

การใช้คลื่น FMCW และ AI สำหรับการตรวจสอบอัตราการเต้นของหัวใจ
โดยไม่ต้องสัมผัส

HARNESSING FMCW AND AI FOR NON-INVASIVE HEART RATE
MONITORING



วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2566

KMITL-2023-EN-D-047-154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARNESSING FMCW AND AI FOR NON-INVASIVE HEART RATE
MONITORING



A THESIS SUBMITTED IN FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2022

KMITL-2023-EN-D-047-154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2022

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การใช้คลื่น FMCW และ AI สำหรับการตรวจสอบ อัตราการเต้นของหัวใจโดยไม่ต้องสัมผัส
นักศึกษา	นายรัชพล แคมภูเขียว
รหัสประจำตัว	62601077
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
พ.ศ.	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.ภัทรพงษ์ ผาสุขกิจ

บทคัดย่อ

การศึกษานี้เน้นการพัฒนาโมเดล AI เพื่อวัดสัญญาณหัวใจโดยไม่ต้องสัมผัสผิวหนังของผู้ป่วย โดยใช้คลื่นแม่เหล็กไฟฟ้าที่มีความถี่ 77GHz เพื่อตรวจจับการเต้นของหัวใจ และแยกสัญญาณออกเป็น 4 กลุ่มสำคัญ ได้แก่ ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด, ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง, ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด และช่วงที่ไม่ต้องวัดสัญญาณ เพื่อช่วยในการเลือกช่วงเวลาที่มีความแม่นยำที่สุดในการวัดสัญญาณ และเปรียบเทียบผล F1 ของโมเดล 3 รูปแบบ (BiLSTM classification, BiRNN classification และ BiGRU classification) ในคลาสช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุดและช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง พบว่าโมเดล BiLSTM classification มีความแม่นยำ F1 ที่ 97.14% (CLASS I) และ 98.02% (CLASS II) และโมเดล BiGRU classification มีความแม่นยำ F1 ที่ 98.10% (CLASS I) และ 97.74% (CLASS II) ซึ่งชี้ให้เห็นถึงความเป็นไปได้ในการพัฒนาเทคนิคในการวัดสัญญาณชีพโดยไม่ต้องสัมผัสผิวหนัง เพื่อเพิ่มความแม่นยำในการวัดและเพิ่มความสะดวกสบายให้กับผู้ป่วยโดยไม่ต้องสัมผัสผิวหนังได้ ซึ่งโมเดล BiLSTM classification และ BiGRU classification มีผลลัพธ์ที่ดีที่สุดและใกล้เคียงกัน เป็นทางเลือกที่เหมาะสมในการประยุกต์ใช้ในงานจริง

Thesis	HARNESSING FMCW AND AI FOR NON-INVASIVE HEART RATE MONITORING
Student	Mr. Ratchaphon Khaemphukhiao
Student ID.	62601077
Degree	Master of Engineering
Program	Electronics Engineering
Year	2019
Thesis Advisor	Assoc. Prof. Dr. Pattarapong Phasukkit

ABSTRACT


This study focuses on the development of an AI model for measuring heart signals without the need for direct skin contact with patients. It utilizes electromagnetic waves with a frequency of 77GHz to detect heartbeats and separates signals into four crucial intervals: the interval with the highest accuracy, the interval with moderate accuracy, the interval with the lowest accuracy, and the interval without signal measurement. This approach aids in selecting the most accurate time intervals for signal measurement, and the F1 score comparison of three model formats (BiLSTM classification, BiRNN classification, and BiGRU classification) in the intervals with the highest and moderate accuracy reveals that the BiLSTM classification model achieves an F1 accuracy of 97.14% (CLASS I) and 98.02% (CLASS II). Additionally, the BiGRU classification model achieves an F1 accuracy of 98.10% (CLASS I) and 97.74% (CLASS II). These results indicate the potential for developing non-contact physiological signal measurement techniques to enhance accuracy and convenience for patients. Both the BiLSTM and BiGRU classification models exhibit the best and closely comparable results, making them suitable options for real-world applications.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้เนื่องจากความช่วยเหลือจากท่านอาจารย์ที่ปรึกษา รศ.ดร.ภัทรพงษ์ ผาสุขกิจ ที่คอยให้คำปรึกษา ข้อเสนอแนะและสถานที่สำหรับทำงานวิจัย ตลอดจนถึงการช่วยแก้ไขปัญหาที่พบในระหว่างการทำวิจัย ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ในการทำงานวิจัยเพื่อนำไปใช้ในงานวิจัยขั้นสูงต่อไป จึงขอขอบพระคุณเป็นอย่างสูง

ข้าพเจ้าขอขอบพระคุณครอบครัวของข้าพเจ้า ที่สนับสนุนกำลังทรัพย์ แรงใจ แก่ข้าพเจ้าเสมอมา ขอขอบพระคุณพี่น้องชาวสมาชิกห้องปฏิบัติการ BURNLAB ที่คอยช่วยเหลือแลกเปลี่ยนความรู้กับข้าพเจ้าเสมอมาในตลอดช่วงที่กำลังศึกษาในระดับปริญญาโท จึงขอขอบพระคุณเป็นอย่างสูง มา ณ โอกาสนี้ด้วย

ข้าพเจ้าคาดหวังว่าวิทยานิพนธ์เล่มนี้จะเป็นประโยชน์ต่อบุคลากร นักศึกษาหรือผู้ที่สนใจในงานที่เกี่ยวข้อง เพื่อเป็นแนวทางในการพัฒนางานวิจัยให้สอดคล้องกับแนวทางเทคโนโลยีในปัจจุบัน



รัชพล แคมภูเขียว

สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	V
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญญา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ขอบเขตการวิจัย.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 ประวัติความเป็นมา.....	3
2.2 ปัญญาประดิษฐ์.....	7
2.2.1 ประวัติของปัญญาประดิษฐ์ หรือ Artificial Intelligence.....	8
2.2.2 การเรียนรู้ของเครื่อง (Machine learning).....	9
2.2.2.1 รูปแบบการเรียนรู้ของ Machine Learning.....	10
2.2.2.2 ตัวอย่างการประยุกต์ใช้การเรียนรู้ของเครื่อง (Machine learning)....	11
2.2.3 การเรียนรู้เชิงลึก (Deep Learning).....	12
2.2.3.1 การเรียนรู้ของโมเดล.....	13
2.2.3.2 ชนิดของโครงข่ายการเรียนรู้เชิงลึก (Deep Learning).....	13
2.2.3.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks :CNN).....	14
2.2.3.4 ตัวอย่างการประยุกต์ใช้การเรียนรู้เชิงลึก (Deep Learning).....	14
2.2.3.5 ข้อจำกัดของการเรียนรู้เชิงลึก (Deep Learning).....	15
2.3 ฟังก์ชันการกระตุ้น (Activation function).....	15
2.3.1 ฟังก์ชันการกระตุ้นแบบเชิงเส้น (Linear activation function).....	17
2.3.2 ฟังก์ชันการกระตุ้นที่ไม่ใช่เชิงเส้น (Non-Linear activation function).....	17
2.3.2.1 ฟังก์ชันซิกมอยด์ (Sigmoid function).....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.2.2 ฟังก์ชันแทน (Tanh function).....	18
2.3.2.3 ฟังก์ชัน ReLU (Rectified Linear Unit).....	18
2.3.2.4 ฟังก์ชันการรั่วของ ReLU (Leaky ReLU function).....	19
2.3.2.5 ฟังก์ชัน Softmax	20
2.4 การเคลื่อนลงตามความชัน (Gradient Descent).....	20
2.4.1 อัตราการเรียนรู้ (Learning rate).....	21
2.4.2 ฟังก์ชันความผิดพลาด (Error Function).....	22
2.4.2.1 การคำนวณความผิดพลาดโดยใช้ Cross Entropy.....	22
2.5 การแบ่งส่วนคลื่นแม่เหล็กไฟฟ้า 24GHz ของการหายใจโดยใช้ GMM.....	23
2.6 ทฤษฎีเกี่ยวกับ Long Short-Term Memory (LSTM).....	27
2.7 วัดประสิทธิภาพ Model จาก Confusion Matrix.....	31
2.8 โครงสร้างของหัวใจ.....	32
2.9 ปัจจัยที่มีผลต่อค่าอัตราการเต้นของหัวใจ (Heart Rate : HR).....	33
บทที่ 3 ระเบียบวิธีวิจัย.....	35
3.1 ระบบวิธีการวิจัย.....	35
3.2 การเก็บข้อมูล.....	37
3.3 การเตรียมชุดข้อมูล (Data preparing).....	41
3.3.1 การทำ Data Labeling.....	41
3.3.2 การเตรียมข้อมูลสำหรับเทรนโมเดล.....	43
3.4 การฝึกสอนโมเดล BiLSTM, BiRNN, BiGRU Classification.....	44
3.4.1 การประยุกต์ใช้งาน KFold.....	44
3.4.2 การฝึกสอนโมเดล.....	45
3.4.3 การประยุกต์ใช้ Decision Tree เพื่อทำ Ensemble Model.....	49
3.4.4 การวัดผล Ensemble Model.....	50
บทที่ 4 ผลการดำเนินงาน.....	52
4.1 การประเมินผลลัพธ์โมเดล.....	52
บทที่ 5 สรุปผลการดำเนินงาน.....	67
เอกสารอ้างอิง.....	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 PARAMETER FEATURE EXTRACTION LIST.....	24
ตารางที่ 2.2 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC ของการไม่หายใจ MFCC + 1 ST DELTA + 2 ND DELTA OF NOISE SIGNAL	25
ตารางที่ 2.3 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC ของการหายใจ MFCC + 1 ST DELTA + 2 ND DELTA OF RESPIRATION SIGNAL	25
ตารางที่ 3.1 สัญลักษณ์สำหรับการ label สัญญาณ	42
ตารางที่ 3.2 ผลลัพธ์จากโค้ดสำหรับการจัดเตรียมข้อมูลจากไฟล์ annotation.....	43
ตารางที่ 3.3 ตัวอย่างข้อมูลจำนวน 1 Batch สำหรับใช้ฝึกสอนโมเดล.....	44
ตารางที่ 4.1 Classification Report ของโมเดล BiLSTM.....	53
ตารางที่ 4.2 F1-score และค่าเฉลี่ย F1-score หรือค่าความแม่นยำในการพยากรณ์เฉลี่ยรวมทุกคลาสของโมเดล BiRNN.....	58
ตารางที่ 4.3 F1-score และค่าเฉลี่ย F1-score หรือค่าความแม่นยำในการพยากรณ์เฉลี่ยรวมทุกคลาสของโมเดล BiGRU.....	63
ตารางที่ 5.1 ตารางเทียบเทียบผล F1 ทั้ง 3 โมเดล โดยเจาะจงที่คลาสช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด และช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 (ก) การตรวจจับการหายใจ (ข) การตรวจจับการเต้นของหัวใจ	1
รูปที่ 2.1 สิทธิบัตรของอุปกรณ์ “เทเลเอ็นโมบิโลสโคป (Telemobiloscope)” ที่จดทะเบียนในปี ค.ศ. 1905 และผู้ประดิษฐ์เจ้าของสิทธิบัตร นายคริสเตียน ฮัลซ์เมเยอร์ (Christian Huelsmeyer) ..	3
รูปที่ 2.2 เรดาร์ภาคพื้นดิน (ก) เรดาร์เซนโฮม [6] (ข) เรดาร์เอส ซี อาร์-270.....	4
รูปที่ 2.3 เรดาร์ซีทัก ได้ถูกติดตั้งบนเรือรบของกองทัพเยอรมนี.....	5
รูปที่ 2.4 เรดาร์เนปจูน ได้ถูกติดตั้งบนเครื่องบินรบของกองทัพเยอรมนี.....	5
รูปที่ 2.5 เครื่องตีบลูของบริษัทไอบีเอ็ม	8
รูปที่ 2.6 เครื่องวัดสันของบริษัทไอบีเอ็ม.....	9
รูปที่ 2.7 ชนิดของปัญญาประดิษฐ์	9
รูปที่ 2.8 รูปแบบการทำงานของ Machine learning	10
รูปที่ 2.9 รูปแบบการเรียนรู้ของ Machine learning	11
รูปที่ 2.10 รูปแบบโครงสร้างระบบโครงข่ายประสาทของการเรียนรู้เชิงลึก (Deep Learning)	13
รูปที่ 2.11 ตัวอย่างโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent neural networks : RNN)	14
รูปที่ 2.12 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks : CNN)	14
รูปที่ 2.13 (ก) คือตัวอย่างโครงข่ายประสาทของมนุษย์, (ข) คือแบบจำลองทางคณิตศาสตร์ของโครงข่ายประสาท-เทียม.....	16
รูปที่ 2.14 กราฟลือตเอาต์พุตของข้อมูลโดยใช้ฟังก์ชันการกระตุ้นเชิงเส้น (Linear activation function)	17
รูปที่ 2.15 กราฟของฟังก์ชันซิกมอยด์ (Sigmoid)	18
รูปที่ 2.16 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชันซิกมอยด์และฟังก์ชันแทน.....	18
รูปที่ 2.17 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชันซิกมอยด์, ฟังก์ชันแทนและฟังก์ชัน ReLU.....	19
รูปที่ 2.18 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชัน ReLU และฟังก์ชันการรั่วของ ReLU.....	19
รูปที่ 2.19 รูปแบบการอัปเดตค่าน้ำหนักโดยใช้เทคนิคการเคลื่อนลงตามความชัน.....	21
รูปที่ 2.20 (ก) คืออัตราการเรียนรู้ที่มีค่าที่สูงเกินไป, (ข) คืออัตราการเรียนรู้ที่มีค่าน้อย	21
รูปที่ 2.21 กราฟลือตแสดงความสัมพันธ์ระหว่างค่าความคลาดเคลื่อน(Loss), ค่าเอาต์พุตที่เป็นค่าจริง (y) และค่าที่ได้จากการพยากรณ์ (yhat).....	22
รูปที่ 2.22 การสร้างแบบจำลองของการตรวจจับสัญญาณชีพเรดาร์ดอปเปลอร์ด้วยตัวรับส่งสัญญาณ.....	23
รูปที่ 2.23 ภาพอุปกรณ์สำหรับเก็บข้อมูล และแบบจำลองทารก	24
รูปที่ 2.24 สัญญาณจากอุปกรณ์ BIOPAC MP35	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.25 สัญญาณจากอุปกรณ์ RF transceiver.....	24
รูปที่ 2.26 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC สำหรับนำมาพัฒนาปัญญาประดิษฐ์	26
รูปที่ 2.27 แบบจำลอง AIC / BIC	26
รูปที่ 2.28 การเปรียบเทียบการแบ่งส่วนด้วยเทคนิค GMM ระหว่างสัญญาณหายใจ และไม่หายใจ	27
รูปที่ 2.29 แสดงโครงสร้างการทำงานของ RNN.....	27
รูปที่ 2.30 แสดงโครงสร้างการทำงานของ LSTM	28
รูปที่ 2.31 แสดงโครงสร้าง Forget gate layer	28
รูปที่ 2.32 แสดงโครงสร้าง Input gate layer.....	29
รูปที่ 2.33 แสดงโครงสร้าง Output gate layer	30
รูปที่ 2.34 แสดงความสัมพันธ์ระหว่าง Actual values กับ Predicted values.....	31
รูปที่ 2.35 ภายวิภาคของหัวใจ	32
รูปที่ 2.36 ตำแหน่งของลิ้นหัวใจ	33
รูปที่ 3.1 ระบบวิธีการวิจัย.....	35
รูปที่ 3.2 ภาพรวมการทำงาน	36
รูปที่ 3.3 ลักษณะการเก็บข้อมูลสัญญาณชีพ	38
รูปที่ 3.4 อุปกรณ์ mmWave EVM Kit	38
รูปที่ 3.5 อุปกรณ์ BioRadio	38
รูปที่ 3.6 อาสาสมัครหายใจแบบปกติต่อเนื่องและไม่ขยับตัวเป็นเวลา 5 นาที.....	39
รูปที่ 3.7 อาสาสมัครหยุดหายใจต่อเนื่องเป็นเวลา 15 วินาที สลับกับหายใจปกติเป็นเวลา 1 นาที และไม่ขยับตัว จำนวน 5 รอบ	40
รูปที่ 3.8 อาสาสมัครไม่อยู่ในพื้นที่เก็บสัญญาณชีพ เป็นเวลา 5 นาที.....	40
รูปที่ 3.9 อาสาสมัครหายใจแบบปกติและขยับตัวต่อเนื่องเป็นเวลา 5 นาที	41
รูปที่ 3.10 การทำ Label ด้วย label studio.....	41
รูปที่ 3.11 time series classification templates สำหรับ label studio.....	42
รูปที่ 3.12 ผลลัพธ์จากการ export ข้อมูลจาก label studio	42
รูปที่ 3.13 โค้ดสำหรับการจัดเตรียมข้อมูลจากไฟล์ annotation.....	43
รูปที่ 3.14 โค้ดสำหรับสร้างข้อมูลลำดับและป้ายกำกับ	44
รูปที่ 3.15 แบบจำลอง KFold.....	44
รูปที่ 3.16 โค้ดสำหรับสร้างโมเดล BiLSTM Classification	45
รูปที่ 3.17 โค้ดสำหรับสร้างโมเดล BiRNN Classification.....	46

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาตจากทางเรา

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.18 โค้ดสำหรับสร้างโมเดล BiGRU Classification.....	47
รูปที่ 3.19 โครงสร้างโมเดล BiRNN, BiRNN, BiGRU Classification	48
รูปที่ 3.20 โค้ดสำหรับสร้างเทรน Decision Tree	49
รูปที่ 3.21 กราฟ Decision Tree.....	49
รูปที่ 3.22 โค้ดสำหรับวัดผล precision, recall, f1-score.....	50
รูปที่ 3.23 โค้ดสำหรับวัดผล Ensemble Model ด้วย confusion matrix	51
รูปที่ 4.1 Confusion Matrix ของโมเดล BiLSTM Classification.....	52
รูปที่ 4.2 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiLSTM Classification	54
รูปที่ 4.3 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiLSTM Classification	55
รูปที่ 4.4 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiLSTM Classification	56
รูปที่ 4.5 Confusion Matrix ของโมเดล BiRNN Classification	57
รูปที่ 4.6 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiRNN Classification	59
รูปที่ 4.7 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiRNN Classification	60
รูปที่ 4.8 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiRNN Classification	61
รูปที่ 4.9 Confusion Matrix ของโมเดล BiGRU Classification	62
รูปที่ 4.10 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiGRU Classification	64
รูปที่ 4.11 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiGRU Classification	65
รูปที่ 4.12 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiGRU Classification	66

บทที่ 1

บทนำ (INTRODUCTION)

1.1 ความเป็นมาและความสำคัญของปัญหา

การวัดสัญญาณหัวใจ (Heart signal) มีบทบาทสำคัญอย่างยิ่งในการวินิจฉัยสภาพหัวใจและหลอดเลือด ส่งผลให้สามารถป้องกันความเสี่ยงของภาวะหัวใจวายฉับพลันที่เป็นสาเหตุการเสียชีวิตอันดับสูงในประชากรทั่วโลก การวัดสัญญาณดังกล่าวในอดีตมักใช้เซ็นเซอร์ที่ต้องสัมผัสกับผิวหนัง ซึ่งส่งผลให้เกิดปัญหาในกลุ่มเด็กทารกแรกเกิดและผู้ป่วยที่เป็นโรคผื่นสัมผัส อีกทั้งราคาของอุปกรณ์เหล่านี้ยังคงค่อนข้างสูง



รูปที่ 1.1 (ก) การตรวจจับการหายใจ (ข) การตรวจจับการเต้นของหัวใจ

ในปัจจุบัน มีงานวิจัยที่ต้องการนำคลื่นแม่เหล็กไฟฟ้ามาประยุกต์ใช้ในการวัดสัญญาณหัวใจ โดยไม่ต้องสัมผัสผิวหนัง ซึ่งเป็นเทคนิคที่สามารถสกัดคัดกรองได้อย่างรวดเร็ว มีขนาดเล็กและราคาถูก โดยใช้คลื่นแม่เหล็กไฟฟ้าที่ความถี่ 77GHz ในการตรวจจับการเปลี่ยนแปลงของคลื่นดอปเปลอร์จากการเต้นของหัวใจ แต่การวิจัยในเวลานี้กำลังเผชิญปัญหาในการที่ผู้ป่วยเคลื่อนไหวทำให้การวัดไม่แม่นยำ

ในงานวิจัยนี้ เราจะนำเสนอการใช้โมเดล AI ในการแยกสัญญาณออกเป็น 4 กลุ่ม ได้แก่ 1) ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด 2) ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง 3) ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด และ 4) ช่วงที่ไม่ต้องวัดสัญญาณ ซึ่งจะช่วยให้เราสามารถเลือกช่วงเวลาที่มีความแม่นยำที่สุดในการวัดได้

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 การพัฒนาโมเดล AI ที่สามารถทำการวัดสัญญาณหัวใจโดยไม่ต้องสัมผัสผิวหนังผู้ป่วย: โดยเฉพาะในกลุ่มเช่น เด็กทารกแรกเกิด และผู้ที่มีโรคผิวหนังสัมผัส

1.2.2 การทดลองและปรับปรุงโมเดล AI ที่สามารถจำแนกสัญญาณหัวใจในหลากหลายสถานะ: รวมถึงช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด, ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง, ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด และช่วงที่ไม่ต้องวัดสัญญาณ

1.2.3 การวิจัยและพัฒนาเทคนิคในการใช้คลื่นแม่เหล็กไฟฟ้า FMCW ความถี่ 77GHz ในการตรวจจับดอปเพลอร์ของหัวใจ: ซึ่งจะเป็นวิธีการที่ไม่ต้องสัมผัสผิวหนัง

1.3 สมมติฐานของการศึกษา

1.3.1 โมเดล AI ที่ถูกฝึกให้รับรู้สัญญาณในช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด และช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง จะสามารถวัดสัญญาณของหัวใจได้อย่างแม่นยำ ด้วยคลื่นแม่เหล็กไฟฟ้าความถี่ 77 GHz

1.3.2 การรวมกลุ่มสัญญาณหัวใจเป็นส่วนต่าง ๆ ที่มีความแตกต่างในการหายใจและการเคลื่อนไหวจะช่วยให้โมเดล AI ทำงานได้ดียิ่งขึ้น

1.3.3 โมเดล AI ที่ปรับปรุงแล้วจะสามารถปรับใช้ได้กับเทคโนโลยีการวัดหัวใจที่ไม่ต้องสัมผัสผิวหนัง ทำให้สามารถลดความเสี่ยงที่เกิดจากการสัมผัสผิวหนังและเพิ่มความสะดวกสบายให้กับผู้ป่วย

1.4 ขอบเขตการวิจัย

1.4.1 ศึกษาและการพัฒนาโมเดล AI สำหรับการจำแนกสัญญาณหัวใจ: สิ่งนี้จะรวมถึงการทดสอบและปรับปรุงโมเดล AI ในหลาย ๆ สถานะเช่น ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด, ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง, ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด และช่วงที่ไม่ต้องวัดสัญญาณ

1.4.2 การใช้คลื่นแม่เหล็กไฟฟ้าความถี่ 77GHz ในการวัดสัญญาณหัวใจ: รวมถึงการทดลองและทดสอบวิธีการที่เหมาะสมที่สุดสำหรับการจับความถี่ดอปเพลอร์

โปรดทราบว่า การวิจัยนี้ไม่ได้รวมถึงการพัฒนาเซ็นเซอร์หรืออุปกรณ์ที่ใช้ในการวัดสัญญาณหัวใจ เราจะมุ่งเน้นไปที่การปรับปรุงและประยุกต์ใช้เทคนิคที่มีอยู่แล้วในการวัดสัญญาณดอปเพลอร์และการพัฒนาโมเดล AI เพื่อการจำแนกสัญญาณหัวใจที่เป็นอิสระจากการสัมผัสผิวหนัง

1.5 งานวิจัยที่เกี่ยวข้อง

1.5.1 Segmentation of 24GHz RF Respiration Signal Using GMM: งานวิจัยนี้นำเสนอเทคนิคในการ Segmentation เพื่อสามารถคำนวณ Respiration rate ให้เกิดความแม่นยำมากขึ้น

1.5.2 A Survey of mmWave-based Human Sensing: Technology, Platforms and Applications: งานวิจัยนี้ศึกษา FMCW Radar รวมถึงเทคนิคต่างๆ ได้แก่ Signal Processing, Feature Extraction และ Sensing Model ตลอดจน application ด้าน Smart Health

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

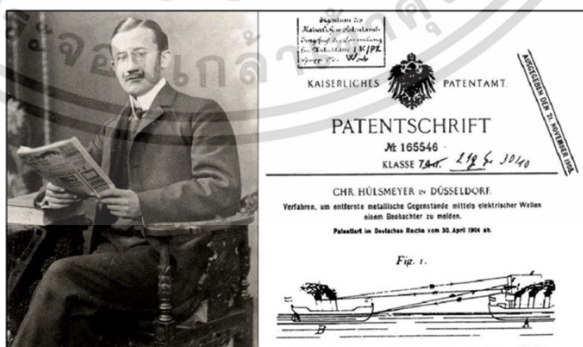
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงประวัติและความเป็นมาทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับ FMCW เรดาร์ในการตรวจจับการการเต้นของหัวใจ และความรู้การแพทย์ รวมถึงทฤษฎีที่เกี่ยวข้องกับการพัฒนาโมเดลที่ใช้จำแนกสัญญาณ

2.1 ประวัติความเป็นมา

ความเป็นมาของเรดาร์นั้นเริ่มต้นในช่วงปลายศตวรรษที่ 19 เมื่อมีการค้นพบคลื่นแม่เหล็กไฟฟ้า โดยในปี ค.ศ. 1864 นักฟิสิกส์ชาวสก็อตแลนด์ นายเจมส์ เคลิร์ก แมกซ์เวล (James Clerk Maxwell) ได้ทำการค้นคว้าทฤษฎีและสมการอธิบายพฤติกรรมคลื่นแม่เหล็กไฟฟ้าขึ้น ซึ่งต่อมาในปี ค.ศ. 1886 นักฟิสิกส์ชาวเยอรมัน นายไฮน์ริช เฮิร์ตซ์ (Heinrich Hertz) ได้แสดงการทดลองเพื่อยืนยันทฤษฎีคลื่นแม่เหล็กไฟฟ้าของแมกซ์เวลล์ (Maxwell's equation) ทั้งนี้ยังได้ค้นพบว่าคลื่นแม่เหล็กไฟฟ้ามีคุณสมบัติสามารถสะท้อนกับวัตถุที่เป็นโลหะได้ [1] ในปี ค.ศ. 1900 นายนิโคลาส เทสลา (Nikola Tesla) วิศวกรชาวสหรัฐฯ ได้เสนอแนวความคิดเรื่องการใช้คลื่นแม่เหล็กไฟฟ้าเพื่อการตรวจจับและวัดความเร็วของเป้าหมาย โดยมีหลักการเกี่ยวกับการสะท้อนก้องของเสียง (echo) [2]

ในช่วงศตวรรษที่ 20 ตอนต้น ได้มีพัฒนาการนำมาใช้อย่างเป็นทางการ ในปี ค.ศ. 1904 นายคริสเตียน ฮัลซ์เมเยอร์ (Christian Huelsmeyer) นักฟิสิกส์ชาวเยอรมันทำการประดิษฐ์อุปกรณ์ ที่เรียกว่า “เทเลเอ็นโมบิโลสโคป (Telemobiloscope)” ซึ่งใช้สำหรับการตรวจจับเรือในทะเลท่ามกลางหมอกหนา ภายในระยะ 2 ไมล์ ด้วยการส่งคลื่นแม่เหล็กไฟฟ้าแบบต่อเนื่อง (Continuous wave; CW) ที่ความถี่ 650 MHz แต่อย่างไรก็ตามอุปกรณ์ยังไม่สามารถบอกระยะห่างของเป้าหมายได้ เพื่อป้องกันปัญหาเรือชนกัน เป็นครั้งแรกที่มีการใช้คลื่นแม่เหล็กไฟฟ้าในการตรวจจับวัตถุที่เป็นโลหะที่ระบบมีทั้งภาคส่งและภาครับ [3], [4]



รูปที่ 2.1 สิทธิบัตรของอุปกรณ์ “เทเลเอ็นโมบิโลสโคป (Telemobiloscope)” ที่จดทะเบียนในปี ค.ศ. 1905 และผู้ประดิษฐ์เจ้าของสิทธิบัตร นายคริสเตียน ฮัลซ์เมเยอร์ (Christian Huelsmeyer)[9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จนกระทั่งในปี ค.ศ.1924 มีการทดลองเพื่อระบุความสูงของชั้นบรรยากาศไอโอโนสเฟียร์ ด้วยหลักการสะท้อนของคลื่นแม่เหล็กไฟฟ้าย่านความถี่วิทยุ (radio echoes) โดยนักฟิสิกส์ชาวอังกฤษ เซอร์ เอ็ดเวิร์ด วิกเตอร์ แอปเปิ้ลตัน (Sir Edward Victor Appleton) ซึ่งประสบความสำเร็จและถือเป็นครั้งแรกที่มีการใช้คลื่นแม่เหล็กไฟฟ้าในการวัดระยะทาง ต่อมาในปี ค.ศ. 1935 นายโรเบิร์ต วัตสัน วัตต์ (Robert Watson-Watt) ได้เสนอวิธีการตรวจจับอากาศยานโดยใช้คลื่นแม่เหล็กไฟฟ้าและมีการทดลองสาธิตในงานวิจัยหัวข้อ “The Detection of Aircraft by Radio Methods” ซึ่งใช้ตั้งอยู่บนหลักการส่งคลื่นแม่เหล็กไฟฟ้าไปกระทบวัตถุ (radio wave bouncing) แล้ววัดข้อมูลจากการเดินทางของคลื่นในขณะเดียวกันทางฝั่งสหรัฐอเมริกา นายโรเบิร์ต เอ็ม เพจ (Robert M. Page) ได้ทำการทดลองส่งคลื่นแม่เหล็กไฟฟ้าเป็นสัญญาณพัลส์ความถี่ 60 MHz สามารถตรวจจับเครื่องบินที่อยู่ห่างออกไป 1.6 กิโลเมตร และในปี ค.ศ. 1936 สามารถพัฒนาเป็นการใช้สัญญาณความถี่ 28.6 MHz ตรวจจับเครื่องบินที่อยู่ห่างออกไป 40 กิโลเมตร ซึ่งนายเพจเป็นบุคลากรคนสำคัญในการพัฒนาเทคโนโลยีทางการตรวจจับโดยคลื่นแม่เหล็กไฟฟ้าของห้องปฏิบัติการวิจัยกองทัพเรือสหรัฐอเมริกา จนต่อมาในปี ค.ศ. 1940 กองทัพเรือสหรัฐอเมริกาได้ให้ชื่อเรียกเทคโนโลยีนี้ว่า “เรดาร์ (Radar)” [3], [5]



(ก)

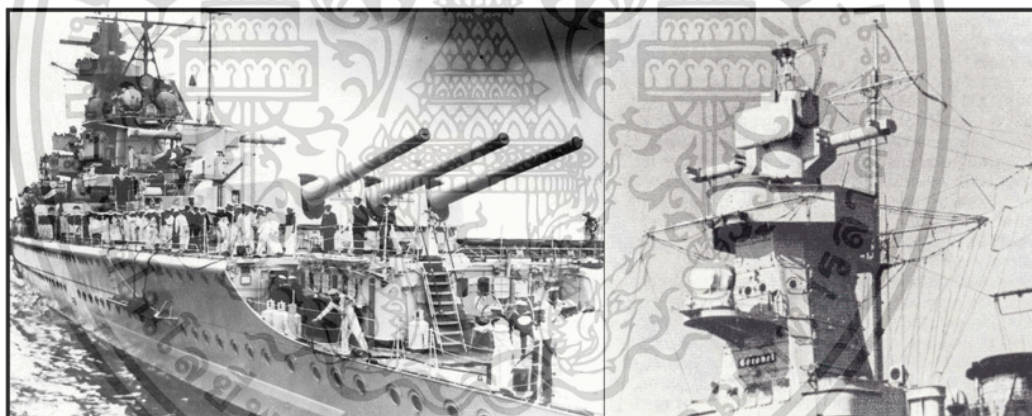
(ข)

รูปที่ 2.2 เรดาร์ภาคพื้นดิน (ก) เรดาร์เซนโฮม [6] (ข) เรดาร์เอส ซี อาร์-270[3]

ในยุคสงครามโลกครั้งที่ 2 (World war II) เรดาร์ได้มีวิวัฒนาการอย่างต่อเนื่องและเริ่มมีบทบาทเป็นอย่างมากโดยชาติมหาอำนาจในยุคนั้นที่มีการศึกษาระบบเรดาร์ เช่น สหรัฐอเมริกา สหราชอาณาจักร เยอรมนี เป็นต้น โดยเมื่อปี ค.ศ. 1937 ประเทศสหราชอาณาจักรได้นำผลงานที่นายโรเบิร์ต วัตสัน วัตต์ ได้พัฒนาขึ้นและจดสิทธิบัตรไปเมื่อปี ค.ศ. 1935 มาเป็นต้นแบบในการพัฒนาระบบเรดาร์สำหรับการเตือนภัยคุกคาม เรียกว่า “เซนโฮม (Chain Home; CH)” ซึ่งมีความถี่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ใช้งาน 22 MHz สามารถตรวจจับอากาศยานได้ที่ความสูง 3,000 เมตร และระยะทางการตรวจจับที่ 150 กิโลเมตร [6] ภายใต้ความเกรงกลัวที่ว่าทางประเทศเยอรมนีจะสามารถพัฒนาอาวุธที่เรียกว่า “รังสีแห่งความตาย (death-ray)” ซึ่งเกรงว่าจะเป็นภัยคุกคามต่อทางสหราชอาณาจักร ต่อมาระบบเรดาร์เตือนภัยคุกคามนี้ได้ถูกนำมาติดตั้งเป็นระบบเตือนภัยชายฝั่งในบริเวณทางตะวันออกเฉียงใต้ของเกาะอังกฤษในปี ค.ศ. 1939 และมีบทบาทสำคัญในยุทธการบริเตน (Battle of Britain) ที่เพิ่มความแม่นยำในการค้นหาและโจมตีข้าศึก [3], [5]

ในทางฝั่งสหรัฐอเมริกาได้มีการพัฒนาเรดาร์ภาคพื้นดินชื่อว่า “เอส ซี อาร์-270 (SCR-270)” ซึ่งเตือนภัยคุกคามก่อนการโจมตีของกองทัพญี่ปุ่นเป็นเวลาประมาณชั่วโมงก่อนเกิดเหตุการณ์โจมตีเพิร์ลฮาร์เบอร์ (Attack on Pearl Harbor) เมื่อปี ค.ศ. 1941 นอกจากนี้ยังได้มีการโครงการร่วมระหว่าง สหราชอาณาจักรและสหรัฐอเมริกา เพื่อการพัฒนาเครื่องแมกนีตรอนกำลังสูงสำหรับเรดาร์ (magnetron for high-power microwave radar) ซึ่งประสบผลสำเร็จและประดิษฐ์ “แมกนีตรอนเรโซแนนซ์ คิววิตี้ (Resonant cavity magnetron)” สามารถผลิตสัญญาณพัลส์ความถี่สูงที่มีกำลังสูงถูกนำไปใช้เป็นส่วนหนึ่งของระบบตรวจจับ “ลิดาร์ (Light detection and ranging)” ที่สามารถตรวจจับในสภาพอากาศทุกรูปแบบโดยปัจจุบันได้ถูกนำมาใช้ในการสื่อสารและวัดคุณภาพอากาศ [3], [5]



รูปที่ 2.3 เรดาร์ซีทัก ได้ถูกติดตั้งบนเรือรบของกองทัพเยอรมนี [7]



รูปที่ 2.4 เรดาร์เนปจูน ได้ถูกติดตั้งบนเครื่องบินรบของกองทัพเยอรมนี [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเทศเยอรมนีเป็นอีกชาติมหาอำนาจที่มีความเชี่ยวชาญทางด้านการพัฒนาเทคโนโลยีเรดาร์ ซึ่งได้พัฒนาระบบปฏิบัติการเรดาร์อีกหลายแบบในระหว่างยุคสงครามโลกครั้งที่ 2 จุดเริ่มต้นในการพัฒนาเริ่มขึ้นในปี ค.ศ. 1933 นายรูดอล์ฟ कुโนด์ (Rudolph Kuhnold) หัวหน้าโครงการวิจัยทางด้านการวิเคราะห์สัญญาณของกองทัพเรือเยอรมัน ได้ทำการค้นคว้าวิธีการตรวจจับเป้าหมายใต้น้ำโดยใช้คลื่นเสียง ที่เรียกว่า “โซนาร์ (sonar)” หลังจากนั้นเขาได้ทำการพัฒนาเรดาร์สำหรับการตรวจจับเรือในทะเลด้วยตนเองมาจนถึงปี ค.ศ. 1935 มีการทดลองระบบซึ่งสามารถระบุตำแหน่งเรือที่จอดอยู่ห่างจากฝั่งได้ไกล 12 ไมล์ และเรือที่เคลื่อนที่ได้ไกล 5 ไมล์ หลังจากนั้น บริษัทเจมา (Gema company) ได้ดำเนินการพัฒนาโดยการเพิ่มความถี่การใช้งานเป็น 600 MHz และสูงกว่า พบว่าสามารถตรวจจับเป้าหมายได้ไกลขึ้นและละเอียดมากขึ้น ต่อมาในปี ค.ศ. 1936 มีการพัฒนาระบบเรดาร์เตือนภัยคุกคามที่สำคัญของเยอรมนีในช่วงแรกของวิวัฒนาการเรดาร์เยอรมัน สามารถตรวจจับอากาศยานที่อยู่ไกลได้ 50 ไมล์ มีชื่อเรียกว่า “เรดาร์เฟรย่า (Freya radar)” ต่อมาในปี ค.ศ. 1937 ได้มีการประดิษฐ์ระบบเรดาร์ปฏิบัติการบนเรือสำเร็จเป็นครั้งแรกของโลก เรียกว่า “เรดาร์ซีทัก (Seetakt radar)” มีความถี่ปฏิบัติการที่ 375 MHz ตรวจจับได้ในระยะสั้นภายใน 9 ไมล์ เรดาร์ทั้งสองถูกนำมาใช้โดยกองทัพเรือเยอรมนี ในปี ค.ศ. 1938 [38] นอกจากนี้ยังมีการพัฒนาระบบเรดาร์ปฏิบัติการบนอากาศยาน เช่น เรดาร์เนปจูน (Neptun radar) และเรดาร์ลิกเตนสไตน์ (Lichtenstein radar) [3], [7]

จากรายละเอียดที่กล่าวมาสามารถระบุได้ว่าในยุคสงครามโลกครั้งที่ 2 เป็นยุคที่มีการพัฒนาระบบการตรวจจับเรดาร์และนำไปใช้งานอย่างกว้างขวางมาก โดยเฉพาะการตรวจจับทางไกล (remote sensing) เพื่อการสังเกตการณ์ข้าศึกการเตือนภัยก่อนการจู่โจมแก่พลเรือนช่วยระบุตำแหน่งเพื่อการโจมตีทั้งในภาคพื้นดิน ทางน้ำ และอากาศ ในสภาพแวดล้อมทั้งกลางวัน กลางคืน และสภาพอากาศทุกแบบซึ่งบางครั้งเป็นการยากที่มนุษย์จะสังเกตการณ์ด้วยตนเอง

หลังจากสงครามโลกครั้งที่ 2 สิ้นสุดลง ได้มีการพัฒนาเทคนิคสำหรับระบบการตรวจจับเรดาร์ในยุคใหม่ เช่น ระบบช่องเปิดสังเคราะห์ (Synthetic aperture radar; SAR) ระบบเรดาร์เฟสอาร์เรย์ (phase array radar) ระบบระบุเป้าหมายเคลื่อนที่ (Moving target indication; MTI) ระบบเรดาร์ตรวจตราอันดับสอง (Secondary surveillance radar; SSR) และระบบเรดาร์พัลส์ดอปเปลอร์ (Pulsedoppler radar) เป็นต้น ซึ่งนำมาสู่การพัฒนาเทคโนโลยีที่เป็นประโยชน์ต่อการนำมาใช้ประโยชน์ในชีวิตประจำวันด้านอื่นมาจนถึงปัจจุบัน เช่น เรดาร์ตรวจสอบสภาพอากาศเพื่อใช้ในการพยากรณ์อากาศ เรดาร์ตรวจจับความเร็วยานพาหนะสำหรับการปฏิบัติหน้าที่ของเจ้าหน้าที่ตำรวจ เรดาร์ในระบบรถยนต์ [8], [9] เป็นต้น

2.2 ปัญญาประดิษฐ์ (Artificial Intelligence : AI)

ปัญญาประดิษฐ์ (Artificial Intelligence) คือ เครื่องจักร(machine) ที่มีฟังก์ชันที่มีความสามารถในการทำความเข้าใจ เรียนรู้องค์ความรู้ต่างๆ อาทิเช่น การรับรู้ การเรียนรู้ การให้เหตุผลและการแก้ปัญหาต่างๆ เครื่องจักรที่มีความสามารถเหล่านี้ก็คือว่าเป็นปัญญาประดิษฐ์ เพราะฉะนั้นจึงสามารถกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ว่าปัญญาประดิษฐ์ถือกำเนิดขึ้นเมื่อเครื่องจักรมีความสามารถที่จะเรียนรู้ตัวเอง ซึ่งปัญญาประดิษฐ์ก็ถูกแบ่งออกเป็นหลายระดับตามความสามารถหรือความฉลาด โดยจะวัดจากความสามารถในการให้เหตุผล การพูดและทัศนคติของปัญญาประดิษฐ์ตัวนั้นๆ เมื่อเปรียบเทียบกับมนุษย์อย่างเราๆ ปัญญาประดิษฐ์ถูกจำแนกเป็น 3 ระดับตามความสามารถหรือความฉลาด ดังนี้

1. Artificial Narrow Intelligence (ANI) หรือ Weak AI คือ ปัญญาประดิษฐ์ที่มีความสามารถเฉพาะทางได้ดีกว่ามนุษย์ อาทิ เช่นปัญญาประดิษฐ์ที่ช่วยในการผ่าตัด (AI-assisted robotic surgery) ที่อาจจะเชี่ยวชาญเรื่องการผ่าตัดกว่าหมอยุคปัจจุบัน แต่แน่นอนว่าปัญญาประดิษฐ์ตัวนี้ไม่สามารถที่จะทำอาหาร ร้องเพลง หรือทำสิ่งอื่นที่นอกเหนือจากการผ่าตัดได้นั่นเอง ซึ่งผลงานวิจัยด้านปัญญาประดิษฐ์ ณ ปัจจุบันยังอยู่ที่ระดับนี้

2. Artificial General Intelligence (AGI) หรือ Strong AI คือ ปัญญาประดิษฐ์ที่มีสติปัญญาและสามารถทำงานต่าง ๆ ได้เทียบเท่ากับสมองของมนุษย์ เช่น สามารถคิดวิเคราะห์ วางแผน แก้ปัญหาที่ซับซ้อน เข้าใจเรื่องราวที่เป็นนามธรรม รวมถึงสามารถเรียนรู้จากประสบการณ์ในอดีตได้เหมือนกับที่มนุษย์ทำได้ แต่ในปัจจุบันเรายังไม่สามารถพัฒนา AI ชนิดนี้ขึ้นมาได้สำเร็จ

3. Artificial Super intelligence (ASI) เป็นสุดยอด AI ที่มีปัญญาเหนือกว่ามนุษย์ และแน่นอนว่าในปัจจุบันเรายังไม่สามารถพัฒนา AI ประเภทนี้ขึ้นมาได้ แต่ไอเดียการมีอยู่ของ Artificial Super intelligence (ASI) นั้น ซึ่งมักจะปรากฏอยู่ในสื่อต่าง ๆ มากมาย อาทิ เช่น เกมส์ ภาพยนตร์ ซีรีส์ หรือนวนิยาย

โดยสรุปปัญญาประดิษฐ์หรือ AI เป็นเทคโนโลยีที่ล้ำสมัยที่สามารถรับมือกับปัญหาที่ซับซ้อนเกินกว่าที่มนุษย์จะสามารถรับมือได้และปัญญาประดิษฐ์ยังเป็นเครื่องมือที่สามารถทำงานที่ซับซ้อนแทนมนุษย์ได้อย่างดีเยี่ยม ช่วยให้เราสามารถมีเวลาไปโฟกัสงานที่สำคัญและสามารถสร้างมูลค่าได้มากกว่า นอกจากนี้การประยุกต์ใช้ปัญญาประดิษฐ์ในระดับอุตสาหกรรม ยังช่วยลดต้นทุนและเพิ่มรายได้มหาศาล

2.2.1 ประวัติของปัญญาประดิษฐ์ หรือ Artificial Intelligence

สาขาปัญญาประดิษฐ์นั้นเริ่มก่อตั้งขึ้นในที่ประชุมวิชาการที่วิทยาลัยดาร์ตมัธ สหรัฐอเมริกาในช่วงหน้าร้อน ค.ศ. 1956 โดยผู้ร่วมในการประชุมครั้งนั้น ได้แก่ จอห์น แม็กคาร์ธีย์ มาร์วิน มินสกี อัลเลน นิวเวลล์ อาเธอร์ ซามูเอล และเฮอริเบิร์ต ไซมอน ที่ได้กลายมาเป็นผู้นำทางสาขาปัญญาประดิษฐ์ในอีกหลายสิบปีต่อมา นักวิทยาศาสตร์และนักศึกษาของพวกเขาเหล่านี้เขียนโปรแกรมที่หลายคนทิ้ง ไม้ว่าจะเป็น คอมพิวเตอร์ที่สามารถเอาชนะคนเล่นหมากรุก แก้ไขปัญหาเกี่ยวกับคำด้วยพีชคณิต พิสูจน์ทฤษฎีทางตรรกวิทยา หรือแม้กระทั่งพูดภาษาอังกฤษได้ ผู้ก่อตั้งสาขาปัญญาประดิษฐ์กลุ่มนี้เชื่อมั่นในอนาคตของเทคโนโลยีใหม่นี้มาก โดยเฮอริเบิร์ต ไซมอนคาดว่าจะมีเครื่องจักรที่สามารถทำงานทุกอย่างได้เหมือนมนุษย์ภายใน 20 ปีข้างหน้า และมาร์วิน มินสกีก็เห็นพ้องโดยการเขียนว่า "เพียงชั่วอายุคน ปัญหาของการสร้างความฉลาดเทียมจะถูกแก้ไขอย่างยั่งยืน"

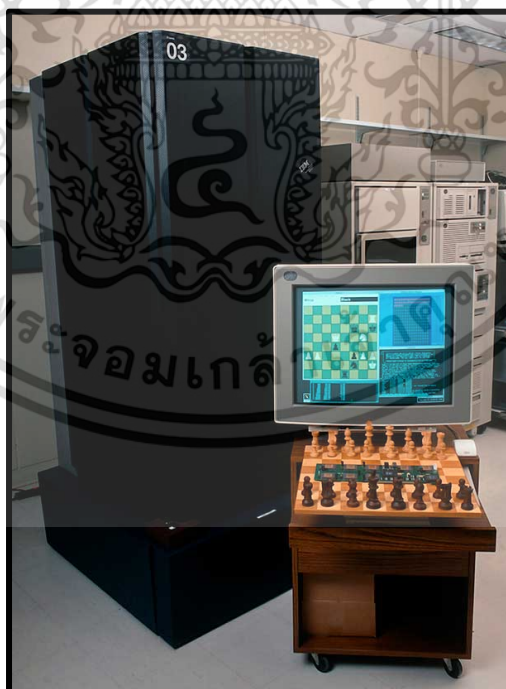
อย่างไรก็ตาม นักวิทยาศาสตร์กลุ่มนี้กลับไม่ได้พิจารณาถึงความยากของปัญหาที่จะพบมาก

นัก ในปี ค.ศ. 1974 เซอร์ เจมส์ โลทฮิลล์ ได้เขียนวิพากษ์วิจารณ์สาขาปัญญาประดิษฐ์ ประกอบกับมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรงกดดันจากสภาคองเกรสของสหรัฐฯให้ไปให้เงินสนับสนุนโครงการมีผลผลิตออกมาเป็นรูปธรรมมากกว่า ดังนั้นรัฐบาลสหรัฐอเมริกาและสหราชอาณาจักรจึงได้ตั้งงบประมาณการวิจัยที่ไร้ทิศทางของสาขาปัญญาประดิษฐ์ไป จนเป็นยุคที่เรียกว่า หน้าหนาวของปัญญาประดิษฐ์ (AI winter) กินเวลาหลายปีซึ่งโครงการด้านปัญญาประดิษฐ์แต่ละโครงการนั้นหาเงินทุนสนับสนุนยากมาก

ในช่วงต้นคริสต์ทศวรรษ 1980 งานวิจัยด้านปัญญาประดิษฐ์ประสบความสำเร็จในเชิงพาณิชย์เป็นครั้งแรกด้วยระบบที่ชื่อว่า “ระบบผู้เชี่ยวชาญ” อันเป็นระบบคอมพิวเตอร์ที่ช่วยในการหาคำตอบ อธิบายความไม่ชัดเจน ซึ่งปกตินั้นจะใช้ผู้เชี่ยวชาญในแต่ละสาขาตอบคำถามนั้น ในปี ค.ศ. 1985 ตลาดของปัญญาประดิษฐ์ทะยานขึ้นไปแตะระดับ 1 พันล้านดอลลาร์สหรัฐ ในขณะที่เดียวกันโครงการคอมพิวเตอร์รุ่นที่ 5 ของญี่ปุ่นก็ได้จุดประกายให้รัฐบาลสหรัฐอเมริกาและสหราชอาณาจักรหันมาให้เงินทุนสนับสนุนงานวิจัยในสาขาปัญญาประดิษฐ์อีกครั้ง

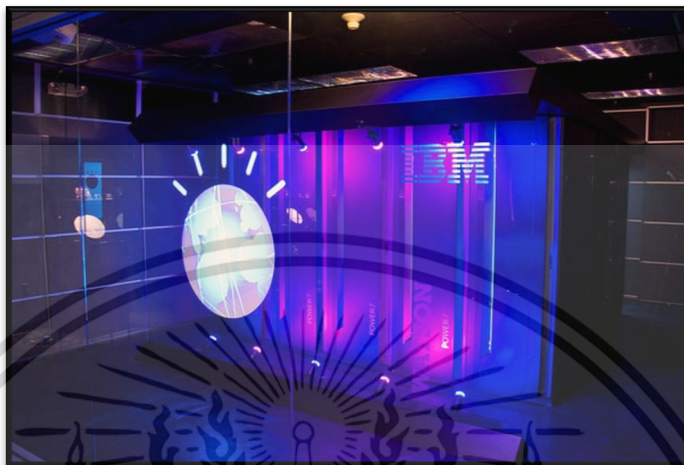
ในคริสต์ทศวรรษ 1990 และช่วงต้นคริสต์ศตวรรษที่ 21 ปัญญาประดิษฐ์ประสบความสำเร็จอย่างสูงแม้ว่าจะมีหลายอย่างที่อยู่เบื้องหลัง มีการนำปัญญาประดิษฐ์มาใช้ในด้านการขนส่ง การทำเหมืองข้อมูล การวินิจฉัยทางการแพทย์ และในอีกหลายสาขาหลายอุตสาหกรรม ความสำเร็จของปัญญาประดิษฐ์นั้นได้รับการผลักดันมาจากหลายปัจจัย ไม่ว่าจะเป็นเรื่องของความเร็วของคอมพิวเตอร์ที่มีการประมวลผลที่เร็วขึ้น (ตามกฎของมัวร์) การให้ความสำคัญกับการแก้ปัญหาย่อยบางปัญหา การสร้างความเชื่อมโยงระหว่างปัญญาประดิษฐ์กับสาขาอื่น ๆ ที่ทำงานอยู่กับปัญหาที่คล้าย ๆ กัน ตลอดจนความมุ่งมั่นของนักวิจัยที่ใช้วิธีการทางคณิตศาสตร์และวิทยาศาสตร์ที่มีหลักการ [12]



รูปที่ 2.5 เครื่องตีปบลูของบริษัทไอบีเอ็ม [12]

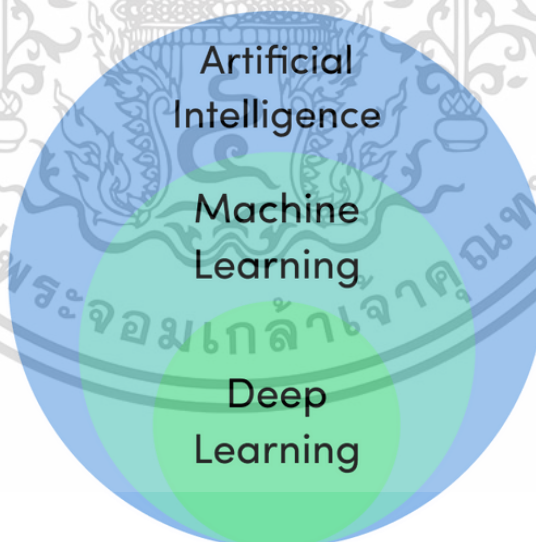
เมื่อวันที่ 11 พฤษภาคม ค.ศ. 1997 เครื่องตีปบลูของบริษัทไอบีเอ็ม กลายมาเป็นคอมพิวเตอร์เครื่องแรกของโลกที่สามารถเล่นหมากรุกเอาชนะ แกรี คาสปารอฟ แชมป์โลกในขณะนั้น เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการแข่งขันเท่านั้น เมื่ออนุญาตให้เผยแพร่ในอินเทอร์เน็ต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ และในเดือนกุมภาพันธ์ ค.ศ. 2011 เครื่องวัดสั่นของบริษัทไอบีเอ็มก็สามารถเอาชนะแชมป์รายการตอบคำถามจีโอพาร์ดีได้แบบขาดลอย นอกจากนี้ เครื่องเล่นเกมอย่าง Kinect ก็ใช้เทคโนโลยีของปัญญาประดิษฐ์มาใช้ในการสร้างส่วนติดต่อกับผู้ใช้ผ่านทาง การเคลื่อนไหวร่างกายใน 3 มิติ เช่นกัน [11]



รูปที่ 2.6 เครื่องวัดสั่นของบริษัทไอบีเอ็ม [13]

จากรูปที่ 2.7 จะเห็นได้ว่าปัญญาประดิษฐ์ (Artificial Intelligence) มีการเรียนรู้ของเครื่อง (Machine learning) เป็นซับเซตและการเรียนรู้เชิงลึก (Deep Learning) เป็นซับเซตของ Machine learning โดยมีรายละเอียดดังนี้



รูปที่ 2.7 ชนิดของปัญญาประดิษฐ์ [14]

2.2.2 การเรียนรู้ของเครื่อง (Machine learning)

การเรียนรู้ของเครื่อง (Machine learning) คือระบบที่สามารถเรียนรู้ได้จากตัวอย่างด้วยตนเองโดยปราศจากการป้อนคำสั่งของโปรแกรมเมอร์ ความก้าวหน้าในครั้งนี้นำมาพร้อมกับความคิด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ว่าเครื่องคอมพิวเตอร์สามารถเรียนรู้เพียงแค่ว่าจากข้อมูลอย่างเดียวเพื่อที่ให้นายผลลัพธ์ที่แม่นยำออกมาได้โดยการทำงานคือ Machine learning จะเรียนรู้ว่าข้อมูลขาเข้าและข้อมูลขาออก มีความสัมพันธ์กันอย่างไร และรู้อย่างไรจะเขียนกฎซึกกฎหนึ่งขึ้นมาอย่างไร โปรแกรมเมอร์ไม่จำเป็นต้องเขียนกฎใหม่ทุกครั้งที่ข้อมูลใหม่ อัลกอริทึมจะปรับเข้ากับข้อมูลใหม่เพื่อปรับปรุงประสิทธิภาพผลลัพธ์ในเวลาต่อมาดังรูปที่ 2.8



รูปที่ 2.8 รูปแบบการทำงานของ Machine learning [15]

การเรียนรู้ของเครื่อง (Machine learning) มีรูปแบบในการเรียนรู้เหมือนกับมนุษย์ คือ เรียนรู้จากประสบการณ์ ยิ่งรู้มากยิ่งง่ายต่อการพยากรณ์ว่าสิ่งต่อไปอะไรจะเกิดขึ้นโดยการเปรียบเทียบเมื่อพวกเขาประสบกับเหตุการณ์ที่ไม่เคยเจอมาก่อน มีความเป็นไปได้ที่ความสำเร็จจะลดลงกว่าเหตุการณ์ที่เคยเจอมาแล้ว machine สามารถถูกฝึก (train) ได้ในรูปแบบเดียวกันกับมนุษย์ เพื่อที่จะเพิ่มความแม่นยำในการพยากรณ์ [15]

2.2.2.1 รูปแบบการเรียนรู้ของ Machine Learning

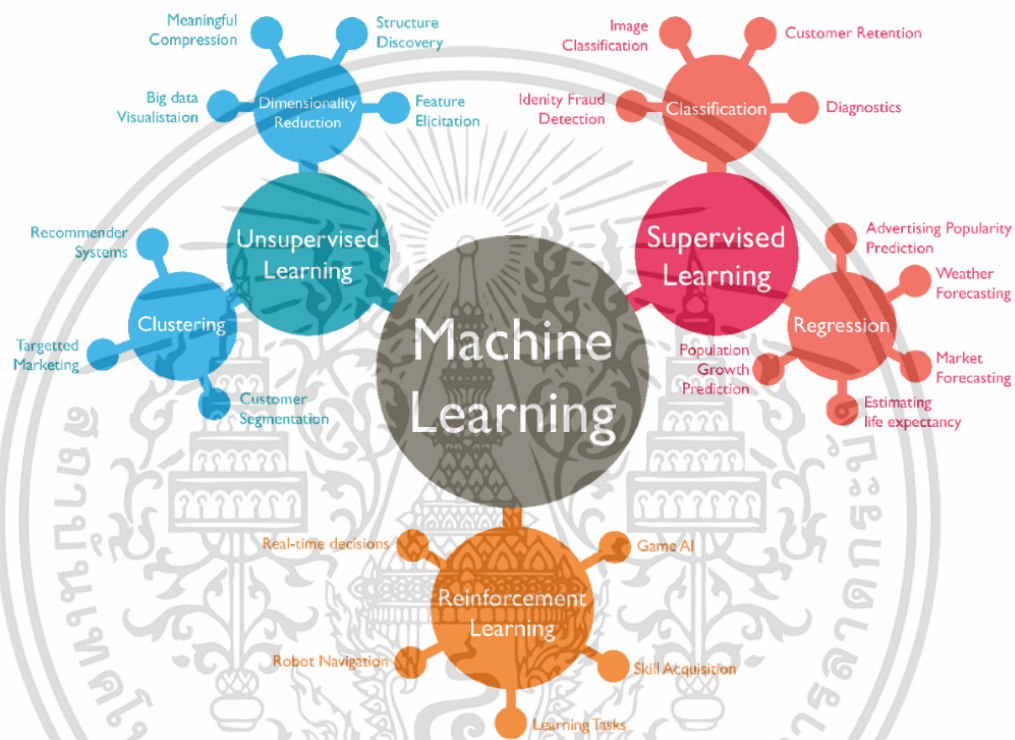
แบ่งเป็นได้ 3 แบบ ดังรูปที่ 2.9 ได้แก่

1. supervised machine learning คือ การเรียนรู้ที่ต้องมีข้อมูลสอนก่อน คอมพิวเตอร์จะสามารถหาคำตอบหรือผลลัพธ์ได้ด้วยตัวมันเอง หลังจากที่ได้เรียนรู้จากข้อมูลที่ใส่และสอนมันไปก่อนหน้า เมื่อมันรู้เรียนรู้แล้ว ในรอบต่อไปเมื่อใส่ข้อมูลลงไปมันก็จะบอกผลลัพธ์ได้ทันทีตามที่ฝึก(train) ซึ่งหากข้อมูลที่ฝึก (train) มันมีจำนวนมาก ในครั้งต่อไปมันก็จะยิ่งให้ผลลัพธ์ที่แม่นยำมากยิ่งขึ้นเช่นกัน การเรียนรู้แบบ Supervised Learning ตัวอย่างเช่น หากแบบจำลองของคุณกำลังพยายามทำนายว่าเพื่อนของคุณจะเล่นกอล์ฟหรือไม่คุณอาจมีตัวแปร เช่นอุณหภูมิ วันในสัปดาห์ เป็นต้น โดยจะมีตัวแปรที่มีค่าเป็น 1 ถ้าเพื่อนของคุณไปเล่นกอล์ฟจริง ๆ หรือ 0 ถ้าพวกเขาไม่ได้

2. Unsupervised Learning คือ การเรียนรู้ที่ไม่ต้องอาศัยข้อมูลสอน โดยใส่ข้อมูลลงไปให้คอมพิวเตอร์และไม่ได้กำหนดผลลัพธ์ว่าจะต้องเป็นแบบไหน ปล่อยให้คอมพิวเตอร์มันเรียนรู้ที่จะแยกกลุ่มได้ด้วยตัวมันเอง ซึ่งการเรียนรู้แบบ Unsupervised Learning ตัวอย่างเช่น เพื่อนของคุณเล่นกอล์ฟหรือไม่ขึ้นอยู่กับคอมพิวเตอร์เพื่อค้นหารูปแบบผ่านทางแบบจำลองเพื่อคาดเดาสิ่งที่เกิดขึ้นหรือทำนายว่าจะเกิดอะไรขึ้น [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Reinforcement learning จัดเป็นรูปแบบหนึ่งของ machine learning โดยใช้รูปแบบของการให้รางวัล (reward) และลงโทษ (punishments) จากการเรียนรู้แบบลองผิดลองถูก (trial and error) Google's DeepMind ก็ใช้ Reinforcement Learning ในการ train model ในการเล่นเกม ล้อมจมนชนะแชมป์โลกมาแล้ว นอกจากนี้ยังมีการนำ Reinforcement Learning มาใช้ในวงการเกมเพื่อ train ให้ bot มีความฉลาดมากยิ่งขึ้น โดยมีการใช้ algorithm ที่น่าสนใจได้แก่ Q-learning , Deep Q network , State-Action-Reward-State-Action (SARSA) และ Deep Deterministic Policy Gradient (DDPG)



รูปที่ 2.9 รูปแบบการเรียนรู้ของ Machine learning

2.2.2.2 ตัวอย่างการประยุกต์ใช้การเรียนรู้ของเครื่อง (Machine learning)

1. การเพิ่มเติมขึ้นมา (Augmentation) คือ Machine Learning ซึ่งช่วยเหลือมนุษย์ในงานต่าง ๆ ทุกวันนี้ทั้งเชิงบุคคลและเชิงพาณิชย์ก็ตามนั้นจะปราศจากการควบคุมโดยสมบูรณ์ของผลลัพธ์ที่เกิดขึ้นซึ่ง Machine Learning ถูกใช้ในหลากหลายรูปแบบ เช่น ผู้ช่วยเสมือน (Virtual Assistant), การวิเคราะห์ข้อมูล (Data Analysis), Software solution เป็นต้น โดยผู้ใช้งานแรก ๆ จะเป็นผู้ลดความผิดพลาดเนื่องจากความลำเอียงของมนุษย์เอง

2. การทำงานอัตโนมัติ (Automation) คือ Machine Learning ซึ่งทำงานทั้งหมดแต่อย่างอัตโนมัติในด้านต่าง ๆ โดยปราศจากการแทรกแซงของมนุษย์ ยกตัวอย่างเช่น หุ่นยนต์ที่ดำเนินการตามกระบวนการตามขั้นตอนในโรงงานการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อุตสาหกรรมการเงิน (Finance Industry) คือ Machine Learning กำลังเติบโตและเป็นที่ต้องการในอุตสาหกรรมการเงินอย่างมาก ธนาคารส่วนใหญ่ใช้ Machine Learning (ML) เพื่อหา pattern ของข้อมูลเพื่อป้องกันการฉ้อโกงจะเกิดขึ้น

4. องค์กรภาครัฐ (Government organization) คือ รัฐบาลใช้ ML เพื่อจัดการความปลอดภัยของสาธารณะและสาธารณูปโภค ยกตัวอย่างประเทศจีนมีการใช้ระบบการจดจำใบหน้า (face recognition) เป็นจำนวนมาก โดยรัฐบาลใช้ปัญญาประดิษฐ์ (Artificial Intelligence : AI) เพื่อป้องกันผู้ข้ามถนนอย่างไม่ระมัดระวัง

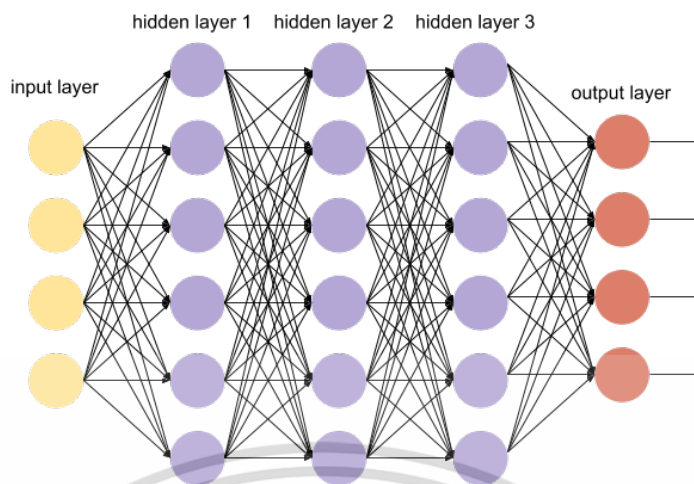
5. อุตสาหกรรมการดูแลสุขภาพ (Healthcare industry) คือ การดูแลสุขภาพเป็นหนึ่งในอุตสาหกรรมแรก ๆ ที่นำ Machine Learning กับการตรวจจับภาพ (image detection) เข้ามาใช้งาน

6. การตลาด (Marketing) คือ AI ถูกใช้อย่างกว้างขวางในด้านการตลาดโดยจะใช้เวลาเข้าถึงข้อมูลจำนวนมากในการทำงาน ก่อนที่จะถึงยุคที่มีข้อมูลมหาศาลขนาดนี้ นักวิจัยได้พัฒนาเครื่องมือด้านคณิตศาสตร์ขั้นสูงเหมือนกับการวิเคราะห์ของ Bayes (Bayesian analysis) เพื่อประเมินมูลค่าของลูกค้าแต่ละคน ฝ่ายการตลาดจะใช้ AI เป็นส่วนใหญ่เพื่อประสิทธิภาพการหาความสัมพันธ์ของลูกค้าและการโฆษณาด้านการตลาด

2.2.3 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึก (Deep Learning) คือ ซอร์ฟแวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท(neural network) ในสมองมนุษย์ถือเป็นซับเซ็ตของ machine learning อัลกอริทึมของ deep learning ถูกสร้างขึ้นจากการนำเอาระบบโครงข่ายประสาท (neural network) หลายๆชั้นมาต่อกัน โดยชั้น(layer) แรกสุดจะทำหน้าที่ในการรับข้อมูล(Input layer) ชั้นสุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา(Output layer) ส่วนชั้นที่อยู่ระหว่างชั้นแรกสุดและชั้นสุดท้ายจะถูกเรียกว่า Hidden layer หรือชั้นซ่อนดังรูปที่ 2.10

คำว่า deep learning หรือการเรียนรู้เชิงลึก มีที่มาจากการใช้ชั้นของ neural network หลายอันๆมาต่อกัน (มี Hidden layer มากกว่า 2 layer ก็ถือเป็น deep learning แล้ว) เนื่องจากชั้นเหล่านี้เป็นโครงสร้างที่ถูกจัดเก็บแบบเป็นกองซ้อน (stack) จึงเปรียบได้ว่าชั้นที่จำนวนเยอะๆ ทำให้มีโครงสร้างที่ลึกยิ่งขึ้น [16]



รูปที่ 2.10 รูปแบบโครงสร้างระบบโครงข่ายประสาทของการเรียนรู้เชิงลึก (Deep Learning) [16]

โดย Hidden layer ของแต่ละชั้นจะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท(neural) จำนวนมากซึ่งมีหน้าที่ในการประมวลผลรับข้อมูลจากชั้นที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยังชั้นที่อยู่ล่างกว่า ข้อดีของการส่งข้อมูลแบบนี้ก็คือชั้นแต่ละชั้นสามารถที่จะมีค่าถ่วงน้ำหนัก (weight), ค่าความเอนเอียงของข้อมูล (bias) และวิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกันได้ ถ้าหากป้อนข้อมูลให้กับโมเดลมากเท่าไรจะส่งผลให้แต่ละชั้นสามารถสกัด feature ที่มีความซับซ้อนมากยิ่งขึ้นซึ่งจะส่งผลโดยตรงกับความแม่นยำของการพยากรณ์

2.2.3.1 การเรียนรู้ของโมเดล นั้นเกิดขึ้นจาก 2 เฟส ประกอบด้วย

เฟสแรก คือ การประยุกต์ใช้ การแปลงแบบไม่เป็นเชิงเส้น (nonlinear transformation) กับข้อมูลที่ได้รับ (input) และได้ผลลัพธ์ (output) ออกมาอยู่ในรูปของโมเดลทางสถิติ (statistical model)

เฟสที่สอง คือ การนำโมเดลมาผ่านวิธีการทางคณิตศาสตร์ อย่างเช่นการใช้อนุพันธ์ (derivative) โดยทั้ง 2 เฟสนี้จะถูกทำซ้ำแล้วซ้ำเล่าจนกว่าจะได้มาซึ่งโมเดลที่มีความแม่นยำในระดับที่น่าพึงพอใจ ซึ่งการทำซ้ำๆ ของ 2 เฟสนี้มีชื่อว่าการทำซ้ำ (iteration)

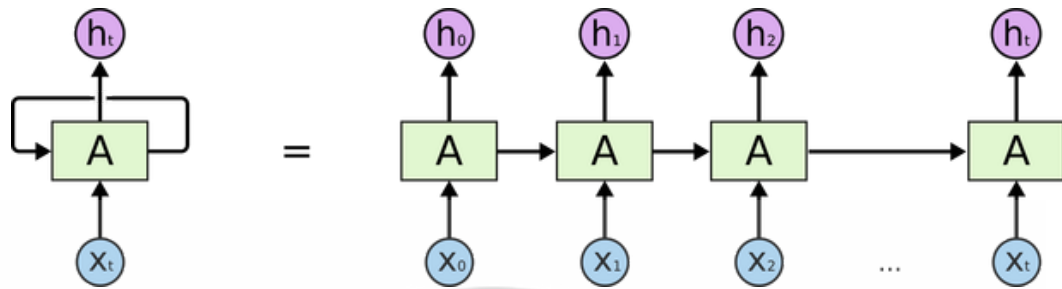
2.2.3.2 ชนิดของโครงข่ายการเรียนรู้เชิงลึก (Deep Learning) แบ่งออกเป็น 2 ส่วน ได้แก่

1. โครงข่ายประสาทแบบป้อนไปหน้า (Feed-forward neural networks) ถือเป็นโมเดลที่มีโครงสร้างที่เรียบง่ายที่สุด เพราะว่าการดำเนินการของข้อมูลจะเป็นไปในทิศทางเดียว ก็คือรับข้อมูลจาก input layer แล้วส่งต่อไปยัง hidden layer เรื่อยๆ จนกระทั่งถึง output layer ก็จะหยุด

2. วนซ้ำวนซ้ำ (Recurrent neural networks : RNN) คือ โครงข่ายประสาท (neural networks) หลายชั้นที่สามารถเก็บข้อมูลไว้ที่ node จึงทำให้มันสามารถรับข้อมูลเป็นแบบลำดับ (data sequences) และให้ผลลัพธ์ออกเป็นลำดับของข้อมูลได้ อธิบายอย่างง่ายๆ RNN ก็คือ

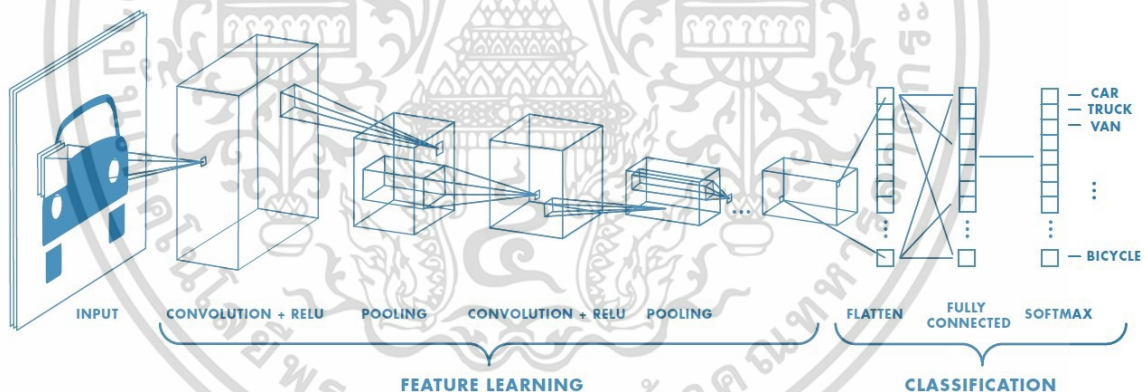
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาท (neural networks) เชื่อมต่อกันหลายๆอันและยังสามารถต่อกันเป็นวงวนรูปได้นั้นเอง เพราะฉะนั้น RNN จึงเหมาะสมในการประมวลผลข้อมูลที่เป็นลำดับอย่างมากตัวอย่างดังรูปที่ 2.11 [11]



รูปที่ 2.11 ตัวอย่างโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent neural networks : RNN) [17]

2.2.3.3 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks :CNN) คือ neural network หลายชั้นที่มีโครงสร้างเฉพาะตัว โดยถูกออกแบบมาเพื่อการเพิ่มความสามารถในการสกัดเอา feature ที่มีความซับซ้อนมากยิ่งขึ้นจากข้อมูล โดยโครงข่ายประสาทแบบคอนโวลูชันนั้นตอบโจทย์ปัญหาประเภทการรับรู้ (perceptual tasks) อย่างมาก มักจะถูกใช้เพื่อการสกัด feature จากข้อมูลประเภทที่ไม่ค่อยเป็นระเบียบหรือไม่ได้มีโครงสร้างเป็นรูปแบบเฉพาะตัว (unstructured data) อย่างเช่น รูปภาพ เป็นต้น



รูปที่ 2.12 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural networks : CNN)

2.2.3.4 ตัวอย่างการประยุกต์ใช้การเรียนรู้เชิงลึก (Deep Learning)

1. บทบาทของ AI ในการเงิน (finance) : ปัจจุบันเทคโนโลยีการเงินได้มีการนำเอา AI มาเข้ามาใช้เพื่อประหยัดเวลา ลดต้นทุนและเพิ่มมูลค่า การเรียนรู้เชิงลึก (Deep Learning) ได้ทำการปฏิวัติระบบการกู้ยืมเงินแบบดั้งเดิมโดยการให้ AI เป็นผู้ประมวลผลถึงเครดิตของผู้กู้ยืม

2. บทบาทของ AI ในการจัดการทรัพยากรมนุษย์ (Human result) : Under Armour คือบริษัทขายอุปกรณ์การกีฬาได้ปฏิวัติวิธีการรับคนเข้าทำงาน นอกจากนี้ Under Armour ยังลดเวลาจ้างงานในการทำสต็อกสินค้า(retail stores) ได้มากถึง 35 เปอร์เซ็นต์ , Under Armour เติบโตและเป็นที่รู้จักอย่างมากในช่วงปี 2012 ถึงขนาดที่ว่ามีโบสมักรเข้ามาถึง 30000 ใบต่อเดือน การที่จะไล่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ่านใบสมัครทั้ง 30000 ใบทุกเดือนจะเป็นการเสียเวลาและทรัพยากรด้านบุคคลมากเกินไป Under Armour จึงหันมาสนใจ AI ที่ช่วยงานด้านทรัพยากรมนุษย์ โดยเลือกที่จะใช้ "HireVue" ซึ่งเป็น AI ที่ตอบโจทย์ปัญหาด้านทรัพยากรมนุษย์อย่างมาก โดยการคัดกรองผู้สมัครที่มีคุณสมบัติตามความต้องการและทำการสัมภาษณ์ (live interview)

3. บทบาทของ AI ในการตลาด : AI ถือเป็นเครื่องมือที่มีประโยชน์อย่างมากในการจัดการเรื่องของการให้บริการลูกค้าเป็นรายบุคคล การพัฒนาของ speech recognition ใน call-center management และติดต่อไปยังฝ่ายที่เกี่ยวข้องกับสิ่งที่ลูกค้าต้องการ ก็ถือเป็น การตอบโจทย์ลูกค้าอย่างหนึ่ง

จาก 3 ตัวอย่างข้างต้น จะเห็นได้ว่า deep learning ถูกนำมาประยุกต์ใช้ในหลากหลายวงการ ตั้งแต่การเงิน (finance) ไปถึงการตลาด (marketing) ห่วงโซ่อุปทาน (supply chain) และบริษัทยักษ์ใหญ่จะเป็นบริษัทแนวหน้าที่สามารถใช้ deep learning ได้อย่างมีประสิทธิภาพเพราะมีข้อมูลมากพอที่จะใช้ในการ train model นั้นเอง

2.2.3.5 ข้อจำกัดของการเรียนรู้เชิงลึก (Deep Learning)

1. ต้องการระบุชื่อหรือจัดประเภทข้อมูล (Data labeling) ปัจจุบัน AI ส่วนมากอาศัยการ train โดยใช้หลักการของ supervise learning ซึ่งนั่นก็หมายความว่า มนุษย์จะต้องตั้งชื่อและจัดประเภทของข้อมูล (data labeling) ด้วยตัวเองก่อนจะนำไป train ซึ่งข้อมูลเหล่านี้ก็มีจำนวนมากจึงไม่แปลกที่อาจเกิดความผิดขึ้นได้ ตัวอย่างเช่น รถยนต์ขับเคลื่อนอัตโนมัติ (self-driving car) จำคนหลายร้อยคนเพื่อที่จะ labeling วิดีโอที่จะใช้ในการ train ระบบ

2. ต้องการชุดข้อมูลจำนวนมาก (Obtain huge training datasets) การเรียนรู้เชิงลึกนั้นต้องการข้อมูลจำนวนมากในการ train ข้อมูล ยกตัวเช่น ต้องการข้อมูลจำนวน 1000 ตัวอย่าง เพื่อให้ได้โมเดล classification ที่มีประสิทธิภาพและในบางกรณีก็ต้องการมากกว่า 1 ล้านตัวอย่างเพื่อที่จะทำให้ model ทำงานได้อย่างมีประสิทธิภาพใกล้เคียงกับมนุษย์ แน่แน่นอนว่าการเรียนรู้เชิงลึกนั้นเป็นที่นิยมในบริษัทเทคโนโลยีชั้นนำทั่วโลก ซึ่งพวกเขาใช้ big data ที่สะสมข้อมูลที่มีเนื้อที่ไม่น้อยกว่า petabytes (1 ล้าน gigabytes) นั่นเป็นเหตุผลว่าทำไม บริษัทเทคโนโลยีชั้นนำจึงสามารถสร้าง deep learning model ที่มีประสิทธิภาพและมีความแม่นยำสูง

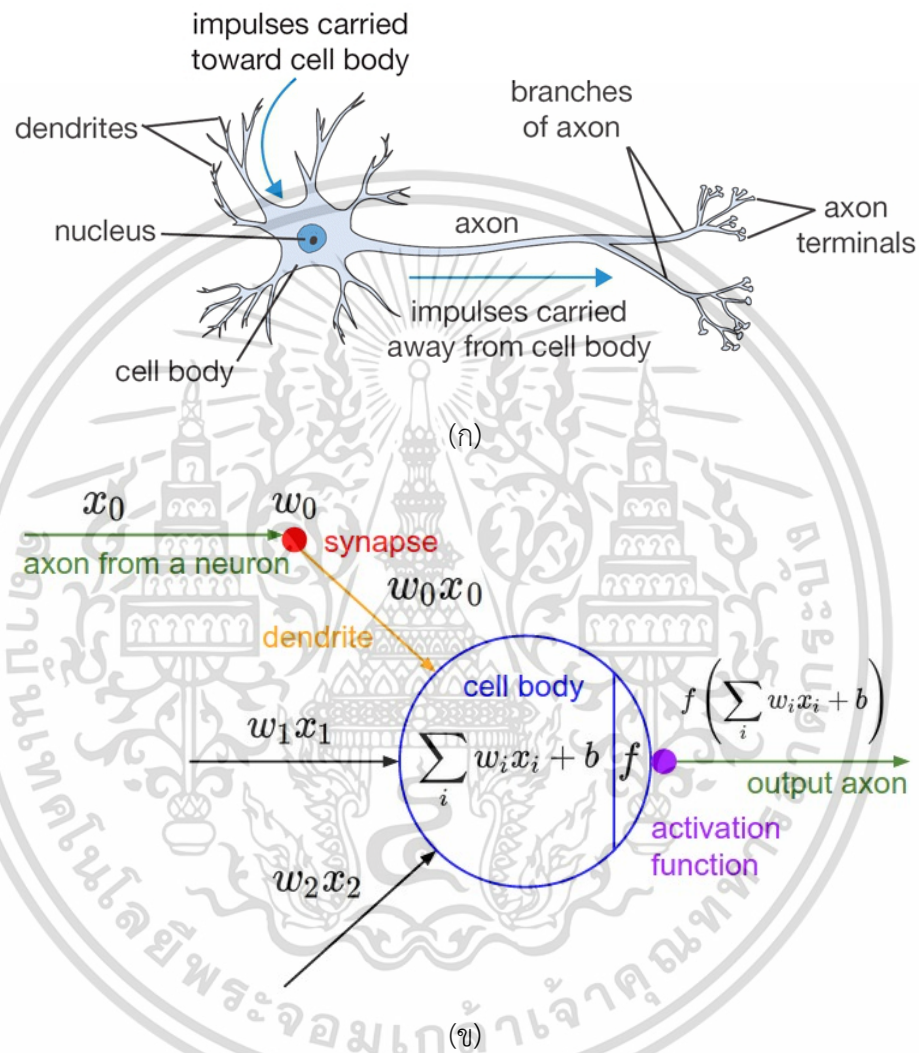
3. ปัญหาในการอธิบาย (Explain a problem) ไม่ใช่เรื่องง่ายที่มนุษย์เราจะอธิบาย AI model ที่ซับซ้อนได้ยกตัวอย่างเช่น คำถามว่าทำไม model ถึงเกิดการตัดสินใจแบบนั้น นี่ก็เป็นอีกเหตุผลหนึ่งที่ทำไม AI ถึงได้รับการยอมรับช้านักก็เพราะบางอย่างมันยากที่มนุษย์เราจะสามารถอธิบายนั่นเอง

2.3 ฟังก์ชันการกระตุ้น (Activation function)

ในสมองของมนุษย์ประกอบด้วยหน่วยเล็ก ๆ เรียกว่า นิวรอน (Neuron) จำนวนประมาณ 8 หมื่น 6 พันล้านนิวรอน โดยแต่ละนิวรอนจะเชื่อมต่อกันด้วยเส้นประสาทเรียกว่า ไสแวนป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Synapse) รวมแล้วประมาณ 1 พันล้านล้านไซแนปส์ ซึ่งนักวิทยาศาสตร์คอมพิวเตอร์ได้นำมาเป็นแนวคิดในการออกแบบโครงข่ายประสาทเทียม (Artificial Neural Network) แต่ละนิวรอนจะได้รับอินพุตหลายๆอันจากหลายกิ่งก้านสาขาของเดน-ไดท์ (Dendrite) แล้วประมวลผลออกมาเป็นเอาต์พุต 1 เอาต์พุตออกไปที่แอกซอน(Axon) เพื่อส่งต่อไปให้เดนไดท์ของนิวรอนอื่นๆ ใช้เป็นอินพุตต่อไป [19] ดังรูป 2.13



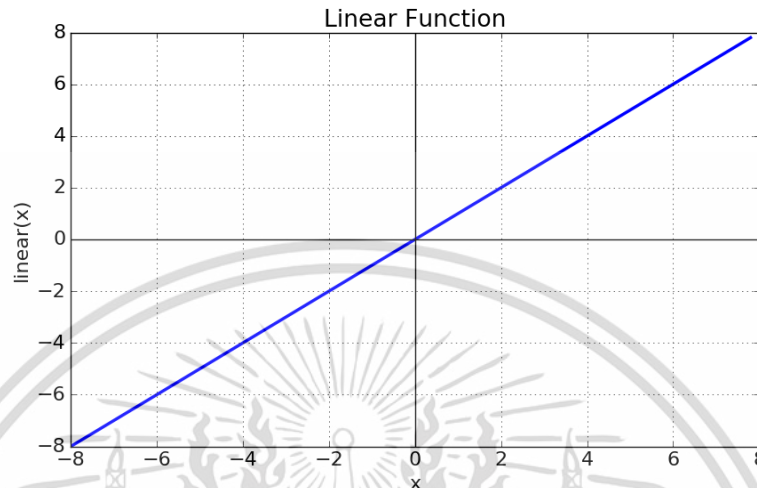
รูปที่ 2.13 (ก) คือตัวอย่างโครงข่ายประสาทของมนุษย์, (ข) คือแบบจำลองทางคณิตศาสตร์ของโครงข่ายประสาท-เทียม [19]

ดังนั้นจึงกล่าวได้ว่าฟังก์ชันการกระตุ้น (Activation function) คือฟังก์ชันที่รับผลรวมการประมวลผลทั้งหมดจากทุกอินพุตภายใน 1 นิวรอน แล้วส่งต่อเป็นเอาต์พุตของโนดหรือรู้จักอีกชื่อว่าฟังก์ชันการถ่ายโอน (Transfer function) โดยทั่วไปแล้วฟังก์ชันการกระตุ้นสามารถแบ่งออกเป็น 2 ประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ฟังก์ชันการกระตุ้นแบบเชิงเส้น (Linear activation function)

ในปัจจุบันไม่ค่อยเป็นที่นิยมเนื่องจากเอาต์พุตของฟังก์ชันจะไม่ถูกจำกัดไว้ระหว่างช่วงใดๆ และไม่ได้ช่วยลดความซับซ้อนหรือพารามิเตอร์ต่างๆ ของข้อมูลปกติที่เป็นอินพุตให้กับโครงข่ายประสาทเทียมและสามารถสังเกตเอาต์พุต ดังรูปที่ 2.14



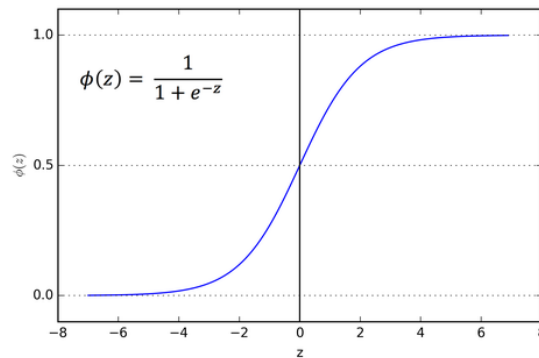
รูปที่ 2.14 กราฟพล็อตเอาต์พุตของข้อมูลโดยใช้ฟังก์ชันการกระตุ้นเชิงเส้น (Linear activation function)

2.3.2 ฟังก์ชันการกระตุ้นที่ไม่ใช่เชิงเส้น (Non-Linear activation function)

ฟังก์ชันการกระตุ้นแบบไม่ใช่เชิงเส้น (Non-Linear activation function) ซึ่งเป็นฟังก์ชันการกระตุ้น มีการใช้งานมากที่สุดในปัจจุบัน ความไม่เป็นเชิงเส้นช่วยให้โมเดลสามารถปรับให้เข้ากับข้อมูลอินพุตที่มีความหลากหลายได้ง่ายเพื่อแยกความแตกต่างระหว่างผลลัพธ์ฟังก์ชันการกระตุ้นแบบไม่ใช่เชิงเส้นส่วนใหญ่แบ่งตามช่วงหรือเส้นโค้งสามารถแบ่งออกได้ ดังนี้

2.3.2.1 ฟังก์ชันซิกมอยด์ (Sigmoid function)

เส้นโค้งฟังก์ชันซิกมอยด์มีลักษณะเป็นรูปตัวเอส ดังรูปที่ 2.15 สาเหตุที่นิยมในปัจจุบัน เนื่องจากมีค่าเอาต์พุตอยู่ในช่วง 0 ถึง 1 เหมาะสำหรับการพยากรณ์ความน่าจะเป็นที่มีผลลัพธ์อยู่ในระหว่างช่วง 0 และ 1 เท่านั้น ข้อเสียของฟังก์ชันซิกมอยด์ คืออาจจะส่งผลให้โครงข่ายติดขัดได้ขณะเวลาที่ฝึกโมเดล สามารถใช้ฟังก์ชันซอร์ฟแมก (Softmax function) ในกรณีที่เราต้องการทำการจำแนกแบบหลายคลาส (Multiclass classification) ซึ่งจะทำการอธิบายในขั้นตอนต่อไป

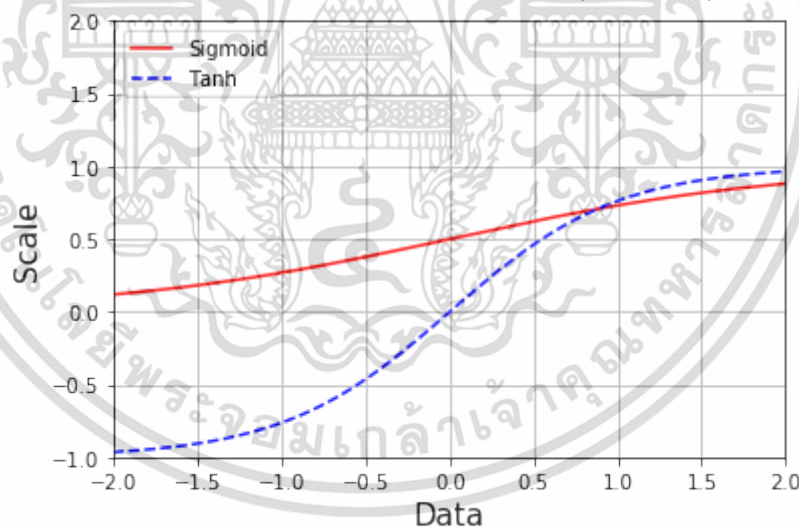


รูปที่ 2.15 กราฟของฟังก์ชันซิกมอยด์ (Sigmoid)

2.3.2.2 ฟังก์ชันแทน (Tanh function)

ฟังก์ชันแทนมีลักษณะของกราฟคล้ายกับฟังก์ชันซิกมอยด์ แต่ช่วงเอาต์พุตของฟังก์ชันแทนมีค่าระหว่าง -1 ถึง 1 และฟังก์ชันแทนยังเป็นรูป Sigmoidal (รูปตัว s) ดังรูปที่ 2.16 จุดเด่นของฟังก์ชันนี้คือถ้าค่าที่ได้ออกมาเป็นค่าติดลบก็จะมีแนวโน้มไปแมพ (maps) กับค่าที่เป็นลบสูง โดยถ้าค่าเป็น 0 ก็จะมีแนวโน้มไปแมพกับค่าใกล้ๆ 0 ในกราฟฟังก์ชันแทนส่วนใหญ่ใช้การจำแนกระหว่างสองคลาส สามารถคำนวณได้ดังสมการ (2.1)

$$\text{Tanh}(z) = \frac{(e^z - e^{-z})}{(e^z + e^{-z})} \quad (2.1)$$



รูปที่ 2.16 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชันซิกมอยด์และฟังก์ชันแทน

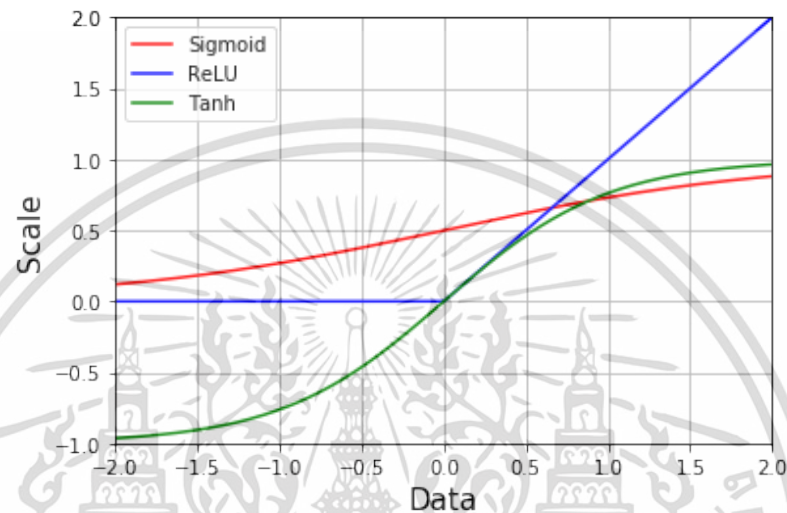
2.3.2.3 ฟังก์ชัน ReLU (Rectified Linear Unit)

ฟังก์ชัน ReLU ย่อมาจาก Rectified Linear Unit เป็นฟังก์ชันเส้นตรงที่ถูกปรับแก้ (Rectified) ไม่ได้เป็นรูปตัว S เหมือน 2 ฟังก์ชันก่อนหน้าดังในรูปที่ 2.17 ReLU เป็นฟังก์ชันที่เรียบง่ายกว่าทุก Activation Function ที่ผ่านมาเนื่องจากถ้าอินพุตเป็นบวกความชันจะมีค่าเท่ากับ 1 ตลอดกาลทำให้ไม่เกิด Vanishing Gradient ส่งผลให้โมเดลทำงานได้เร็วขึ้นมาก สามารถคำนวณ

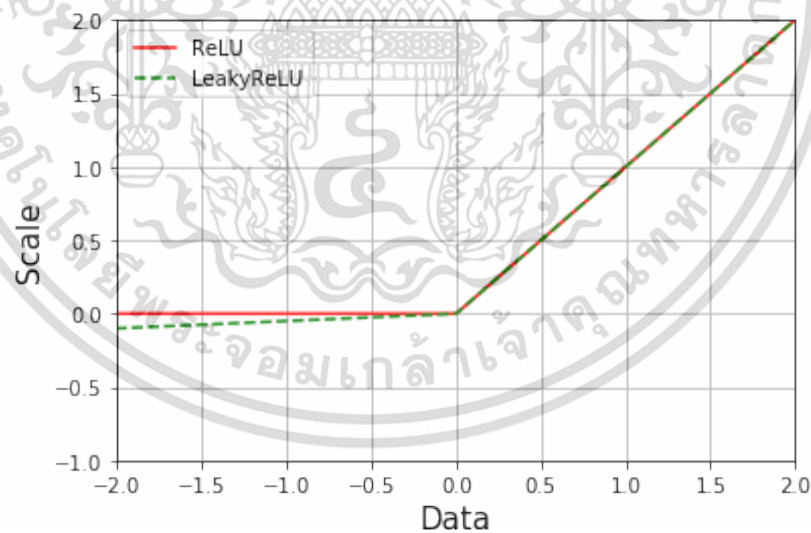
ได้ดังสมการ (2.2)
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} \quad (2.2)$$

ปัญหาของการแปลงค่าลบเป็น 0 ทั้งหมดนั้นจะลดความสามารถในการฝึกของโมเดลจากข้อมูล ซึ่งไม่ว่าค่าจะเป็นค่าเท่าใดก็ตามหากติดลบแล้วเมื่อเข้าสู่ฟังก์ชัน ReLU ก็จะกลายเป็น 0 ทันที ทำให้ส่งผลให้ไม่สามารถแมปค่าที่ติดลบได้



รูปที่ 2.17 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชันซิกมอยด์, ฟังก์ชันแทนและฟังก์ชัน ReLU



รูปที่ 2.18 กราฟเปรียบเทียบเอาต์พุตระหว่างฟังก์ชัน ReLU และฟังก์ชันการรั่วของ ReLU

2.3.2.4 ฟังก์ชันการรั่วของ ReLU (Leaky ReLU function)

ฟังก์ชันการรั่วของ ReLU คือวิธีหนึ่งที่ใช้ในการแก้ไขปัญหาที่เกิดขึ้นจากฟังก์ชัน ReLU แทนที่จะให้ฟังก์ชันเป็นศูนย์เมื่อ x น้อยกว่า 0 แต่ฟังก์ชันการรั่วของ ReLU จะมีความชันลบเล็กน้อยแทน(เท่ากับ 0.01 หรือมากกว่านั้น) ดังรูปที่ 2.18 นั่นคือถ้าอินพุตติดลบก็ให้เป็นค่าที่ติดลบน้อยๆ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถแปลงค่ากลับได้แต่ในปัจจุบันก็ยังไม่ค่อยได้รับความนิยมเท่าที่ควร สามารถคำนวณฟังก์ชันได้ดังสมการ (2.3)

$$f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x) \quad (2.3)$$

โดยที่ α คือ ค่าคงที่ขนาดเล็ก

2.3.2.5 ฟังก์ชัน Softmax

ฟังก์ชัน Softmax หรือที่เรียกว่า Softargmax เป็นฟังก์ชันทางคณิตศาสตร์ที่ใช้ในการแปลงค่าน้ำหนักจากค่าหนึ่งไปเป็นอีกค่าหนึ่งด้วยฟังก์ชันที่เรากำหนด โดยเอาต์พุตถูกรวมด้วยฟังก์ชันเอ็กซ์โปเนนเชียล (e ยกกำลัง x) ซึ่งเอาต์พุตคือความน่าจะเป็นที่ผลรวมเท่ากับ 1 หรือจะให้เข้าใจได้ง่ายขึ้นก็คือ ฟังก์ชัน Softmax รับค่าอินพุตที่เป็นตัวเลขเข้าไปแล้วแปลงเอาต์พุตออกมาในรูปแบบของความน่าจะเป็น ดังสมการที่ (2.4)

$$\text{Softmax}(z) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (2.4)$$

โดยที่ $i = 1, \dots, K$ และ $\mathbf{z} = (z_1, \dots, z_k) \in \mathbb{R}^k$

ฟังก์ชัน Softmax มักถูกนำไปไว้ขั้นสุดท้ายของของโครงข่ายประสาทเทียม เพื่อให้เอาต์พุตออกมาในรูปแบบของความน่าจะเป็นเพื่อไปคำนวณ Negative Log Likelihood สำหรับหาค่าความสูญเสีย โดยใช้เทคนิค Cross Entropy ฟังก์ชัน Softmax เป็นที่นิยมในงานทางด้านการแยกแยะ (Classification) มีชื่อเรียกว่า Softmax Classifier หรือ Softmax Loss [18]

2.4 การเคลื่อนลงตามความชัน (Gradient Descent)

การเคลื่อนลงตามความชัน หรือ Gradient Descent เป็นกลยุทธ์การเพิ่มประสิทธิภาพที่ได้รับความนิยมมากที่สุดที่ใช้ในแมชชีนเลิร์นนิงและการเรียนรู้เชิงลึกในปัจจุบันเนื่องจากสามารถใช้ร่วมกับทุกอัลกอริทึมและง่ายต่อการเข้าใจและนำไปใช้ โดยในหัวข้อนี้อธิบายถึงวิธีการทำงานของการเคลื่อนลงตามความชันตามประเภทของการใช้งาน การเคลื่อนลงตามความชันเป็นอัลกอริทึมที่ใช้ในการลดฟังก์ชันฟังก์ชันความผิดพลาด (Error Function) โดยการจะวัดการเปลี่ยนแปลงของน้ำหนักทั้งหมดโดยพิจารณาการเปลี่ยนแปลงของฟังก์ชันความผิดพลาด (Error Function) เพื่อให้ได้ค่าความผิดพลาดที่น้อยที่สุดในการอัปเดตน้ำหนักและพารามิเตอร์อื่นๆ ในชั้นต่างๆของโครงข่ายประสาทเทียมเพื่อให้ค่าความผิดพลาดที่ได้ (Error) ลดลงไปสู่จุดที่ต่ำสุดในค่าน้ำหนักที่อัปเดตในครั้งต่อไป โดยที่การเคลื่อนลงตามความชันดังรูปที่ 2.19 สามารถคำนวณได้ดังสมการ (2.5)

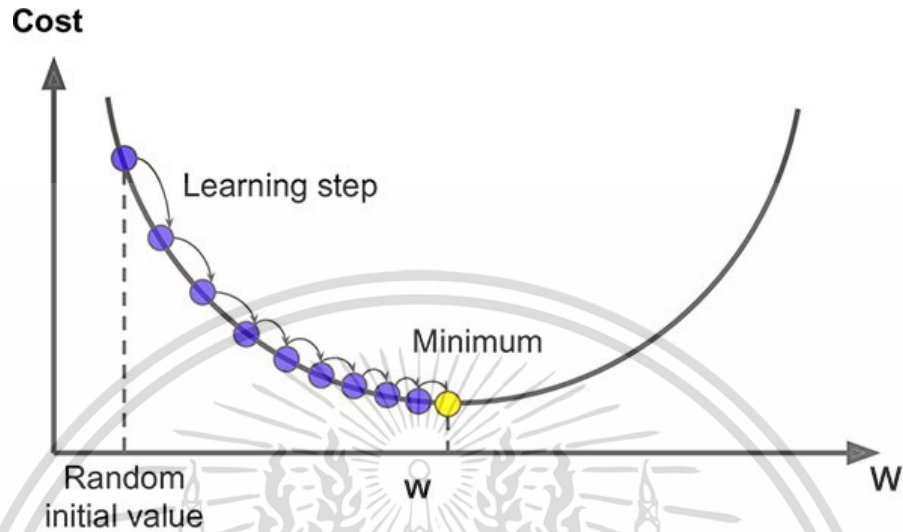
$$w_p^q = w_p^q - \alpha \frac{d\text{Error}}{dw_p^q} \quad (2.5)$$

โดยที่ w_p^q คือค่าน้ำหนักที่ทำการอัปเดต

α คือค่าอัตราการเรียนรู้ (Learning rate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

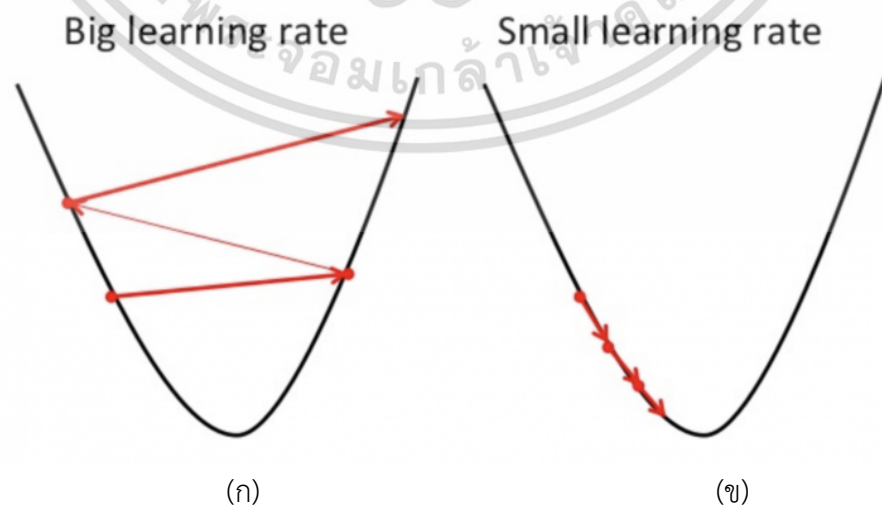
$\frac{dError}{dw_p^q}$ คือการหาอนุพันธ์ย่อย(Partial derivative)ของค่าความผิดพลาดโดยใช้ Error function



รูปที่ 2.19 รูปแบบการอัปเดตค่าน้ำหนักโดยใช้เทคนิคการเคลื่อนลงตามความชัน

2.4.1 อัตราการเรียนรู้ (Learning rate)

อัตราการเรียนรู้ (Learning rate) คือ ขนาดของอัตราการเรียนรู้เพื่อหาค่าความผิดพลาดต่ำสุดของโลคอล(Local) ในการกำหนดอัตราการเรียนรู้จำเป็นต้องเป็นค่าที่เหมาะสมซึ่งไม่สูงหรือต่ำจนเกินไป เพราะถ้าหากมีค่าสูงเกินไปอาจจะทำให้ไม่ถึงค่าต่ำสุดในพื้นที่เนื่องจากมันตีกลับไปมา ระหว่างฟังก์ชันนูน (Convex function) ของการเคลื่อนลงตามความชันดังรูปที่ 2.20 (ก) แต่ถ้าหากอัตราการเรียนรู้ที่น้อยมากของการเคลื่อนลงตามความชันค่าความผิดพลาดจะถึงค่าต่ำสุดแต่อาจใช้เวลาในการคำนวณ ดังรูปที่ 2.20 (ข)



รูปที่ 2.20 (ก) คืออัตราการเรียนรู้ที่มีค่าที่สูงเกินไป, (ข) คืออัตราการเรียนรู้ที่มีค่าน้อย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ฟังก์ชันความผิดพลาด (Error Function)

ฟังก์ชันความผิดพลาด (Error Function) คือฟังก์ชันคณิตศาสตร์ที่เป็นตัวชี้ความแม่นยำในการทำนายของโมเดล โดยเปรียบเทียบระหว่างเอาต์พุตที่เป็นค่าจริงและเอาต์พุตที่ได้จากการทำนายของโมเดลเพื่อทำการหาอนุพันธ์หรือความชันสำหรับการอัปเดตค่าน้ำหนักและพารามิเตอร์อื่นๆในโมเดล

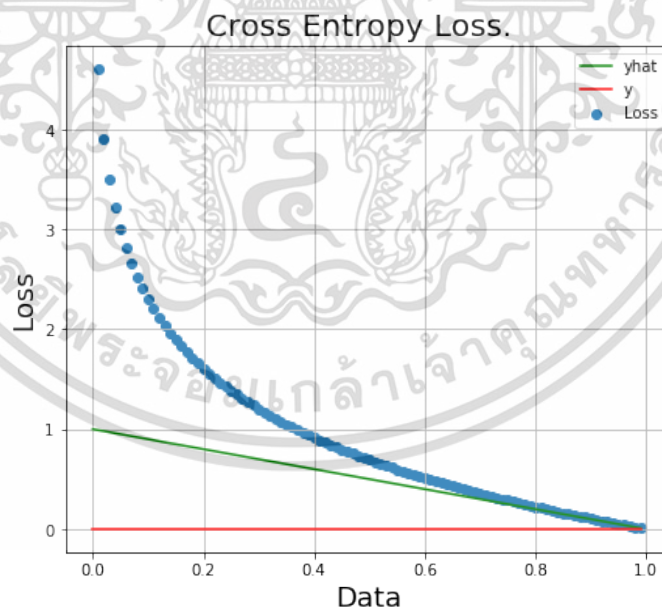
2.4.2.1 การคำนวณความผิดพลาดโดยใช้ Cross Entropy

การคำนวณความผิดพลาดโดยใช้ Cross Entropy หรือ Cross Entropy Loss คือการคำนวณความผิดพลาดของค่าที่พยากรณ์ว่าแตกต่างจากค่าจริงมากน้อยเพียงใดเพื่อทำการอัปเดตค่าน้ำหนักในกระบวนการเกรเดียนต์เดสเซนส์ (Gradient descent) ซึ่งทำการคำนวณโดยใช้ความน่าจะเป็น (Probability) คือ ถ้าหากทายถูกแต่ไม่มั่นใจหรือถ้าหากทายผิดแต่มั่นใจมาก ก็จะมีผิดพลาดมากโดยคำนวณทั้งแบบ (Batch) แล้วทำการหาค่าเฉลี่ยและผลลัพธ์ที่ได้ทุกคลาสต้องรวมกันได้เท่ากับ 1 โดยที่การคำนวณในรูปแบบดังที่กล่าวมาเหมาะสำหรับงานทางด้านแยกแยะ (Classification model) ดังสมการที่ (2.6)

$$J(w) = -\frac{1}{N} \sum_{n=1}^N [Y_n \log(\hat{Y}_n)] \quad (2.6)$$

โดยที่ Y_n คือค่าเอาต์พุตที่เป็นค่าจริง

\hat{Y}_n คือค่าเอาต์พุตที่เป็นค่าพยากรณ์

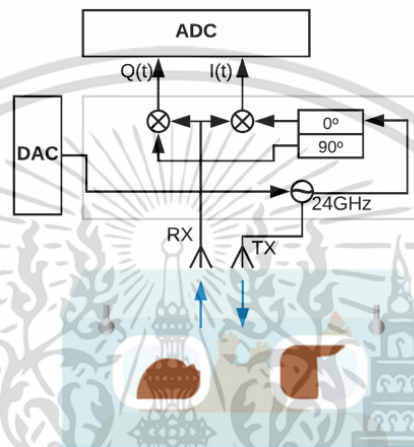


รูปที่ 2.21 กราฟพล็อตแสดงความสัมพันธ์ระหว่างค่าความคลาดเคลื่อน (Loss), ค่าเอาต์พุตที่เป็นค่าจริง (y) และค่าที่ได้จากการพยากรณ์ (yhat)

จากรูปที่ 2.21 กราฟพล็อตแสดงความสัมพันธ์ระหว่างค่าความคลาดเคลื่อน (Loss), ค่าเอาต์พุตที่เป็นค่าจริง (y) และค่าที่ได้จากการพยากรณ์ (yhat) โดยจำลองข้อมูลตัวอย่างที่เป็นค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จริง (y) มีค่าเท่ากับ 0 (เส้นสีแดง) เทียบกับค่าที่ได้จากการพยากรณ์ไล่ตั้งแต่ 1 ไปถึง 0 โดยทยอยผิดไล่ไปถึงทยอยถูก (เส้นสีเขียว) พบว่าค่าความคลาดเคลื่อน(เส้นสีฟ้า) เริ่มจากซ้ายสุด Ground Truth มีค่าเท่ากับเท่ากับ 0 (เส้นสีแดง) เนื่องจากโมเดลทยอยผิดคือทยอยว่าเป็น 1 (เส้นสีเขียว) ด้วยความมั่นใจทำให้ความคลาดเคลื่อนพุ่งขึ้นถึงอินฟินิตี้ ส่วนถัดมาคือช่วงตรงกลางความคลาดเคลื่อนจะลดลงอย่างรวดเร็วเมื่อโมเดลทยอยผิดแต่มีค่าใกล้เคียงกับค่าจริงมากขึ้น และสุดท้ายคือด้านขวาสุดความคลาดเคลื่อนลดลงเรื่อยๆ จนเป็นศูนย์เมื่อโมเดลทยอยถูก

2.5 การแบ่งส่วนคลื่นแม่เหล็กไฟฟ้า 24GHz ของการหายใจโดยใช้ GMM

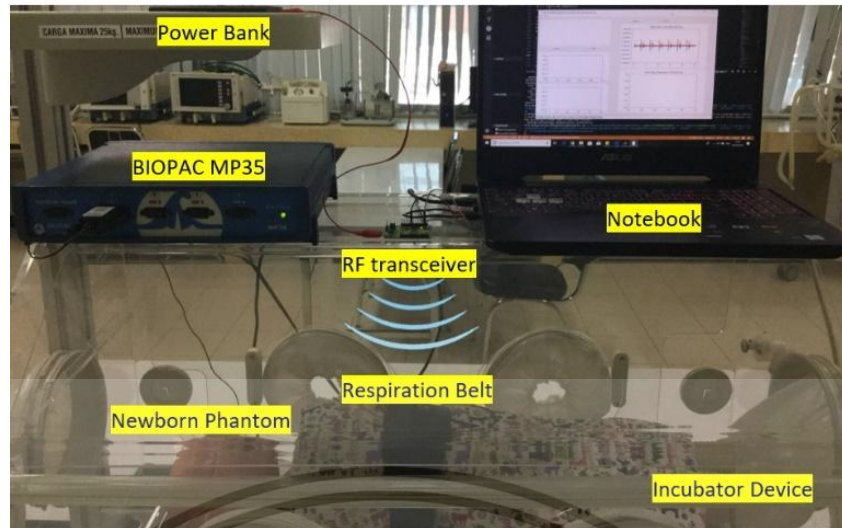


รูปที่ 2.22 การสร้างแบบจำลองของการตรวจจับสัญญาณชีพเรดาร์ดอปเปลอร์ด้วยตัวรับส่งสัญญาณ

ในปัจจุบันสัญญาณอัตราการหายใจ (respiration rate signal) มีความสำคัญอย่างมากในการวิเคราะห์ภาวะหัวใจและหลอดเลือด ซึ่งเป็นประโยชน์ในการป้องกันความเสี่ยงของภาวะหัวใจวายฉับพลันที่เป็นสาเหตุการเสียชีวิตอันดับต้นๆของคนทั่วโลก นอกจากนี้สัญญาณของอัตราการหายใจสามารถนำมาวิเคราะห์กายภาพของปอด อาการน้ำท่วมปอด (Pulmonary Edema) และศึกษาปริมาณความจุของปอด

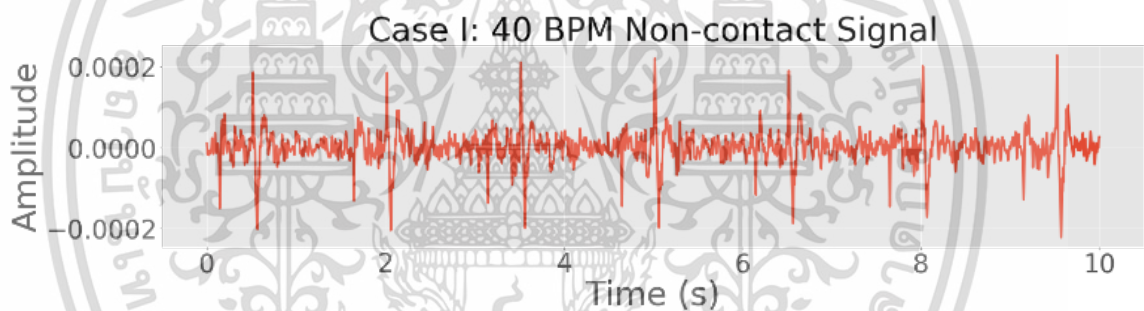
และในการวัดสัญญาณการหายใจแบบดั้งเดิมคือการใช้เซ็นเซอร์สัมผัสติดกับตัวผู้ป่วยไม่ว่าจะเป็นการติดอิเล็กโทรดวัดคลื่นไฟฟ้าหัวใจ การรัดเข็มขัดเพื่อวัดการเคลื่อนไหวของผนังทรวงอกหรือช่องท้องกับตัวผู้ป่วยอย่างไรก็ตามวิธีการเหล่านี้เป็นอุปสรรคกับกลุ่มเด็กทารกแรกเกิดและกลุ่มผู้ป่วยโรคผื่นสัมผัส (contact dermatitis) เป็นต้น และนอกจากนี้ยังพบอีกว่าอุปกรณ์ที่ใช้วัดสัญญาณดังกล่าวนี้ไม่มีราคาแพง

จากปัญหาที่กล่าวมาข้างต้นนี้จึงเป็นที่มาของปริณญาณิพนธ์ที่จะนำการประยุกต์ใช้คลื่นแม่เหล็กไฟฟ้าในการวัดอัตราการหายใจแบบไร้การสัมผัสผิวหนัง (non-contact respiration measurement) ซึ่งช่วยในการคัดกรองได้อย่างรวดเร็วมีขนาดเล็กและราคาถูก โดยใช้ความถี่ 24GHz มาประยุกต์ในการตรวจจับดอปเปลอร์ของการหายใจเทียบกับวิธีการวัดแบบดั้งเดิม

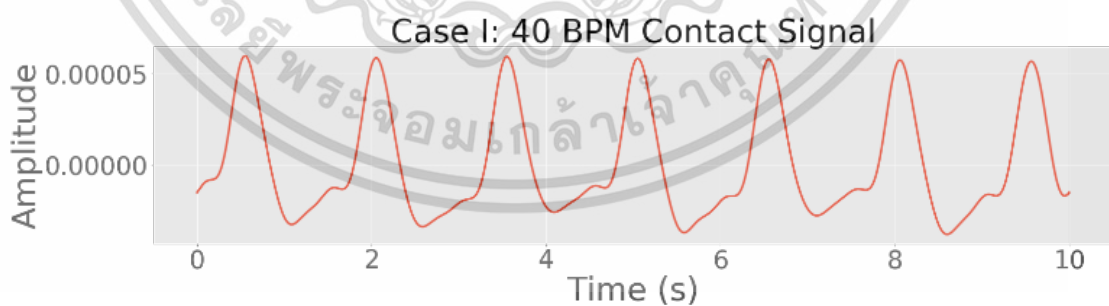


รูปที่ 2.23 ภาพอุปกรณ์สำหรับเก็บข้อมูล และแบบจำลองทารก

ในการเก็บข้อมูลในครั้งนี้ จะมีการเปรียบเทียบสัญญาณระหว่าง 2 อุปกรณ์ได้แก่ BIOPAC MP35 ดังรูปที่ 2.24 และ RF transceiver ดังรูปที่ 2.25



รูปที่ 2.24 สัญญาณจากอุปกรณ์ BIOPAC MP35



รูปที่ 2.25 สัญญาณจากอุปกรณ์ RF transceiver

ตารางที่ 2.1 PARAMETER FEATURE EXTRACTION LIST	
Algorithm	MFCC + 1ST DELTA + 2ND DELTA
Size of signal window 20 ms	20 ms
Number of Mel filters	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Number of MFCC coefficients	40
HOP SIZE	80
Number of FFT	256

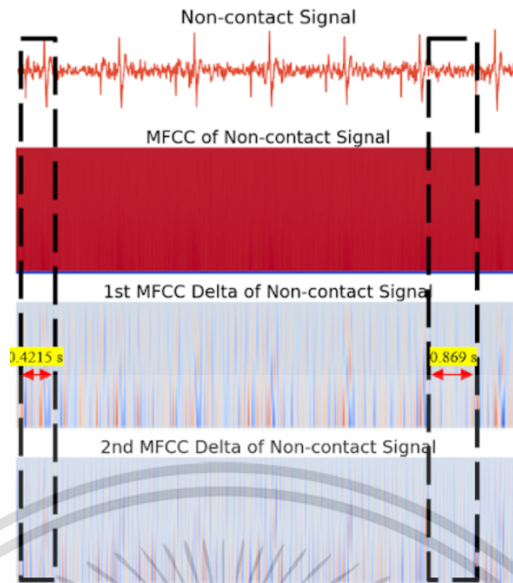
และในแบบจำลองนี้ยังมีการสกัดคุณสมบัติของสัญญาณด้วยเทคนิค MFCC โดยใช้พารามิเตอร์ดังตารางที่ 2.1 เพื่อเป็นข้อมูลสำหรับนำมาพัฒนาปัญญาประดิษฐ์ ที่สามารถแยกแยะข้อมูล ช่วงเวลาที่หายใจ และไม่หายใจ ดังตารางที่ 2.2 และ 2.3

ตารางที่ 2.2 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC ของการไม่หายใจ										
MFCC + 1 ST DELTA + 2 ND DELTA OF NOISE SIGNAL										
	0	1	2	3	...	115	116	117	118	119
0	-627.87781	6.443981	6.354864	6.208219	...	0.062651	0.050236	0.037733	0.025177	0.012594
1	-628.30822	5.858839	5.839834	5.808276	...	0.062651	0.050236	0.037733	0.025177	0.012594
...

ตารางที่ 2.3 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC ของการหายใจ										
MFCC + 1 ST DELTA + 2 ND DELTA OF RESPIRATION SIGNAL										
	0	1	2	3	...	115	116	117	118	119
0	-333.65447	90.54193	33.67920	23.71586	...	-0.04914	-	-	-	-
1	391.23765	51.78551	48.07298	42.51066	...	-0.04914	-	-	-	-
...
107	-626.30315	8.672255	8.587021	8.446114	...	0.03452	0.02764	0.02075	0.0138	0.00692
3	624.70817	10.92472	10.83002	10.67342	...	0.03452	0.02764	0.02075	0.0138	0.00692
4	624.70817	9	8	6	...	5	9	7	5	9

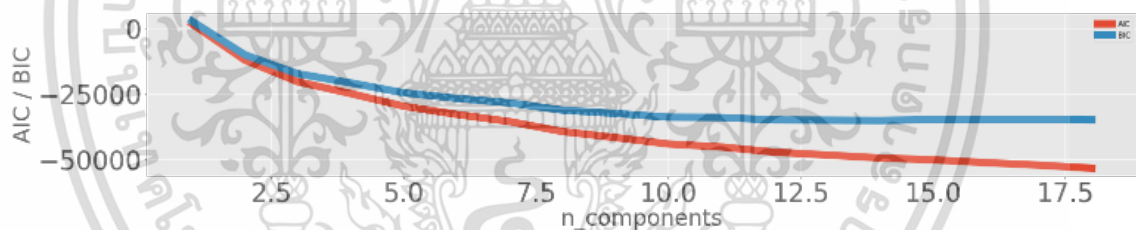
จากรูปที่ 2.26 รูปภาพจะแสดงคุณสมบัติที่ MFCC, MFCC Delta 1 และ MFCC Delta 2 ในกระบวนการนี้เราจะสร้างชุดข้อมูลเพื่อเข้าสู่กระบวนการเรียนรู้ของเครื่องเพื่อสร้างแบบจำลองการแบ่งส่วน ระหว่างการหายใจและไม่หายใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



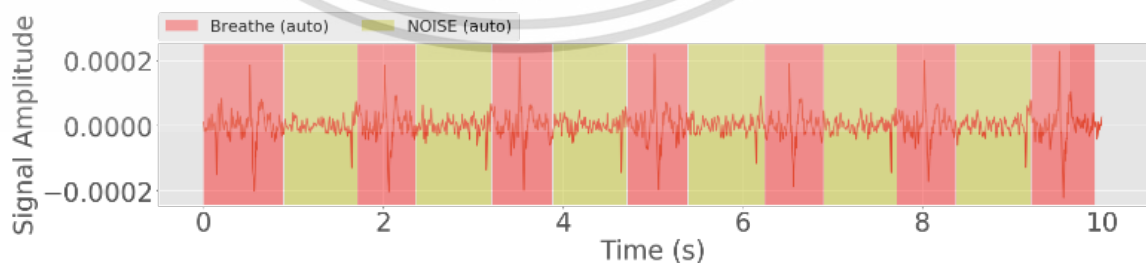
รูปที่ 2.26 คุณสมบัติของสัญญาณด้วยเทคนิค MFCC สำหรับนำมาพัฒนาปัญญาประดิษฐ์

การสร้างชุดข้อมูลแบ่งออกเป็น 2 คลาส ชั้นแรกคือหายใจ เราได้สร้างหน้าต่าง 0.4215 วินาที หรือ 3,372 ตัวอย่างเช่นสัญญาณที่ถูกครอบตัดโดยเฉพาะสัญญาณการหายใจและคุณลักษณะที่แสดงในตารางที่ 2.2 คลาสที่สองคือ ไม่หายใจ ตัวอย่าง 0.869 วินาที หรือ 6,952 สำหรับการครอบตัดเฉพาะเสียงและลักษณะที่แสดงในตารางที่ 2.1

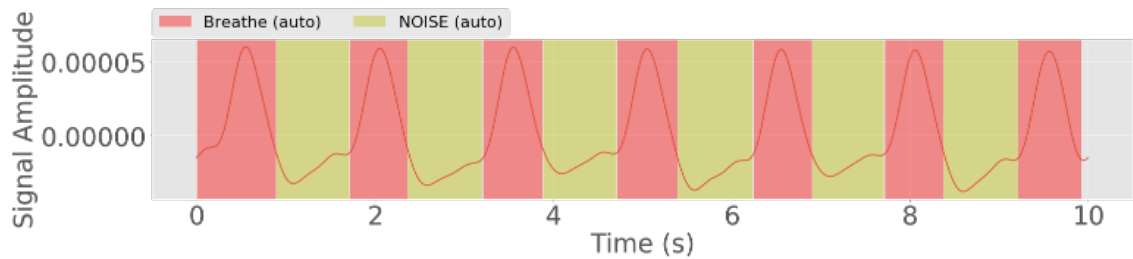


รูปที่ 2.27 แบบจำลอง AIC / BIC

จากภาพที่ 2.27 แสดงแบบจำลองที่เหมาะสมที่สุด โดยใช้ค่า AIC และ BIC ต่ำที่สุดจะเป็นตัวแบบที่ให้ค่าพยากรณ์แม่นยำ และถูกต้อง

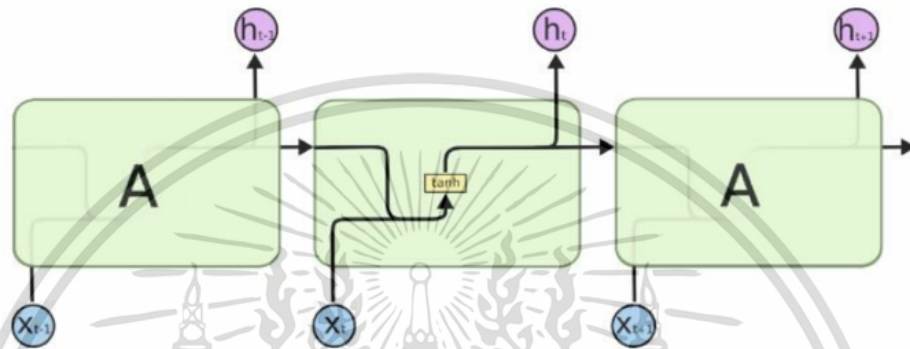


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 การเปรียบเทียบการแบ่งส่วนด้วยเทคนิค GMM ระหว่างสัญญาณหายใจ และไม่หายใจ

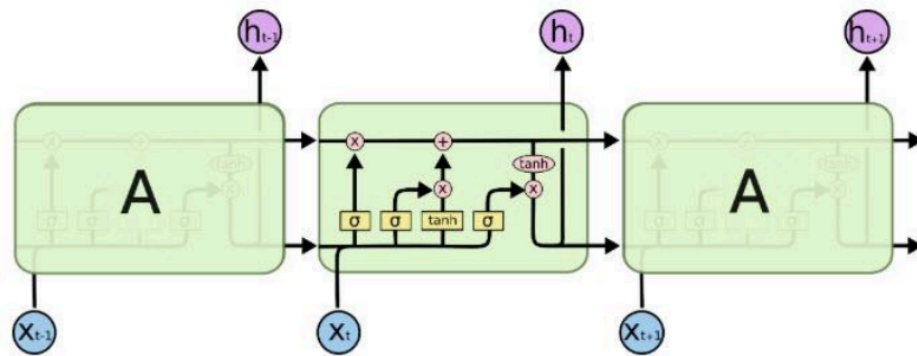
2.6 ทฤษฎีเกี่ยวกับ Long Short-Term Memory (LSTM)



รูปที่ 2.29 แสดงโครงสร้างการทำงานของ RNN

Long Short-Term Memory Model (LSTM) เป็นเทคนิคหนึ่งที่ถูกพัฒนาจาก Recurrent neural network (RNN) ซึ่ง RNN นั้นมีหลักการทำงาน คือ การนำ output (ผลลัพธ์) ที่ได้จากการคำนวณจากโหนดก่อนหน้านี้กลับมาใช้เป็นข้อมูล input ของโหนดถัดไป ซึ่งแต่ละโหนดของ RNN นั้นจะมีข้อมูลที่เข้ามา 2 ส่วน คือ ข้อมูล input ของโหนดนั้น ๆ กับ output ที่ผ่านการคำนวณจากโหนดก่อนหน้านี้ โดยข้อมูลทั้ง 2 ชุดที่เข้ามาในโหนดจะถูกรวมเข้าด้วยกัน ก่อนจะถูกแยกผลลัพธ์ออกเป็น 2 ส่วน คือ ผลลัพธ์ที่ได้จากโหนดนั้น ๆ และผลลัพธ์ที่จะถูกนำไปเป็นข้อมูล input ของโหนดถัดไป เทคนิค RNNs นั้นเหมาะนำมาใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (sequence) หรือข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลอนุกรมเวลา (Time Series), ข้อมูลเสียง, ข้อมูลประเภท ข้อความ, ข้อมูลรูปภาพและวิดีโอ เป็นต้น

ข้อดีของ RNN คือ สามารถนำข้อมูลก่อนหน้า (ในอดีต) มาใช้ในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคตได้ ส่วนข้อเสียของ RNN คือ จะสามารถดูข้อมูลย้อนหลังได้แค่เพียงระยะสั้น ๆ เท่านั้น ซึ่งทำให้เกิดปัญหาในการทำ Backpropagation หรือการคำนวณหาค่าความผิดพลาดย้อนหลังของแต่ละโหนดเมื่อสิ้นสุดการทำงาน เพราะการทำ Backpropagation นั้นจะต้องทำย้อนกลับไปหลายขั้นตอนและหลายโหนด จึงทำให้เกิดปัญหา Vanishing Gradient Problem ดังนั้นเพื่อแก้ปัญหาดังกล่าวจึงทำให้เกิดเทคนิค LSTM ขึ้น



รูปที่ 2.30 แสดงโครงสร้างการทำงานของ LSTM

Long short-term memory (LSTM) เป็นโครงข่ายประเภท RNN รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น LSTM เริ่มเป็นที่รู้จักในปี ค.ศ.1997 โดย Hochreiter และ Schmidhuber โดยมีหลักการทำงาน คือ สามารถเก็บ 'สถานะ' หรือข้อมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับไปได้จะดูย้อนถึงที่มา ของข้อมูลค่าดังกล่าวว่าเดิมเป็นค่าอะไร และจุดเด่นของเทคนิค LSTM คือฟังก์ชันพิเศษที่มีหน้าที่ เหมือน 'ประตู (Gate)' ที่คอยควบคุมข้อมูลที่เข้ามาในแต่ละโหนด ซึ่งประกอบไปด้วย Forget gate layer, Input gate layer และ Output gate layer



รูปที่ 2.31 แสดงโครงสร้าง Forget gate layer

เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน cell state นั้นควรจะถูกลบไว้ หรือ ควรที่จะทิ้งไป ซึ่งข้อมูลที่ถูกตัดสินใจว่าจะเก็บไว้นั้นจะถูกประเมินจากข้อมูล input ที่เข้ามาในโหนด นั้น ๆ รวมกับผลลัพธ์ที่ได้จากการคำนวณของโหนดก่อนหน้า ผ่านฟังก์ชัน sigmoid ดังสมการที่ (2.7)

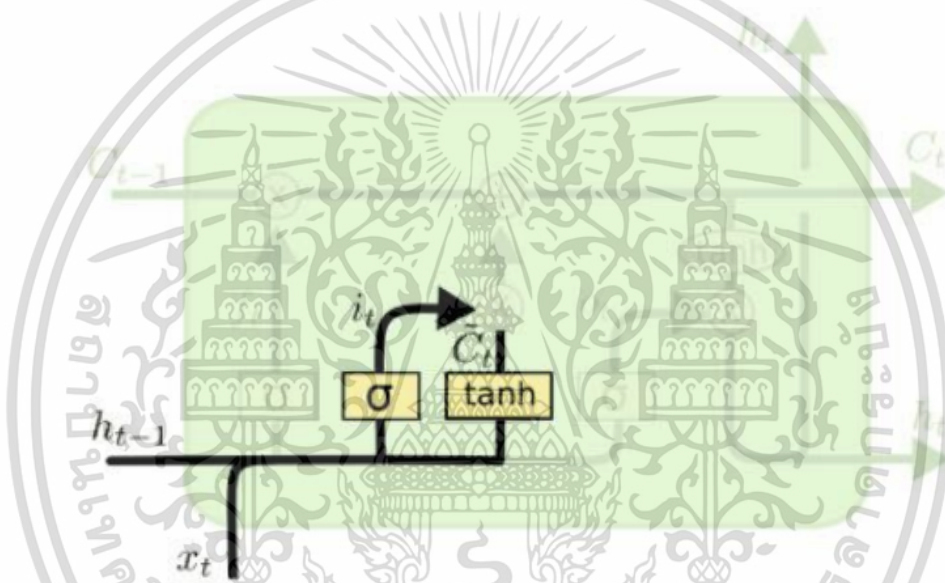
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ

ft	คือ Forget gate
σ	คือ ฟังก์ชัน sigmoid
Wf	คือ ค่าน้ำหนักของ matrices
$ht-1$	คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp t-1)
xt	คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t และ bf คือ ค่า bias

ผลลัพธ์ที่ได้จาก Forget gate layer จะอยู่ระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้นหมายถึงให้ลบค่า cell state เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า cell state นี้ต่อไป



รูปที่ 2.32 แสดงโครงสร้าง Input gate layer

เป็น Gate ที่มีหน้าที่รับข้อมูล input เข้ามาใหม่แล้วจึงทำการบันทึกหรือ 'เขียน (write)' ข้อมูลลงในแต่ละโหนด โดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ ถ้าต้องการ update cell state เมื่อทำการรับข้อมูล input เข้ามาแล้วฟังก์ชัน sigmoid ที่เป็นตัวควบคุมจะเรียกใช้ input gate เพื่อเลือกว่าจะให้ update cell state หรือไม่ต้อง และในส่วนที่สอง ถ้า input gate เลือกที่จะทำการ update cell state ฟังก์ชัน tanh ก็จะทำให้การสร้าง candidate values (C_t) ขึ้นมาใน state ดังสมการที่ (2.8) และ (2.9)

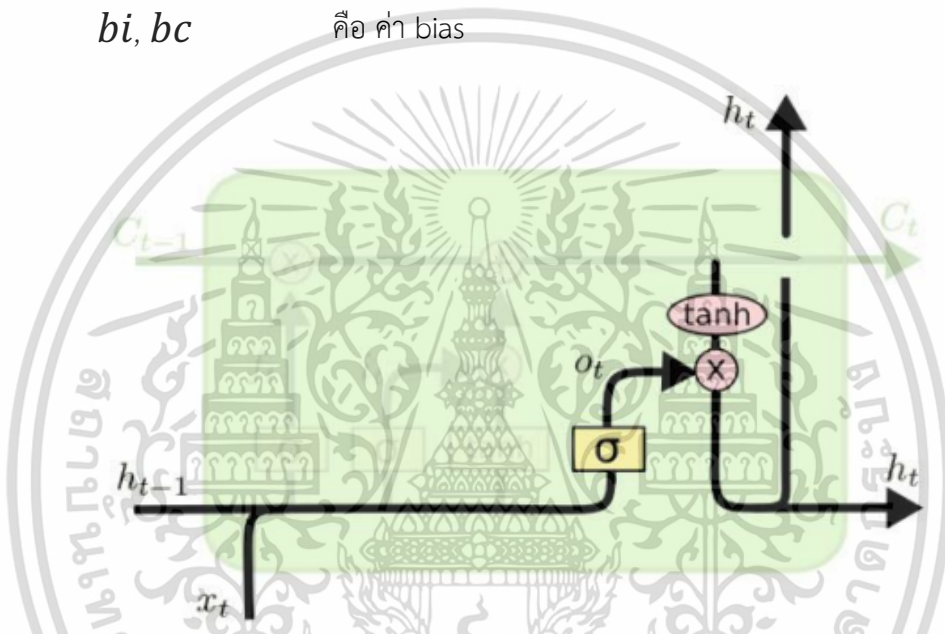
$$it = \sigma (Wi \cdot [ht-1, xt] + bi) \quad (2.8)$$

$$Ct = \tanh (Wc \cdot [ht-1, xt] + bc) \quad (2.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ

it	คือ Input gate
σ	คือ ฟังก์ชัน sigmoid
C_t	คือ ค่า candidate ของ cell state ที่เวลา t
\tanh	คือ ฟังก์ชัน tanh
W_i, W_c	คือ ค่าน้ำหนักของ matrices
h_{t-1}	คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp $t-1$)
x_t	คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t
b_i, b_c	คือ ค่า bias



รูปที่ 2.33 แสดงโครงสร้าง Output gate layer

เป็น Gate ที่มีหน้าที่เตรียมทำการส่งออกข้อมูล (output data) โดยข้อมูลที่จะทำการ output นั้นจะดูจาก cell state ที่ผ่านกระบวนการคำนวณต่างๆ แล้ว โดยฟังก์ชัน sigmoid จะเป็นตัวเลือกว่าข้อมูลส่วนไหนใน cell state ที่จะถูก output จากนั้นก็นำค่า cell state เข้าฟังก์ชัน tanh (เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1) แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า output ที่ได้จาก sigmoid gate จากนั้นก็จะได้ค่า output ที่ต้องการดังสมการที่ (2.10)

$$ot = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.10)$$

จากสมการ

ft	คือ Forget gate
σ	คือ ฟังก์ชัน sigmoid
W_f	คือ ค่าน้ำหนักของ matrices
h_{t-1}	คือ ค่า output ของ cell state ก่อนหน้า (ที่ timestamp $t-1$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

xt คือ ค่า input ที่เข้ามาใน cell state ณ เวลา t
 bo คือ ค่า bias

ซึ่งค่า output ที่ได้ออกมาจะถูกแบ่งออกเป็น 2 ส่วน คือค่า output ที่ได้จากโหนดนั้นๆ กับค่า output ที่จะถูกส่งไปเป็นข้อมูล input ของโหนดถัดไป

2.7 วัดประสิทธิภาพ Model จาก Confusion Matrix

Confusion Matrix คือตารางสำคัญในการวัดความสามารถของ machine learning ในการแก้ปัญหา classification

การที่จะนำโมเดลใช้งานจริงได้นั้น จำเป็นต้องมีการวัดประสิทธิภาพโมเดล ก่อนว่าโมเดลนั้นมีประสิทธิภาพเพียงพอที่จะนำมาพัฒนา หรือนำไปใช้งานด้านต่างๆ ซึ่งการวัดประสิทธิภาพนั้นส่วนใหญ่จะวัดค่าจากใน Table ข้อมูลที่มี (Confusion Matrix)

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

รูปที่ 2.34 แสดงความสัมพันธ์ระหว่าง Actual values กับ Predicted values

True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” และมีค่าเป็น “จริง”

True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” และมีค่า “ไม่จริง”

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” แต่มีค่าเป็น “ไม่จริง”

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” แต่มีค่าเป็น “จริง”

โดยทั่วไปแล้วจะมีตัววัดที่นิยมใช้กันในงานวิจัยและการทำงานต่างๆ อยู่ 3 ค่า และสมการ คือ

1. Precision เป็นการวัดความแม่นยำของข้อมูล โดยพิจารณาแยกที่ละคลาส

$$\frac{TP}{TP+FP} \quad (2.11)$$

2. Recall เป็นการวัดความถูกต้องของ Model โดยพิจารณาแยกที่ละคลาส

$$\frac{TP}{TP+FN} \quad (2.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F1-Score คือค่าเฉลี่ยแบบ harmonic mean ระหว่าง precision และ recall สร้าง F1 ขึ้นมาเพื่อเป็น single metric ที่วัดความสามารถของโมเดล (ไม่ต้องเลือกระหว่าง precision, recall เพราะเฉลี่ยให้แล้ว)

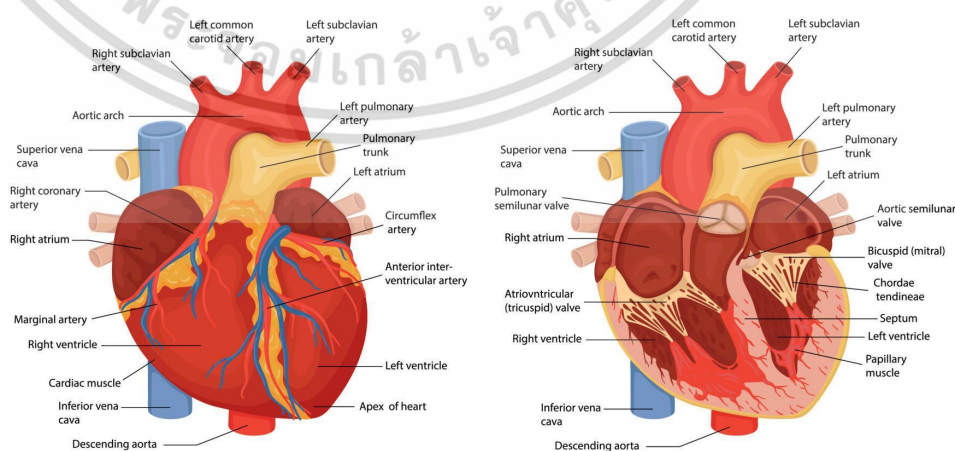
$$F1 = 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \right) \quad (2.13)$$

2.8 โครงสร้างของหัวใจ

ลักษณะของหัวใจเป็นอวัยวะที่ทำหน้าที่ในการสูบฉีดเลือดออกไปเลี้ยงทั่วร่างกายตั้งอยู่ในช่องอกด้านหลังของกระดูกหน้าอก (Sternum) อยู่ระหว่างซี่โครงซี่ที่ 2-6 โดยอยู่ระหว่างปอดทั้ง 2 ข้างเหนือกระบังลมมีรูปร่างคล้ายกรวย (cone shape) ส่วนปลายของหัวใจเรียกว่า apex มีปลายชี้ไปทางด้านซ้าย มีขนาดโตเท่ากับกำปั้นของเจ้าของหัวใจ

หัวใจประกอบด้วยเซลล์กล้ามเนื้อเนื้อเป็นส่วนใหญ่ แบ่งห้องหัวใจออกเป็น 4 ห้อง 2 ซีก คือ ซีกซ้าย และซีกขวา โดยมีผนังกล้ามเนื้อเนื้อกัน เรียกว่า septum และแต่ละซีกแบ่งเป็นห้องบนและห้องล่าง สองห้องบนเรียกว่า atrium สองห้องล่างเรียกว่า ventricle ผนังของ atrium บางเมื่อเทียบกับ ผนังของ ventricle จึงยืดขยายได้ง่ายกว่า ทำให้รับเลือดที่กับสู่หัวใจได้แม้เพียงความดันต่ำๆ ส่วนผนังของ ventricle หนา เพราะจะต้องบีบตัวแรงเพื่อเอาชนะความดันในหลอดเลือดแดงใหญ่ จึงจะสามารถบีบเลือดออกไปได้ หัวใจซีกขวามีหน้าที่ขับเลือดที่ร่างกายใช้แล้วไปพอกที่ปอด ส่วนหัวใจซีกซ้ายมีหน้าที่รับเลือดจากปอดที่พอกแล้วไปเลี้ยงส่วนต่างๆของร่างกาย ดังนั้นหัวใจซีกล่างซ้าย ต้องทำงานหนักมากในการบีบตัวดันเลือดไปสู่ระบบหลอดเลือดแดง ซึ่งมีความดันและความต้านทานสูง เมื่อเทียบกับหัวใจห้องล่างขวา ซึ่งบีบตัวดันเลือดไปปอด มีความดันและความต้านทานต่ำกว่าส่งผลทำให้หัวใจห้องล่างซ้ายจึงหนากว่าห้องล่างขวาประมาณ 2-3 เท่า ดังรูปที่ 2.34

Human heart anatomy



รูปที่ 2.35 กายวิภาคของหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

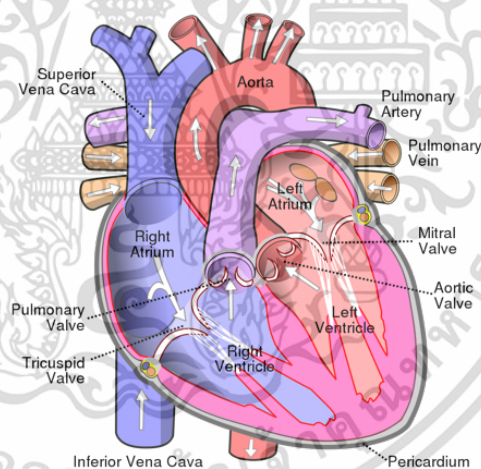
และจากที่กล่าวมาแล้วว่าหัวใจคนเรามี 4 ห้องแบ่ง ซ้าย-ขวา และแบ่งห้องบน-ล่าง ก็จะมี ลิ้นหัวใจกันระหว่างห้องแบ่งเป็น 4 ลิ้นด้วยกันแสดงดังรูปที่ 2.35 ดังนี้ คือ

1. ลิ้นไตรคัสปิด (Tricuspid valve) อยู่ระหว่างหัวใจห้องบนขวาและล่างขวามีลักษณะเป็นแผ่น 3 แผ่น ที่ขอบของแต่ละลิ้นจะยึดติดกับเนื้อเยื่อเกี่ยวพันและผนังของห้องล่างขวา เพื่อควบคุมการปิดเปิดลิ้น ป้องกันไม่ให้เลือดในหัวใจห้องล่างขวา ไหลย้อนกลับขึ้นสู่หัวใจห้องบนขวา

2. ลิ้นพัลโมนารี เซมิลูนาร์ (Pulmonary semilunar valve) อยู่ที่โคนของเส้นเลือดแดง พัลโมนารีมีลักษณะเป็นถุงรูปพระจันทร์ครึ่งเสี้ยว 3 ใบบรรจบกันแต่ไม่ได้ยึดติดด้วยเนื้อเยื่อเกี่ยวพันทำหน้าที่ป้องกันไม่ให้เลือดไหลกลับสู่หัวใจห้องล่างขวา

3. ลิ้นไบคัสปิดหรือลิ้นไมตรัล (Bicuspid valve or Mitral valve) อยู่ระหว่างหัวใจห้องบนซ้ายและห้องล่างซ้ายมีลักษณะเป็นแผ่น 2 แผ่น ที่ขอบของแต่ละลิ้นยึดติดกับเนื้อเยื่อเกี่ยวพัน และผนังของหัวใจห้องล่างซ้ายเพื่อควบคุมการปิดเปิดลิ้น ทำหน้าที่ป้องกันไม่ให้เลือดในหัวใจห้องล่างซ้าย ไหลย้อนขึ้นไปหัวใจห้องบนซ้าย

4. ลิ้นเอออร์ติก เซมิลูนาร์ (Aortic semilunar valve) อยู่ที่โคนของหลอดเลือดเอออร์ตารมีลักษณะเป็นถุงรูปพระจันทร์ครึ่งเสี้ยว 3 ใบบรรจบกันไม่ได้ยึดติดกันด้วยเนื้อเยื่อเกี่ยวพัน ทำหน้าที่ป้องกันไม่ให้เลือดไหลกลับสู่หัวใจห้องล่างซ้าย



รูปที่ 2.36 ตำแหน่งของลิ้นหัวใจ

2.9 ปัจจัยที่มีผลต่อค่าอัตราการเต้นของหัวใจ (Heart Rate : HR)

โดยปกติแล้วค่าอัตราการเต้นของหัวใจในผู้ใหญ่ขณะพักประมาณ 60-100 ครั้งต่อนาที แต่อาจขึ้นกับอายุ หรือการมีกิจกรรมต่าง ๆ เช่น ในเด็กจะมีอัตราการเต้นของหัวใจสูงกว่าในวัยผู้ใหญ่ หรือในขณะที่หลับอัตราการเต้นของหัวใจอาจจะลดลงประมาณ 10-20 ครั้งต่อนาทีค่าอัตราการเต้นของหัวใจแสดงถึงคุณสมบัติทางไฟฟ้าของหัวใจ ดังนั้นปัจจัยที่มีผลทำให้เกิดการเปลี่ยนแปลงของอัตราการเต้นของหัวใจต้องเป็นปัจจัยที่มีผลต่อการสร้างการเปลี่ยนแปลงศักย์ไฟฟ้าภายในเซลล์ (Action potential) ของตำแหน่งการเกิดไฟฟ้าในหัวใจ (SA node) ซึ่งในภาวะปกติถือว่าเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวกำหนดอัตราการเต้นของซึ่งโดยปกติอัตราการเต้นหัวใจขณะพักในผู้ใหญ่เฉลี่ยจะมีค่าอยู่ระหว่าง 60-100 ครั้งต่อนาทีซึ่งปัจจัยที่มีอิทธิพลต่อค่าการเต้นของหัวใจมีดังนี้

1. อายุ เมื่ออายุเพิ่มขึ้นอัตราการเต้นของหัวใจจะลดลง ในผู้ใหญ่อัตราการเต้นชีพจรจะเท่ากับ 60-100 (เฉลี่ย 80 ครั้งต่อนาที)

2. เพศ หลังวัยรุ่นค่าเฉลี่ยของอัตราการเต้นของหัวใจของผู้ชายจะต่ำกว่าหญิงเล็กน้อย

3. การออกกำลังกาย อัตราการเต้นของหัวใจจะเพิ่มขึ้นเมื่อออกกำลังกายซึ่งถูกกระตุ้นโดยการตอบสนองของเคมีรีเซปเตอร์ (Chemoreceptor) กับการขาดเลือด (ischemia) ของกล้ามเนื้อ ทำให้มีการเพิ่มความดันเลือดและจังหวะการเต้นของหัวใจ

4. ไข้ อัตราการเต้นของหัวใจจะเพิ่มขึ้นเพื่อปรับตัวให้เข้ากับความดันเลือดที่ต่ำลงซึ่งเป็นผลมาจากเส้นเลือดส่วนปลายขยายตัวทำให้อุณหภูมิร่างกายสูงขึ้นเพิ่มอัตราการเผาผลาญพลังงาน (Metabolic rate)

5. ยา ยาบางชนิดลดอัตราการเต้นของหัวใจ เช่น ยาโรคหัวใจ (Digitalis) ซึ่งมีผลไปลดการนำกระแสไฟฟ้าผ่านทางเดินของกระแสผ่านในหัวใจ (Conducting system) ผ่านทางเอ-วี โหนด (AV node) และมีผลในการลดอัตราการเต้นของหัวใจ แต่ยาจะมีผลในการเพิ่มการบีบตัวของหัวใจ ซึ่งมีผลในการเพิ่มปริมาณเลือดที่ออกจากหัวใจใน 1 นาที (cardiac output) นอกจากนี้ยังมีผลทำให้ระบบไหลเวียน (circulation) และเลือดที่ไหลไปเลี้ยงเนื้อเยื่อ (tissue perfusion) ได้ดีขึ้นส่วนยาที่มีผลกระตุ้นหัวใจได้แก่ยาที่มีส่วนผสมของคอฟฟี่ (coffee, caffeine) และโทแบคโค (tobacco)

6. ภาวะเลือดออก (Hemorrhage) การสูญเสียเลือดจะมีผลทำให้เพิ่มการกระตุ้นระบบประสาทพาราซิมพาเทติก ทำให้อัตราการเต้นของหัวใจสูงขึ้น โดยปกติในผู้ใหญ่มีเลือดประมาณ 5 ลิตร การสูญเสียเลือดไปน้อยกว่า 10% จึงจะไม่มีผลข้างเคียง

7. อารมณ์ การตอบสนองของระบบการทำงานของหัวใจต่อสภาวะอารมณ์เกิดขึ้นในสมองส่วนซีรีบรอล คอเทก (Cerebral cortex) และส่งไปยังเม็ดดูลลารี คาร์ดิโอ วาสคูลาร์ เซ็นเตอร์ (medullary cardiovascular center) เมื่อเกิดความกลัว ความวิตกกังวล จะกระตุ้นระบบประสาทซิมพาเทติก (sympathetic nervous) ทำให้มีผลเพิ่มความดันเลือด มีผลเพิ่มอัตราเต้นของหัวใจและการทำงานของระบบประสาทพาราซิมพาเทติก

8. ความเจ็บปวด โดยปกติความรู้สึกในระดับตื้น (Superficial or subcutaneous pain) จะเพิ่มการทำงานของระบบประสาทซิมพาเทติก มีผลทำให้ความดันเลือดสูงขึ้น และการเต้นของหัวใจเพิ่มขึ้น แต่ความรู้สึกเจ็บในระดับลึก (deep pain) ในอวัยวะภายในหรือภายในข้อต่อมีผลในการลดการทำงานของระบบประสาทซิมพาเทติกแต่เพิ่มการทำงานของระบบพาราซิมพาเทติก มีผลทำให้ความดันเลือดลดลงและอัตราการเต้นของหัวใจลดลงอย่างรวดเร็ว ซึ่งอาจทำให้เกิดภาวะช็อค ได้

9. ท่าทาง เมื่ออยู่ในท่ายืนหรือนั่งหัวใจจะเต้นเพิ่มขึ้น (เร็วขึ้น) ท่านอนหัวใจจะเต้นลดลง (ช้า)

10. สิ่งแวดล้อม ได้แก่ เสียง การเปลี่ยนแปลงของอุณหภูมิ มลภาวะทางอากาศ ก็จะทำให้

อัตราการเต้นของหัวใจเพิ่มขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ระเบียบวิธีวิจัย

3.1 ระบบวิธีการวิจัย

ในหัวข้อนี้เราจะพูดถึงวิธีการศึกษาที่เกี่ยวข้องกับการใช้คลื่น FMCW และ AI เพื่อตรวจสอบสัญญาณหัวใจโดยไม่ต้องสัมผัสในการทดสอบนั้นเราใช้โมเดลทั้งหมด 3 รูปแบบ คือ Bidirectional LSTM Classification, Bidirectional RNN Classification, และ Bidirectional GRU Classification เพื่อทำการทำนายกลุ่มของช่วงสัญญาณที่วัดได้ ซึ่งกลุ่มช่วงสัญญาณถูกแบ่งออกเป็น 4 สถานะ คือ ช่วงที่มีความแม่นยำสูงสุดในการวัดสัญญาณ, ช่วงที่มีความแม่นยำปานกลางในการวัดสัญญาณ, ช่วงที่มีความแม่นยำต่ำสุดในการวัดสัญญาณและช่วงที่ไม่ต้องการวัดสัญญาณ

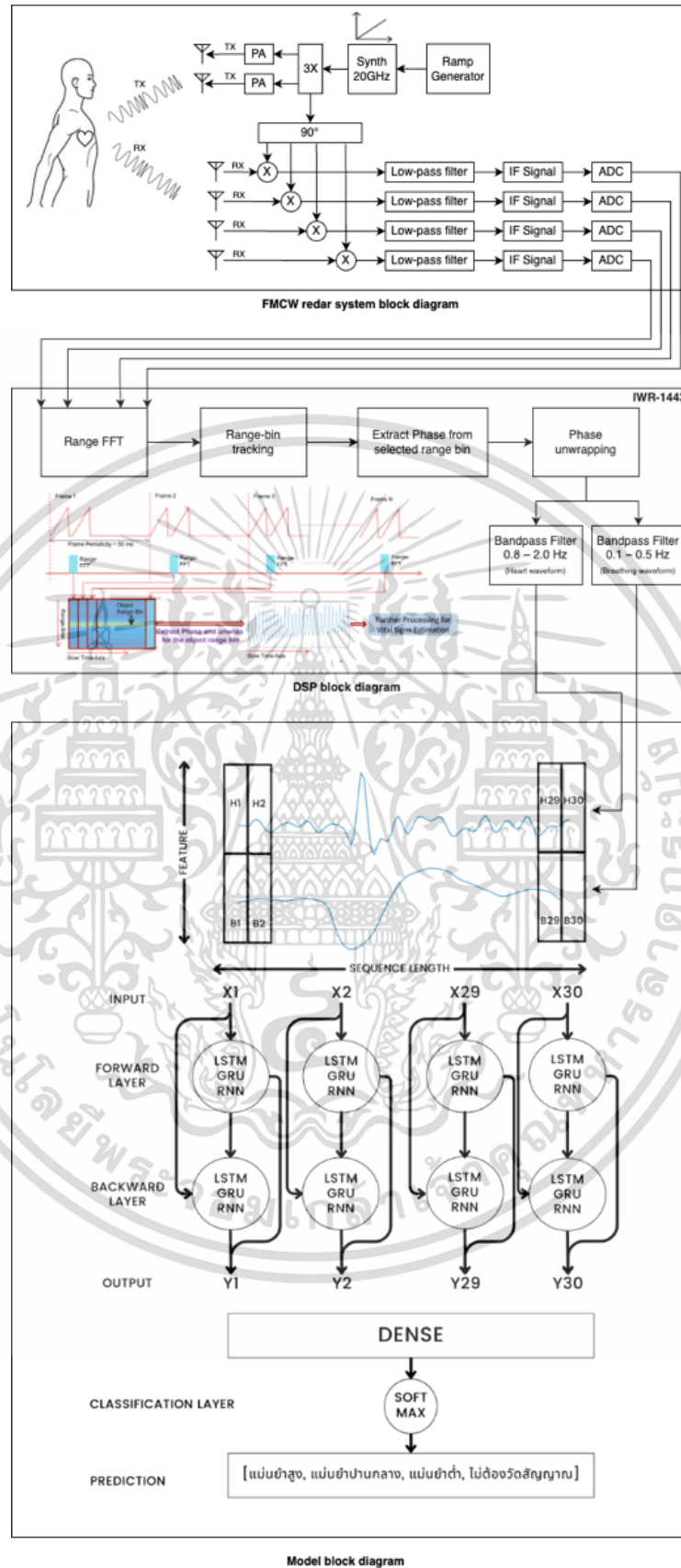
เพื่อประเมินผลการทดสอบของโมเดลที่ได้แล้ว ผู้วิจัยได้ใช้วิธี k-fold cross validation โดยใช้จำนวนครั้ง 5 ครั้งต่อโมเดล และนำผลลัพธ์ที่ได้จากแต่ละรอบมาสร้าง ensemble model โดยใช้ decision tree เพื่อช่วยในการตัดสินใจ

เพื่อวัดผลการทดสอบของโมเดลที่ได้ ผู้วิจัยได้ใช้เมตริกการประเมินผลคือ precision, recall, f1-score และ confusion matrix เพื่อแสดงภาพรวมของการศึกษา ซึ่งรายละเอียดเพิ่มเติมสามารถดูได้ในรูปที่ 3.1



รูปที่ 3.1 ระบบวิธีการวิจัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ภาพรวมการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับภาพรวมการทำงานของระบบ ภาพที่ 3.2 เราได้แบ่งออกเป็น 3 ส่วน ได้แก่

1. FMCW radar system block diagram: ในส่วนนี้ผู้วิจัยได้เลือกใช้งานอุปกรณ์ Batman BM101 mmWave EVM kit ที่มีวงจร RF ของสัญญาณ FMCW โดยประกอบด้วย TX จำนวน 2 เสาสัญญาณ และ RX 4 เสาสัญญาณ ซึ่งสร้างความถี่ 77 GHz โดยมี แบนด์วิดท์ 4 GHz

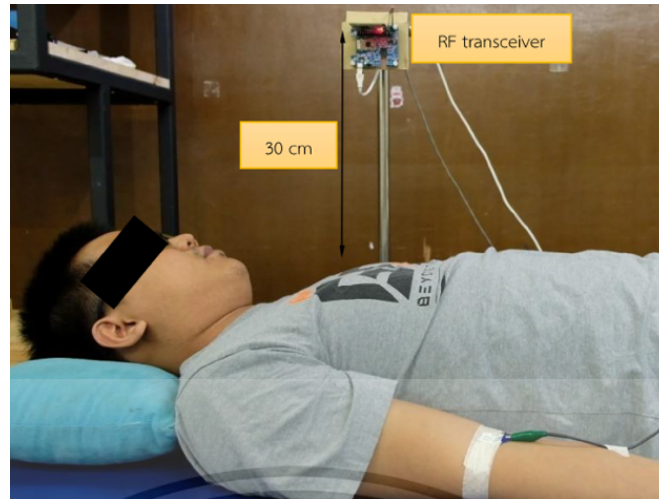
2. DSP block diagram: ในส่วนนี้ผู้วิจัยได้เลือกใช้งานอุปกรณ์ Batman BM101 mmWave EVM kit โดยมีชิป IWR-1443 โดยมีการโปรแกรม Digital Signal Processor สำหรับการนำสัญญาณที่ได้จาก ADC จาก FMCW radar system block diagram นำมาคำนวณเพื่อให้ได้ Heart Waveform และ Breathing Waveform

3. Model block diagram: ในส่วนนี้ผู้วิจัยได้เลือกนำโมเดล sequence classification ทั้งหมด 3 โมเดล ได้แก่ BiRNN + classification, BiLSTM + classification, BiGRU + classification โดยใช้ Heart Waveform และ Breathing Waveform มาเทรนเพื่อให้โมเดลสามารถแยกแยะช่วงเวลาในการวัดสัญญาณที่มีความแม่นยำสูงสุด

3.2 การเก็บข้อมูล

ขั้นตอนในการวัดสัญญาณชีพเพื่อนำมาใช้ในการพัฒนาโมเดลเป็นดังนี้

1. คัดเลือกอาสาสมัครระดับปริญญาตรีทั้งหมด 30 คนที่มีสุขภาพดีเพื่อให้เข้าร่วมทดลองนี้
2. แบ่งกลุ่มอาสาสมัครเป็นผู้หญิง 15 คนและผู้ชาย 15 คน
3. กำหนดช่วงอายุของอาสาสมัครให้เป็นระหว่าง 18 ถึง 25 ปี
4. เริ่มต้นกระบวนการวัดสัญญาณชีพของอาสาสมัครโดยใช้เครื่องมือที่เกี่ยวข้อง เช่น อุปกรณ์วัดชีพสัญญาณหัวใจ
5. ทำการวัดสัญญาณชีพที่แต่ละอาสาสมัครในกลุ่ม โดยบันทึกผลลัพธ์เช่น อัตราการเต้นของหัวใจ ความดันโลหิต เป็นต้น
6. เก็บข้อมูลสัญญาณชีพที่วัดได้จากทุกคนในกลุ่มอาสาสมัคร
7. นำข้อมูลสัญญาณชีพที่ได้มาใช้ในการกระบวนการพัฒนาโมเดล โดยอาจต้องใช้เวลาในการอธิบายแบบเชิงวิชาการเพื่อให้เห็นภาพรวมของกระบวนการวัดสัญญาณชีพ และการเลือกอาสาสมัครให้ชัดเจนยิ่งขึ้น



รูปที่ 3.3 ลักษณะการเก็บข้อมูลสัญญาณชีพ



รูปที่ 3.4 อุปกรณ์ mmWave EVM Kit

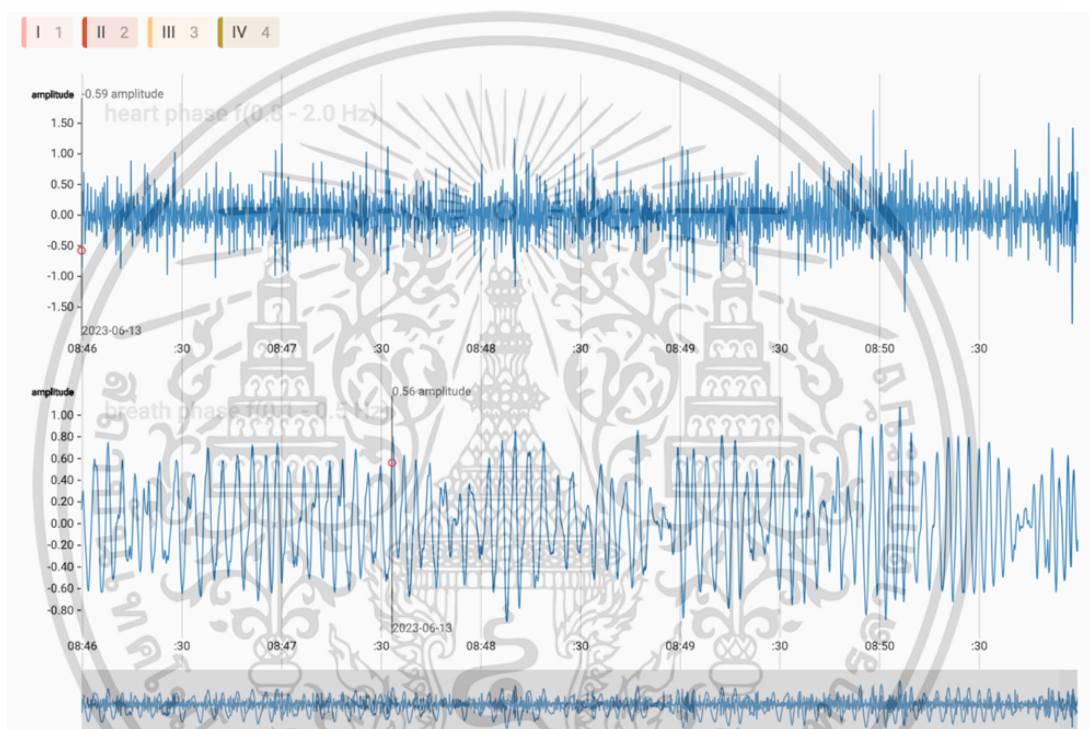


รูปที่ 3.5 อุปกรณ์ BioRadio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

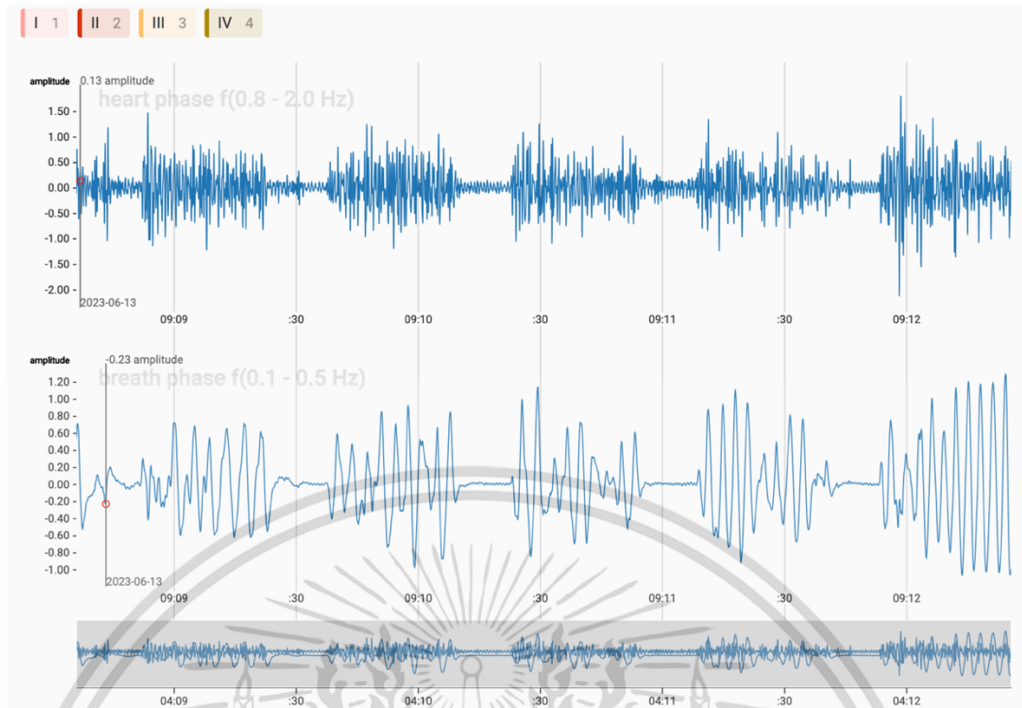
ก่อนการทดลองอาสาสมัครต้องนอนหลับพักผ่อนอย่างน้อย 8 ชั่วโมงก่อนเข้ารับการบันทึกสัญญาณชีพผ่านอุปกรณ์ที่ได้จัดเตรียมไว้ดังรูปที่ 3.4 และ 3.5 เมื่อตั้งค่าเรียบร้อยแล้ว เริ่มต้นขั้นตอนให้อาสาสมัครเก็บข้อมูลเป็น 4 เรือนไข ดังนี้

1. อาสาสมัครหายใจแบบปกติต่อเนื่องและไม่ขยับตัวเป็นเวลา 5 นาที
2. อาสาสมัครหยุดหายใจต่อเนื่องเป็นเวลา 15 วินาที สลับกับหายใจปกติเป็นเวลา 30 วินาที และไม่ขยับตัวจำนวน 5 รอบ
3. อาสาสมัครไม่อยู่ในพื้นที่เก็บสัญญาณชีพ เป็นเวลา 5 นาที
4. อาสาสมัครหายใจแบบปกติและขยับตัวต่อเนื่องเป็นเวลา 5 นาที

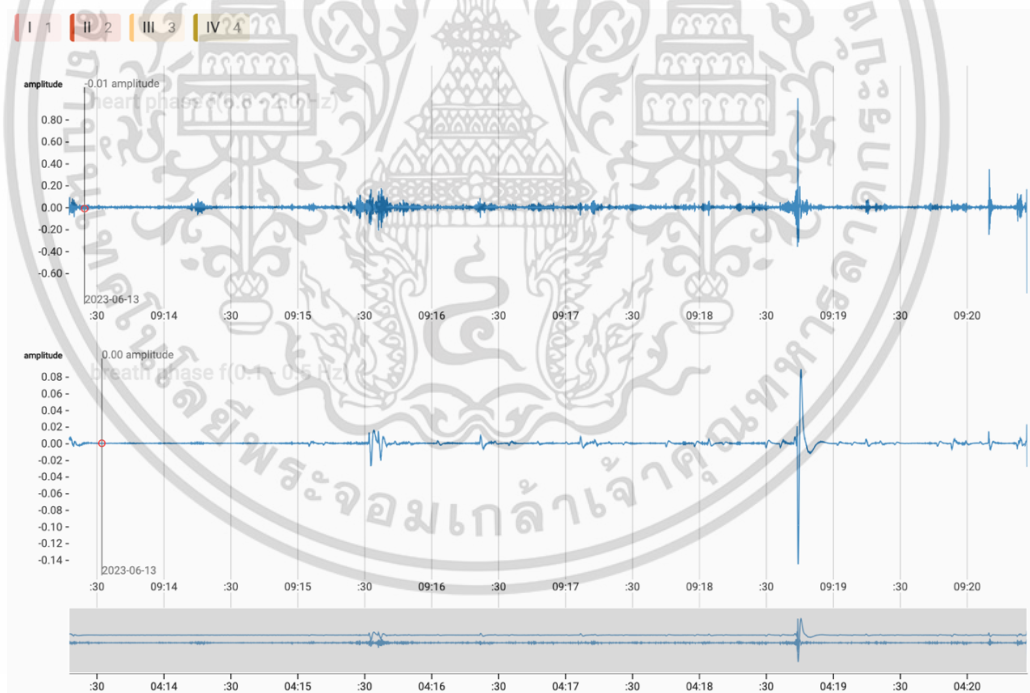


รูปที่ 3.6 อาสาสมัครหายใจแบบปกติต่อเนื่องและไม่ขยับตัวเป็นเวลา 5 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

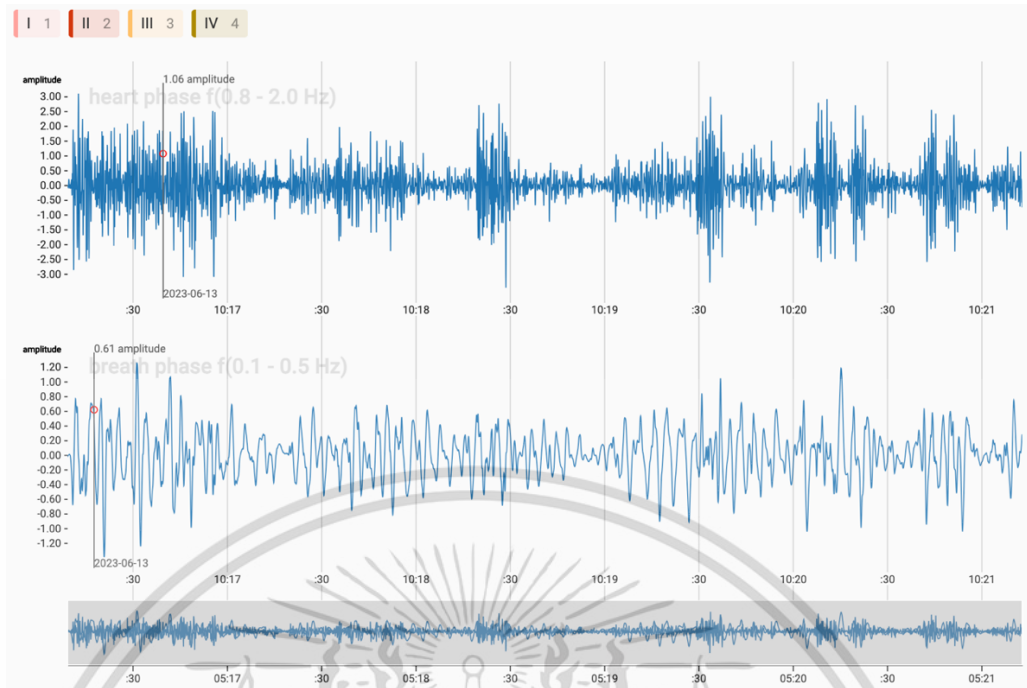


รูปที่ 3.7 อาสาสมัครหยุดหายใจต่อเนื่องเป็นเวลา 15 วินาที สลับกับหายใจปกติเป็นเวลา 1 นาที และไม่ขยับตัว จำนวน 5 รอบ



รูปที่ 3.8 อาสาสมัครไม่อยู่ในพื้นที่เก็บสัญญาณชีพ เป็นเวลา 5 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

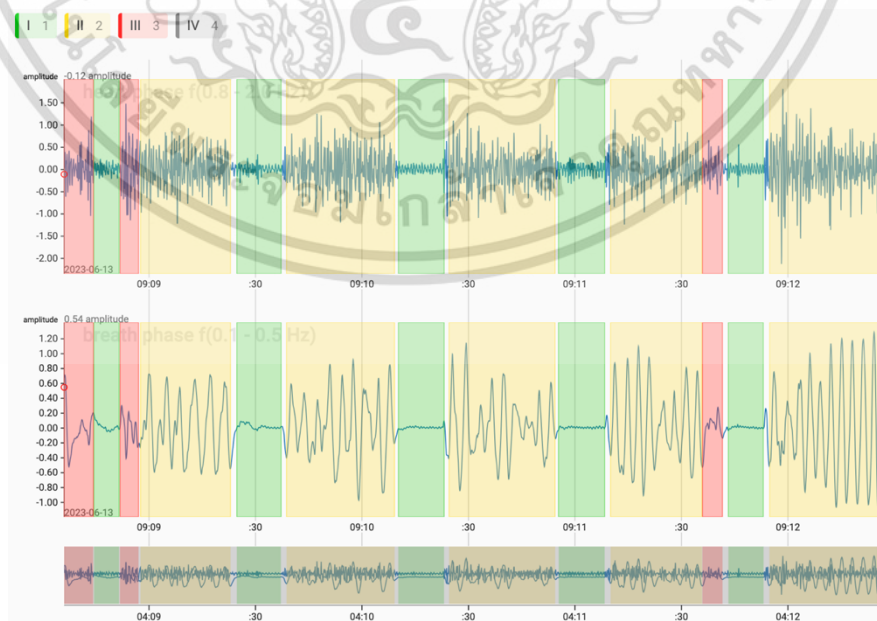


รูปที่ 3.9 อาสาสมัครหายใจแบบปกติและขยับตัวต่อเนื่องเป็นเวลา 5 นาที

3.3 การเตรียมชุดข้อมูล (Data preparing)

Data Preprocessing คือเป็นการเตรียมข้อมูลที่ได้จากการบันทึกหรือข้อมูลดิบตั้งที่ได้กล่าวมาในข้างต้น ให้พร้อมนำไปใช้กับโมเดล เช่น การเลือก Feature หรือ การแปลงข้อความเป็น และเพื่อให้ข้อมูลอยู่ในฟอร์มที่ถูกต้องมากขึ้น และเพื่อความแม่นยำและถูกต้องมากที่สุดของโมเดล โดยจะแยกออกมาได้ 2 หัวข้อ ได้แก่

3.3.1 การทำ Data Labeling



รูปที่ 3.10 การทำ Label ด้วย label studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 สัญลักษณ์สำหรับการ label สัญญาณ		
สัญลักษณ์	สีระบุ label	ความหมาย
I	สีเขียว	ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด
II	สีเหลือง	ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง
III	สีแดง	ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด
IV	สีเทา	ช่วงที่ไม่ต้องวัดสัญญาณ

label studio ความสามารถในการนำสัญญาณมาทำการ Label segment โดยกำหนด label ทั้งหมด 4 class ดังรูปที่ 3.10 โดยใช้สัญลักษณ์สำหรับการ label สัญญาณ ดังตารางที่ 3.1

```

<View>
  <TimeSeriesLabels name="label" toName="ts">
    <Label value="I" background="#FFA39E" fontColor="#FFFFFF"/> // ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด
    <Label value="II" background="#D4380D" fontColor="#FFFFFF"/> // ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง
    <Label value="III" background="#FFC069" fontColor="#000000"/> // ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด
    <Label value="IV" background="#AD8B00" fontColor="#000000"/> // ช่วงที่ไม่ต้องวัดสัญญาณ
  </TimeSeriesLabels>
  <TimeSeries name="ts" valueType="url" value="$timeseriesUrl" sep="," timeColumn="ts" timeFormat="%Y-%m-%d %H:%M:%S.%f"
  timeDisplayFormat="%Y-%m-%d" overviewChannels="non-contact">
    <Channel column="heart_phase" units="amplitude" displayFormat=".,2f" strokeColor="#1f77b4" legend="heart phase"/>
    <Channel column="breath_phase" units="amplitude" displayFormat=".,2f" strokeColor="#1f77b4" legend="breath phase"/>
  </TimeSeries>
</View>

```

รูปที่ 3.11 time series classification templates สำหรับ label studio

โดยผู้วิจัยได้นำออกข้อมูลจาก label studio ในรูปแบบ json ดังรูปที่ 3.12 เพื่อนำข้อมูลไปใช้งานต่อในขั้นตอนการเตรียมข้อมูลสำหรับเทรนโมเดล

```

{
  "timeseriesUrl": "อาสาสมัครสุขภาพใจต่อเมืองเป็น  
เวลา15วินาทีติดต่อกับหัวใจปกติเป็นเวลา30นาทีและในขณะจำนวน5รอบ-  
ญ-1.csv",
  "id": 2585,
  "label": [
    {
      "start": "2023-06-13 21:08:44.645000",
      "end": "2023-06-13 21:08:51.696000",
      "instant": false,
      "timeserieslabels": [
        "I"
      ]
    },
    {
      "start": "2023-06-13 21:08:51.945000",
      "end": "2023-06-13 21:08:57.196000",
      "instant": false,
      "timeserieslabels": [
        "III"
      ]
    },
    {
      "start": "2023-06-13 21:09:24.897000",
      "end": "2023-06-13 21:09:37.297000",
      "instant": false,
      "timeserieslabels": [
        "I"
      ]
    }
  ]
}

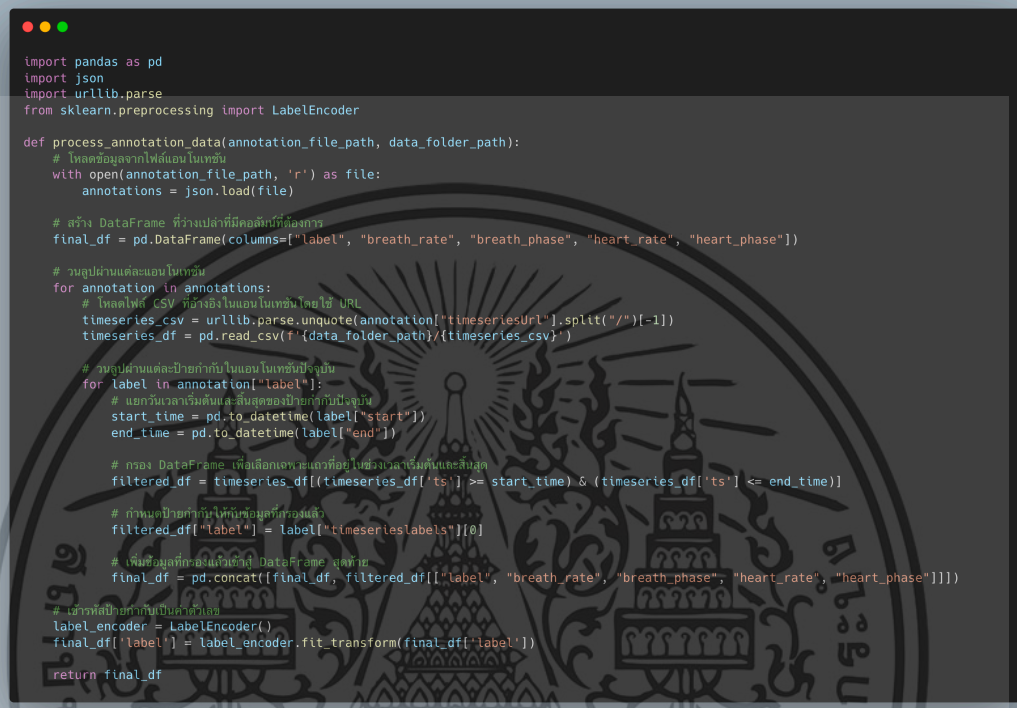
```

รูปที่ 3.12 ผลลัพธ์จากการ export ข้อมูลจาก label studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การเตรียมข้อมูลสำหรับเทรนโมเดล

เมื่อได้ทำงาน segment ข้อมูลด้วย label studio แล้วเราจะได้ไฟล์ json ดังรูปที่ 3.12 มาดังนั้นเราจะทำการสร้างตารางใหม่ขึ้นมา เพื่อสำหรับการเทรนโมเดล โดยใช้โค้ด ดังรูปที่ 3.13 ซึ่งผลลัพธ์จะได้ดังตารางที่ 3.2



```
import pandas as pd
import json
import urllib.parse
from sklearn.preprocessing import LabelEncoder

def process_annotation_data(annotation_file_path, data_folder_path):
    # โหลดข้อมูลจากไฟล์แอนโนเทชัน
    with open(annotation_file_path, 'r') as file:
        annotations = json.load(file)

    # สร้าง DataFrame ที่ว่างเปล่าที่มีคอลัมน์ที่ต้องการ
    final_df = pd.DataFrame(columns=["label", "breath_rate", "breath_phase", "heart_rate", "heart_phase"])

    # วนลูปผ่านแต่ละแอนโนเทชัน
    for annotation in annotations:
        # โหลดไฟล์ CSV ที่อ้างอิงในแอนโนเทชันโดยใช้ URL
        timeseries_csv = urllib.parse.unquote(annotation["timeseriesURL"].split("/")[-1])
        timeseries_df = pd.read_csv(f'{data_folder_path}/{timeseries_csv}')

        # วนลูปผ่านแต่ละป้ายกำกับในแอนโนเทชันปัจจุบัน
        for label in annotation["label"]:
            # แยกวันเวลาเริ่มต้นและสิ้นสุดของป้ายกำกับปัจจุบัน
            start_time = pd.to_datetime(label["start"])
            end_time = pd.to_datetime(label["end"])

            # กรอง DataFrame เพื่อเลือกเฉพาะแถวที่อยู่ในช่วงเวลาที่เริ่มและสิ้นสุด
            filtered_df = timeseries_df[(timeseries_df["ts"] >= start_time) & (timeseries_df["ts"] <= end_time)]

            # กำหนดป้ายกำกับใหม่กับข้อมูลที่กรองแล้ว
            filtered_df["label"] = label["timeseries_labels"][0]

            # เพิ่มข้อมูลที่กรองแล้วเข้า DataFrame สุดท้าย
            final_df = pd.concat([final_df, filtered_df[["label", "breath_rate", "breath_phase", "heart_rate", "heart_phase"]]])

    # เติบโตป้ายกำกับเป็นค่าตัวเลข
    label_encoder = LabelEncoder()
    final_df["label"] = label_encoder.fit_transform(final_df["label"])

    return final_df
```

รูปที่ 3.13 โค้ดสำหรับการจัดเตรียมข้อมูลจากไฟล์ annotation

ตารางที่ 3.2 ผลลัพธ์จากโค้ดสำหรับการจัดเตรียมข้อมูลจากไฟล์ annotation

index	heart_phase	breath_phase	label
0	-0.588177800178528	0.150319695472717	1
1	-0.156419858336449	0.123810410499573	1
2	0.50360494852066	0.119762778282166	1
.	.	.	.
.	.	.	.
.	.	.	.
23183	-0.037074	-0.110262	2
23184	-0.017523	-0.103980	2
23185	-0.051659	-0.099740	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import numpy as np
from tensorflow.keras.utils import to_categorical

def create_sequence_data(dataframe, sequence_length):
    sequences = []
    labels = []

    for i in range(sequence_length, len(dataframe)):
        sequences.append(dataframe.iloc[i-sequence_length:i, :-1].values)
        labels.append(dataframe.iloc[i, :-1])

    input_data = np.array(sequences)
    target_data = to_categorical(np.array(labels))

    return input_data, target_data

# กำหนดค่าคงที่
SEQUENCE_LENGTH = 30

# สร้างข้อมูลลำดับ
input_data, target_data = create_sequence_data(dataframe, SEQUENCE_LENGTH)
```

รูปที่ 3.14 โค้ดสำหรับสร้างข้อมูลลำดับและป้ายกำกับ

โดยในกระบวนการเตรียมข้อมูลสำหรับเทรนโมเดล เราได้สร้างข้อมูลลำดับและป้ายกำกับโดยใช้โค้ด ดังรูปที่ 3.14 โดยผู้วิจัยได้ใช้ทั้งหมด 2 ทีเจอร์ในการฝึกสอนโมเดล ได้แก่ h คือ สัญญาณการเต้นของหัวใจในย่านความถี่ 0.8 – 2.0 Hz และสัญญาณของการหายใจในย่านความถี่ 0.1 - 0.5 Hz โดยมีทั้งหมด 6 คลาส ดังตารางที่ 3.3

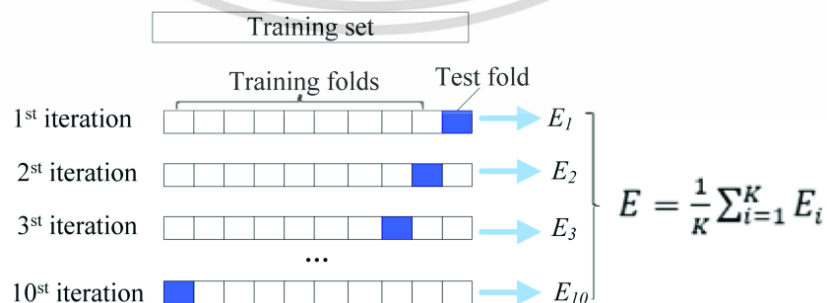
ตารางที่ 3.3 ตัวอย่างข้อมูลจำนวน 1 Batch สำหรับใช้ฝึกสอนโมเดล

sequences					labels			
h_1	h_2	...	h_{29}	h_{30}	y_1	y_2	y_3	y_4
b_1	b_2	...	b_{29}	b_{30}				

3.4 การฝึกสอนโมเดล BiLSTM, BiRNN, BiGRU Classification

3.4.1 การประยุกต์ใช้งาน KFold

ในการเก็บข้อมูลเพื่อฝึกสอนโมเดลเราพบปัญหาอยู่สองประเด็น คือ (1) เรามีข้อมูลสอนน้อย และ (2) ในข้อมูลทีน้อยนั้นเรายังมี Imbalance Data ดังนั้นเราจึงจำเป็นต้องออกแบบกระบวนการวัด performance ด้วย K-Folds โดยกำหนดค่าเท่ากับ 5 ครั้ง



รูปที่ 3.15 แบบจำลอง KFold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การฝึกสอนโมเดล

BiLSTM หรือ Bidirectional Long Short-Term Memory นั้นเป็นทฤษฎีแบบเชิงความลึกหรือ Deep Learning ที่มีประสิทธิภาพสูงในการทำงานที่เกี่ยวข้องกับข้อมูลแบบเวลาหรือ Sequence ดังนั้น มีหลายเหตุผลที่คนเลือกใช้ BiLSTM สำหรับการทำให้ classification ดังนี้

1. **ทำงานด้วยข้อมูลเวลา (Sequential Data):** BiLSTM ถูกออกแบบมาเพื่อทำงานร่วมกับข้อมูลแบบ sequence อย่างการพูดคุย, ข้อความ, เสียง, หรือข้อมูลที่มีลำดับเวลาอื่น ๆ ซึ่งในการทำให้ classification บางครั้ง ข้อมูลลำดับหรือความสัมพันธ์ที่มีต่อเวลาอาจสำคัญ

2. **รับรู้ความสัมพันธ์ระยะยาว (Long-Term Dependencies):** LSTM สามารถรับรู้ความสัมพันธ์ระยะยาวในข้อมูลลำดับ ดังนั้น BiLSTM ซึ่งเป็นการทำ LSTM ในทางทั้งสอง จึงสามารถรับรู้ความสัมพันธ์ทั้งจากแหล่งข้อมูลที่มาก่อนและหลังจากตำแหน่งปัจจุบัน

3. **ปรับปรุงการรับรู้ความหมายทางบริบท (Contextual Understanding):** ในบริบทที่ต้องการให้ระบบเข้าใจความหมายที่อยู่ภายใต้บริบทที่กว้างขึ้น, BiLSTM สามารถทำให้เราได้ผลดีขึ้นเนื่องจากการทำงานในทางสองทิศทางในข้อมูลลำดับ ทำให้เราได้ความหมายจากบริบทที่กว้างขึ้น

4. **ความสามารถในการทำงานร่วมกับข้อมูลที่มีความไม่แน่นอน (Handling of Uncertain Data):** หากมีความไม่แน่นอนหรือเสียงความผิดพลาดในข้อมูลลำดับ, BiLSTM สามารถช่วยจัดการได้ดีกว่าเทคนิคการเรียนรู้เชิงลึกทั่วไปอื่น ๆ

โดยรวม BiLSTM เหมาะสมสำหรับงานการจำแนกประเภท (classification) โดยเฉพาะอย่างยิ่งสำหรับข้อมูลที่มีลำดับหรือที่ต้องการความรู้ความสัมพันธ์ระหว่างเวลา โดยใช้โค้ดที่สร้างโมเดล BiLSTM classification ดังรูปที่ 3.16 และมีโครงสร้างดังรูปที่ 3.19

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Bidirectional
from tensorflow.keras.optimizers import Adam

def LSTM_MODEL(NUM_CLASSES, NUM_FEATURES, SEQUENCE_LENGTH, LEARNING_RATE):
    # Model definition
    model = Sequential()
    model.add(LSTM(128, activation='tanh', input_shape=(SEQUENCE_LENGTH, NUM_FEATURES)))
    model.add(Dense(units=128, activation='tanh'))
    model.add(Dense(NUM_CLASSES, activation='softmax'))

    # Model compilation
    optimizer = Adam(learning_rate=LEARNING_RATE)
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

รูปที่ 3.16 โค้ดสำหรับสร้างโมเดล BiLSTM Classification

BiRNN หรือ Bidirectional Recurrent Neural Networks คือ โมเดลสมองเทียมแบบเวียนเกิดที่เรียนรู้จากข้อมูลลำดับทั้งในทางไปและทางกลับมา. ในงานการทำให้ classification, BiRNN มีประโยชน์ที่สำคัญดังนี้:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

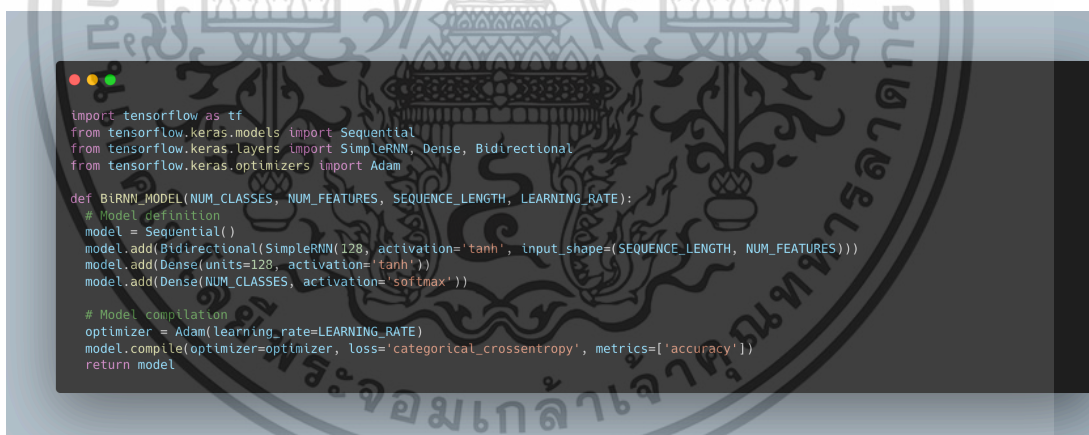
1. **การจัดการข้อมูลลำดับ (Sequential Data Handling):** ด้วยคุณสมบัติสำคัญในการจัดการกับข้อมูลแบบลำดับ, BiRNN กลายเป็นเทคนิคที่ยอดเยี่ยมสำหรับงานที่ต้องมีการแปลความหมายจากข้อมูลลำดับ, เช่น การจำแนกประเภทข้อความ, การแปลภาษา, หรืองานที่เกี่ยวข้องกับเสียง

2. **รับรู้บริบททั้งหมด (Full Contextual Understanding):** BiRNN ทำงานไปในทางทั้งสองของข้อมูลลำดับ, นั่นคือ ในทางที่แล้วและทางถัดไป. ผลลัพธ์คือ, มันสามารถทำความเข้าใจความหมายของบริบทที่ครอบคลุมไปจากทั้งสองทิศทาง

3. **การจำแนกประเภทที่ครอบคลุม (Comprehensive Classification):** ด้วยการทำความเข้าใจบริบททั้งหมด, BiRNN สามารถจำแนกประเภทข้อมูลอย่างรอบคอบ. นั่นคือ, ถ้ามีข้อมูลในบริบทที่จะมีผลต่อการจำแนกประเภท, BiRNN จะมีโอกาสที่ดีในการรับรู้และใช้ข้อมูลเหล่านี้ในการทำสิ่งที่ถูกต้อง

4. **จัดการกับความยาวของลำดับที่ไม่คงที่ (Variable Sequence Length):** BiRNN สามารถรับรู้และปรับตัวเองตามข้อมูลลำดับที่มีความยาวแปรผัน ซึ่งทำให้มันเป็นเทคนิคที่มีประสิทธิภาพในการจัดการกับข้อมูลลำดับของข้อมูลจริงที่มักจะมีมีความยาวที่แตกต่างกัน

โดยรวม, BiRNN เป็นเทคนิคที่มีประสิทธิภาพสำหรับงานการจำแนกประเภท, โดยเฉพาะสำหรับข้อมูลที่มีลำดับหรือความเชื่อมโยงกับเวลา โดยใช้โค้ดที่สร้างโมเดล BiRNN classification ดังรูปที่ 3.17 และมีโครงสร้างดังรูปที่ 3.19



```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense, Bidirectional
from tensorflow.keras.optimizers import Adam

def BiRNN_MODEL(NUM_CLASSES, NUM_FEATURES, SEQUENCE_LENGTH, LEARNING_RATE):
    # Model definition
    model = Sequential()
    model.add(Bidirectional(SimpleRNN(128, activation='tanh', input_shape=(SEQUENCE_LENGTH, NUM_FEATURES))))
    model.add(Dense(units=128, activation='tanh'))
    model.add(Dense(NUM_CLASSES, activation='softmax'))

    # Model compilation
    optimizer = Adam(learning_rate=LEARNING_RATE)
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

รูปที่ 3.17 โค้ดสำหรับสร้างโมเดล BiRNN Classification

BiGRU หรือ Bidirectional Gated Recurrent Units เป็นเทคนิคของโมเดลสมองเทียมแบบเวียนเกิดที่เรียนรู้ข้อมูลลำดับจากทั้งสองทิศทางในงานการทำ classification, BiGRU มีข้อดีอย่างมากมาย ดังนี้:

1. **จัดการข้อมูลลำดับ (Sequential Data Handling):** BiGRU สามารถทำงานได้ดีกับข้อมูลลำดับ เช่น ข้อความ, การพูดคุย, เสียง, หรือข้อมูลที่มีลำดับเวลาอื่น ๆ ทำให้เหมาะสมสำหรับการทำ classification ที่ต้องการทราบความสัมพันธ์เวลาหรือลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **รับรู้บริบททั้งหมด (Full Contextual Understanding):** BiGRU ทำงานไปในทางทั้งสองของข้อมูลลำดับ, นั่นคือ ในทางที่แล้วและทางถัดไป. ผลลัพธ์คือ, มันสามารถทำความเข้าใจความหมายของบริบทที่ครอบคลุมไปจากทั้งสองทิศทาง

3. **การจัดการกับความสัมพันธ์ระยะยาว (Handling of Long-Term Dependencies):** จุดเด่นของ GRU (และดังนั้น BiGRU ด้วย) คือการจัดการกับความสัมพันธ์ระยะยาวในข้อมูลลำดับ. คุณลักษณะนี้ทำให้ GRU และ BiGRU เป็นเทคนิคที่มีประสิทธิภาพในการจัดการกับปัญหาของความจำระยะสั้น (short-term memory) ในโมเดล RNN ดั้งเดิม

4. **ความซับซ้อนของโมเดล (Model Complexity):** GRU มีความซับซ้อนที่น้อยกว่า LSTM (และดังนั้น BiGRU มีความซับซ้อนน้อยกว่า BiLSTM) ทำให้การฝึกฝนโมเดลเร็วขึ้นและต้องการทรัพยากรน้อยกว่า. แต่ยังคงรักษาความสามารถในการจัดการกับความสัมพันธ์ระยะยาว

ดังนั้น, ถ้าเราเลือกใช้ BiGRU ในงานทำ classification, เราสามารถรับประโยชน์จากการรับรู้บริบททั้งหมด, การจัดการกับความสัมพันธ์ระยะยาว, และความซับซ้อนของโมเดลที่ลดลง โดยใช้โค้ดที่สร้างโมเดล BiGRU classification ดังรูปที่ 3.18 และมีโครงสร้างดังรูปที่ 3.19



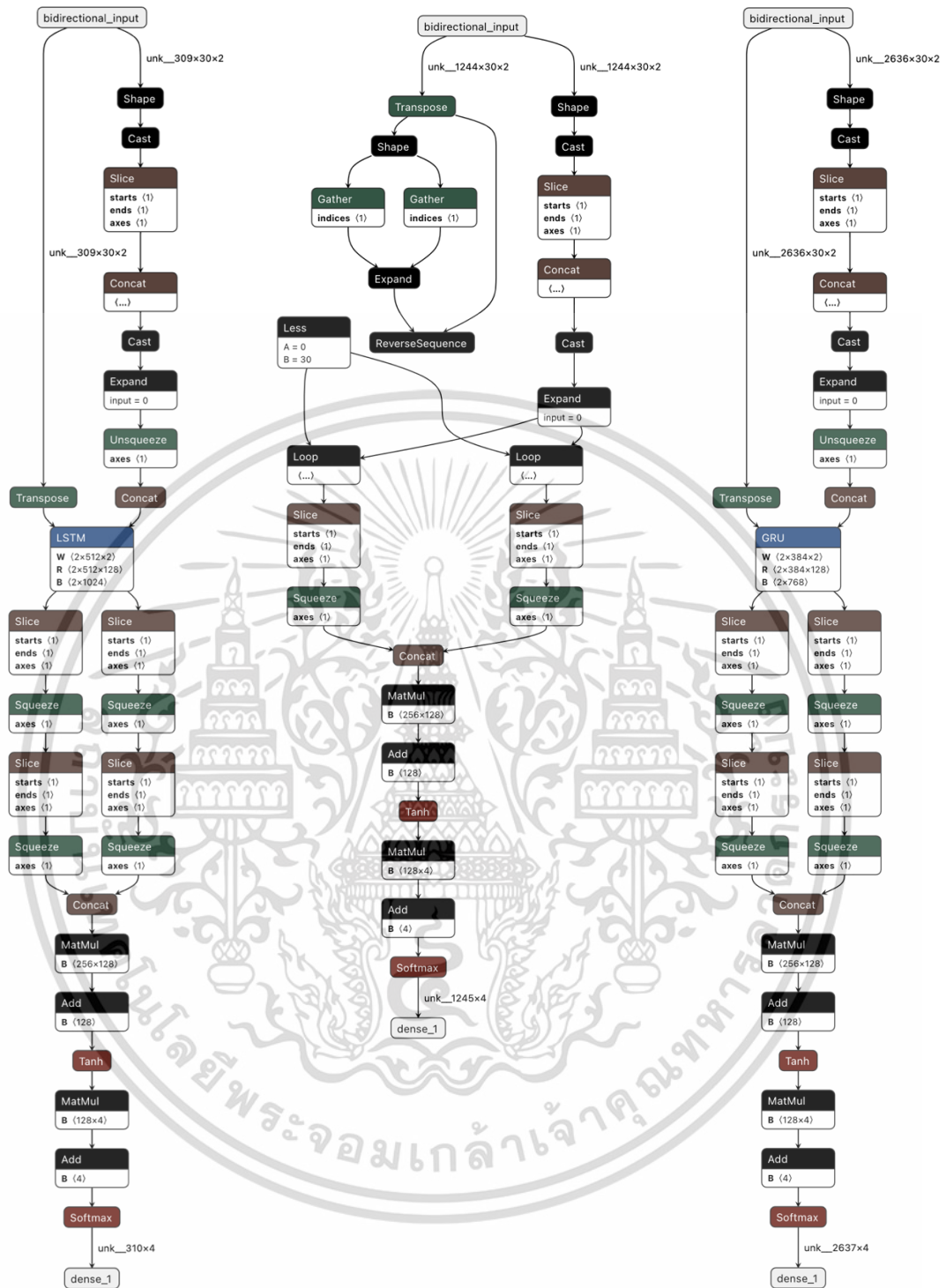
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense, Bidirectional
from tensorflow.keras.optimizers import Adam

def BiRNN_MODEL(NUM_CLASSES, NUM_FEATURES, SEQUENCE_LENGTH, LEARNING_RATE):
    # Model definition
    model = Sequential()
    model.add(Bidirectional(SimpleRNN(128, activation='tanh', input_shape=(SEQUENCE_LENGTH, NUM_FEATURES))))
    model.add(Dense(units=128, activation='tanh'))
    model.add(Dense(NUM_CLASSES, activation='softmax'))

    # Model compilation
    optimizer = Adam(learning_rate=LEARNING_RATE)
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

รูปที่ 3.18 โค้ดสำหรับสร้างโมเดล BiGRU Classification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 โครงสร้างโมเดล BiRNN, BiRNN, BiGRU Classification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การประยุกต์ใช้ Decision Tree เพื่อทำ Ensemble Model

Ensemble Model ที่ใช้ Decision Tree เพื่อรวมผลลัพธ์ของโมเดล BiLSTM, BiRNN, BiGRU Classification ที่ได้จาก K-fold Cross-validation ช่วยเพิ่มประสิทธิภาพในการทำนาย ลดความไม่แน่นอนในการทำนาย และให้ความยืดหยุ่นในการปรับแต่งโมเดล อีกทั้งยังช่วยเพิ่มความเชื่อมั่นในผลลัพธ์ที่ได้ โดยใช้โค้ดดังรูปที่ 3.20 และตัวอย่างกราฟที่ได้จากการเทรน Decision Tree ของโมเดล BiLSTM จำนวน 5 K-fold ดังรูปที่ 3.21

```

import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

# Define constants
SEQUENCE_LENGTH = 20
NUM_FOLDS = 5 # Number of folds for cross-validation

# Load and preprocess data
dataframe = pd.read_csv("../test_sets.csv")[["heart_phase", "breath_phase", "label"]]

# Convert labels to numerical
label_encoder = LabelEncoder()
dataframe["label"] = label_encoder.fit_transform(dataframe["label"])

# Create sequence data
sequences = []
labels = []

for i in range(SEQUENCE_LENGTH, len(dataframe)):
    sequences.append(dataframe.iloc[i-SEQUENCE_LENGTH:i, :-1].values)
    labels.append(dataframe.iloc[i, -1])

# Convert to numpy arrays
input_data = np.array(sequences)
target_data = to_categorical(np.array(labels)) # Perform one-hot encoding for labels

# Perform k-fold cross-validation
kf = KFold(n_splits=NUM_FOLDS, shuffle=True, random_state=42)
all_fold_predictions = []
y_true_test = []

for fold, (train_indices, val_indices) in enumerate(kf.split(input_data)):
    print(f"Fold: {fold+1}")
    train_data, val_data = input_data[train_indices], input_data[val_indices]
    train_target, val_target = target_data[train_indices], target_data[val_indices]

    # Create sequence data for test set
    test_sequences = []
    for i in range(SEQUENCE_LENGTH, len(dataframe)):
        test_sequences.append(dataframe.iloc[i-SEQUENCE_LENGTH:i, :-1].values)
    X_test_seq = np.array(test_sequences)
    y_true_test.append(dataframe["label"].iloc[SEQUENCE_LENGTH:i].values)

    # Save the best model from each fold
    best_model_path = f"../fold_{fold+1}.h5"
    # Load the best model for this fold
    best_model = tf.keras.models.load_model(best_model_path)

    # Predict on the validation set
    y_pred_test = best_model.predict(X_test_seq)
    y_pred_test_classes = np.argmax(y_pred_test, axis=-1)

    # Store the predictions for each fold
    all_fold_predictions.append(y_pred_test_classes)

# Concatenate the true test labels
y_true_test = np.concatenate(y_true_test)

# Ensemble the predictions using a Decision Tree
ensemble_predictions = np.array(all_fold_predictions).T
ensemble_decision_tree = DecisionTreeClassifier()
ensemble_decision_tree.fit(ensemble_predictions, y_true_test)
ensemble_pred_test = ensemble_decision_tree.predict(ensemble_predictions)

# Calculate accuracy of ensemble model
ensemble_accuracy = accuracy_score(y_true_test, ensemble_pred_test)
print("Ensemble Accuracy:", ensemble_accuracy)
    
```

รูปที่ 3.20 โค้ดสำหรับสร้างเทรน Decision Tree



รูปที่ 3.21 กราฟ Decision Tree

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การวัดผล Ensemble Model

การวัดผลด้วย precision, recall, และ F1-score เป็นเครื่องมือที่ใช้ในการประเมินประสิทธิภาพของระบบหรือโมเดลการเรียนรู้ของเครื่องจักร (machine learning) ในงานปัญญาประดิษฐ์ (artificial intelligence) โดยเฉพาะอย่างยิ่งในงานจำแนกประเภท (classification tasks) ซึ่งมักใช้กันอย่างแพร่หลายในการวัดประสิทธิภาพของโมเดลการจำแนกหมวดหมู่ต่าง ๆ โดยใช้โค้ดดังรูปที่ 3.22 และตัวอย่างตารางที่ 4.1 Classification Report ของโมเดล BiLSTM

```

from sklearn.metrics import classification_report

# Calculate predictions for the ensemble model
ensemble_pred_test = ensemble_decision_tree.predict(ensemble_predictions)

# Calculate and print the classification report for the ensemble model
ensemble_report = classification_report(y_true_test, ensemble_pred_test, target_names=label_encoder.classes_, digits=4)
print("Ensemble Model Classification Report:")
print(ensemble_report)

```

รูปที่ 3.22 โค้ดสำหรับวัดผล precision, recall, f1-score

การวัดผลด้วย confusion matrix ในการทำ Ensemble Model BiLSTM classification ด้วย 6 Kfold มีข้อดีที่สำคัญต่อไปนี้:

1. ประสิทธิภาพในการประเมินความแม่นยำ: Confusion matrix ช่วยให้เราสามารถวัดประสิทธิภาพของโมเดลในการจำแนกหมวดหมู่ต่าง ๆ ได้อย่างละเอียด โดยการนับจำนวนตัวอย่างที่ถูกตรวจจับและจำแนกให้ถูกต้อง รวมถึงจำนวนตัวอย่างที่ผิดพลาดในการจำแนกเชิงเสียง (false positives และ false negatives) ที่ช่วยให้เราได้ภาพรวมที่มีประสิทธิภาพในการประเมินโมเดลได้อย่างถูกต้องและเต็มที่
2. การตรวจจับความครอบคลุม: Confusion matrix ช่วยให้เราวัดความครอบคลุมของโมเดลในการตรวจจับตัวอย่างที่เป็นความจริงได้อย่างชัดเจน โดยการนับจำนวนตัวอย่างที่ถูกตรวจจับเทียบกับจำนวนตัวอย่างทั้งหมดที่เป็นความจริง ซึ่งช่วยให้เราได้ข้อมูลที่มีประสิทธิภาพในการประเมินความครอบคลุมของโมเดล
3. การปรับแต่งโมเดล: โดยการวิเคราะห์ confusion matrix เราสามารถระบุประเภทของข้อผิดพลาดที่โมเดลทำได้ เช่น ตรวจพบเทียบกับข้อผิดพลาดที่ไม่ตรงกับความจริง นี้ช่วยให้เราสามารถปรับแต่งและพัฒนาโมเดลได้อย่างมีประสิทธิภาพ
4. การประเมินที่ถูกต้อง: Confusion matrix ช่วยให้เราได้รับข้อมูลที่แม่นยำและรายละเอียดเกี่ยวกับประสิทธิภาพของโมเดล ซึ่งสามารถนำมาเปรียบเทียบกับเป้าหมายหรือค่าที่กำหนดไว้ เพื่อปรับปรุงโมเดลให้ดียิ่งขึ้นในการจำแนก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ confusion matrix ในการวัดผลจะช่วยให้เราได้รับความเข้าใจที่ลึกซึ้งและแม่นยำเกี่ยวกับประสิทธิภาพของโมเดลในการจำแนกและตรวจจับหมวดหมู่ต่าง ๆ ในการใช้งาน Ensemble Model BiLSTM classification ด้วย 6 Kfold โดยใช้โค้ดดังรูปที่ 3.23 และตัวอย่าง Confusion Matrix ของโมเดล BiLSTM Classification ดังรูปที่ 4.1

```

from sklearn.metrics import confusion_matrix

# Calculate confusion matrix for the ensemble model
ensemble_cm = confusion_matrix(y_true_test, ensemble_pred_test)
ensemble_cm = ensemble_cm / ensemble_cm.sum(axis=1, keepdims=True) # Normalize the confusion matrix

# Plot the confusion matrix
plt.figure(figsize=(8, 6))
plt.imshow(ensemble_cm, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Ensemble Model Confusion Matrix')
plt.colorbar()
tick_marks = np.arange(NUM_CLASSES)
plt.xticks(tick_marks, label_encoder.classes_, rotation=45)
plt.yticks(tick_marks, label_encoder.classes_)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')

thresh = ensemble_cm.max() / 2.
for i in range(NUM_CLASSES):
    for j in range(NUM_CLASSES):
        plt.text(j, i, f'{ensemble_cm[i, j]:.2f}', ha='center', va='center', color='white' if ensemble_cm[i, j] > thresh else 'black')

plt.tight_layout()
plt.show()

```

รูปที่ 3.23 โค้ดสำหรับวัดผล Ensemble Model ด้วย confusion matrix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

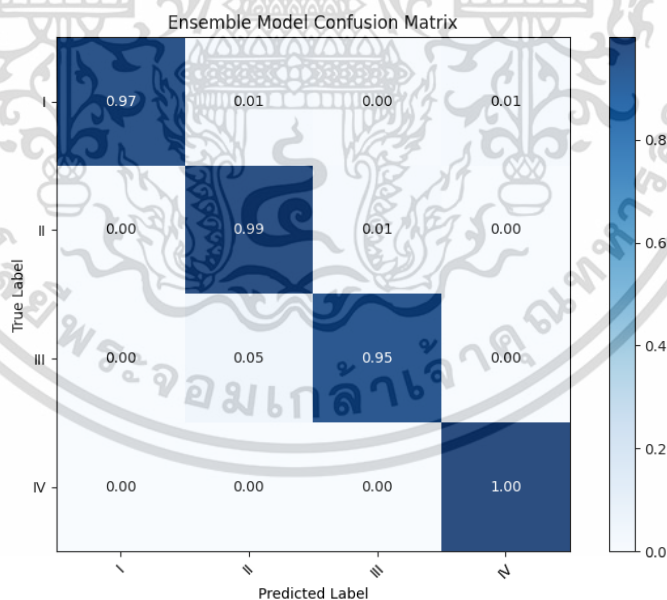
บทที่ 4

ผลการดำเนินงาน

บทนี้จะนำเสนอผลการดำเนินงานซึ่งทำตามรายละเอียดที่ระบุไว้ในระเบียบวิจัยบทที่ 3 ดังนั้น เนื้อหาที่อยู่ในบทนี้จะบรรยายผลที่เกิดจากการพยากรณ์โดยใช้โมเดลทั้งสาม ได้แก่ Bidirectional LSTM Classification, Bidirectional RNN Classification, และ Bidirectional GRU Classification ผลที่ได้จะถูกประเมินหรือวัดค่าความแม่นยำในการจัดหมวดหมู่ ด้วยเทคนิค Confusion matrix เพื่อทำการเปรียบเทียบความแม่นยำ โมเดลทั้งสาม ได้ผลลัพธ์ที่แบ่งออกเป็น 4 สถานะ ได้แก่ I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด), II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง), III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) และ I (ช่วงที่ไม่จำเป็นต้องวัดสัญญาณ) โดยการฝึกสอนโมเดลใช้ข้อมูลทั้งหมดออกเป็น 3 ชุด คือ training set จำนวน 13,894 ชุด, validation set จำนวน 4,632 ชุด และ test set จำนวน 23,158 ชุด

4.1 การประเมินผลลัพธ์โมเดล

การประเมินผลโมเดล คือ ขั้นตอนการวัดประสิทธิภาพการพยากรณ์ของโมเดลเพื่อตรวจสอบความแม่นยำการพยากรณ์ โดยใช้ Confusion matrix เพื่อทดสอบประสิทธิภาพและความแม่นยำการพยากรณ์เอาต์พุตในรูปแบบ classification โดยใช้ข้อมูลสำหรับทดสอบดังที่กล่าวมาข้างต้น



รูปที่ 4.1 Confusion Matrix ของโมเดล BiLSTM Classification

จากรูปที่ 4.1 คือ ผลลัพธ์ของ Confusion Matrix สำหรับโมเดล BiLSTM จำนวนข้อมูลเทรน 13,894 ชุด และใช้ข้อมูลสำหรับทดสอบ 23,158 ชุด พบว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลาส I หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด สามารถทำนายได้ถูกต้อง (1,066) 97 % ชุดจากข้อมูลจริงทั้งหมด 1,104 ชุด
- คลาส II หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง สามารถทำนายได้ถูกต้อง (8,726) 99 % ชุดจากข้อมูลจริงทั้งหมด 8,821 ชุด
- คลาส III หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด สามารถทำนายได้ถูกต้อง (4,411) 95 % ชุดจากข้อมูลจริงทั้งหมด 4,644 ชุด
- คลาส IV หรือ ช่วงที่ไม่ต้องวัดสัญญาณ สามารถทำนายได้ถูกต้อง (8,589) 100 % ชุดจากข้อมูลจริงทั้งหมด 8,589 ชุด

จากผลลัพธ์ Confusion matrix ดังรูปที่ 4.1 สามารถคำนวณประสิทธิภาพในการทำนายของโมเดล โดยใช้ F1 score ซึ่ง F1 score คือการแสดงผลประสิทธิภาพการพยากรณ์ในแต่ละคลาสใช้ Precision กับ Recall เพื่อคำนวณหาค่าเฉลี่ย Harmonic mean ดังตารางที่ (4.1)

ตารางที่ 4.1 : Classification Report ของโมเดล BiLSTM

Evaluate \ Class	I (แม่นยำสูง)	II (แม่นยำปานกลาง)	III (แม่นยำต่ำ)	IV (ไม่ต้องวัดสัญญาณ)
Precision (%)	97.45	97.46	97.23	99.81
Recall (%)	96.83	98.58	95.13	99.87
F1-score (%)	97.14	98.02	96.17	99.84
Support (%)	1,104	8,821	4,644	8,589
Weighted avg (%)	98.28			

จากตารางที่ 4.1 สรุปโดยโฟกัสที่กลุ่ม I และ II ซึ่งเป็นกลุ่มหลักในการวิเคราะห์ผลของ ensemble model ในงานวิจัยนี้ ได้ผลการวัดประสิทธิภาพดังนี้:

กลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด):

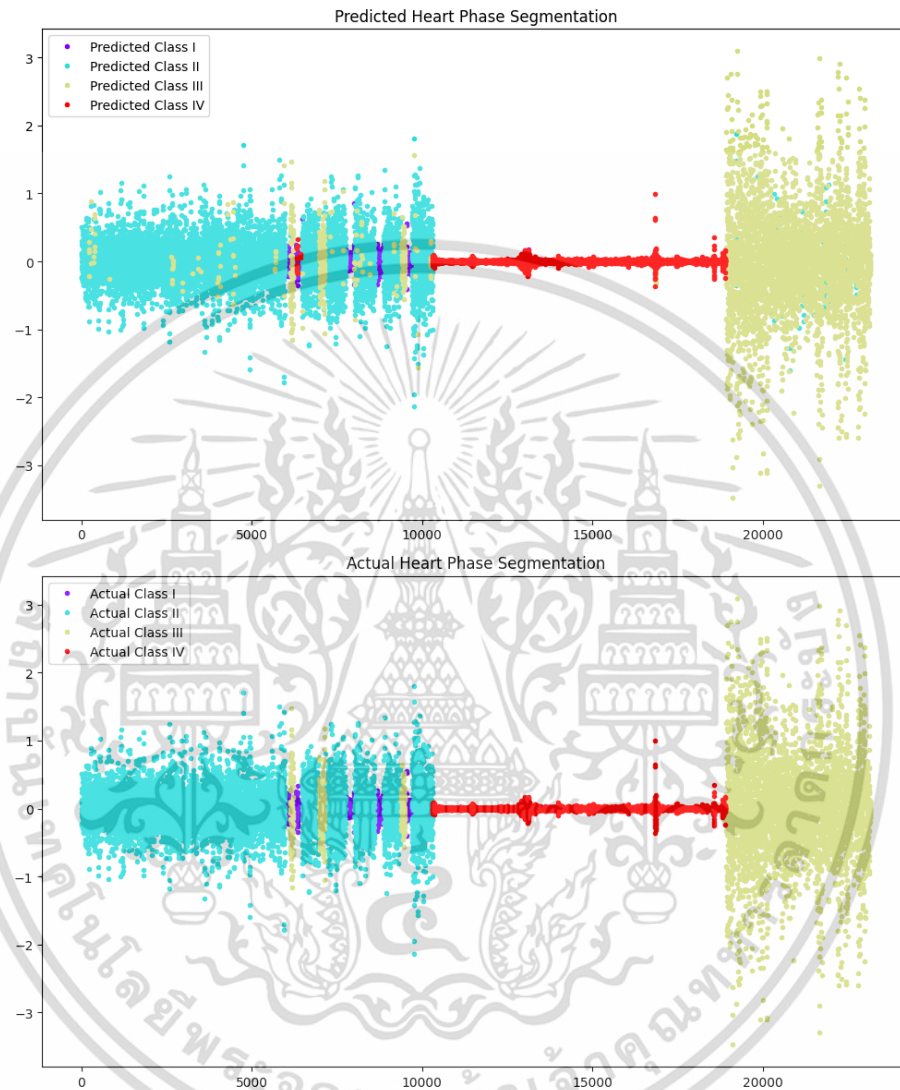
- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 97.45%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 96.83%
- ค่า F1-score: มีค่าประมาณ 97.14 ซึ่งเป็นค่าเฉลี่ย harmonic mean

กลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง):

- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 97.46%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 98.58%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

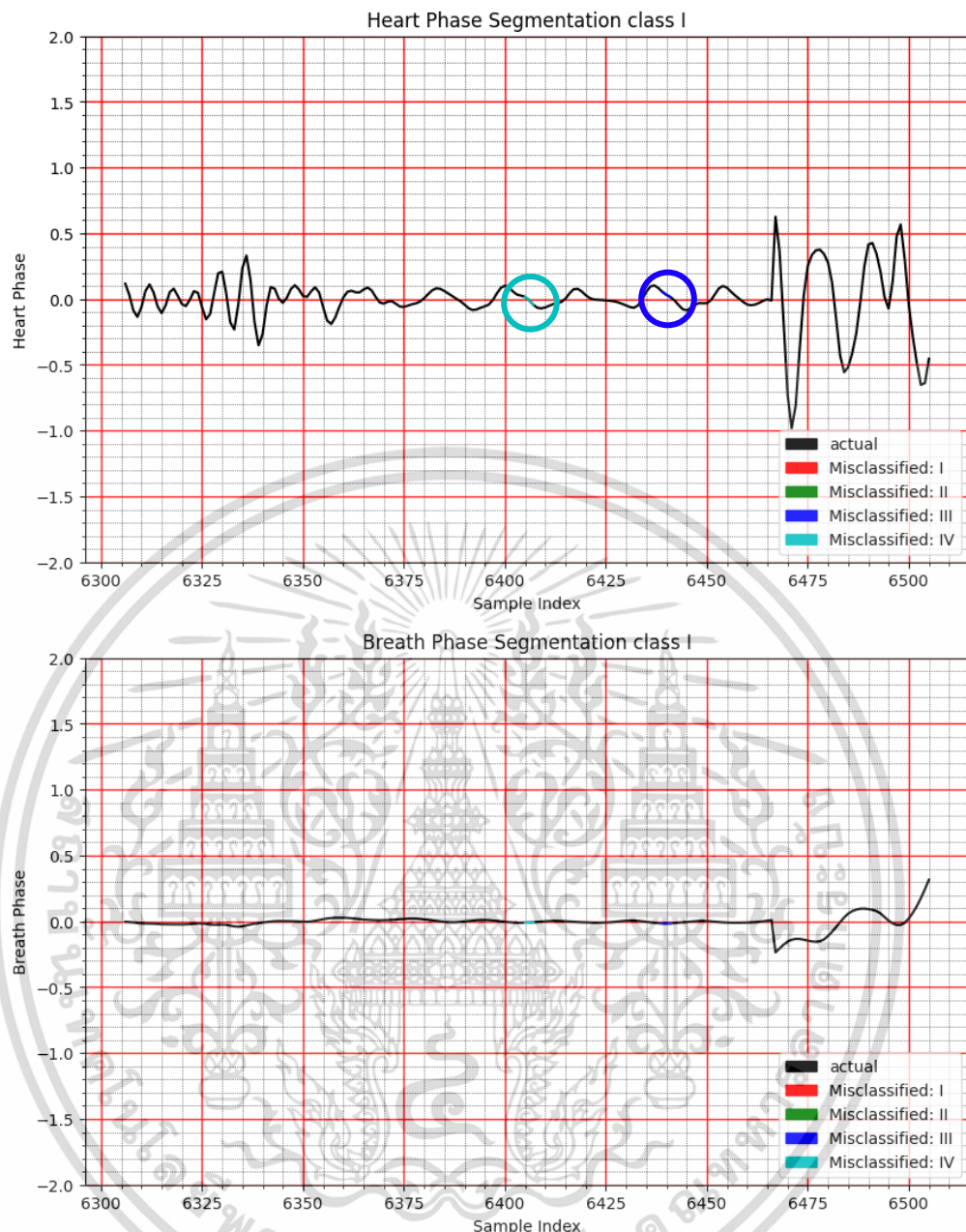
- ค่า F1-score: มีค่าประมาณ 98.02 ซึ่งเป็นค่าเฉลี่ย harmonic mean จากผลการวัดประสิทธิภาพของโมเดลสำหรับกลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด) และกลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง) สรุปได้ว่าโมเดล BiLSTM สามารถแยกแยะสัญญาณกลุ่มที่ I และ II ได้ใกล้เคียงกันและอยู่ในเกณฑ์ที่สูง และเชื่อถือได้



รูปที่ 4.2 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiLSTM Classification

จากรูปที่ 4.2 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือกราฟที่แสดงผลลัพธ์ของโมเดล และกราฟแสดงผลลัพธ์จาก label ทั้ง 4 class จะพบว่าในกรอบสีดำ คือตัวอย่างที่โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้อง คือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

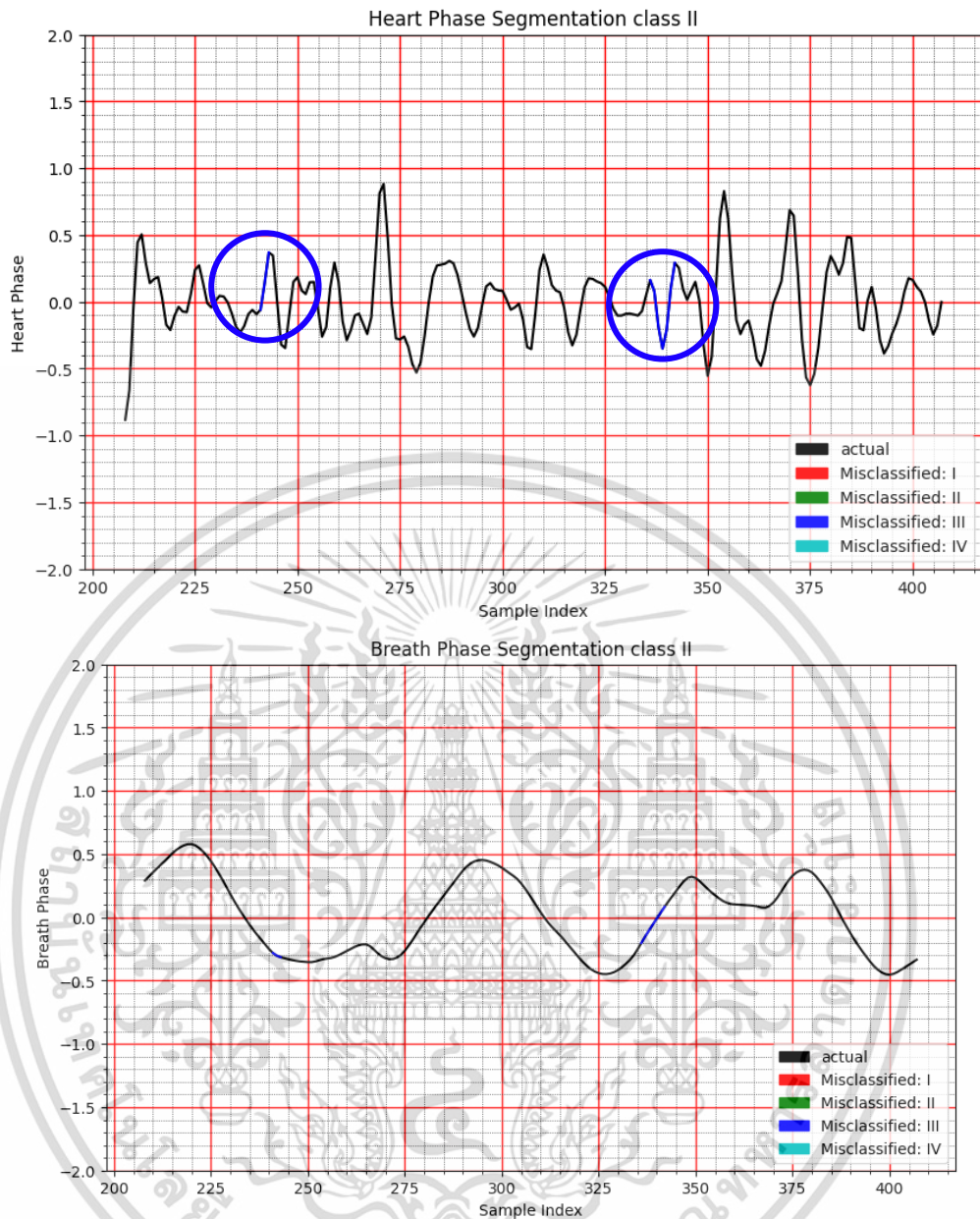
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiLSTM Classification

จากรูปที่ 4.3 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือกราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.3 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีฟ้า โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class IV (ช่วงที่ไม่ต้องวัดสัญญาณ) แต่ผลที่ถูกต้องคือ class I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด) และกรอบสีน้ำเงิน โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด)

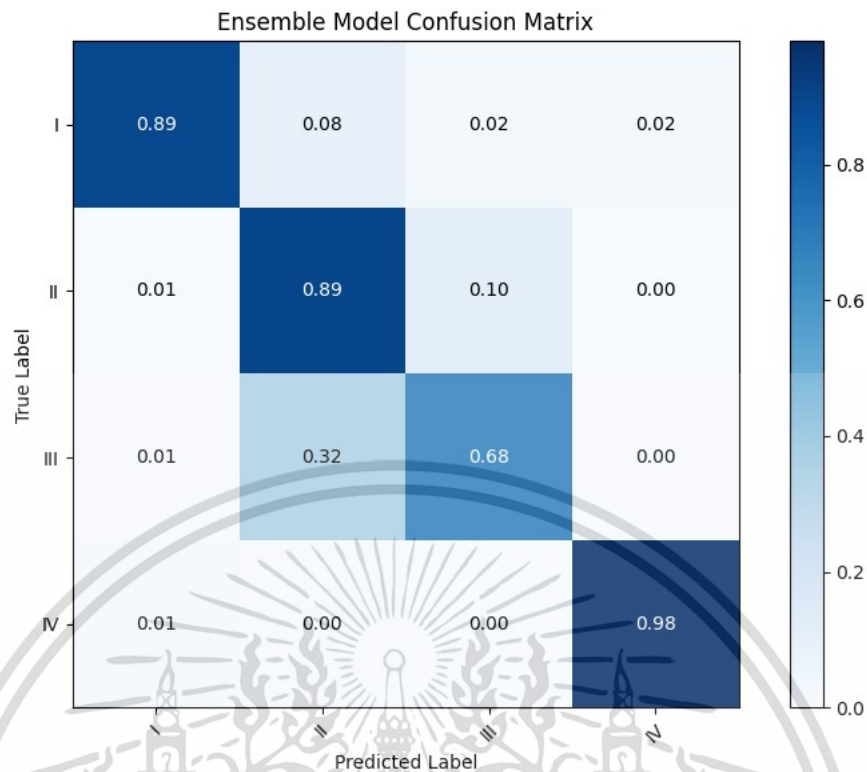
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiLSTM Classification

จากรูปที่ 4.4 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือกราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.4 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีน้ำเงิน โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Confusion Matrix ของโมเดล BiRNN Classification

จากรูปที่ 4.5 ผลลัพธ์ของ Confusion matrix สำหรับโมเดล BiRNN จำนวนข้อมูลเทรน 13,894 ชุด และใช้ข้อมูลสำหรับทดสอบ 23,158 ชุด พบว่า

- คลาส I หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด สามารถทำนายได้ถูกต้อง (1,059) 89 % ชุดจากข้อมูลจริงทั้งหมด 1104 ชุด
- คลาส II หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง สามารถทำนายได้ถูกต้อง (8,085) 89 % ชุดจากข้อมูลจริงทั้งหมด 8821 ชุด
- คลาส III หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด สามารถทำนายได้ถูกต้อง (3,622) 68 % ชุดจากข้อมูลจริงทั้งหมด 4644 ชุด
- คลาส IV หรือ ช่วงที่ไม่ต้องวัดสัญญาณ สามารถทำนายได้ถูกต้อง (8,584) 98 % ชุดจากข้อมูลจริงทั้งหมด 8589 ชุด

จากผลลัพธ์ Confusion matrix ดังรูปที่ 4.5 สามารถคำนวณประสิทธิภาพในการทำนายของโมเดล โดยใช้ F1 score ซึ่ง F1 score คือการแสดงผลประสิทธิภาพการพยากรณ์ในแต่ละคลาสใช้ Precision กับ Recall เพื่อคำนวณหาค่าเฉลี่ย Harmonic mean ดังตารางที่ (4.2)

ตารางที่ 4.2 : F1-score และค่าเฉลี่ย F1-score หรือค่าความแม่นยำในการพยากรณ์เฉลี่ยรวมทุกคลาส ของโมเดล BiRNN

Class \ Evaluate	I (แม่นยำสูง)	II (แม่นยำปานกลาง)	III (แม่นยำต่ำ)	IV (ไม่ต้องวัดสัญญาณ)
Precision (%)	83.75	83.15	77.70	99.74
Recall (%)	88.68	89.48	67.66	98.16
F1-score (%)	86.14	86.20	72.33	98.94
Support (%)	1104	8821	4644	8589
Weighted avg (%)	88.14			

จากตารางที่ 4.2 สรุปโดยโฟกัสที่กลุ่ม I และ II ซึ่งเป็นกลุ่มหลักในการวิเคราะห์ผลของ ensemble model ในงานวิจัยนี้ ได้ผลการวัดประสิทธิภาพดังนี้:

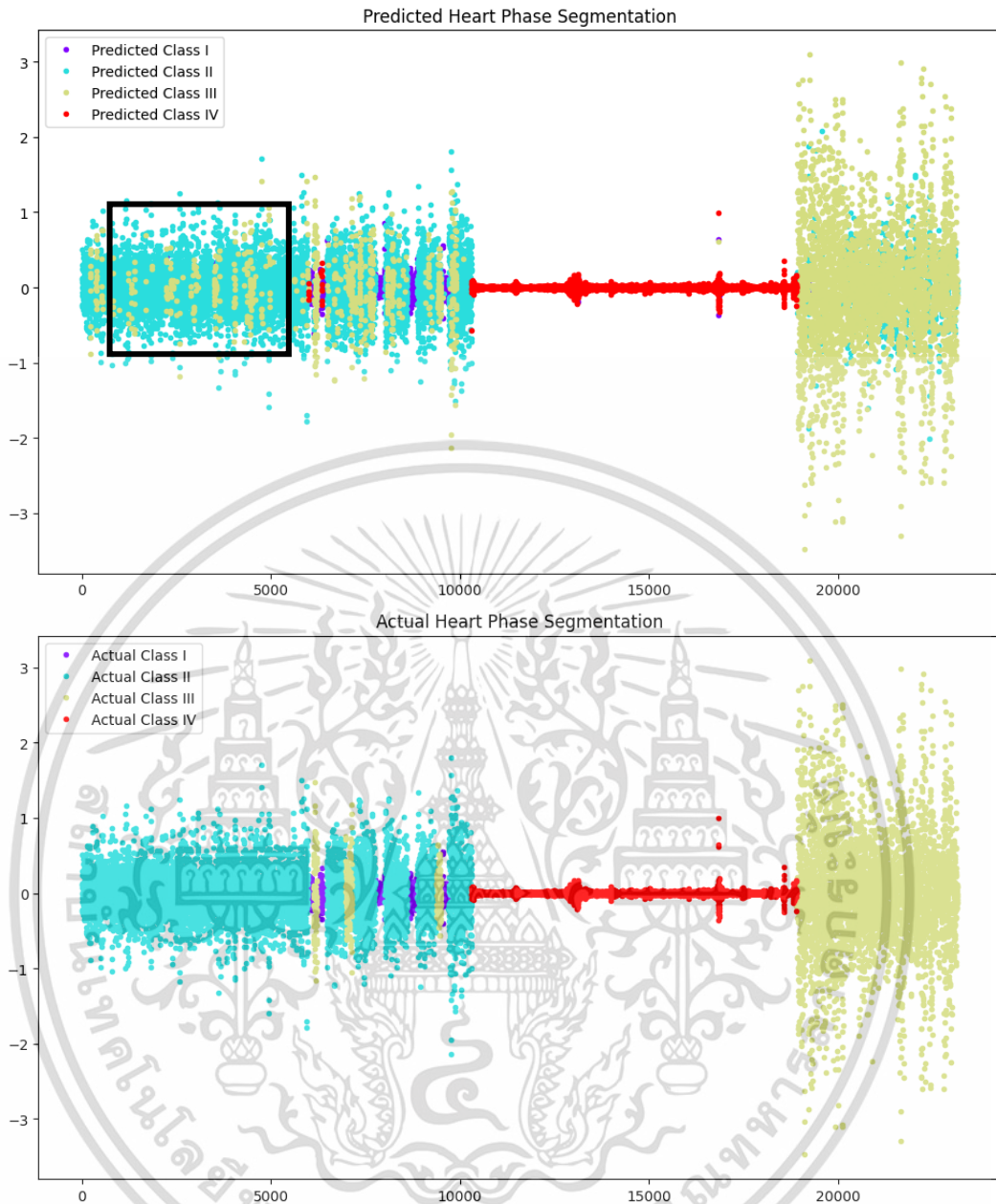
กลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด):

- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 83.75%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 88.68%
- ค่า F1-score: มีค่าประมาณ 86.14 ซึ่งเป็นค่าเฉลี่ย harmonic mean

กลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง):

- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 83.15%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 89.48%
- ค่า F1-score: มีค่าประมาณ 86.20 ซึ่งเป็นค่าเฉลี่ย harmonic mean

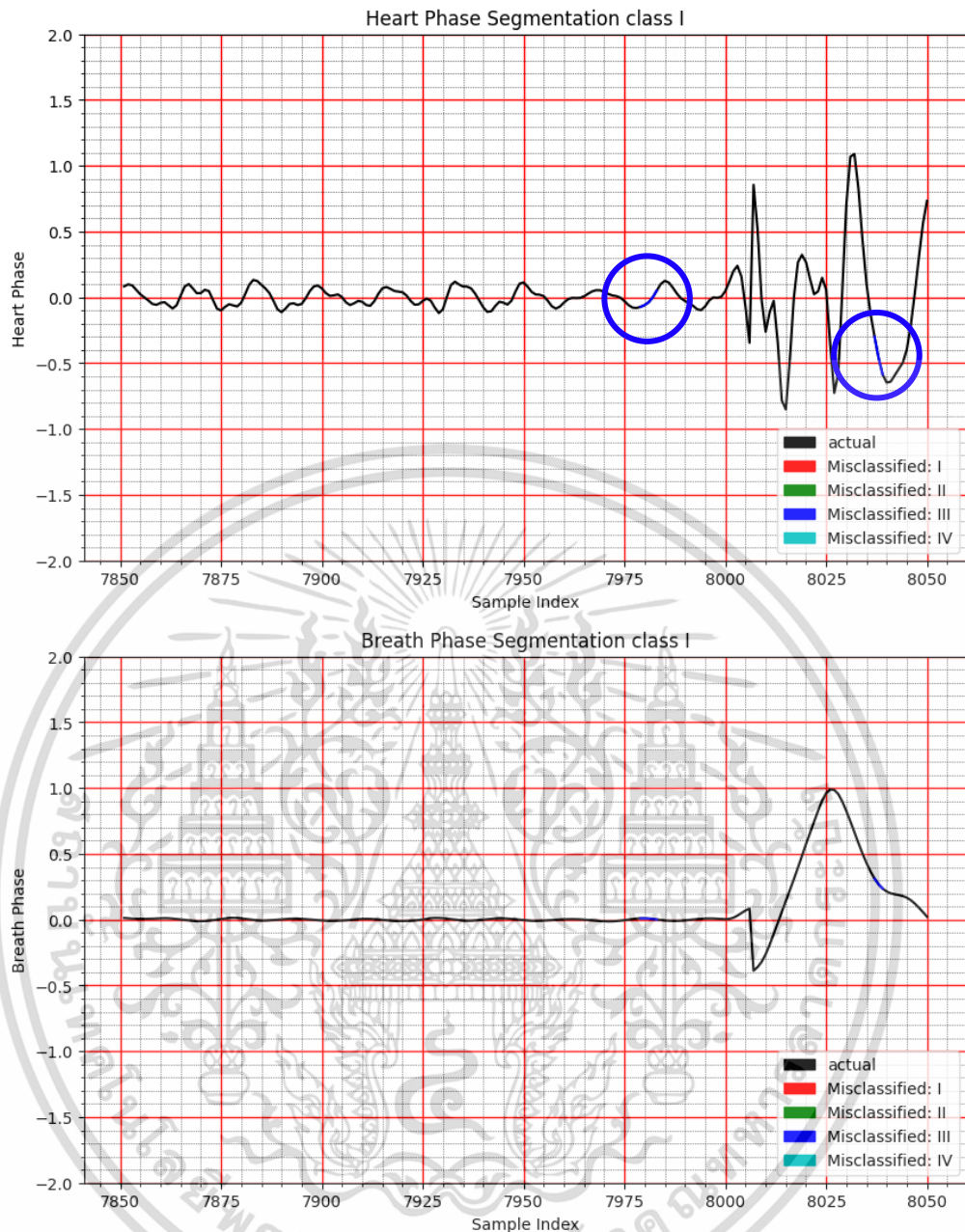
จากผลการวัดประสิทธิภาพของโมเดลสำหรับกลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด) และกลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง) สรุปได้ว่าโมเดล BiRNN สามารถแยกแยะสัญญาณกลุ่มที่ I ได้ไม่ดีกว่า กลุ่มที่ II -0.6%



รูปที่ 4.6 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiRNN Classification

จากรูปที่ 4.6 ผลทดสอบกับสัญญาณย่านความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือกราฟที่แสดงผลลัพธ์ของโมเดล และกราฟแสดงผลลัพธ์จาก label ทั้ง 4 class จะพบว่าในกรอบสีดำ คือตัวอย่างที่โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้อง คือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

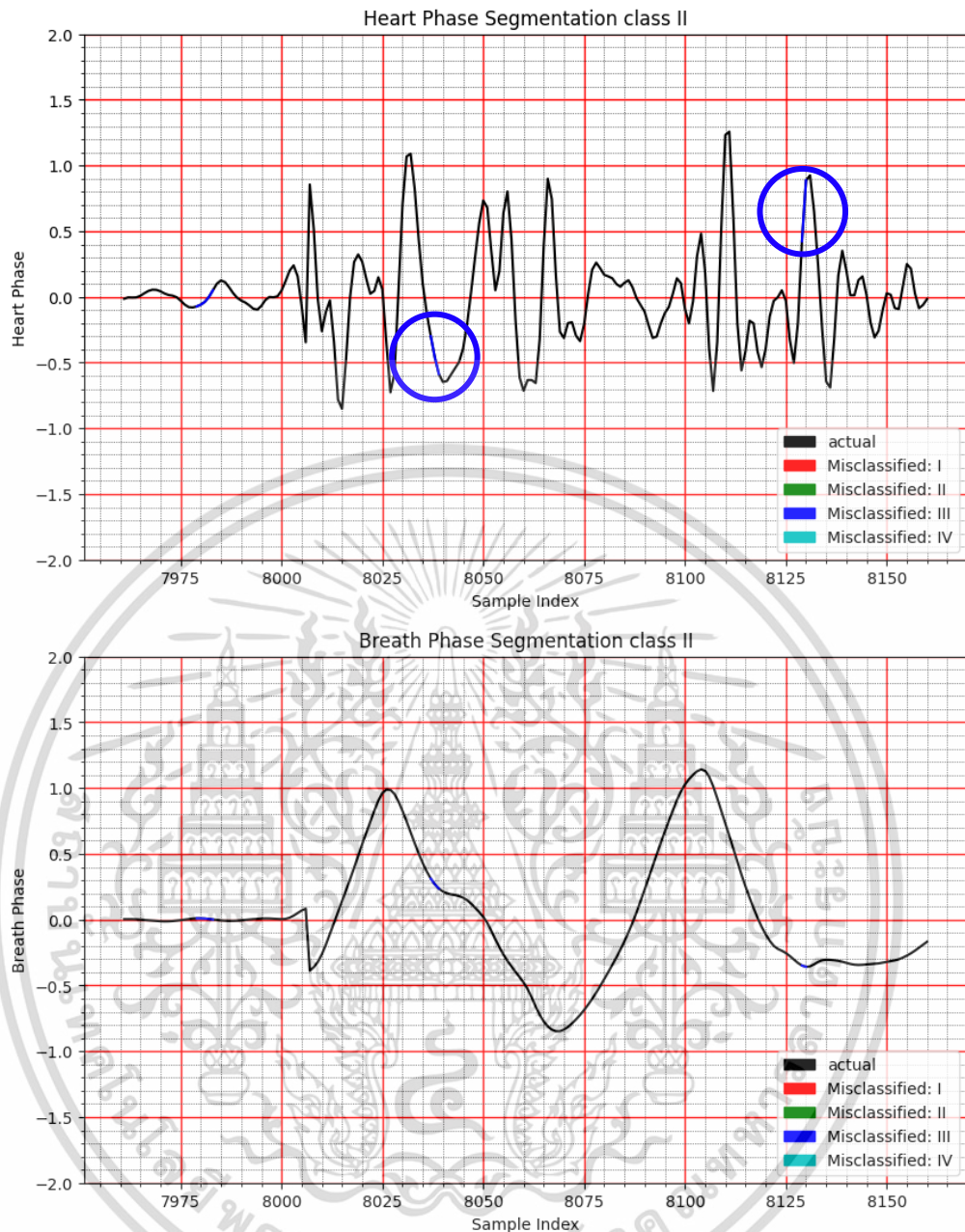
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiRNN Classification

จากรูปที่ 4.7 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiRNN Classification จะประกอบด้วย 2 กราฟ คือกราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.4 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีน้ำเงิน โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด)

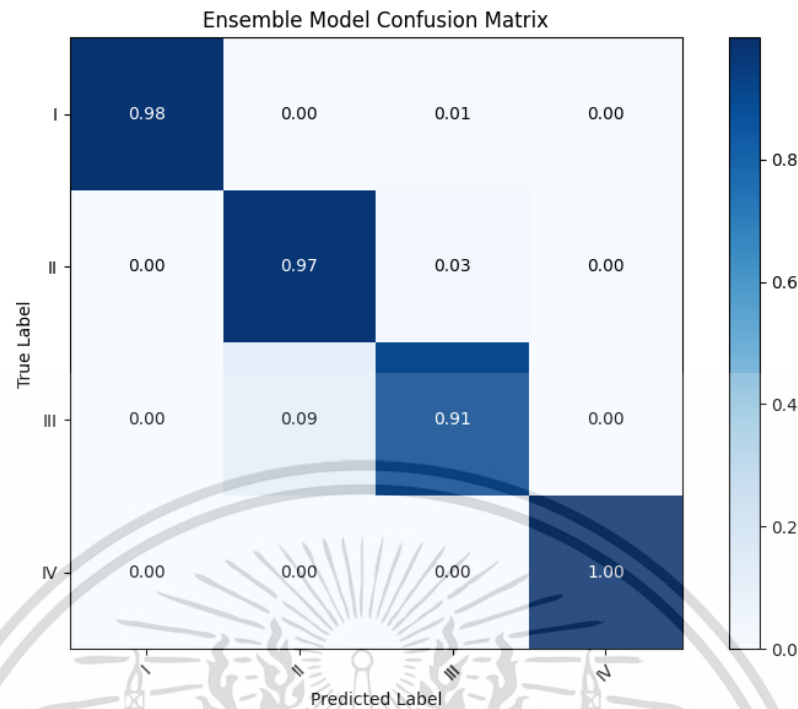
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiRNN Classification

จากรูปที่ 4.8 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือกราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.4 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีน้ำเงิน โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 Confusion Matrix ของโมเดล BiGRU Classification

จากรูปที่ 4.9 คือผลลัพธ์ของ Confusion matrix สำหรับโมเดล BiGRU จำนวนข้อมูลเทรน 13,894 ชุด และใช้ข้อมูลสำหรับทดสอบ 23,158 ชุด พบว่า

- คลาส I หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด สามารถทำนายได้ถูกต้อง (1,081) 98 % ชุดจากข้อมูลจริงทั้งหมด 1104 ชุด
- คลาส II หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง สามารถทำนายได้ถูกต้อง (8,532) 97 % ชุดจากข้อมูลจริงทั้งหมด 8821 ชุด
- คลาส III หรือ ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด สามารถทำนายได้ถูกต้อง (4,214) 91 % ชุดจากข้อมูลจริงทั้งหมด 4644 ชุด
- คลาส IV หรือ ช่วงที่ไม่ต้องวัดสัญญาณ สามารถทำนายได้ถูกต้อง 100 % ชุดจาก (8,589) ข้อมูลจริงทั้งหมด 8589 ชุด

จากผลลัพธ์ Confusion matrix ดังรูปที่ 4.9 สามารถคำนวณประสิทธิภาพในการทำนายของโมเดล โดยใช้ F1 score ซึ่ง F1 score คือการแสดงผลประสิทธิภาพการพยากรณ์ในแต่ละคลาสใช้ Precision กับ Recall เพื่อคำนวณหาค่าเฉลี่ย Harmonic mean ดังตารางที่ (4.3)

ตารางที่ 4.3 : F1-score และค่าเฉลี่ย F1-score หรือค่าความแม่นยำในการพยากรณ์เฉลี่ยรวมทุกคลาส ของโมเดล BiGRU

Class	I (แม่นยำสูง)	II (แม่นยำปานกลาง)	III (แม่นยำต่ำ)	IV (ไม่ต้องวัดสัญญาณ)
Evaluate				
Precision (%)	97.75	96.52	98.14	99.81
Recall (%)	98.46	98.99	92.96	99.94
F1-score (%)	98.10	97.74	95.48	99.88
Support (%)	1104	8821	4644	8589
accuracy (%)	98.10			

จากตารางที่ 4.3 สรุปโดยโฟกัสที่กลุ่ม I และ II ซึ่งเป็นกลุ่มหลักในการวิเคราะห์ผลของ ensemble model ในงานวิจัยนี้ ได้ผลการวัดประสิทธิภาพดังนี้:

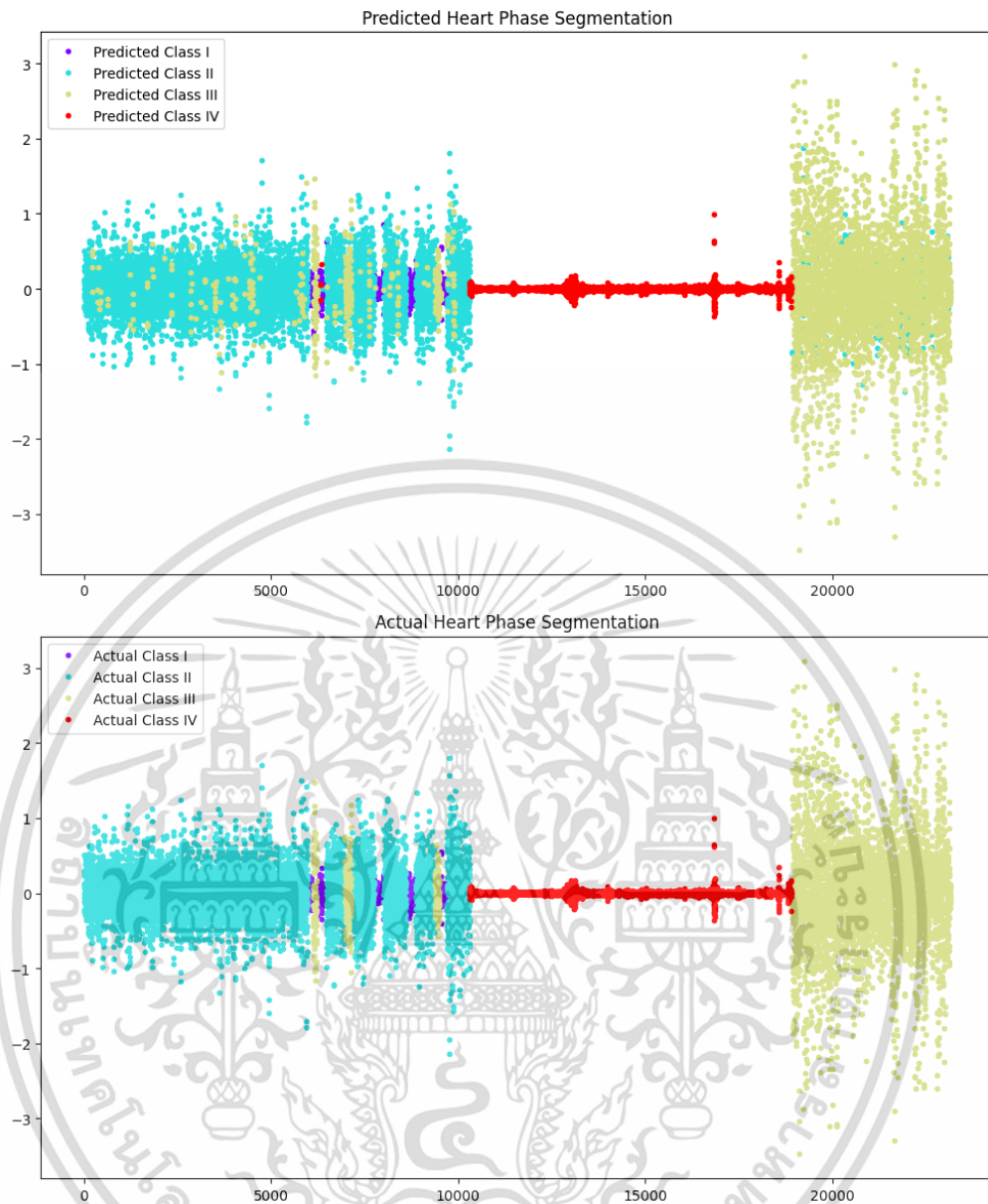
กลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด):

- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 97.75%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 98.46%
- ค่า F1-score: มีค่าประมาณ 98.10 ซึ่งเป็นค่าเฉลี่ย harmonic mean

กลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง):

- ค่า Precision: โมเดลมีความแม่นยำในการทำนายกลุ่ม I ประมาณ 96.52%
- ค่า Recall: โมเดลความน่าจะเป็นที่โมเดลสามารถตรวจจับกลุ่ม I จากจำนวนทั้งหมด ประมาณ 98.99%
- ค่า F1-score: มีค่าประมาณ 97.74 ซึ่งเป็นค่าเฉลี่ย harmonic mean

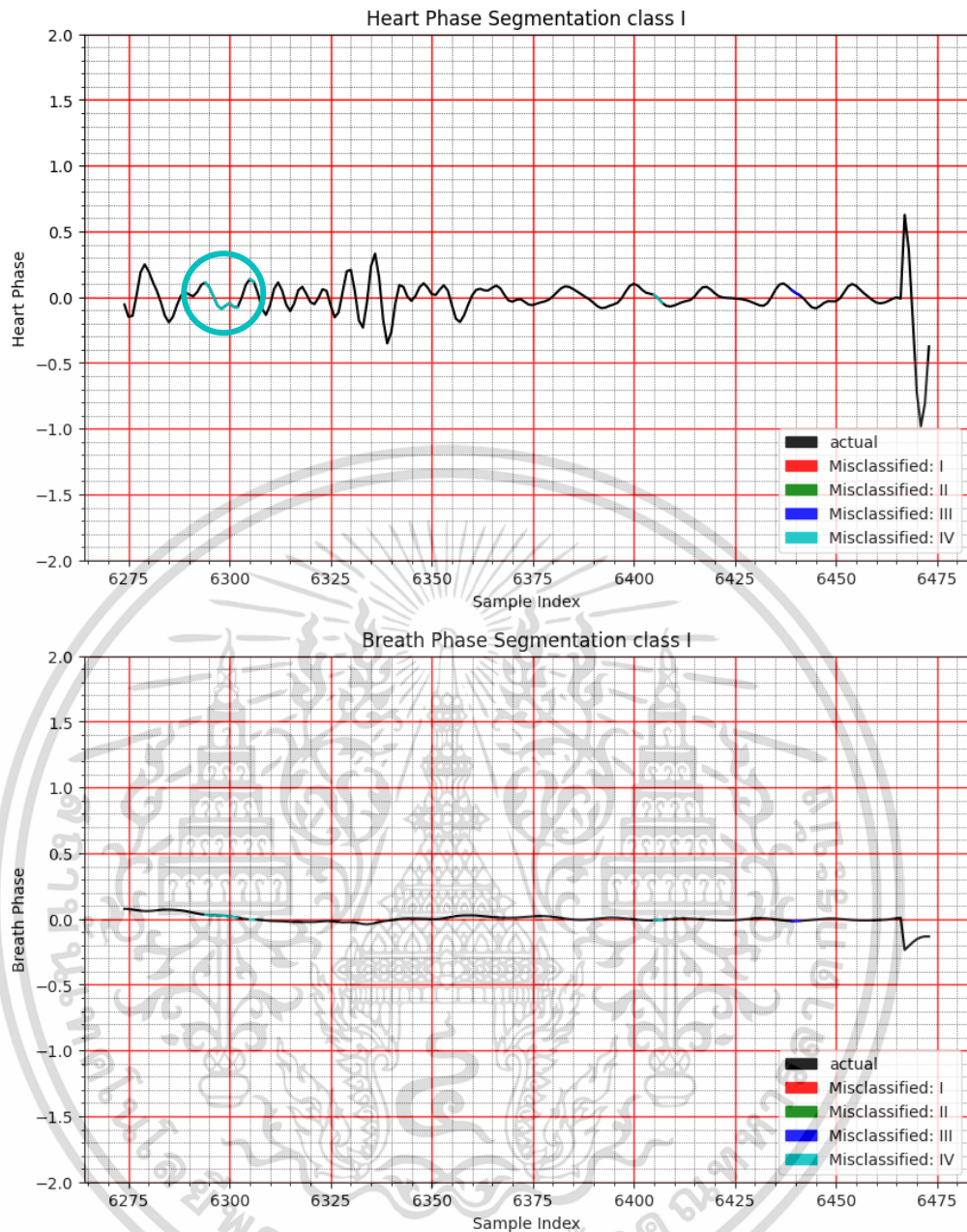
จากผลการวัดประสิทธิภาพของโมเดลสำหรับกลุ่ม I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด) และกลุ่ม II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง) สรุปได้ว่าโมเดล BiGRU สามารถแยกแยะสัญญาณกลุ่มที่ I ได้ดีกว่า กลุ่มที่ II 0.36% ซึ่งใกล้เคียงกันและอยู่ในเกณฑ์ที่สูง และเชื่อถือได้



รูปที่ 4.10 ผลทดสอบกับสัญญาณความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiGRU Classification

จากรูปที่ 4.10 ผลทดสอบกับสัญญาณความถี่ของการเต้นหัวใจทั้ง 4 class ของโมเดล BiLSTM Classification จะประกอบด้วย 2 กราฟ คือ กราฟที่แสดงผลลัพธ์ของโมเดล และกราฟแสดงผลลัพธ์จาก label ทั้ง 4 class จะพบว่าในกรอบสีดำ คือตัวอย่างที่โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

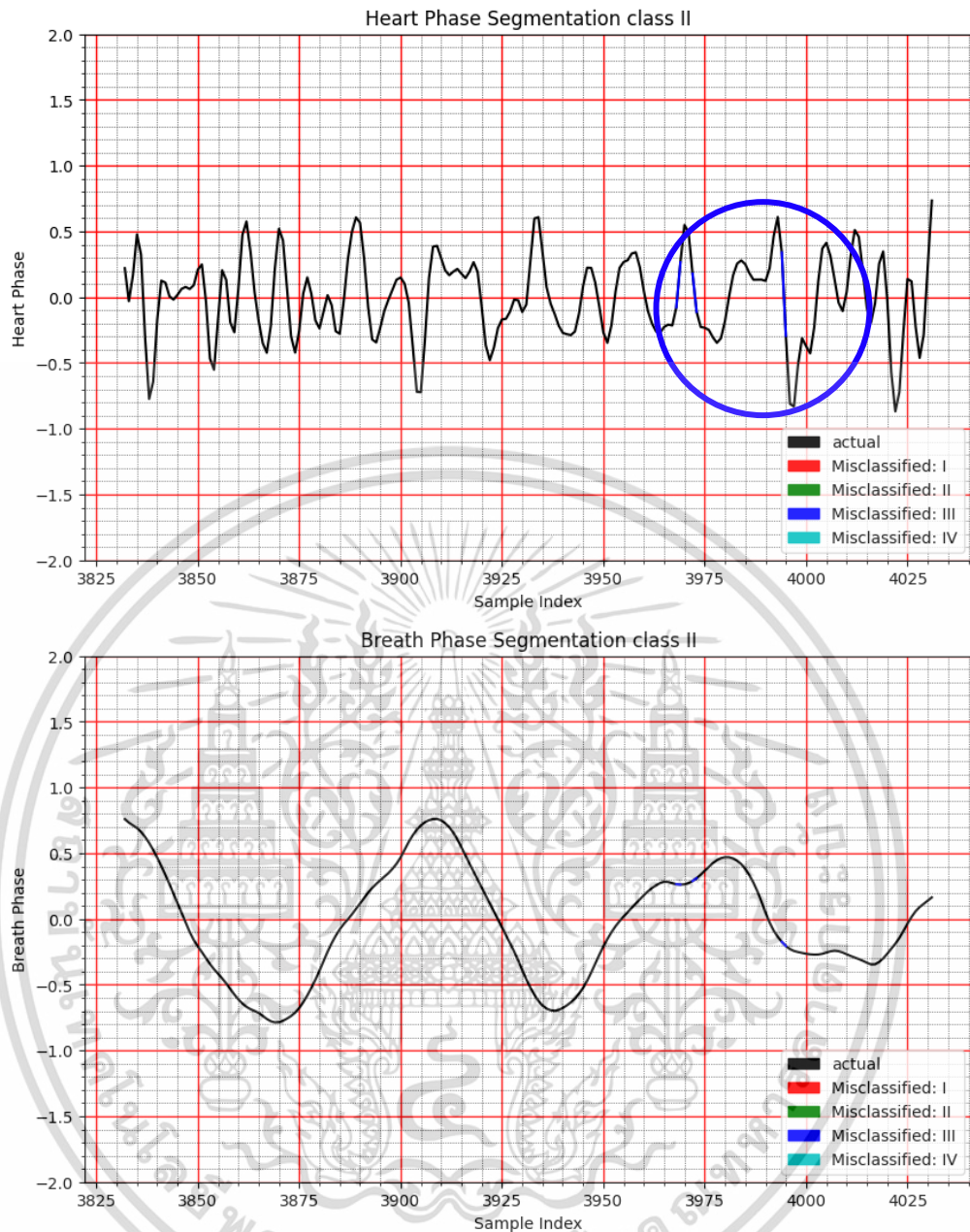
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ผลทดสอบกับสัญญาณ Class I จำนวน 200 sample ของโมเดล BiGRU Classification

จากรูปที่ 4.11 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiGRU Classification จะประกอบด้วย 2 กราฟ คือ กราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.4 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีฟ้า โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class IV (ช่วงที่ไม่ต้องวัดสัญญาณ) แต่ผลที่ถูกต้องคือ class I (ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiGRU Classification

จากรูปที่ 4.12 ผลทดสอบกับสัญญาณ Class II จำนวน 200 sample ของโมเดล BiGRU Classification จะประกอบด้วย 2 กราฟ คือ กราฟบนย่านความถี่หัวใจ และกราฟล่างย่านความถี่หายใจ โดยจากรูปที่ 4.4 คือช่วงที่อาสาสมัครหยุดหายใจ จะพบว่าในกรอบสีน้ำเงิน โมเดลได้ผลลัพธ์ที่ผิดจากที่สอนไว้ โดยโมเดลระบุว่าคือ class III (ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด) แต่ผลที่ถูกต้องคือ class II (ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงาน

ในการวิจัยนี้ เรามีวัตถุประสงค์หลักที่เน้นในการพัฒนาโมเดล AI ที่สามารถวัดสัญญาณหัวใจโดยไม่ต้องสัมผัสผิวหนังผู้ป่วย โดยการใช้คลื่นแม่เหล็กไฟฟ้าความถี่ 77GHz ในการตรวจจับตอบสนองของหัวใจ และแยกสัญญาณออกเป็น 4 กลุ่มสำคัญ คือ ช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด, ช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง, ช่วงที่วัดสัญญาณที่มีความแม่นยำต่ำสุด และช่วงที่ไม่ต้องวัดสัญญาณ

ผลลัพธ์ที่คาดหวังจากการวิจัยนี้คือการพัฒนาโมเดล AI ที่สามารถจัดการกับการเคลื่อนไหวของผู้ป่วยได้ ไม่ว่าจะเป็นการยืนยันจากผู้ป่วยหรือการเคลื่อนไหวที่ไม่คาดคิด โดยใช้โมเดล AI ที่ได้รับการปรับปรุงแล้ว นอกจากนี้ การวิจัยยังมุ่งเน้นการค้นหาวีธีที่ดีที่สุดในการจัดการกับผู้ป่วยที่มีการเคลื่อนไหวโดยไม่คาดคิด และการปรับใช้เทคโนโลยีใหม่ๆ ในการวัดสัญญาณหัวใจที่ไม่ต้องสัมผัสผิวหนัง เพื่อลดความเสี่ยงที่เกิดจากการสัมผัสผิวหนังและเพิ่มความสะดวกสบายให้กับผู้ป่วย

ในสรุปผลการดำเนินงานนี้ เราได้ศึกษาและพัฒนาโมเดล AI ที่สามารถวัดสัญญาณหัวใจโดยไม่ต้องสัมผัสผิวหนังผู้ป่วย โดยใช้คลื่นแม่เหล็กไฟฟ้าความถี่ 77GHz เพื่อตรวจจับตอบสนองของหัวใจ และแยกสัญญาณออกเป็น 4 กลุ่มสำคัญ เพื่อช่วยในการเลือกช่วงเวลาที่มีความแม่นยำที่สุดในการวัดสัญญาณ จากผลการทดสอบในตารางที่ 5.1 ที่เปรียบเทียบผล F1 ของโมเดล 3 รูปแบบ (BiLSTM classification, BiRNN classification, และ BiGRU classification) ในคลาสช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุดและช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง พบว่า BiLSTM classification และ BiGRU classification มีผลลัพธ์ที่ดีที่สุดและใกล้เคียงกัน

ดังนั้น สรุปผลการดำเนินงานในงานวิจัยนี้คือโมเดล BiLSTM classification และ BiGRU classification เป็นโมเดลที่เหมาะสมที่สุดในการใช้งานจริง

ตารางที่ 5.1 : ตารางเปรียบเทียบผล F1 ทั้ง 3 โมเดล โดยเจาะจงที่คลาสช่วงที่วัดสัญญาณที่มีความแม่นยำสูงสุด และช่วงที่วัดสัญญาณที่มีความแม่นยำปานกลาง

BiRNN classification		BiLSTM classification		BiGRU classification	
I (สูงสุด)	II (ปานกลาง)	I (สูงสุด)	II (ปานกลาง)	I (สูงสุด)	II (ปานกลาง)
86.14%	86.20%	97.14%	98.02%	98.10%	97.74%

เอกสารอ้างอิง

- [1] H. Rahman, Fundamental Principles of Radar, Boca Raton, FL: CRC Press Taylor&Francis Group, 2019, pp. 1-10..
- [2] T. K. Sarkar, M. S. Palma and E. L. Mokole, "Echoing Across the Years: A History of Early Radar Evolution," IEEE Microwave Magazine, vol. 17, no. 10, pp. 46 - 60, 2016.
- [3] M. Guarnieri, "The Early History of Radar[Historical]," IEEE Industrial Electronics Magazine, vol. IV, no. 3, pp. 36-42, 2010.
- [4] S. Swords, Technical History of the Beginnings of Radar, London, UK: Peter Peregrinus, 1986.
- [5] H. Rahman, Fundamental Principles of Radar, Boca Raton, FL: CRC Press Taylor&Francis Group, 2019, pp. 1-10.
- [6] T. K. Sarkar, M. S. Palma and E. L. Mokole, "Echoing Across the Years: A History of Early Radar Evolution," IEEE Microwave Magazine, vol. 17, no. 10, pp. 46 - 60, 2016..
- [7] H. Rohling, "From Huelsmeyer's telemobiloskop to the digital radar," in 2014 11th European Radar Conference, Rome, Italy, 2014.
- [8] Z. Xu, C. J. Baker and S. Pooni, "Range and Doppler Cell Migration in Wideband Automotive Radar," IEEE Transactions on Vehicular Technology, vol. 68 , no. 6, pp. 5527 - 5536, 2019..
- [9] Z. Feng, M. Li, M. Stolz, M. Kunert and W. Wiesbeck, "Lane Detection With a High-Resolution Automotive Radar by Introducing a New Type of Road Marking," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 7, pp. 2430 - 2447, 2019 .
- [10] "Machine learning คืออะไร ??? ," Stackpython,18 ธันวาคม 2561. [Online].Available: [https:// www.thaiprogrammer.org/2018/12/](https://www.thaiprogrammer.org/2018/12/) [Accessed 25 เมษายน 2564]
- [11] "ปัญญาประดิษฐ์ ," wikipedia, 4 มกราคม 2564. [Online].Available: <https://th.wikipedia.org/wiki/ปัญญาประดิษฐ์> [Accessed 27 เมษายน 2564]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [12] "How IBM's Deep Blue Beat World Champion Chess Player Garry Kasparov," IEEE Spectrum, [Online]. Available: <https://spectrum.ieee.org/the-institute/ieee-history/how-ibms-deep-blue-beat-world-champion-chess-player-garry-kasparov>. [Accessed 28 April 2021].
- [13] "IBM's Watson comes to computer science department at the University of Toronto," UNIVERSITY OF TORONTO - SINCE 1827, [Online]. Available: <https://www.utoronto.ca/news/ibms-watson-comes-computer-science-department-university-toronto>. [Accessed 29 April 2021].
- [14] "Deep learning vs. machine learning – What's the difference?,"Levity, 3 May 2021,[Online]. Available: <https://levity.ai/blog/difference-machine-learning-deep-learning>. [Accessed 30 April 2021].
- [15] "Deep learning คืออะไร ?," สมาคมโปรแกรมเมอร์ไทย, 16 ธันวาคม 2561.[Online].Available:<https://www.thaiprogrammer.org/2018/12/deep-learning->[Accessed 1 พฤษภาคม 2564].
- [16] "Neural Networks & Backpropagation with ND4J," Drewnoff, [Online]. Available: <https://github.com/drewnoff/spark-notebook-ml-labs/blob/master/labs/DLFramework/README.md>. [Accessed 2 May 2021].
- [17] "DeepLearning,"TechTarget,[Online].Available:<https://searchenterpriseai.techtarget.com/definition/deep-learning-deep-neural-network>. [Accessed 2 May 2021].
- [18] " Convolutional Neural Networks for Visual Recognition," stanford-edu,[Online].Available: <https://cs231n.github.io/neural-networks-1/>. [Accessed 5 May 2021].
- [19] R. Murugadoss and M. Ramakrishnan, "Universal approximation using probabilistic neural networks with sigmoid activation functions," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), 2014, pp. 1-4, doi: 10.1109/ICAETR.2014.7012920.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายรัชพล แคมภูเขียว
วัน เดือน ปีเกิด	28 ธันวาคม 2539
ที่อยู่	174 ม.8 ต.บ้านหัน อ.เกษตรสมบูรณ์ จ.ชัยภูมิ 36120
ประวัติการศึกษา	พ.ศ. 2558 มัธยมศึกษาตอนปลาย โรงเรียนภูเขียว พ.ศ. 2562 วิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงานและผลงานวิจัย

พ.ศ. 2562	ตีพิมพ์หัวข้อวิจัย “Segmentation of 24GHz RF Respiration Signal Using GMM” ในงานประชุมวิชาการ “2019 12th Biomedical Engineering International Conference (BMEiCON)”
-----------	---



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้