

การประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน
โดยใช้โครงข่ายประสาทเทียม

State of Charge Estimation of a Lithium Ion Battery
Using the Artificial Neural Network



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

State of Charge Estimation of a Lithium Ion Battery

Using the Artificial Neural Network



THIS PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE BACHELOR DEGREE IN ELECTRICAL ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2565

การประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน

โดยการใช้โครงข่ายประสาทเทียม

State of Charge Estimation of a Lithium Ion Battery

Using the Artificial Neural Network



นางสาวกนกวรรณ

นางสาวกฤษณา

นางสาวกาญจนาพร

นางสาวพิมพ์ทอง

โดย

ตั้งกาญจนกิจ

พันธ์แดง

พูลเกิด

กวางสีบสาม

อาจารย์ที่ปรึกษา

รศ.ดร.สุพัฒน์ กิตติรัตน์สัจจา

ดร.สมภพ ผลไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2565


ภาควิชาวิศวกรรมไฟฟ้า


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเธียมไอออนโดยการใช้โครงข่ายประสาทเทียม

ผู้จัดทำ

1. นางสาวกนกวรรณ ตั้งกาญจนกิจ
2. นางสาวกฤษณา พันธุ์แดง
3. นางสาวกาญจนพร พูลเกิด
4. นางสาวพิมพ์ทอง กวารสิบสาม


..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.สุพัฒน์ กิตติรัตน์สัจจา)


..... อาจารย์ที่ปรึกษาร่วม
(ดร.สมภพ ผลไม้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน
โดยการใช้โครงข่ายประสาทเทียม

นางสาวกนกวรรณ	ตั้งกาญจนกิจ	
นางสาวกฤษณา	พันธ์แดง	
นางสาวกาญจนาวพร	พูลเกิด	
นางสาวพิมพ์ทอง	กาวาสืบสาม	
รศ.ดร.สุพัฒน์	กิตติรัตน์สัจจา	อาจารย์ที่ปรึกษา
ดร.สมภพ	ผลไม้	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2565

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอเกี่ยวกับการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมเพื่อการประมาณสถานะที่แม่นยำและทำให้แบตเตอรี่มีอายุการใช้งานที่ยาวนานขึ้น โดยในปริญญานิพนธ์ฉบับนี้เสนอการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM) , Feed-forward Neural Network (FNN) และ Recurrent Neural Network (RNN) โดยการใช้โปรแกรม MATLAB ซึ่งกำหนดสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้วกระแสไฟฟ้าและอุณหภูมิ และกำหนดสัญญาณขาออกคือ สถานะประจุ และมีการกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับและมีการทดลองใช้ชุดข้อมูลที่ซ้ำกันและสลับชุดข้อมูล โดย WLTP ย่อมาจาก Worldwide Harmonized Light Vehicle Test Procedure ซึ่งเป็นมาตรฐานการวัดระยะทางรถยนต์ไฟฟ้า ในการประเมินประสิทธิภาพของการประมาณสถานะประจุโดยโครงข่าย LSTM FNN และ RNN ซึ่งพิจารณาจากค่า Root Mean Squared Error (RMSE) พบว่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุดและรองลงมาคือ SoC ที่ได้จากโครงข่าย RNN และ FNN ตามลำดับ ซึ่งแสดงให้เห็นถึงความแม่นยำในการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

State of Charge Estimation of a Lithium Ion Battery Using the Artificial Neural Network

Kanokwan	Tanganjanakit
Kritsana	Pantang
Kanchanapoon	Poonkoed
Pimtong	Kwaosibsam
Asspc.Prof.Dr.Supat	Kittiratsatcha Advisor
Dr.Sompob	Polmai Co-Advisor

ABSTRACT

This thesis presents the estimation of the state of charge of lithium-ion batteries using artificial neural networks for accurate state estimation and longer battery life. The thesis proposes the estimation of the state of charge of a lithium-ion battery through the utilization of Long Short-Term Memory (LSTM) neural network, Feed-forward Artificial Neural Network (FNN), and Recurrent Neural Network (RNN). The estimation is implemented through a MATLAB program, where the input values entered into the network consist of three variables: inter-polar voltage, current, and temperature. The output is set to the state of charge (SoC).

The training set and testing set are defined as WLTP Class 3 and WLTP Class 2, respectively. Additionally, duplicate, and alternating datasets were also experimented with. WLTP stands for Worldwide Harmonized Light Vehicle Test Procedure, which is an electric vehicle mileage measurement standard. To evaluate the efficiency of charge state estimation by LSTM, FNN, and RNN networks, the Root Mean Squared Error (RMSE) was considered. It was found that the SoC obtained from the LSTM network was the most accurate, followed by the SoC obtained from the RNN and FNN networks, respectively. This finding demonstrates the potential for estimating the state of charge of lithium-ion batteries using the Long Short-Term Memory (LSTM) neural network.

กิตติกรรมประกาศ

ปริญญาโทเล่มนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความเมตตาและความช่วยเหลือจากท่านอาจารย์ที่ปรึกษาและอาจารย์ที่ปรึกษาร่วม กราบขอบพระคุณ รองศาสตราจารย์ ดร.สุพัฒน์ กิตติรัตน์สัจจา และ ดร.สมภพ ผลไม้ ที่ได้กรุณารับเป็นอาจารย์ที่ปรึกษาและเสียสละเวลาอันมีค่าในการให้คำแนะนำและการชี้แนะแนวทางในการทำให้ปริญญาโทเล่มนี้เป็นไปในขอบเขตที่คณะผู้จัดทำหวังไว้ ตลอดจนการตรวจสอบแก้ไขความเรียบร้อยของปริญญาโทเล่มนี้เป็นอย่างดี

ขอขอบพระคุณคณะอาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ให้ความช่วยเหลือและให้ คำแนะนำแนวทางในการทำปริญญาโทให้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ นายกฤตยชญ์ หรุพานวงษ์ และนายอัศววัฒน์ ไพโรสันต์ นักศึกษาปริญญาโท ที่ให้ความช่วยเหลือและคอยให้คำแนะนำมาโดยตลอด

สุดท้ายนี้ คณะผู้จัดทำขอขอบพระคุณบิดามารดาและครอบครัว ผู้ซึ่งเปิดโอกาสให้ได้รับการศึกษา ตลอดจนให้การสนับสนุน ช่วยเหลือและให้กำลังใจระหว่างการดำเนินรวมทั้งเพื่อน ๆ ที่ช่วยเหลือทางความคิดเห็นช่วยแนะนำแนวทางที่เป็นประโยชน์ต่อปริญญาโทเล่มนี้จึงขอขอบคุณมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตและข้อกำหนดของโครงการ	2
1.4 วิธีการที่ใช้ในโครงการ	2
1.5 แผนการดำเนินโครงการ	3
1.6 ประโยชน์ที่ได้รับจากการวิจัย	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 แบตเตอรี่	5
2.2 แบตเตอรี่ลิเธียมไอออน	5
2.2.1 โครงสร้างของแบตเตอรี่ลิเธียมไอออน	6
2.2.2 หลักการทำงานของแบตเตอรี่ลิเธียมไอออน	7
2.3 วิธีการประมาณสถานะประจุ	8
2.3.1 วิธีการประมาณสถานะประจุของแบตเตอรี่	8
2.4 โครงข่ายประสาทเทียม	10
2.4.1 โครงสร้างของโครงข่ายประสาทเทียม	10
2.4.2 ประเภทของโครงข่ายประสาทเทียม	11
2.5 The Worldwide harmonized Light vehicles Test Procedure (WLTP)	14
2.6 โปรแกรม MATLAB	15
บทที่ 3 การออกแบบและสร้างโปรแกรมในการประมาณสถานะประจุของแบตเตอรี่	17
3.1 Input Profile	18

สารบัญ (ต่อ)

	หน้า
3.2 แผนผัง และ Code ในการประมาณสถานะประจุของแบตเตอรี่	20
3.2.1 แผนผัง Feed-forward neural network	20
3.2.2 Code Feed-forward neural network	23
3.2.3 แผนผัง Recurrent neural network	27
3.2.4 Code Recurrent neural network	30
3.2.5 แผนผัง Long short-term memory neural network	34
3.2.6 Code Long short-term memory neural network	37
บทที่ 4 ผลการทดลอง	41
4.1 ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	41
4.2 ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	44
4.3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	47
4.4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	50
บทที่ 5 สรุปผลการทดลอง	54
5.1 สรุปผลการทดลอง	55
5.2 ปัญหาอุปสรรคและแนวทางแก้ไข	55
5.3 แนวทางการพัฒนาต่อ	55
เอกสารอ้างอิง	56
ภาคผนวก	59
ประวัติผู้เขียน	69

สารบัญรูปภาพ

ภาพที่	หน้า	
2.1	โครงสร้างของแบตเตอรี่ลิเทียมไอออน	7
2.2	หลักการทำงานของแบตเตอรี่ลิเทียมไอออน	8
2.3	โครงข่ายประสาทเทียม (Artificial neural network : ANN)	10
2.4	โครงข่ายประสาทเทียม (Artificial neural network : ANN)	11
2.5	โครงข่ายประสาทที่คำนวณแบบไปข้างหน้า (feed-forward artificial neural network)	12
2.6	โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent neural network)	12
2.7	โครงสร้าง Long Short-Term Memory	14
2.8	โลโก้ของโปรแกรม MATLAB	15
3.1	ลักษณะการประมาณสถานะประจุของแบตเตอรี่โดยการใช้โครงข่ายประสาท เทียมชนิด FNN, RNN และ LSTM	17
3.2	กราฟกระแสไฟฟ้าและแรงดันไฟฟ้าระหว่างขั้วของโปรไฟล์อินพุต	18
4.1	กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	41
4.2	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	42
4.3	กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	42
4.4	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	43
4.5	กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	43
4.6	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2	44
4.7	กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	44

สารบัญรูปภาพ (ต่อ)

ภาพที่	หน้า	
4.8	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	45
4.9	กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	45
4.10	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	46
4.11	กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	46
4.12	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3	47
4.13	กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	47
4.14	กราฟการทำนาย SoC และ เปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	48
4.15	กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	48
4.16	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	49
4.17	กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	49
4.18	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2	50
4.19	กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	50

สารบัญรูปภาพ (ต่อ)

ภาพที่		หน้า
4.20	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	51
4.21	กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	51
4.22	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	52
4.23	กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	52
4.24	กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3	53

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินโครงการ เดือนสิงหาคม 2565 ถึง เดือนธันวาคม 2565	3
1.2 แผนการดำเนินโครงการ เดือนมกราคม 2566 ถึง เดือนเมษายน 2566	3
2.1 ข้อดีและข้อเสียของแบตเตอรี่ชนิดลิเทียมไอออน	6
4.1 สูตรค่า Root Mean Squared Error (RMSE) ของแต่ละกรณี	53



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันรถยนต์ไฟฟ้า (Electric Vehicles : EV) เป็นที่นิยมและได้รับความสนใจมากขึ้น โดยเฉพาะอย่างยิ่งในการนำมาใช้แทนรถยนต์ที่ใช้เครื่องยนต์แบบสันดาปภายใน (Internal Combustion Engine : ICE) เนื่องจากเป็นมิตรกับสิ่งแวดล้อมและลดมลพิษทางอากาศและประกอบกับในปัจจุบันประเทศไทยเริ่มให้ความสำคัญกับพลังงานทางเลือกมากขึ้นเพราะมองเห็นถึงความสำคัญของสถานการณ์ภาวะโลกร้อนและมลพิษของประเทศไทยที่มีผลต่อสังคมและชุมชนซึ่งเกิดขึ้นเพิ่มมากขึ้นทุกปี เหตุผลหนึ่งของสถานการณ์ภาวะโลกร้อนมาจากกิจกรรมการเผาผลาญเชื้อเพลิงฟอสซิล (Fossil) ไม่ว่าจะเป็นน้ำมัน ถ่านหิน หรือก๊าซธรรมชาติ ด้วยเหตุนี้รถยนต์ไฟฟ้าจึงเข้ามามีบทบาทสำคัญต่ออุตสาหกรรมยานยนต์ทั้งในปัจจุบันและอนาคต ซึ่งส่วนประกอบที่สำคัญที่สุดของรถยนต์ไฟฟ้าคือแบตเตอรี่ โดยแบตเตอรี่ที่นิยมใช้ในรถยนต์ไฟฟ้ามีทั้งสิ้น 3 ประเภทคือ แบตเตอรี่ตะกั่ว แบตเตอรี่นิกเกิล และแบตเตอรี่ลิเทียม โดยแบตเตอรี่ลิเทียมมีคุณสมบัติที่เหมาะสมกับการนำมาใช้งานในยานยนต์ไฟฟ้ามากที่สุดเนื่องจากคุณสมบัติด้านความหนาแน่น พลังงานที่สูง และในส่วนประกอบของธาตุลิเทียมนั้นมีเซลล์ไฟฟ้าเคมีที่สูงกว่าเซลล์จากโลหะอื่น เป็นสาเหตุให้แบตเตอรี่มีแรงดันไฟฟ้าที่สูงกว่าแบตเตอรี่ชนิดอื่น อีกทั้งแบตเตอรี่ลิเทียมมีขนาดและน้ำหนักน้อยกว่าแบตเตอรี่ตะกั่วและแบตเตอรี่นิกเกิลที่มีความจุเท่ากัน [1]

ดังนั้นจึงจำเป็นต้องศึกษาค้นคว้าพฤติกรรมที่ไม่เป็นเชิงเส้นของแบตเตอรี่เพื่อพัฒนาประสิทธิภาพการทำงานของแบตเตอรี่สำหรับรถยนต์ไฟฟ้าให้ดียิ่งขึ้น ซึ่งระบบการจัดการแบตเตอรี่ (Battery Management System : BMS) เป็นสิ่งจำเป็นในการรองรับความปลอดภัยสำหรับแบตเตอรี่ โดยฟังก์ชันที่สำคัญอย่างหนึ่งของระบบการจัดการแบตเตอรี่คือการประมาณสถานะประจุของแบตเตอรี่ เพื่อตรวจสอบว่าแบตเตอรี่ยังคงเหลือประจุให้ใช้อีกมากน้อยเพียงใดในรถยนต์ไฟฟ้า

โดยในปัจจุบัน Artificial Intelligence ได้เข้ามามีบทบาทและเป็นที่ยอมรับในด้านต่าง ๆ อาทิเช่นด้านการแพทย์ ด้านอุตสาหกรรม เป็นต้น โดยในปริญญาโทฉบับนี้ Artificial intelligence ถูกนำมาใช้สำหรับการประมาณสถานะประจุของแบตเตอรี่ โดย Artificial Neural Network (ANN) เลียนแบบการทำงานของระบบประสาทของมนุษย์จะจำลองการทำงานเหมือนกลุ่มเซลล์ประสาทที่เชื่อมโยงกัน และเนื่องจากชุดข้อมูลที่ป้อนเข้าสู่โครงข่ายเป็นชุดข้อมูลลำดับที่มีความยาวมาก จึงใช้ ANN ชนิดหนึ่งที่มีชื่อว่า Recurrent Neural Network (RNN) [2] ในการประมาณสถานะประจุ ซึ่ง RNN มีโครงสร้างคล้ายกับ ANN ปกติ แต่มีข้อแตกต่างคือมีการเก็บค่าสถานะก่อนหน้าเอาไว้ใช้สำหรับป้อนเข้าสู่โครงข่ายสถานะปัจจุบัน อย่างไรก็ตาม RNN ไม่สามารถประมาณสถานะประจุให้มีความแม่นยำได้เพียงพอ เนื่องจากไม่มีความทรงจำระยะยาวทำให้ลืมข้อมูลในช่วงแรก จึงมีการเสนอใช้

Long short-term memory (LSTM) ซึ่งที่มีประสิทธิภาพทางการคำนวณมากกว่าสามารถจัดการข้อมูลในหน่วยความจำได้เป็นเวลานานกว่าซึ่งเป็น Recurrent Neural Network (RNN) ชนิดหนึ่งที่มีทั้งความทรงจำระยะสั้นและระยะยาวทำให้สถานะประจุกู้ที่ได้จากการประมาณมีความแม่นยำมากยิ่งขึ้น [3]

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาโครงสร้างของโครงข่ายประสาทเทียม (Artificial neural network)
2. เพื่อศึกษาแบบจำลองการทำงานและคุณสมบัติของแบตเตอรี่ชนิดลิเธียมไอออน
3. เพื่อออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเธียมไอออนโดยใช้โครงข่ายประสาทเทียม และเปรียบเทียบโครงข่ายประสาทเทียม 3 ชนิด ได้แก่ Feed-forward Neural Network (FNN), Recurrent Neural Network (RNN) และ Long Short-Term Memory (LSTM) จากโปรแกรม MATLAB

1.3 ขอบเขตและข้อกำหนดของโครงการ

1. ศึกษาการประมาณสถานะประจุของแบตเตอรี่และหลักการทำงานของโครงข่ายประสาทเทียม
2. ออกแบบและสร้างโปรแกรมในการประมาณสถานะประจุของแบตเตอรี่

1.4 วิธีการที่ใช้ในโครงการ

1. ศึกษาทฤษฎี หลักการ และบทความทางวิชาการที่เกี่ยวข้องกับการประมาณค่าสถานะประจุ แบตเตอรี่ชนิดลิเธียมไอออนโดยการใช้โครงข่ายประสาทเทียม
2. ศึกษาวิธีการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเธียมไอออนโดยการใช้โครงข่ายประสาทเทียมด้วยชนิด Long Short-Term Memory (LSTM)
3. ศึกษาการใช้โปรแกรม MATLAB ในการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเธียมไอออน
4. ออกแบบโปรแกรมในการประมาณค่าสถานะประจุของแบตเตอรี่
5. สร้างโปรแกรมในการประมาณค่าสถานะประจุของแบตเตอรี่
6. สรุปผลการวิจัย ข้อเสนอแนะ และจัดทำรูปเล่มปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 แผนการดำเนินโครงการ

ตารางที่ 1.1 แผนการดำเนินโครงการ เดือนสิงหาคม 2565 ถึง เดือนธันวาคม 2565

หัวข้องานดำเนินโครงการ	เดือน	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาโครงข่ายประสาทเทียมจากกล่องเครื่องมือในโปรแกรม MATLAB		↔				
2. ศึกษาโครงข่ายประสาทเทียมแบบ Feed-forward			↔			
3. ศึกษาการ fitting ข้อมูล			↔			
4. ศึกษาโครงข่ายประสาทเทียมแบบ Recurrent				↔		
5. ศึกษาการ Prediction สถานะการชาร์จแบตเตอรี่โดยใช้โครงข่ายประสาทเทียม				↔	↔	

ตารางที่ 1.2 แผนการดำเนินโครงการ เดือนมกราคม 2566 ถึง เดือนเมษายน 2566

หัวข้องานดำเนินโครงการ	เดือน	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
6. ออกแบบการประมาณค่า SoC ของแบตเตอรี่ลิเทียมไอออนโดยใช้ Long Short-Term Memory (LSTM)		↔	↔			
7. ออกแบบการประมาณค่า SoC ของแบตเตอรี่ลิเทียมไอออนโดยใช้ Feed-forward Neural Network (FNN)			↔	↔		
8. ออกแบบการประมาณค่า SoC ของแบตเตอรี่ลิเทียมไอออนโดยใช้ Recurrent Neural Network (RNN)			↔	↔		
9. นำโปรแกรมที่สร้างไว้มาทำการทดสอบ โดยมีการทดสอบทั้งหมด 4 กรณี				↔	↔	
10. จัดทำรูปเล่มปริญญานิพนธ์						↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์ที่ได้รับจากการวิจัย

1. สามารถประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้การจำลองโครงข่ายประสาทเทียม (Artificial neural network) จากโปรแกรม MATLAB
2. องค์ความรู้เกี่ยวกับโครงข่ายประสาทเทียม (Artificial neural network)
3. งานวิจัยนี้เป็นการศึกษาการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียม โดยการบูรณาการความรู้ทางวิศวกรรมกับระบบโครงข่ายประสาทเทียม ซึ่งผลลัพธ์ของงานวิจัยนี้สามารถนำไปประยุกต์ใช้งานกับแบตเตอรี่ยานยนต์ไฟฟ้า (Electric vehicle : EV)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 แบตเตอรี่

แบตเตอรี่ถือเป็นอุปกรณ์ทางไฟฟ้าที่สำคัญชิ้นหนึ่งของยานยนต์ไฟฟ้า เนื่องจากเป็นอุปกรณ์ที่ใช้ปฏิกิริยาทางเคมีไฟฟ้ากักเก็บพลังงานที่ใช้ในการขับเคลื่อน โดยส่วนประกอบหลักของแบตเตอรี่ประกอบด้วยส่วนหลัก ๆ 3 ส่วนคือ ขั้วบวก (แคโทด) ขั้วลบ (แอโนด) และสารละลายอิเล็กโทรไลต์ ซึ่งเมื่อแบตเตอรี่ถูกใช้งานหรือเกิดการคายประจุไฟฟ้าขั้วลบหรือขั้วแอโนดจะเป็นขั้วที่เกิดปฏิกิริยาเคมี ส่งผลให้เกิดการเคลื่อนที่ของอิเล็กตรอน (ปฏิกิริยาออกซิเดชัน) และขั้วไฟฟ้าบวกหรือขั้วแคโทดเป็นขั้วที่เกิดปฏิกิริยาเคมีที่รับอิเล็กตรอน (ปฏิกิริยารีดักชัน) โดยอิเล็กตรอนที่ได้จากปฏิกิริยาออกซิเดชัน จะวิ่งผ่านลวดโลหะก่อให้เกิดการไหลของกระแสไฟฟ้าขึ้น โดยอิเล็กโทรไลต์จะต้องเป็นสารละลายที่ไม่นำอิเล็กตรอน แต่มีหน้าที่ส่งผ่านไอออนที่จำเป็นต่อการเกิดปฏิกิริยาดังกล่าวได้อย่างต่อเนื่องจนกว่าแบตเตอรี่จะหมด

แบตเตอรี่แบ่งเป็น 2 ประเภทหลัก คือ แบตเตอรี่ปฐมภูมิ (Primary battery) เป็นแบตเตอรี่ที่เมื่อผ่านการใช้งานแล้วจะไม่สามารถนำกลับมาชาร์จเพื่อนำมาใช้อีกครั้งได้ เช่น แบตเตอรี่แบบอัลคาไลน์ หรือแบตเตอรี่แบบลิเทียม หรือที่เรียกด้วยคำทั่วไปว่า "ถ่าน" สำหรับใช้ในวิทยุ นาฬิกา หรือรีโมทโทรทัศน์ และแบตเตอรี่ทุติยภูมิ (Secondary battery) เป็นแบตเตอรี่ที่เมื่อผ่านการใช้งานแล้วสามารถนำกลับมาชาร์จเพื่อนำกลับมาใช้งานได้ใหม่ ตัวอย่างเช่น แบตเตอรี่ลิเทียมไอออน (Li-ion) ที่ใช้ในโทรศัพท์เคลื่อนที่หรือโน้ตบุ๊กคอมพิวเตอร์ หรือแบตเตอรี่รถยนต์ [4]

2.2 แบตเตอรี่ลิเทียมไอออน (Lion Battery)

แบตเตอรี่ลิเทียมไอออนเป็นแบตเตอรี่ที่นิยมใช้ในรถยนต์ไฟฟ้ามากที่สุดเนื่องจากมีสมบัติเป็นโลหะที่มีน้ำหนักเบาที่สุด นอกจากนี้ลิเทียมยังมีศักย์ไฟฟ้าทางไฟฟ้าเคมีที่มาก รวมถึงการทำปฏิกิริยาทางเคมีที่มีความว่องไวจากสมบัติต่าง ๆ ของลิเทียม เมื่อนำมาทำเป็นแบตเตอรี่จึงได้แบตเตอรี่ที่เก็บพลังงานได้สูงและจ่ายพลังงานได้มาก เนื่องจากโลหะลิเทียมนี้มีความไวต่อการทำปฏิกิริยาเคมีในธรรมชาติจึงไม่พบลิเทียมในรูปของโลหะบริสุทธิ์ แต่มักพบลิเทียมในรูปของสารประกอบ นอกจากนี้แล้วโลหะลิเทียมยังทำปฏิกิริยารุนแรงกับน้ำ ดังนั้นเมื่อนำโลหะลิเทียมมาทำแบตเตอรี่จึงทำให้แบตเตอรี่ชนิดนี้มีความเสี่ยงสูง โดยเฉพาะถ้าปิดผนึกแบตเตอรี่ไม่ดีพอจะทำให้อากาศหรือความชื้นจากภายนอกสามารถหลุดเข้าไปในเซลล์ของแบตเตอรี่ได้ เมื่อโลหะลิเทียมสัมผัสกับน้ำหรือความชื้นในอากาศจะทำให้เกิดการทำให้ปฏิกิริยาเคมีกันจนอาจเกิดการระเบิดหรือเกิดการลุกเป็นไฟได้ ในยุคแรก ๆ เคยเกิดเหตุระเบิดขึ้นจนบริษัทผู้ผลิตต้องเลิกผลิตและหันไปพัฒนาแบตเตอรี่ที่ใช้ลิเทียมในรูปไอออน ซึ่งมีความปลอดภัยมากกว่า โดยข้อดีและข้อเสียของแบตเตอรี่ชนิดลิเทียมไอออนสามารถสรุปได้ในตาราง 2.1 [5,6]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 ข้อดีและข้อเสียของแบตเตอรี่ชนิดลิเทียมไอออน

ข้อดี	ข้อเสีย
1. ความหนาแน่นของพลังงานและกำลังไฟฟ้าสูง 2. มีน้ำหนักเบา 3. มีอัตราการคายประจุตัวเองต่ำเมื่อเทียบกับแบตเตอรี่ชนิดอื่น ๆ	1. ต้องใช้วงจรป้องกันแรงดันและกระแสให้อยู่ในระดับที่ปลอดภัย 2. มีการเสื่อมอายุตามเวลาแม้ว่าจะไม่มีการใช้งาน แบตเตอรี่ลิเทียมมีอายุการใช้งานประมาณ 2-3 ปี 3. แบตเตอรี่ลิเทียมไอออนมีต้นทุนการผลิตและการผลิตที่สูงกว่า ทำให้มีราคาแพงกว่า แบตเตอรี่นิกเกิลแคดเมียมเกือบ 40%

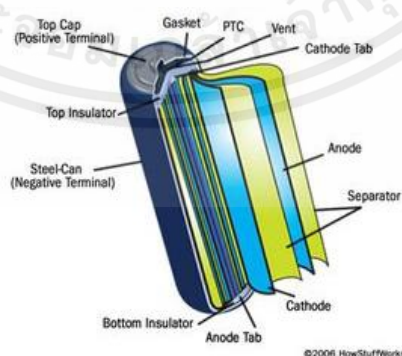
2.2.1 โครงสร้างของแบตเตอรี่ลิเทียมไอออน

2.2.1.1 ขั้วไฟฟ้าจะประกอบไปด้วยขั้วแคโทด (Cathode) เป็นส่วนที่ไว้กักเก็บพลังงานขึ้นอยู่กับวัสดุที่นำมาใช้ทำและอีกส่วนคือขั้วแอโนด (Anode) ส่วนมากจะทำมาจากวัสดุประเภทคาร์บอน เช่น แกรไฟต์

2.2.1.2 Separator เป็นส่วนที่ป้องกันไม่ให้ขั้วแคโทดสัมผัสกับขั้วแอโนดเพราะจะทำให้เกิดการลัดวงจรได้

2.2.1.3 อิเล็กโทรไลต์ (Electrolyte) เป็นสารละลายเกลือของลิเทียมซึ่งเป็นตัวนำไฟฟ้าที่ยอมให้ไอออนไหลผ่านแต่ ไม่ยอมให้อิเล็กตรอนไหลผ่าน จึงเป็นตัวนำไอออนที่ดีแต่เป็นตัวนำอิเล็กตรอนที่ไม่ดี

2.2.1.4 Current collector เป็นส่วนโลหะตัวนำที่ทำหน้าที่ให้อิเล็กตรอนไหลผ่านออกสู่วงจรภายนอกและเกิดการนำพลังงานไฟฟ้าจนสามารถนำไปใช้ได้ [7]

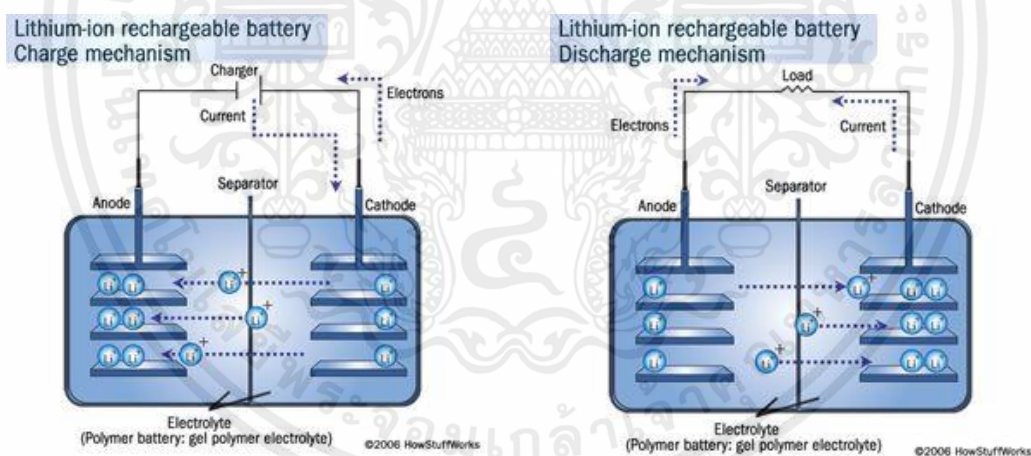


รูปที่ 2.1 โครงสร้างของแบตเตอรี่ลิเทียมไอออน [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 หลักการทำงานของแบตเตอรี่ลิเทียมไอออน

แบตเตอรี่ชนิดลิเทียมไอออนทำงานโดยอาศัยหลักการทางไฟฟ้าเคมี โดยขณะที่ประจุพลังงานไฟฟ้าที่เราประจุไฟเข้าไปทำให้เกิดปฏิกิริยาเคมีภายในแบตเตอรี่ ซึ่งจะบังคับให้ลิเทียมไอออนไหลออกจากโครงสร้างของวัสดุที่ใช้ทำขั้วแคโทด แล้วไหลผ่านอิเล็กโทรไลต์ แล้วผ่าน Separator และเข้าไปสอดตัว (Intercalate) อยู่ในโครงสร้างของวัสดุที่ใช้ทำขั้วแอโนด ดังแสดงในรูปที่ 2.2 ผลของปฏิกิริยานี้จะทำให้วัสดุที่ใช้ทำขั้วแคโทด เช่น LiMO_2 , LiM_2O_2 , LiMPO_4 และวัสดุที่ใช้ทำขั้วแอโนด เช่น C, Sn, Si อยู่ในสถานะไม่เสถียร ส่วนในขณะที่ใช้งาน (คายประจุ) ปฏิกิริยาเคมีในแบตเตอรี่จะสามารถเกิดขึ้นได้เอง (Spontaneous reaction) กล่าวคือลิเทียมไอออนที่ไหลออกจากโครงสร้างของวัสดุที่ใช้ทำขั้วแคโทด และไปแทรกตัวอยู่ที่ขั้วแอโนดนั้นจะไหลออกจากโครงสร้างของวัสดุที่ใช้ทำขั้วแอโนดและเข้าไปอยู่ในโครงสร้างของวัสดุที่ใช้ทำขั้วแคโทดตามเดิมทำให้ระบบมีสภาพเสถียรอีกครั้ง พร้อมกับให้อิเล็กตรอนผ่านวงจรไฟฟ้า (โดยที่อิเล็กตรอนจะไหลผ่านโลหะ Current collector) และให้พลังงานไฟฟ้าออกมา เมื่อใดก็ตามที่ลิเทียมไอออนไหลกลับไปเต็มหมดปฏิกิริยาก็จะสิ้นสุดหรือถ่านหมดนั่นเอง หากต้องการนำแบตเตอรี่ไปใช้ใหม่อีกต้องประจุไฟอีกครั้ง และเกิดเป็นเช่นนี้ต่อเนื่องไปจนกว่าแบตเตอรี่จะหมดสภาพและอายุการใช้งาน [9]



รูปที่ 2.2 หลักการทำงานของแบตเตอรี่ลิเทียมไอออน [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วิธีการประมาณสถานะประจุ

การประมาณสถานะประจุของการชาร์จแบตเตอรี่มีความสำคัญต่อการกำหนดช่วงที่เหลือของรถยนต์ที่ใช้ไฟฟ้าและระยะเวลาใช้งานของอุปกรณ์ที่ใช้พลังงานจากแบตเตอรี่ กล่าวคือความจุของแบตเตอรี่จะค่อย ๆ ลดลงตามจำนวนการใช้งานที่เพิ่มขึ้น ซึ่งมีหลายสาเหตุที่ทำให้ความจุของแบตเตอรี่ลดลง เช่น อุณหภูมิที่มากเกินไป การชาร์จมากเกินไป การคายประจุมากเกินไป และการไม่ได้ใช้งานเป็นเวลานานหลังจากชาร์จเต็มแล้ว โดยปริมาณไฟฟ้าที่เหลืออยู่ในแบตเตอรี่สามารถประมาณได้จากการหาสถานะประจุ (State-Of-Charge : SoC) ของแบตเตอรี่ โดยค่าสถานะประจุคือจำนวนประจุหรือพลังงานที่เหลืออยู่ภายในแบตเตอรี่ (Q) ซึ่งแสดงในรูปเปอร์เซ็นต์ต่อพิกัดความจุของแบตเตอรี่ (Rated Capacity : C_R) ดังสมการด้านล่าง [11,12]

$$SOC = \frac{Q}{C_R} \times 100\% \quad (2.1)$$

ซึ่งค่าของ SoC จะแตกต่างกันไประหว่าง 0% ถึง 100% หาก SoC เป็น 100% แสดงว่าเซลล์ถูกชาร์จจนเต็มในขณะที่ SoC 0% แสดงว่าเซลล์ถูกคายประจุจนหมด ในทำนองเดียวกันเมื่อเซลล์มีอายุมากขึ้น SoC สูงสุดจะเริ่มลดลง ซึ่งหมายความว่าสำหรับเซลล์ที่มีอายุมาก SoC 100% จะเทียบเท่ากับ SoC 75%-80% ของเซลล์ใหม่

2.3.1 วิธีการประมาณค่าสถานะประจุของแบตเตอรี่

2.3.1.1 วิธีการคายประจุไฟฟ้าสถิต (Static Discharge Method) เป็นการประมาณค่า SoC ของแบตเตอรี่ผ่านกระบวนการคายประจุแบตเตอรี่ โดยแบตเตอรี่จะถูกคายประจุอย่างช้า ๆ จนกว่าจะหมด และความจุเริ่มต้นของแบตเตอรี่สามารถหาได้จากการวัดและการรวมปริมาณการคายประจุ วิธีการนี้ต้องการการปล่อยประจุแบตเตอรี่อย่างสมบูรณ์ ซึ่งไม่สามารถตอบสนองความต้องการตามเวลาจริงได้ด้วยระยะเวลาการคายประจุที่นานขึ้นได้ นอกจากนี้แบตเตอรี่จะไม่สามารถใช้งานได้หลังจากการคายประจุจนหมด

2.3.1.2 วิธีการประมาณค่าสถานะประจุจากแรงดันขณะเปิดวงจร (Open-Circuit Voltage: OCV) โดยที่แรงดันขณะเปิดวงจรคือแรงดันขั้วของแบตเตอรี่ขณะที่แบตเตอรี่ไม่ได้จ่ายกระแสหรือขณะเปิดวงจร ซึ่งแรงดันขนาดเปิดวงจรจะมีความสัมพันธ์เป็นเชิงเส้นกับค่าสถานะประจุ ดังนั้นถ้าทราบค่าแรงดันขณะเปิดวงจรของแบตเตอรี่ก็จะสามารถประมาณค่าสถานะประจุได้ แต่ข้อเสียของวิธีนี้คือไม่สามารถนำมาใช้ได้ขณะที่แบตเตอรี่จ่ายกระแส

2.3.1.3 วิธีการนับคูลอมบ์ (Coulomb Counting) วิธีที่ใช้งานง่ายที่สุดคือวิธีนี้อาศัยการอ่านค่ากระแส จากนั้นคำนวณการวัดอัตราการไหลของกระแสไฟฟ้าทั้งอัดและคายประจุและทำการหาปริพันธ์เทียบกับเวลา ดังสมการ 2.2 จึงทำให้สามารถวัดค่า SoC ของแบตเตอรี่ได้อย่างแม่นยำ

ซึ่งแตกต่างจากวิธี OCV วิธีนี้สามารถตรวจสอบวิวัฒนาการของสถานะการชาร์จระหว่างการใช้แบตเตอรี่ไม่จำเป็นต้องพักแบตเตอรี่เพื่อทำการวัดที่แม่นยำ

$$\text{SOC}(t) = \text{SOC}(t_0) + \frac{\int_{t_0}^t I_{\text{batt}}(t) dt}{Q_{\text{Rated}}} \times 100\% \quad (2.2)$$

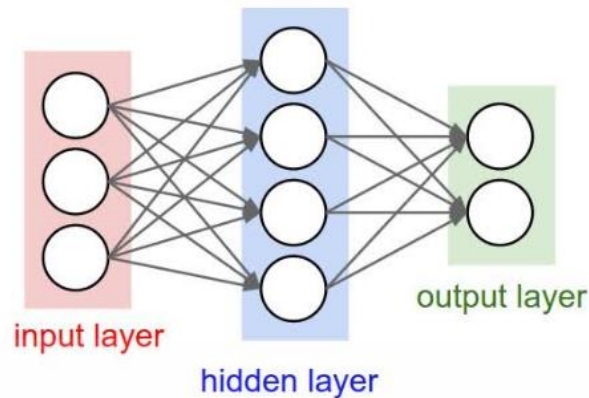
2.3.1.4 การวัดแอมแปร์-ชั่วโมง (Ampere-Hour Measurement) หรือวิธีการนับคูโลมบ์ (Coulomb counting method) ใช้สำหรับการประมาณค่า SoC ซึ่งกระแสไฟฟ้าที่คายประจุจะวัดจากแบตเตอรี่ และกระแสไฟฟ้าที่คายประจุจะถูกรวมเข้าด้วยกันเมื่อเวลาผ่านไป การดำเนินการนี้ไม่ซับซ้อนแต่จำเป็นต้องมีความรู้เบื้องต้นเกี่ยวกับ SoC เริ่มต้น และมีความยากลำบากจากการรวบรวมข้อผิดพลาดในการวัด

2.3.1.5 การประยุกต์ใช้ระเบียบวิธีของ Kalman Filter เป็นการประมาณ SoC ที่ตรวจสอบได้สำหรับแบตเตอรี่โดยการประมาณสถานะตามเวลาจริง มีความแม่นยำ แต่ข้อเสียคือต้องใช้ชุดสมการที่ซับซ้อนมากเพื่อกระตุ้นผลลัพธ์

2.3.1.6 วิธีโครงข่ายประสาทเทียม (Neural Network Algorithm) มีลักษณะเฉพาะแบบไม่เชิงเส้นเพื่อประเมิน SoC ของแบตเตอรี่ผ่านการฝึกอบรมข้อมูลการฝึกอบรมจำนวนมาก ความแม่นยำในการประมาณ SoC ขึ้นอยู่กับข้อมูลการฝึกอบรมและวิธีการฝึกอบรมเป็นส่วนใหญ่ แต่มีข้อเสียเนื่องจากกระบวนการนี้ต้องใช้การคำนวณสูงและซับซ้อนมาก [13,14]

2.4 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial neural network : ANN) คือโมเดลทางคณิตศาสตร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์ซึ่งเซลล์ประสาทเหล่านี้จะมีการเชื่อมโยงกับน้ำหนัก (weight) ที่เป็นสัญญาณที่ส่งผ่านจากเซลล์ประสาทหนึ่ง ไปยังอีกเซลล์ประสาทหนึ่ง ซึ่งเซลล์ประสาทแต่ละเซลล์จะรับตัวเลขหนึ่งตัวเพื่อส่งต่อไปยังปลายทาง แต่ผลลัพธ์ที่ได้มักจะมีมากกว่าหนึ่งจำนวน ซึ่งผลลัพธ์ที่ได้นี้จะส่งต่อไปยังเซลล์ประสาทอื่น ๆ ที่เชื่อมต่อกัน ซึ่งการส่งสัญญาณออกจะสิ้นสุดเมื่อมีการรับข้อมูลจากเซลล์ประสาทอื่น ๆ เข้ามาภายในเซลล์ รูปที่ 2.3 แสดงถึงการเชื่อมโยงเซลล์ประสาทต่าง ๆ [15]



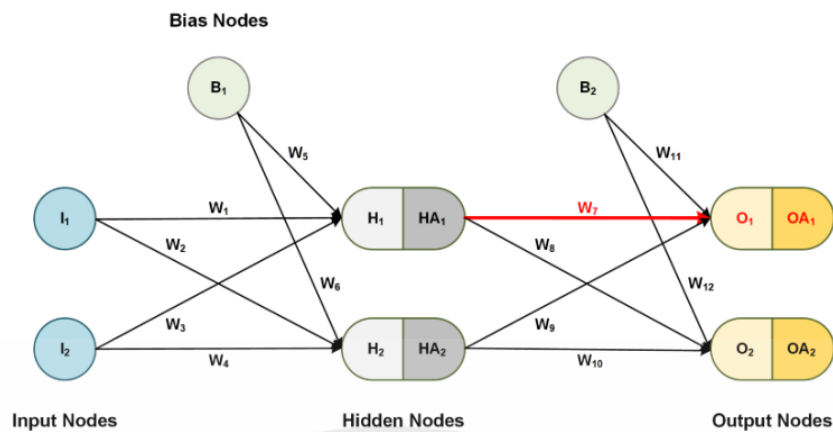
รูปที่ 2.3 โครงข่ายประสาทเทียม (Artificial neural network : ANN) [16]

2.4.1 โครงสร้างของโครงข่ายประสาทเทียม

2.4.1.1 ชั้นข้อมูลเข้า (Input Layer) ที่ชั้นนี้จะเป็นข้อมูลเข้า จำนวนของโหนดในชั้นนี้ขึ้นอยู่กับจำนวนของข้อมูลเข้า เช่น ถ้าข้อมูลของลูกค้าเป็นข้อมูลเข้าที่ประกอบด้วย อายุ เพศ จังหวัด ที่อาศัย รวมทั้งสิ้น 4 อย่าง ดังนั้นชั้นข้อมูลเข้าก็จะมี 4 โหนด ซึ่งอาจจะเรียกปัจจัยที่นำมาวิเคราะห์เหล่านี้ว่าคุณลักษณะ (Feature)

2.4.1.2 ชั้นซ่อน (Hidden Layer) เป็นชั้นที่อยู่ระหว่างกลาง มีหน้าที่รับข้อมูลจาก Layer ก่อนหน้า จะสังเกตว่า Hidden Layer สามารถมีจำนวนมากกว่า 1 ได้ และโดยพื้นฐาน ถ้าเราต้องการความแม่นยำที่มากขึ้นเราก็จะเพิ่มจำนวนชั้นของ Hidden Layer และจำนวน Neuron ให้มากขึ้น

2.4.1.3 ชั้นข้อมูลออก (Output Layer) เป็น Layer ที่อยู่ท้ายสุดรอรับค่าจาก Hidden Layer อันสุดท้าย โดยในชั้น Output นั้น แต่ละ Neuron จะมีค่าน้ำหนัก (Weight) ของคลาสอยู่ และใน Hidden Layer มีอีกสองสิ่งที่ถูกซ่อนไว้คือ Bias และ Weight แสดงดังรูปที่ 2.4 โดยทุก ๆ Hidden Layer จะมี Bias เชื่อมต่ออยู่เพื่อให้ทุก ๆ การคำนวณเพื่อส่งต่อมีความเท่าเทียมกันตีความหมายง่าย ๆ คือทุก ๆ Neuron ใน Hidden Layer จะต้องมี Bias เข้าไปคำนวณเพื่อให้ Decision Boundary ไม่จำเป็นต้องผ่านจุด Origin ส่วน Weight จะเป็นน้ำหนักซึ่งมันจะส่งผลทุก ๆ Neuron มีค่า Output ที่ไม่เท่ากัน ทำให้แต่ละคลาสมีน้ำหนักไม่เท่ากัน เวลาคำนวณว่าเป็นคลาสไหนทำให้เราสามารถแยกว่าข้อมูลนี้เป็นคลาสอะไรได้ด้วยการดูตัวเลขที่ Output [17]



รูปที่ 2.4 โครงข่ายประสาทเทียม (Artificial neural network : ANN) [18]

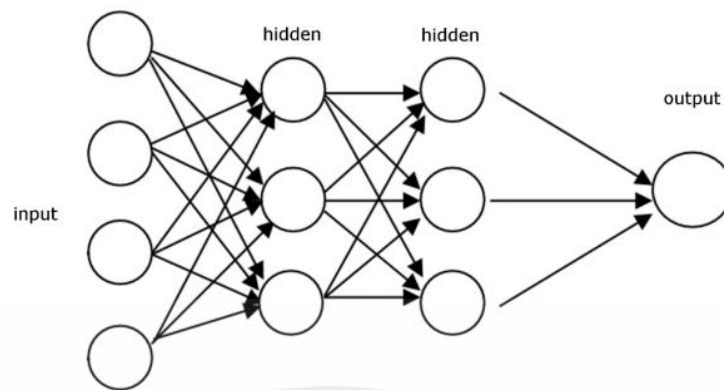
2.4.2 ประเภทของโครงสร้างของโครงข่ายประสาทเทียม

2.4.2.1 การคำนวณแบบไปข้างหน้า

โครงข่ายประสาทที่คำนวณแบบไปข้างหน้า (Feed-forward artificial neural network) ดังรูปที่ 2.5 สัญญาณจะถูกส่งไปทางเดียวจากอินพุตไปยังเอาต์พุตไม่มีการย้อนกลับ ผลของข้อมูลในชั้นต่าง ๆ จะไม่มีการคำนวณซ้ำ ค่าที่ได้จากการคำนวณของอินพุตจะแสดงในเอาต์พุตทันที ส่วนใหญ่จะใช้สำหรับปัญหาการแยกประเภทอย่างง่าย

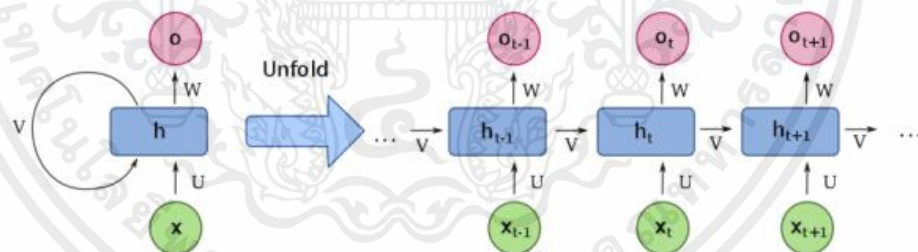
2.4.2.1.1 โครงข่ายประสาทแบบป้อนไปข้างหน้าชั้นเดียว (Single-layer feed forward neural networks) ประกอบด้วยชั้นสัญญาณประสาทขาเข้าและชั้นสัญญาณประสาทขาออกเท่านั้น

2.4.2.1.2 โครงข่ายประสาทแบบป้อนไปข้างหน้าหลายชั้น (Multi-layer feed forward neural networks) มีลักษณะเช่นเดียวกับโครงข่ายประสาทแบบป้อนไปข้างหน้าชั้นเดียว แต่จะมีชั้นซ่อนเพิ่มขึ้น โดยอยู่ระหว่างชั้นข้อมูลนำเข้าและชั้นส่งออก [19,20]



รูปที่ 2.5 โครงข่ายประสาทที่คำนวณแบบไปข้างหน้า
(Feed-forward artificial neural network) [21]

2.4.2.2 โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent neural network) เป็นโครงสร้างเครือข่ายการเรียนรู้เชิงลึกที่ใช้ผลลัพธ์ที่ได้จากการคำนวณ ย้อนกลับมาใช้เป็นข้อมูลเข้าอีกครั้งซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง [22]



รูปที่ 2.6 โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent neural network) [23]

สำหรับความแตกต่างของการคำนวณแบบไปข้างหน้า (Feed-forward Artificial Neural Network : FNN) กับ โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent Neural Network : RNN) คือ FNN เป็นฟังก์ชันที่รับข้อมูลเข้าและประมวลผลออกมาเป็นข้อมูลออก โดย FNN จะมองข้อมูลเข้าแต่ละตัวแยกกันชัดเจน ไม่ขึ้นต่อกัน ข้อมูลเข้าที่รับเข้าไปจะเข้ามีการเรียงลำดับอย่างไรก็ได้ FNN จะไม่สนใจและจะประมวลผลให้ข้อมูล ออกมาเหมือนเดิมซึ่ง FNN จะมีปัญหาเกี่ยวกับข้อมูลที่เป็นลำดับ เช่น ข้อความ (ลำดับตัวอักษร), เสียง (ลำดับแรงดันอากาศ), วิดีโอ (ลำดับของภาพและเสียง) และข้อมูลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแบบ Time Series ต่าง ๆ เป็นต้น แต่ RNN ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับงานที่ข้อมูลที่มีลำดับ โดยใช้หลักการนำสถานะภายในของโมเดลกลับมาเป็นข้อมูลเข้าใหม่คู่กับข้อมูลเข้าแบบปกติ เรียกว่า สถานะซ่อน (Hidden State) หรือสถานะภายใน (Internal State) ช่วยให้โมเดลรู้จักรูปแบบ (Pattern) ของลำดับข้อมูลเข้า (Input Sequence) ได้แสดงดังรูปที่ 2.6

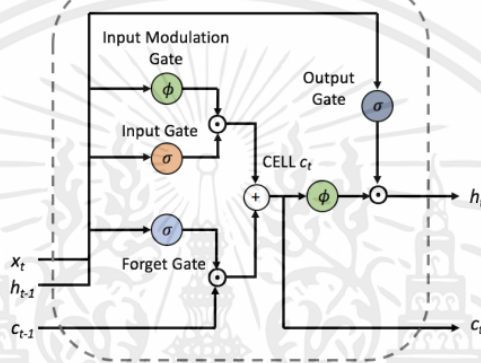
ในแต่ละโหนดของ RNN จะมีข้อมูลเข้าสองอย่าง ได้แก่ 1) ข้อมูลเข้า ณ โหนด นั้น ๆ และ 2) ผลลัพธ์ที่ได้จากการคำนวณในโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทางคือ 1) ผลลัพธ์ที่ออก ณ โหนดนั้น ๆ และออกเพื่อไปเข้าเป็นข้อมูลเข้าในโหนดถัดไป ข้อดีของ RNN คือ มันมีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคต ซึ่งหมายถึง อะไรที่เคยเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย แม้ RNN จะมีข้อดีในการทำงานของข้อมูลที่มีความต่อเนื่อง แต่ข้อเสียของ RNN คือมันสามารถย้อนกลับได้แค่เพียงในช่วงระยะเวลาสั้น ๆ เท่านั้น ซึ่งปัญหาหลัก ๆ ของ RNN เกิดมาจากค่าเกรเดียนที่เริ่มน้อยลง ในข้อมูลที่มีความยาวมากขึ้น จนแทบจะไม่สามารถเห็นความเปลี่ยนแปลงของเกรเดียนได้เลย จึงมีการเสนอใช้วิธีที่มีประสิทธิภาพทางการคำนวณมากกว่าสามารถจัดการข้อมูลในหน่วยความจำได้เป็นเวลานานกว่า และช่วยให้ผู้ใช้สามารถฝึกโมเดลโดยใช้ลำดับที่มีขั้นตอนเวลาหลายร้อยขั้นตอนได้ นั่นก็คือ Long Short-Term Memory (LSTM) ซึ่งเป็น Recurrent Neural Network (RNN) ชนิดหนึ่งใน RNN เอาต์พุตจากขั้นตอนสุดท้ายจะถูกป้อนเป็นอินพุตในขั้นตอนปัจจุบัน ใจสำคัญของ RNN ก็เพื่อจะใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) เช่น Video (Sequence of images) หรือ Text (Sequence of words) ยกตัวอย่างเปรียบเทียบกับการอ่านหนังสือซึ่งก็คือลำดับของคำที่ต่อกัน (Sequence of words) อ่านทีละคำจากซ้ายไปขวา (สำหรับภาษาไทยหรือภาษาอังกฤษ) การที่จะสามารถรู้เรื่องได้ว่าประโยคที่กำลังอ่านนั้นเกี่ยวกับอะไร ก็ต้องเอาเรื่องราวจากสิ่งที่อ่านผ่านไปแล้ว (Hidden state) มาผสมกับคำที่เพิ่งอ่านไป (Input data) ทำให้สามารถเข้าใจความหมายในส่วนที่กำลังอ่านได้ ซึ่ง RNN ก็ใช้หลักการเดียวกันคือการปรับรูปแบบของ Neural network เดิม เพื่อให้สามารถเอา State ก่อนหน้า มาบวกกับ Input data ตัวใหม่ที่เข้ามา เพื่อทำความเข้าใจอะไรสักอย่างต่อไปเรื่อย ๆ การใช้การเกิดซ้ำสามารถปรับปรุงความสามารถของเครือข่ายในการเข้ารหัสข้อมูลเชิงบริบทให้กับแบบจำลอง ทำให้เหมาะสมในการแก้ปัญหาที่ข้อมูลในอดีตเกี่ยวข้องกับการประเมินหรือคาดการณ์ผลลัพธ์ เช่น การพยากรณ์ตลาดหุ้น SoC ของแบตเตอรี่ และสภาพการขับขี่ [24,25]

หลักการทำงานของ LSTM จะมี Cell state เป็นตัวเก็บ State ของ Memory cell ใน LSTM และ Gate เป็นตัวที่ควบคุมการไหลของข้อมูล ซึ่งก็คือ ค่า Analog ที่คอยควบคุมว่าเมื่อไหร่ควรจะ Read, Write หรือ Forget ซึ่งมันก็เหมือนกับประตูที่จะดูว่าเมื่อไหร่ควรเปิดให้ข้อมูลไหลเข้า-ออก หรือไหลหายไปเลย

1. Forget. การล้าง Cell state เดิมออกไป เพื่อเคลียร์พื้นที่รับข้อมูลใหม่ โดยมี Forget gate เป็นเหมือนตัวตัดสินใจ ถ้า Forget gate ให้ค่าเป็น 0 ก็จะสั่งให้ลบ Cell state เดิมออกไป แต่ถ้า Forget gate ให้ค่าเป็น 1 ก็จะเก็บ Cell state นี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Write: เมื่อมี Input data ใหม่เข้ามา ต้องคิดว่าจะอัปเดต Cell state ด้วย Input data ใหม่หรือไม่ ซึ่งจะควบคุมโดย Input gate จะยังคงใช้ Sigmoid function เป็นตัวตัดสินใจว่าจะอนุญาตให้อัปเดตหรือไม่ โดยใช้ค่า Input data ที่เข้ามา กับ Hidden state ก่อนหน้า มาคำนวณ และต้องคิดว่าถ้าจะอัปเดตต้องอัปเดตด้วยค่าอะไร ซึ่งรอบนี้จะใช้ Input modulation gate เป็นตัวจัดการ
 3. Updating cell state: ใช้ข้อมูลที่ได้จาก Forget gate, Input gate และ Input modulation gate เพื่อนำมาอัปเดต Cell state
 4. Read: การที่จะอนุญาตให้สามารถอ่านข้อมูลได้หรือไม่ได้ ซึ่งจะมี Output gate มาช่วยตัดสินใจ
- [26]



รูปที่ 2.7 โครงสร้าง Long Short-Term Memory [27]

2.5 The Worldwide harmonized Light vehicles Test Procedure (WLTP)

เป็นระบบทดสอบรถยนต์ระดับโลกที่คาดการณ์การประหยัดเชื้อเพลิงการปล่อยมลพิษ และช่วงของรถยนต์ไฟฟ้าอัตราประหยัดของรถยนต์เครื่องยนต์ปกติและแบบ Hybrid โดยจะเอารูปแบบการทดสอบจากสถานการณ์จริงบนท้องถนนเข้ามาเป็นส่วนประกอบในการทดสอบด้วย เพื่อให้ได้ค่า

การทดสอบของทาง WLTP จะมีการทดสอบในห้องทดลองที่มีการควบคุมสภาพแวดล้อมของรถที่ใช้ทดสอบให้อยู่ในสภาวะเดียวกันทั้งอุณหภูมิห้อง ความหนาแน่นของอากาศ เชื้อเพลิงคุณภาพองน้ำมันเชื้อเพลิง ความเร็วลม และเครื่องวัดความเร็ว เป็นต้น เพื่อให้การทดสอบเป็นธรรมกับรถทุกคัน และในการทดสอบนั้นจะถูกแบ่งประเภทรถมาเป็น 3 แบบคือ

Class 1 คือรถที่มีกำลังเครื่องยนต์ต่ำกว่า 22 วัตต์/น้ำหนักรถ 1 กิโลกรัม

Class 2 คือรถที่มีกำลังเครื่องยนต์ระหว่าง 22-34 วัตต์/น้ำหนักรถ 1 กิโลกรัม

Class 3 คือรถที่มีกำลังเครื่องยนต์มากกว่า 34 วัตต์/น้ำหนักรถ 1 กิโลกรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 WLTP Range

ช่วง WLTP คำนวณโดยเรียกใช้รอบการทดสอบสองครั้งและวัดการใช้พลังงานจากสิ่งที่มีอยู่ในแบตเตอรี่ ดังสมการด้านล่าง [28,29,30,31]

$$\text{ช่วง (กม.)} = \frac{\text{พลังงานแบตเตอรี่ที่ใช้งานได้ (Wh)}}{\text{การใช้พลังงานจากแบตเตอรี่ (Wh/km)}} \quad (2.3)$$

2.6 โปรแกรม MATLAB



รูปที่ 2.8 โลโก้ของโปรแกรม MATLAB [32]

MATLAB เป็นโปรแกรมสำหรับการคำนวณเชิงตัวเลขและ Visualization ที่มีประสิทธิภาพสูง สามารถทำได้ง่าย เมื่อเทียบกับภาษาโปรแกรมอื่น ๆ เราสามารถสร้างและกำหนดค่าตัวแปรได้ทันทีโดยไม่ต้องประกาศตัวแปรก่อน ซึ่งสะดวกในการเขียนโปรแกรม เพราะผู้ใช้ไม่ต้องกังวลในเรื่องโครงสร้างของภาษา เช่น ในเรื่องการต้องประกาศตัวแปรก่อนใช้งาน แต่มาสนใจอัลกอริทึมในการแก้โจทย์ปัญหาเป็นหลัก ข้อมูลไม่ว่าจะเป็นตัวเลข หรือตัวอักษร (Stings) จะถูกจัดเก็บในรูปแบบของแถวและหลัก หรือ Array ซึ่งก็คือ Matrix นั่นเอง เช่น จำนวนสเกลลาร์ (Scalar) จะถูกแทนด้วยเมตริกซ์ขนาด 1x1 ข้อมูลที่เป็นเวกเตอร์จะถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 แถว ในกรณีที่เป็นเวกเตอร์แบบแถว (Row vector) หรือ ถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 หลัก ในกรณีที่เป็นเวกเตอร์แบบหลัก (Column vector) เป็นต้น การที่ MATLAB ถูกออกแบบมาให้มีการทำงานภายในเช่นนี้ ทำให้การเขียนโปรแกรมแก้โจทย์ปัญหาที่มีลักษณะของ Vector และ Matrix เป็นเรื่องง่าย ตัวอย่างเช่น การแก้ระบบสมการเชิงเส้น ซึ่งระบบสมการ สรุปความสามารถของโปรแกรม MATLAB เบื้องต้น ดังนี้

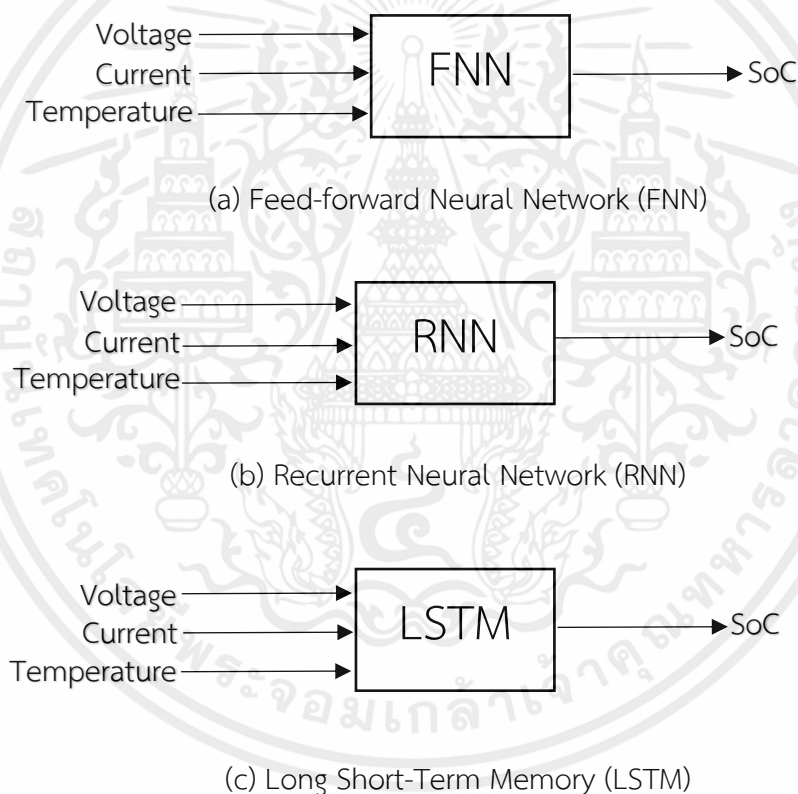
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MATLAB เป็นโปรแกรมเพื่อการคำนวณและแสดงผลได้ทั้งตัวเลข และรูปภาพซึ่งมีประสิทธิภาพสูง สามารถทำการเขียนกราฟทั้ง 2 มิติ และ 3 มิติ ได้อย่างง่ายดาย และมีประสิทธิภาพ
- สามารถควบคุมการทำงานของ MATLAB ด้วยชุดคำสั่ง (Command line) และยังสามารถรวบรวมชุดคำสั่งเป็นโปรแกรม (Script file) ได้ด้วย
- ลักษณะการเขียนโปรแกรมใน MATLAB จะใกล้เคียงการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคย จึงง่ายกว่าการเขียนโปรแกรมด้วยภาษาชั้นสูง เช่น ภาษา C, Pascal, Fortran และอื่น ๆ
- MATLAB มีฟังก์ชันสำเร็จรูป (Built-in function) เพื่อทำงานเฉพาะทางมากมาย นอกจากนี้ผู้ใช้ยังสามารถเขียนฟังก์ชันขึ้นมาใหม่โดยใช้ประโยชน์จากฟังก์ชันที่มีอยู่เดิมได้เพื่อให้เหมาะสมกับงานของผู้ใช้แต่ละกลุ่มสำหรับผู้ใช้ที่ต้องการใช้งานเฉพาะทางชั้นสูง เช่น งานด้าน Control, Image Processing, Artificial Neural Network หรืออื่น ๆ MATLAB ก็มี Toolbox หรือชุด Function พิเศษ เพื่อทำงานเฉพาะทางนั้น ๆ [33]

บทที่ 3

การออกแบบและสร้างโปรแกรมในการประมาณสถานะประจุของแบตเตอรี่

การออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเธียมไอออนโดยใช้โครงข่ายประสาทเทียม โดยการใช้โปรแกรม MATLAB จะกำหนดสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้ว กระแสไฟฟ้า และอุณหภูมิ และกำหนดสัญญาณขาออกคือสถานะประจุ แสดงดังรูปที่ 3.1 ลักษณะการประมาณสถานะประจุของแบตเตอรี่โดยใช้โครงข่ายประสาทเทียมชนิด FNN, RNN และ LSTM และมีการกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับ



รูปที่ 3.1 ลักษณะการประมาณสถานะประจุของแบตเตอรี่โดยใช้โครงข่ายประสาทเทียมชนิด

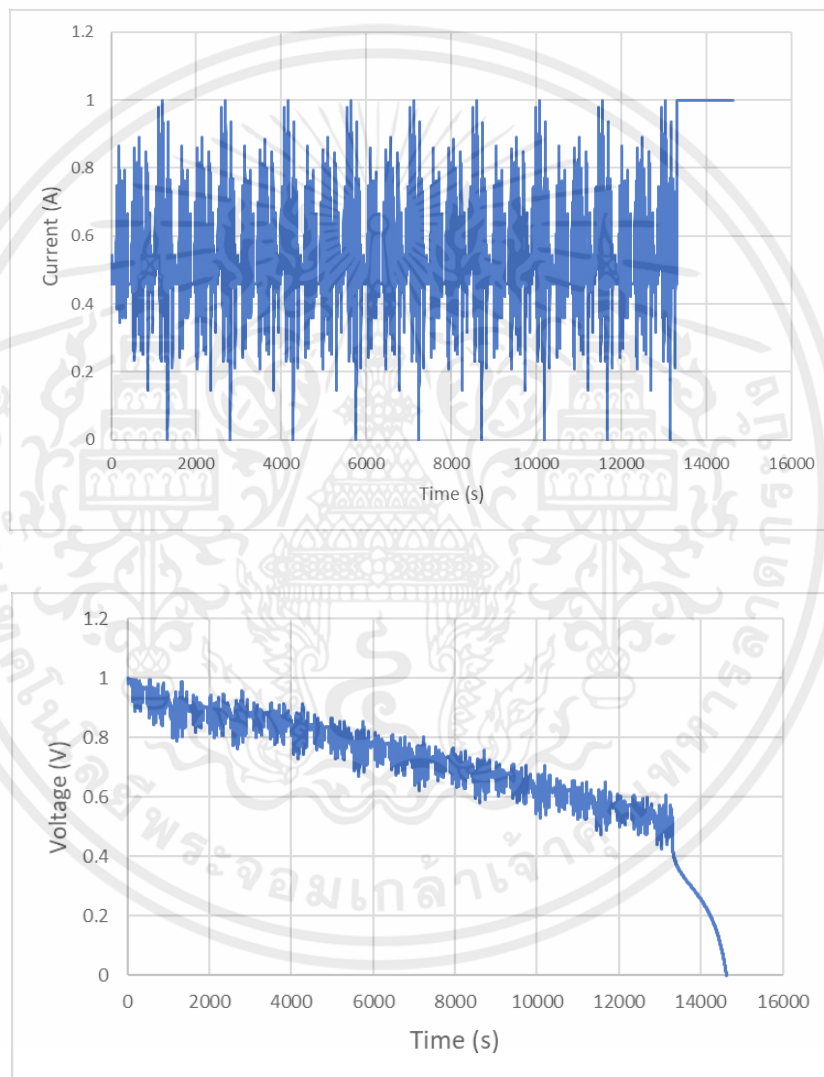
(a) Feed-forward Neural Network (FNN), (b) Recurrent Neural Network (RNN)

(c) Long Short-Term Memory (LSTM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

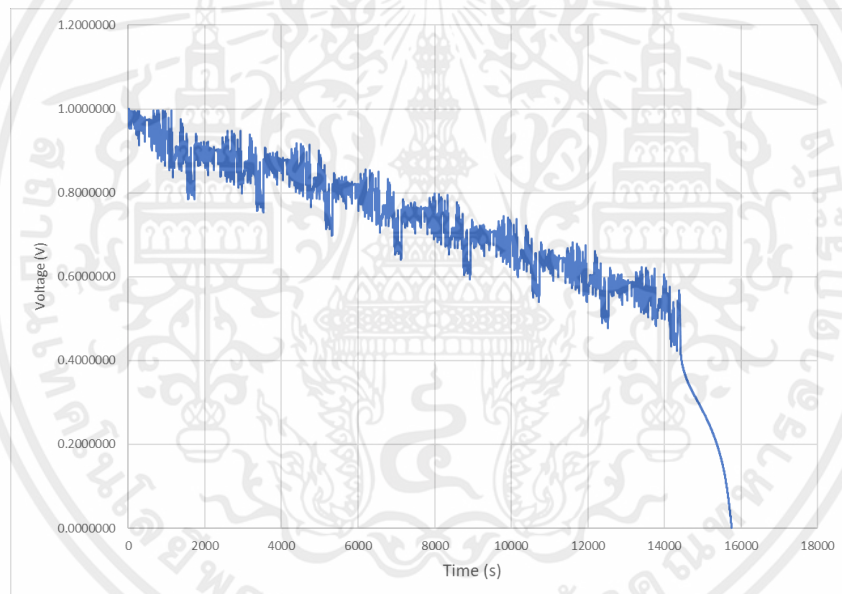
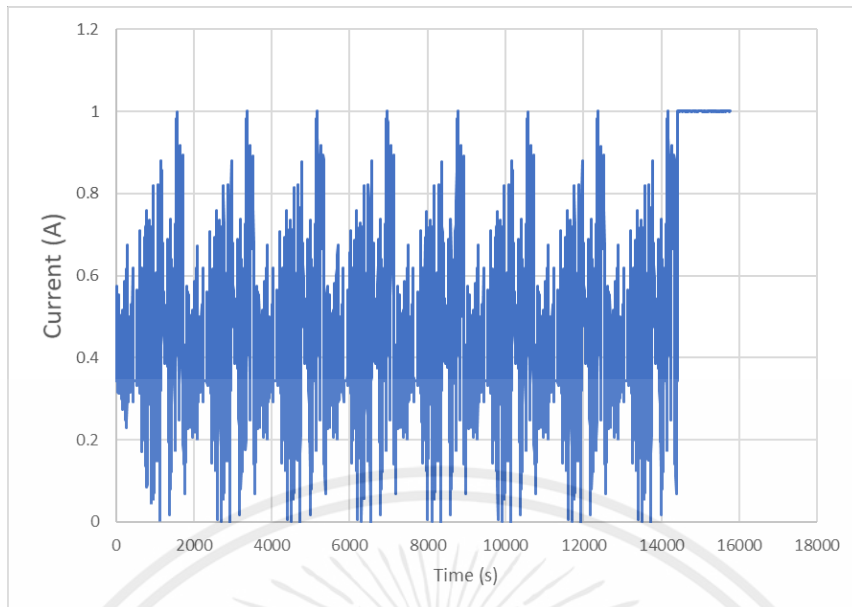
3.1 Input Profile

โพรไฟล์ที่ใช้เป็นสัญญาณเข้าสำหรับการประมาณสถานะประจุคือโพรไฟล์ที่จำลองการวิ่งของรถยนต์ไฟฟ้าในห้องทดสอบรถยนต์ไฟฟ้าโดยใช้ EV Reference Applications Model และเลือกใช้ Driving Cycle Source เป็น WLTP Class 2 และ WLTP Class 3 [25] ซึ่งสัญญาณเข้าที่ใช้ในการประมาณสถานะประจุโดย FNN, RNN และ LSTM คือแรงดันไฟฟ้าระหว่างขั้วและกระแสไฟฟ้าแสดงดังรูปที่ 3.2 (a) และ (b) กราฟแรงดันไฟฟ้าระหว่างขั้วและกระแสไฟฟ้าของโพรไฟล์ WLTP Class 2 และ WLTP Class 3



(a) กราฟกระแสไฟฟ้าและแรงดันไฟฟ้าระหว่างขั้วของโพรไฟล์ WLTP Class 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b) กราฟกระแสไฟฟ้าและแรงดันไฟฟ้าระหว่างขั้วของโปรไฟล์ WLTP Class 3

รูปที่ 3.2 กราฟกระแสไฟฟ้าและแรงดันไฟฟ้าระหว่างขั้วของโปรไฟล์อินพุต

(a) โปรไฟล์ WLTP Class2, (b) โปรไฟล์ WLTP Class3

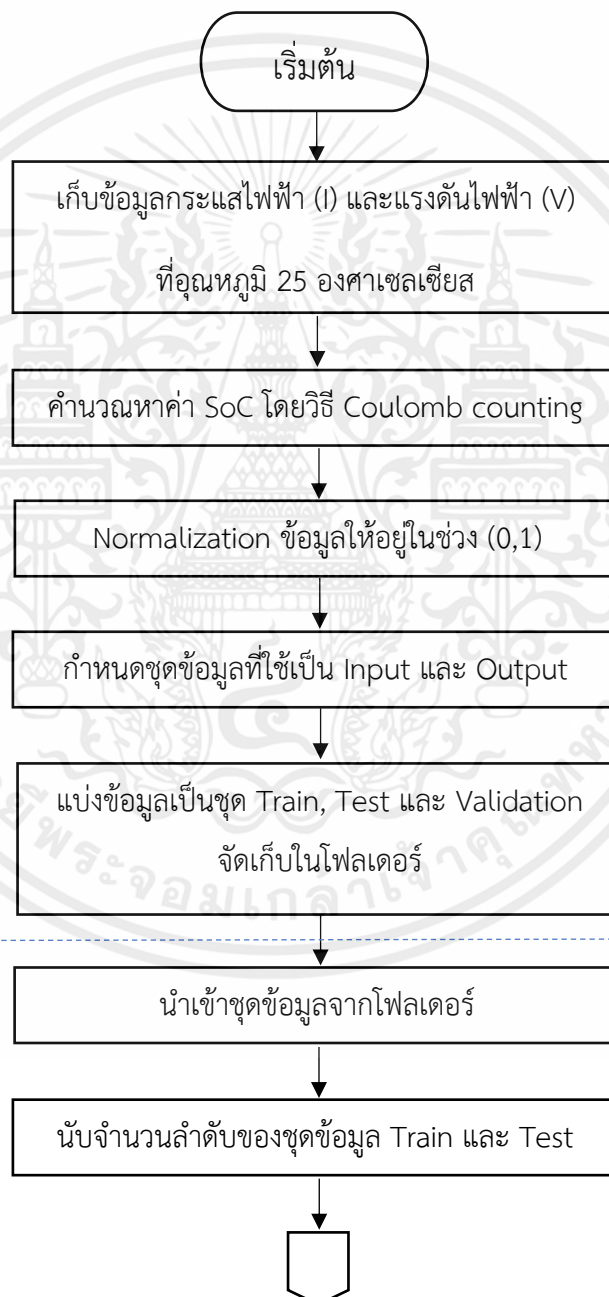
WLTP Class 3 ถูกนำมาใช้เป็น Training set และ WLTP Class 2 ถูกนำมาใช้เป็น Testing set สำหรับการประเมินสถานะประจุโดยโครงข่าย FNN RNN และ LSTM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แผนผัง และ Code ในการประมาณสถานะประจุของแบตเตอรี่

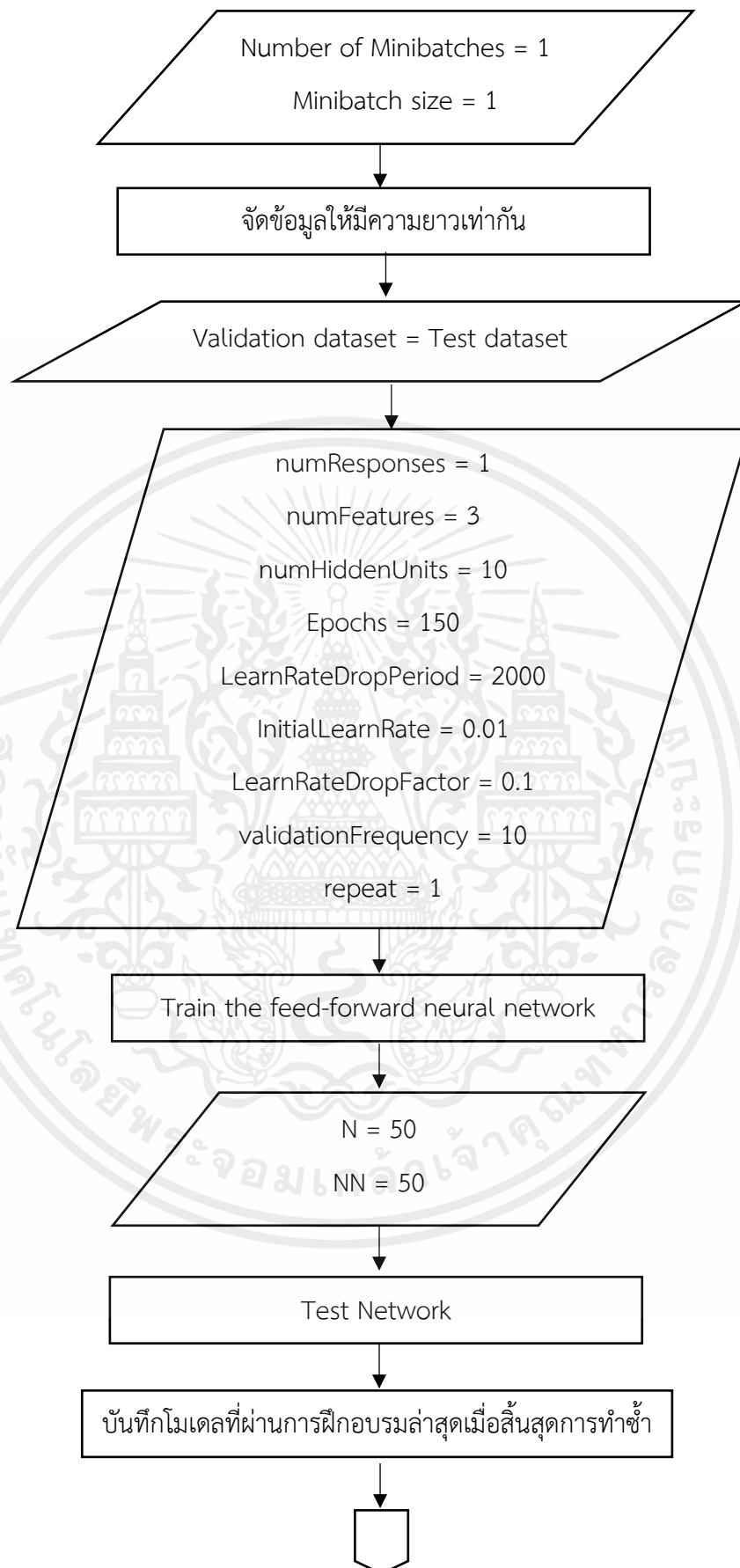
โดยมีค่า Parameter ที่ใช้ในการออกแบบดังต่อไปนี้ จำนวนเซลล์ประสาทที่เลือกใช้ใน Hidden layer คือ 10 เซลล์, ค่า Learning rate เท่ากับ 0.01, จำนวนของ Learn Rate Drop Period เท่ากับ 2000 และกำหนด Epoch เท่ากับ 150 จากนั้นทำการ Train และ Test โคจรข่าย โดยใช้ชุดข้อมูลเป็น WLTP Class 2 และ WLTP Class 3

3.2.1 แผนผัง Feed-forward Neural Network

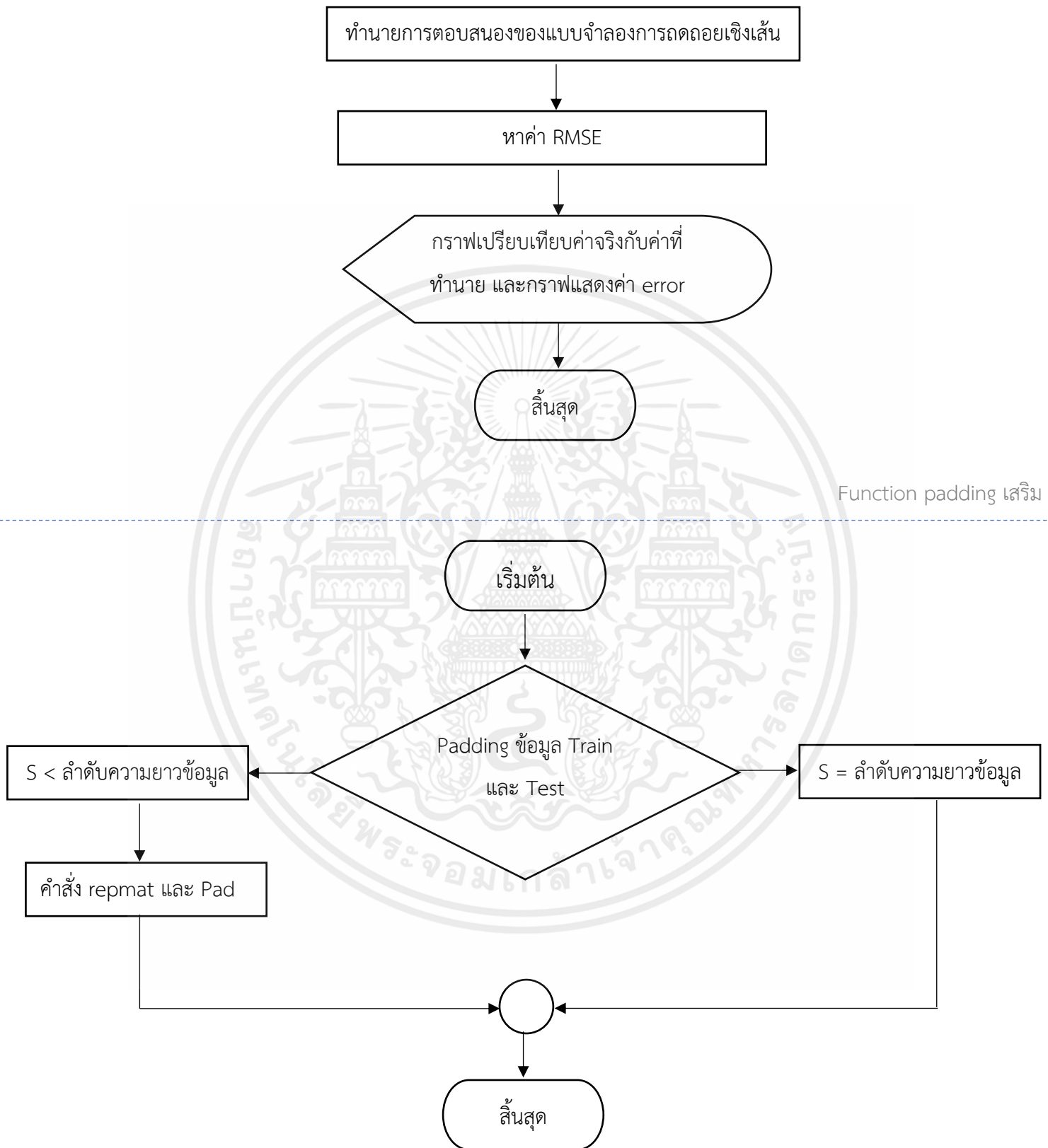


ส่วน Program MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Code Feed-forward Neural Network

```

clc;
close all;
clearvars;
fclose all;
%***load file folder data && define directory for save result
load ="fnn"; % Add the Dataset File name
outputFolder = fullfile('C:\Users\User\Desktop\Project\code ทั้งหมด\fnn');%
The temporary location and name where the training data unzipped
mkdir("FNN");
NET_Path = "FNN";
%select the Datafolder created in the next popup window to allow storing the
%the data each iteration
NET_Path=uigetdir('','Select folder to store data from each training
iteration');
%***set Train datastore
folderTrain = fullfile(outputFolder,"Train");
fdsPredictorTrain = fileDatastore(folderTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); %Is a datastore, for the training dataset,
which is a repository for collections of data that are too large to fit in
memory.
%preview(fdsPredictorTrain)
folderYTrain = fullfile(outputFolder,"Train");
fdsObjectiveTrain = fileDatastore(folderYTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the training dataset.
%preview(fdsObjectiveTrain);
%set Test datastore

%***set Test datastore
folderTest = fullfile(outputFolder,"Test");
fdsPredictorTest = fileDatastore(folderTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); % Is a datastore, for the Testing dataset,
which is a repository for collections of data that are too large to fit in
memory
%preview(fdsPredictorTest)
folderYTest = fullfile(outputFolder,"Test");
fdsObjectiveTest = fileDatastore(folderYTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the testing dataset.
% preview(fdsObjectiveTest);
%*** Transform and Combine Data set
RR=readall(fdsPredictorTrain); % read all training data files
numObservations = numel(RR);
for i=1:numObservations
    sequence = RR{i}; %combine training data set files into one structure.
    sequenceLengths(i) = size(sequence.X,2);
end

%***Minibatch size
miniBatchSize = 1; % Size of Mini-batches

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

sequenceLength = max(sequenceLengths);%Max length of the training dataset
files if the files are not in the same length, this will determine the
amount of padding on the shorter files. If all files have the same length,
there will be no padding .
RRR=readall(fdsPredictorTest); % read all Test data files
numObservationsY = numel(RRR);
for i=1:numObservationsY
    sequence = RRR{i}; %Combine Test data set files into one structure.
    sequenceLengthsY(i) = size(sequence.X,2);
end
sequenceLengthY= max(sequenceLengthsY); %Max length of the objective
function of the training dataset (SOC or Ah)
RRTest=readall(fdsPredictorTest);
numObservationsTest = numel(RRTest);

%*** Transforming Datastore dataset
tdsObjectiveTrain = transform(fdsObjectiveTrain,@(data)
padSequenceY(data,sequenceLength)); % Padding function - Objective function
Training Dataset
tdsObjectiveTest = transform(fdsObjectiveTest,@(data)
padSequenceY(data,sequenceLengthY)); % Padding function - Objective
function Testing Dataset
ObjectiveTest=readall(tdsObjectiveTest); %Objective function Test dataset,
this is only to help manipulate the dataset for plotting and calculate the
Errors.
ObjectiveTrain=readall(tdsObjectiveTrain);
tdsTrain = transform(fdsPredictorTrain,@(data)
padSequence(data,sequenceLength)); % Padding function - Training Dataset
tdsTest = transform(fdsPredictorTest,@(data)
padSequence(data,sequenceLengthY)); % Padding function - Testing Dataset
% preview(tdsTrain);
PredictorTest=readall(tdsTest);
PredictorTrain=readall(tdsTrain);
cdsTrain = combine(tdsTrain,tdsObjectiveTrain);
cdsTest = combine(tdsTest,tdsObjectiveTest);
% preview(cdsTrain)

%***Validation dataset
V=1; % Selection of the Validation dataset, in this script the validation
dataset is the same as Test dataset, but it can be changed
XTest=PredictorTest{V}; % predictors Validation dataset
YTest=ObjectiveTest{V}; % Objective function "Y == SOC" for the Validation
dataset
XTrain=PredictorTrain{V};
YTrain=ObjectiveTrain{V};

%***Specify NN training parameter
numResponses = 1; %Number of outputs (SOC)
numFeatures = 3; %Number of inputs features (V, I, T)
numHiddenUnits = 10; %Number of hidden units 'N', where each hidden unit
for FNN represents a Neuron.
Epochs = 150; %Number of Epochs
LearnRateDropPeriod = 1000; %Number of epochs for dropping the learning
rate
InitialLearnRate=0.01; %Initial Learning rate. Very important parameter
LearnRateDropFactor=0.1; %Factor for dropping the learning rate, ex.:
reduce to 10% of the current learning rate at every 'LearnRateDropPeriod
validationFrequency = 10; %Validate the model at each Iterations when
values is equal to 1.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

repeat=1; %Select the amount the training process will be repeated

%***FNN Structure
layers = [sequenceInputLayer(numFeatures)
         fullyConnectedLayer(numHiddenUnits)
         tanhLayer % Hyperbolic Tangent
         fullyConnectedLayer(numHiddenUnits)
         leakyReluLayer(0.1) % The leaky rectified linear unit
         (ReLU) activation operation performs a nonlinear threshold operation, where
         any input value less than zero is multiplied by a fixed scale factor
         fullyConnectedLayer(numResponses)
         clippedReluLayer(1) % A clipped ReLU layer performs a
         threshold operation, where any input value less than zero is set to zero
         and any value above the clipping ceiling, this case '1', is set to that
         clipping ceiling.
         regressionLayer];
options = trainingOptions('adam', ... % Adam optimizer
                          'MaxEpochs', Epochs, 'ExecutionEnvironment', 'auto', ... % Select GPU if
                          you works on computer with GPU
                          'GradientThreshold', 1, ...
                          'InitialLearnRate', InitialLearnRate, ...
                          'LearnRateSchedule', 'piecewise', ...
                          'LearnRateDropPeriod', LearnRateDropPeriod, ...
                          'LearnRateDropFactor', LearnRateDropFactor, ...
                          'L2Regularization', 1, ...
                          'ValidationData', {XTest, YTest}, ...
                          'Shuffle', 'never', ...
                          "Verbose", 1, ...
                          'ValidationFrequency', validationFrequency, ...
                          'MiniBatchSize', miniBatchSize, ...
                          'Plots', 'training-progress', ...
                          'CheckpointPath', NET_Path); % to save each trained model during the
training process, if necessary it can be "commented-out" and will only the
last model will be available to be save.

%***Train the FNN network with the specified training options.
for i=1:repeat
    net = trainNetwork(cdsTrain, layers, options);
    N=50; % To remove the first 'N' and last 'NN' datapoints to compensate
the lack of previous data, which wouldn't normally occur in a real life xEV
application
    NN=50;
%*Test Network
    YT=readall(fdsObjectiveTest);
    XT=readall(fdsPredictorTest);
    X_T=[]; Y_Test=[]; X_Test=[]; Y_T=[];
    Error_Test=[];
    Nets{i} = net; % save the last trained model at the end of each repeat
    Y_Pred_25degC = predict(net, XT{1, 1}.X, 'MiniBatchSize', miniBatchSize);
    Error_Test_25degC = Y.Pred_25degC(:, N:end-NN) - XT{1, 1}.Y(:, N:end-NN);

    E_Test_RMSE_25degC{i} = sqrt(mean((Error_Test_25degC).^2))*100; %SOC
Test dataset root mean square error
    E_Test_MAE_25degC{i} = mae(Error_Test_25degC)*100; %SOC Test dataset
Mean abs error
    E_Test_MAX_25degC{i} = max(abs(Error_Test_25degC)*100); %max SOC error
of Test dataset

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

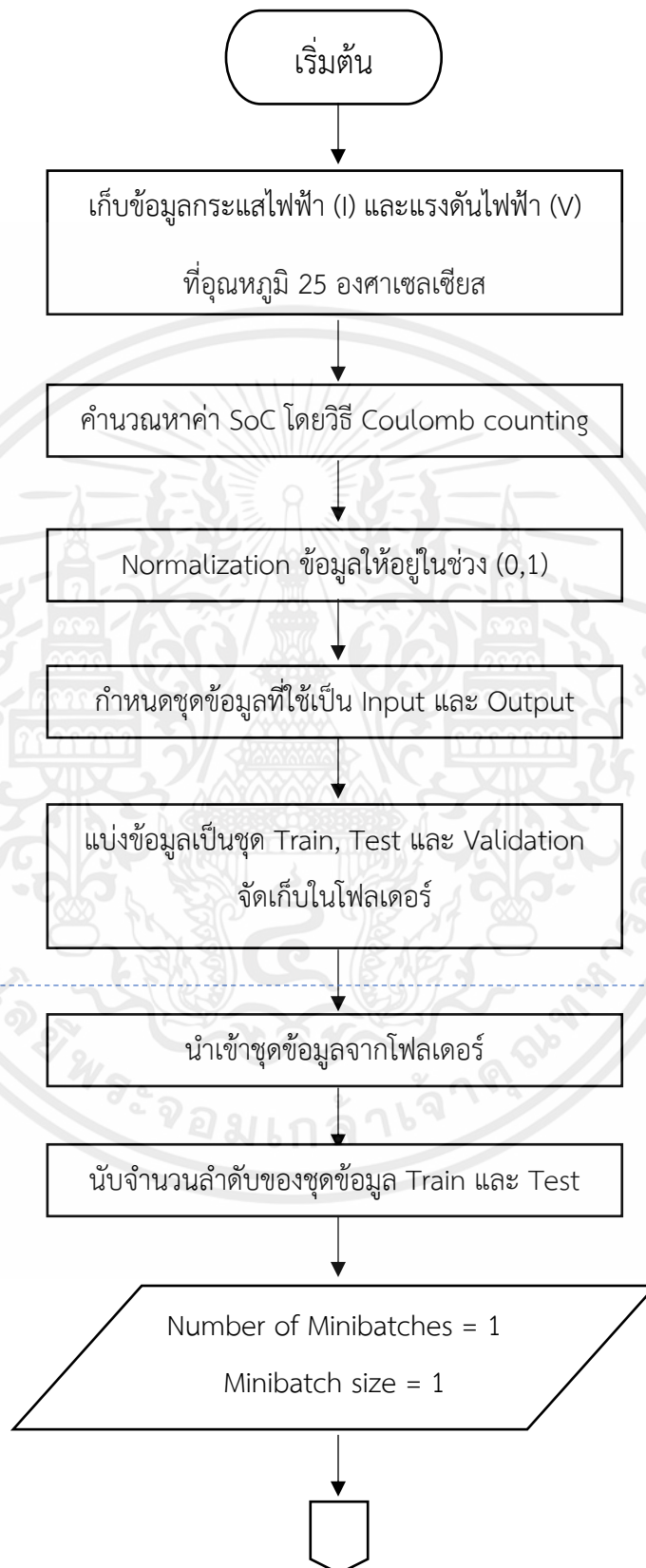
RMSE_SUMMARY = ['25degC_RMSE',E_Test_RMSE_25degC];
MAE_SUMMARY = ['25degC_MAE',E_Test_MAE_25degC];
MAX_SUMMARY = ['25degC_MAX',E_Test_MAX_25degC];
***Based on the best result from the above figure select the best #repeat
to plot the on time domain
[Minimum_25degC, idx] = min(cell2mat(E_Test_RMSE_25degC));
SelectNet =idx;
    %SelectNet = 1; % Manual selection
Y_Pred_25degC = predict(Nets{SelectNet},XT{1,
1}.X,'MiniBatchSize',miniBatchSize);
Error_Test_25degC = Y_Pred_25degC(:,N:end-NN)-XT{1, 1}.Y(:,N:end-NN);
figure
subplot(2,1,1)
grid on
hold all;
plot(XT{1, 1}.Y(:,N:end-NN)*100,'LineWidth',3,'LineStyle','-');
plot(Y_Pred_25degC*100,'LineWidth',2,'LineStyle','-');
legend(["Observed" "Predicted"])
ylabel("(SOC)")
xlabel('Time(s)')
title("Test - SOC@25degC - Estimation")
grid on
subplot(2,1,2)
grid on
hold all;
plot((Error_Test_25degC)*100,'LineWidth',2,'LineStyle','-');
legend("Error")
ylabel("(%)")
xlabel('Time(s)')
title("Error")
***Function
function sequence = padSequence(data,sequenceLength)
sequence = data.X;
[C,S] = size(sequence);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequence,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequence = [sequence padding];
else
    sequence = sequence(:,1:sequenceLength);
end
sequence = {sequence};
end
function sequenceY = padSequenceY(data,sequenceLength)
sequenceY = data.Y;
[C,S] = size(sequenceY);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequenceY,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequenceY = [sequenceY padding];
else
    sequenceY = sequenceY(:,1:sequenceLength);
end
sequenceY = {sequenceY};

end

```

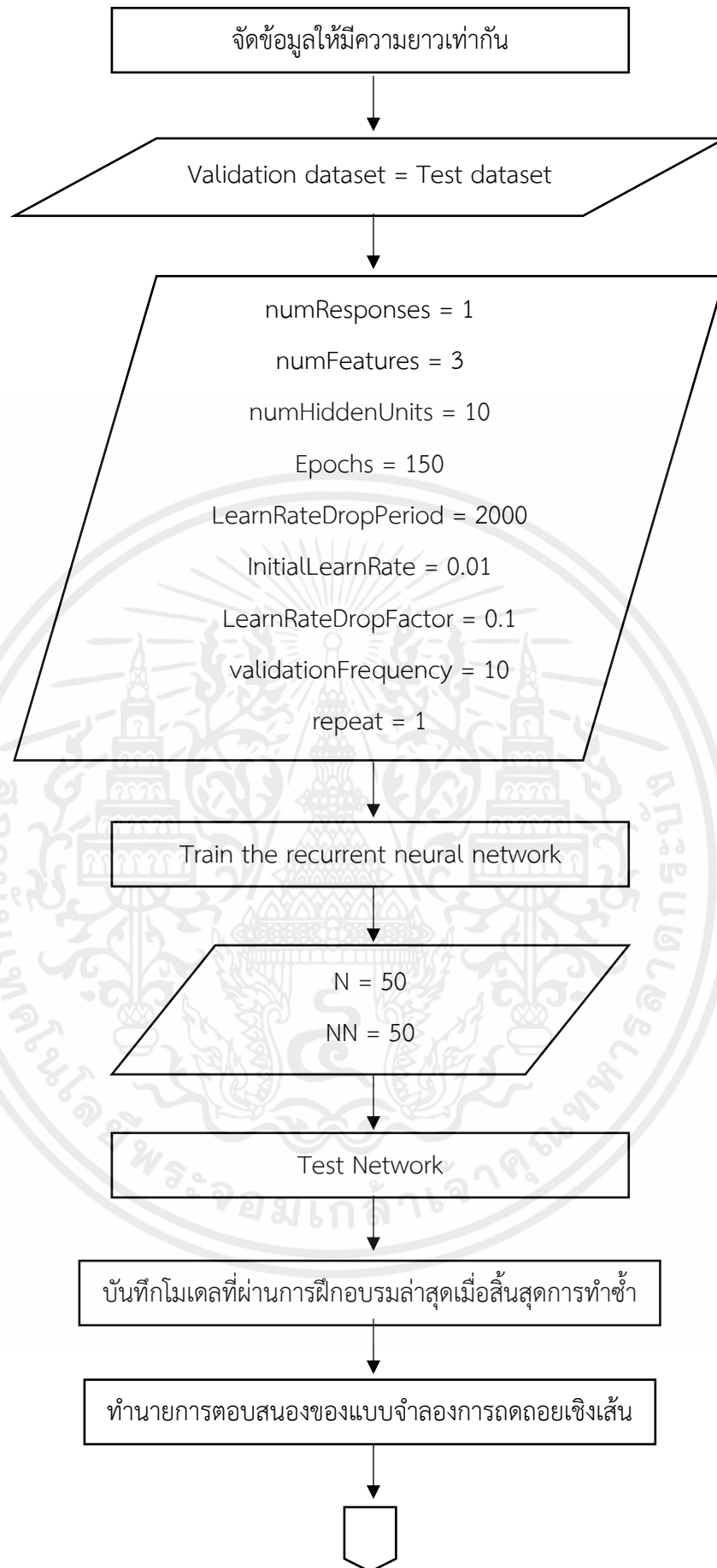
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แผนผัง Recurrent Neural Network

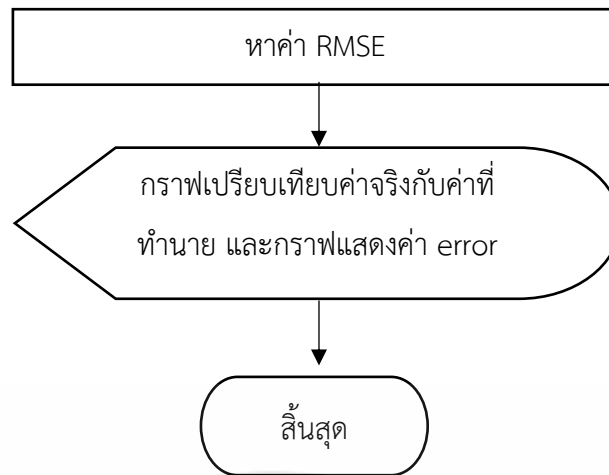


ส่วน Program MATLAB

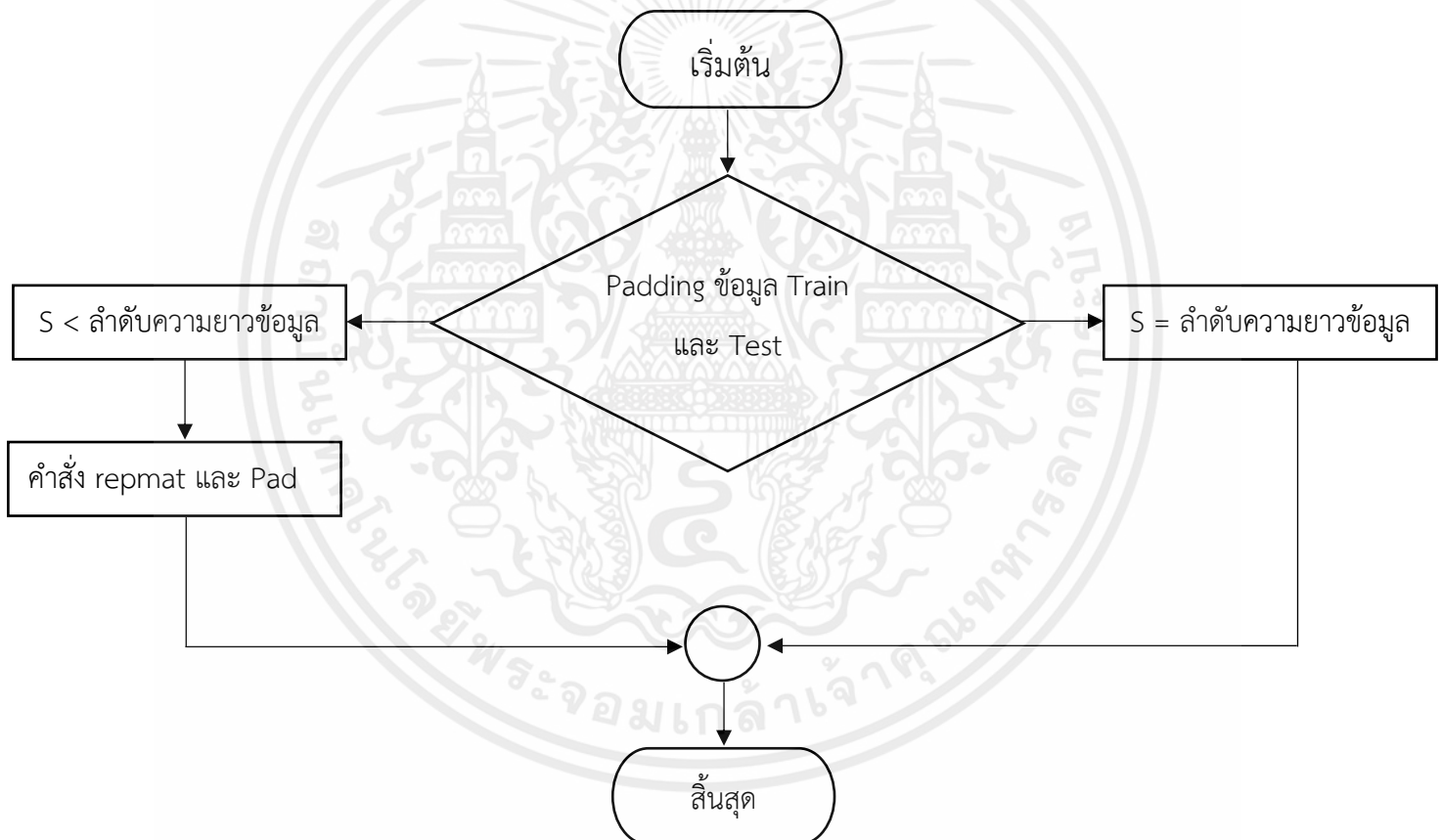
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Function padding เสริม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Code Recurrent Neural Network

```

clc;
close all;
clearvars;
fclose all;
%***load file folder data && define directory for save result
load ="rnn"; % Add the Dataset File name
outputFolder = fullfile('C:\Users\User\Desktop\Project\mix\rnn');% The
temporary location and name where the training data unzipped
mkdir("RNN");
NET_Path = "RNN";
%select the Datafolder created in the next popup window to allow storing the
%the data each iteration
NET_Path=uigetdir('','Select folder to store data from each training
iteration');
%***set Train datastore
folderTrain = fullfile(outputFolder,"Train");
fdsPredictorTrain = fileDatastore(folderTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); %Is a datastore, for the training dataset,
which is a repository for collections of data that are too large to fit in
memory.
%preview(fdsPredictorTrain)
folderYTrain = fullfile(outputFolder,"Train");
fdsObjectiveTrain = fileDatastore(folderYTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the training dataset.
%preview(fdsObjectiveTrain);
%set Test datastore
%***set Test datastore
folderTest = fullfile(outputFolder,"Test");
fdsPredictorTest = fileDatastore(folderTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); % Is a datastore, for the Testing dataset,
which is a repository for collections of data that are too large to fit in
memory
%preview(fdsPredictorTest)
folderYTest = fullfile(outputFolder,"Test");
fdsObjectiveTest = fileDatastore(folderYTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the testing dataset.
% preview(fdsObjectiveTest);

%*** Transform and Combine Data set
RR=readall(fdsPredictorTrain); % read all training data files
numObservations = numel(RR);
for i=1:numObservations
    sequence = RR{i}; %combine training data set files into one structure.
    sequenceLengths(i) = size(sequence,X,2);
end

%***Minibatch size
NumberOfMinibatches = 1; % If the dataset doesn't fit the memory it's
necessary to increase the number of MiniBatches. Need to be a integer
number 1 or >1.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

miniBatchSize = 1; % Size of Mini-batches
sequenceLength = max(sequenceLengths);%Max length of the training dataset
files if the files are not in the same length, this will determine the
amount of padding on the shorter files. If all files have the same length,
there will be no padding .
RRR=readall(fdsPredictorTest); % read all Test data files
numObservationsY = numel(RRR);
for i=1:numObservationsY
    sequence = RRR{i}; %Combine Test data set files into one structure.
    sequenceLengthsY(i) = size(sequence.X,2);
end
sequenceLengthY= max(sequenceLengthsY); %Max length of the objective
function of the training dataset (SOC or Ah)
RRTest=readall(fdsPredictorTest);
numObservationsTest = numel(RRTest);
*** Transforming Dastore dataset
tdsObjectiveTrain = transform(fdsObjectiveTrain,@(data)
padSequenceY(data,sequenceLength)); % Padding function - Objective function
Training Dataset
tdsObjectiveTest = transform(fdsObjectiveTest,@(data)
padSequenceY(data,sequenceLengthY)); % Padding function - Objective
function Testing Dataset
ObjectiveTest=readall(tdsObjectiveTest); %Objective function Test dataset,
this is only to help manipulate the dataset for plotting and calculate the
Errors.
ObjectiveTrain=readall(tdsObjectiveTrain);
tdsTrain = transform(fdsPredictorTrain,@(data)
padSequence(data,sequenceLength)); % Padding function - Training Dataset
tdsTest = transform(fdsPredictorTest,@(data)
padSequence(data,sequenceLengthY)); % Padding function - Testing Dataset
% preview(tdsTrain);
PredictorTest=readall(tdsTest);
PredictorTrain=readall(tdsTrain);
cdsTrain = combine(tdsTrain,tdsObjectiveTrain);
cdsTest = combine(tdsTest,tdsObjectiveTest);
% preview(cdsTrain)

***Validation dataset
V=1; % Selection of the Validation dataset, in this script the validation
dataset is the same as Test dataset, but it can be changed
XTest=PredictorTest{V}; % predictors Validation dataset
YTest=ObjectiveTest{V}; % Objective function "Y == SOC" for the Validation
dataset
XTrain=PredictorTrain{V};
YTrain=ObjectiveTrain{V};

***Specify NN training parameter
numResponses = 1; %Number of outputs (SOC)
numFeatures = 3; %Number of inputs features (V, I, T)
numHiddenUnits = 10; %Number of hidden units 'N', where each hidden unit
for LSTM represents a Neuron.
Epochs = 150; %Number of Epochs
LearnRateDropPeriod = 2000; %Number of epochs for dropping the learning
rate
InitialLearnRate=0.01; %Initial Learning rate. Very important parameter
LearnRateDropFactor=0.1; %Factor for dropping the learning rate, ex.:
reduce to 10% of the current learning rate at every 'LearnRateDropPeriod
validationFrequency =2; %Validate the model at each Iterations when values
is equal to 1.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

repeat=1; %Select the amount the training process will be repeated

%***RNN Structure
layers = [sequenceInputLayer(numFeatures)
          gruLayer(numHiddenUnits,'OutputMode','sequence') % number of hidden
          units in this example is '10'
          fullyConnectedLayer(numResponses)
          clippedReluLayer(1)
          regressionLayer];

options = trainingOptions('adam', ... % Adam optimizer
                          'MaxEpochs',Epochs,'ExecutionEnvironment','auto', ...% Select GPU if
you works on computer with GPU
                          'GradientThreshold',1, ...
                          'InitialLearnRate',InitialLearnRate, ...
                          'LearnRateSchedule','piecewise', ...
                          'LearnRateDropPeriod',LearnRateDropPeriod, ...
                          'LearnRateDropFactor',LearnRateDropFactor, ...
                          'L2Regularization',1, ...
                          'ValidationData', {XTest,YTest}, ...
                          'Shuffle','never',...
                          "Verbose",1,...
                          'ValidationFrequency',validationFrequency, ...
                          'MiniBatchSize',miniBatchSize, ...
                          'Plots','training-progress',...
                          'CheckpointPath', NET_Path); % to save each trained model during the
training process, if necessary it can be "commented-out" and will only the
last model will be available to be save.

%***Train the LSTM network with the specified training options.
for i=1:repeat
    net = trainNetwork(cdsTrain,layers,options);
    N=50; % To remove the first 'N' and last 'NN' datapoints to compensate
the lack of previous data, which wouldn't normally occur in a real life xEV
application
    NN=50;
%*Test Network
    YT=readall(fdsObjectiveTest);
    XT=readall(fdsPredictorTest);
    X_T=[];Y_Test=[];X_Test=[];Y_T=[];
    Error_Test=[];
    Nets{i} = net;% save the last trained model at the end of each repeat
    Y_Pred_25degC = predict(net,XT{1, 1}.X,'MiniBatchSize',miniBatchSize);

    Error_Test_25degC = Y_Pred_25degC(:,N:end-NN)-XT{1, 1}.Y(:,N:end-NN);

    E_Test_RMSE_25degC{i} = sqrt(mean((Error_Test_25degC).^2))*100; %SOC
Test dataset root mean square error
    E_Test_MAE_25degC{i} = mae(Error_Test_25degC)*100; %SOC Test dataset
Mean abs error
    E_Test_MAX_25degC{i} = max(abs(Error_Test_25degC)*100); %max SOC error
of Test dataset

end
RMSE_SUMMARY = ['25degC_RMSE',E_Test_RMSE_25degC;];
MAE_SUMMARY = ['25degC_MAE',E_Test_MAE_25degC;];
MAX_SUMMARY = ['25degC_MAX',E_Test_MAX_25degC;];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

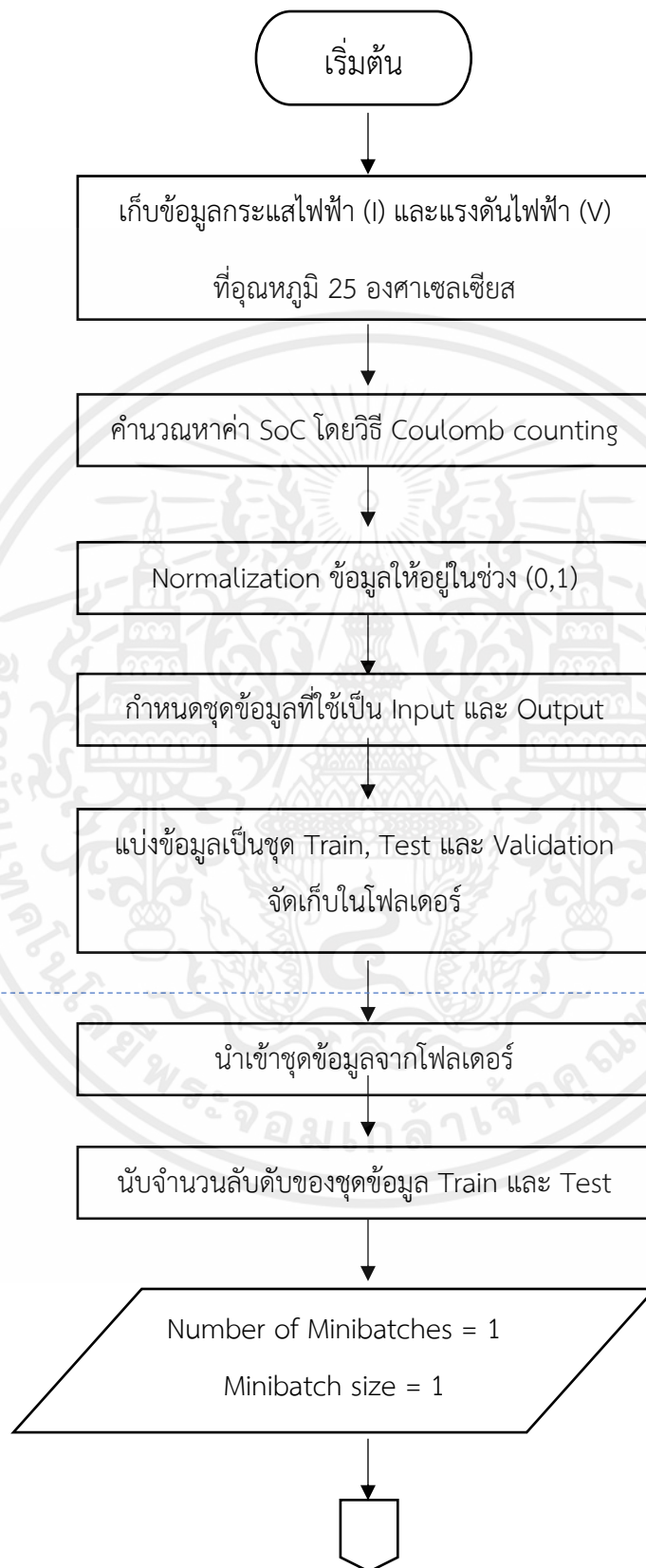
%***Based on the best result from the above figure select the best #repeat
to plot the on time domain
[Minimum_25degC, idx] = min(cell2mat(E_Test_RMSE_25degC));
SelectNet =idx;
    %SelectNet = 1; % Manual selection
Y_Pred_25degC = predict(Nets{SelectNet},XT{1,
1}.X, 'MiniBatchSize',miniBatchSize);

Error_Test_25degC = Y_Pred_25degC(:,N:end-NN)-XT{1, 1}.Y(:,N:end-NN);
figure
subplot(2,1,1)
grid on
hold all;
plot(XT{1, 1}.Y(:,N:end-NN)*100,'LineWidth',3,'LineStyle','-');
plot(Y_Pred_25degC*100,'LineWidth',2,'LineStyle','-');
legend(["Observed" "Predicted"])
ylabel("(SOC)")
xlabel('Time(s)')
title("Test - SOC@25degC - Estimation")
grid on
subplot(2,1,2)
grid on
hold all;
plot((Error_Test_25degC)*100,'LineWidth',2,'LineStyle','-');
legend("Error")
ylabel("(%")
xlabel('Time(s)')
title("Error")
%***Function
function sequence = padSequence(data,sequenceLength)
sequence = data.X;
[C,S] = size(sequence);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequence,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequence = [sequence padding];
else
    sequence = sequence(:,1:sequenceLength);
end
sequence = {sequence};
end
function sequenceY = padSequenceY(data,sequenceLength)
sequenceY = data.Y;
[C,S] = size(sequenceY);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequenceY,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequenceY = [sequenceY padding];
else
    sequenceY = sequenceY(:,1:sequenceLength);
end
sequenceY = {sequenceY};
end

```

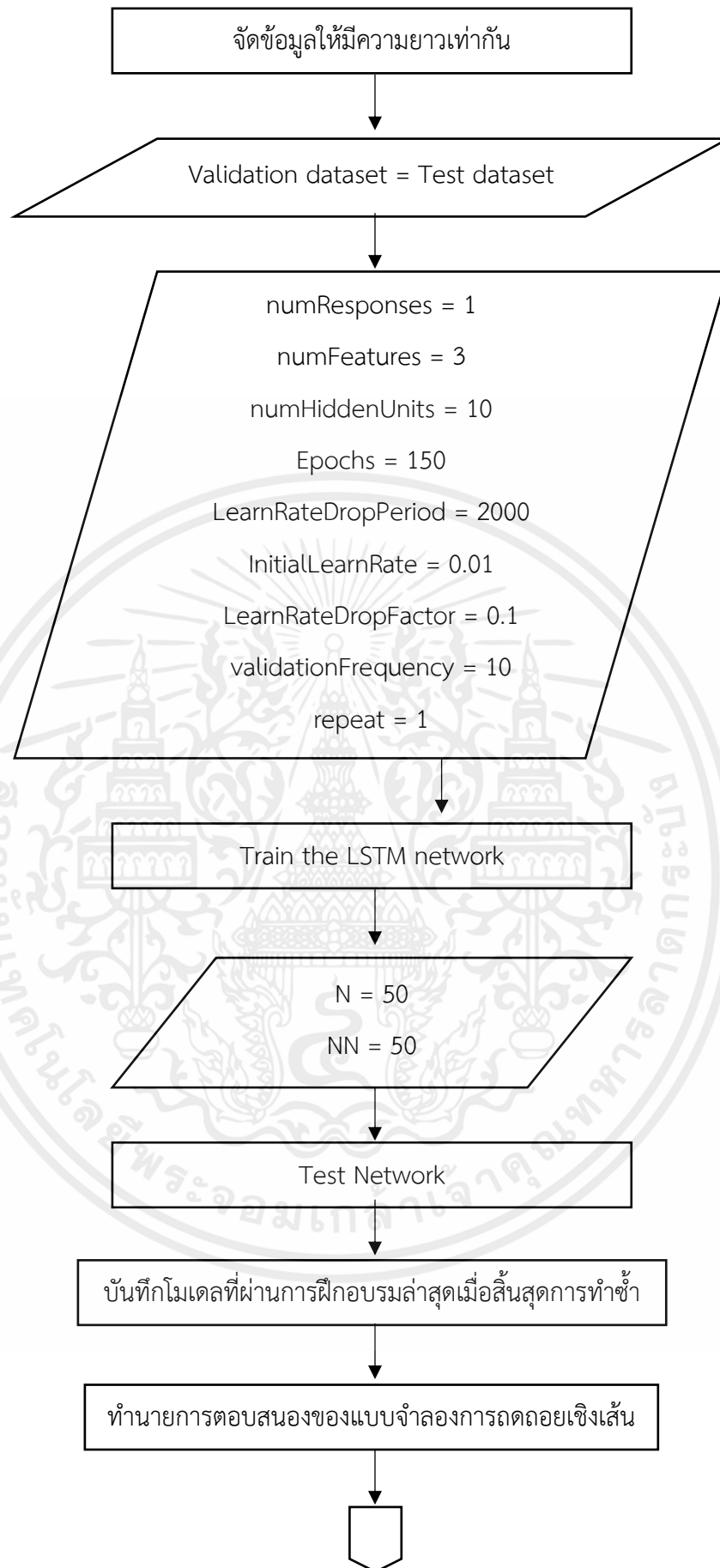
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 แผนผัง Long Short-Term Memory

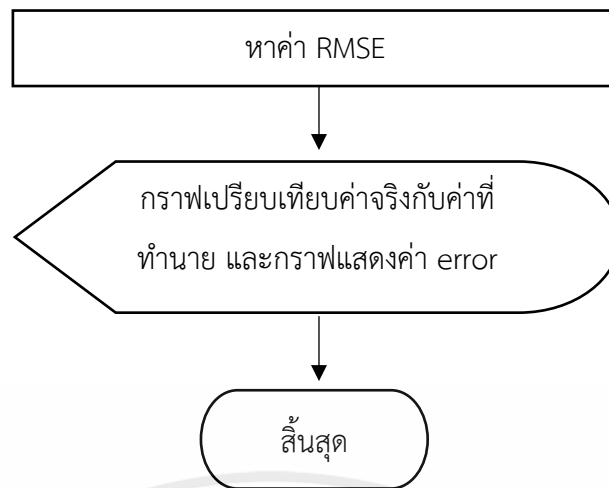


ส่วน Program MATLAB

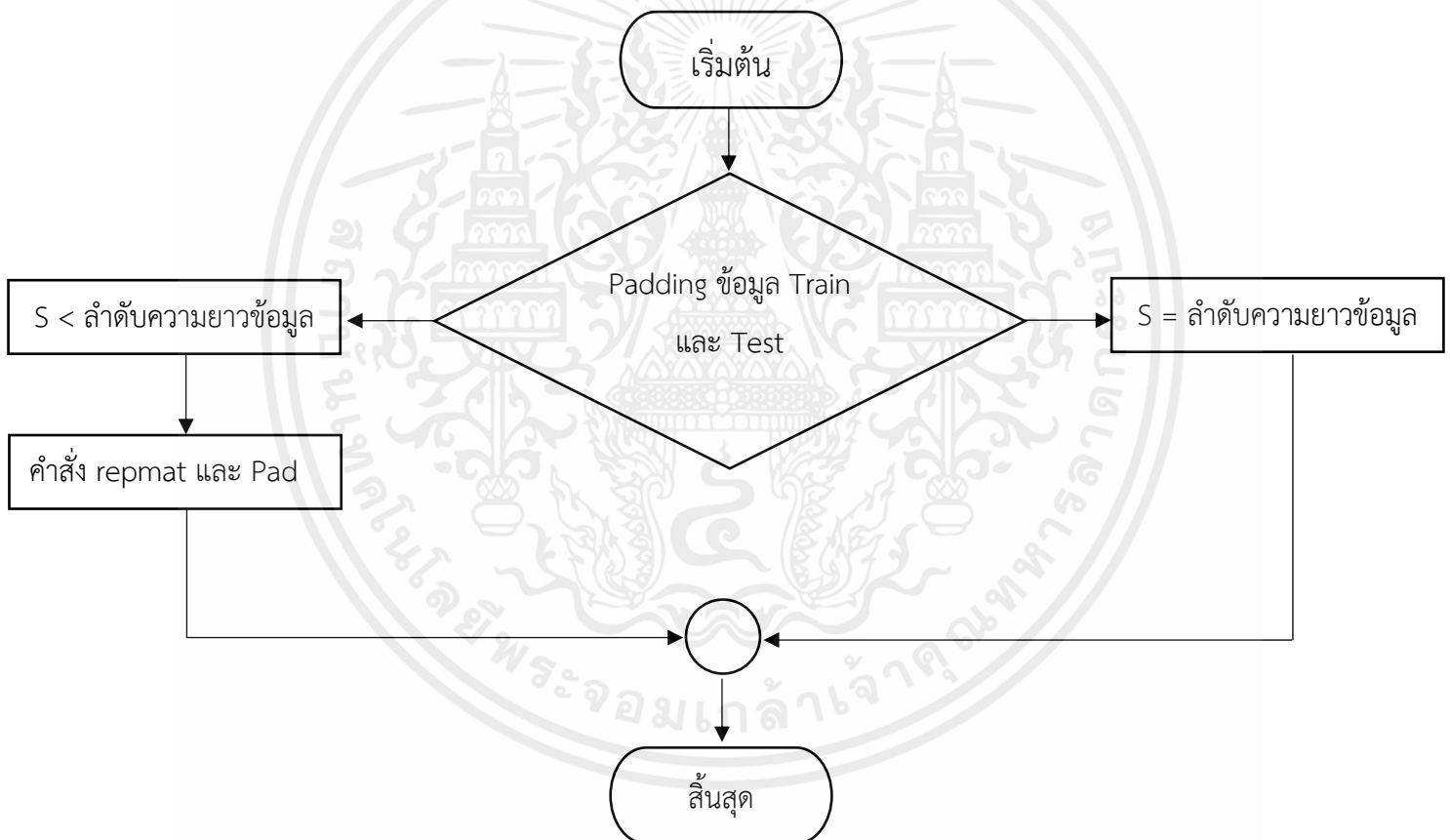
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Function padding เสริม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 Code Long Short-Term Memory

```

clc;
close all;
clearvars;
fclose all;
%***load file folder data && define directory for save result
load ="lstm"; % Add the Dataset File name
outputFolder = fullfile('C:\Users\User\Desktop\Project\code ทั้งหมด\lstm');%
The temporary location and name where the training data unzipped
mkdir("LSTM");
NET_Path = "LSTM";
%select the Datafolder created in the next popup window to allow storing the
%the data each iteration
NET_Path=uigetdir('','Select folder to store data from each training
iteration');
%***set Train datastore
folderTrain = fullfile(outputFolder,"Train");
fdsPredictorTrain = fileDatastore(folderTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); %Is a datastore, for the training dataset,
which is a repository for collections of data that are too large to fit in
memory.
%preview(fdsPredictorTrain)
folderYTrain = fullfile(outputFolder,"Train");
fdsObjectiveTrain = fileDatastore(folderYTrain, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the training dataset.
%preview(fdsObjectiveTrain);
%set Test datastore
%***set Test datastore
folderTest = fullfile(outputFolder,"Test");
fdsPredictorTest = fileDatastore(folderTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true); % Is a datastore, for the Testing dataset,
which is a repository for collections of data that are too large to fit in
memory
%preview(fdsPredictorTest)
folderYTest = fullfile(outputFolder,"Test");
fdsObjectiveTest = fileDatastore(folderYTest, ...
    'ReadFcn',@load, ...
    'IncludeSubfolders',true);% Is a datastore for the objective function,
in this case SOC reference data for the testing dataset.
% preview(fdsObjectiveTest);

%*** Transform and Combine Data set
RR=readall(fdsPredictorTrain); % read all training data files
numObservations = numel(RR);
for i=1:numObservations
    sequence = RR{i}; %combine training data set files into one structure.
    sequenceLengths(i) = size(sequence.X,2);
end

%***Minibatch size
NumberOfMinibatches = 1; % If the dataset doesn't fit the memmory it'e
necessary to increase the number of MiniBatches. Need to be a integer
number 1 or >1.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

miniBatchSize = 1; % Size of Mini-batches
sequenceLength = max(sequenceLengths);%Max length of the training dataset
files if the files are not in the same length, this will determine the
amount of padding on the shorter files. If all files have the same length,
there will be no padding .
RRR=readall(fdsPredictorTest); % read all Test data files
numObservationsY = numel(RRR);
for i=1:numObservationsY
    sequence = RRR{i}; %Combine Test data set files into one structure.
    sequenceLengthsY(i) = size(sequence.X,2);
end
sequenceLengthY= max(sequenceLengthsY); %Max length of the objective
function of the training dataset (SOC or Ah)
RRTest=readall(fdsPredictorTest);
numObservationsTest = numel(RRTest);

%*** Transforming Datastore dataset
tdsObjectiveTrain = transform(fdsObjectiveTrain,@(data)
padSequenceY(data,sequenceLength)); % Padding function - Objective function
Training Dataset
tdsObjectiveTest = transform(fdsObjectiveTest,@(data)
padSequenceY(data,sequenceLengthY)); % Padding function - Objective
function Testing Dataset
ObjectiveTest=readall(tdsObjectiveTest); %Objective function Test dataset,
this is only to help manipulate the dataset for plotting and calculate the
Errors.
ObjectiveTrain=readall(tdsObjectiveTrain);
tdsTrain = transform(fdsPredictorTrain,@(data)
padSequence(data,sequenceLength)); % Padding function - Training Dataset
tdsTest = transform(fdsPredictorTest,@(data)
padSequence(data,sequenceLengthY)); % Padding function - Testing Dataset
% preview(tdsTrain);
PredictorTest=readall(tdsTest);
PredictorTrain=readall(tdsTrain);
cdsTrain = combine(tdsTrain,tdsObjectiveTrain);
cdsTest = combine(tdsTest,tdsObjectiveTest);
% preview(cdsTrain)

%***Validation dataset
V=1; % Selection of the Validation dataset, in this script the validation
dataset is the same as Test dataset, but it can be changed
XTest=PredictorTest{V}; % predictors Validation dataset
YTest=ObjectiveTest{V}; % Objective function "Y == SOC" for the Validation
dataset
XTrain=PredictorTrain{V};
YTrain=ObjectiveTrain{V};

%***Specify NN training parameter
numResponses = 1; %Number of outputs (SOC)
numFeatures = 3; %Number of inputs features (V, I, T)
numHiddenUnits = 10; %Number of hidden units 'N', where each hidden unit
for LSTM represents a Neuron.
Epochs = 150; %Number of Epochs
LearnRateDropPeriod = 2000; %Number of epochs for dropping the learning
rate
InitialLearnRate=0.01; %Initial Learning rate. Very important parameter
LearnRateDropFactor=0.1; %Factor for dropping the learning rate, ex.:
reduce to 10% of the current learning rate at every 'LearnRateDropPeriod

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

validationFrequency =2; %Validate the model at each Iterations when values
is equal to 1.
repeat=1; %Select the amount the training process will be repeated

%***LSTM Structure
layers = [sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits,'OutputMode','sequence') % number of hidden
units in this example is '10'
    fullyConnectedLayer(numResponses)
    clippedReluLayer(1)
    regressionLayer];

options = trainingOptions('adam', ... % Adam optimizer
    'MaxEpochs',Epochs,'ExecutionEnvironment','auto', ...% Select GPU if
you works on computer with GPU
    'GradientThreshold',1, ...
    'InitialLearnRate',InitialLearnRate, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',LearnRateDropPeriod, ...
    'LearnRateDropFactor',LearnRateDropFactor, ...
    'L2Regularization',1, ...
    'ValidationData', {XTest,YTest}, ...
    'Shuffle','never',...
    "Verbose",1,...
    'ValidationFrequency',validationFrequency, ...
    'MiniBatchSize',miniBatchSize, ...
    'Plots','training-progress',...
    'CheckpointPath', NET_Path); % to save each trained model during the
training process, if necessary it can be "commented-out" and will only the
last model will be available to be save.

%***Train the LSTM network with the specified training options.
for i=1:repeat
    net = trainNetwork(cdsTrain,layers,options);
    N=50; % To remove the first 'N' and last 'NN' datapoints to compensate
the lack of previous data, which wouldn't normally occur in a real life xEV
application
    NN=50;
%*Test Network
    YT=readall(fdsObjectiveTest);
    XT=readall(fdsPredictorTest);
    X_T=[];Y_Test=[];X_Test=[];Y_T=[];
    Error_Test=[];
    Nets{i} = net;% save the last trained model at the end of each repeat
    Y_Pred_25degC = predict(net,XT{1, 1}.X,'MiniBatchSize',miniBatchSize);

    Error_Test_25degC = Y_Pred_25degC(:,N:end-NN)-XT{1, 1}.Y(:,N:end-NN);

    E_Test_RMSE_25degC{i} = sqrt(mean((Error_Test_25degC).^2))*100; %SOC
Test dataset root mean square error
    E_Test_MAE_25degC{i} = mae(Error_Test_25degC)*100; %SOC Test dataset
Mean abs error
    E_Test_MAX_25degC{i} = max(abs(Error_Test_25degC)*100); %max SOC error
of Test dataset

end
RMSE_SUMMARY = ['25degC_RMSE',E_Test_RMSE_25degC;];
MAE_SUMMARY = ['25degC_MAE',E_Test_MAE_25degC;];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAX_SUMMARY = ['25degC_MAX',E_Test_MAX_25degC];
***Based on the best result from the above figure select the best #repeat
to plot the on time domain
[Minimum_25degC, idx] = min(cell2mat(E_Test_RMSE_25degC));
SelectNet =idx;
    %SelectNet = 1; % Manual selection
Y_Pred_25degC = predict(Nets{SelectNet},XT{1,
1}.X,'MiniBatchSize',miniBatchSize);
Error_Test_25degC = Y_Pred_25degC(:,N:end-NN)-XT{1, 1}.Y(:,N:end-NN);

figure
subplot(2,1,1)
grid on
hold all;
plot(XT{1, 1}.Y(:,N:end-NN)*100,'LineWidth',3,'LineStyle','-');
plot(Y_Pred_25degC*100,'LineWidth',2,'LineStyle','-');
legend(["Observed" "Predicted"])
ylabel("(SOC)")
xlabel('Time(s)')
title("Test - SOC@25degC - Estimation")
grid on
subplot(2,1,2)
grid on
hold all;
plot((Error_Test_25degC)*100,'LineWidth',2,'LineStyle','-');
legend("Error")
ylabel("(%)")
xlabel('Time(s)')
title("Error")
***Function
function sequence = padSequence(data,sequenceLength)
sequence = data.X;
[C,S] = size(sequence);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequence,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequence = [sequence padding];
else
    sequence = sequence(:,1:sequenceLength);
end
sequence = {sequence};
end
function sequenceY = padSequenceY(data,sequenceLength)
sequenceY = data.Y;
[C,S] = size(sequenceY);
if S < sequenceLength
    N=round((sequenceLength/S)*1);
    Pad= repmat(sequenceY,1,N);
    padding = Pad(:,1:sequenceLength-S);
    sequenceY = [sequenceY padding];
else
    sequenceY = sequenceY(:,1:sequenceLength);
end
sequenceY = {sequenceY};
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้จะแสดงผลของการดำเนินงานที่ได้จากการออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน มีการทดสอบโครงข่ายประสาทเทียมแบบ FNN, RNN และ LSTM โดยทำการใช้ชุดข้อมูลเดียวกันและสลับชุดข้อมูลทั้งหมด 4 กรณี ได้แก่

กรณีที่ 1 ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

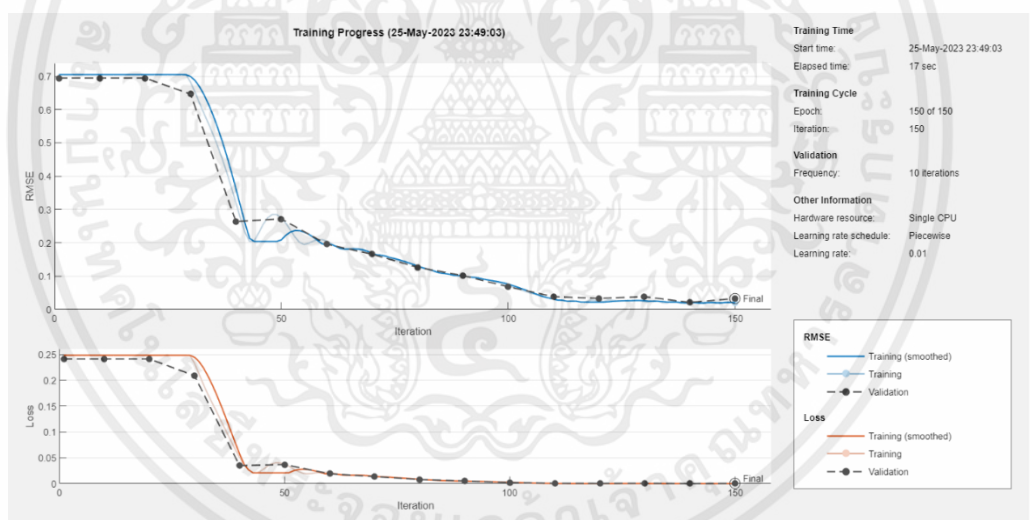
กรณีที่ 2 ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

กรณีที่ 3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

และกรณีที่ 4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3 ได้ผลการทดลองดังนี้

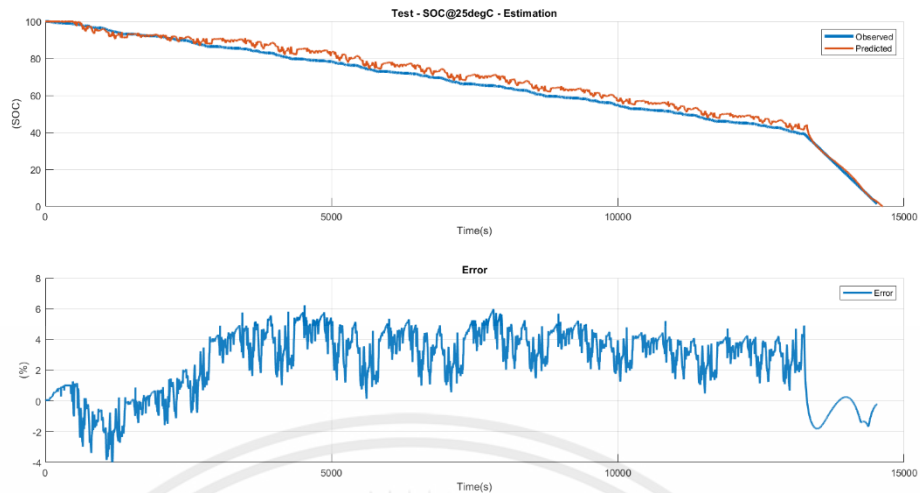
4.1 ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

- Feed-forward neural network ได้ค่า RMSE = 3.27 %



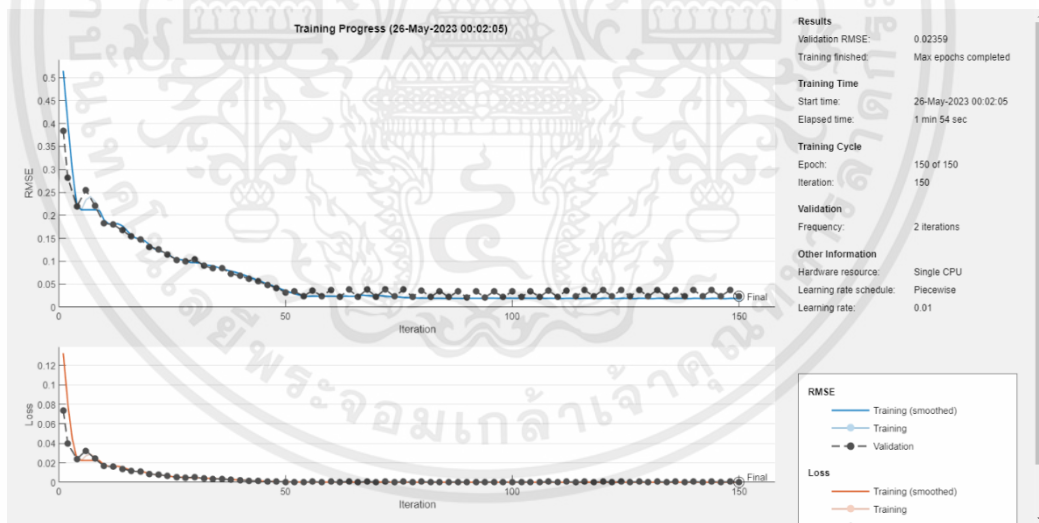
รูปที่ 4.1 กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



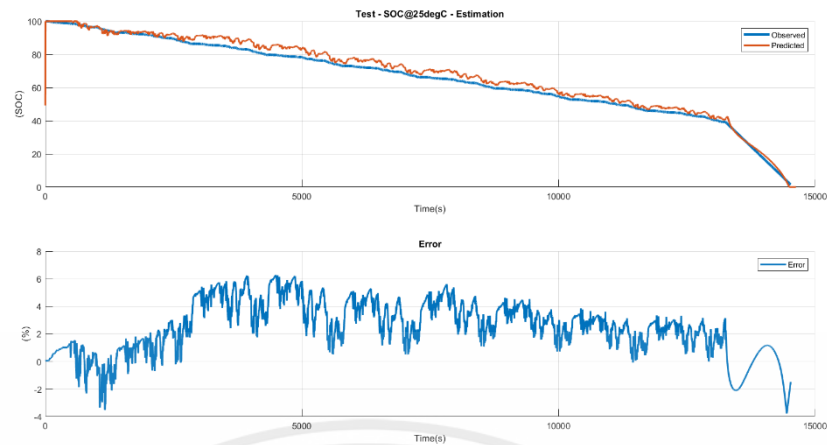
รูปที่ 4.2 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

- Recurrent neural network ได้ค่า RMSE = 3.12 %



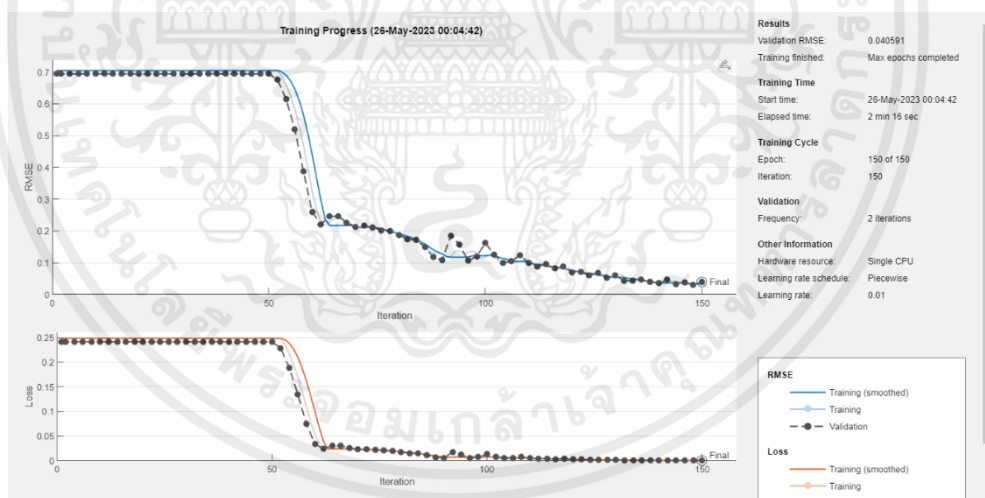
รูปที่ 4.3 กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



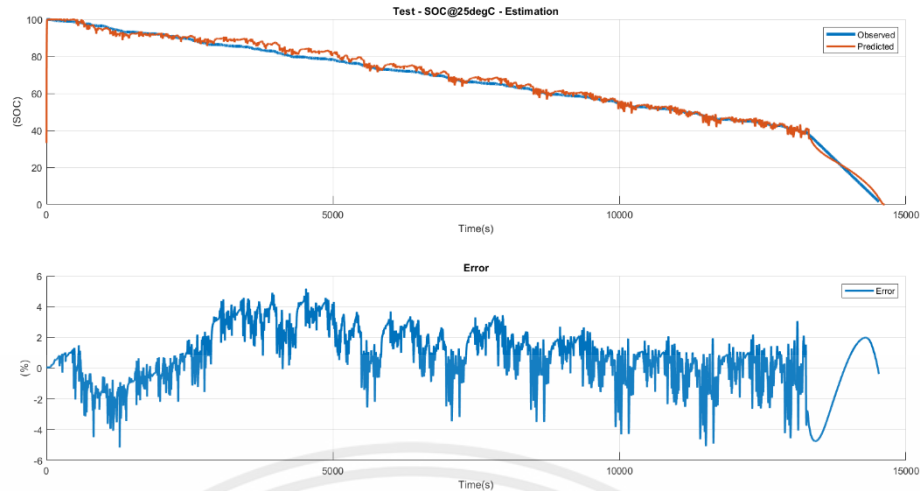
รูปที่ 4.4 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

- Long short-term memory neural network ได้ค่า RMSE = 2.02 %



รูปที่ 4.5 กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

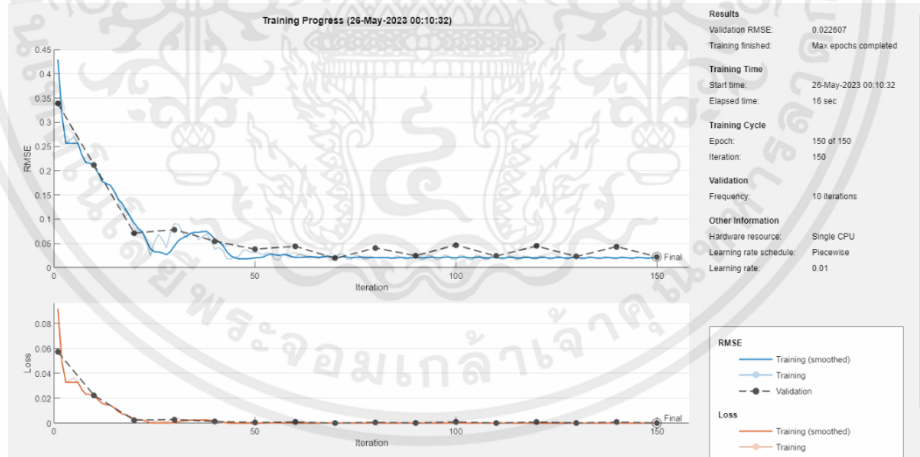
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

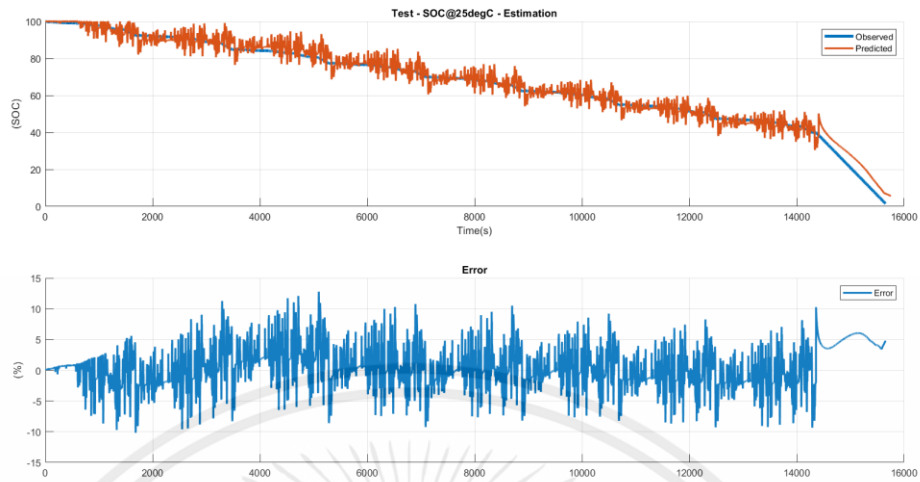
4.2 ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

- Feed-forward neural network ได้ค่า RMSE = 3.03 %

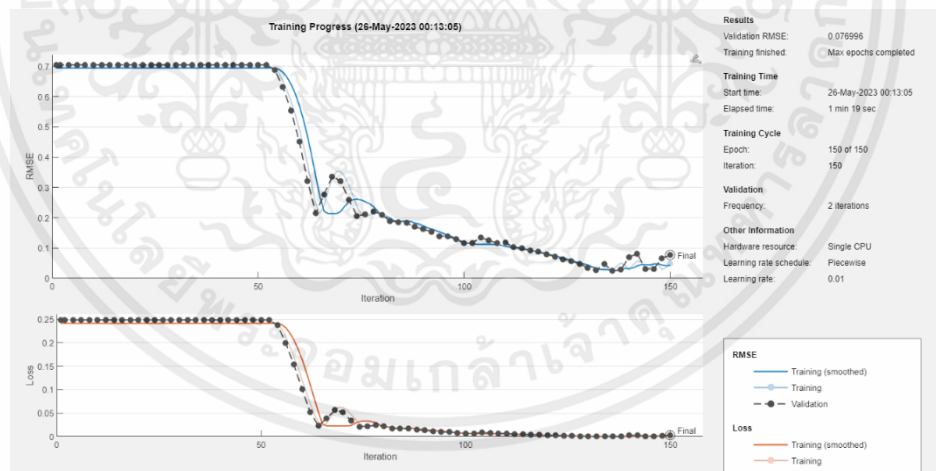


รูปที่ 4.7 กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

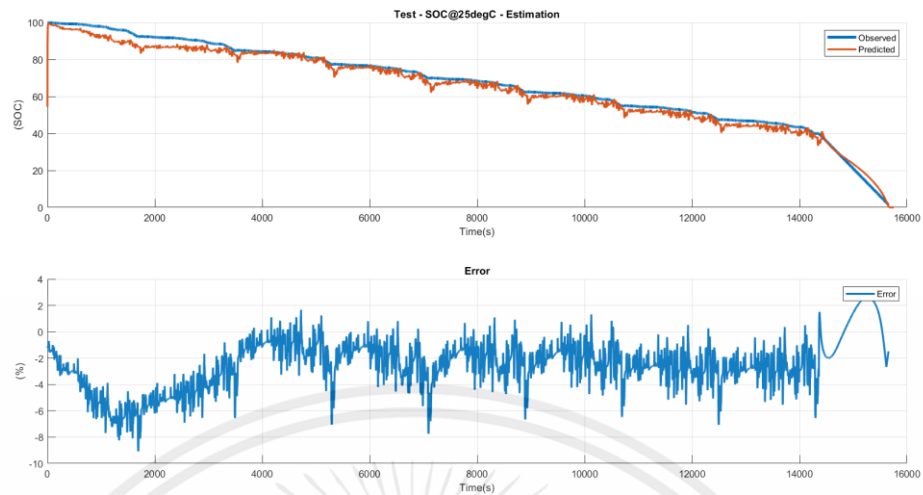


รูปที่ 4.8 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3
- Recurrent neural network ได้ค่า RMSE = 3.11 %

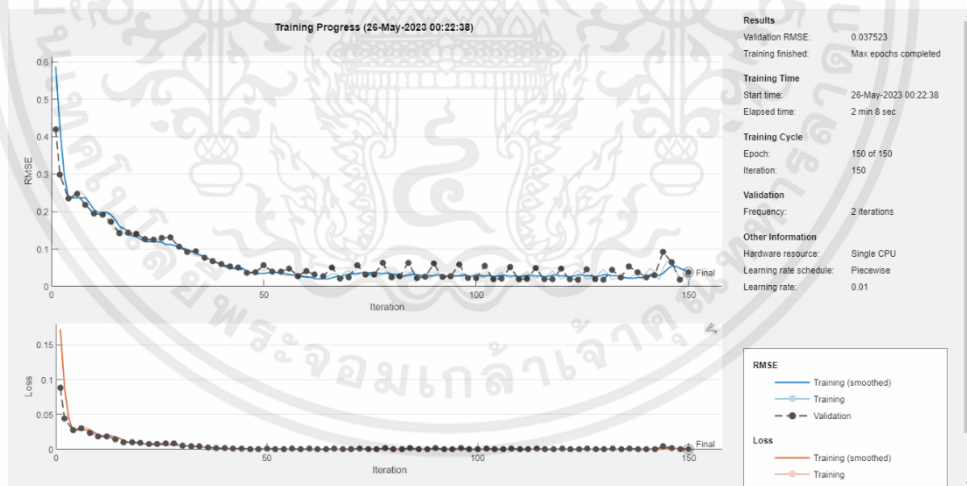


รูปที่ 4.9 กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

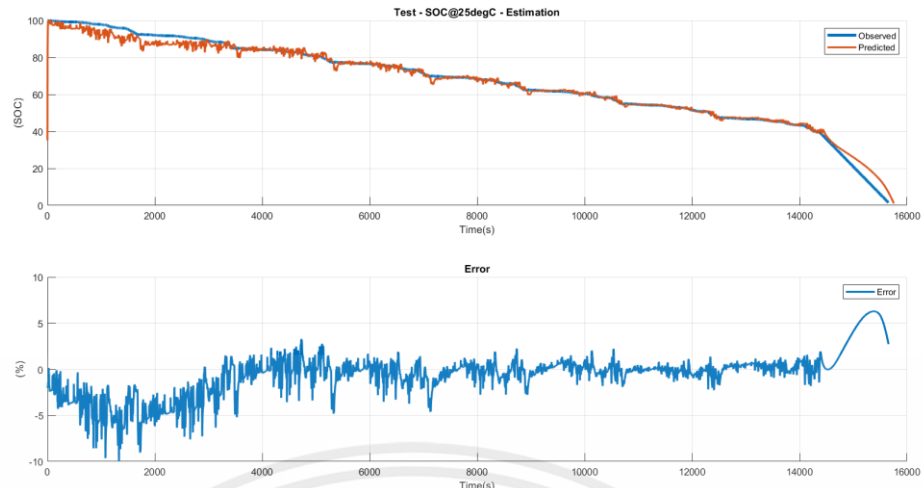


รูปที่ 4.10 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3 - Long short-term memory neural network ได้ค่า RMSE = 2.39 %



รูปที่ 4.11 กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

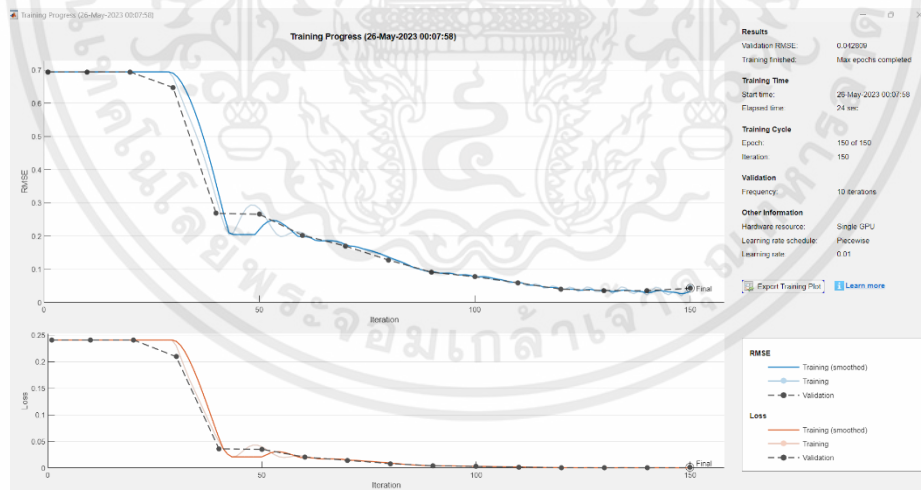
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

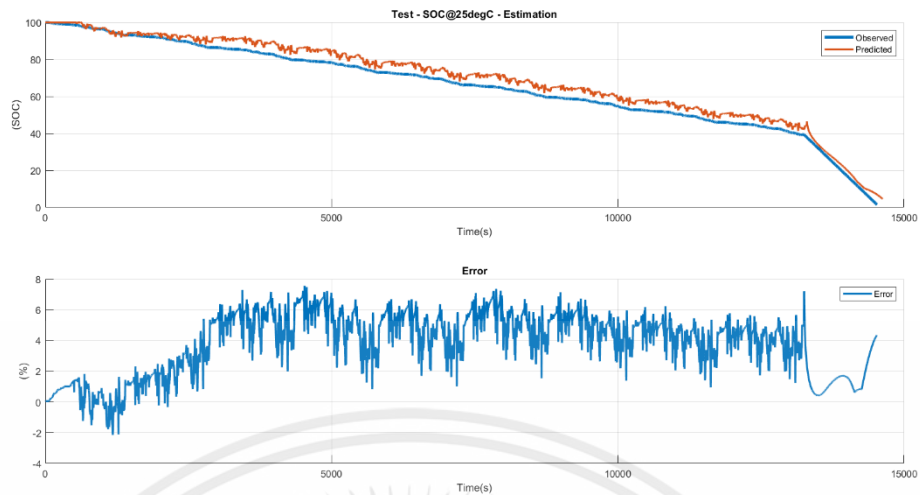
4.3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

- Feed-forward neural network ได้ค่า RMSE = 4.29 %



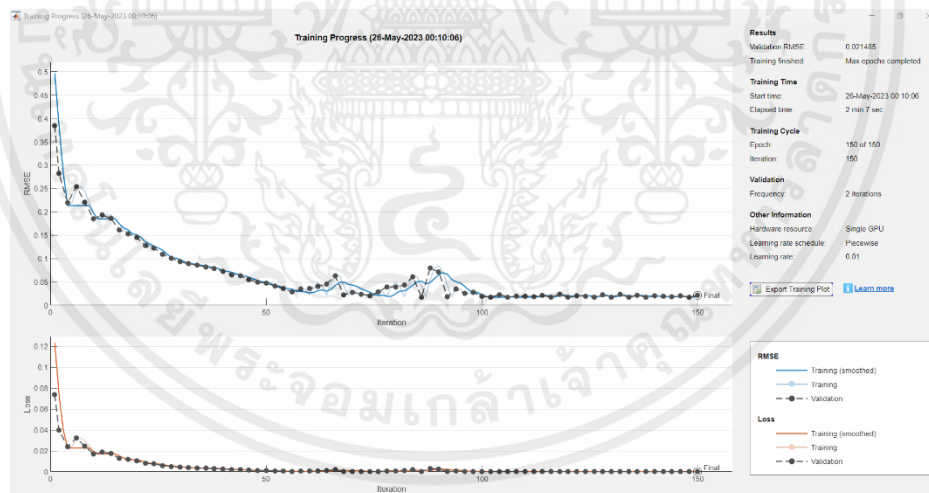
รูปที่ 4.13 กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



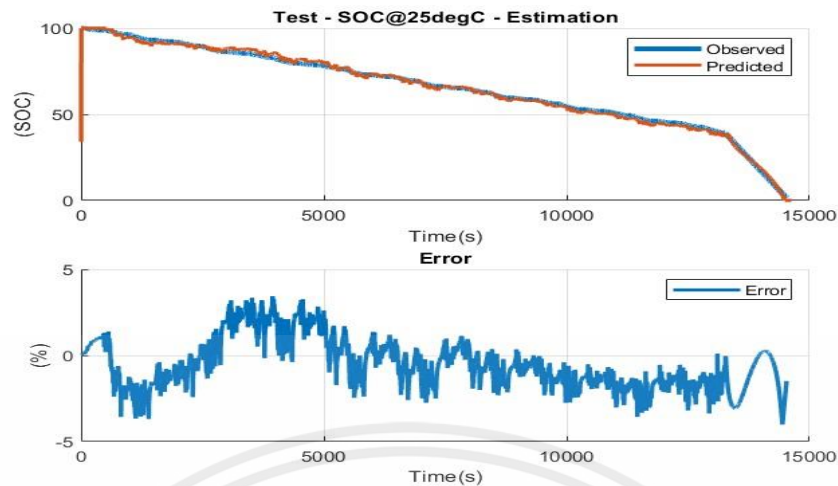
รูปที่ 4.14 กราฟการทำนาย SoC และ เปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

- Recurrent neural network ได้ค่า RMSE = 2.18 %



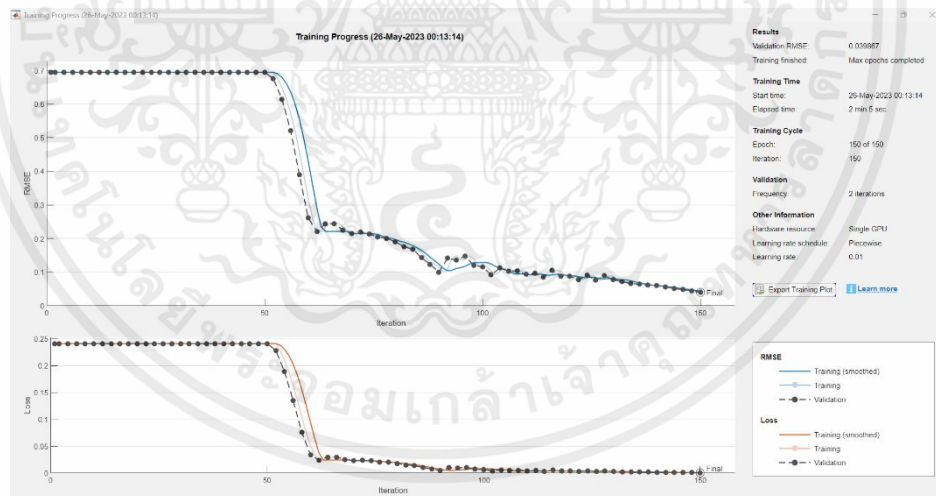
รูปที่ 4.15 กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



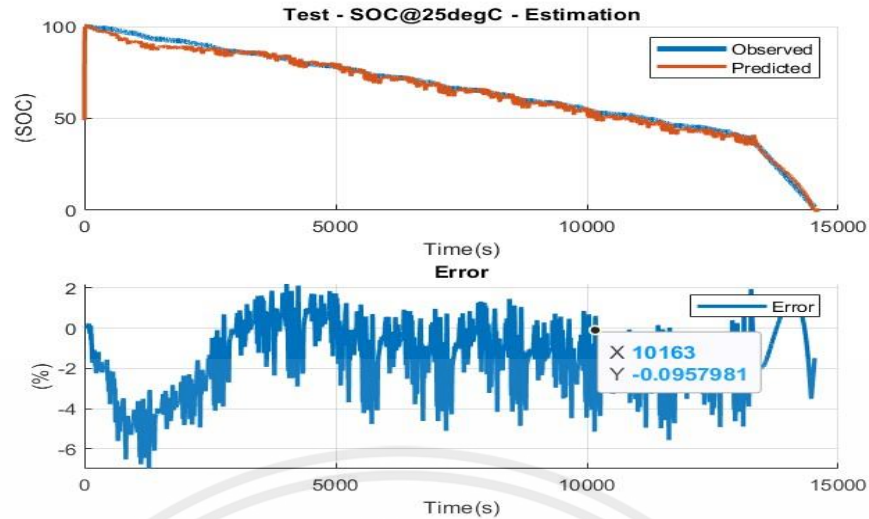
รูปที่ 4.16 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

- Long short-term memory neural network ได้ค่า RMSE = 1.55 %



รูปที่ 4.17 กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

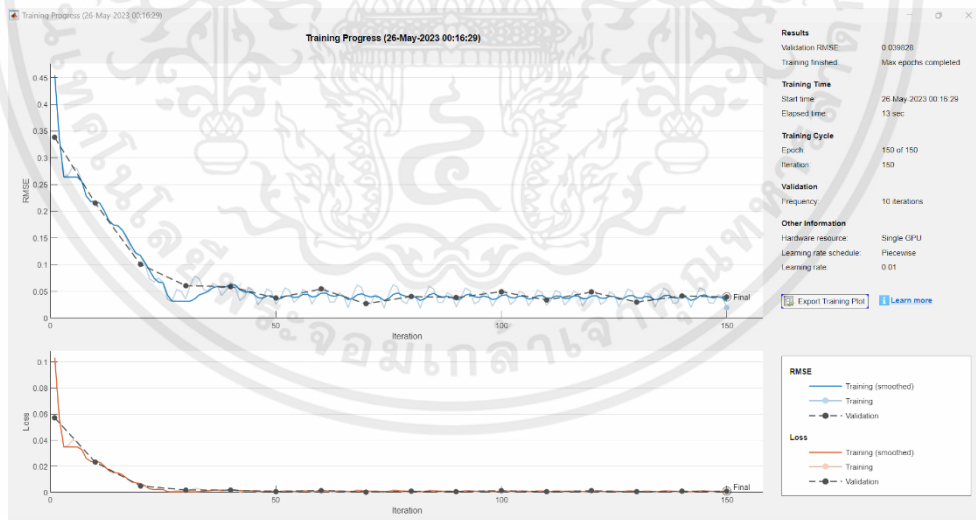
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

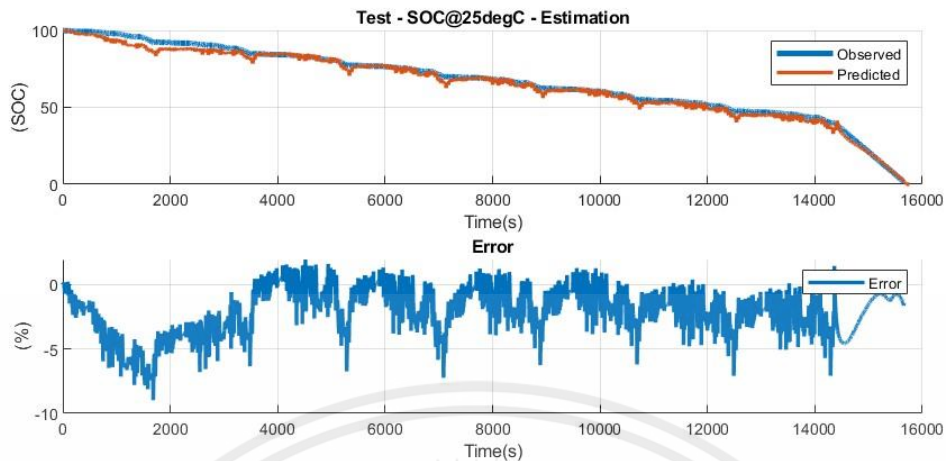
4.4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

- Feed-forward neural network ได้ค่า RMSE = 2.69 %



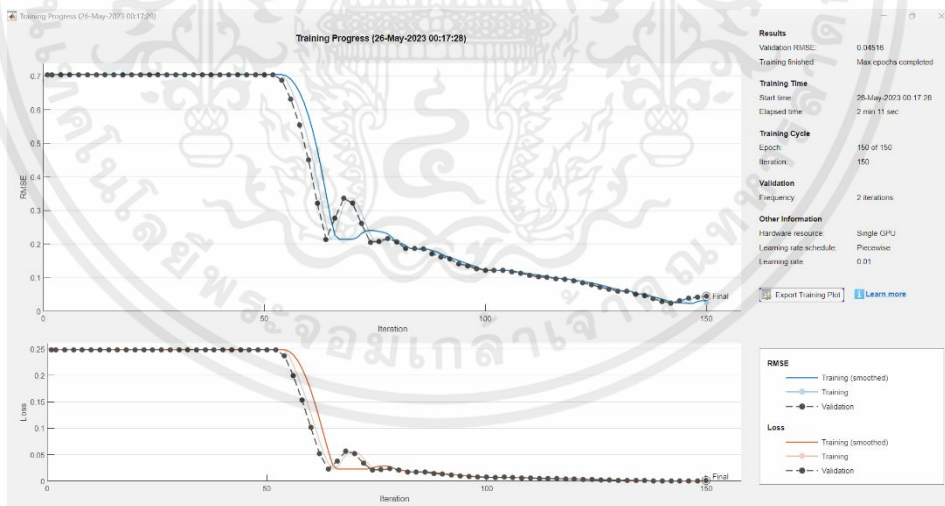
รูปที่ 4.19 กราฟผลลัพธ์การ train โครงข่าย Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



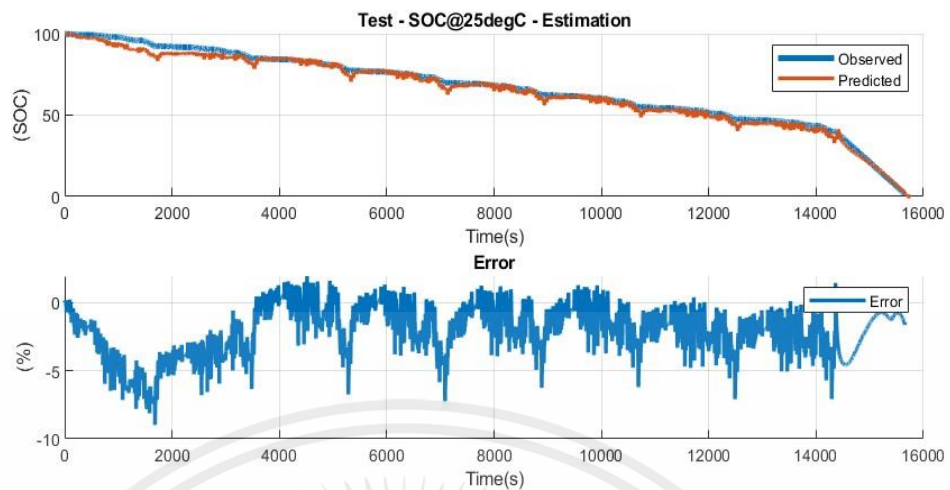
รูปที่ 4.20 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Feed-forward neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

- Recurrent neural network ได้ค่า RMSE = 1.67 %



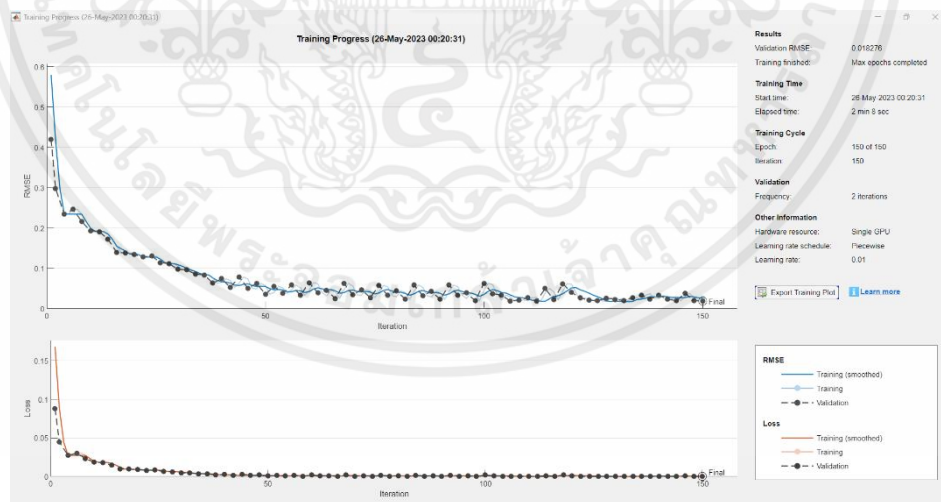
รูปที่ 4.21 กราฟผลลัพธ์การ train โครงข่าย Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



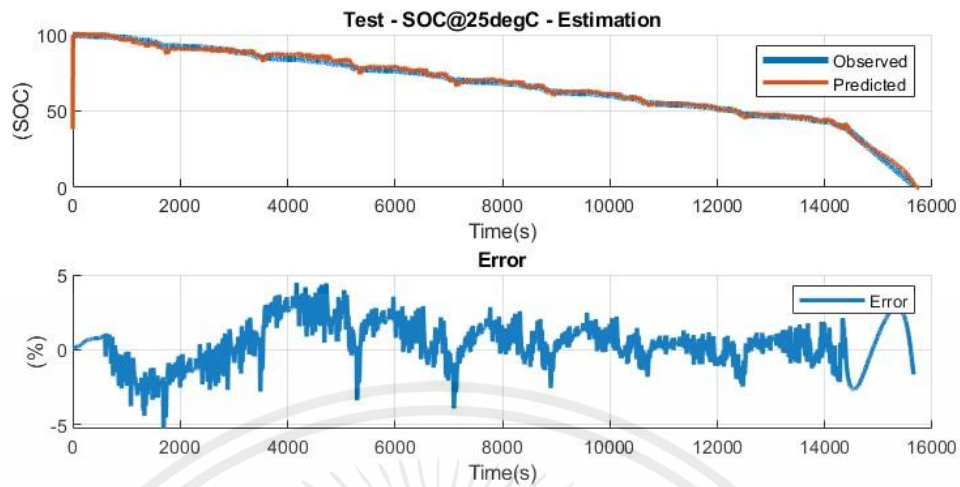
รูปที่ 4.22 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Recurrent neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

- Long short-term memory neural network ได้ค่า RMSE = 1.48 %



รูปที่ 4.23 กราฟผลลัพธ์การ train โครงข่าย Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 กราฟการทำนาย SoC และเปอร์เซ็นต์ Error ของ Long short-term memory neural network ที่ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

ตารางที่ 4.1 สรุปค่า Root Mean Squared Error (RMSE) ของแต่ละกรณี

ประเภทโครงข่ายประสาทเทียม	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3	กรณีที่ 4
FNN (%)	3.27	3.03	4.29	2.69
RNN (%)	3.12	3.11	2.18	1.67
LSTM (%)	2.02	2.39	1.55	1.48

จากการทดสอบโครงข่ายประสาทเทียมแบบ FNN RNN และ LSTM โดยทำการใช้ชุดข้อมูลเดียวกันและสลับชุดข้อมูลทั้งหมด 4 กรณี พบว่าโครงข่ายประสาทเทียมแบบ LSTM ให้ค่า RMSE น้อยที่สุดในทุกกรณี ซึ่งแสดงให้เห็นถึงความแม่นยำในการประมาณสถานะประจุแบตเตอรี่ชนิดลิเทียมไอออนโดยใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM)

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

ปริญญาานิพนธ์ฉบับนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อการออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน เนื่องจากแบตเตอรี่ลิเทียมเป็นที่นิยมใช้ในรถยนต์ไฟฟ้า ซึ่งมีคุณสมบัติที่เหมาะสมกับการนำมาใช้งานในยานยนต์ไฟฟ้ามากที่สุดเนื่องจากคุณสมบัติด้านความหนาแน่น พลังงานที่สูง และในส่วนของประกอบของธาตุลิเทียมนั้นมีเซลล์ไฟฟ้าเคมีที่สูงกว่าเซลล์จากโลหะอื่น เป็นสาเหตุให้แบตเตอรี่มีแรงดันไฟฟ้าที่สูงกว่าแบตเตอรี่ชนิดอื่น อีกทั้งแบตเตอรี่ลิเทียมมีขนาดและน้ำหนักน้อยกว่าแบตเตอรี่ตะกั่วและแบตเตอรี่นิกเกิลที่มีความจุเท่ากัน การประมาณสถานะประจุสามารถทำได้หลายวิธี ตัวอย่างเช่น วิธีการนับคูลอมบ์ (Coulomb Counting) แต่ด้วยวิธีนี้จะเกิดข้อผิดพลาดจากค่าออฟเซตและการคายประจุด้วยตัวเอง และ วิธี Kalman-Filter มีความยุ่งยากและซับซ้อน อีกทั้งต้องใช้ชุดสมการที่ซับซ้อนมาก คณะผู้จัดทำจึงเสนอการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมที่มีการทำงานที่ไม่ซับซ้อนและเปรียบเสมือนกับ Black box ที่ต้องการแค่ Input และ Output ป้อนเข้าสู่ระบบ โดยไม่สนใจกระบวนการทำงานภายในว่าทำงานอย่างไร เพื่อทำการประมาณสถานะประจุแบตเตอรี่ลิเทียมไอออน

จากการทดลองเพื่อออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการทดลองนี้จะเป็นการดำเนินการผ่านโปรแกรม MATLAB โดยกำหนดสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้ว กระแสไฟฟ้า และอุณหภูมิ และกำหนดสัญญาณขาออก คือ State of Charge (SoC) และมีการกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับ โดยมีการทดสอบทั้งหมด 4 กรณี ได้แก่

กรณีที่ 1 ใช้ชุดข้อมูล Train เป็น WLTP Class 3 และ Test เป็น WLTP Class 2

กรณีที่ 2 ใช้ชุดข้อมูล Train เป็น WLTP Class 2 และ Test เป็น WLTP Class 3

กรณีที่ 3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 2

และกรณีที่ 4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP Class 3

เมื่อพิจารณาและเปรียบเทียบค่า Root Mean Squared Error (RMSE) พบว่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุดและรองลงมาคือ SoC ที่ได้จากโครงข่าย RNN และ FNN ตามลำดับ เนื่องจากข้อมูลที่ใช้ในการประมาณสถานะประจุแบตเตอรี่ลิเทียมไอออนเป็นชุดข้อมูลอนุกรมเวลา กล่าวคือข้อมูลเวลาในปัจจุบันกับข้อมูลเวลาก่อนหน้ามีความสัมพันธ์กัน ซึ่งข้อมูลที่มีลักษณะเป็นลำดับ เช่น Text (Sequence of words) ยกตัวอย่างเปรียบเทียบกับกรอ่านหนังสือ ซึ่งก็คือลำดับของคำที่ต่อกันอ่านทีละคำจากซ้ายไปขวา (สำหรับภาษาไทยหรือภาษาอังกฤษ) การที่จะสามารถรู้เรื่องได้ว่าประโยคที่กำลังอ่านนั้นเกี่ยวกับอะไร ก็ต้องเอาเรื่องราวจากสิ่งที่อ่านผ่านไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาผสมกับค่าที่เพิ่งอ่านไปทำให้สามารถเข้าใจความหมายในส่วนที่กำลังอ่านได้ ซึ่งการประมาณโดยใช้ Feed-forward Neural Network (FNN) จะมองแค่ข้อมูลในปัจจุบันจึงมีปัญหาเกี่ยวกับข้อมูลที่เป็นลำดับ แต่ Recurrent Neural Network (RNN) ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับข้อมูลที่เป็นลำดับ โดย RNN มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคตซึ่งหมายถึงข้อมูลในปัจจุบัน เป็นผลมาจากข้อมูลในอดีต ถึงแม้ว่า RNN จะสามารถใช้ข้อมูลก่อนหน้ามาคาดการณ์สถานะประจวบแต่เตอร์ได้ แต่ก็ยังเป็นข้อมูลก่อนหน้าที่เป็นช่วงระยะเวลาสั้น ๆ เท่านั้น จึงมีการใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM) ซึ่งสามารถเก็บข้อมูลที่เก่ากว่าไว้ในหน่วยความจำได้ ด้วยสาเหตุนี้จึงทำให้ค่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุด และรองลงมาคือค่า SoC จาก โครงข่าย RNN ที่มี Short term memory และอันดับสุดท้ายคือค่า SoC จาก โครงข่าย FNN

5.2 ปัญหาอุปสรรคและแนวทางแก้ไข

1. ค่า Root Mean Squared Error (RMSE) ที่ได้จากการประมาณสถานะประจวบแต่เตอร์ยังถือว่ามีความสูง ซึ่งสามารถแก้ไขได้โดยการป้อนข้อมูล Input ให้มากขึ้น และเพิ่มข้อมูลอุณหภูมิในการทดสอบหรือปรับพารามิเตอร์ของโครงสร้างของโครงข่ายประสาทเทียม
2. ชุดข้อมูลที่นำมาทำการทดสอบมีแค่อุณหภูมิเดียวคือ 25 องศาเซลเซียส จึงไม่สามารถทำการทดลองเปรียบเทียบได้ว่าหากมีหลายอุณหภูมิโครงข่ายประสาทเทียมแต่ละชนิดจะมีค่า RMSE ไกล่เคียงกันหรือไม่ และโครงข่ายประสาทเทียมชนิด LSTM จะยังสามารถประมาณค่า SoC ได้แม่นยำที่สุดหรือไม่

5.3 แนวทางการพัฒนาต่อ

1. ทำการศึกษาหรือทำการทดลองที่อุณหภูมิอื่น ๆ เพื่อให้โครงข่ายประสาทเทียมมีความแม่นยำมากยิ่งขึ้น
2. นำโค้ดที่สร้างไปพัฒนาต่อในการทดลองใช้กับ Hardware เพื่อจำลองการใช้งานการประมาณสถานะประจวบแต่เตอร์ลิเทียมไอออน

เอกสารอ้างอิง

- [1] สถาบันยานยนต์ กระทรวงอุตสาหกรรม. ความรู้ยานยนต์ไฟฟ้าเบื้องต้น. <http://www.thaiauto.or.th/th/services/ev/default.asp>, 2012.
- [2] Manjot Kaur and Aakash Mohta, “A Review of Deep Learning with Recurrent Neural Network,” 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019, pp 460-465.
- [3] Sepp Hochreiter and Jürgen Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, issue 8, 1997, pp 1735-1780.
- [4] คลังความรู้ Scimath. แบตเตอรี่: ตอนที่ 2 อุปกรณ์ให้พลังงานแห่งอนาคต [ออนไลน์]. สืบค้นจาก <https://www.scimath.org/article-physics/item/12467-2-2> .
- [5] JB BATTERY. ข้อดีและข้อเสียของแบตเตอรี่ลิเทียมไอออน.[ออนไลน์]. สืบค้นจาก <https://www.lithiumbatterychina.com/th/blog/the-pros-and-cons-of-ion-batteries-with-advantages-and-disadvantages/>, 2021.
- [6] สถาบันยานยนต์ กระทรวงอุตสาหกรรม. ความรู้ยานยนต์ไฟฟ้าเบื้องต้น. [ออนไลน์]. สืบค้นจาก <http://www.thaiauto.or.th/th/services/ev/default.asp>, 2012.
- [7] Ana-Irina Stan, Maciej Swierczynski. Lithium-Ion Battery Chemistries from Renewable-Energy Storage to Automotive and Back-up Power Applications - An Overview. [ออนไลน์]. สืบค้นจาก <https://ieeexplore.ieee.org/document/6850936>, 2014.
- [8] howstuffworks. How Lithium-ion Batteries Work [ออนไลน์]. สืบค้นจาก <https://ics-howstuff-works.com/everyday-tech/lithium-ion-battery.htm#pt1>.
- [9] howstuffworks. How Lithium-ion Batteries Work [ออนไลน์]. สืบค้นจาก <https://ics-howstuff-hworks.com/everyday-tech/lithium-ion-battery.htm#pt1>.
- [10] howstuffworks. How Lithium-ion Batteries Work [ออนไลน์]. สืบค้นจาก <https://ics-howstuffworks.com/everyday-tech/lithium-ion-battery.htm#pt1>.
- [11] M. Einhorn, Member, IEEE, F.V. Conte, C. Kral, Senior Member, IEEE, and J. Fleig, A Method for Online Capacity Estimation of Lithium-Ion Battery Cells Using the State of Charge and the Transferred Charge. [ออนไลน์]. สืบค้นจาก <https://ieeexplore.ieee.org/document>.
- [12] Hiroyuki Miyamoto. On-line SOC Estimation of Battery for Wireless Tram Car. [ออนไลน์]. สืบค้นจาก <https://ieeexplore.ieee.org/document/4487927>, 2007.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- [13] Yimin Zhou and Xiaoyun Li. Overview of Lithium-ion Battery SOC Estimation. [ออนไลน์]. สืบค้นจาก <https://ieeexplore.ieee.org/abstract/document>, 2015.
- [14] BSLBATT. สถานะการชาร์จของแบตเตอรี่ลิเทียม [ออนไลน์]. สืบค้นจาก <https://www.lithium-battery-factory.com/th/lithium-battery-state-of-charge/>
- [15] ธนาวุฒิ ประกอบผล. “โครงข่ายประสาทเทียม = Artificial neural networks”. วารสาร มฉก. วิชาการ. ปีที่ 12 ฉบับที่ 24 (ม.ค.-มิ.ย. 2552) หน้า 72-87, 2552.
- [16] L. Thanakrit. เบื้องหลังเทคโนโลยีอัจฉริยะ ตอนที่2: Neural Network. สืบค้นจาก <https://thanakrit.net/neural-network/>, 2016
- [17] Mr. L. Deep Learning แบบฉบับคนสามัญชน EP 1: Neural Network History. [ออนไลน์]. สืบค้นจาก <https://medium.com/mmp-li/deep-learning>, 2019.
- [18] Medium. Deep Learning แบบฉบับคนสามัญชน EP 1: Neural Network History. สืบค้นจาก <https://medium.com/mmp-li/deep-learning>, 2019
- [19] SELDON. Neural Network Models Explained. [ออนไลน์]. สืบค้นจาก <https://www.seldon.io/neural-network-models-explained>, 2022.
- [20] pack. Feed-forward and feedback networks. [ออนไลน์]. สืบค้นจาก <https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781788397872/1/-/ch01/vl1sec21/feed-forward-and-feedback-networks>
- [21] HashDork. ความรู้เบื้องต้นเกี่ยวกับโครงข่ายประสาทเทียม. สืบค้นจาก <https://hashdork.com/-th/>, 2022.
- [22] MathWorks. Recurrent Neural Network (RNN). [ออนไลน์]. สืบค้นจาก https://www.mathworks.com/discovery/rnn.html?s_tid=srchtitle_Recurrent%20neural%20networks_1, 2022.
- [23] BUA Labs. Recurrent Neural Network (RNN) คืออะไร. สืบค้นจาก <https://www.bualab.com/archives/3103/what-is-rnn-recurrent-neural-network-what-is-gru-gated-recurrent-un-it-teach-how-to-build-rnn-gru-with-python-nlp-ep-9/>, 2019
- [24] Tiago Miguel. How the LSTM improves the RNN. [ออนไลน์]. สืบค้นจาก <https://towardsdatascience.com/how-the-lstm-improves-the-rnn-1ef156b75121>, 2021
- [25] Ashutosh Tripathi. WHAT IS THE MAIN DIFFERENCE BETWEEN RNN AND LSTM | NLP | RNN VS LSTM. [ออนไลน์]. สืบค้นจาก <https://ashutoshtripathi.com/2021/07/02/-what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/>, 2021.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- [26] Sirinart Tangruamsub. Long Short-Term Memory (LSTM). [ออนไลน์]. สืบค้นจาก <https://-medium.com/@sinart.t/long-short-term-memory-lstm>, 2017.
- [27] Medium. Long Short-Term Memory (LSTM). สืบค้นจาก <https://medium.com/@sinart.-/long-short-term-memory-lstm-e6cb23b494c6>, 2017
- [28] AJ AUTODEFT KNOW IT BETTER. ทำความรู้จักการมาตรฐานระยะทางของรถไฟฟ้าจาก EPA, NEDC และ WLTP. [ออนไลน์]. สืบค้นจาก <https://www.autodeft.com/deftanswer/how-to-epa-nedc-wltp-test-electric-cars>.
- [29] Emission Test Cycles. Worldwide Harmonized Light Vehicles Test Cycle (WLTC). [ออนไลน์]. สืบค้นจาก <https://dieselnet.com/standards/cycles/wltp.php>.
- [30] Z. Dongchang, R. Chunjia, C.Chuan, Y. Rujie. From NEDC to WLTP for Vehicles: The Impact on Fuel Efficiency Calculation and Algorithm Optimization. [ออนไลน์]. สืบค้นจาก <https://ieeexplore.ieee.org/document/9974136>, 2022.
- [31] John Deere. Worldwide Harmonized Light Vehicles Test Cycle (WLTC), [ออนไลน์]. สืบค้นจาก <https://dieselnet.com/standards/cycles/wltp.php>, 2019.
- [32] Engineering Computer Center. โปรแกรม MATLAB. สืบค้นจาก <https://www.ecc.eng.kmut-nb.ac.th/index.php/2016/12/19/software-matlab-20161219/>, 2016.
- [33] โศรฎา แข็งการ, กนต์ธร ชำนิประศาสน์. การใช้ Matlab ในการคำนวณทางวิศวกรรม. Tools online, 2011 หน้า1-2.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน

โดยการใช้โครงข่ายประสาทเทียม

State of Charge Estimation of a Lithium Ion Battery

Using the Artificial Neural Network

นางสาวกนกวรรณ ตั้งกาญจนกิจ, นางสาวกฤษณา พันธุ์แดง, นางสาวกาญจนาพร พูลเกิด และ นางสาวพิมพ์ทอง กวาวลีสาม

อาจารย์ที่ปรึกษา รศ.ดร.สุวัฒน์ กิตติรัตน์สังจา, ดร.สมภพ ผลไม้

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1 ซอย ฉลองกรุง 1 ลาดกระบัง กรุงเทพมหานคร 10520 โทรศัพท์ 02-329-8000 ต่อ 3925 E-Mail: power@kmit.ac.th

ABSTRACT

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอเกี่ยวกับการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมเพื่อการประมาณสถานะที่แม่นยำและทำให้แบตเตอรี่มีอายุการใช้งานที่ยาวนาน โดยในปริญญานิพนธ์ฉบับนี้เสนอการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM) , Feed-forward Artificial Neural Network (FNN) และ Recurrent Neural Network (RNN) ดำเนินการผ่านโปรแกรม MATLAB โดยสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้ว กระแสไฟฟ้าและอุณหภูมิ และกำหนดสัญญาณขาออกคือ State of Charge (SoC) และมีกรกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับและมีการทดลองใช้ชุดข้อมูลที่ซ้ำกันและสลับชุดข้อมูล โดย WLTP ย่อมาจาก Worldwide Harmonized Light Vehicle Test Procedure ซึ่งเป็นมาตรฐานการวัดระยะทางรถยนต์ไฟฟ้า ในการประเมินประสิทธิภาพของการประมาณสถานะประจุโดยโครงข่าย LSTM FNN และ RNN ซึ่งพิจารณาจากค่า Root Mean Squared Error (RMSE) พบว่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุดและรองลงมาคือ SoC ที่ได้จากโครงข่าย RNN และ FNN ตามลำดับ ซึ่งแสดงให้เห็นถึงศักยภาพในการประมาณสถานะประจุแบตเตอรี่ชนิดลิเทียมไอออนโดยใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM)

คำสำคัญ : การประมาณสถานะประจุ, โครงข่ายประสาทเทียม

This thesis presents the estimation of the state of charge of lithium-ion batteries using artificial neural networks for accurate state estimation and longer battery life. The thesis proposes the estimation of the state of charge of a lithium-ion battery through the utilization of Long Short-Term Memory (LSTM) neural network, Feed-forward Artificial Neural Network (FNN), and Recurrent Neural Network (RNN). The estimation is implemented through a MATLAB program, where the input values entered into the network consist of three variables: inter-polar voltage, current, and temperature. The output is set to the state of charge (SoC).

The training set and testing set are defined as WLTP Class 3 and WLTP Class 2, respectively. Additionally, duplicate, and alternating datasets were also experimented with. WLTP stands for Worldwide Harmonized Light Vehicle Test Procedure, which is an electric vehicle mileage measurement standard. To evaluate the efficiency of charge state estimation by LSTM, FNN, and RNN networks, the Root Mean Squared Error (RMSE) was considered. It was found that the SoC obtained from the LSTM network was the most accurate, followed by the SoC obtained from the RNN and FNN networks, respectively. This finding demonstrates the potential for estimating the state of charge of lithium-ion batteries using the Long Short-Term Memory (LSTM) neural network.

Keywords: state of charge Estimation, neural network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

ในปัจจุบันรถยนต์ไฟฟ้า (Electric Vehicles : EV) ได้รับความสนใจมากขึ้น เนื่องจากเป็นมิตรกับสิ่งแวดล้อมและลดมลพิษทางอากาศ จึงจำเป็นต้องศึกษาค้นคว้าพฤติกรรมที่ไม่เป็นเชิงเส้นของแบตเตอรี่เพื่อพัฒนาประสิทธิภาพการทำงานของแบตเตอรี่สำหรับรถยนต์ไฟฟ้าให้ดีขึ้น โดยฟังก์ชันที่สำคัญอย่างหนึ่งของระบบการจัดการแบตเตอรี่คือการประมาณสถานะประจุของแบตเตอรี่ เพื่อตรวจสอบว่าแบตเตอรี่ยังคงเหลือประจุให้ใช้อีกเท่าไรในรถยนต์ไฟฟ้า จึงต้องมีการประมาณสถานะประจุที่มีความแม่นยำเพื่อรองรับความปลอดภัยในการใช้รถยนต์ไฟฟ้า

คณะผู้จัดทำนำเสนอการประมาณสถานะประจุของแบตเตอรี่โดยใช้ Artificial Neural Network (ANN) เนื่องจากชุดข้อมูลที่ป้อนเข้าสู่โครงข่ายเป็นชุดข้อมูลลำดับที่มีความยาวมาก จึงใช้ ANN ชนิดหนึ่งที่มีชื่อว่า Recurrent Neural Network (RNN) ในการประมาณสถานะประจุ แต่ RNN ไม่สามารถประมาณสถานะประจุที่มีความแม่นยำได้เพียงพอเนื่องจากไม่มีความทรงจำระยะยาวทำให้ลืมข้อมูลในช่วงแรก จึงมีการใช้ Long short-term memory (LSTM) ซึ่งมีประสิทธิภาพทางการคำนวณมากกว่าและสามารถจัดการข้อมูลในหน่วยความจำได้เป็นเวลานานกว่า ซึ่งเป็น RNN ชนิดหนึ่งที่มีทั้งความทรงจำระยะสั้นและระยะยาวทำให้สถานะประจุที่ได้จากการประมาณมีความแม่นยำมากยิ่งขึ้น

2. ทฤษฎีที่เกี่ยวข้อง

2.1 แบตเตอรี่

แบตเตอรี่ถือเป็นอุปกรณ์ทางไฟฟ้าที่สำคัญชิ้นหนึ่งของยานยนต์ไฟฟ้า เนื่องจากเป็นอุปกรณ์ที่ใช้ปฏิกิริยาทางเคมีไฟฟ้ากักเก็บพลังงานที่ใช้ในการขับเคลื่อน โดยส่วนประกอบหลักของแบตเตอรี่ประกอบด้วยส่วนหลักๆ 3 ส่วนคือ ขั้วบวก (แคโทด) ขั้วลบ (แอโนด) และสารละลายอิเล็กโทรไลต์ ซึ่ง เมื่อแบตเตอรี่ถูกใช้งานหรือเกิดการคายประจุไฟฟ้า ขั้วลบหรือขั้วแอโนดจะเป็นขั้วที่เกิดปฏิกิริยาเคมี ส่งผลให้เกิดการเคลื่อนที่ของอิเล็กตรอน (ปฏิกิริยาออกซิเดชัน) และขั้วไฟฟ้าบวกหรือขั้วแคโทดเป็น ขั้วที่เกิดปฏิกิริยาเคมีที่รับอิเล็กตรอน (ปฏิกิริยรีดักชัน) โดยอิเล็กตรอนที่ได้จากปฏิกิริยาออกซิเดชัน จะวิ่งผ่านลวดโลหะก่อให้เกิดการไหลของกระแสไฟฟ้าขึ้น โดยอิเล็กโทรไลต์จะต้องเป็นสารละลายที่ไม่ นำอิเล็กตรอน แต่มีหน้าที่ส่งผ่านไอออนที่จำเป็นต่อการเกิดปฏิกิริยาดังกล่าวได้อย่างต่อเนื่องจนกว่า แบตเตอรี่จะหมด

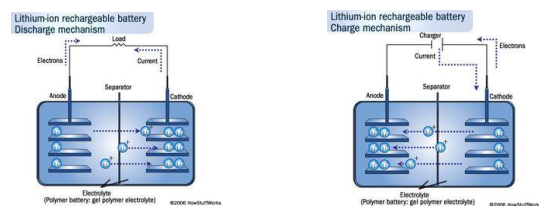
2.2 แบตเตอรี่ลิเทียมไอออน (Lion Battery)

แบตเตอรี่ลิเทียมไอออนเป็นแบตเตอรี่ที่นิยมใช้ในรถยนต์ไฟฟ้ามากที่สุดเนื่องจากมีสมบัติเป็นโลหะ ที่มี น้ำหนักเบาที่สุด

นอกจากนี้ลิเทียมยังมีศักย์ไฟฟ้าทางไฟฟ้าเคมีที่มาก รวมถึงการทำปฏิกิริยาทางเคมีที่มีความว่องไว จากสมบัติต่างๆ ของลิเทียม เมื่อนำมาทำเป็นแบตเตอรี่จึงได้แบตเตอรี่ที่เก็บพลังงานได้ สูงและจ่ายพลังงานได้มาก เนื่องจากโลหะลิเทียมนี้มีความไวต่อการทำปฏิกิริยาเคมี ในธรรมชาติจึงไม่ พบลิเทียมในรูปของโลหะบริสุทธิ์ แต่มักพบลิเทียมในรูปของสารประกอบ นอกจากนี้แล้วโลหะลิเทียม ยังทำปฏิกิริยารุนแรงกับน้ำ ดังนั้นเมื่อนำโลหะลิเทียมมาทำแบตเตอรี่ จึงทำให้แบตเตอรี่ชนิดนี้มีความเสี่ยงสูง โดยเฉพาะถ้าปิดผนึกแบตเตอรี่ไม่ดีพอ จะทำให้อากาศหรือความชื้นจากภายนอกสามารถ หลุดลอดเข้าไปในเซลล์ของแบตเตอรี่ได้ เมื่อโลหะลิเทียมสัมผัสกับน้ำหรือความชื้นในอากาศ จะทำให้เกิดการทำปฏิกิริยาเคมีกันจนอาจเกิดการระเบิดหรือเกิดการลุกเป็นไฟได้ ในยุคแรกๆเคยเกิดเหตุ ระเบิดขึ้น จนบริษัทผู้ผลิตต้องเลิกผลิตและหันไปพัฒนาแบตเตอรี่ที่ใช้ลิเทียมในรูปไอออนซึ่งมีความปลอดภัยมากกว่า

2.2.1 หลักการทำงานของแบตเตอรี่ลิเทียมไอออน

แบตเตอรี่ชนิดลิเทียมไอออนทำงานโดยอาศัยหลักการทางไฟฟ้าเคมี โดยขณะที่ ประจุ พลังงาน ไฟฟ้าที่เราประจุไฟเข้าไป ทำให้เกิดปฏิกิริยาเคมีภายในแบตเตอรี่ ซึ่งจะบังคับให้ลิเทียมไอออนไหลออกจากโครงสร้างของวัสดุที่ใช้ทำขั้วแคโทด แล้วไหลผ่านอิเล็กโทรไลต์ แล้วผ่าน Separator และเข้าไปสอดตัว (Intercalate) อยู่ในโครงสร้างของวัสดุที่ใช้ ทำขั้วแอโนด ดังแสดงในรูปที่ 2 ผลของปฏิกิริยานี้จะทำให้วัสดุที่ใช้ทำขั้วแคโทด เช่น LiMO_2 , LiM_2O_2 , LiMPO_4 และวัสดุที่ใช้ทำขั้วแอโนด เช่น C, Sn, Si อยู่ในสภาวะไม่เสถียร ส่วนในขณะที่ใช้งาน (คายประจุ) ปฏิกิริยาเคมีในแบตเตอรี่จะสามารถเกิดขึ้น ได้เอง (Spontaneous reaction) กล่าวคือ ลิเทียมไอออนที่ไหลออกจากโครงสร้างของวัสดุที่ใช้ทำ ขั้วแคโทด และไปแทรกตัวอยู่ที่ขั้วแอโนดนั้น จะไหลออกจากโครงสร้างของวัสดุที่ใช้ทำ ขั้วแอโนดและเข้าไปอยู่ในโครงสร้างของวัสดุที่ใช้ทำ ขั้วแคโทดตามเดิม ทำให้ระบบมีสภาพเสถียรอีกครั้ง พร้อมกับให้อิเล็กตรอนผ่านวงจรไฟฟ้า (โดยที่อิเล็กตรอนจะไหลผ่านโลหะ current collector) และให้พลังงานไฟฟ้าออกมา เมื่อใดก็ตามที่ลิเทียมไอออนไหลกลับไปที่ เดิมหมด ปฏิกิริยาที่จะสิ้นสุดหรือถ่านหมดนั่นเอง หากต้องการน แบตเตอรี่ไปใช้ใหม่ก็ต้อง ประจุไฟอีกครั้ง และเกิดเป็นเช่นนี้ต่อเนื่องไปจนกว่าแบตเตอรี่จะหมดสภาพและอายุการใช้งาน



รูปที่ 1 หลักการทำงานของแบตเตอรี่ลิเทียมไอออน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วิธีการประมาณสถานะประจุ

การประมาณสถานะประจุของการชาร์จแบตเตอรี่มีความสำคัญต่อการกำหนดช่วงที่เหลือของรถยนต์ที่ใช้ไฟฟ้าและระยะเวลาใช้งานของอุปกรณ์ที่ใช้พลังงานจากแบตเตอรี่ กล่าวคือความจุของแบตเตอรี่จะค่อยๆลดลงตามจำนวนการใช้งานที่เพิ่มขึ้น ซึ่งมีหลายสาเหตุที่ทำให้ความจุของแบตเตอรี่ลดลง เช่น อุณหภูมิที่มากเกินไป การชาร์จมากเกินไป การคายประจุมากเกินไป และการไม่ได้ใช้งานเป็นเวลานานหลังจากชาร์จเต็มแล้ว โดยปริมาณไฟฟ้าที่เหลืออยู่ในแบตเตอรี่สามารถประมาณได้จากกรหาสถานะประจุ (State-Of-Charge : SoC) ของแบตเตอรี่ โดยค่าสถานะประจุคือจำนวนประจุหรือพลังงานที่เหลืออยู่ภายในแบตเตอรี่ (Q) ซึ่งแสดงในรูปเปอร์เซ็นต์ต่อพิถักความจุของแบตเตอรี่ (Rated Capacity : C_R) ดังสมการด้านล่าง

$$\text{soc} = \frac{Q}{C_R} \times 100\%$$

ซึ่งค่าของ SoC จะแตกต่างกันไประหว่าง 0% ถึง 100% หาก SoC เป็น 100% แสดงว่าเซลล์ถูกชาร์จจนเต็มในขณะที่ SoC 0% แสดงว่าเซลล์ถูกคายประจุจนหมด ในทำนองเดียวกัน เมื่อเซลล์มีอายุมากขึ้น SoC สูงสุดจะเริ่มลดลง ซึ่งหมายความว่าสำหรับเซลล์ที่มีอายุมากขึ้น SoC 100% จะเทียบเท่ากับ SoC 75%-80% ของเซลล์ใหม่

2.4 โครงข่ายประสาทเทียม

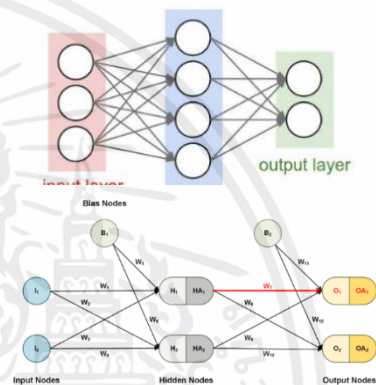
โครงข่ายประสาทเทียม (Artificial neural network : ANN) คือโมเดลทางคณิตศาสตร์ที่พัฒนาขึ้นเพื่อจำลองการทำงานของโครงข่ายประสาทในสมองมนุษย์ซึ่งเซลล์ประสาทเหล่านี้จะมีการเชื่อมโยงกับน้ำหนัก (weight) ที่เป็นสัญญาณที่ส่งผ่านจากเซลล์ประสาทหนึ่ง ไปยังอีกเซลล์ประสาทหนึ่ง ซึ่งเซลล์ประสาทแต่ละเซลล์จะรับตัวเลขหนึ่งตัวเพื่อส่งต่อไปยังปลายทาง แต่ผลลัพธ์ที่ได้มักจะมากกว่าหนึ่งจำนวน ซึ่งผลลัพธ์ที่ได้นี้จะส่งต่อไปยังเซลล์ประสาทอื่นๆ ที่เชื่อมต่ออยู่ ซึ่งการส่งสัญญาณออกจะสิ้นสุดเมื่อมีการรับข้อมูลจากเซลล์ประสาทอื่นๆ เข้ามาภายในเซลล์ รูปที่ 2 แสดงถึงการเชื่อมโยงของเซลล์ประสาทต่างๆ

โครงข่ายประสาทเทียมมีโครงสร้างดังนี้

1. ชั้นข้อมูลเข้า (Input Layer) ที่ชั้นนี้จะเป็นข้อมูลเข้า จำนวนของโหนดในชั้นนี้ขึ้นอยู่กับจำนวนของข้อมูลเข้า เช่น ถ้าข้อมูลของลูกค้านี้เป็นข้อมูลเข้าที่ประกอบด้วย อายุ เพศ จังหวัด ที่อาศัย รวมทั้งสิ้น 4 อย่าง ดังนั้นชั้นข้อมูลเข้าก็จะมี 4 โหนด ซึ่งอาจจะเรียกปัจจัยที่นำมาวิเคราะห์เหล่านี้ว่าคุณลักษณะ (Feature)
2. ชั้นซ่อน (Hidden Layer) เป็นชั้นที่อยู่ระหว่างกลาง มีหน้าที่รับข้อมูลจาก Layer ก่อนหน้า จะสังเกตว่า Hidden Layer สามารถมีจำนวนมากกว่า 1 ได้ และโดยพื้นฐาน ถ้าเรายังต้องการความแม่นยำที่มากขึ้นเราก็จะเพิ่มจำนวนชั้นของ Hidden Layer และจำนวน Neuron ให้มากขึ้น

3. ชั้นข้อมูลออก (Output Layer) เป็น Layer ที่อยู่ท้ายสุดรอรับค่าจาก Hidden Layer อันสุดท้าย โดยในชั้น Output นั้น แต่ละ Neuron จะมีค่าน้ำหนัก (Weight) ของคลาสอยู่ แต่ใน Hidden Layer มีอีกสองสิ่งที่ถูกซ่อนไว้คือ Bias และ Weight แสดงดังรูปที่ 2 โดยทุก ๆ Hidden Layer จะมี Bias เชื่อมต่ออยู่เพื่อให้ทุก ๆ การคำนวณเพื่อส่งต่อมีความเท่าเทียมกัน มีความหมายง่าย ๆ ก็คือทุก ๆ Neuron ใน Hidden Layer จะต้องมีการ Bias เข้าไปคำนวณเพื่อให้ Decision Boundary ไม่จำเป็นต้องผ่านจุด Origin ส่วน Weight จะเป็นน้ำหนักซึ่งมันจะส่งผลทุก ๆ Neuron มีค่า Output ที่ไม่เท่ากัน ทำให้แต่ละคลาสมีน้ำหนักไม่เท่ากัน เวลาคำนวณว่าเป็นคลาสไหนทำให้เราสามารถแยกแยะว่าข้อมูลนี้เป็นคลาสอะไรได้ด้วยการดูตัวเลขที่

Output

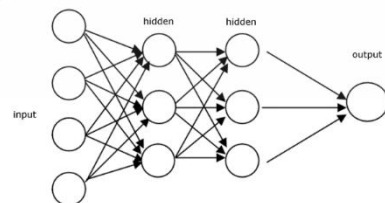


รูปที่ 2 โครงข่ายประสาทเทียม (Artificial neural network : ANN)

ประเภทของโครงสร้างของโครงข่ายประสาทเทียม

1. การคำนวณแบบไปข้างหน้า

โครงข่ายประสาทที่คำนวณแบบไปข้างหน้า (Feed-forward artificial neural network) ดังรูปที่ 3 สัญญาณจะถูกส่งไปทางเดียวจากอินพุตไปยังเอาต์พุต ไม่มีการย้อนกลับ ผลของข้อมูลในชั้นต่างๆ จะไม่มีการคำนวณซ้ำ ค่าที่ได้จากการคำนวณของอินพุตจะแสดงในเอาต์พุตทันที ส่วนใหญ่จะใช้สำหรับปัญหาการแยกประเภทอย่างง่าย

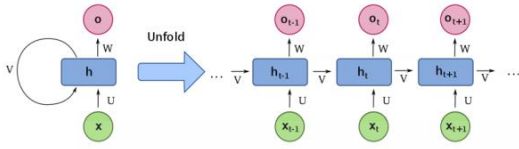


รูปที่ 3 โครงข่ายประสาทที่คำนวณแบบไปข้างหน้า

(feed-forward artificial neural network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent neural network) เป็นโครงสร้างเครือข่ายการเรียนรู้เชิงลึกที่ใช้ผลลัพธ์ที่ได้จากการคำนวณย้อนกลับมาใช้เป็นข้อมูลขาเข้าอีกครั้งซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง



รูปที่ 4 โครงข่ายประสาทแบบเกิดซ้ำ
(Recurrent neural network)

สำหรับความแตกต่างของ การคำนวณแบบไปข้างหน้า (Feed-forward Artificial Neural Network : FNN) กับ โครงข่ายประสาทแบบเกิดซ้ำ (Recurrent Neural Network : RNN) คือ FNN เป็นฟังก์ชันที่รับข้อมูลเข้าและประมวลผลออกมาเป็นข้อมูลออก โดย FNN จะมองข้อมูลเข้าแต่ละตัวแยกกันชัดเจน ไม่ขึ้นต่อกัน ข้อมูลเข้าที่รับเข้าไปจะเข้ามีการเรียงลำดับอย่างไรก็ได้ FNN จะไม่สนใจและจะประมวลผลให้ข้อมูลออกมาเหมือนเดิมซึ่ง FNN จะมีปัญหาเกี่ยวกับข้อมูลที่เป็นลำดับ เช่น ข้อความ (ลำดับตัวอักษร), เสียง (ลำดับแรงดันอากาศ), วิดีโอ (ลำดับของภาพและเสียง) และข้อมูลที่เป็นแบบ Time Series ต่าง ๆ เป็นต้น แต่ RNN ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับงานที่มีลำดับ โดยใช้หลักการนำสถานะภายในของโมเดล กลับมาเป็นข้อมูลขาเข้าใหม่คู่กับข้อมูลขาเข้าแบบปกติ เรียกว่า สถานะซ่อน (Hidden State) หรือสถานะภายใน (Internal State) ช่วยให้โมเดลรู้จำรูปแบบ (Pattern) ของลำดับข้อมูลเข้า (Input Sequence) ได้แสดงดังรูปที่ 4

ในแต่ละ โหนดของ RNN จะมีข้อมูลเข้าสองอย่าง ได้แก่ 1) ข้อมูลเข้า ณ โหนด นั้น ๆ และ 2) ผลลัพธ์ที่ได้จากการคำนวณในโหนดก่อนหน้า ซึ่งทั้งสองข้อมูลจะถูกนำมารวมเข้าด้วยกันและออกผลลัพธ์มาเป็นสองทาง คือ 1) ผลลัพธ์ที่ออก ณ โหนดนั้น ๆ และออกเพื่อไปเข้าเป็นข้อมูลขาเข้าในโหนดถัดไป ข้อดีของ RNN คือ มันมีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่จะเกิดขึ้นในอนาคต ซึ่งหมายถึง อะไรที่เพิ่งเกิดขึ้นในอดีตย่อมส่งผลต่อเหตุการณ์ที่จะเกิดขึ้นในอนาคตด้วย แม้ RNN จะมีข้อดีในการทำงานของข้อมูลที่มีความต่อเนื่อง แต่ข้อเสียของ RNN ก็คือมันสามารถดูย้อนกลับได้แค่เพียงในช่วงระยะเวลาสั้น ๆ เท่านั้น ซึ่งปัญหาหลัก ๆ ของ RNN เกิดมาจากค่าเกรเดียนที่เริ่มน้อยลงในข้อมูลที่มีความยาวมากขึ้น จนแทบจะไม่สามารถเห็นความเปลี่ยนแปลงของเกรเดียนได้เลย จึงมีการเสนอใช้วิธีที่มีประสิทธิภาพทางการคำนวณมากกว่าสามารถจัดการข้อมูลในหน่วยความจำได้เป็นเวลานานกว่า และช่วยให้ผู้ใช้สามารถฝึกโมเดลโดยใช้ลำดับที่มีขึ้นตอนเวลาหลายร้อยขั้นตอนได้ นั่นก็คือ Long Short-Term Memory (LSTM) ซึ่งเป็น Recurrent Neural

Network (RNN) ชนิดหนึ่ง ใน RNN เอาต์พุตจากขั้นตอนสุดท้ายจะถูกป้อนเป็นอินพุตในขั้นตอนปัจจุบัน ไอเดียหลักของ RNN ก็เพื่อจะใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) เช่น Video (Sequence of images) หรือ Text (Sequence of words) ยกตัวอย่างเปรียบเทียบกับ การอ่านหนังสือซึ่งก็คือลำดับของคำที่ต่อกัน (Sequence of words) อ่านทีละคำจากซ้ายไปขวา (สำหรับภาษาไทยหรือภาษาอังกฤษ) การที่จะสามารถรู้เรื่องได้ว่าประโยคที่กำลังอ่านนั้นเกี่ยวกับอะไร ก็ต้องเอาเรื่องราวจากสิ่งที่อ่านผ่านไปแล้ว (Hidden state) มาผสมกับคำที่เพิ่งอ่านไป (Input data) ทำให้สามารถเข้าใจความหมายใน

ส่วนที่กำลังอ่านได้ ซึ่ง RNN ก็ใช้หลักการเดียวกันคือการปรับรูปแบบของ Neural network เดิม เพื่อให้สามารถเอา State ก่อนหน้า มาบวกกับ Input data ตัวใหม่ที่เข้ามา เพื่อทำความเข้าใจอะไรสักอย่างต่อไปเรื่อยๆ การใช้การเกิดซ้ำสามารถปรับปรุงความสามารถของเครือข่ายในการเข้ารหัสข้อมูลเชิงบริบทให้กับแบบจำลอง ทำให้เหมาะสมในการแก้ปัญหาที่ข้อมูลในอดีตเกี่ยวข้องกับการประมวลหรือคาดการณ์ผลลัพธ์ เช่น การพยากรณ์ตลาดหุ้น SoC ของแบตเตอรี่ และสภาพการขับขี่

3. การออกแบบและสร้างโปรแกรม ในการประมาณสถานะประจุของแบตเตอรี่

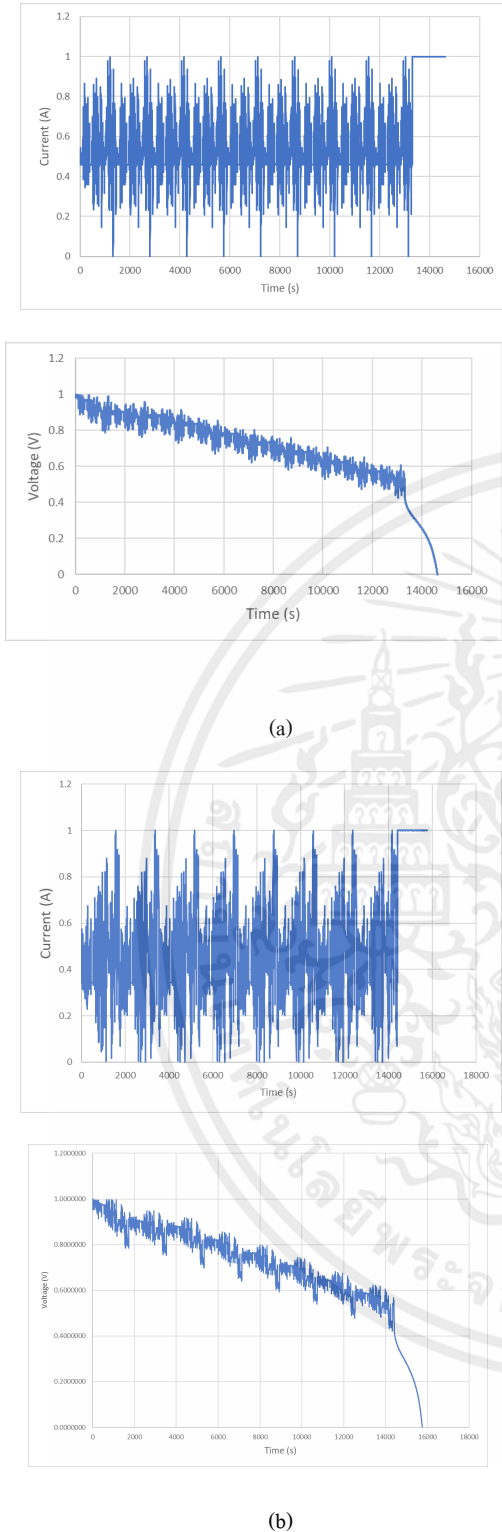
การออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียม โดยการใช้โปรแกรม MATLAB จะกำหนดสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้ว กระแสไฟฟ้าและอุณหภูมิ และกำหนดสัญญาณขาออกคือ สถานะประจุ มีการกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับ

3.1 Input Profile

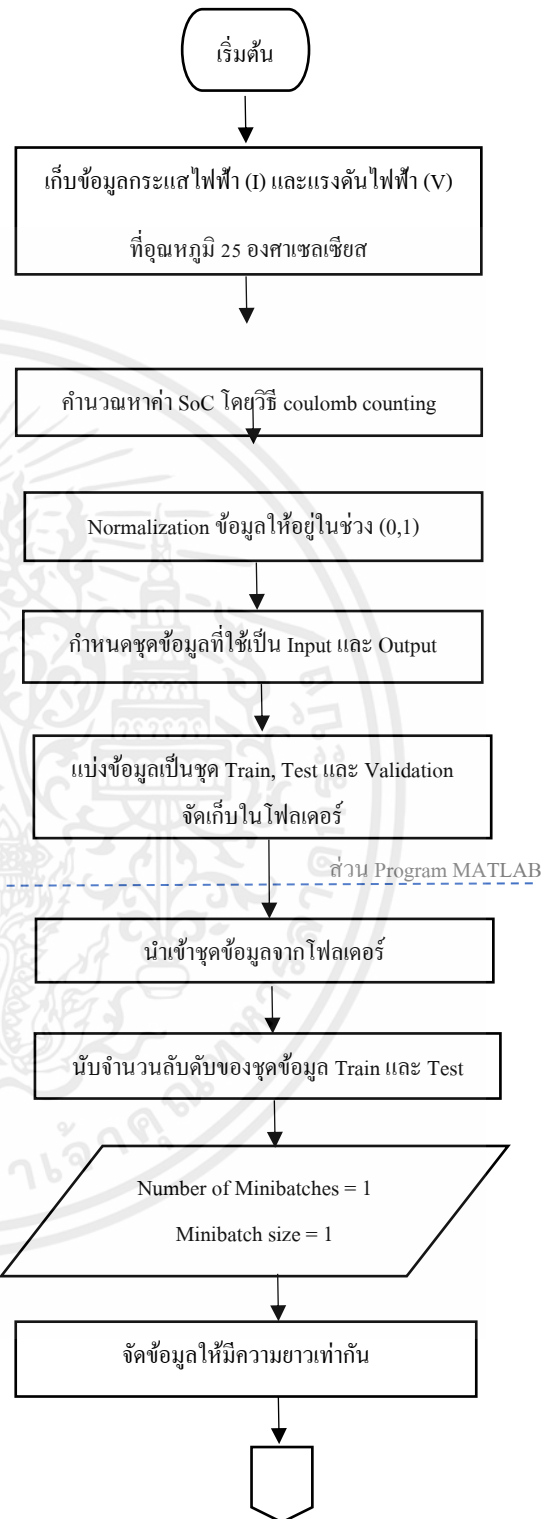
โปรไฟล์ที่ใช้เป็นอินพุตสำหรับการประมาณสถานะประจุคือ โปรไฟล์ที่จำลองการวิ่งของรถยนต์ ไฟฟ้าในห้องทดสอบรถยนต์ไฟฟ้า โดยใช้ EV Reference Applications Model และเลือกใช้ Driving Cycle Source เป็น WLTP3 Class2 และ WLTP Class3 ซึ่งอินพุตที่สำคัญในการประมาณสถานะ ประจุโดย FNN, LSTM และ LSTM คือ แรงดันไฟฟ้าระหว่างขั้วและกระแสไฟฟ้า รูปที่ 9 (a) และ (b) แสดงกราฟแรงดันไฟฟ้าระหว่างขั้วและกระแสไฟฟ้าของโปรไฟล์ WLTP Class2 และ WLTP Class3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

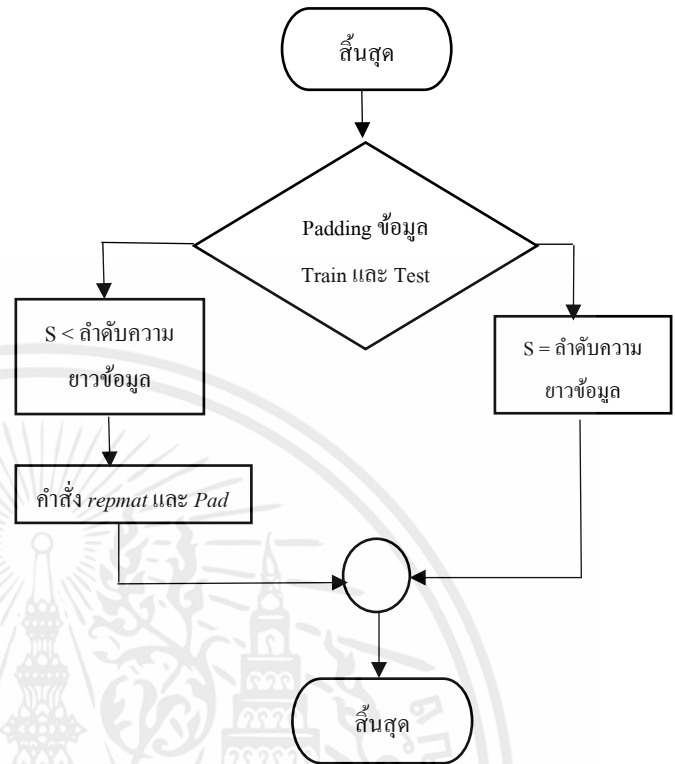
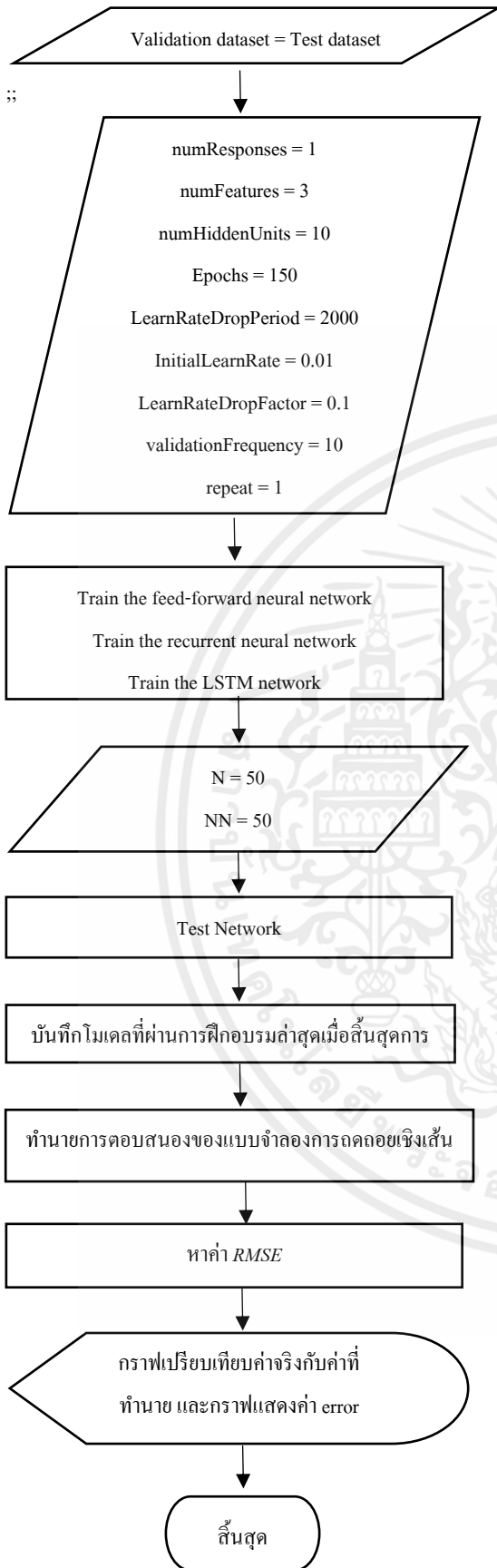
3.2 แผนผังในการประมาณสถานะประจุของแบตเตอรี่



รูปที่ 5 กราฟกระแสไฟฟ้าและแรงดันไฟฟ้าระหว่างขับของโปรไฟล์อินพุต (a) โปรไฟล์ WLTP Class2, (b) โปรไฟล์ WLTP Class3



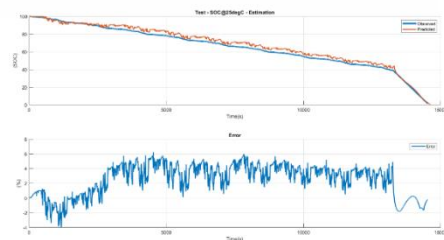
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4. ผลการทดลอง

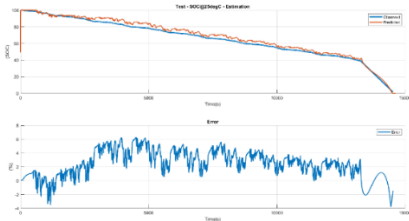
แสดงผลของการดำเนินงานที่ได้จากการออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน มีการทดสอบโครงข่ายประสาทเทียมแบบ FNN RNN และ LSTM โดยทำการใช้ชุดข้อมูลเดียวกันและสลับชุดข้อมูลทั้งหมด 4 กรณี ได้แก่ กรณีที่ 1 ใช้ชุดข้อมูล Train เป็น WLTP class 3 และ Test เป็น WLTP class 2 กรณีที่ 2 ใช้ชุดข้อมูล Train เป็น WLTP class 2 และ Test เป็น WLTP class 3 กรณีที่ 3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP class 2 และ กรณีที่ 4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP class 3 ได้ผลการทดลองดังนี้

4.1 ใช้ชุดข้อมูล Train เป็น WLTP class 3 และ Test เป็น WLTP class 2

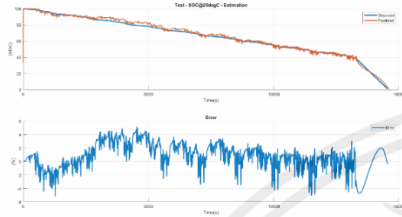


(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b)

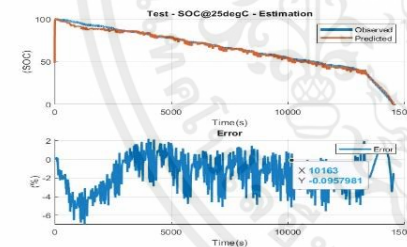


(c)

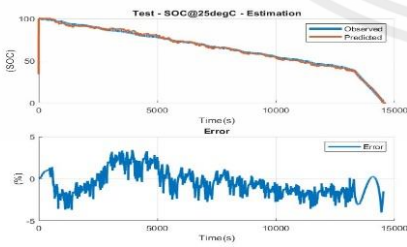
รูปที่ 6 กราฟการทำนายค่า SoC และ เปอร์เซ็นต์ Error โดยใช้ชุดข้อมูล Train เป็น WLTP class 3 และ Test เป็น WLTP class 2

- (a) Feed-forward neural network
- (b) Recurrent neural network
- (c) Long short-term memory neural network

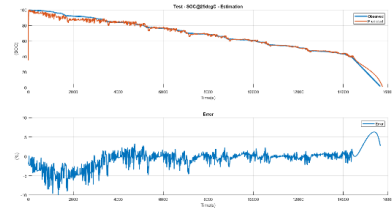
4.2 ใช้ชุดข้อมูล Train เป็น WLTP class 2 และ Test เป็น WLTP class 3



(a)



(b)

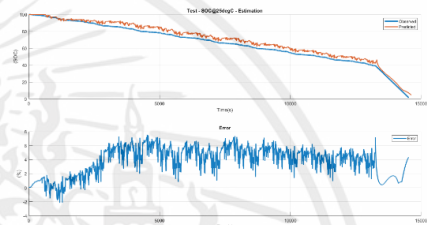


(c)

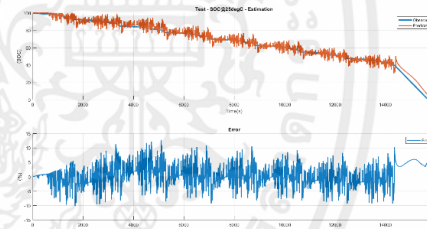
รูปที่ 7 กราฟการทำนายค่า SoC และ เปอร์เซ็นต์ Error โดยใช้ชุดข้อมูล Train เป็น WLTP class 2 และ Test เป็น WLTP class 3

- (a) Feed-forward neural network
- (b) Recurrent neural network
- (c) Long short-term memory neural network

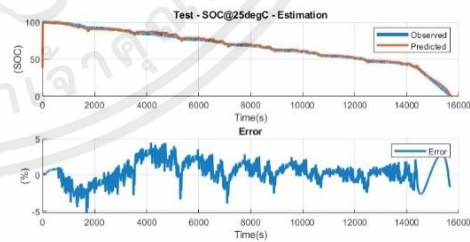
4.3 ใช้ชุดข้อมูล Train และ Test เป็น WLTP class 2



(a)



(b)



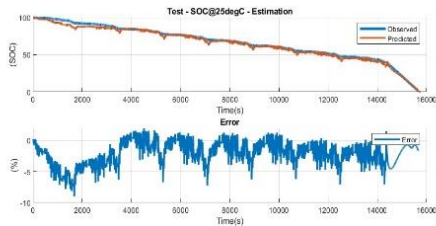
(c)

รูปที่ 8 กราฟการทำนายค่า SoC และ เปอร์เซ็นต์ Error โดยใช้ชุดข้อมูล Train และ Test เป็น WLTP class 2

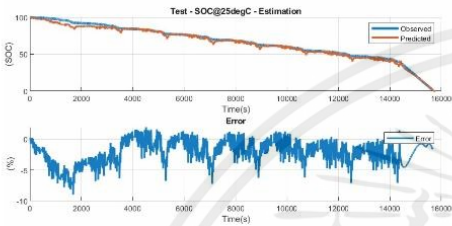
- (a) Feed-forward neural network
- (b) Recurrent neural network
- (c) Long short-term memory neural network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

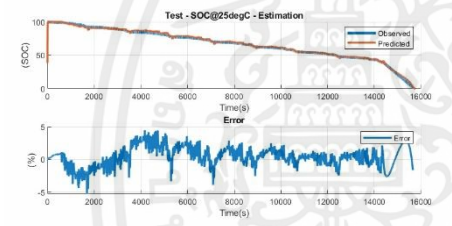
4.4 ใช้ชุดข้อมูล Train และ Test เป็น WLTP class 3



(a)



(b)



(c)

รูปที่ 3 กราฟการทำนายค่า SoC และ เปอร์เซ็นต์ Error โดยใช้ชุดข้อมูล

Train และ Test เป็น WLTP class 3

(a) Feed-forward neural network, (b) Recurrent neural network

(c) Long short-term memory neural network

ตารางที่ 1 สรุปค่า RMSE ของแต่ละกรณี

ประเภท โครงข่าย ประสาทเทียม	กรณีที่ 1	กรณีที่ 2	กรณีที่ 3	กรณีที่ 4
FNN (%)	3.27	3.03	4.29	2.69
RNN (%)	3.12	3.11	2.18	1.67
LSTM (%)	2.02	2.39	1.55	1.48

จากการทดสอบโครงข่ายประสาทเทียมแบบ FNN RNN และ LSTM โดยทำการใช้ชุดข้อมูลเดียวกันและสลับชุดข้อมูลทั้งหมด 4 กรณี พบว่าโครงข่ายประสาทเทียมแบบ LSTM ให้ค่า RMSE น้อยที่สุดในทุก

กรณี ซึ่งแสดงให้เห็นถึงความแม่นยำในการประมาณสถานะประจุแบตเตอรี่ชนิดลิเทียมไอออนโดยใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM)

5. สรุปผลการทดลอง

ปัญหานี้พบกันบ่อยครั้งเกิดขึ้นโดยมีวัตถุประสงค์เพื่อการออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออน เนื่องจากปัจจุบันแบตเตอรี่ลิเทียมไอออนถูกเลือกใช้เป็นส่วนมากในรถยนต์ไฟฟ้า ซึ่งการประมาณสถานะประจุสามารถทำได้หลายวิธี ตัวอย่างเช่น วิธีการนับคูลอมบ์ (Coulomb Counting) แต่ด้วยวิธีนี้จะเกิดข้อผิดพลาดจากค่าออฟเซตและการคายประจุด้วยตัวเอง วิธี Kalman-Filter มีความยุ่งยากและซับซ้อน และต้องใช้ชุดสมการที่ซับซ้อนมากเพื่อกระตุ้นผลลัพธ์คณะผู้จัดทำจึงเสนอการประมาณค่าสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการใช้โครงข่ายประสาทเทียมซึ่งเป็นวิธีที่สามารถประมาณสถานะประจุได้อย่างแม่นยำและไม่มีความยุ่งยากเปรียบเสมือนกับ black box ที่ต้องการแค่อินพุตและเอาต์พุตป้อนเข้าสู่ระบบ โดยไม่สนใจกระบวนการทำงานภายในว่าทำงานอย่างไร

จากการทดลองเพื่อออกแบบการประมาณสถานะประจุของแบตเตอรี่ชนิดลิเทียมไอออนโดยการทดลองนี้จะเป็นการดำเนินการผ่านโปรแกรม MATLAB โดยกำหนดสัญญาณขาเข้าที่ป้อนเข้าสู่โครงข่าย 3 ค่า ได้แก่ แรงดันระหว่างขั้ว กระแสไฟฟ้า และอุณหภูมิ และกำหนดสัญญาณขาออก คือ State of Charge (SoC) และมีการกำหนด Training set และ Testing set เป็น WLTP Class 3 และ WLTP Class 2 ตามลำดับ โดยมีการทดสอบทั้งหมด 4 กรณี

เมื่อพิจารณาและเปรียบเทียบค่า Root Mean Squared Error (RMSE) พบว่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุดและรองลงมาคือ SoC ที่ได้จากโครงข่าย RNN และ FNN ตามลำดับ เนื่องจากข้อมูลที่ใช้ในการประมาณสถานะประจุแบตเตอรี่ลิเทียมไอออนเป็นชุดข้อมูลอนุกรมเวลา กล่าวคือข้อมูลเวลาในปัจจุบันกับข้อมูลเวลา ก่อนหน้ามีความสัมพันธ์กัน ซึ่งข้อมูลที่มีลักษณะเป็นลำดับ เช่น Text (Sequence of words) ยกตัวอย่างเปรียบเทียบกับกรอ่านหนังสือซึ่งก็คือลำดับของคำที่ต่อกันอ่านทีละคำจากซ้ายไปขวา (สำหรับภาษาไทยหรือภาษาอังกฤษ) การที่จะสามารถรู้เรื่องได้ว่าประโยคที่กำลังอ่านนั้นเกี่ยวกับอะไร ก็ต้องเอาเรื่องราวจากสิ่งที่อ่านผ่านไป แล้ว มาผสมกับคำที่เพิ่งอ่านไปทำให้สามารถเข้าใจความหมายในส่วนที่กำลังอ่านได้ ซึ่งการประมาณโดยใช้ Feed-forward Neural Network (FNN) จะมองแค่ข้อมูลในปัจจุบันจึงมีปัญหาเกี่ยวกับข้อมูลที่เป็นลำดับ แต่ Recurrent Neural Network (RNN) ถูกออกแบบมาเพื่อแก้ปัญหาสำหรับข้อมูลที่เป็นลำดับ โดย RNN มีการใช้ข้อมูลก่อนหน้าในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคตซึ่งหมายถึงข้อมูลในปัจจุบันเป็นผลมาจากข้อมูลในอดีต ถึงแม้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RNN จะสามารถใช้ข้อมูลก่อนหน้ามาคาดการณ์สถานะประจุแบตเตอรี่ได้ แต่ก็ยังเป็นข้อมูลก่อนหน้าที่เป็นช่วงระยะเวลาสั้น ๆ เท่านั้น จึงมีการใช้โครงข่ายประสาทเทียมชนิด Long Short-Term Memory (LSTM) ซึ่งสามารถเก็บข้อมูลที่เก่ากว่าไว้ในหน่วยความจำได้ด้วยสาเหตุนี้จึงทำให้ค่า SoC ที่ได้จากโครงข่าย LSTM มีความแม่นยำมากที่สุด และรองลงมาคือค่า SoC จาก โครงข่าย RNN ที่มี Short term memory และอันดับสุดท้ายคือค่า SoC จาก โครงข่าย FNN

กิตติกรรมประกาศ

ปริญญาานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความเมตตาและความช่วยเหลือจากท่าน อาจารย์ที่ปรึกษาและอาจารย์ที่ปรึกษาร่วมกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุวัฒน์ กิตติรัตนสังจา และ ดร.สมภพ ผลไม้ รวมไปถึงอาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่าน ที่ได้กรุณาเสียสละเวลาอันมีค่า ในการให้คำแนะนำและการชี้แนะแนวทางในการทำให้ปริญญาานิพนธ์นี้เป็นไปในขอบเขตที่คณะผู้จัดทำหวังไว้ ตลอดจนการตรวจสอบแก้ไขความ เรียบร้อยของปริญญาานิพนธ์เล่มนี้เป็นอย่างดี

สุดท้ายนี้ คณะผู้จัดทำขอขอบพระคุณบิดามารดาและครอบครัว ผู้ซึ่งเปิดโอกาสให้ได้รับ การศึกษา ตลอดจนให้การสนับสนุนช่วยเหลือและให้กำลังใจระหว่างการดำเนิน รวมทั้งเพื่อน ๆ ที่ ช่วยเหลือทางความคิดเห็น ช่วยแนะแนวทางที่เป็นประโยชน์ต่อปริญญาานิพนธ์เล่มนี้ จึงขอขอบคุณมา ณ ที่นี้ด้วย

เอกสารอ้างอิง

- [1] M.Einhorn, Member, IEEE, F.V. Conte, C. Kral, Senior Member, IEEE, and J. Fleig, A Method for Online Capacity Estimation of Lithium Ion Battery Cells Using the State of Charge and the Transferred Charge.
- [2] Estimation of Lithium Ion Battery Cells Using the State of Charge and the Transferred Charge. Lithium-Ion Battery Chemistries from Renewable Energy Storage to Automotive and Back-up Power Applications – An Overview

ประวัติผู้เขียนบทความ



นางสาวกนกรรณ ตั้งกาญจนกิจ
สำเร็จการศึกษาระดับปริญญาตรี
ภาควิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง



นางสาวกฤษณา พันธุ์แดง
สำเร็จการศึกษาระดับปริญญาตรี
ภาควิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง



นางสาวกาญจนนาพร พูลเกิด
สำเร็จการศึกษาระดับปริญญาตรี
ภาควิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง



นางสาวพิมพ์ทอง กวาลิบบสาม
สำเร็จการศึกษาระดับปริญญาตรี
ภาควิชาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน



ชื่อ-นามสกุล กนกวรรณ ตั้งกาญจนกิจ

วัน/เดือน/ปี 8 พฤษภาคม พ.ศ. 2543

ประวัติการศึกษา

ปีการศึกษา 2561 สำเร็จระดับมัธยมศึกษาตอนปลาย โรงเรียนธัญรัตน์ ปทุมธานี

ปีการศึกษา 2565 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตภาควิชา
วิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง

ฝึกงานฤดูร้อน

บริษัท สยามพาวเวอร์ จำกัด

ระหว่างวันที่ 1 มิถุนายน ถึง 29 กรกฎาคม พ.ศ. 2565



ชื่อ-นามสกุล กฤษณา พันธุ์แดง

วัน/เดือน/ปี 27 มีนาคม พ.ศ. 2543

ประวัติการศึกษา

ปีการศึกษา 2561 สำเร็จระดับมัธยมศึกษาตอนปลาย โรงเรียนบางพลีราษฎร์บำรุง
สมุทรปราการ

ปีการศึกษา 2565 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตภาควิชา
วิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง

ฝึกงานฤดูร้อน

บริษัท สยามพาวเวอร์ จำกัด

ระหว่างวันที่ 1 มิถุนายน ถึง 29 กรกฎาคม พ.ศ. 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ชื่อ-นามสกุล กาญจนนาพร พูลเกิด

วัน/เดือน/ปี 24 ธันวาคม พ.ศ. 2543

ประวัติการศึกษา

ปีการศึกษา 2561 สำเร็จระดับมัธยมศึกษาตอนปลาย โรงเรียนรัชชวิทยา
ประจวบคีรีขันธ์

ปีการศึกษา 2565 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตภาควิชา
วิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง

ฝึกงานฤดูร้อน บริษัท สยามพาวเวอร์ จำกัด
ระหว่างวันที่ 1 มิถุนายน ถึง 29 กรกฎาคม พ.ศ. 2565



ชื่อ-นามสกุล พิมพ์ทอง กวาวสิบสาม

วัน/เดือน/ปี 4 เมษายน พ.ศ. 2542

ประวัติการศึกษา

ปีการศึกษา 2561 สำเร็จระดับมัธยมศึกษาตอนปลาย โรงเรียนสา น่าน

ปีการศึกษา 2565 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตภาควิชา
วิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง

ฝึกงานฤดูร้อน บริษัท ทีทีซีแอล จำกัด (มหาชน)

ระหว่างวันที่ 1 มิถุนายน ถึง 29 กรกฎาคม พ.ศ. 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้