การเรียนรู้เชิงลึกสำหรับการแยกสัญญาณแบบไม่รู้แหล่งที่มา

Deep Learning for Blind Source Separation

ศักย์ศรณ์ สร้อยระย้า

Saksorn Soiraya

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียนรู้เชิงลึกสำหรับการแยกสัญญาณแบบไม่รู้แหล่งที่มา

Deep Learning for Blind Source Separation

โดย

ศักย์ศรณ์ สร้อยระย้า

อาจารย์ที่ปรึกษา

ผศ.ดร. ยุทธนา คิดใจเดียว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2565

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเรียนรู้เชิงลึกสำหรับการแยกสัญญาณแบบไม่รู้แหล่งที่มา

Deep Learning for Blind Source Separation

ผู้จัดทำ นายศักย์ศรณ์ สร้อยระย้า รหัสนักศึกษา 62010860

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ผศ.ดร. ยุทธนา คิดใจเดียว

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	การเรียนรู้เชิงลึกสำหรับการแยกสัญญาณแบบไม่รู้แหล่งที่มา
นักศึกษา	นายศักย์ศรณ์ สร้อยระย้า รหัสนักศึกษา 62010860
ปริญญา	วิศวกรรมศาสตร์บัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษาโครงงาน	ผศ.ดร. ยุทธนา คิดใจเดียว

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอเกี่ยวกับการใช้ปัญญาประดิษฐ์มาประยุกต์ใช้ร่วมกับกระบวนการประมวลผลสัญญาณโดยมุ่งเน้นไปที่สัญญาณรวม โดยทางผู้จัดทำได้นำข้อมูลที่มีสัญญาณหลายแหล่งสัญญาณหรือสัญญาณรวม นำมาผ่านกระบวนการจำแนกสัญญาณออกด้วยวิธีการวิเคราะห์ส่วนประกอบอิสระ และอีกวิธีซึ่งนำปัญญาประดิษฐ์มาปรับใช้เพิ่มเติม เพื่อให้ได้ผลลัพธ์ที่คาดว่าจะมีประสิทธิภาพมากขึ้น ซึ่งผลลัพธ์ที่ออกมานั้นยังไม่เป็นที่น่าพอใจแต่สามารถคัดแยกสัญญาณออกจากกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Deep Learning for Blind Source Separation
Student	Mr.Saksorn Soiraya Student ID 62010860
Degree	Bachelor of Engineering
Program	Electronics Engineering
Academic Year	2022
Advisor	Asst. Prof. Yuttana Kitjaidure, Ph.D

ABSTRACT

This paper reports application of artificial intelligence in signal processing, particularly in the context of Mixed signals, has garnered significant interest. However, organizers have encountered challenges when dealing with information derived from multiple signal sources or combined signals. To address this issue, independent components analysis (ICA) has been utilized for signal classification. Nonetheless, the effectiveness of ICA has been limited, prompting the exploration of alternative approaches, such as the use of artificial intelligence, to achieve more efficient results. Despite these efforts, the outcome remains suboptimal, albeit with some success in separating signals from each other.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ได้ลุล่วงไปได้ด้วยความกรุณาจากอาจารย์ยุทธนา คิดใจเดียว อาจารย์ที่ปรึกษาที่คอยให้คำแนะนำ แนวทาง แนวคิดการทำโครงงานและการแก้ไขปัญหามาโดยตลอด

ขอขอบคุณเพื่อนๆที่สนับสนุนให้คำแนะนำที่เป็นประโยชน์สำหรับการทำงานและการช่วยเหลือต่างๆทำให้ได้รับประสบการณ์จากการทำโครงงานในครั้งนี้เป็นอย่างดี

ขอขอบคุณหนังสือตำรา ข้อมูลอ้างอิงและงานวิจัยต่างๆที่ทำให้ศึกษาเพื่อเป็นแนวทางในการนำมาปรับใช้ในการทำงานและส่งผลให้งานนี้ผ่านพ้นไปได้

ศักย์ศรณ์ สร้อยระย้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

ABSTRACT II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VI

สารบัญตาราง VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 สมมุติฐานของการศึกษา 1

1.4 ขอบเขตของโครงงาน 1

1.5 ประโยชน์ที่คาดว่าจะได้รับ 1

1.6 ระยะเวลาทำการวิจัยและแผนการดำเนินงานโครงงาน 2

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3

2.1 Basic of Signal Processing 3

2.1.1 ความหมายของสัญญาณ 3

2.1.2 การประมวลผลสัญญาณ(Signal Processing) 3

2.2 การแยกสัญญาณแบบไม่รู้แหล่งที่มา(Blind Source Separation) 3

2.2.1 การวิเคราะห์ส่วนประกอบหลัก(Principal Component Analysis) 4

2.2.2 การวิเคราะห์ส่วนประกอบอิสระ (Independent Component Analysis) 5

2.2.3 การแยกตัวประกอบเมทริกซ์ที่ไม่ใช่เชิงลบ (Non-negative Matrix Factorization) 6

2.2.4 การประยุกต์ใช้งาน 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ปัญญาประดิษฐ์ (Artificial Intelligence) 7
2.4 แมชชีนเลิร์นนิง (Machine learning) 7
2.4.1. การเรียนรู้ภายใต้การดูแล (Supervised Learning) 8
2.4.2. การเรียนรู้ที่ไม่ได้รับการดูแล (Unsupervised Learning) 8
2.4.3. การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) 8
2.5 การเรียนรู้เชิงลึก (Deep learning) 8
2.6 เครือข่ายประสาทเทียม (Neural Network) 8
2.7 Recurrent Neural Network (RNN) 8
2.8 Generative Adversarial Networks (GANs) 9
2.9 Long Short-Term Memory (LSTM) 9
บทที่ 3 วิธีการดำเนินงาน 11
3.1 เครื่องมืออุปกรณ์และข้อมูล 11
3.2 ขั้นตอนการดำเนินงาน 12
บทที่ 4 ผลการดำเนินงาน 13
4.1 ผลการดำเนินงาน 13
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 21
5.1 สรุปผลการทดลอง 21
5.2 วิจารณ์ผลการทดลอง 21
บรรณานุกรม 22
ภาคผนวก 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างของการแยกสัญญาณแบบไม่รู้แหล่งที่มา 4

2.2 ตัวอย่างของ PCA 4

2.3 ภาพประกอบการทำงานของ ICA 5

2.4 ความแตกต่างระหว่าง PCA และ ICA 5

2.5 หลักการทำงานของ NMF 6

2.6 ตัวอย่าวการใช้งาน NMF 6

2.7 โครงสร้างของปัญญาประดิษฐ์ 7

2.8 โครงสร้างของ RNN 9

2.9 ความแตกต่างระหว่าง RNN และ LSTM 10

4.1 ข้อมูลสัญญาณเบส 13

4.2 ข้อมูลสัญญาณเสียงนักร้อง 14

4.3 ข้อมูลสัญญาณเสียงเครื่องดนตรีอื่นๆ 15

4.4 ข้อมูลสัญญาณเสียงกลอง 16

4.5 ข้อมูลสัญญาณเสียงรวม 17

4.6 ผลลัพธ์จากโปรแกรม ICA 18

4.7 ผลลัพธ์จากโปรแกรม LSTM 18

4.8 ผลลัพธ์จากโปรแกรม GAN 18

4.9 ผลลัพธ์จากโปรแกรม Deep RNN 19

4.10 ผลลัพธ์จากโปรแกรม RNN 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

1.1 ระยะเวลาการดำเนินงาน 2

3.1 ข้อมูล Dataset ที่นำมาใช้งาน 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

การแยกสัญญาณแบบไม่รู้แหล่งที่มา(Blind Source Separation/BSS) ได้กลายเป็นเทคนิคการประมวลผลสัญญาณที่ได้รับความนิยมในด้านต่างๆ เนื่องจากความสามารถในการแยกสัญญาณผสมออกเป็นสัญญาณเดี่ยวแต่ละส่วน ข้อได้เปรียบหลักของ BSS คือความสามารถในการดึงข้อมูลที่เป็นประโยชน์จากข้อมูลที่มีเสียงดังและซับซ้อน เทคนิค BSS ถูกใช้อย่างแพร่หลายในสาขาต่างๆ เช่น โทรคมนาคม วิศวกรรมชีวการแพทย์ และการรู้จำเสียง ซึ่งความสามารถในการแยกสัญญาณต่างๆ ออกเป็นสิ่งสำคัญสำหรับการวิเคราะห์และตีความข้อมูลอย่างแม่นยำยิ่งขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 ศึกษาทฤษฎีหลักการประมวลผลสัญญาณ

1.2.2 ศึกษาทฤษฎีโครงข่ายประสาทเทียม(Neural Network)

1.2.3 ศึกษาการออกแบบโมเดลเพื่อจำแนกสัญญาณ

1.2.4 เพื่อใช้ AI มอดูเลตและหาพาหะของสัญญาณได้

1.3 สมมุติฐานของการศึกษา

สมมุติฐานของการศึกษาอาจเกิดปัญหาในการนำข้อมูลมาจำแนกเนื่องจากการจำแนกสัญญาณรวมนั้น อาจจะสามารถคาดการณ์โดเมนความถี่ได้ที่ซึ่งสังเกตได้ง่ายกว่าการคาดการณ์สัญญาณในโดเมนเวลา

1.4 ขอบเขตของโครงงาน

1.4.1 สามารถถอดรหัสสัญญาณจากสัญญาณรวมได้

1.4.2 โมเดลที่ออกแบบสามารถทำนายผลได้แม่นยำเกิน 70 %

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้รับทักษะกระบวนการคิดการออกแบบโครงสร้างของโมเดล

1.5.2 ได้รับทักษะในการใช้โปรแกรม Visual Studio Code และ MATLAB

1.5.3 ความเข้าใจและสามารถออกแบบอัลกอริทึมที่ใช้ในการวิเคราะห์สัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ระยะเวลาทำการวิจัยและแผนการดำเนินงานโครงงาน

เริ่มต้นตั้งแต่ 3 กุมภาพันธ์ 2566 จนถึง 10 พฤษภาคม 2566

ตารางที่ 1.1 ระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	สัปดาห์ที่														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. ศึกษาทฤษฎี		◄	—	—	—	—	►	สอบกลางภาค	◄	—	—	—	►		
2. ออกแบบและเทรนโมเดล				◄	—	—	►		◄	—	—	—	—	►	
3. ทดสอบโมเดล									◄	—	—	—	—	►	
4. แก้ไขปัญหา									◄	—	—	—	—	►	
5. สรุปผลโครงงาน														◄►	
6. เขียนรายงานฉบับสมบูรณ์														◄	►

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Basic of Signal Processing

2.1.1 ความหมายของสัญญาณ

สัญญาณ คือ ปริมาณทางฟิสิกส์ที่มีการเปลี่ยนแปลงไปกับตัวแปรต้นที่มักจะเป็นเวลาหรือระยะทาง ปรากฏการณ์ทางกายภาพหลายอย่าง สามารถอธิบายได้ด้วยสิ่งที่เรียกว่าสัญญาณ โดยข้อมูลที่ต้องการจะแฝงอยู่ในรูปแบบการเปลี่ยนแปลงของสัญญาณ ตัวอย่างเช่น สัญญาณทางไฟฟ้า สัญญาณเสียง หรือสัญญาณภาพเป็นต้น

2.1.2 การประมวลผลสัญญาณ(Signal Processing)

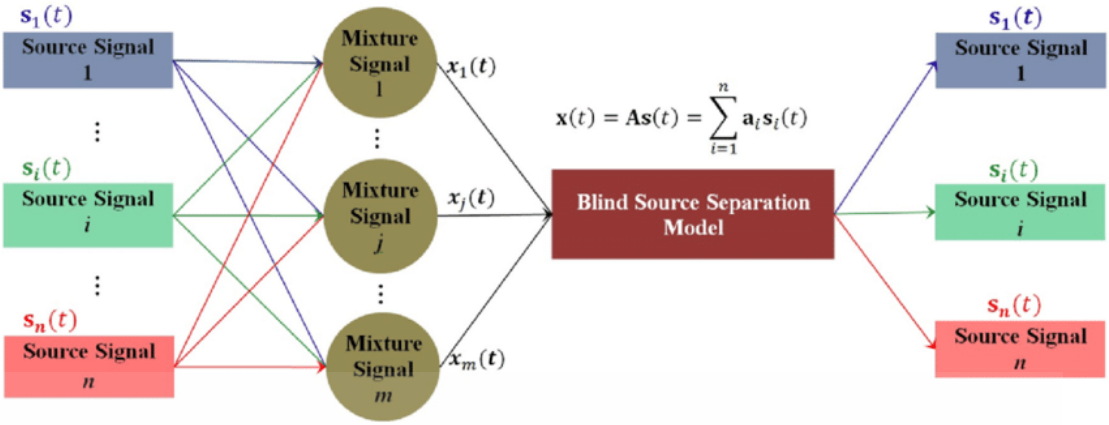
การประมวลผลสัญญาณ คือ การจัดการสัญญาณประเภทต่างๆ ด้วยการกรอง การวัด การบีบอัด และการผลิตสัญญาณอะนาล็อก สัญญาณอะนาล็อกจะแตกต่างกันไปตามการรับข้อมูลและแปลงเป็นพัลส์ไฟฟ้าของแอมพลิจูดที่แตกต่างกัน ในขณะที่ข้อมูลสัญญาณดิจิตอลจะถูกแปลงเป็นรูปแบบไบนารี่ (Binary) ความแตกต่างที่เห็นได้ชัดเจนอีกอย่างหนึ่งก็คือสัญญาณอะนาล็อกสามารถถูกแทนด้วยคลื่นไซน์และ สัญญาณดิจิตอลจะแสดงเป็นคลื่นสี่เหลี่ยม DSP สามารถพบได้ในเกือบทุกสาขาไม่ว่าจะเป็น การแปรรูปน้ำมัน การสร้างเสียงเรดาร์ และโซนาร์การประมวลผลภาพทางการแพทย์หรือการสื่อสารโทรคมนาคม เป็นต้น

2.2 การแยกสัญญาณแบบไม่รู้แหล่งที่มา(Blind Source Separation)

การแยกสัญญาณแบบไม่รู้แหล่งที่มา(Blind Source Separation) เป็นเทคนิคที่ใช้ในการประมวลผลสัญญาณและแมชชีนเลิร์นนิ่งเพื่อแยกสัญญาณผสมออกเป็นส่วนประกอบต้นทางแต่ละส่วนโดยไม่ต้องมีความรู้มาก่อนเกี่ยวกับกระบวนการผสม

อัลกอริธึมการแยกสัญญาณแบบไม่รู้แหล่งที่มานี้ใช้เทคนิคทางสถิติเช่น การวิเคราะห์ส่วนประกอบหลัก(PCA), การวิเคราะห์ส่วนประกอบอิสระ(ICA), การแยกตัวประกอบเมทริกซ์ที่ไม่เป็นเชิงลบ(NMF) และอื่น ๆ เพื่อแยกแต่ละสัญญาณออกจากสัญญาณผสม

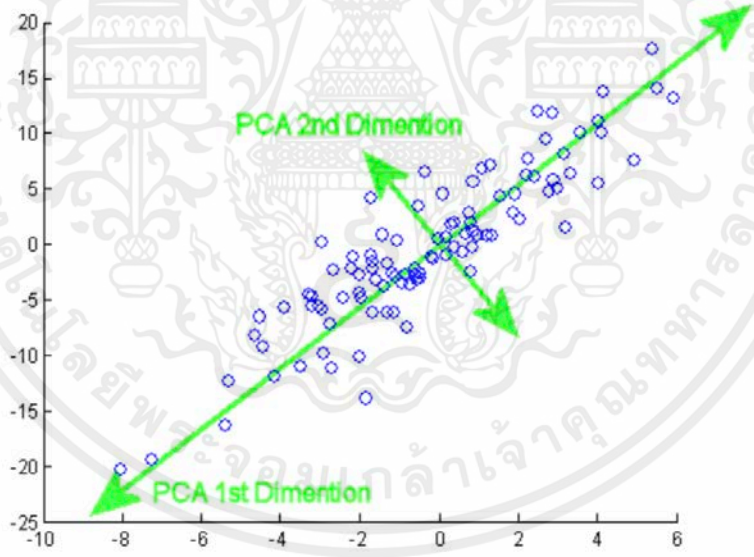
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 โครงสร้างของการแยกสัญญาณแบบไม่รู้แหล่งที่มา

2.2.1 การวิเคราะห์ส่วนประกอบหลัก(Principal Component Analysis)

การวิเคราะห์ส่วนประกอบหลัก(Principal Component Analysis/PCA) เป็นเทคนิคทางสถิติที่ใช้เพื่อลดมิติของชุดข้อมูลมิติสูงโดยการระบุคุณสมบัติหรือส่วนประกอบที่สำคัญที่สุดของข้อมูล การวิเคราะห์ส่วนประกอบหลักทำงานโดยการระบุทิศทางในข้อมูลที่มีรูปแบบมากที่สุดและฉายข้อมูลไปยังทิศทางเหล่านี้เพื่อสร้างการแสดงข้อมูลที่มีมิติต่ำกว่า

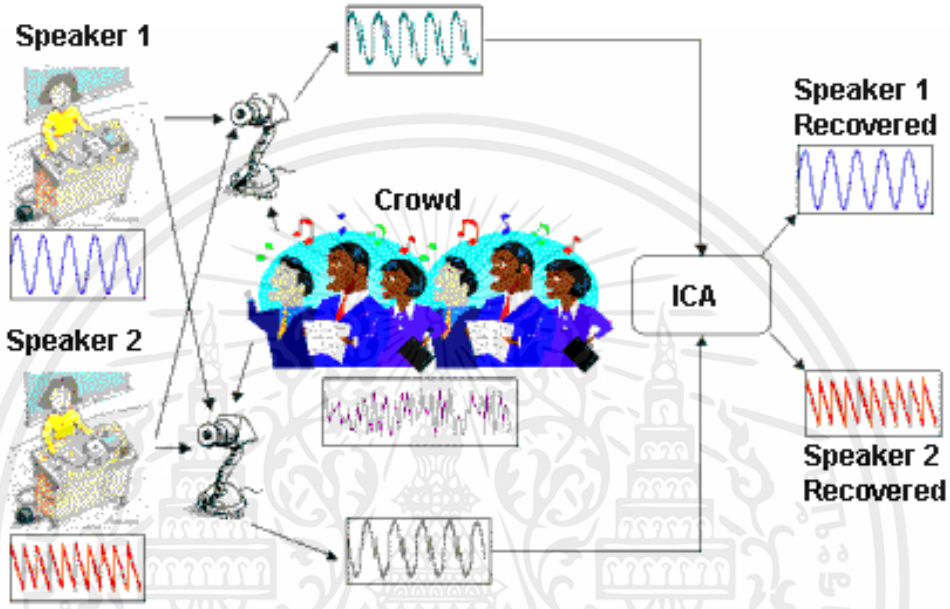


รูปที่ 2.2 ตัวอย่างของ PCA

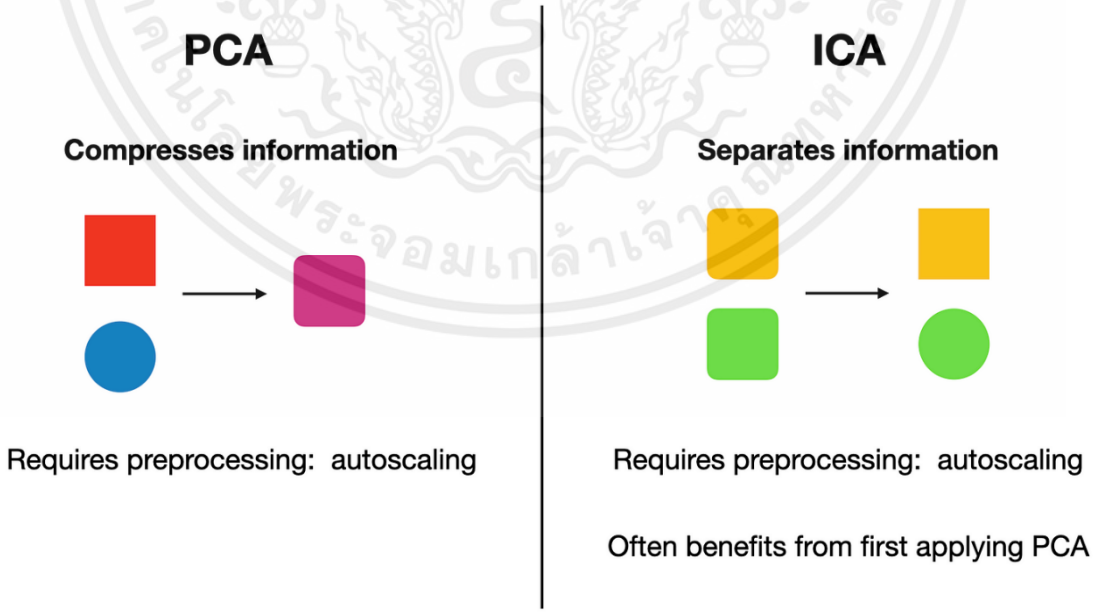
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การวิเคราะห์ส่วนประกอบอิสระ (Independent Component Analysis)

การวิเคราะห์ส่วนประกอบอิสระ(Independent Component Analysis/ICA) คือ เทคนิคการคำนวณสำหรับการย่อยสลายสัญญาณหลายตัวแปรเป็นส่วนประกอบย่อยสารเติมแต่ง โดยสมมติว่าส่วนประกอบย่อยเป็นอิสระจากกันทางสถิติและมีเพียงองค์ประกอบย่อยเดียวเท่านั้นคือเกาส์ ตัวอย่างเฉพาะของการแยกสัญญาณแบบไม่รู้แหล่งที่มา เช่น การแก้ไขปัญหาเสียงหลายแหล่งผสมกัน



รูปที่ 2.3 ภาพประกอบการทำงานของ ICA

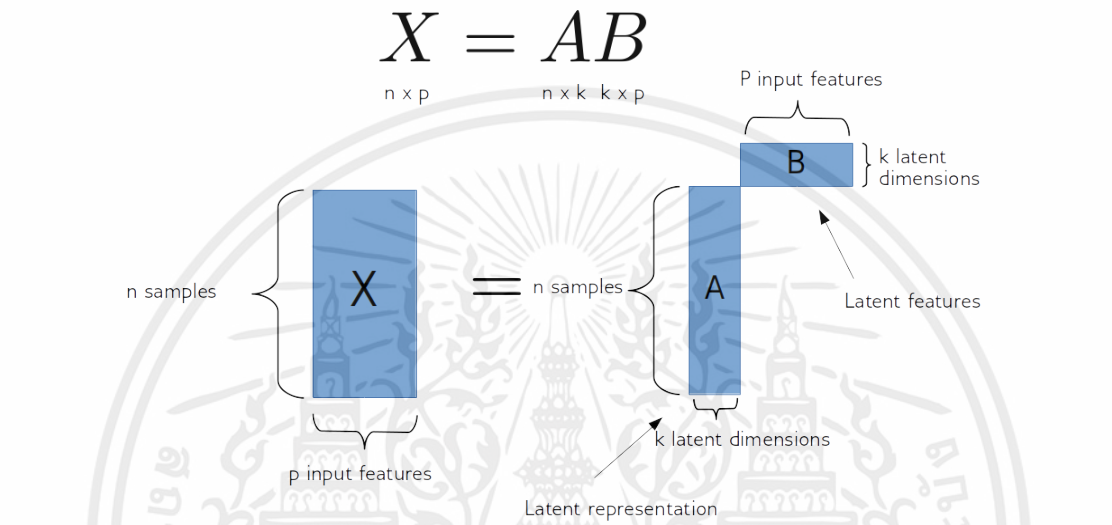


รูปที่ 2.4 ความแตกต่างระหว่าง PCA และ ICA

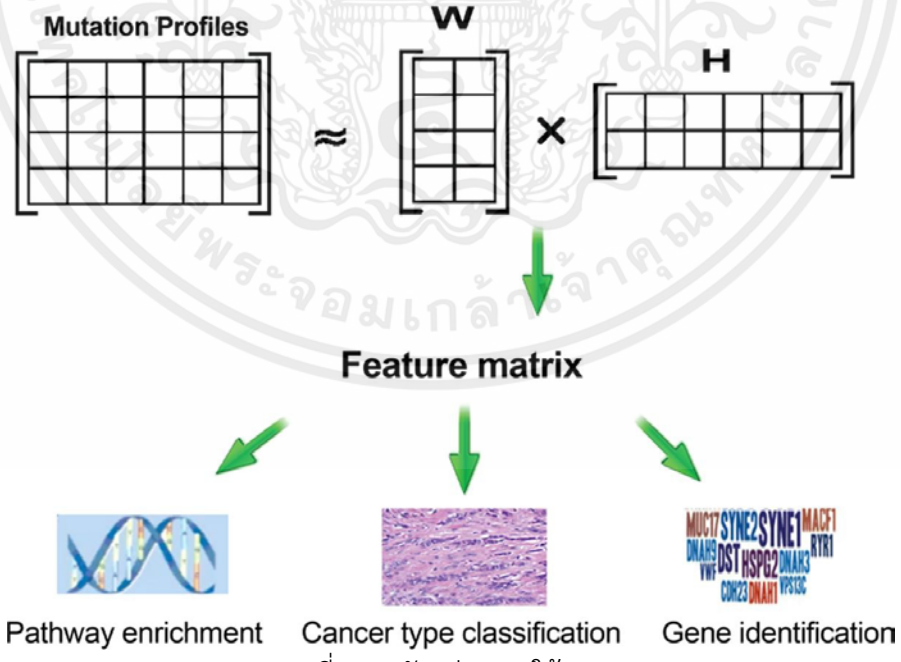
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การแยกตัวประกอบเมทริกซ์ที่ไม่ใช่เชิงลบ (Non-negative Matrix Factorization)

การแยกตัวประกอบเมทริกซ์ที่ไม่เป็นเชิงลบ(Non-negative Matrix Factorization/NMF) เป็นเทคนิคการเรียนรู้ของเครื่องที่ใช้สำหรับการแยกตัวประกอบเมทริกซ์ซึ่งเป้าหมาย คือ การแยกเมทริกซ์ที่ไม่ใช่เชิงลบออกเป็นเมทริกซ์ที่ไม่ใช่ลบสองตัว การแยกตัวประกอบเมทริกซ์ที่ไม่เป็นเชิงลบใช้เพื่อแยกรูปแบบหรือคุณสมบัติพื้นฐานออกจากข้อมูลและมีแอปพลิเคชันในด้านต่างๆเช่นการประมวลผลภาพการประมวลผลภาษาธรรมชาติและชีวสารสนเทศศาสตร์



รูปที่ 2.5 หลักการทำงานของ NMF



รูปที่ 2.6 ตัวอย่างการใช้งาน NMF

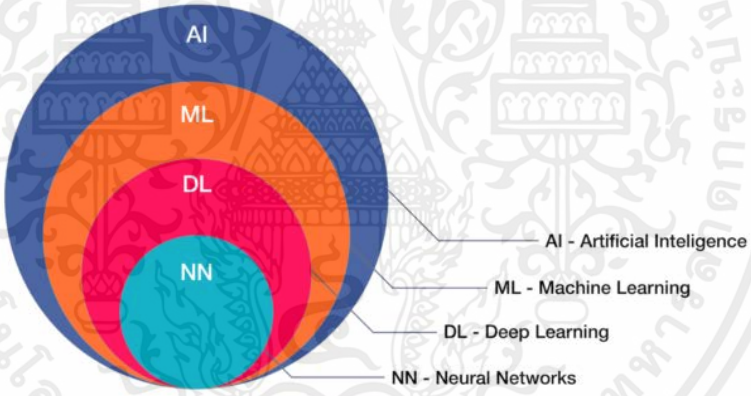
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การประยุกต์ใช้งาน

การแยกสัญญาณแบบไม่รู้แหล่งที่มานั้นสามารถนำไปประยุกต์ใช้ได้หลายด้านรวมถึงการประมวลผลเสียงการประมวลผลภาพและการวิเคราะห์สัญญาณชีวการแพทย์ ตัวอย่างเช่นในการประมวลผลเสียงการแยกสัญญาณแบบไม่รู้แหล่งที่มา สามารถใช้เพื่อแยกแต่ละแหล่งในการบันทึกเสียงแบบผสมเช่นการแยกเสียงร้องออกจากเพลงประกอบ ในการวิเคราะห์สัญญาณทางชีวการแพทย์สามารถใช้ BSS เพื่อแยกส่วนประกอบแต่ละส่วนในสัญญาณผสมเช่นการแยกสัญญาณคลื่นไฟฟ้าหัวใจ (ECG) ออกจากสัญญาณ electromyogram (EMG) ในการบันทึก electromyography (EMG)

2.3 ปัญญาประดิษฐ์ (Artificial Intelligence)

ปัญญาประดิษฐ์(Artificial Intelligence) คือ ศาสตร์แขนงหนึ่งทางด้านเทคโนโลยีที่มีพื้นฐานมาจากวิชาวิทยาการคอมพิวเตอร์และ วิทยาศาสตร์ ชีววิทยา จิตวิทยา ภาษาศาสตร์คณิตศาสตร์ และวิศวกรรมศาสตร์ ซึ่งเป้าหมายของปัญญาประดิษฐ์ คือ การพัฒนาระบบคอมพิวเตอร์ให้มีพฤติกรรมเลียนแบบกระบวนการการคิดของมนุษย์



รูปที่ 2.7 โครงสร้างของปัญญาประดิษฐ์

2.4 แมชชีนเลิร์นนิง (Machine learning)

แมชชีนเลิร์นนิงเป็นปัญญาประดิษฐ์ประเภทหนึ่งที่เกี่ยวข้องกับอัลกอริธึมการฝึกอบรมเพื่อเรียนรู้รูปแบบและความสัมพันธ์ในข้อมูลโดยไม่ต้องตั้งโปรแกรมไว้อย่างชัดเจน กล่าวอีกนัยหนึ่งการเรียนรู้ของเครื่องช่วยให้คอมพิวเตอร์เรียนรู้จากประสบการณ์และปรับปรุงประสิทธิภาพในการทำงานเมื่อเวลาผ่านไป

แมชชีนเลิร์นนิงมีสามประเภทหลักๆ ได้แก่ การเรียนรู้ภายใต้การดูแล (Supervised Learning), การเรียนรู้ที่ไม่ได้รับการดูแล (Unsupervised Learning) และการเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1. การเรียนรู้ภายใต้การดูแล (Supervised Learning)

เกี่ยวข้องกับการฝึกอบรมแบบจำลองบนชุดข้อมูลที่มีป้ายกำกับ ซึ่งแต่ละจุดข้อมูลจะเชื่อมโยงกับเป้าหมายหรือป้ายชื่อ แบบจำลองเรียนรู้ที่จะทำนายเป้าหมายที่ได้รับข้อมูลอินพุตใหม่ ตัวอย่างของงานการเรียนรู้ภายใต้การควบคุมดูแล ได้แก่ การจำแนกรูปภาพ การรู้จำเสียง และการแปลภาษา

2.4.2. การเรียนรู้ที่ไม่ได้รับการดูแล (Unsupervised Learning)

เกี่ยวข้องกับการฝึกอบรมแบบจำลองบนชุดข้อมูลที่ไม่มีป้ายกำกับ ซึ่งเป้าหมายคือการค้นหารูปแบบหรือโครงสร้างในข้อมูล ตัวอย่างของงานการเรียนรู้ที่ไม่ได้รับการดูแล ได้แก่ การจัดกลุ่มและการลดมิติ

2.4.3. การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning)

เกี่ยวข้องกับการฝึกอบรมแบบจำลองเพื่อตัดสินใจตามรางวัลหรือข้อเสนอแนะ แบบจำลองเรียนรู้ที่จะเพิ่มผลตอบแทนสูงสุดเมื่อเวลาผ่านไปโดยการดำเนินการที่นำไปสู่ผลลัพธ์เชิงบวก ตัวอย่างของงานการเรียนรู้เสริมกำลัง ได้แก่ การเล่นเกมและการควบคุมหุ่นยนต์

2.5 การเรียนรู้เชิงลึก (Deep learning)

การเรียนรู้เชิงลึกเป็นแมชชีนเลิร์นนิงประเภทหนึ่งที่เกี่ยวข้องกับการฝึกอบรมเครือข่ายประสาทเทียมเพื่อทำงานที่หลากหลายเช่นการจำแนกการถดถอยและการจัดกลุ่ม แบบจำลองการเรียนรู้เชิงลึกมักประกอบด้วยเซลล์ประสาทที่เชื่อมต่อกันหลายชั้น และกระบวนการฝึกอบรมเกี่ยวข้องกับการปรับน้ำหนักและอคติของเซลล์ประสาทเหล่านี้เพื่อเพิ่มประสิทธิภาพการทำงานของแบบจำลองในงานที่กำหนด

2.6 เครือข่ายประสาทเทียม (Neural Network)

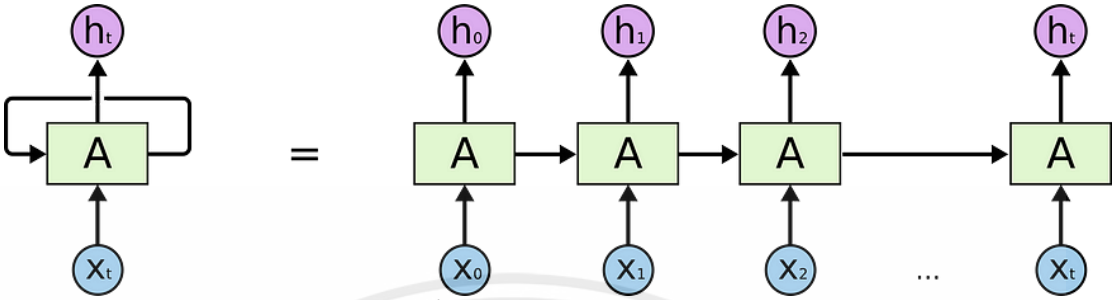
เครือข่ายประสาทเทียม(Neural Network) เป็นปัญญาประดิษฐ์ประเภทหนึ่งที่จำลองมาจากโครงสร้างและการทำงานของสมองมนุษย์ เครือข่ายประสาทเทียมประกอบด้วยโหนดที่เชื่อมต่อกันหรือ "เซลล์ประสาท" ที่ประมวลผลและส่งข้อมูล เซลล์ประสาทถูกจัดเรียงเป็นชั้น ๆ โดยแต่ละชั้นมีหน้าที่รับผิดชอบด้านต่างๆของการคำนวณ

2.7 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) เป็นเครือข่ายประสาทเทียมประเภทหนึ่งที่ออกแบบมาเพื่อจัดการข้อมูลตามลำดับ RNN ซึ่งแตกต่างจากประสาทเทียมอื่นๆตรงที่มีลูปภายในเลเยอร์ ทำให้สามารถประมวลผลลำดับอินพุตและรักษาสถานะหรือหน่วยความจำของอินพุตก่อนหน้าได้ กลไก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

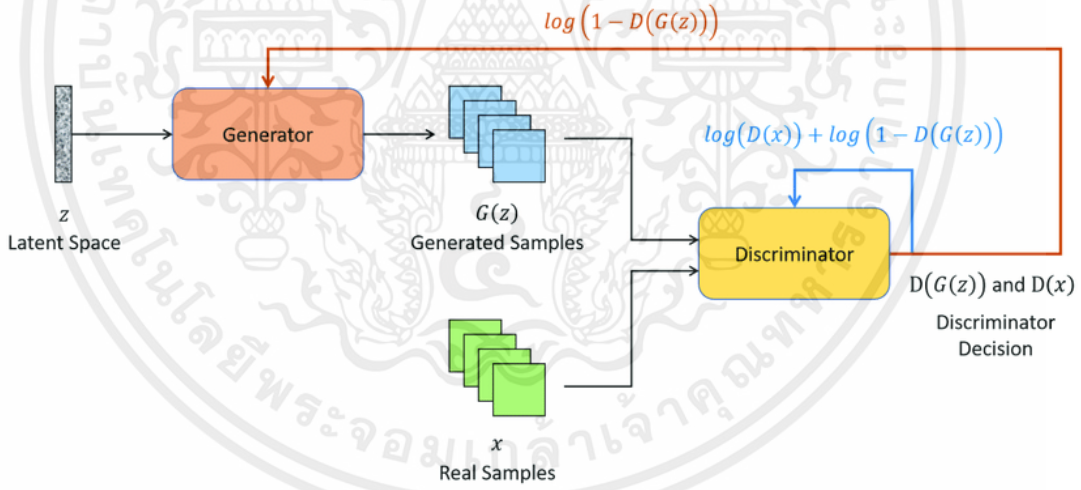
หน่วยความจำนี้ช่วยให้ RNN สามารถวิเคราะห์และคาดการณ์ข้อมูลตามลำดับ เช่น การรู้จำเสียง การรู้จำลายมือ การประมวลผลภาษาธรรมชาติ และการวิเคราะห์วิดีโอ



รูปที่ 2.8 โครงสร้างของ RNN

2.8 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) คือ เครือข่ายประสาทเทียมการเรียนรู้เชิงลึกที่สามารถสร้างข้อมูลใหม่โดยรูปแบบการเรียนรู้และลักษณะจากข้อมูลที่มีอยู่ GANs ประกอบด้วยเครือข่ายประสาทเทียมสองเครือข่ายเครื่องกำเนิดไฟฟ้าและผู้เลือกปฏิบัติซึ่งได้รับการฝึกฝนร่วมกันในกระบวนการแข่งขัน

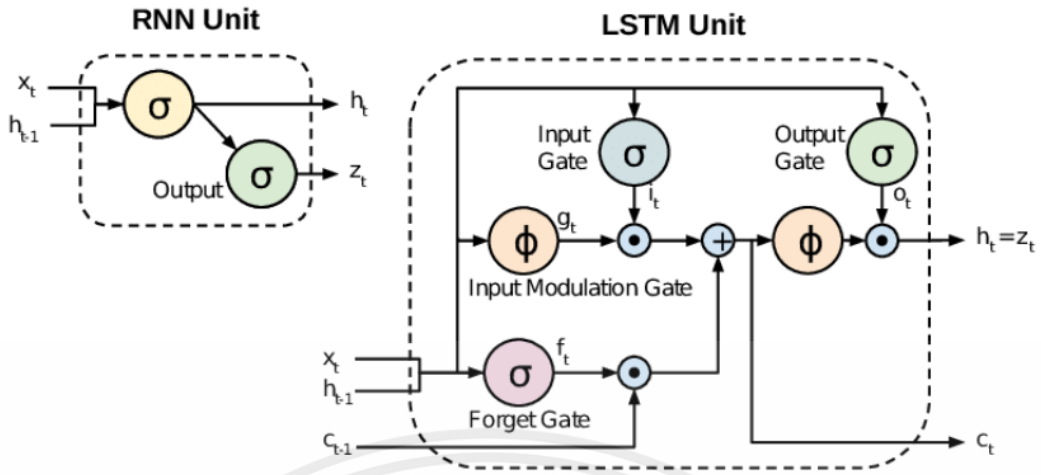


รูปที่ 2.8 โครงสร้างของ GAN

2.9 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) คือ ประเภทของ Recurrent Neural Network (RNN) ที่ออกแบบมาเพื่อเอาชนะปัญหาการไล่ระดับสีที่หายไปซึ่งเกิดขึ้นเมื่อการไล่ระดับสีมีขนาดเล็กเกินไปที่จะเป็นประโยชน์ในระหว่างการ backpropagation ในเครือข่ายประสาทเทียมลึก เครือข่าย LSTM เหมาะอย่างยิ่งสำหรับการประมวลผลข้อมูลตามลำดับและถูกนำมาใช้อย่างประสบความสำเร็จในแอปพลิเคชันที่หลากหลายรวมถึงการรู้จำเสียงการประมวลผลภาษาธรรมชาติและคำบรรยายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ความแตกต่างระหว่าง RNN และ LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

3.1 เครื่องมืออุปกรณ์และข้อมูล

Sample Dataset

DSD Dataset

MUSDB18 Dataset

Visual Studio Code

MATLAB

ซึ่งข้อมูล Sample Dataset นั้น คือ ข้อมูลที่กำหนดขึ้นมาเองเพื่อทดสอบการทำงานของโปรแกรม และ MUSDB18 Dataset กับ DSD Dataset คือ ข้อมูลสัญญาณเสียงดนตรี โดย MUSDB18 เป็นสัญญาณทั้งหมด 100 เพลง แต่ละเพลงจะมีไฟล์ Mixture คือ สัญญาณรวม, Vocal คือ สัญญาณของเสียงนักร้อง, Drum คือ สัญญาณของกลอง, bass คือ สัญญาณของเบส และOther คือ สัญญาณเครื่องดนตรีอื่นๆ DSD Dataset คือ เพลงที่สุ่มมาจาก MUSDB18 Dataset ซึ่งมีเพลงทั้งหมด 4 เพลง ดังนี้

ตารางที่ 3.1 ข้อมูล Dataset ที่นำมาใช้งาน

ชื่อเพลง	แนวเพลง	ระยะเวลา
Angels In Amplifiers - I'm Alright	Smooth Acoustic Rock	3'00"
AM Contra - Heart Peripheral	Electronic Dance Pop	3'30"
Cnoc An Tursa - Bannockburn	Death Metal	4'54"
Patrick Talbot - Set Free Me	Jazz/Singer-Songwriter	4'50"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการดำเนินงาน

Sample Dataset ประกอบไปด้วยสัญญาณดังนี้

$$S = \begin{cases} s_1 = \sin(2t) \\ s_2 = sign(\sin(3t)) \\ s_3 = sawtooth(2\pi t) \end{cases}$$

โดยกำหนดให้จำนวน sampling points เป็น 2000 แล้วนำสัญญาณทั้งหมดมารวมกันและเพิ่มสัญญาณรบกวนเล็กน้อย จากนั้นนำเข้าสู่กระบวนการ ICA กับ PCA เพื่อเปรียบเทียบผลลัพธ์และแสดงผลลัพธ์ในรูปของโดเมนเวลา

แยกสัญญาณแบบไม่รู้แหล่งที่มาโดยใช้การเรียนรู้เชิงลึก โดยนำหลักการมาทดสอบประยุกต์ใช้ด้วยกัน 3 โมเดล ได้แก่ RNN LSTM GAN

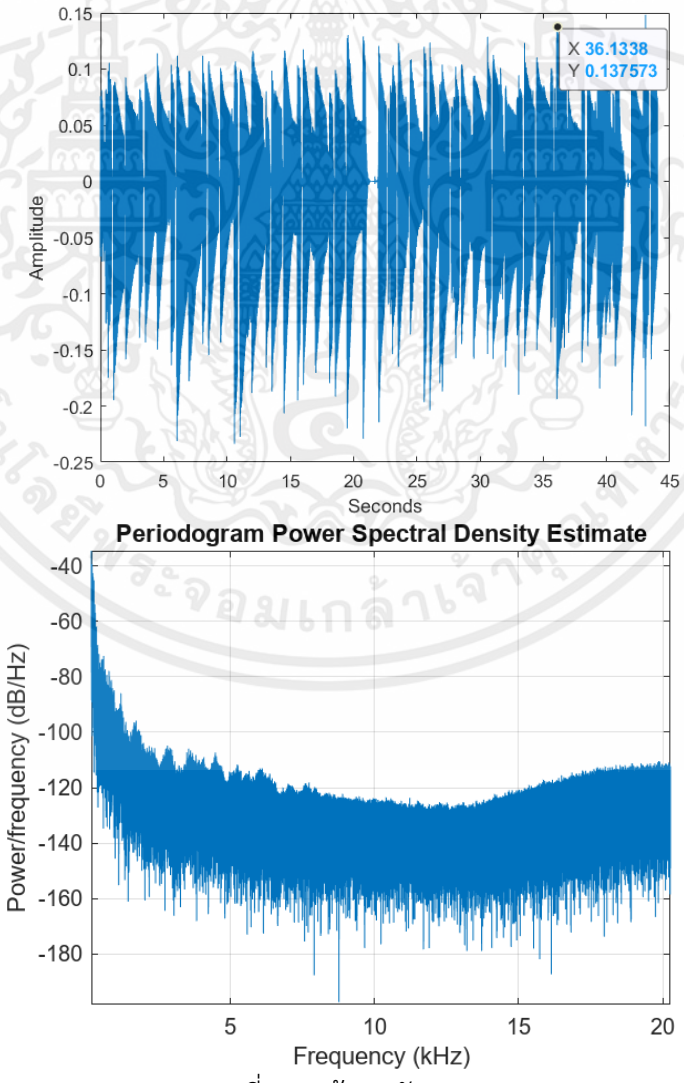
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

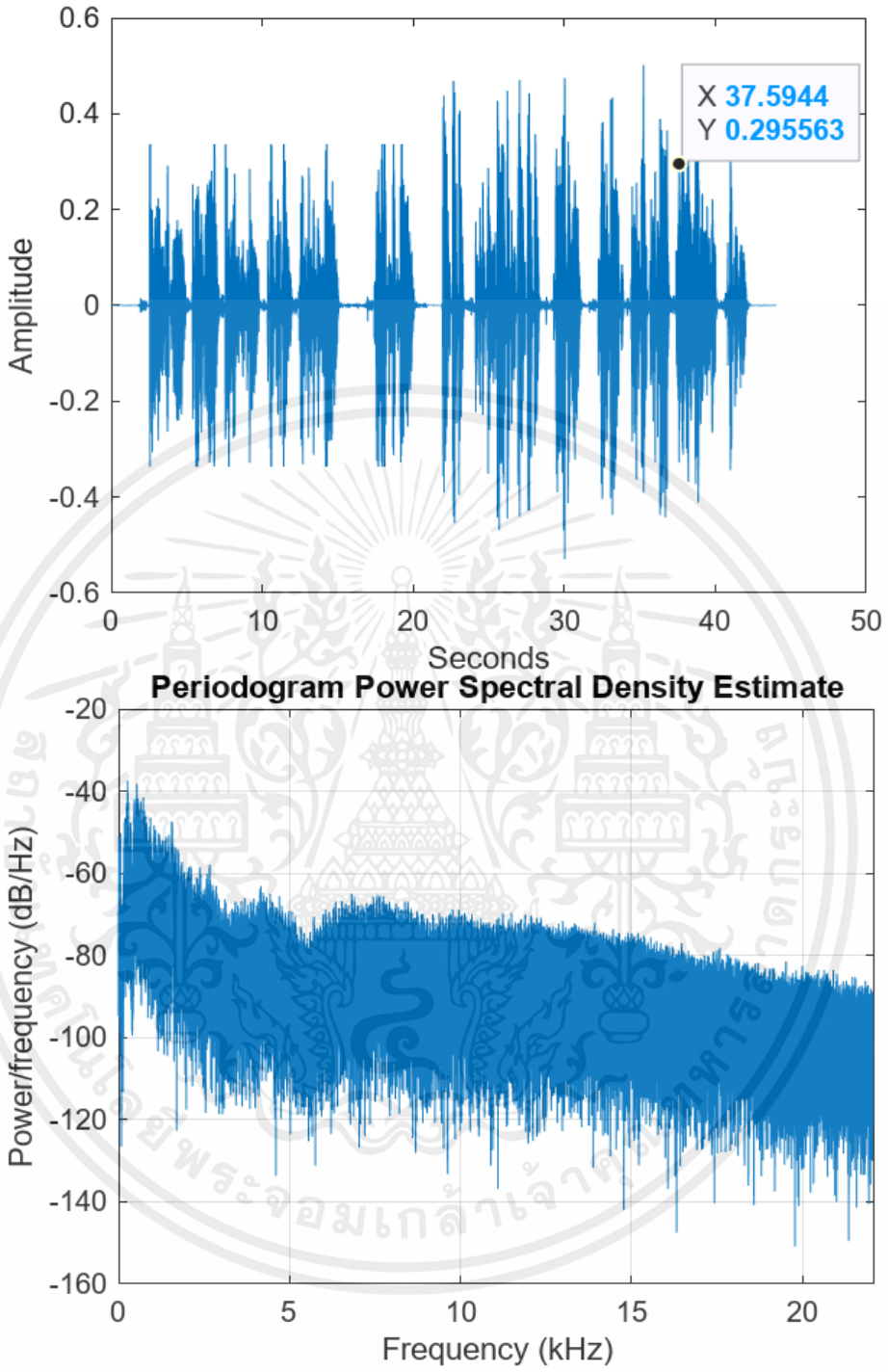
4.1 ผลการดำเนินงาน

ผลการดำเนินงานนั้น ผลลัพธ์ของข้อมูลที่กำหนดขึ้นมาเองนั้น สามารถแสดงผลลัพธ์ได้ แต่ผลลัพธ์จากการใช้ข้อมูลสัญญาณจริงนั้น ผลลัพธ์ไม่มีการแสดงผลออกมา ซึ่งอาจเกิดจากการออกแบบโปรแกรม



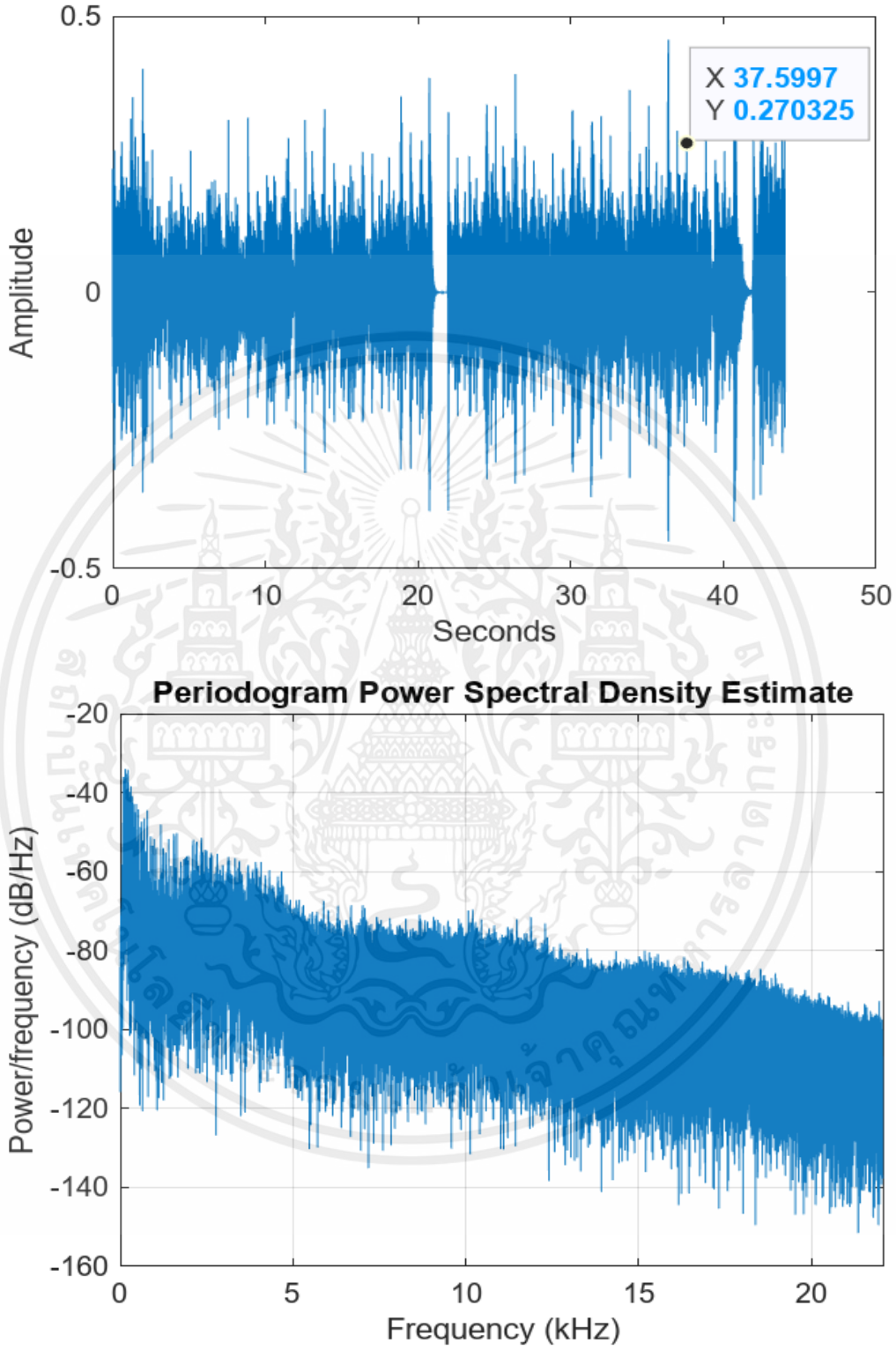
รูปที่ 4.1 ข้อมูลสัญญาณเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



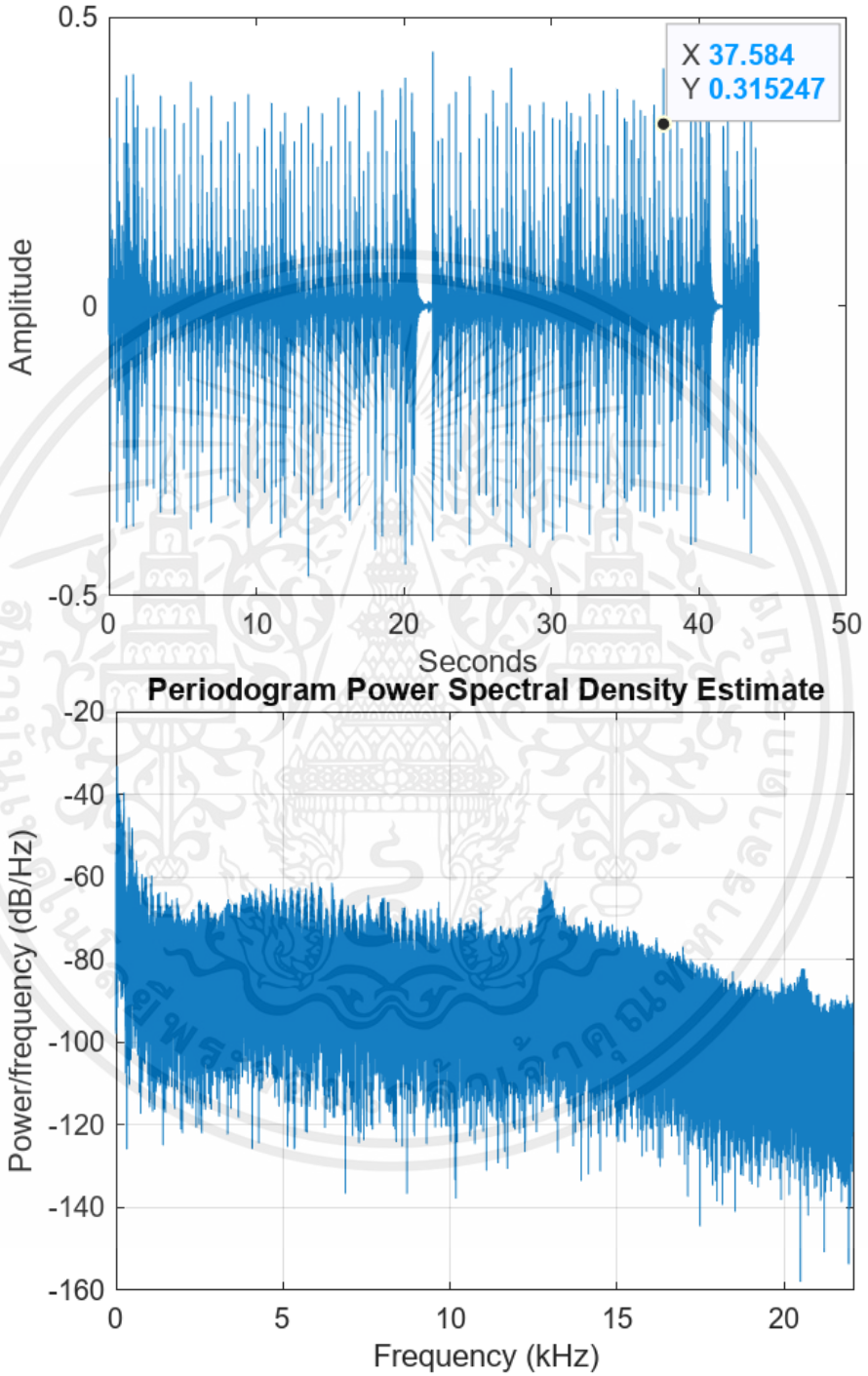
รูปที่ 4.2 ข้อมูลสัญญาณเสียงนักร้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



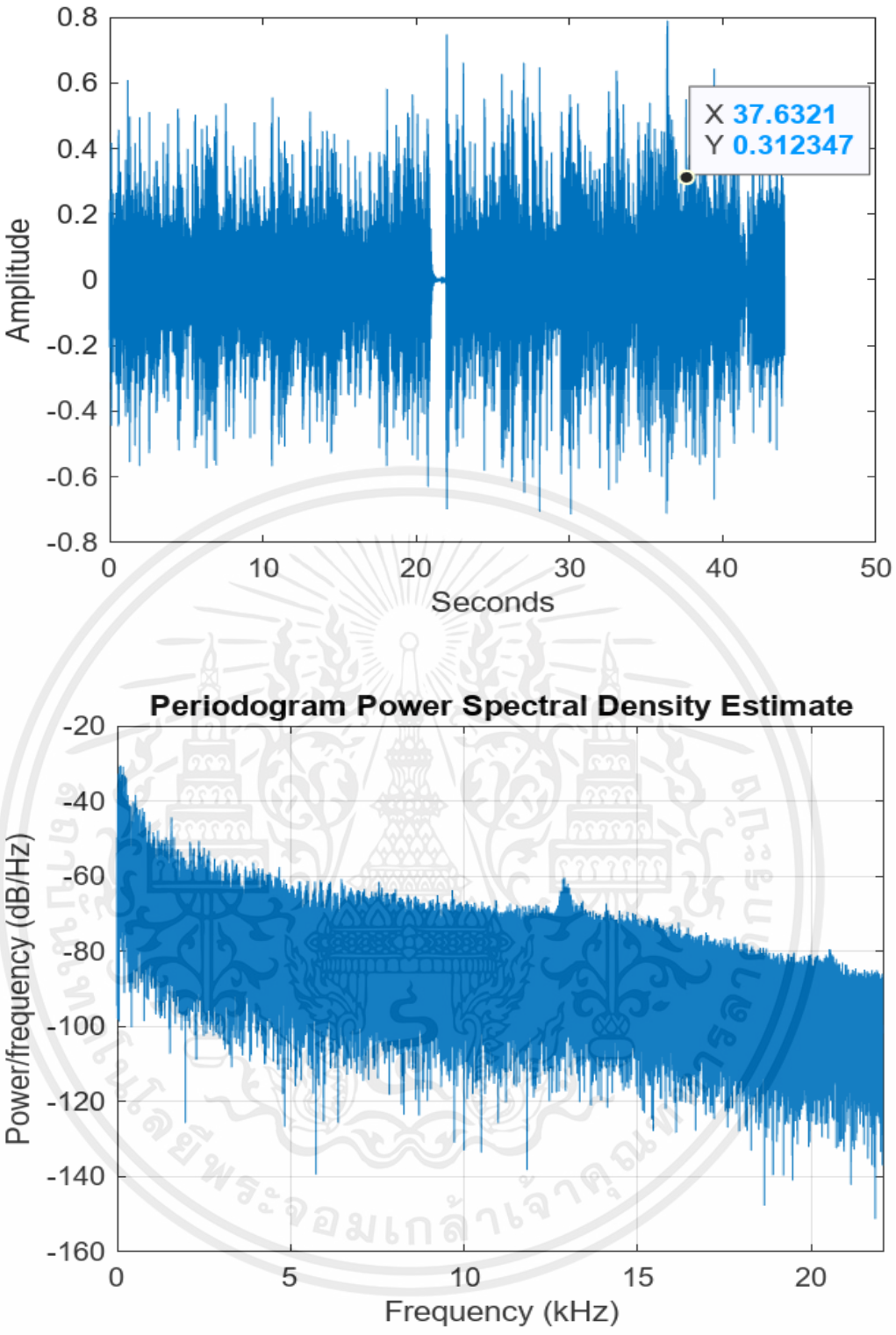
รูปที่ 4.3 ข้อมูลสัญญาณเสียงเครื่องดนตรีอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



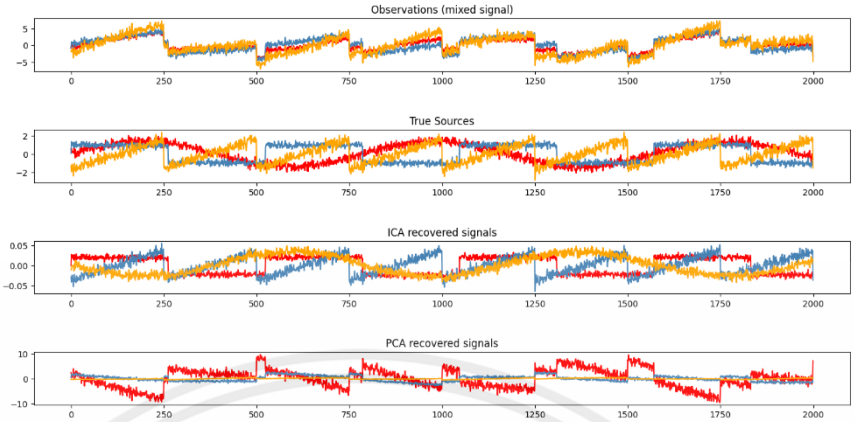
รูปที่ 4.4 ข้อมูลสัญญาณเสียงกลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

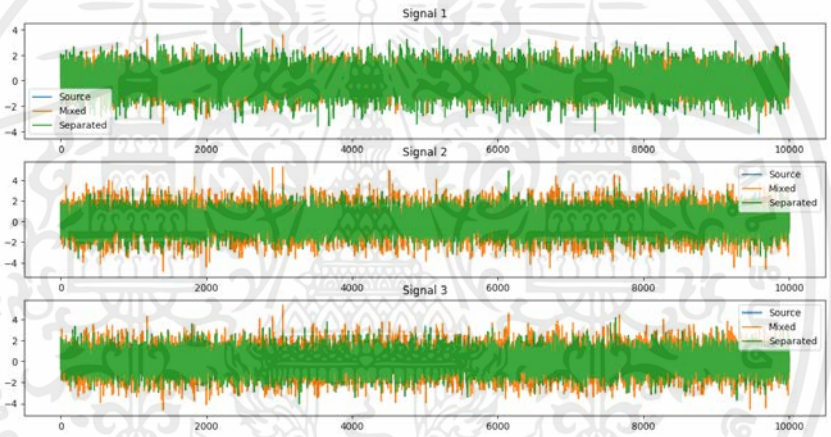


รูปที่ 4.5 ข้อมูลสัญญาณเสียงรวม

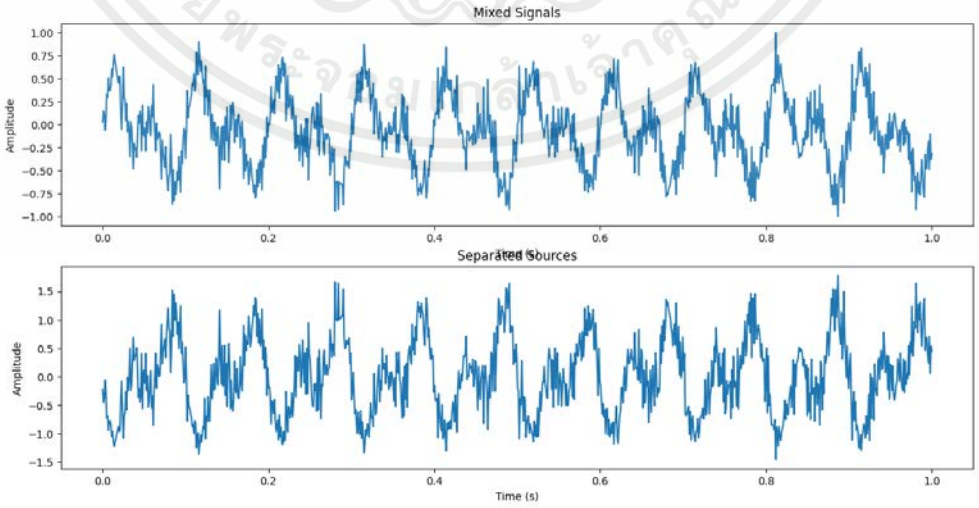
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ผลลัพธ์จากโปรแกรม ICA

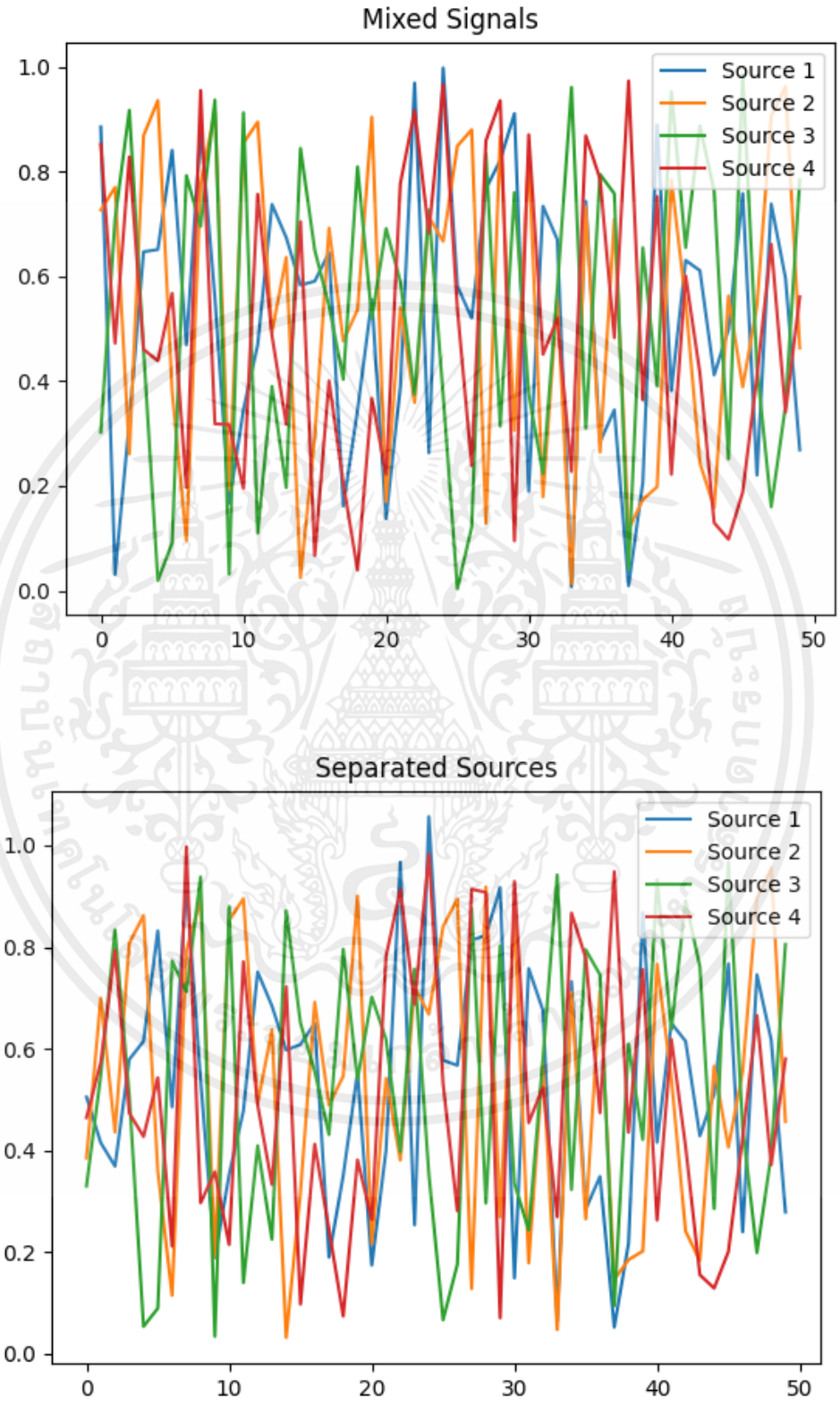


รูปที่ 4.7 ผลลัพธ์จากโปรแกรม LSTM



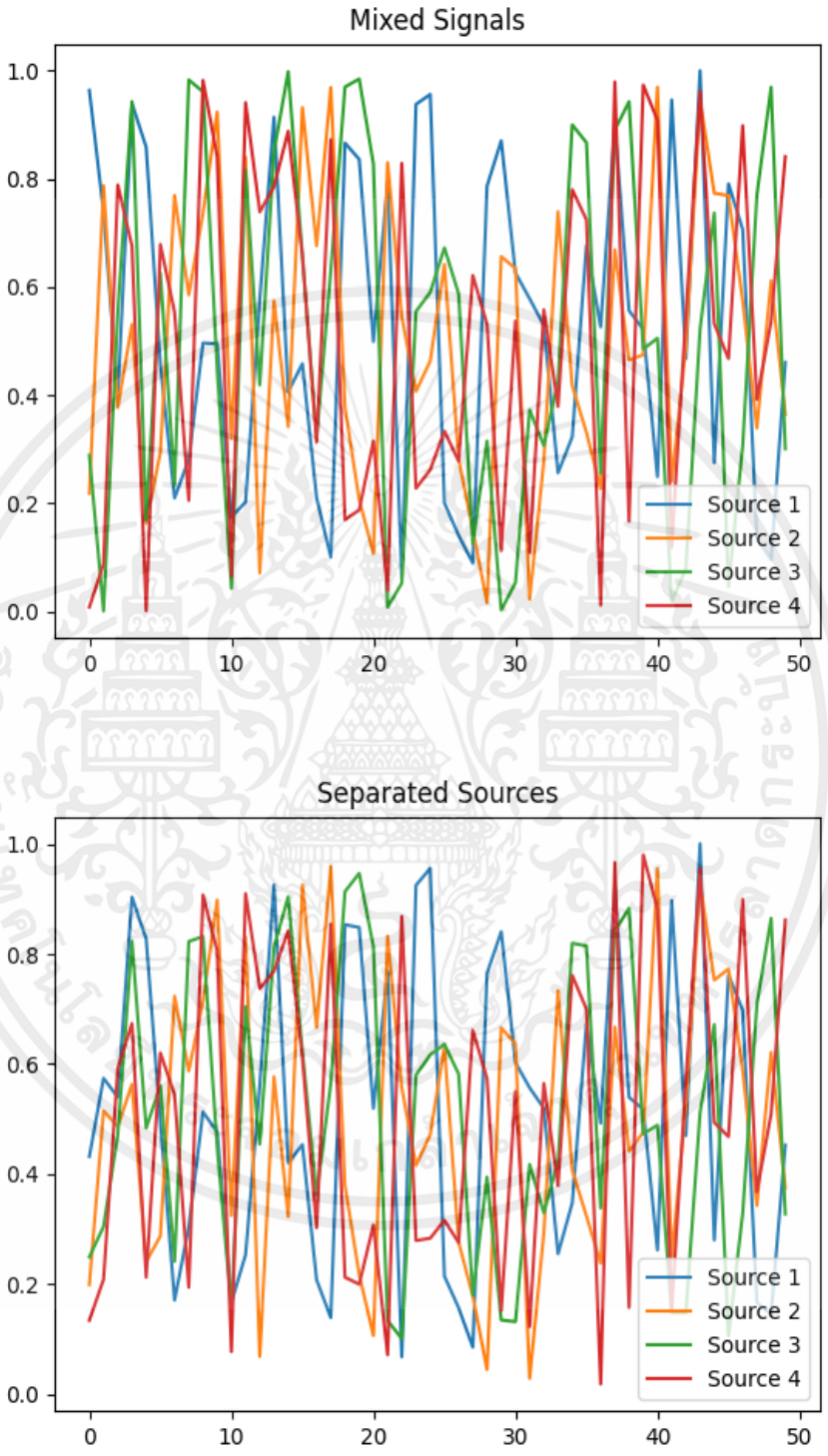
รูปที่ 4.8 ผลลัพธ์จากโปรแกรม GAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผลลัพธ์จากโปรแกรม Deep RNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลลัพธ์จากโปรแกรม RNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากผลการดำเนิน สรุปได้ว่าการแยกสัญญาณแบบไม่รู้แหล่งที่มาในแบบต่างๆนั้น โดยใช้ข้อมูลที่กำหนดขึ้นมาเอง ผลลัพธ์ที่สังเกตได้ค่อนข้างชัดเจนมากไปหาน้อย โดยสังเกตจากการมองเห็น ได้แก่ โปรแกรม ICA, โปรแกรม GAN, โปรแกรม LSTM และโปรแกรม Deep RNN ตามลำดับ โดยทั่งนี้เหตุที่ผลลัพธ์ออกมาลักษณะนี้อาจเกิดจากการออกแบบที่ยังไม่สมบูรณ์ของผู้จัดทำ

5.2 วิจารณ์ผลการทดลอง

ผลลัพธ์ของ DSD Dataset นั้น อาจเรียบเรียงลำดับการทำงานหรือคำสั่งที่ผิดขั้นตอน ส่งผลให้ผลลัพธ์โปรแกรมไม่แสดงผล และข้อมูลอ้างอิงนั้น มีต่อยข้างน้อยเพราะเนื้อหามีความเป็นเอกลักษณ์เฉพาะ จึงมีบททความที่สามารถนำมาประยุกต์ใช้ได้นั้นน้อยตาม

เนื่องจากเริ่มต้นนั้น ผู้จัดทำต้องการนำโครงงานครั้งนี้ไปประยุกต์ใช้กับการแก้ปัญหาของ RFID และการทำ Carrier Recovery ซึ่ง การค้นหาข้อมูลตัวอย่างนั้นค่อนข้างมีน้อยและไม่มีความหลากหลายบในการนำมาใช้เทรนข้อมูล จึงทำให้เสียเวลาในช่วงเริ่มต้นค่อนข้างมาก ส่งผลให้โครงงานนี้มีความก้าวหน้าที่ช้ากว่าเดิมเป็นอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] scikit-learn developers (BSD License), (2007-2023). “Blind source separation using FastICA”. แหล่งข้อมูล:https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_blind_source_separation.html. สืบค้นวันที่ 21 มีนาคม 2566

[2] Sheng Liu, Bangmin Wang and Lanyong Zhang. (2021). “Blind Source Separation Method Based on Neural Network with Bias Term and Maximum Likelihood Estimation Criterion”. แหล่งข้อมูล: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7867157/. สืบค้นวันที่ 21 มีนาคม 2566

[3] ATASPINAR. (2018). “Machine Learning with Signal Processing Techniques”. แหล่งข้อมูล: https://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/. สืบค้นวันที่ 21 มีนาคม 2566

[4] Rafii, Zafar and Liutkus, Antoine and Fabian-Robert St{\"o}ter and Mimilakis, Stylianos Ioannis and Bittner, Rachel. (2017). “The {MUSDB18} corpus for music separation”. แหล่งข้อมูล: https://sigsep.github.io/datasets/musdb.html. สืบค้นวันที่ 1 เมษายน 2566

[5] SiSEC16. (2017). “The 2016 Signal Separation Evaluation Campaign”. แหล่งข้อมูล: https://sigsep.github.io/datasets/dsd100.html. สืบค้นวันที่ 1 เมษายน 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Plot Dataset

```
    y = data(:,1);


    dt = 1/fs;
    t = 0:dt:(length(y)*dt)-dt;
    plot(t,y); xlabel('Seconds'); ylabel('Amplitude');


ax = gca;
chart = ax.Children(1);
datatip(chart,37.59,0.281);
    figure
    plot(psd(spectrum.periodogram,y,'Fs',fs,'NFFT',length(y)));
xlim([0.0 22.1])
ylim("auto")
```

ICA

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from sklearn.decomposition import FastICA, PCA

np.random.seed(0)
n_samples = 2000
time = np.linspace(0, 8, n_samples)

s1 = np.sin(2 * time)  # Signal 1 : sinusoidal signal
s2 = np.sign(np.sin(3 * time))  # Signal 2 : square signal
s3 = signal.sawtooth(2 * np.pi * time)  # Signal 3: saw tooth signal

S = np.c_[s1, s2, s3]
S += 0.2 * np.random.normal(size=S.shape)  # Add noise

S /= S.std(axis=0)  # Standardize data
# Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]])  # Mixing matrix
X = np.dot(S, A.T)  # Generate observations

#ICA
```

เอกสารนี้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ica = FastICA(n_components=3, whiten="arbitrary-variance")

S_ = ica.fit_transform(X)  # Reconstruct signals
A_ = ica.mixing_  # Get estimated mixing matrix

assert np.allclose(X, np.dot(S_, A_.T) + ica.mean_)

# PCA
pca = PCA(n_components=3)
H = pca.fit_transform(X)  # Reconstruct signals based on orthogonal components
plt.figure()
models = [X, S, S_, H]
names = [
    "Observations (mixed signal)",
    "True Sources",
    "ICA recovered signals",
    "PCA recovered signals",
]
colors = ["red", "steelblue", "orange"]

for ii, (model, name) in enumerate(zip(models, names), 1):
    plt.subplot(4, 1, ii)
    plt.title(name)
    for sig, color in zip(model.T, colors):
        plt.plot(sig, color=color)

plt.tight_layout()
plt.show()
```

GAN

```
import numpy as np
from keras.layers import Dense, Reshape, Flatten, Input, Lambda
from keras.models import Model
from keras.optimizers import Adam
from keras.backend import clear_session
from sklearn.preprocessing import MinMaxScaler
from keras import backend as K
import matplotlib.pyplot as plt

# Generate random mixed signals
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
num_samples = 1000
t = np.linspace(0, 1, num_samples)
f1 = 10
f2 = 20
mixed_signals = np.sin(2 * np.pi * f1 * t) + np.sin(2 * np.pi * f2 * t)

# Add noise to the mixed signal
noise_factor = 0.5
mixed_signals += noise_factor * np.random.randn(num_samples)

# Normalize the data between -1 and 1
scaler = MinMaxScaler(feature_range=(-1, 1))
mixed_signals = scaler.fit_transform(mixed_signals.reshape(-1, 1))

# Define the generator network
gen_input = Input(shape=(1,))
gen_hidden = Dense(128, activation='relu')(gen_input)
gen_output = Dense(1, activation='tanh')(gen_hidden)
generator = Model(gen_input, gen_output)

# Define the discriminator network
dis_input = Input(shape=(1,))
dis_hidden = Dense(128, activation='relu')(dis_input)
dis_output = Dense(1, activation='sigmoid')(dis_hidden)
discriminator = Model(dis_input, dis_output)

# Define the combined GAN model
z = Input(shape=(1,))
gen_sample = generator(z)
dis_output = discriminator(gen_sample)
gan = Model(z, dis_output)

# Define the loss functions
def generator_loss(y_true, y_pred):
    return -K.mean(y_pred)

def discriminator_loss(y_true, y_pred):
    return -K.mean(y_true*K.log(y_pred+1e-8)+(1-y_true)*K.log(1-y_pred+1e-8))

# Compile the models
discriminator.compile(optimizer=Adam(lr=0.0002, beta_1=0.5), loss=discriminator_loss)
discriminator.trainable = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
gan.compile(optimizer=Adam(lr=0.0002, beta_1=0.5), loss=generator_loss)

# Train the GAN
num_epochs = 1000
batch_size = 64
for epoch in range(num_epochs):
    # Train discriminator
    real_samples = mixed_signals[np.random.randint(0, mixed_signals.shape[0], size=batch_size), :]
    fake_samples = generator.predict(np.random.uniform(-1, 1, size=(batch_size, 1)))
    X = np.concatenate((real_samples, fake_samples))
    y = np.concatenate((np.ones((batch_size, 1)), np.zeros((batch_size, 1))))
    discriminator_loss = discriminator.train_on_batch(X, y)

    # Train generator
    gan_loss = gan.train_on_batch(np.random.uniform(-1, 1, size=(batch_size, 1)), np.ones((batch_size, 1)))

    # Print the loss
    print('Epoch: {}, Discriminator Loss: {}, GAN Loss: {}'.format(epoch+1, K.eval(discriminator_loss), K.eval(gan_loss)))

# Separate the sources
separated_sources = generator.predict(mixed_signals)

# Denormalize the data
separated_sources = scaler.inverse_transform(separated_sources)

# Plot the original mixed signals and the separated sources
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.plot(t, mixed_signals)
plt.title('Mixed Signals')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.subplot(2, 1, 2)
plt.plot(t, separated_sources)
plt.title('Separated Sources')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.show()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RNN

```
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import LSTM, Dense, TimeDistributed

# Define the input data dimensions
num_samples = 1000
num_timesteps = 50
num_features = 4

# Generate some random input data
X = np.random.rand(num_samples, num_timesteps, num_features)

# Define the BSS-RNN model architecture
model = Sequential()
model.add(LSTM(64, return_sequences=True, input_shape=(num_timesteps, num_features)))
model.add(TimeDistributed(Dense(num_features)))

# Compile the model
model.compile(loss='mse', optimizer='adam')

# Train the model
model.fit(X, X, epochs=10, batch_size=32)

# Use the model to separate the sources from mixed signals
Y = model.predict(X)

# Plot the first example in the input data (mixed signals)
mixed_signal = X[0]
plt.plot(mixed_signal[:,0], label='Source 1')
plt.plot(mixed_signal[:,1], label='Source 2')
plt.plot(mixed_signal[:,2], label='Source 3')
plt.plot(mixed_signal[:,3], label='Source 4')
plt.title('Mixed Signals')
plt.legend()
plt.savefig(f'save/rnnmix.png')
plt.show()

# Plot the separated sources for the first example in the input data
separated_sources = Y[0]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
plt.plot(separated_sources[:,0], label='Source 1')
plt.plot(separated_sources[:,1], label='Source 2')
plt.plot(separated_sources[:,2], label='Source 3')
plt.plot(separated_sources[:,3], label='Source 4')
plt.title('Separated Sources')
plt.legend()
plt.savefig(f'save/rnnsep.png')
plt.show()
```

Deep RNN

```
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import LSTM, Dense, TimeDistributed, Dropout

# Define the input data dimensions
num_samples = 1000
num_timesteps = 50
num_features = 4

# Generate some random input data
X = np.random.rand(num_samples, num_timesteps, num_features)

# Define the BSS-DRNN model architecture
model = Sequential()
model.add(LSTM(64, return_sequences=True, input_shape=(num_timesteps, num_features)))
model.add(Dropout(0.2))
model.add(LSTM(64, return_sequences=True))
model.add(TimeDistributed(Dense(num_features)))

# Compile the model
model.compile(loss='mse', optimizer='adam')

# Train the model
model.fit(X, X, epochs=10, batch_size=32)

# Use the model to separate the sources from mixed signals
Y = model.predict(X)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# Plot the first example in the input data (mixed signals)
mixed_signal = X[0]
plt.plot(mixed_signal[:,0], label='Source 1')
plt.plot(mixed_signal[:,1], label='Source 2')
plt.plot(mixed_signal[:,2], label='Source 3')
plt.plot(mixed_signal[:,3], label='Source 4')
plt.title('Mixed Signals')
plt.legend()
plt.savefig(f'save/drnnmix.png')
plt.show()

# Plot the separated sources for the first example in the input data
separated_sources = Y[0]
plt.plot(separated_sources[:,0], label='Source 1')
plt.plot(separated_sources[:,1], label='Source 2')
plt.plot(separated_sources[:,2], label='Source 3')
plt.plot(separated_sources[:,3], label='Source 4')
plt.title('Separated Sources')
plt.legend()
plt.savefig(f'save/drnnsep.png')
plt.show()
```

LSTM

```
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.decomposition import FastICA

# Set the parameters of the audio signals
n_samples = 10000   # Number of audio samples
n_sources = 3       # Number of audio sources
duration = 5        # Duration of audio signals in seconds
sample_rate = 44100 # Sampling rate of audio signals in Hz

# Generate random audio signals
sources = np.random.randn(n_sources, n_samples)

# Mix the audio signals randomly
mixing_matrix = np.random.rand(n_sources, n_sources)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mixed_signals = np.dot(mixing_matrix, sources)

# Reshape the data for LSTM input
X = mixed_signals.T.reshape(n_samples, 1, n_sources)
y = sources.T.reshape(n_samples, 1, n_sources)

# Build the LSTM model
model = Sequential()
model.add(LSTM(units=128, input_shape=(1, n_sources), return_sequences=True))
model.add(Dense(n_sources))

# Compile the model
model.compile(loss='mse', optimizer='adam')

# Train the model
history = model.fit(X, y, epochs=50, batch_size=32, validation_split=0.1)

# Use the trained model to separate the mixed signals
predicted_signals = model.predict(X).reshape(n_samples, n_sources)

# Plot the results
fig, axs = plt.subplots(n_sources, 1, figsize=(8, 6))
for i in range(n_sources):
    axs[i].plot(sources[i], label='Source')
    axs[i].plot(mixed_signals[i], label='Mixed')
    axs[i].plot(predicted_signals[:, i], label='Separated')
    axs[i].set_title(f'Signal {i+1}')
    axs[i].legend()
plt.show()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้