โครงการวิจัยประเภทงบประมาณเงินรายได้จากเงินอุดหนุนรัฐบาล, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยบูรพา, 2561.
- [9] PradyaSin. “Convolution Neural Network คืออะไร”  
<https://medium.com/@pradyasin/what-is-convolution-neural-network-bf2e525089f5>
- [10] กานต์กมล ทวีผล. “แบบจำลองโครงข่ายประสาทเทียมแบบลึกสำหรับการทำนายปริมาณความหนาแน่นของฝุ่นละออง PM2.5 บริเวณพื้นที่จังหวัดกรุงเทพมหานครชั้นใน.” วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต, สาขาวิชาเทคโนโลยีสารสนเทศ, คณะวิทยาศาสตร์, มหาวิทยาลัยศรีนครินทรวิโรฒ, 2562.

- [11] C. Cesaroni, L. Spogli, A. Aragon-Angel, M. Fiocca, V. Dear, G. De Franceschi, and V. Romano, “Neural network-based model for global Total Electron Content forecasting,” C. Cesaroni et al.: J. Space Weather Space Clim., Vol. 10, No. 1, 2020
- [12] ชิตพงษ์ กิตตินราดร. “Neural Network Regularisation.”  
<https://guopai.github.io/ml-blog18.html>
- [13] Christoph Bartelmus. “What is LIRC ?”  
<https://lirc.org/>
- [14] marcuscode.com. “ภาษา Python.”  
<http://marcuscode.com/lang/python>
- [15] medium.com. “Linux คืออะไร.”  
<https://medium.com/@sprizebnz/linux>
- [16] seeedstudio.com. “Meet The Brand New Raspberry Pi 4 8GB.”  
<https://www.seeedstudio.com/blog/2020/05/28/meet-the-brand-new-raspberry-pi-4-8gb-ram/>
- [17] seeedstudio.com. “Specification of Raspberry Pi 4 Model B 8GB.”  
<https://www.seeedstudio.com/blog/2020/05/28/meet-the-brand-new-raspberry-pi-4-8gb-ram/>
- [18] raspberrypi.org. “GPIO”  
<https://www.raspberrypi.org/documentation/usage/gpio/>
- [19] ArduinoModules. “KY-005 INFRARED TRANSMITTER MODULE”  
<https://arduinomodules.info/ky-005-infrared-transmitter-sensor-module/>
- [20] ArduinoModules. “KY-022 INFRARED RECEIVER MODULE”  
<https://arduinomodules.info/ky-022-infrared-receiver-module/>
- [21] Jesada Winnaroch. “อินฟราเรด (Infrared)”  
<https://sites.google.com/site/jesadawin/xinfrared-infrared>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในงาน

<https://github.com/Clean610/AI-project->



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้