

ประสิทธิภาพของแบบจำลองแบบผสมของการเรียนรู้เชิงลึกสำหรับการพยากรณ์ราคาหุ้น



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาสถิติและวิทยาการข้อมูล ภาควิชาสถิติ
คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2567

PERFORMANCE OF HYBRID DEEP LEARNING MODELS FOR STOCK PRICE FORECASTING



A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Statistics and Data Science
Department of Statistics
Faculty of Commerce and Accountancy
Chulalongkorn University
Academic Year 2024

หัวข้อวิทยานิพนธ์

ประสิทธิภาพของแบบจำลองแบบสมมติการเรียนรู้เชิงลึก
สำหรับการพยากรณ์ราคาหุ้น

โดย

นายกิตติคุณ ทัดประดิษฐ์

สาขาวิชา

สถิติและวิทยาการข้อมูล

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร.วิชรา พึงพาพงศ์

คณะกรรมการบัญชี จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

คณะกรรมการนิชยศาสตร์และการบัญชี

(รองศาสตราจารย์ ดร.ธารทัศน์ โมกขมรรคกุล)

คณะกรรมการสอบบวิทยานิพนธ์

ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.นพ. กุลวนิช)

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.วิชรา พึงพาพงศ์)

กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ภูริพันธุ์ รุจิ哈尔)

กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร. ทันตแพทย์หญิงนัจฉลี ศรีเมฆภรณ์จน)

กิตติคุณ ทัศประดิษฐ์ : ประสิทธิภาพของแบบจำลองแบบผสมของการเรียนรู้เชิงลึกสำหรับการพยากรณ์ราคาหุ้น. (PERFORMANCE OF HYBRID DEEP LEARNING MODELS FOR STOCK PRICE FORECASTING) อ.ที่ปรึกษาหลัก : รศ. ดร.วิชูรา พึงพา พงศ์

งานวิจัยนี้มีวัตถุประสงค์เพื่อวิเคราะห์และเปรียบเทียบประสิทธิภาพและระยะเวลาที่ใช้ของแบบจำลองการเรียนรู้เชิงลึกแบบผสม (Hybrid Deep Learning Models) ได้แก่ RNN, LSTM และ GRU ในรูปแบบการใช้โครงสร้างแบบผสม (Stacked layer) และเครือข่ายประสาทเทียมแบบซ้อนกัน (Cascaded neural network) ในการพยากรณ์ราคาปิดหุ้น รวมถึงศึกษาผลกระทบของ การสลับลำดับของแบบจำลองภายใน Hybrid model เพื่อพิจารณาความแตกต่างของ ประสิทธิภาพการพยากรณ์ ทั้งในระยะสั้น 7 วัน และ ระยะยาว 30 วัน โดยมีการใช้ข้อมูลราคาหุ้น ทั้งหมด 5 อุตสาหกรรม เลือกกลุ่มอุตสาหกรรมละ 3 หุ้นตามระดับความผันผวนของราคาหุ้นเมื่อ เทียบกับตลาด (Beta) รวมทั้งหมด 15 ชุดข้อมูล ผลการศึกษาพบว่า ในภาพรวมการพยากรณ์ระยะ สั้น 7 วันและระยะยาว 30 วัน การสร้างแบบจำลองผสมด้วยวิธี Cascaded neural network มี ประสิทธิภาพที่ไม่แตกต่างกับวิธี Stacked layers อย่างมีนัยสำคัญ แต่ถ้าหากพิจารณาด้วย ระยะเวลาที่ใช้ในการสร้างแบบจำลองจะพบว่า การสร้างแบบจำลองผสมด้วยวิธี Stacked layers ใช้เวลาสร้างแบบจำลองน้อยกว่าวิธี Cascaded neural network อย่างมีนัยสำคัญ ซึ่งจะช่วย ประหยัดทรัพยากรในการสร้างแบบจำลองในการพยากรณ์เป็นอย่างมาก นอกจากนี้การสลับลำดับ ของแบบจำลองที่ใช้ในการผสมแบบจำลอง ด้วยวิธี Cascaded neural network และ Stacked layers ใน การพยากรณ์ระยะสั้น 7 วันและระยะยาว 30 วัน ไม่ส่งผลต่อประสิทธิภาพของ Hybrid Deep Learning Models อย่างมีนัยสำคัญ จึงสามารถสรุปได้ว่าการสร้าง Hybrid Deep Learning Models ด้วยวิธี Stacked layers ให้ประสิทธิภาพและความคุ้มค่าที่มากกว่าการผสม ด้วยวิธี Cascaded neural network ทั้งในระยะสั้นและระยะยาว

สาขาวิชา สถิติและวิทยาการข้อมูล
ปีการศึกษา 2567

ลายมือชื่อนิสิต
ลายมือชื่อ อ.ที่ปรึกษาหลัก

6680025026 : MAJOR STATISTICS AND DATA SCIENCE

KEYWORD: HYBRID DEEP LEARNING MODEL, TIME SERIES ANALYSIS, RNN, LSTM, GRU

Kittikun Tadpradit : PERFORMANCE OF HYBRID DEEP LEARNING MODELS FOR STOCK PRICE FORECASTING. Advisor: Assoc. Prof. Vitara Pungpapong, Ph.D.

This research examines and compares the performance and training time of Hybrid Deep Learning Models, including RNN, LSTM, and GRU. These models are implemented using two architectures: Stacked Layers and Cascaded Neural Networks, for forecasting stock closing prices. Including the study of the impact of changing the sequencing of models within the Hybrid model to assess differences in forecasting performance for both the short-term (7 days) and the long-term (30 days). The dataset consists of stock prices from five industries, selecting three stocks from volatility levels (Beta), resulting in a total of 15 datasets. The findings indicate that Stacked Layers and Cascaded Neural Networks perform similarly in both short-term and long-term forecasts, with no significant differences in accuracy. However, in terms of training time, Stacked Layers require significantly less time than Cascaded Neural Networks, making them more efficient and resource-saving. Additionally, rearranging the order of models in hybrid architecture does not significantly affect forecasting performance. Overall, the results suggest that Stacked Layers are the better approach for building Hybrid Deep Learning Models, as they provide similar accuracy while being more efficient in terms of computation and resource usage, making them preferable for both short-term and long-term stock price forecasting.

Field of Study: Statistics and Data Science Student's Signature

Academic Year: 2024 Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์และผ่านไปได้ด้วยดี เพราะได้รับความอนุเคราะห์ ความช่วยเหลือและคำแนะนำจากบุคคลหลายท่าน ผู้วิจัยจึงจัดทำกิตติกรรมประกาศ เพื่อแสดงคำขอบคุณทุกท่านซึ่งเป็นส่วนสำคัญที่ช่วยให้วิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์

ขอขอบคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร.วิชรา พึงพาพงศ์ ผู้ที่ให้การช่วยเหลือทางการวิจัยอย่างต่อเนื่อง ท่านได้ให้คำแนะนำที่มีคุณค่า และบ่งชี้ถึงปัญหาต่าง ๆ พร้อมทั้งสละเวลาให้คำปรึกษาแนวทางการปรับปรุงและพัฒนาแก้ไขวิทยานิพนธ์อย่างเอาใจใส่มาตลอดตั้งแต่เริ่มต้นโครงการจนกระทั่งสำเร็จลุล่วง ลำดับถัดมา ขอขอบคุณประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. นพ. กุลวนิช กรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. ภริพันธ์ รุจิชจร และ กรรมการภายนอกมหาวิทยาลัย รองศาสตราจารย์ ดร. ทันตแพทย์หญิง นัจลี ศรีเมืองกานุจ เป็นอย่างสูงที่ได้ให้ข้อเสนอแนะและคำแนะนำที่เป็นประโยชน์ในการปรับปรุงและพัฒนางานวิจัยนี้ ขอขอบคุณเพื่อนนิสิตทุกคนที่ให้การสนับสนุนและกำลังใจตลอดระยะเวลาที่ผ่านมา การแบ่งปันความคิดเห็นและประสบการณ์ต่าง ๆ เป็นแรงผลักดันให้มีความมุ่งมั่นในการทำงานวิจัยนี้

สุดท้ายนี้ ขอขอบคุณครอบครัวที่ให้การสนับสนุนและความเข้าใจตลอดระยะเวลาการศึกษาและการวิจัย ความรักและกำลังใจจากท่านเป็นแรงบันดาลใจที่สำคัญในการดำเนินงานวิจัยนี้ จันสำเร็จ



กิตติคุณ ทัศประดิษฐ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉบับ
สารบัญตาราง	ช
สารบัญรูปภาพ	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 บริบทของปัญหา.....	4
1.3 วัตถุประสงค์ของการวิจัย	4
1.4 สมมติฐานการวิจัย	4
1.5 ขอบเขตของการวิจัย	5
1.6 เส้นทางที่สำคัญของการวิจัย.....	6
1.7 ระยะเวลาในการดำเนินการวิจัย	7
1.8 ประโยชน์ที่คาดว่าจะได้รับ	7
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	8
2.1 แนวคิดและทฤษฎี.....	8
2.1.1 การวิเคราะห์อนุกรมเวลา (Time series analysis).....	8
2.1.2 เครือข่ายประสาทเทียม (Artificial Neural Networks - ANN).....	9
2.1.3 เครือข่ายประสาทเทียมแบบย้อนกลับ (Recurrent Neural Networks: RNN)	15
2.1.4 เครือข่ายหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM) ...	17

2.1.5 เครือข่ายประตุกกลับ (Gated Recurrent Units - GRU)	19
2.1.6 แบบจำลองการเรียนรู้เชิงลึกแบบผสม (Hybrid Deep Learning Model).....	22
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง	25
บทที่ 3 วิธีการดำเนินการวิจัย	31
3.1 ศึกษาทฤษฎีและกรอบแนวคิดที่เกี่ยวข้อง	31
3.2 การเก็บรวบรวมข้อมูล	31
3.3 การจัดเตรียมข้อมูลและสร้างแบบจำลองสำหรับการพยากรณ์	37
3.3.1 การจัดเตรียมข้อมูล	37
3.3.2 สร้างแบบจำลองสำหรับการพยากรณ์.....	40
3.4 การวัดประสิทธิภาพการพยากรณ์	45
3.5 สรุปผลและวิเคราะห์ผลงานวิจัย	46
3.6 แผนผังการดำเนินงานวิจัย	47
บทที่ 4 ผลการวิจัย.....	49
4.1 ผลการวิจัยของหุ้นที่มีความผันผวนต่ำ (Low Beta)	50
4.2 ผลการวิจัยของหุ้นที่มีความผันผวนปานกลาง (Medium beta)	51
4.3 ผลการวิจัยของหุ้นที่มีความผันผวนสูง (High beta).....	52
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	63
5.1 สรุปผลการวิจัย	63
5.2 ข้อเสนอแนะ	71
บรรณานุกรม.....	72
ภาคผนวก.....	74
ประวัติผู้เขียน	119

สารบัญตาราง

หน้า

ตารางที่ 1 แสดงหุ้นในกลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และ กลุ่มเทคโนโลยี แบ่งตามระดับความผันผวนของหุ้น (Beta).....	5
ตารางที่ 2 แสดงคำจำกัดความของแบบจำลองการเรียนรู้เชิงลึก	6
ตารางที่ 3 แสดงขั้นตอนและระยะเวลาในการดำเนินการวิจัย	7
ตารางที่ 4 แบบจำลองและวิธีการทดสอบที่มีประสิทธิภาพที่สุดในการพยากรณ์ระยะสั้น 7 วัน ของแต่ละ หุ้นโดยวัดจาก MAPE และ RMSE.....	53
ตารางที่ 5 แบบจำลองและวิธีการทดสอบที่มีประสิทธิภาพที่สุดในการพยากรณ์ระยะยาว 30 วัน ของแต่ ละหุ้นโดยวัดจาก MAPE และ RMSE	55
ตารางที่ 6 Hybrid Deep Learning Model ที่ดีที่สุดสำหรับพยากรณ์ระยะสั้น 7 วันโดยวัดจาก ระยะเวลาในการฝึกฝนแบบจำลอง	58
ตารางที่ 7 Hybrid Deep Learning Model ที่ดีที่สุดสำหรับพยากรณ์ระยะยาว 30 วัน โดยวัดจาก ระยะเวลาในการฝึกฝนแบบจำลอง	60
ตารางที่ 8 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ ของการสลับแบบจำลองที่ใช้การทดสอบแบบ Stacked layer	63
ตารางที่ 9 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ ของการสลับแบบจำลองที่ใช้การทดสอบแบบ Cascaded neural network	63
ตารางที่ 10 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ..	64
ตารางที่ 11 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ใน กลุ่มความผันผวนสูง	65
ตารางที่ 12 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ใน กลุ่มความผันผวนปานกลาง	66

ตารางที่ 13 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ในกลุ่มความผันผวนต่ำ.....	66
ตารางที่ 14 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลอง.....	67
ตารางที่ 15 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลอง.....	67
ตารางที่ 16 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนสูง	68
ตารางที่ 17 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนสูง	68
ตารางที่ 18 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวน ปานกลาง	68
ตารางที่ 19 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวน ปานกลาง	69
ตารางที่ 20 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนต่ำ	69
ตารางที่ 21 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนต่ำ	69

สารบัญรูปภาพ

หน้า

ภาพที่ 1 กระบวนการทำงานภายในเซลล์ประสาท	9
ภาพที่ 2 โครงสร้างการทำงานของเครือข่ายประสาทเทียม	11
ภาพที่ 3 โครงสร้างของเครือข่ายระบบประสาทแบบย้อนกลับ (RNN).....	15
ภาพที่ 4 ความแตกต่างระหว่าง Feedforward Neural Networks และ RNN	16
ภาพที่ 5 สถาปัตยกรรมของเครือข่ายประสาทเทียมหน่วยความจำระยะยาว LSTM	17
ภาพที่ 6 สถาปัตยกรรมของเครือข่ายประทุกกลับ GRU	20
ภาพที่ 7 กระบวนการสร้าง Hybrid Deep Learning Model แบบ Cascaded neural network	23
ภาพที่ 8 สถาปัตยกรรมของของเครือข่ายประสาทเทียมแบบชั้นอย่างง่าย RNN และ RNN.....	24
ภาพที่ 9 ราคากลางรายวันของหุ้น XPG ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567 .	32
ภาพที่ 10 ราคากลางรายวันของหุ้น MTC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	32
ภาพที่ 11 ราคากลางรายวันของหุ้น KBANK ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	32
ภาพที่ 12 ราคากลางรายวันของหุ้น PTTGC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	33
ภาพที่ 13 ราคากลางรายวันของหุ้น AMC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	33
ภาพที่ 14 ราคากลางรายวันของหุ้น STANLY ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.	
2567	33
ภาพที่ 15 ราคากลางรายวันของหุ้น TOP ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	34
ภาพที่ 16 ราคากลางรายวันของหุ้น BCP ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	34
ภาพที่ 17 ราคากลางรายวันของหุ้น PTT ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	34

ภาพที่ 18 ราคาปิดรายวันของหุ้น RCL ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	35
ภาพที่ 19 ราคาปิดรายวันของหุ้น MINT ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567 35
ภาพที่ 20 ราคาปิดรายวันของหุ้น BDMS ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567 35
ภาพที่ 21 ราคาปิดรายวันของหุ้น SMT ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567	36
ภาพที่ 22 ราคาปิดรายวันของหุ้น HANA ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567 36
ภาพที่ 23 ราคาปิดรายวันของหุ้น TRUE ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567 36
ภาพที่ 24 ลักษณะและลำดับของข้อมูลที่ใช้เป็นตัวแปรต้นและตัวแปรตาม.	38



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการลงทุนมีบทบาทสำคัญมากต่อชีวิตของผู้คน เนื่องจากมีอัตราผลตอบแทนที่สูง และมีบทบาทสำคัญในการช่วยสร้างความมั่งคั่งและบรรลุเป้าหมายทางการเงิน เช่น การเกณฑ์อายุ การซื้อบ้าน เป็นต้น การลงทุนในปัจจุบันมีทางเลือกมากมายไม่ว่าจะเป็นตราสารทุน ตราสารหนี้ หรืออนุพันธ์ นักลงทุนทุกคนมีเป้าหมาย ระยะเวลา และความสามารถในการยอมรับความเสี่ยงแตกต่าง กัน อย่างไรก็ตาม การลงทุนที่ต้องการผลตอบแทนสูงมักจะมาพร้อมกับความเสี่ยงที่สูงขึ้น การทำนายราคาหุ้นจึงได้รับความสนใจจากนักลงทุนภาคเอกชน สถาบันหรือนักลงทุนรายย่อย นักลงทุนอาจใช้เทคนิคการลงทุนและการบริหารความเสี่ยงที่แตกต่างกัน ดังนั้นการใช้เครื่องมือทางสถิติและคณิตศาสตร์เพื่อช่วยในการคาดการณ์และการพยากรณ์จึงมีความสำคัญต่อการวางแผนการลงทุนและการบริหารความเสี่ยงในอนาคต

ในช่วงเวลาต่าง ๆ ตั้งแต่อดีตจนถึงปัจจุบันนี้มีการวิจัยมากมายเกี่ยวกับการพยากรณ์ราคาปิดหุ้น เพื่อใช้ในการตัดสินใจลงทุนเพื่อให้ได้ผลตอบแทนที่คาดหวัง นักวิจัยได้นำเสนอแบบจำลองทางคณิตศาสตร์หลายประเภทเพื่อการพยากรณ์ตั้งนี้ แบบจำลองอารีมา (ARIMA), แบบจำลองเวกเตอร์เชิงเส้น (Support Vector Machines : SVM), และในช่วงเวลาปัจจุบันได้มีการนำเสนอแบบจำลองการเรียนรู้เชิงลึก เช่น เครือข่ายประสาทเทียม (Artificial Neural Network : ANN), เครือข่ายประสาทเทียมแบบบวกกลับ (Recurrent Neural Network : RNN), Long Short-Term Memory (LSTM) รวมถึงการปรับปรุงประสิทธิภาพในการพยากรณ์อนุกรมเวลาโดยผสมแบบจำลอง ARIMA กับ Deep learning ซึ่งได้ผลการพยากรณ์ที่แม่นยำกว่า ARIMA เดิมที่เป็นแบบจำลองทางสถิติทั่วไปในการพยากรณ์ชุดข้อมูลอนุกรมเวลา

ในงานวิจัยของกุลวนิช (2564) ได้ทำการศึกษาแบบจำลองอนุกรมเวลาแบบสมรรถว่าง ARIMA และ ANN ร่วมกับข้อมูลหุ้นของ HMPRO และ SIRI โดยใช้วิธีการทดสอบแบบ Additive และ Linear Regression Combination Methods และประเมินประสิทธิภาพด้วย MSE, RMSE, MAE, MAPE และ SD ผลการศึกษาพบว่า การทดสอบแบบ Linear Regression มีประสิทธิภาพสูงสุดในการพยากรณ์ รองลงมาคือ Additive Combination Method และ ARIMA ตามลำดับ สรุปได้ว่าการใช้

แบบจำลองผสมเป็นทางเลือกที่มีประสิทธิภาพในการพยากรณ์ แต่ต้องแลกกับเวลาประมาณที่นานกว่าแบบจำลองเดี่ยว ซึ่งทางผู้วิจัยได้นำแนวคิดการผสมแบบ Additive Combination Method มาประยุกต์ใช้กับแบบจำลองการเรียนรู้เชิงลึก โดยแบบจำลองการเรียนรู้เชิงลึกสามารถวิเคราะห์ได้ทั้งสององค์ประกอบ คือ เชิงเส้นตรง และไม่เป็นเส้นตรง อย่างไรก็ตามยังคงมีส่วนเหลือ (Residual) จากการพยากรณ์อยู่ ดังนั้นผู้วิจัยจึงนำแบบจำลองการเรียนรู้เชิงลึกอื่น ๆ มาช่วยในการพยากรณ์องค์ประกอบที่ไม่เป็นเส้นตรงที่เหลืออยู่ เพื่อเพื่อปรับปรุงผลลัพธ์ในการพยากรณ์ ซึ่งในงานวิจัยนี้จะเรียกวิธีการนี้ว่า Cascaded neural network

ในงานวิจัยของ Liu et al. (2019) ได้ทำการเปรียบเทียบประสิทธิภาพในการพยากรณ์ระหว่างแบบจำลอง Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) และแบบจำลองโครงสร้างผสม GRU-LSTM สำหรับการพยากรณ์ราคาปิดของหุ้นสองตัว ได้แก่ Dalian Thermal Power Co.Ltd (600719) และ Dalian Friendship Co.Ltd (000679) โดยใช้ตัววัดประสิทธิภาพของแบบจำลองคือค่า Root Mean Square Error (RMSE) ผลการศึกษาพบว่าแบบจำลองโครงสร้างผสม GRU-LSTM ให้ค่า RMSE ต่ำที่สุด นอกจากนี้ยังพบว่าแบบจำลองโครงสร้างผสม GRU-LSTM ใช้เวลาในการฝึกฝนใกล้เคียงกับแบบจำลอง GRU ซึ่งเป็นแบบจำลองที่ใช้เวลาในการฝึกฝนน้อยที่สุด ดังนั้นสามารถสรุปได้ว่าแบบจำลองโครงสร้างผสม GRU-LSTM มีประสิทธิภาพในการพยากรณ์สูงกว่าแบบจำลอง GRU อย่างมีนัยสำคัญ

มีงานวิจัยที่นำผลการศึกษาโดย Liu et al. (2019) ไปต่อยอดในการพยากรณ์ราคาหุ้น แต่ได้ผลที่ขัดแย้งกัน โดยงานวิจัยของเกรว่อง (2566) ได้ศึกษาและเปรียบเทียบประสิทธิภาพของ Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) และแบบจำลองโครงสร้างผสม GRU-LSTM ในการพยากรณ์ราคาปิดหุ้นในอีก 30 วันข้างหน้า เพื่อครอบคลุมชุดข้อมูลอนุกรมเวลาที่มีความแตกต่างกันจึงคัดเลือกหุ้นจากค่าความผันผวน (Beta) ที่สูงท่อนถึงความผันผวนเมื่อเปรียบเทียบกับผลตอบแทนเฉลี่ยของตลาดหุ้นไทย โดยแบ่งหุ้นออกเป็น 3 กลุ่ม คือ กลุ่มที่มีความผันผวนต่ำ ได้แก่ ADVANCE และ BDMS กลุ่มที่มีความผันผวนปานกลาง ได้แก่ CPALL และ MINT และกลุ่มที่มีความผันผวนสูง ได้แก่ DELTA และ SAWAD การประเมินประสิทธิภาพของแบบจำลองใช้ตัวชี้วัด เช่น Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Reduction Rate of RMSE และระยะเวลาในการฝึกสอนแบบจำลอง จากการวิจัยพบว่า ในกลุ่มหุ้นที่มีความผันผวนต่ำ หุ้น ADVANCE ให้แบบจำลองโครงสร้างผสม GRU-LSTM เป็นแบบจำลองที่มี

ประสิทธิภาพสูงสุด ขณะที่หุ้น BDMS พบร่วมแบบจำลอง Recurrent Neural Network (RNN) ให้ประสิทธิภาพสูงสุด ในกลุ่มหุ้นที่มีความผันผวนปานกลาง หุ้น CPALL แบบจำลองโครงสร้างผสม GRU-LSTM เป็นแบบจำลองที่มีประสิทธิภาพสูงสุด ขณะที่หุ้น MINT พบร่วมแบบจำลอง GRU ให้ประสิทธิภาพสูงสุด ในกลุ่มหุ้นที่มีความผันผวนสูง หุ้น SAWAD แบบจำลอง LSTM เป็นแบบจำลองที่มีประสิทธิภาพสูงสุด ขณะที่หุ้น DELTA แบบจำลอง RNN ให้ประสิทธิภาพสูงสุด ดังนั้นสรุปได้ว่า แบบจำลองโครงสร้างผสม GRU-LSTM ไม่ได้มีประสิทธิภาพในการพยากรณ์สูงกว่าแบบจำลองการเรียนรู้เชิงลึกโครงสร้างเดี่ยวในทุกรอบ

ในงานวิจัยนี้มีจุดแตกต่างจากงานวิจัยของเกรว่อง (2566) ทั้งหมด 3 ข้อ ดังนี้ ประการแรก งานวิจัยของเกรว่อง (2566) ศึกษาเฉพาะแบบจำลองการเรียนรู้เชิงลึกแบบผสมในรูปแบบการใช้โครงสร้างผสม (Stacked layer) เท่านั้น ในขณะที่งานวิจัยนี้ศึกษาแบบจำลองการเรียนรู้เชิงลึกแบบผสมทั้งในรูปแบบการใช้โครงสร้างผสม (Stacked layer) และเครือข่ายประสาทเทียมแบบซ้อนกัน (Cascaded neural network) ประการที่สอง งานวิจัยของเกรว่อง (2566) ทำการพยากรณ์ราคาปิดในอีก 30 วันข้างหน้า ในขณะที่งานวิจัยนี้ทำการพยากรณ์ราคาปิดใน 2 ระยะเวลา ได้แก่ ระยะสั้น 7 วัน และระยะยาว 30 วัน เนื่องจากผู้วิจัยมองว่า nakong ทุนอาจไม่ได้ซื้อขายรายวันเสมอไป บางหุ้นอาจถือครองในระยะยาวถึงรายสัปดาห์หรือรายเดือน การพยากรณ์ในระยะเวลานี้จึงอาจตอบโจทย์นักลงทุนได้ดียิ่งขึ้น และสุดท้าย ตัวอย่างชุดข้อมูลที่เกรว่อง (2566) ใช้เลือกหุ้นจาก SET50 โดยอิงตามความผันผวนผ่านเกณฑ์เบอร์เซ็นต์ใกล้ๆ ในขณะที่ผู้วิจัยเลือกหุ้นจาก 5 กลุ่มอุตสาหกรรม คือ กลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และกลุ่มเทคโนโลยี โดยแต่ละกลุ่มมีหุ้น 3 ตัว จำแนกตามความผันผวนสูง ปานกลาง และต่ำ เนื่องจากหุ้นในอุตสาหกรรมเดียวกันมักมีปัจจัยพื้นฐานที่คล้ายคลึงกัน เช่น แนวโน้มการเติบโตและความเสี่ยงเฉพาะอุตสาหกรรม ซึ่งอาจส่งผลให้ราคาหุ้นเปลี่ยนแปลงในทิศทางที่คล้ายกัน

1.2 บริบทของปัญหา

งานวิจัยนี้การพยากรณ์ของแบบจำลองการเรียนรู้เชิงลึกแบบผสม RNN-LSTM, LSTM-RNN, RNN-GRU, GRU-RNN, LSTM-GRU และ GRU-LSTM ในรูปแบบการใช้โครงสร้างแบบผสม (Stacked layer) และเครือข่ายประสาทเทียมแบบซ้อนกัน (Cascaded neural network) ในการพยากรณ์ราคาปิดหุ้น เพื่อช่วยให้เลือกแบบจำลองและรูปแบบการผสมผสานให้เหมาะสมกับลักษณะของชุดข้อมูลที่มีความแตกต่างกันในแต่ละอุตสาหกรรมและระดับความผันผวน ภายใต้ระยะเวลาและความเสี่ยงที่นักลงทุนสามารถยอมรับได้

1.3 วัตถุประสงค์ของการวิจัย

- เพื่อเปรียบเทียบประสิทธิภาพและเวลาที่ใช้ของแบบจำลองการเรียนรู้เชิงลึกแบบผสม RNN-LSTM, LSTM-RNN, RNN-GRU, GRU-RNN, LSTM-GRU และ GRU-LSTM ในรูปแบบการใช้โครงสร้างแบบผสม (Stacked layer) และเครือข่ายประสาทเทียมแบบซ้อนกัน (Cascaded neural network) ในการพยากรณ์ราคาปิดหุ้น
- เพื่อสร้างแบบจำลองการเรียนรู้เชิงลึกแบบผสมที่เหมาะสมกับการพยากรณ์ราคาปิดหุ้นทั้งในระยะสั้น 7 วัน และ ระยะยาว 30 วัน

1.4 สมมติฐานการวิจัย

- แบบจำลองการเรียนรู้เชิงลึกแบบผสม RNN-LSTM, LSTM-RNN, RNN-GRU, GRU-RNN, LSTM-GRU และ GRU-LSTM ในรูปแบบการใช้ Cascaded neural network มีประสิทธิภาพที่แตกต่างจากการใช้ Stacked layer
- การสลับแบบจำลองที่ใช้ใน Stacked layers และ Cascaded neural network ส่งผลกระทบต่อประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึกแบบผสม

1.5 ขอบเขตของการวิจัย

- การศึกษาครั้งนี้ใช้ข้อมูลราคาปิดรายวันของหุ้น 5 ปีย้อนหลังตั้งแต่วันที่ 1 กรกฎาคม พ.ศ. 2563 ถึงวันที่ 30 มิถุนายน พ.ศ.2567 จาก <https://finance.yahoo.com/>
- คัดเลือกหุ้นจากความผันผวนของหุ้น (Beta 5 Years monthly) ณ วันที่ 1 กรกฎาคม 2567 ทั้งหมด 5 กลุ่มอุตสาหกรรม ได้แก่ กลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และกลุ่มเทคโนโลยี เลือกกลุ่มอุตสาหกรรมละ 3 หุ้น ทั้งหมด 15 ชุด ข้อมูล ดังตารางที่ 1

ตารางที่ 1 แสดงหุ้นในกลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และกลุ่มเทคโนโลยี แบ่งตามระดับความผันผวนของหุ้น (Beta)

ชุดที่	ความผันผวน	บริษัท	ชื่อหุ้น
กลุ่มธุรกิจการเงิน			
1.	สูง	บริษัท เอ็กซ์ซปริง แคปปิตอล จำกัด (มหาชน)	XPG
2.	ปานกลาง	บริษัท เมืองไทย แคปปิตอล จำกัด (มหาชน)	MTC
3.	ต่ำ	ธนาคารกสิกรไทย จำกัด (มหาชน)	KBANK
กลุ่มสินค้าอุตสาหกรรม			
4.	สูง	บริษัท พีทีที โกลบอล เคมิคอล จำกัด (มหาชน)	PTTGC
5.	ปานกลาง	บริษัท เอเชีย เมทัล จำกัด (มหาชน)	AMC
6.	ต่ำ	บริษัท ไทยสแตนเลสการไฟฟ้า จำกัด (มหาชน)	STANLY
กลุ่มทรัพยากร			
7.	สูง	บริษัท ไทยอยล์ จำกัด (มหาชน)	TOP
8.	ปานกลาง	บริษัท บางจาก คอร์ปอเรชั่น จำกัด (มหาชน)	BCP
9.	ต่ำ	บริษัท ปตท. จำกัด (มหาชน)	PTT
กลุ่มบริการ			
10.	สูง	บริษัท อาร์ ซี แอล จำกัด (มหาชน)	RCL
11.	ปานกลาง	บริษัท ไมเนอร์ อินเตอร์เนชันแนล จำกัด (มหาชน)	MINT
12.	ต่ำ	บริษัท กรุงเทพดุสิตเวชการ จำกัด(มหาชน)	BDMS
กลุ่มเทคโนโลยี			
13.	สูง	บริษัท สตาร์ส ไมโครอิเล็กทรอนิกส์ (ประเทศไทย) จำกัด (มหาชน)	SMT
14.	ปานกลาง	บริษัท hana ไมโครอิเล็กทรอนิกส์ จำกัด (มหาชน)	HANA
15.	ต่ำ	บริษัท ทรู คอร์ปอเรชั่น จำกัด (มหาชน)	TRUE

ที่มา: ตลาดหลักทรัพย์แห่งประเทศไทย (SET), Yahoo Finance

1.6 เงื่อนไขที่สำคัญของการวิจัย

ข้อจำกัดของการวิจัย

- ตัวแปรในงานวิจัยนี้ได้ใช้ตัวแปรอิสระ 5 ตัว คือ ราคาเปิด ราคาปิด ราคาสูงสุด ราคาต่ำสุด และปริมาณการซื้อขายรายวัน ซึ่งอาจไม่ได้ใช้ตัวแปรอื่นที่ส่งผลต่อการพยากรณ์ราคาปิดหุ้น อาจมีตัวแปร/ปัจจัยอื่น ๆ ที่ส่งผลต่อราคาปิดของหุ้นที่ไม่ได้นำมาวิเคราะห์ ซึ่งอาจส่งผลต่อประสิทธิภาพในการพยากรณ์ราคาหุ้น
- โปรแกรมค้นหาชุดพารามิเตอร์ที่เหมาะสมที่สุด (Search engine) ผู้วิจัยเลือกใช้ Randomized search เป็นการสุ่มค่าต่างๆ ที่ไม่แนบทรัพยากรคอมพิวเตอร์ของผู้วิจัยมีจำกัด

คำจำกัดความของงานวิจัย

ตารางที่ 2 แสดงคำจำกัดความของแบบจำลองการเรียนรู้เชิงลึก

คำศัพท์	คำจำกัดความของคำศัพท์
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
MLP	Multilayer perceptron

- EPOCH คือ จำนวนรอบของการฝึกฝนโดยนับ 1 รอบเมื่อใช้ข้อมูลทดสอบครบทั้งหมด
- Batch size คือ จำนวนข้อมูลตัวอย่างที่ใช้ฝึกฝนแบบจำลองในแต่ละรอบ
- ค่าเบต้า (Beta) คือ ค่าทางสถิติที่ใช้วัดความผันผวนของราคาหุ้น เมื่อเทียบกับดัชนีตลาดโดยรวม เช่น ค่าเบต้าของหุ้นในตลาดหลักทรัพย์แห่งประเทศไทย ก็คือ ค่าที่วัดความผันผวนของหุ้นตัวนั้นเทียบกับดัชนีตลาดหุ้นไทย โดยแบ่งออกเป็น 3 กรณีดังนี้
 - ค่าเบต้ามากกว่า 1: หุ้นมีความผันผวนมากกว่าตลาด หมายความว่า ถ้าตลาดขึ้น 1% หุ้นมีโอกาสขึ้นมากกว่า 1% และกลับกันถ้าตลาดลง 1% หุ้นมีโอกาสลงมากกว่า 1%
 - ค่าเบต้าเท่ากับ 1: หุ้นมีความผันผวนเท่ากับตลาด หมายความว่า ราคาหุ้นมีการเปลี่ยนแปลงไปในทิศทางเดียวกับตลาด
 - ค่าเบต้าน้อยกว่า 1: หุ้นมีความผันผวนน้อยกว่าตลาด หมายความว่า ถ้าตลาดขึ้น 1% หุ้นมีโอกาสขึ้นน้อยกว่า 1% และกลับกัน ถ้าตลาดลง 1% หุ้นมีโอกาสลงน้อยกว่า 1%

1.7 ระยะเวลาในการดำเนินการวิจัย

กรกฎาคม พ.ศ. 2567 ถึง เมษายน พ.ศ. 2568 เป็นระยะเวลาทั้งสิ้น 10 เดือน

ตารางที่ 3 แสดงขั้นตอนและระยะเวลาในการดำเนินการวิจัย

แผนการดำเนินงาน	ระยะเวลา									
	พ.ศ. 2567					พ.ศ. 2568				
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
กำหนดปัญหา										
ศึกษาทฤษฎีและกรอบแนวคิดที่เกี่ยวข้อง										
เก็บรวบรวมข้อมูลราคาหุ้น										
สร้างแบบจำลองพยากรณ์และเปรียบเทียบประสิทธิภาพ										
เขียนโครงร่างวิทยานิพนธ์										
เสนอโครงร่างวิทยานิพนธ์										
วิเคราะห์และสรุปผลการศึกษา										
นำเสนอวิทยานิพนธ์										
จัดทำรูปเล่มวิทยานิพนธ์										

1.8 ประโยชน์ที่คาดว่าจะได้รับ

- เพื่อเป็นแนวทางในการพิจารณาเลือกใช้แบบจำลองการเรียนรู้เชิงลึกแบบสมมาตร Stacked layers หรือ Cascaded neural network ในการพยากรณ์ราคาหลักทรัพย์ตัวอื่นในอนาคต
- นำแบบจำลองแบบสมนัยประยุกต์ใช้กับข้อมูลอนุกรมเวลาอื่น ๆ ได้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

2.1.1 การวิเคราะห์อนุกรมเวลา (Time series analysis)

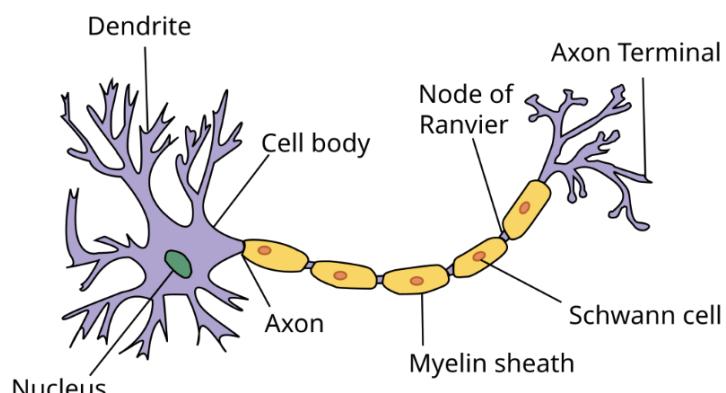
เทคนิคทางสถิติที่ใช้ในการวิเคราะห์และทำความเข้าใจข้อมูลที่ขึ้นอยู่กับเวลา รวมถึงการแยกองค์ประกอบของอนุกรมเวลาออกเป็นแนวโน้ม ฤดูกาล และโครงสร้างพื้นฐานอื่น ๆ ที่ແengอยู่ภายในข้อมูล เพื่อนำไปใช้ในการพยากรณ์อนาคต รวมถึงการตัดสินใจอย่างชาญฉลาดโดยอาศัยข้อมูลตามช่วงเวลา การวิเคราะห์องค์ประกอบเหล่านี้ทำให้สามารถระบุรูปแบบ ตรวจจับสิ่งผิดปกติ และตัดสินใจบนพื้นฐานของข้อมูลได้อย่างมีประสิทธิภาพ นอกจากนี้ ยังสามารถนำไปประยุกต์ใช้ในหลายสาขา เช่น เศรษฐศาสตร์ การเงิน การบริหารจัดการ และวิทยาศาสตร์ เพื่อเพิ่มความแม่นยำในการคาดการณ์และการวางแผนเชิงกลยุทธ์ ตัวอย่างเช่น การวิเคราะห์แนวโน้มราคาหุ้น การคาดการณ์ยอดขายสินค้า หรือการพยากรณ์สภาพอากาศ เป็นต้น (Pandian, 2025) โดยข้อมูลอนุกรมเวลาแบ่งออกเป็น 4 ส่วนประกอบหลัก ได้แก่

- 1) แนวโน้ม (Trend) เป็นการเคลื่อนไหวในระยะยาวของอนุกรมเวลา มีแนวโน้มเพิ่มขึ้นหรือลดลงไปในทิศทางเดียวกันหนึ่ง ซึ่งแนวโน้มอาจเป็นแบบเชิงเส้นหรือไม่เชิงเส้น และสามารถแสดงถึงการเติบโต การเสื่อมถอยลง หรือลักษณะอื่น ๆ ของข้อมูลในระยะยาว
- 2) ฤดูกาล (Seasonality) เป็นรูปแบบที่เกิดขึ้นซ้ำในช่วงเวลาที่แน่นอน (เช่น รายวัน รายเดือน รายไตรมาส หรือรายปี) ฤดูกาลอาจได้รับอิทธิพลจากปัจจัยต่าง ๆ เช่น สภาพอากาศ วันหยุด หรือพฤติกรรมของผู้บริโภค ซึ่งสามารถทำนายได้ตามช่วงเวลาที่เกิดขึ้นซ้ำ ๆ
- 3) วัฏจักร (Cyclicity) เป็นรูปแบบที่เกิดขึ้นซ้ำ (คล้ายคลึงกับฤดูกาล) ในช่วงเวลาที่ไม่แน่นอน มีการเปลี่ยนแปลงระยะยาวที่ยาวนานกว่าฤดูกาลเพียงอย่างเดียว วัฏจักรอาจเกิดจากปัจจัยทางเศรษฐกิจ แนวโน้มของอุตสาหกรรม หรือปัจจัยภายนอกอื่น ๆ
- 4) ค่าคลาดเคลื่อน (Residuals) เป็นการเปลี่ยนแปลงที่เหลืออยู่ในอนุกรมเวลาหลังจากคำนึงถึงแนวโน้ม ฤดูกาล และวัฏจักรแล้ว โดยทั่วไปค่าคลาดเคลื่อนจะถูกสมมติให้เป็นสัญญาณรบกวนหรือการเปลี่ยนแปลงที่ไม่สามารถอธิบายได้ เช่น วิกฤตโรคระบาด ภัยธรรมชาติ หรือเหตุการณ์ที่ไม่คาดคิด

2.1.2 เครือข่ายประสาทเทียม (Artificial Neural Networks - ANN)

เครือข่ายประสาทเทียมเป็นแบบจำลองทางคณิตศาสตร์ที่มีหลักการเบื้องต้นมาจากการเลียนแบบความสามารถของสมองมนุษย์ ซึ่งถูกคิดค้นขึ้นในปี ค.ศ. 1943 โดย Warren S. McCulloch และ Walter Pitts เพื่อใช้อธิบายลักษณะการทำงานของระบบประสาทของมนุษย์ โดยมีจุดประสงค์ในการสร้างเครื่องมือที่มีความสามารถในการเรียนรู้ที่ซับซ้อน เช่น การจดจำรูปแบบ (Pattern recognition) และการอนุมานความรู้ (Knowledge deduction) ซึ่งเครือข่ายประสาทเทียมนี้ได้รับความนิยมในการพยากรณ์ด้านธุรกิจ โดยเฉพาะด้านการเงิน เนื่องจากมีความสามารถในการพยากรณ์ข้อมูลที่ไม่เป็นพังก์ชันเชิงเส้นตรงในท้องของค่าพารามิเตอร์ได้อย่างมีประสิทธิภาพ จนนับได้มีการศึกษาและพัฒนามาอย่างต่อเนื่อง จนกระทั่งปี ค.ศ. 1957 Frank Rosenblatt เสนอแนวคิดเพอเซ็ปตรอน (perceptron) ซึ่งวางรากฐานสำคัญของ Deep Learning ที่ใช้กันอย่างแพร่หลาย (ประกอบด้วย, 2552)

เครือข่ายประสาทเทียมประกอบด้วยชุดของเซลล์ประสาท (Neurons) ที่เชื่อมต่อกันและจุดประสาประสาท (Synapses) โครงสร้างของการส่งสัญญาณประสาทประกอบขึ้นจากการเชื่อมต่อระหว่างตัวเซลล์ประสาท (Cell body) หรือล้านเซลล์ โดยเซลล์ประสาทแต่ละเซลล์ประกอบด้วยแขนงรับสัญญาณประสาทซึ่งเปรียบเสมือนหน่วยรับข้อมูลเข้าเรียกว่า เดนไดร็ต (Dendrites) และส่วนปลายของเซลล์ประสาทในการส่งสัญญาณประสาทซึ่งเปรียบเสมือนหน่วยส่งข้อมูลออกของเซลล์เรียกว่า แอคชอน (Axon) การส่งสัญญาณประสาทดังกล่าวอาจทำให้เกิดทั้งการกระตุ้นและการยับยั้งนอกจากรีวิวการประมวลผลภายในเซลล์ประสาทแต่ละเซลล์ยังมีการขยายหรือลดขนาดของสัญญาณโดยสัญญาณจากเดนไดร็ตต่าง ๆ จะรวมกันเข้าสู่เซลล์ประสาท และหากสัญญาณรวมมีความแรงเกินค่าระดับ (Threshold) ของเซลล์ประสาทนั้น เซลล์ประสาทก็จะส่งสัญญาณ出去ทางแอคชอนต่อไป



ภาพที่ 1 กระบวนการทำงานภายในเซลล์ประสาท

ที่มา : <https://biologydictionary.net/nervous-system/>

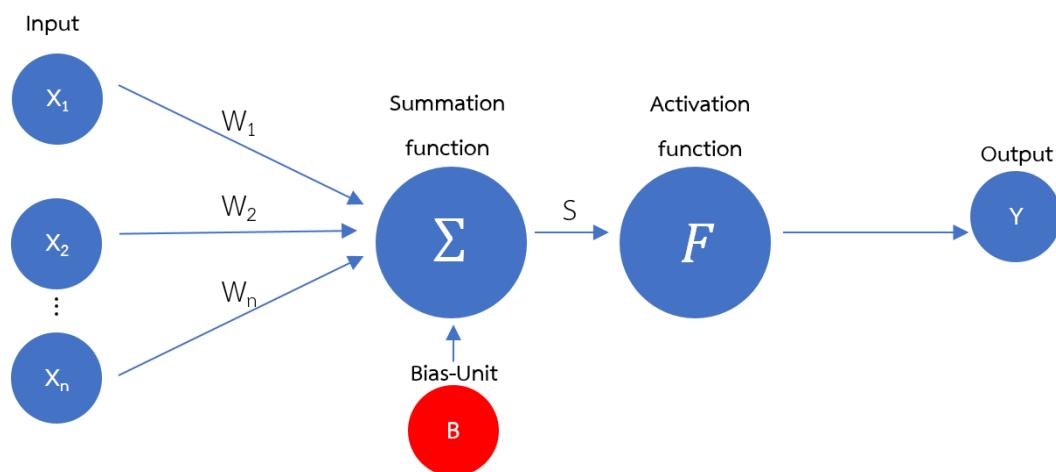
เครือข่ายประสาทเทียมมีคุณลักษณะที่คล้ายคลึงกับการส่งสัญญาณประสาทในสมองของมนุษย์ กล่าวคือ มีความสามารถในการรวบรวมความรู้ (Knowledge) ผ่านกระบวนการเรียนรู้ (Learning process) โดยความรู้เหล่านี้จะถูกจัดเก็บอยู่ในเครือข่ายในรูปแบบของค่าน้ำหนัก (Weights) ซึ่งสามารถปรับเปลี่ยนค่าได้เมื่อมีการเรียนรู้สิ่งใหม่ ๆ ค่าค่าน้ำหนักเหล่านี้ทำหน้าที่เชื่อมต่อความรู้ที่มนุษย์รวบรวมไว้ เพื่อใช้ในการแก้ไขปัญหาเฉพาะทางที่พบเจอ การประมวลผลต่าง ๆ เกิดขึ้นในหน่วยประมวลผลอย่างที่เรียกว่า โหนด (Node) ซึ่งโหนดเหล่านี้เป็นการจำลองลักษณะการทำงานของเซลล์ประสาทในสมองมนุษย์ การส่งสัญญาณระหว่างโหนดที่เชื่อมต่อกัน (Connections) เปรียบเสมือนการเชื่อมต่อของเดินไดร์ตและแอคชันในระบบประสาทมนุษย์

นอกจากนี้ภายในระบบประสาทของมนุษย์จะมีฟังก์ชันที่กำหนดสัญญาณส่งออกเรียกว่า ฟังก์ชันกระตุ้น (Activation function) หรือฟังก์ชันการแปลง (Transfer function) ซึ่งทำหน้าที่เชื่อมกระบวนการทำงานในเซลล์ประสาท (ธนาวุฒิ ประกอบผล, 2009) ดังภาพที่ 1 ซึ่งมีบทบาทสำคัญในการจำลองการทำงานของสมองมนุษย์ในกระบวนการเรียนรู้และการประมวลผลข้อมูล การทำงานของฟังก์ชันเหล่านี้มีความสำคัญในการกำหนดการตอบสนองของโหนดตามข้อมูลที่ได้รับ ซึ่งทำให้เครือข่ายประสาทเทียมสามารถประมวลผลข้อมูลได้อย่างมีประสิทธิภาพ เครือข่ายประสาทเทียมประกอบด้วย 5 องคประกอบหลัก ดังนี้

- 1) ข้อมูลนำเข้า (Input) เป็นข้อมูลที่เป็นตัวเลข หากเป็นข้อมูลเชิงคุณภาพต้องแปลงให้อยู่ในรูปเชิงปริมาณที่โครงข่ายประสาทเทียมยอมรับได้
- 2) ข้อมูลส่งออก (Output) คือผลลัพธ์ที่เกิดขึ้นจริง (Actual output) จากกระบวนการเรียนรู้ของเครือข่ายประสาทเทียม
- 3) ค่าน้ำหนัก (Weights) คือสิ่งที่ได้จากการเรียนรู้ของเครือข่ายประสาทเทียมหรือเรียกอีกอย่างหนึ่งว่าค่าความรู้ (Knowledge) ค่านี้อาจถูกเก็บเป็นทักษะเพื่อใช้ในการจัดจำข้อมูลอื่น ๆ ที่อยู่ในรูปแบบเดียวกัน
- 4) ฟังก์ชันผลรวม (Summation function) เป็นผลรวมของข้อมูลป้อนเข้าและค่าน้ำหนัก
- 5) ฟังก์ชันการแปลง (Transfer function) เป็นการคำนวณการจำลองการทำงานของโครงข่ายประสาทเทียม เช่น โดยฟังก์ชันกระตุ้นที่นิยมใช้ในเครือข่ายประสาทเทียม ได้แก่ ฟังก์ชัน Sigmoid, Tanh และ ReLU

เพอร์เซปตรอน (Perceptron) เป็นอัลกอริทึมพื้นฐานที่ใช้ในการเรียนรู้ข้อมูลจากตัวอย่างหลักการทำงานของเพอร์เซปตรอนคือการจำลองเครือข่ายระบบประสาทของมนุษย์ โดยมีโนนด (Node) เปรียบเสมือนนิวรอน (Neuron) ที่ทำหน้าที่เก็บรวบรวมข้อมูลเข้ามา (Input) และนำค่าข้อมูลเข้าในแต่ละโนนดไปผ่านการคำนวณค่าน้ำหนัก (Weight) ในแต่ละแคนประสาท จากนั้นจะถูกส่งไปรวมกันที่ปลายประสาทของเซลล์ (Synapse) เพื่อนำค่าข้อมูลที่รวมกันนั้นไปผ่านฟังก์ชันกระตุ้น (Activation Function) และส่งออกมายเป็นข้อมูลนำออก (Output) เพอร์เซปตรอนประกอบด้วยองค์ประกอบหลัก 3 ส่วน ได้แก่ ชั้นนำเข้าข้อมูล (Input Layer) ชั้นซ่อน (Hidden Layer) และชั้นนำออกข้อมูล (Output Layer) ซึ่งแต่ละชั้นจะประกอบด้วยโนนดที่มีจำนวนต่างกันออกไป และโนนดใน input layer จะเชื่อมไปข้างหน้ากับทุก node ในชั้นถัดไปตามลำดับจนถึง output layer (กุลวนิช, 2564)

ทั้งนี้แบบจำลองเครือข่ายประสาทเทียมที่นิยมใช้ในการพยากรณ์อนุกรมเวลา คือ เครือข่ายประสาทเทียมเพอร์เซปตรอนหลายชั้น (Multilayer Perceptron: MLP) โดย MLP จะมีความแตกต่างกับเพอร์เซปตรอนที่จะมีการเพิ่มจำนวนชั้นซ่อนเข้าไป เพื่อให้แบบจำลองสามารถเข้าใจรูปแบบที่มีความซับซ้อนของข้อมูลได้ดียิ่งขึ้น การคำนวณภายในแต่ละชั้นยังคงลักษณะเดิม โดยการรับข้อมูลจากชั้นก่อนหน้า ผ่านการคำนวณค่าน้ำหนัก และฟังก์ชันกระตุ้นก่อนส่งต่อไปชั้นถัดไป ดังภาพที่ 2



ภาพที่ 2 โครงสร้างการทำงานของเครือข่ายประสาทเทียม

- 1) ชั้นข้อมูลนำเข้า (Input layer) ทำหน้าที่นำเข้าข้อมูลสู่เครือข่ายประสาทเทียมโดยข้อมูลนี้จะนำไปประมวลผลในแต่ละหน่วยของชั้นถัดไป
- 2) ชั้นซ่อน (Hidden layer) ทำหน้าที่รับข้อมูลจากชั้นนำเข้าข้อมูลโดยกำหนดค่าน้ำหนักของข้อมูลจากชั้นนำเข้าก่อนรับข้อมูลเข้าสู่ชั้นซ่อนชั้นจะทำหน้าที่เพิ่มประสิทธิภาพในการจัดกลุ่มข้อมูลก่อนจะส่งต่อข้อมูลไปยังชั้นถัดไป
- 3) ชั้นข้อมูลส่งออก (Output layer) ทำหน้าที่ส่งออกข้อมูลโดยผ่านการประมวลผลจากฟังก์ชันรวม(Summation function: S) เป็นผลรวมของข้อมูลที่ป้อนเข้ามาโดยการถ่วงค่าน้ำหนัก (W) กับค่าความเออนเอียง (Bias unit: B) และพังก์ชันการแปลง (Transfer function) ดังสมการที่ (1)

$$S = \sum_{i=1}^n W_i X_i + B \quad (1)$$

แบบจำลอง MLP, ANN หรือแบบจำลองการเรียนรู้เชิงลึกชนิดอื่น ๆ มีคำจำกัดความบางคำที่ใช้กันอย่างแพร่หลาย เพื่อความเข้าใจในหลักการทำงานของแบบจำลองการเรียนรู้เชิงลึกมากขึ้น ดังนั้น จึงขออธิบายความหมายของคำจำกัดความที่สำคัญ (ภูสีเจีย哥, 2564) ดังนี้

- 1) ค่าน้ำหนัก (Weight: W) คือ ค่าที่แสดงถึงความสำคัญระหว่างโหนดต้นทางและโหนดปลายทาง หากโหนดต้นทางมีผลกระทบต่อโหนดปลายทางมาก ค่าถ่วงน้ำหนักจะมีค่าสูงขึ้น ตามไปด้วย ค่าถ่วงน้ำหนักเป็นพารามิเตอร์ที่ต้องเรียนรู้จากข้อมูลที่ถูกนำมาฝึกฝนในแบบจำลองการเรียนรู้เชิงลึก เพื่อปรับปรุงความแม่นยำในการพยากรณ์ โดยการปรับค่าถ่วงน้ำหนักนี้จะเกิดขึ้นในกระบวนการเรียนรู้ ซึ่งเป็นการปรับเปลี่ยนค่าน้ำหนักให้เหมาะสมกับลักษณะของข้อมูลที่ได้รับ
- 2) ค่าความเออนเอียง (Bias-Unit: B) คือ หน่วยปรับข้อมูล ณ จุดตัดแกน เนื่องจากการคำนวณของแบบจำลอง MLP มีลักษณะเป็นการคำนวณเชิงเส้นตรงก่อนเข้าพังก์ชันกระตุน Bias-Unit จึงทำหน้าที่เป็นจุดตัดแกนที่ช่วยเลื่อนค่าจากการคำนวณเชิงเส้นตรงไปยังตำแหน่งที่เหมาะสม ทำให้แบบจำลองการเรียนรู้เชิงลึกสามารถฝึกฝนและหารูปแบบจากข้อมูลตัวอย่างได้อย่างมีประสิทธิภาพมากขึ้น โดย Bias-Unit จะมีค่าเท่ากับ 1 เสมอและอยู่ในทุกชั้นของแบบจำลองการเรียนรู้เชิงลึก ยกเว้นชั้นนำออกข้อมูล การใช้ Bias-Unit ช่วยให้แบบจำลองสามารถเข้าใจและจับรูปแบบของข้อมูลได้ดีขึ้น

3) พังก์ชันกระตุ้น (Activation Function: F) คือพังก์ชันที่แปลงค่าจากสมการเชิงเส้นให้เป็นค่าที่มีความไม่เป็นเชิงเส้น พังก์ชันนี้มีความสำคัญต่อการเรียนรู้เชิงลึก เพราะช่วยให้โมเดลสามารถจับความสัมพันธ์ที่ไม่เป็นเชิงเส้นระหว่างตัวแปรต้นและตัวแปรเป้าหมายได้ดีขึ้น โดยแต่ละประเภทของพังก์ชันกระตุ้นมีข้อดีและข้อเสียที่แตกต่างกัน ขึ้นอยู่กับลักษณะการใช้งานของแต่ละโดยพังก์ชันกระตุ้นที่ใช้งานในงานวิจัยนี้มีรายละเอียดดังนี้

3.1 พังก์ชัน Sigmoid เป็นพังก์ชันรูปตัว S ที่แปลงค่าข้อมูลให้อยู่ในช่วง 0 ถึง 1 ซึ่งสามารถตีความค่าเป็นความน่าจะเป็นได้ เนื่องจากค่าผลลัพธ์ของพังก์ชัน Sigmoid จะอยู่ในช่วงระหว่าง 0 ถึง 1 จึงเหมาะสมสำหรับการใช้งานในขั้นข้อมูลนำออกของโมเดลที่เกี่ยวข้องกับการแบ่งกลุ่ม (classification) หรือการตีความเชิงความน่าจะเป็น แต่พังก์ชัน Sigmoid ไม่เหมาะสมสำหรับการใช้งานใน hidden layer เนื่องจาก การบีบอัดข้อมูลอาจทำให้สูญเสียข้อมูลสำคัญและทำให้การเรียนรู้ช้าลง โดยเฉพาะเมื่อค่าจากสมการเชิงเส้นมีค่าสูงหรือต่ำเกินไป ซึ่งจะทำให้อนุพันธ์ของพังก์ชัน Sigmoid ใกล้ศูนย์ เกิดปัญหา vanishing gradient และการปรับปรุงค่าน้ำหนักเกิดขึ้นช้า เนื่องให้อยู่ในรูปแบบพังก์ชันได้ดังสมการที่ (2)

$$f(s) = \text{sigmoid}(s) = \frac{1}{1 + e^{-s}} \quad (2)$$

3.2 พังก์ชัน Tanh หรือ Hyperbolic Tangent ให้ผลลัพธ์ในช่วง -1 ถึง 1 โดยไม่ลู่ออกและมีค่า 0 อยู่ตรงกลาง ทำให้การเรียนรู้ของแบบจำลองเร็วขึ้นและมีประสิทธิภาพเดียวกับที่แม่นยำกว่า Sigmoid ดังนั้น Tanh มักมีประสิทธิภาพดีกว่า Sigmoid แต่พังก์ชัน Sigmoid ไม่เหมาะสมสำหรับการใช้งานใน hidden layer เช่นกันเนื่องจาก การบีบอัดข้อมูลที่อาจทำให้สูญเสียข้อมูลสำคัญเช่นเดียวกับพังก์ชัน Sigmoid เนื่องให้อยู่ในรูปแบบพังก์ชันได้ดังสมการที่ (3)

$$f(s) = \tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (3)$$

3.3 พังก์ชัน ReLU (Rectified Linear Unit) เป็นพังก์ชันกระตุ้นที่นิยมใช้ในเลเยอร์ช่อนอกของโครงข่ายประสาทเทียมเชิงลึก ข้อดีหลักของพังก์ชันนี้คือ การลดปัญหาการหายไปของกรเดียโนต์ (vanishing gradient), การระเบิดหรือการลุ่อกของกรเดียโนต์ (exploding gradient) รวมถึงการไม่บีบอัดข้อมูล ซึ่งทำให้การเรียนรู้จากชุดข้อมูลมีประสิทธิภาพดีขึ้น อย่างไรก็ตาม พังก์ชัน ReLU อาจทำให้เซลล์ประสาท

บางตัวไม่ทำงานเมื่อค่าป้อนเข้ามีน้อยกว่าศูนย์ เพราะจะปรับค่าดังกล่าวเป็นศูนย์ ส่งผลให้เซลล์ประสาทเหล่านี้ไม่สามารถเรียนรู้จากข้อมูลใหม่ได้ แต่ถ้าค่าป้อนเข้ามีค่ามากกว่าศูนย์ พังก์ชันจะช่วยให้การเรียนรู้เกิดขึ้นได้รวดเร็วและมีประสิทธิภาพ เขียนให้อยู่ในรูปแบบพังก์ชันได้ดังสมการที่ (4)

$$f(s) = \text{relu}(s) = \begin{cases} s_i & , s \geq 0 \\ 0 & , s \leq 0 \end{cases} \quad (4)$$

- 4) พังก์ชันต้นทุน (Cost Function หรือ Loss Function) เป็นสมการที่ใช้ในการประเมินความถูกต้องของโมเดลที่พัฒนาขึ้น โดยอิงจากชุดพารามิเตอร์ที่เป็นข้อมูลนำเข้า พังก์ชันต้นทุนจะคำนวณค่าผิดพลาดของโมเดล ซึ่งค่าที่ได้จากพังก์ชันนี้แสดงถึงระดับของข้อผิดพลาดที่เกิดขึ้น กล่าวอีกนัยหนึ่ง ชุดพารามิเตอร์ที่ให้ค่าพังก์ชันต้นทุนต่ำที่สุดจะเป็นชุดพารามิเตอร์ที่เหมาะสมที่สุดสำหรับการใช้งานรูปแบบของสมการพังก์ชันต้นทุนจะแตกต่างกันไปตามลักษณะของปัญหาที่ต้องการแก้ไข โดยทั่วไปพังก์ชันต้นทุนสามารถแบ่งออกเป็น 2 ประเภท ตามลักษณะของปัญหา ดังสมการที่ (5) และ (6)

Regression Problem

$$J(\theta) = -\frac{1}{n} \sum_{t=1}^n (h_\theta(x_t) - y_t)^2 \quad (5)$$

Classification Problem

$$J(\theta) = -\frac{1}{n} \sum_{t=1}^n y_t \log(h_\theta(x_t)) + (1 - y_t) \log(1 - h_\theta(x_t)) \quad (6)$$

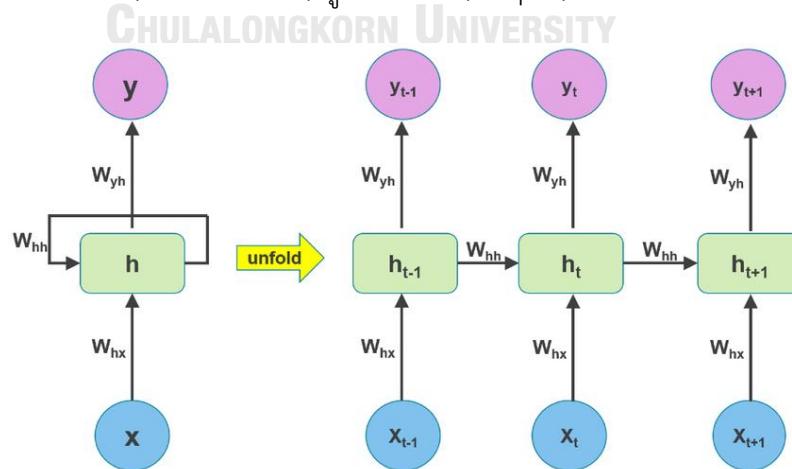
2.1.3 เครือข่ายประสาทเทียมแบบย้อนกลับ (Recurrent Neural Networks: RNN)

เครือข่ายระบบประสาทแบบย้อนกลับ (Recurrent Neural Network: RNN) เป็นสถาปัตยกรรมรูปแบบหนึ่งของเครือข่ายประสาทเทียม (Artificial Neural Network: ANN) ที่ออกแบบมาเพื่อจัดการกับข้อมูลประเภทตรวจจับรูปแบบในลำดับข้อมูล ข้อมูลดังกล่าวอาจเป็นลายมือเขียน จิโนม ข้อความ และอนุกรมเวลา (Time Series Data) มักพบในบริบทอุตสาหกรรม (เช่น ตลาดหุ้นหรือเชื้อเชื้อ) หลักการของ RNN คือการปรับโครงสร้างของเครือข่ายประสาทเทียมแบบตั้งเดิม โดยเพิ่มชั้นซ่อน (Hidden State) เพื่อจดจำข้อมูลหรือความรู้จากอดีตและผสมผสานเข้ากับข้อมูลใหม่ที่เข้ามา (Input Data) เพื่อใช้ในการทำนายหรือพยากรณ์ ซึ่งสามารถแสดงได้ดังสมการที่ (7) และ (8) และภาพที่ 3

$$h_t = f_h(W_{hh}h_{t-1} + W_{hx}x_t) \quad (7)$$

$$y_t = f_y(W_{yh}h_t) \quad (8)$$

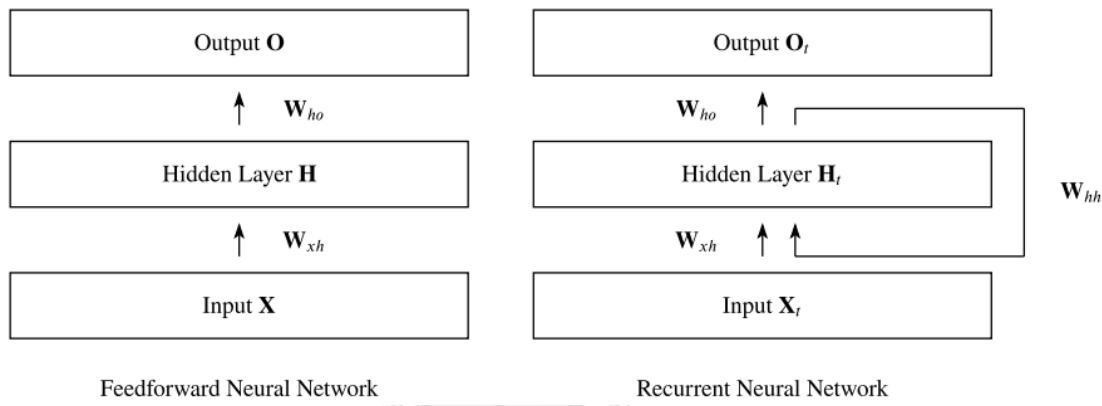
โดยที่ h_t คือ สถานะซ่อน (Hidden State) ณ เวลา t , f_h คือ พังก์ชันกระตุ้น (Activation Function) สำหรับสถานะซ่อน (Hidden State), f_y คือ พังก์ชันกระตุ้น (Activation Function) สำหรับค่าผลลัพธ์ (Output), y_t คือ ค่าผลลัพธ์ (Output) ณ เวลาตอน t , x_t คือ ค่านำเข้า (Input) ณ เวลาตอน t , W_{hx} คือ เมทริกซ์นำหนักสำหรับการเชื่อมต่อจากข้อมูลนำเข้า (Input) สู่สถานะซ่อน (Hidden State), คือ W_{hh} เมทริกซ์นำหนักสำหรับการเชื่อมต่อจากสถานะซ่อน (Hidden State) ในช่วงก่อนหน้าสู่สถานะซ่อน (Hidden State) ปัจจุบัน, W_{yh} คือ เมทริกซ์นำหนักสำหรับการเชื่อมต่อจากสถานะซ่อน (Hidden State) สู่ค่าผลลัพธ์ (Output)



ภาพที่ 3 โครงสร้างของเครือข่ายระบบประสาทแบบย้อนกลับ (RNN)

ที่มา : Prakash Venugopal (2019)

ความแตกต่างระหว่างโครงข่ายระบบประสาทแบบย้อนกลับ (Recurrent Neural Networks: RNN) กับโครงข่ายประสาทเทียมแบบพีดฟอร์เวิร์ด (Feedforward Neural Network) หรือที่รู้จักกันในชื่อ Multi-Layer Perceptron (MLP) อยู่ที่วิธีการส่งข้อมูลผ่านเครือข่าย ในขณะที่ Feedforward Neural Networks ส่งข้อมูลผ่านเครือข่ายโดยไม่มีวัฏจักร RNNs มีการส่งข้อมูลผ่าน เครือข่ายโดยมีวัฏจักรและส่งข้อมูลกลับเข้ามาในตัวเอง ซึ่งช่วยให้ RNNs สามารถขยายความสามารถ ของ Feedforward Neural Networks ให้คำนึงถึงข้อมูลนำเข้าก่อนหน้า $x_{0:t-1}$ ไม่ใช่เพียงข้อมูล นำเข้าปัจจุบัน x_t ดังภาพที่ 4



ภาพที่ 4 ความแตกต่างระหว่าง Feedforward Neural Networks และ RNN

อย่างไรก็ตาม RNN ประสบปัญหาการลดลงของเกรเดียนต์ (Vanishing Gradient) หรือการ ระเบิดของเกรเดียนต์ (Exploding Gradient) ปัญหาเหล่านี้เกิดขึ้นเนื่องจากในโครงข่ายประสาท เทียมจะมีการปรับเปลี่ยนค่าถ่วงน้ำหนัก (Update Weights) โดยอาศัยวิธีการแพร่กลับของความผิด พลาด (Backward Propagation of Errors หรือ Backpropagation) ซึ่งจะคำนวณเกรเดียนต์ของ พังก์ชันการสูญเสีย (Loss Function) เพื่อใช้ในการปรับเปลี่ยนค่าถ่วงน้ำหนัก ในกรณีของ RNN เอาจริงๆ ไม่ได้มาจากช่วงเวลาเดียว ($t=t$) แต่มาจากหลายช่วงเวลา ($t=t-1, t-2, \dots, t$) ทำให้เกิดการ คุณกันของอนุพันธ์หลายตัว ซึ่งหากเกรเดียนต์มีค่าน้อย (น้อยกว่า 1) การคุณกันแบบนี้จะส่งผลให้ค่า เกรเดียนต์ลดลงเรื่อยๆ ตามความยาวของลำดับข้อมูล หรืออีกนัยหนึ่ง RNN จะมีปัญหาในการ ประมวลผลข้อมูลที่มีลำดับยาวเกินไป

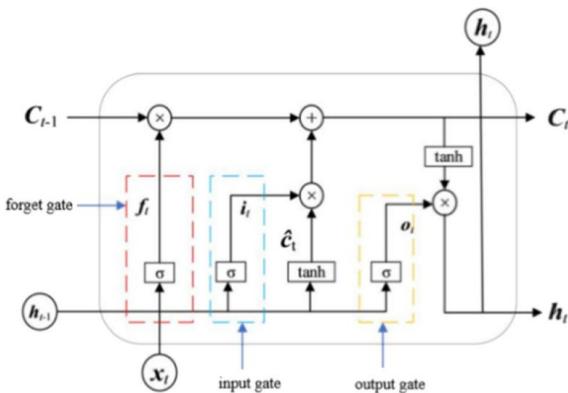
2.1.4 เครือข่ายหน่วยความจำระยะสั้นแบบยาว (Long Short-Term Memory: LSTM)

เครือข่ายประสาทเทียมหน่วยความจำระยะสั้นแบบยาว หรือ LSTM เป็นแบบจำลองเครือข่ายที่ถูกเสนอโดย Hochreiter and Schmidhuber (1997) LSTM ถูกออกแบบมาเพื่อจัดการกับข้อมูลลำดับจำนวนมาก โดยมีเป้าหมายเพื่อแก้ไขปัญหาที่เกิดขึ้นเป็นเวลานานในแบบจำลอง RNN เช่น ปัญหา exploding gradient และ vanishing gradient โดยหลักการสำคัญที่ช่วยให้แบบจำลอง LSTM สามารถลดการเกิดปัญหา exploding gradient และ vanishing gradient ได้คือ

1. การคำนวณค่าสถานะการลืม (Forget gate) เป็นการคิดค่าต่อร่องน้ำหนักเพื่อตัดสินใจนำข้าสถานะเซลล์ก่อนหน้ามาใช้ในสัดส่วนเท่าไหร่
2. การคำนวณค่าสถานะการรับข้อมูลใหม่ (Input gate) เป็นการคิดค่าต่อร่องน้ำหนักเพื่อตัดสินใจนำค่าข้อมูลน้าเข้าใหม่มาใช้ในสัดส่วนเท่าไหร่

หลักการทำงานของแบบจำลอง LSTM (Pirani et al., 2022) องค์ประกอบดังนี้

1. สถานะเซลล์ (Cell State) คือตัวเก็บค่าสถานะของเซลล์ความจำ (Cell memory) โดยจะเก็บค่าสถานะเซลล์เพื่อนำไปใช้คำนวณต่อในประตุลีม และประตุนำเข้าต่อไป
2. ประตุ (Gate) เป็นตัวควบคุมปริมาณนำเข้าข้อมูลของค่าสถานะเซลล์ เพื่อตัดสินใจว่าจะใช้ค่าสถานะเซลล์ในลำดับก่อนหน้าเทียบกับสถานะเซลล์ในลำดับปัจจุบันเป็นสัดส่วนเท่าไหร่ โดยคำนวณจากค่านำเข้าข้อมูลใหม่และค่าชั้นช่องในลำดับก่อนหน้า โดยแต่ละประตุก็มีเงื่อนไขการคิดค่าต่อร่องน้ำหนักของสถานะเซลล์ต่างกัน ประกอบด้วยสามส่วนหลัก ได้แก่ ประตุการลืม (forget gate), ประตุการป้อนเข้า (input gate), และประตุการส่งออก (output gate) ดังภาพที่ 5



ภาพที่ 5 สถาปัตยกรรมของเครือข่ายประสาทเทียมหน่วยความจำระยะยาว LSTM

C_{t-1} คือสถานะเซลล์ของช่วงเวลา ก่อนหน้า, h_{t-1} คือค่าผลลัพธ์สุดท้ายของหน่วยประสาท LSTM ในช่วงเวลา ก่อนหน้า, x_t คือข้อมูลนำเข้า สำหรับช่วงเวลาปัจจุบัน, σ คือฟังก์ชันกระตุ้น, f_t คือผลลัพธ์ของเกตลีมในช่วงเวลาปัจจุบัน, i_t คือผลลัพธ์ของเกตข้อมูลนำเข้าในช่วงเวลาปัจจุบัน, \tilde{C}_t คือสถานะเซลล์ผู้สมัครในช่วงเวลาปัจจุบัน, o_t คือค่าผลลัพธ์ของเกตเอาต์พุต, C_t คือสถานะเซลล์ในช่วงเวลาปัจจุบัน, และ h_t คือผลลัพธ์ของช่วงเวลาปัจจุบัน โดยกระบวนการคำนวณของ LSTM (Lu et al., 2021) เป็นดังนี้

- 1) ค่าผลลัพธ์ของช่วงเวลา ก่อนหน้า และค่าของข้อมูลนำเข้า ในช่วงเวลาปัจจุบัน จะถูกป้อนเข้าสู่ forget gate ผลลัพธ์ของ forget gate จะได้หลังจากการคำนวณตามที่แสดงในสมการ (9)

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (9)$$

โดยที่ช่วงค่าของ f_t อยู่ระหว่าง 0 ถึง 1, W_f คือน้ำหนักของ forget gate, b_f คือค่าเอนเอียงของ forget gate, x_t คือค่าของข้อมูลนำเข้า ณ เวลา t, และ h_{t-1} คือค่าของ Hidden Vector Activation ณ เวลา t-1

- 2) ค่าผลลัพธ์ของเวลาที่ผ่านมา และค่าของข้อมูลนำเข้า ในเวลาปัจจุบัน จะถูกป้อนเข้าสู่เกตข้อมูลนำเข้า ค่าผลลัพธ์และสถานะเซลล์ผู้สมัครของเกตข้อมูลนำเข้า จะได้มาหลังจากการคำนวณซึ่งแสดงในสมการ (10) และ (11)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (10)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (11)$$

โดยที่ช่วงค่าของ i_t อยู่ระหว่าง 0 ถึง 1, W_i คือน้ำหนักของ input gate, b_i คือค่าเอนเอียงของ input gate, W_c คือน้ำหนักของ candidate input gate, และ b_c คือค่าเอนเอียงของ candidate input gate

- 3) Cell state ปัจจุบัน จะได้รับการอัปเดตดังสมการ (12)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (12)$$

โดยที่ช่วงค่าของ C_t อยู่ระหว่าง 0 ถึง 1

- 4) ค่าผลลัพธ์ของช่วงเวลา ก่อนหน้า และค่าของข้อมูลนำเข้าในเวลาปัจจุบันจะถูกป้อนเข้าไปใน output gate ค่าผลลัพธ์ของ output gate จะได้รับหลังจากการคำนวณตามที่แสดงในสมการ (13)

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (13)$$

โดยที่ช่วงค่าของ o_t อยู่ระหว่าง 0 ถึง 1, W_o คือน้ำหนักของ output gate, และ b_o คือค่าเออนเอียงของ output gate

- 5) ค่าผลลัพธ์ของ LSTM ได้มาจากการคำนวณผลลัพธ์ของ output gate และ Cell state ดังที่แสดงในสมการ (14)

$$h_t = o_t * \tanh(C_t) \quad (14)$$

2.1.5 เครือข่ายประตุวากลับ (Gated Recurrent Units - GRU)

เครือข่ายประตุวากลับ หรือ GRU เป็นรูปแบบหนึ่งของโครงข่ายระบบประสาทแบบย้อนกลับ หรือ RNN ที่ถูกพัฒนาขึ้นโดย Chung et al. (2014) เพื่อแก้ไขปัญหา Vanishing Gradient ซึ่งเป็นปัญหาที่เกิดขึ้นในโครงข่าย RNN ที่มีหน่วยความจำระยะสั้น โดยเฉพาะในบริบทของการประมวลผลข้อมูลลำดับยาว ๆ เช่นเดียวกับโครงข่าย LSTM (Long Short-Term Memory) GRU ได้รับการออกแบบมาเพื่อกีบรักษาข้อมูลระยะยาวโดยลดความซับซ้อนของการคำนวณ โดยลดจำนวนประตุ (gate) จากสามประตุใน LSTM ลงเหลือเพียงสองประตุ ได้แก่ ประตุรีเซ็ต (reset gate) และประตุอัพเดต (update gate) เพื่อลดภาระในการคำนวณและเพิ่มความเร็วในการประมวลผล โครงสร้างของเครือข่ายประตุวากลับ GRU ดังภาพที่ 6

- 1) ประตุอัพเดต (Update Gate: z) ในเครือข่าย GRU มีหน้าที่ในการคำนวณค่าถ่วงน้ำหนักของค่าสถานะชั้นช่อนในลำดับก่อนหน้า กับค่าสถานะผู้ท้าชิงในลำดับปัจจุบันเพื่อใช้สำหรับการคำนวณ ค่าสถานะในลำดับปัจจุบัน โดยคำนวณจาก ข้อมูลนำเข้าในลำดับปัจจุบันและค่าสถานะชั้นช่อนในลำดับก่อนหน้า ประตุนี้ถูกควบคุมโดยฟังก์ชัน Sigmoid (σ) ซึ่งให้ผลลัพธ์ที่มีค่าอยู่ระหว่าง 0 ถึง 1 เพื่อทำการตัดสินใจว่าจะมีการรับสัญญาณจากค่าสถานะชั้นช่อนก่อนหน้ามาในอัตราส่วนเท่าไหร่ สมการสำหรับการคำนวณค่าของ Update Gate สามารถแสดงได้ดังสมการที่ (15)

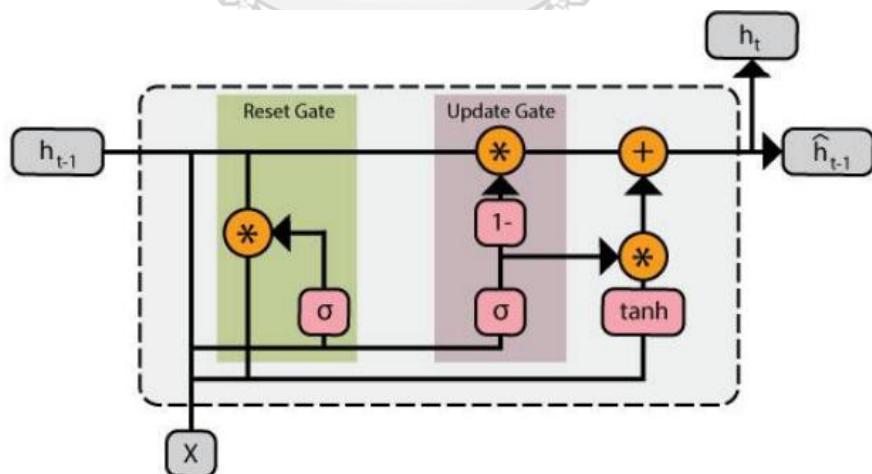
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (15)$$

โดยที่ z_t คือ ค่าถ่วงน้ำหนักของ Update Gate, x_t คือค่าของข้อมูลนำเข้า ณ เวลา t, h_{t-1} คือค่าของ hidden vector activation ณ เวลา t-1, W_{xz} และ W_{hz} คือ เมทริกซ์ น้ำหนักของ update gate, และ b_z คือค่าเอนเอียงของ update gate

- 2) ประตุรีเซ็ต (Reset Gate: r) ในเครือข่าย GRU มีหน้าที่ในการคำนวนค่าถ่วงน้ำหนักของค่าสถานะชั้นช่องในลำดับก่อนหน้า โดยจะคำนวนจากข้อมูลนำเข้าในลำดับปัจจุบันและค่าสถานะชั้นช่องในลำดับก่อนหน้า ตัดสินใจว่าควรปรับเปลี่ยนสถานะของเซลล์เมื่อมีข้อมูลใหม่เข้ามาหรือไม่ ประตุรีเซ็ตควบคุมโดยฟังก์ชัน Sigmoid (σ) ซึ่งให้ผลลัพธ์ที่มีค่าอยู่ระหว่าง 0 ถึง 1 ในกระบวนการคำนวนนี้จะใช้ข้อมูลนำเข้าปัจจุบันและสถานะช่อง (Hidden state) ก่อนหน้า สมการการคำนวนของ reset gate สามารถแสดงได้ดังสมการที่ (16)

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (16)$$

โดยที่ r_t คือ ค่าถ่วงน้ำหนักของ reset gate, x_t คือค่าของข้อมูลนำเข้า ณ เวลา t, h_{t-1} คือค่าของ hidden vector activation ณ เวลา t-1, W_{xr} และ W_{hr} คือ เมทริกซ์ น้ำหนักของ reset gate, และ b_r คือค่าเอนเอียงของ reset gate



ภาพที่ 6 สถาปัตยกรรมของเครือข่ายประตุรีเซ็ต GRU

หลังจากที่ได้สมการของประตุทั้งสองแล้ว ขั้นตอนถัดไปคือการคำนวณ Output ที่ทำนายในสถานะชั้นช่อง ณ เวลาปัจจุบัน (\tilde{h}_t) ซึ่งคำนวณจากการรวมกันของข้อมูลนำเข้าที่เวลาปัจจุบัน (x_t) คูณกับเมตริกซ์ค่าถ่วงน้ำหนักของเออร์พูต (W_{xh}) กับการทำ Element-wise product (\odot) ระหว่าง reset gate (r) และสถานะชั้นช่องก่อนหน้า (Hidden state) เพื่อตัดสินใจว่าเราควรใช้สถานะของชั้นช่องก่อนหน้านี้ในระดับใด ผลลัพธ์จากการทำ Element-wise product (\odot) จะถูกคูณกับเมตริกซ์ค่าถ่วงน้ำหนักของ Update Gate ดังนั้น สมการสำหรับการคำนวณค่าดังกล่าวสามารถแสดงได้ดังสมการที่ (17)

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1})) \quad (17)$$

โดยที่ r_t คือ ค่าถ่วงน้ำหนักของ reset gate, x_t คือค่าของข้อมูลนำเข้า ณ เวลา t, h_{t-1} คือค่าของ hidden vector activation ณ เวลา t-1, \tilde{h}_t คือค่าของ candidate hidden vector activation ณ เวลา t และ W_{xh} และ W_{hh} คือ เมทริกซ์น้ำหนักของ reset gate

เนื่องจากฟังก์ชันกราฟตุ้นที่ใช้คือ ฟังก์ชัน Hyperbolic Tangent ซึ่งเป็น nonlinear function ค่าที่ได้จะอยู่ในช่วง -1 ถึง 1 ฟังก์ชันนี้ช่วยป้องกันปัญหา Vanishing Gradient เนื่องจากมันจำกัดค่าผลลัพธ์ให้อยู่ในช่วงที่กำหนด ทำให้สามารถคำนวณค่าการแพร่รากลับของความผิดพลาด (Backpropagation) ได้โดยค่าที่ได้ไม่เป็นค่าอนันต์ ซึ่งช่วยให้กระบวนการเรียนรู้ของโมเดลมีความเสถียรมากขึ้น

ขั้นตอนสุดท้ายคือการคำนวณสถานะชั้นช่อง ณ เวลาปัจจุบันเพื่อใช้เป็นผลลัพธ์สุดท้าย หรือเพื่อทำการตัดสินใจว่าควรปรับเปลี่ยนค่าเป็นค่าใดระหว่างผลลัพธ์ที่ทำนายได้จากสถานะชั้นช่องก่อนหน้า (\tilde{h}_t) และผลลัพธ์ที่ทำนายได้จากสถานะชั้นช่องปัจจุบัน (h_t) ในสัดส่วนที่กำหนด สมการสำหรับการคำนวณค่าดังกล่าวสามารถแสดงได้ดังสมการที่ (18)

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (18)$$

โดยที่ h_t คือ ค่าของ hidden vector activation ณ เวลา t, h_{t-1} คือค่าของ hidden vector activation ณ เวลา t-1, \tilde{h}_t คือค่าของ candidate hidden vector activation ณ เวลา t, z_t คือ z_t คือ ค่าถ่วงน้ำหนักของ Update Gate

2.1.6 แบบจำลองการเรียนรู้เชิงลึกแบบผสม (Hybrid Deep Learning Model)

แบบจำลองการเรียนรู้เชิงลึกแบบผสม ซึ่งเป็นการผสมผสานของเทคนิคหรือสถาปัตยกรรมของแบบจำลองของการเรียนรู้เชิงลึก (Deep Learning) หลาย ๆ แบบเข้าด้วยกัน เช่น Recurrent Neural Networks (RNN), หรือ Long Short-Term Memory (LSTM) เข้าด้วยกัน เพื่อเพิ่มประสิทธิภาพในการคาดการณ์หรือวิเคราะห์ข้อมูล มีความสามารถในการจัดการกับข้อมูลที่ซับซ้อนมากขึ้น ลดข้อจำกัดของแต่ละแบบจำลองและสามารถปรับตัวให้เข้ากับการใช้งานที่หลากหลายได้ดีขึ้น หมายถึงไม่เดลการเรียนรู้เชิงลึกที่ผสมผสานเทคนิคหรือ เพื่อให้ได้ผลลัพธ์ที่ดียิ่งขึ้น โดยสามารถรวมเอาคุณสมบัติของโมเดลที่มีอยู่

Cascaded neural network

Zhang (2003) ได้นำเสนอแบบจำลองอนุกรมเวลาแบบผสม ซึ่งมีสมมติฐานว่าข้อมูลอนุกรมเวลาที่นำมาวิเคราะห์ประกอบด้วยสององค์ประกอบ ได้แก่ องค์ประกอบเชิงเส้นตรง (linear component) และองค์ประกอบที่ไม่เป็นเส้นตรง (non-linear component) Zhang ได้นำแบบจำลอง ARIMA ซึ่งเป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์มาผสานกับแบบจำลอง ANN ที่ใช้สำหรับการพยากรณ์ข้อมูลที่ไม่เป็นฟังก์ชันเชิงเส้นตรงในเทอมของค่าพารามิเตอร์

ทางผู้วิจัยได้นำแนวคิดดังกล่าวมาประยุกต์ใช้กับการผสมแบบจำลองการเรียนรู้เชิงลึก โดยแบบจำลองการเรียนรู้เชิงลึกสามารถวิเคราะห์ได้ทั้งสององค์ประกอบ คือ องค์ประกอบเชิงเส้นตรง (linear component) และองค์ประกอบที่ไม่เป็นเส้นตรง (non-linear component) อย่างไรก็ตาม ยังคงมีส่วนเหลือ (Residual) จากการพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกอยู่ ดังนั้นผู้วิจัยจึงนำแบบจำลองการเรียนรู้เชิงลึกอื่น ๆ มาช่วยในการพยากรณ์องค์ประกอบที่เหลือ เพื่อปรับปรุงผลลัพธ์ และลดความผิดพลาดในการพยากรณ์ แบบจำลองดังกล่าวมีขั้นตอนการสร้างดังภาพที่ 7 และสามารถแสดงได้ด้วยสมการที่ (19)

$$Y_t = F_t^v + F_t^\varepsilon + \varepsilon_t \quad (19)$$

เมื่อ Y_t คือ ผลลัพธ์สุดท้ายจากการพยากรณ์ข้อมูลอนุกรมเวลา ณ เวลาที่ t , F_t^v คือ ข้อมูลที่ได้จากการพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองที่หนึ่ง, F_t^ε คือ ข้อมูลส่วนเหลือ (Residual) ที่ได้จากการพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองที่สอง และ ε_t คือ ค่าความคลาดเคลื่อนสุดท้าย ณ เวลาที่ t

โดยมีขั้นตอนการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบสมอย่างละเอียดดังนี้

ขั้นตอนที่ 1 วิเคราะห์ข้อมูลอนุกรมเวลาด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองแรก เพื่อให้ได้ค่าพยากรณ์ (Predicted value) ในเทอมของค่าพารามิเตอร์ คือ \hat{F}_t^{DL1}

ขั้นตอนที่ 2 คำนวณค่าส่วนเหลือ (Residual) จากแบบจำลองการเรียนรู้เชิงลึกแบบจำลองแรก ดังสมการ (20)

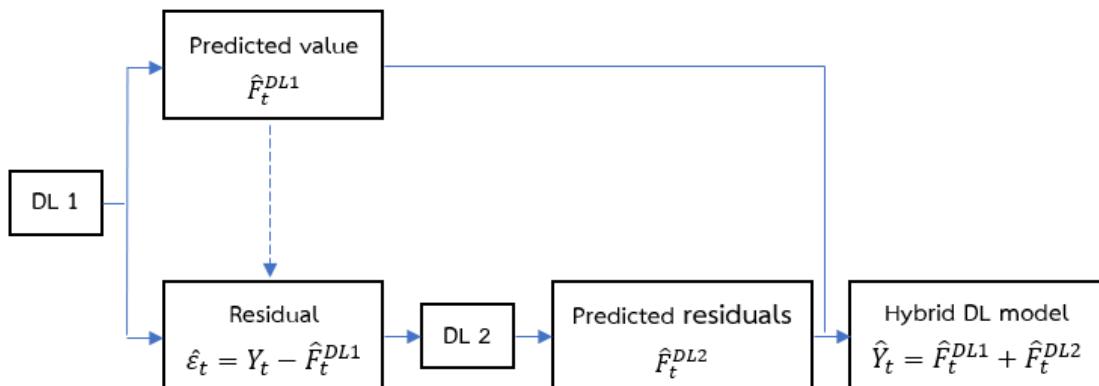
$$\hat{\epsilon}_t = Y_t - \hat{F}_t^{DL1} \quad (20)$$

ขั้นตอนที่ 3 นำค่าส่วนเหลือที่ได้จากแบบจำลองไปพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองที่สอง เพื่อหาค่าพยากรณ์ความคลาดเคลื่อน (Predicted residuals) ในเทอมของค่าพารามิเตอร์ คือ \hat{F}_t^{DL2}

ขั้นตอนที่ 4 คำนวณหาค่าพยากรณ์โดยนำค่าจากขั้นตอนที่ 1 และขั้นตอนที่ 3 มารวมกันดังสมการที่ (21)

$$\hat{Y}_t = \hat{F}_t^{DL1} + \hat{F}_t^{DL2} \quad (21)$$

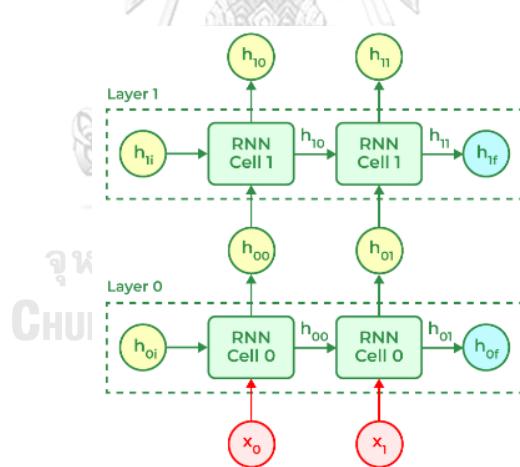
เมื่อ \hat{Y}_t คือ ผลลัพธ์สุดท้ายจากการพยากรณ์ข้อมูลอนุกรมเวลา ณ เวลาที่ t , \hat{F}_t^{DL1} คือ ข้อมูลที่ได้จากการพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองที่หนึ่ง และ \hat{F}_t^{DL2} คือ ข้อมูลส่วนเหลือ (Residual) ที่ได้จากการพยากรณ์ด้วยแบบจำลองการเรียนรู้เชิงลึกแบบจำลองที่สอง



ภาพที่ 7 กระบวนการสร้าง Hybrid Deep Learning Model แบบ Cascaded neural network

Stacked layers in neural network

เป็นการผสมผสานสถาปัตยกรรมหลายแบบเข้าด้วยกันโดยจัดเรียงเป็นชั้น ๆ (layers) ต่อเนื่องกันตามลำดับ ในเครือข่ายประสาทเทียม (Neural Networks) เรียกว่า "stacked" เพื่อให้ข้อมูลสามารถไหลผ่านโมเดลได้อย่างมีประสิทธิภาพ โดยแต่ละชั้นจะรับผลลัพธ์จากชั้นก่อนหน้าเป็นข้อมูลนำเข้า (Input) และส่งผลลัพธ์ของตัวเองไปยังชั้นถัดไป การจัดเรียงชั้nonชั้น (stacked layers) เป็นประโยชน์อย่างยิ่งในการเพิ่มความลึกของโมเดล ทำให้สามารถตรวจสอบรูปแบบที่เกิดขึ้นในระดับตื้นและระดับลึกได้อย่างมีประสิทธิภาพ ด้วยเหตุนี้ โมเดลแบบ Stacked จึงสามารถเรียนรู้และจำรูปแบบข้อมูลในลำดับที่ยาวนานได้ และในขณะเดียวกันยังสามารถวิเคราะห์ข้อมูลจากสถานะปัจจุบันโดยอ้างอิงข้อมูลจากสถานะก่อนหน้าได้อีกด้วย นอกจากนี้ ยังเพิ่มจำนวนเลเยอร์ในโมเดลมากขึ้น เท่าใด เครือข่ายแบบ Stacked ก็ยิ่งสามารถตรวจสอบรูปแบบที่ซับซ้อนมากขึ้นในข้อมูลลำดับได้มากเท่านั้น อีกทั้งโครงสร้างของ stacked layers ยังสามารถปรับเปลี่ยนได้ง่าย โดยสามารถเพิ่มหรือลดจำนวนเครือข่ายได้ตามความต้องการของผู้ใช้งาน โดยตัวอย่างสถาปัตยกรรมของเครือข่ายประสาทเทียมแบบชั้nonชั้นอย่างง่ายเป็นการซ้อนกันระหว่าง RNN (Salalli, 2023) แสดงดังภาพที่ 8



ภาพที่ 8 สถาปัตยกรรมของเครือข่ายประสาทเทียมแบบชั้nonชั้นอย่างง่าย RNN และ RNN

โดย x_0 และ x_1 คือข้อมูลนำเข้า ถูกป้อนเข้าสู่ RNN Cell 0 ซึ่งทำการประมวลผลและส่งผลลัพธ์ออกมาเป็นสถานะช่องที่มีค่า h_{00} และ h_{01} สถานะช่องสุดท้ายของเลเยอร์นี้คือ h_{0f} จากนั้นสถานะช่องที่มีค่า h_{00} และ h_{01} จะถูกนำมาใช้เป็นข้อมูลนำเข้าสำหรับเลเยอร์ 1 ใน RNN Cell 1 เลเยอร์นี้ทำการประมวลผลข้อมูลที่ได้รับและส่งผลลัพธ์ออกมาเป็นสถานะช่อง h_{10} และ h_{11} และได้สถานะช่องสุดท้ายของเลเยอร์นี้คือ h_{1f}

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

ตั้งแต่อดีตจนถึงปัจจุบันมีงานวิจัยมากมายที่ศึกษาเกี่ยวกับการพยากรณ์ราค้าปิดหุ้น โดยในช่วงแรกนักวิจัยได้นำแบบจำลอง ARIMA มาใช้ในการพยากรณ์ราค้าปิดหุ้นเนื่องจากเป็นแบบจำลองพื้นฐานทางสถิติที่ใช้ในการพยากรณ์ข้อมูลประเภทอนุกรมเวลา ถัดมาได้มีการนำแบบจำลองการเรียนรู้เชิงลึกเข้ามาใช้ในการพยากรณ์ราค้าปิดหุ้นมากขึ้น รวมถึงการนำแบบจำลองการเรียนรู้เชิงลึกเข้ามาใช้ร่วมกับแบบจำลองพื้นฐานทางสถิติอย่างแบบจำลอง ARIMA โดยการผสมผสานแบบจำลอง เพื่อเพิ่มความแม่นยำในการพยากรณ์ให้กับข้อมูลอนุกรมเวลา ซึ่งการผสมผสานแบบจำลองนี้ได้รับความสนใจในงานวิจัยต่าง ๆ มา กขึ้น โดยมีการศึกษาและเปรียบเทียบประสิทธิภาพในการพยากรณ์ของแบบจำลอง ARIMA แบบจำลองการเรียนรู้เชิงลึก และแบบจำลองที่ผสมผสานกัน

Wijaya et al. (2010) ได้ทำการเปรียบเทียบประสิทธิภาพการพยากรณ์ระหว่างแบบจำลอง ARIMA และแบบจำลอง ANN โดยใช้ชุดข้อมูลรายวันของหุ้น ANTM (PT Aneka Tambang) สำหรับการพยากรณ์ราค้าปิดหุ้นในวันถัดไป ในการสร้างแบบจำลอง ARIMA ใช้ข้อมูลจากเดือนพฤษภาคม ค.ศ. 2007 ถึงเดือนธันวาคม ค.ศ. 2007 โดยได้แบบจำลอง ARIMA (0,1,2) ส่วนแบบจำลอง ANN ใช้ข้อมูลตั้งแต่ปี 2004 ถึง 2007 โดยมีตัวแปรต้นที่ใช้ในแบบจำลอง ANN จำนวน 7 ตัวแปร ได้แก่ ราคา เปิด ราค้าปิด ราค้าสูงสุด ราค่าต่ำสุดของวันก่อนหน้า ดัชนีตลาดหลักทรัพย์อินโดนีเซีย (IDX) ดัชนี JCI และราคากองในวันปัจจุบัน เพื่อพยากรณ์ราค้าปิดหุ้นในวันเดียวกัน ข้อมูลที่ใช้ในการทดสอบแบบจำลองอยู่ในช่วงเดือนกันยายน ค.ศ. 2007 ถึงธันวาคม ค.ศ. 2007 ผลการวิจัยพบว่า แบบจำลอง ANN มีประสิทธิภาพในการพยากรณ์ที่ดีกว่าแบบจำลอง ARIMA โดยแบบจำลอง ANN ให้ค่า ESS (Error Sum of Squares) เท่ากับ 170.4 ซึ่งน้อยกว่าแบบจำลอง ARIMA ที่มีค่า ESS เท่ากับ 284.95 และพบว่า ANN ใช้เวลาฝึกฝนมากกว่า ARIMA

Adebisi et al. (2014) ได้ทำการเปรียบเทียบประสิทธิภาพการพยากรณ์ระหว่างแบบจำลอง ARIMA และ ANN โดยใช้ข้อมูลรายวันของหุ้น Dell Incorporation ที่เก็บตั้งแต่วันที่ 17 สิงหาคม 1988 ถึง 25 กุมภาพันธ์ 2011 สำหรับแบบจำลอง ANN ได้ใช้ตัวแปรต้น เช่น ราค้าเปิด ราค้าปิด ราค้าสูงสุด ราค่าต่ำสุด ปริมาณการซื้อขายในวันก่อนหน้า และตัวแปรตามคือราค้าปิดของวันนั้น ส่วนแบบจำลอง ARIMA ได้ใช้ราค้าปิดในวันก่อนหน้าในการสร้างแบบจำลอง โดยเลือกใช้แบบจำลอง ARIMA (1,0,0) ซึ่งมีค่า BIC ต่ำที่สุด ในการปรับไฮเปอร์พารามิเตอร์ของแบบจำลอง ANN พบร้า แบบจำลองที่ดีที่สุดประกอบด้วยชั้นนำเข้าข้อมูล 10 หน่วย ชั้นตอน 17 หน่วย และชั้นนำออกข้อมูล

1 หนด งานวิจัยนี้ใช้ค่า Forecast Error (FE) ซึ่งคำนวณจาก (actual – predicted) / (actual) เป็นเกณฑ์ในการวัดประสิทธิภาพการพยากรณ์ ผลการวิจัยพบว่าแบบจำลอง ANN มีความแม่นยำสูงกว่าแบบจำลอง ARIMA ใน การพยากรณ์ แต่ถ้าพิจารณาด้วย FE และการทดสอบทางสถิติพบว่าไม่มีความแตกต่างอย่างมีนัยสำคัญระหว่างค่า FE ของแบบจำลองทั้งสอง

บุญมานะ (2558) ได้ทำการเปรียบเทียบประสิทธิภาพการพยากรณ์ของแบบจำลอง ARIMA, แบบจำลองผสมระหว่าง ARIMA-ANN และแบบจำลองผสมระหว่าง ARIMA-SVM ใน การพยากรณ์ ราคาปิดหุ้นของธนาคารไทยพาณิชย์ จำกัด (มหาชน) ตั้งแต่วันที่ 2 กรกฎาคม พ.ศ. 2550 ถึงวันที่ 8 กุมภาพันธ์ พ.ศ. 2559 งานวิจัยนี้ใช้วิธีการจำลองชุดข้อมูลอนุกรมเวลาด้วยตัวแบบ ARIMA(0,1,1) ARIMA(0,1,2), ARIMA(1,1,0), ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(2,1,0), ARIMA(2,1,1) และ ARIMA(2,1,2) ซึ่งในแต่ละตัวแบบจะทำการจำลองชุดข้อมูลอนุกรมเวลาภายใต้สถานการณ์เดียวกันจำนวน 1,000 รอบและใช้เกณฑ์พิจารณาความแม่นยำของตัวแบบคือ RMSE พบว่าตัวแบบที่เหมาะสมที่สุดสำหรับใช้พยากรณ์ราคาปิดหุ้น SCB คือ ARIMA(1,1,1) จากนั้นนำส่วนเหลือจากการพยากรณ์ด้วย ARIMA(1,1,1) มาพยากรณ์ด้วย ANN และ SVM ผลการวิจัยพบว่าแบบจำลองผสมระหว่าง ARIMA และ SVM มีความแม่นยำในการพยากรณ์สูงสุดรองลงมาคือ ARIMA และแบบจำลองผสมระหว่าง ARIMA และ ANN

จงศิริวิโรจน์ (2564) ได้ทำการเปรียบเทียบประสิทธิภาพการพยากรณ์ของแบบจำลอง ARIMA, ANN และแบบจำลองผสมระหว่าง ARIMA-ANN ใน การพยากรณ์ราคาปิดหุ้นในกลุ่มเทคโนโลยีและอิเล็กทรอนิกส์ โดยศึกษาลักษณะของหุ้นที่มีค่าเบต้าต่างกัน ซึ่งแสดงถึงความผันผวนเมื่อเทียบกับตลาด ได้แก่ HANA ที่มีค่าเบต้าสูง DELTA ที่มีค่าเบต้าปานกลาง และ SVI ที่มีค่าเบต้าต่ำ ในช่วงเวลา ตั้งแต่เดือนตุลาคม พ.ศ. 2559 ถึงเดือนตุลาคม พ.ศ. 2564 ซึ่งเป็นระยะเวลา 5 ปี งานวิจัยนี้ใช้วิธีการตรวจสอบไขว้เพื่อคัดเลือกแบบจำลองที่เหมาะสมกับข้อมูลแต่ละชุด โดยใช้ตัววัดประสิทธิภาพ MAPE และ RMSE เป็นเกณฑ์ในการคัดเลือกแบบจำลอง ผลการวิจัยพบว่าแบบจำลอง ANN มีประสิทธิภาพในการพยากรณ์สูงกว่าแบบจำลอง ARIMA และ ARIMA-ANN และแบบจำลองผสมไม่ได้ให้ผลการพยากรณ์ที่ดีกว่าแบบจำลองเดียวเสมอไป อย่างไรก็ตาม แบบจำลอง ANN ใช้เวลาในการฝึกฝนข้อมูลมากกว่าแบบจำลอง ARIMA

กุลวนิช (2564) ได้นำแบบจำลองอนุกรมเวลาแบบผสมระหว่าง ARIMA และ ANN ใช้ร่วมกับข้อมูลหุ้นของ HMPRO และ SIRI โดยใช้การผสมแบบ Additive Combination Method และ Linear Regression Combination Method (Khairalla et al., 2017) การประเมินประสิทธิภาพของแบบจำลองใช้ตัวชี้วัด MSE RMSE MAE MAPE และ SD ผลการศึกษาพบว่า การใช้แบบจำลองแบบผสมระหว่าง ARIMA และ ANN ด้วยการผสมแบบ Linear Regression Combination Method มีประสิทธิภาพในการพยากรณ์สูงที่สุด รองลงมาคือ แบบจำลองแบบผสมระหว่าง ARIMA และ ANN ด้วยการผสมแบบ Additive Combination Method และ ARIMA ตามลำดับ ดังนั้นสามารถสรุปได้ว่าการพิจารณาเลือกใช้ตัวแบบผสมเพื่อใช้ในการพยากรณ์จัดว่าเป็นอีกทางเลือกที่มีประสิทธิภาพ และสามารถช่วยลดความผิดพลาดในการพยากรณ์ได้ แต่ทั้งนี้ตัวแบบผสมก็มีข้อจำกัดในเรื่องระยะเวลาที่ต้องใช้ในการประมวลผลที่นานกว่าเมื่อเปรียบเทียบกับตัวแบบเดียว

งานวิจัยที่อ้างถึงแสดงให้เห็นว่าโดยส่วนใหญ่แล้วแบบจำลองการเรียนรู้เชิงลึก รวมถึงการใช้แบบจำลองแบบผสมร่วมกับแบบจำลองการเรียนรู้เชิงลึก มีความสามารถในการพยากรณ์ที่ดีกว่าแบบจำลองพื้นฐานทางสถิติทั่วไป เช่น ARIMA หรือ ARIMAX เนื่องจากผู้วิจัยมุ่งหวังในการพัฒนาแบบจำลองที่ให้ประสิทธิภาพในการพยากรณ์สูงสุด จึงไม่ได้ทำการศึกษาและเปรียบเทียบกับ ARIMA หรือ ARIMAX แต่เน้นไปที่การศึกษาและเปรียบเทียบแบบจำลองการเรียนรู้เชิงลึก เช่น RNN, LSTM และ GRU ซึ่งสามารถพยากรณ์ข้อมูลอนุกรมเวลาได้ดี นอกจากนี้ยังมีงานวิจัยจำนวนมากที่มุ่งเน้นการศึกษาและปรับปรุงวิธีการต่างๆ เพื่อเพิ่มประสิทธิภาพในการพยากรณ์ของแบบจำลองการเรียนรู้เชิงลึกให้ประสิทธิภาพในการพยากรณ์สูงขึ้น

ในงานวิจัยของ Al-Thelaya et al. (2018) ได้ทำการวิเคราะห์และเปรียบเทียบความสามารถในการพยากรณ์ของแบบจำลองหลายชนิด เช่น LSTM, Bi-LSTM, และ Stack-LSTM โดยใช้ ANN เป็นเกณฑ์มาตรฐานในการเปรียบเทียบ โดยศึกษาการพยากรณ์ราคาปิดของดัชนี S&P 500 ระหว่างวันที่ 1 มกราคม ค.ศ. 2010 ถึง 30 พฤศจิกายน ค.ศ. 2017 การศึกษาแบ่งออกเป็นการพยากรณ์ในสองระยะเวลา คือ 1) ระยะสั้น (1 วันข้างหน้า) และ 2) ระยะยาว (30 วันข้างหน้า) โดยใช้ข้อมูลราคาปิดย้อนหลัง 10 วันเป็นตัวแปรต้น การประเมินผลใช้ตัวชี้วัดสามตัว ได้แก่ MAE, RMSE และ R-square ผลลัพธ์แสดงให้เห็นว่าการพยากรณ์ระยะสั้นมีความแม่นยำดีกว่าการพยากรณ์ระยะยาว โดยในกลุ่มแบบจำลองที่ศึกษาพบว่า Bi-LSTM มีประสิทธิภาพสูงสุด ส่วน Bi-LSTM และ Stack-LSTM มีความสามารถในการพยากรณ์ทั้งในระยะสั้นและระยะยาวเหนือกว่า LSTM และ ANN

ในงานวิจัยของ Liu et al. (2019) ได้ดำเนินการเปรียบเทียบประสิทธิภาพในการพยากรณ์ของแบบจำลอง Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), และแบบจำลองผสม GRU-LSTM สำหรับการคาดการณ์ราคากลางหุ้นสองตัว ได้แก่ Dalian Thermal Power Co. Ltd (600719) และ Dalian Friendship Co. Ltd (000679) ในช่วงระหว่างวันที่ 15 มีนาคม ค.ศ. 2008 ถึง 15 มีนาคม ค.ศ. 2018 การศึกษานี้ใช้ค่า Root Mean Square Error (RMSE) เป็นตัวชี้วัดประสิทธิภาพ โดยใช้ข้อมูลตัวแปรต้นประกอบด้วย ราคาเปิด ราคาปิด ราคาต่ำสุด ราคาสูงสุด และปริมาณการซื้อขายในช่วง 10 วันก่อนหน้า ส่วนตัวแปรตามคือราคาปิดของวันถัดไปแบบจำลองทั้งสามมีโครงสร้างที่ประกอบด้วยชั้นการเรียนรู้ 4 ชั้น ได้แก่ 3 ชั้นซ่อนและ 1 ชั้นนำออกผล ผลการศึกษาแสดงให้เห็นว่าแบบจำลองผสม GRU-LSTM มีค่า RMSE ต่ำที่สุด นอกจากนี้ยังพบว่า เวลาในการฝึกฝนของแบบจำลอง GRU-LSTM ใกล้เคียงกับแบบจำลอง GRU ซึ่งมีเวลาในการฝึกฝนต่ำที่สุด ดังนั้น จึงสามารถสรุปได้ว่าแบบจำลอง GRU-LSTM มีประสิทธิภาพในการพยากรณ์ที่เหนือกว่าแบบจำลอง GRU อย่างมีนัยสำคัญ

Lu et al. (2021) ได้เสนอแบบจำลองใหม่ที่ชื่อว่า CNN-Bi-LSTM-AM สำหรับการพยากรณ์ราคาปิดหุ้นในวันถัดไปของดัชนี Shanghai Composite Index โดยใช้ข้อมูลในช่วงวันที่ 1 กรกฎาคม ค.ศ. 1991 ถึง 30 มิถุนายน ค.ศ. 2020 งานวิจัยนี้ได้ทำการเปรียบเทียบประสิทธิภาพของแบบจำลอง CNN-Bi-LSTM-AM กับแบบจำลองอื่น ๆ อีก 8 ตัว ได้แก่ MLP, CNN, RNN, LSTM, Bi-LSTM, CNN-LSTM, CNN-Bi-LSTM, และ Bi-LSTM-AM โดยใช้ตัวแปรต้นที่ประกอบด้วยราคาเปิด ราคาปิด ราคาสูงสุด ราคาต่ำสุด ปริมาณการซื้อขายของดัชนี ปริมาณการซื้อขายของทุกหลักทรัพย์ในตลาด ราคาที่เปลี่ยนไป และเบอร์เซ็นต์ที่เปลี่ยนไปของหุ้น ส่วนตัวแปรตามคือราคาปิดของวันถัดไป ในแบบจำลอง CNN-Bi-LSTM-AM นี้, CNN ถูกใช้เพื่อสกัดลักษณะเฉพาะจากข้อมูลนำเข้า, Bi-LSTM ถูกใช้ในการเรียนรู้และคาดการณ์จากลักษณะเฉพาะที่สกัดออกมา, และ AM ถูกใช้เพื่อประเมินอิทธิพลของราคาปิดในอดีตเพื่อเพิ่มความแม่นยำของการพยากรณ์ งานวิจัยนี้ใช้ MAE เป็นฟังก์ชันการสูญเสีย (Loss Function) และใช้ RMSE, MAE, และ R-square เป็นตัววัดประสิทธิภาพในการพยากรณ์ ผลการศึกษาแสดงให้เห็นว่าแบบจำลอง CNN-Bi-LSTM-AM มีประสิทธิภาพสูงสุดในกลุ่มตัวแบบที่ศึกษาโดยมีค่า RMSE และ MAE ต่ำที่สุด และ R-square สูงที่สุด เมื่อเปรียบเทียบกับแบบจำลองอื่น ๆ

Hossain et al. (2018) ได้นำเสนอแบบจำลองการพยากรณ์ Deep Recurrent Neural Network และได้ทำการวิเคราะห์และเปรียบเทียบความสามารถในการพยากรณ์ของแบบจำลอง LSTM และ GRU กำหนดให้มีชั้นช่อง 1, 2 และ 3 ชั้น รวมถึงแบบจำลองโครงสร้างผสมระหว่าง LSTM และ GRU โดยใช้ข้อมูล S&P 500 ตั้งแต่ปี ค.ศ. 1950 ถึง ค.ศ. 2016 เป็นระยะเวลา 66 ปี งานศึกษานี้ใช้ MSE, MAE, และ MAPE เป็นตัววัดประสิทธิภาพในการพยากรณ์ จากผลการวิจัยพบว่า แบบจำลองโครงสร้างผสมระหว่าง LSTM และ GRU ประสิทธิภาพในการพยากรณ์สูงที่สุดเมื่อเปรียบเทียบกับเครือข่ายประสาทเทียมทั้งหมดที่ใช้ในชุดข้อมูลเดียวกันอย่างมีนัยสำคัญ นอกจากนี้การเพิ่มชั้นหรือทำให้เครือข่ายลึกขึ้นด้วยส่วนประกอบเดียวกันอาจทำให้เกิดการ overfitting และลดประสิทธิภาพลง

เกรว่อง (2566) ได้ทำการศึกษาและเปรียบเทียบประสิทธิภาพของแบบจำลอง LSTM, GRU, และแบบจำลองผสม GRU-LSTM ใน การพยากรณ์ราคาปิดของหุ้นสำหรับระยะเวลา 30 วันข้างหน้า เพื่อให้ครอบคลุมความหลากหลายของชุดข้อมูลอนุกรมเวลา, ผู้วิจัยได้เลือกหุ้นตามค่าความผันผวน (Beta) ที่สูงที่สุด ความผันผวนเมื่อเปรียบเทียบกับผลตอบแทนเฉลี่ยของตลาดหุ้นไทย โดยหุ้นสูก แบ่งออกเป็น 3 กลุ่มตามระดับความผันผวน: กลุ่มความผันผวนต่ำ ได้แก่ ADVANCE และ BDMS กลุ่มความผันผวนปานกลาง ได้แก่ CPALL และ MINT และกลุ่มความผันผวนสูง ได้แก่ DELTA และ SAWAD การประเมินประสิทธิภาพของแบบจำลองใช้ตัวชี้วัด RMSE, MAPE, อัตราการลดลงของ RMSE (Reduction Rate of RMSE), และระยะเวลาในการฝึกฝนแบบจำลอง ผลการศึกษาพบว่า สำหรับกลุ่มหุ้นที่มีความผันผวนต่ำ หุ้น ADVANCE มีแบบจำลอง GRU-LSTM ให้ประสิทธิภาพดีที่สุด ขณะที่หุ้น BDMS แบบจำลอง RNN ให้ผลลัพธ์ที่ดีที่สุด ในกลุ่มหุ้นที่มีความผันผวนปานกลาง หุ้น CPALL มีแบบจำลอง GRU-LSTM ที่ให้ประสิทธิภาพสูงสุด ขณะที่หุ้น MINT แบบจำลอง GRU มีประสิทธิภาพดีที่สุด และสำหรับกลุ่มหุ้นที่มีความผันผวนสูง หุ้น SAWAD แบบจำลอง LSTM มีผลลัพธ์ที่ดีที่สุด ขณะที่หุ้น DELTA แบบจำลอง RNN ให้ผลลัพธ์ที่ดีที่สุด ผลการศึกษาแสดงให้เห็นว่า แบบจำลอง GRU-LSTM ไม่ได้มีความสามารถในการพยากรณ์ที่เหนือกว่าแบบจำลองการเรียนรู้เชิงลึกที่ใช้โครงสร้างเดียวในทุกรูปแบบ

จากการวิเคราะห์งานวิจัยที่ศึกษาเกี่ยวกับการพยากรณ์ราคาปิดหุ้นส่วนใหญ่ทำการศึกษาการพยากรณ์ราคาปิดหุ้นในวันถัดไปเท่านั้น มีเพียง 2 งานวิจัยที่ได้ทำการศึกษาการพยากรณ์ราคาปิดหุ้นระยะยาว ซึ่งพบว่าการพยากรณ์ราคาปิดหุ้นในระยะยาวนั้นจะมีประสิทธิภาพลดลงเมื่อเปรียบเทียบกับการพยากรณ์ในระยะสั้น และในงานวิจัยปัจจุบันเทคนิคในการเพิ่มประสิทธิภาพการพยากรณ์ให้กับตัวแบบให้ความสนใจในด้านการสมมติว่างโครงสร้างตัวแบบการเรียนรู้เชิงลึก เช่น งานวิจัยที่ได้ศึกษาโครงสร้างผสาน CNN-Bi-LSTM-AM หรือ ตัวแบบโครงสร้างผสาน GRU-LSTM ซึ่งจากผลงานวิจัยพบว่าตัวแบบโครงสร้างผสานช่วยเพิ่มประสิทธิภาพการพยากรณ์จากตัวแบบปกติและและให้ประสิทธิภาพที่ดีกว่าการเพิ่มชั้นหรือทำให้เครือข่ายลึกขึ้นด้วยเครือข่ายเดียวกัน งานวิจัยนี้จึงต้องการมุ่งเน้นการพัฒนาตัวแบบผสาน 2 แบบ คือ Cascaded neural network และ Stacked layers in neural network ของแบบจำลองการเรียนรู้เชิงลึกที่มีประสิทธิภาพในการพยากรณ์ราคาปิดหุ้น ซึ่งจะพยากรณ์ราคาปิดทั้งหมด 2 ระยะเวลาด้วยกัน ได้แก่ ระยะสั้น 7 วัน และระยะยาว 30 วัน โดยจำนวนชั้นช่องของแบบจำลองการเรียนรู้เชิงลึกที่จะนำมาผสานแบบ Stacked layers in neural network ทั้ง 2 แบบจำลองมีจำนวนเท่ากัน ต่างจากงานวิจัยของ Liu et al. (2019) ที่มีจำนวนชั้นของ LSTM layer และ GRU layer ไม่เท่ากัน เนื่องจากตัวแบบ GRU และตัวแบบ LSTM มีข้อดีที่แตกต่างกัน คือตัวแบบ GRU มีลักษณะที่ใช้เวลาในการฝึกฝนตัวแบบน้อยกว่าตัวแบบ LSTM แต่ตัวแบบ LSTM เหมาะสำหรับข้อมูลที่มีขนาดใหญ่กว่า

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 3

วิธีการดำเนินการวิจัย

การศึกษาวิจัยนี้เป็นการศึกษาและเปรียบเทียบประสิทธิภาพการพยากรณ์ของแบบจำลองการเรียนรู้เชิงลึกแบบผสมของ RNN-LSTM, LSTM-GRU และ RNN-GRU ซึ่งจะมีการผสมหั้งหมด 2 แบบ คือ การใช้โครงสร้างแบบผสม (Stacked layers) และเครือข่ายประสาทเทียมแบบซ้อนกัน (Cascaded neural network) โดยมีขั้นตอนการวิจัยดังนี้

1. ศึกษาทฤษฎีและกรอบแนวคิดที่เกี่ยวข้อง
2. การเก็บรวบรวมข้อมูล
3. การจัดเตรียมข้อมูลและสร้างแบบจำลองสำหรับการพยากรณ์
4. การวัดประสิทธิภาพการพยากรณ์
5. สรุปผลและวิเคราะห์ผลงานวิจัย

3.1 ศึกษาทฤษฎีและกรอบแนวคิดที่เกี่ยวข้อง

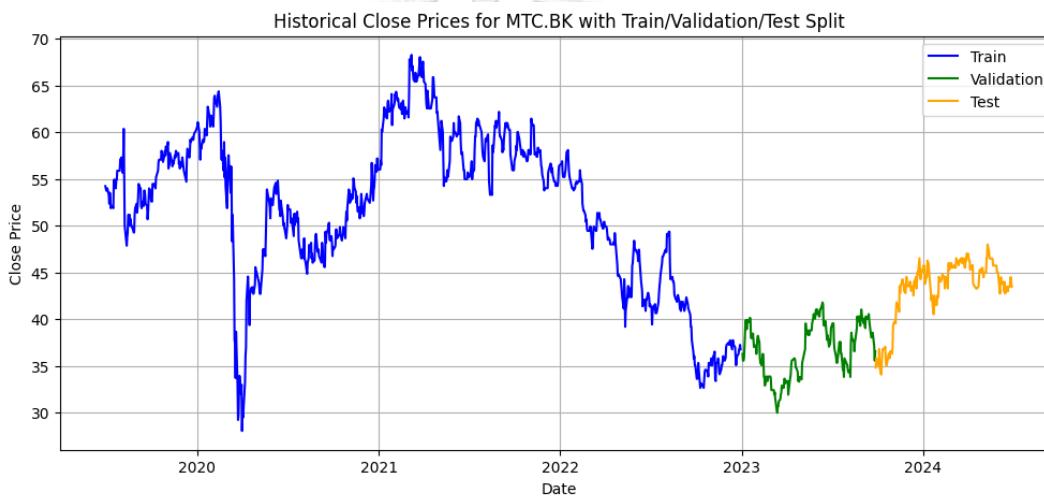
ตั้งหัวข้อทฤษฎีและงานวิจัยที่เกี่ยวข้อง ในบทที่ 2

3.2 การเก็บรวบรวมข้อมูล

ข้อมูลที่ใช้ในงานวิจัยนี้เป็นข้อมูลทุติยภูมิที่เก็บรวบรวมจากเว็บไซต์ Yahoo Finance (<https://finance.yahoo.com/>) เป็นข้อมูลของหุ้นย้อนหลังเป็นระยะเวลา 5 ปีตั้งแต่วันที่ 1 กรกฎาคม พ.ศ.2563 ถึงวันที่ 30 มิถุนายน พ.ศ.2567 เป็นจำนวนตัวอย่างทั้งหมด 1,211 ตัวอย่าง โดยเลือกเก็บข้อมูลหุ้นทั้งหมด 15 หุ้น จากตลาดหลักทรัพย์แห่งประเทศไทย (SET) โดยแบ่งออกเป็น 5 อุตสาหกรรมที่มีลักษณะการซื้อขายและลงของหุ้นที่แตกต่างกัน ได้แก่ กลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และกลุ่มเทคโนโลยี และในทุกกลุ่มอุตสาหกรรมแบ่งออกเป็น 3 กลุ่มตามค่าความผันผวน (Beta) โดยมีรายละเอียดดังตารางที่ 1 ในแต่ละหุ้นทำการเก็บข้อมูล 5 ตัวแปร ได้แก่ ราคาเปิด ราคาปิด ราคาน้ำสูงสุด ราคาน้ำต่ำสุด และปริมาณการซื้อขายในแต่ละวัน ซึ่งข้อมูลที่เก็บมา มีลักษณะเป็นรายวัน ผู้วิจัยจะนำ 5 ตัวแปรจากที่กล่าวไปข้างต้นมาใช้ในการพยากรณ์ราคาปิด โดยลักษณะอนุกรมเวลาของหุ้นทั้ง 15 หุ้นที่ถูกนำมาศึกษาในงานวิจัยมีลักษณะดังภาพที่ 9 ถึงภาพที่ 23



ภาพที่ 9 ราคาปิดรายวันของหุ้น XPG ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 10 ราคาปิดรายวันของหุ้น MTC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 11 ราคาปิดรายวันของหุ้น KBANK ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



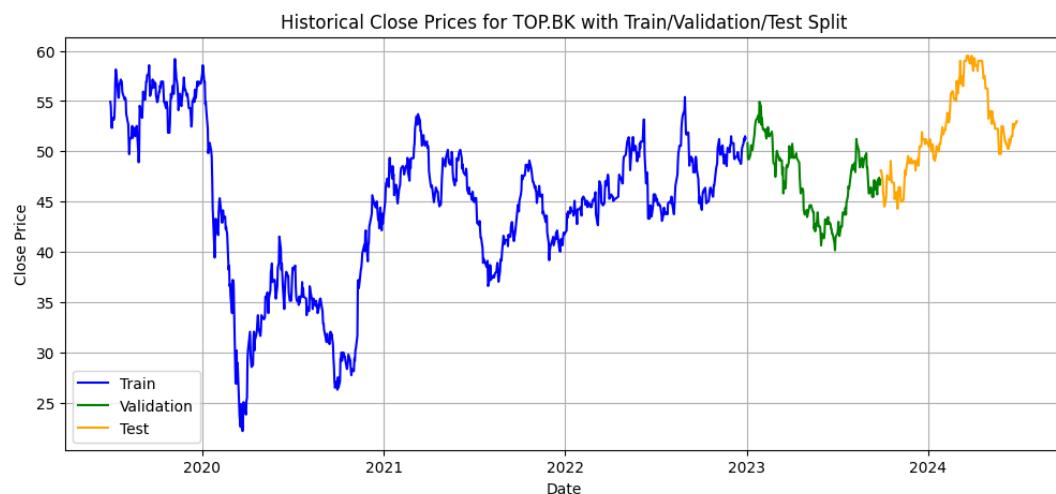
ภาพที่ 12 ราคาปิดรายวันของหุ้น PTTGC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 13 ราคาปิดรายวันของหุ้น AMC ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 14 ราคาปิดรายวันของหุ้น STANLY ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 15 ราคาปิดรายวันของหุ้น TOP ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



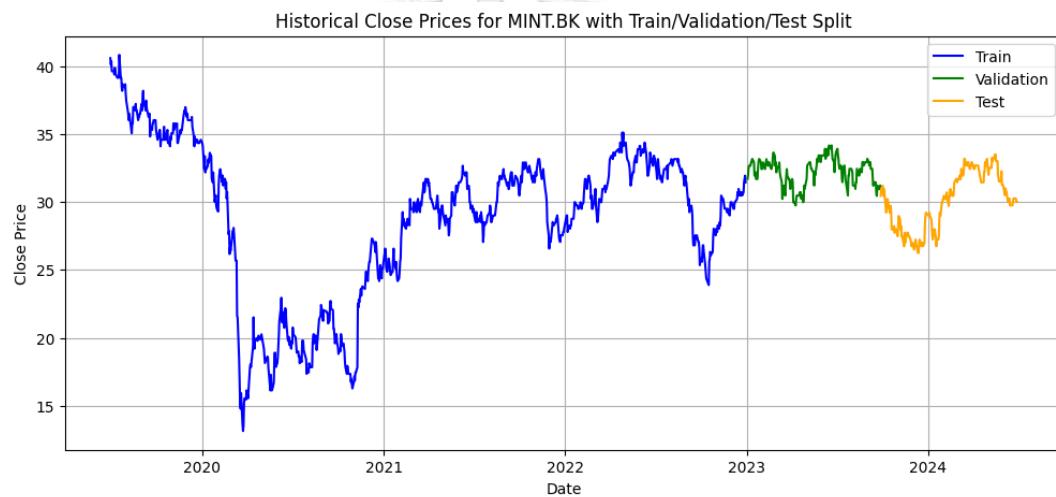
ภาพที่ 16 ราคาปิดรายวันของหุ้น BCP ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



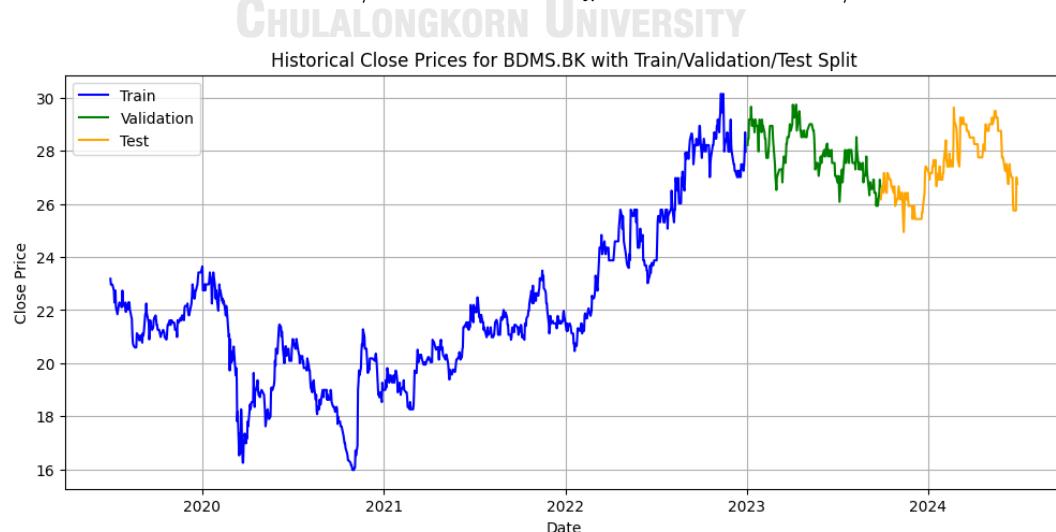
ภาพที่ 17 ราคาปิดรายวันของหุ้น PTT ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 18 ราคาปิดรายวันของหุ้น RCL ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



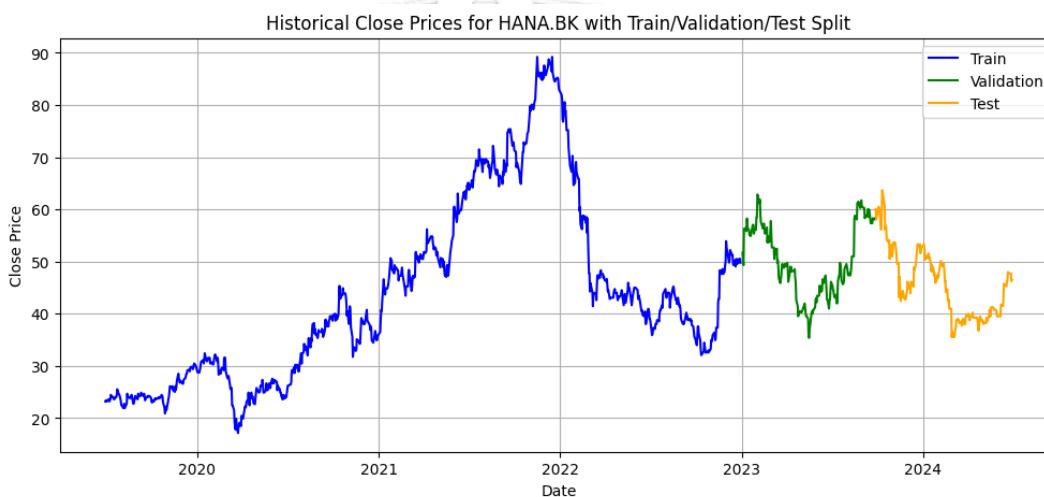
ภาพที่ 19 ราคาปิดรายวันของหุ้น MINT ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 20 ราคาปิดรายวันของหุ้น BDMS ตั้งแต่ 1 กรกฎาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 21 ราคาปิดรายวันของหุ้น SMT ตั้งแต่ 1 กรกฏาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 22 ราคาปิดรายวันของหุ้น HANA ตั้งแต่ 1 กรกฏาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567



ภาพที่ 23 ราคาปิดรายวันของหุ้น TRUE ตั้งแต่ 1 กรกฏาคม พ.ศ.2563 ถึง 30 มิถุนายน พ.ศ.2567

3.3 การจัดเตรียมข้อมูลและสร้างแบบจำลองสำหรับการพยากรณ์

3.3.1 การจัดเตรียมข้อมูล

ขั้นตอนการปรับขนาดข้อมูลด้วย Min-Max Normalization

ขั้นตอนนี้เป็นการปรับข้อมูลตัวแปรต้นทั้ง 5 ตัวแปร ได้แก่ ราคาเปิด ราคาปิด ราคาสูงสุด ราคาต่ำสุด และปริมาณการซื้อขาย ให้อยู่ในช่วง 0 ถึง 1 เนื่องจากตัวแปรทั้ง 5 มีขนาดแตกต่างกันอย่างมาก เพื่อให้แบบจำลองการเรียนรู้เชิงลึกสามารถเรียนรู้ลักษณะของข้อมูลได้อย่างมีประสิทธิภาพ การปรับข้อมูลแต่ละตัวแปรให้อยู่ในขนาดที่ใกล้เคียงกันจึงเป็นสิ่งที่จำเป็น สามารถคำนวณได้ดังสมการที่ (22)

$$X_t^{scaled} = \frac{X_t - \min(X)}{\max(X) - \min(X)} \quad (22)$$

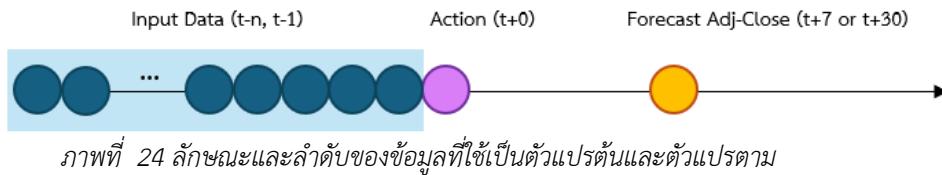
โดย X_t คือ ค่าของข้อมูล

$\min(X)$ คือ ค่าต่ำสุดของข้อมูล

$\max(X)$ คือ ค่าสูงสุดของข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

แปลงข้อมูลนำเข้าให้อยู่ในรูปแบบอาเรย์



ขั้นตอนนี้ทำการแปลงตัวแปรทั้ง 5 ได้แก่ ราคาเปิด ราคาปิด ราคาน้ำสูงสุด ราคาน้ำต่ำสุด และปริมาณการซื้อขายในแต่ละวัน ให้อยู่ในรูปแบบอาเรย์สามมิติ โดยใช้ข้อมูลปัจจุบันและข้อมูลย้อนหลังของทั้ง 5 ตัวแปรเพื่อพยากรณ์ราคาปิดหุ้นทั้งหมด 2 ระยะเวลาด้วยกัน ได้แก่ ระยะสั้น 7 วันข้างหน้า และระยะยาว 30 วันข้างหน้า ดังภาพที่ 24 การแปลงข้อมูลให้อยู่ในรูปแบบอาเรย์ในแต่ละหุ้นจะทำให้ได้ข้อมูลที่แบ่งออกเป็น 2 ส่วน ได้แก่

- ข้อมูลตัวแปรต้นที่จะใช้นำเข้าแบบจำลองมีลักษณะเป็นอาเรย์สามมิติ ที่มีมิติคือ [samples, timesteps, features] โดย
 - sample คือ จำนวนตัวอย่างข้อมูลในแต่ละวันตลอดระยะเวลา 5 ปี
 - timesteps คือ จุดเวลาที่เฉพาะเจาะจงในลำดับของข้อมูล ซึ่งจุดเวลาของข้อมูลที่ผู้วิจัยใช้เป็นรายวันอยู่แล้ว ค่านี้จึงเท่ากับ 1
 - features คือ ตัวแปรทั้ง 5 ตัวแปร ซึ่งแต่ละตัวแปรจะมีค่าของข้อมูลย้อนหลัง (lag) ของตัวแปร ตามความเหมาะสมของลักษณะของข้อมูลและแบบจำลองในทุกตัวแปร

โดยทั้งอาเรย์สามมิตินี้จะถูกเรียกว่าเหตุการณ์สำหรับใช้ในการฝึกฝนแบบจำลองเพื่อพยากรณ์ในราคาปิดหุ้นทั้งหมด 2 ระยะเวลา ได้แก่ ระยะสั้น 7 วัน และระยะยาว 30 วัน

- ข้อมูลตัวแปรตามโดยลักษณะของตัวแปรตามจะมีลักษณะเป็น อาเรย์สองมิติ ที่มีมิติคือ [samples , target] โดย
 - example คือ จำนวนตัวอย่างข้อมูลในแต่ละวันตลอดระยะเวลา 5 ปี
 - target คือ ค่าข้อมูลของตัวแปรราคาปิดในอีก 7 วัน และ 30 วันข้างหน้า เพื่อใช้เป็นตัวแปรเป้าหมายในการพยากรณ์ของแบบจำลอง

การแบ่งชุดข้อมูล

ข้อมูลที่ใช้ในงานวิจัยนี้ใช้ข้อมูลของหุ้นย้อนหลังเป็นระยะเวลา 5 ปีตั้งแต่วันที่ 1 กรกฎาคม พ.ศ.2563 ถึงวันที่ 30 มิถุนายน พ.ศ.2567 เป็นจำนวนตัวอย่างทั้งหมด 1,211 ตัวอย่าง โดยเลือกเก็บข้อมูลหุ้นทั้งหมด 15 หุ้น จากตลาดหลักทรัพย์แห่งประเทศไทย (SET) โดยแบ่งออกเป็น 5 กลุ่ม อุตสาหกรรม ที่มีลักษณะการขึ้นและลงของหุ้นที่แตกต่างกัน ได้แก่ กลุ่มธุรกิจการเงิน, กลุ่มสินค้าอุตสาหกรรม, กลุ่มทรัพยากร, กลุ่มบริการ และกลุ่มเทคโนโลยี และในทุกกลุ่มอุตสาหกรรมแบ่งออกเป็น 3 กลุ่มตามค่าความผันผวน (Beta) เพื่อการสร้างและทดสอบประสิทธิภาพของแบบจำลองการศึกษานี้ได้ทำการแบ่งข้อมูลออกเป็น 3 ส่วน ได้แก่

1. ข้อมูลชุดฝึกฝน (training data) ในอัตราส่วน 70% เพื่อใช้สำหรับการฝึกฝนในแบบจำลองการเรียนรู้เชิงลึก
2. ข้อมูลชุดตรวจสอบ (validation data) ในอัตราส่วน 15% เพื่อใช้สำหรับการทำไฮเปอร์พารามิเตอร์จูนนิ่ง เพื่อหาชุดไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุดในแบบจำลองการเรียนรู้เชิงลึกในแต่ละชนิด
3. ข้อมูลชุดทดสอบ (test data) ในอัตราส่วน 15 % เพื่อใช้ในการรายงานผลและวิเคราะห์ประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึกในแต่ละชนิด

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.3.2 สร้างแบบจำลองสำหรับการพยากรณ์

การศึกษานี้ทางผู้วิจัยได้สร้างแบบจำลองการเรียนรู้เชิงลึกแบบผสมผ่านการเขียนโปรแกรมในภาษา python ผ่าน library ของ TensorFlow และ Keras โดยการสร้างแบบจำลองแบบผสมในแต่ละชนิด ซึ่งมีขั้นตอนดังนี้

การออกแบบโครงสร้างของแบบจำลอง

โดยขั้นตอนแรกของการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบคือสร้างโครงสร้างของแบบจำลองที่สามารถทำการไฮเปอร์พารามิเตอร์จุนนิ่งได้ โดยแบบจำลองการเรียนรู้เชิงลึกแบบผสมที่ต้องการศึกษามีการผสม 2 รูปแบบ คือ Cascaded neural network และ Stacked layers โดยในแต่ละแบบจะมีการจับคู่แบบลง ได้แก่ RNN-LSTM, LSTM-RNN, RNN-GRU, GRU-RNN, LSTM-GRU และ GRU-LSTM รวม 12 แบบจำลอง

- แบบจำลองการเรียนรู้เชิงลึกแบบผสมแบบ Stacked layers

แบบจำลองการเรียนรู้เชิงลึกแบบผสมแบบ Stacked layers คือการพยากรณ์ที่เลือกใช้ชั้นซ่อนที่แตกต่างกันตั้งแต่ 2 ชั้นขึ้นไป แบบจำลองที่กล่าวมาข้างต้น ในแต่ละส่วนจะใช้แบบจำลองการเรียนรู้เชิงลึกที่มีทั้งหมด 3 ชั้น (ไม่รวมชั้นนำเข้าข้อมูล) ประกอบไปด้วย

1. ชั้นซ่อน (Hidden Layer) ซึ่งการวิจัยนี้ผู้วิจัยกำหนดให้มีชั้นซ่อนทั้งหมด 2 ชั้น
2. ชั้นนำออกข้อมูล (Output Layer) 1 ชั้น

ดังนั้นผู้วิจัยจะกำหนดให้แต่ละแบบจำลองทั้ง 6 แบบจำลองที่ต้องการศึกษานั้นมีโครงสร้างในชั้นซ่อนเพียง 2 ชั้นเท่านั้น โดยแต่ละชั้นจะใช้แบบจำลองที่ไม่เหมือนกัน ได้แก่

1. RNN-LSTM ใช้ Simple RNN layer ในชั้นซ่อนแรกและ LSTM layer ในชั้นซ่อนที่สอง
2. LSTM-RNN ใช้ LSTM layer ในชั้นซ่อนแรกและ Simple RNN layer ในชั้นซ่อนที่สอง
3. RNN-GRU ใช้ Simple RNN layer ในชั้นซ่อนแรกและ GRU layer ในชั้นซ่อนที่สอง
4. GRU-RNN ใช้ GRU layer ในชั้นซ่อนแรกและ Simple RNN layer ในชั้นซ่อนที่สอง
5. LSTM-GRU ใช้ LSTM layer ในชั้นซ่อนแรกและ GRU layer ในชั้นซ่อนที่สอง
6. GRU-LSTM ใช้ GRU layer ในชั้นซ่อนแรกและ LSTM layer ในชั้นซ่อนที่สอง

- แบบจำลองการเรียนรู้เชิงลึกแบบผสานแบบ Cascaded neural network

แบบจำลองการเรียนรู้เชิงลึกแบบผสานแบบ Cascaded neural network คือการทำการพยากรณ์โดยผลลัพธ์จากการพยากรณ์ครั้งแรกเรียกว่า “ค่าพยากรณ์ (Predicted value)” จากนั้นทำการคำนวณค่าความคลาดเคลื่อนจากความแตกต่างระหว่างค่าพยากรณ์และค่าจริง ค่าดังกล่าวเรียกว่า “ค่าพยากรณ์ความคลาดเคลื่อน (Predicted residuals)” แบบจำลองที่กล่าวมาข้างต้น ในแต่ละส่วนใช้แบบจำลองการเรียนรู้เชิงลึกที่มีทั้งหมด 2 ชั้น (ไม่รวมชั้นนำเข้าข้อมูล) ประกอบไปด้วย

1. ชั้นซ่อน (Hidden Layer) กำหนดให้มีชั้นซ่อนทั้งหมด 1 ชั้น
2. ชั้นนำออกข้อมูล (Output Layer) 1 ชั้น

หลังจากที่ได้ผลลัพธ์ทั้ง 2 ส่วนแล้วจะทำการนำ Predicted value มารวมกันกับ Predicted residuals เพื่อให้ได้ผลลัพธ์สุดท้าย (Final predicted) ดังนั้นผู้วิจัยจึงกำหนดให้แต่ละแบบจำลองการเรียนรู้เชิงลึกแบบผสานทั้ง 6 แบบจำลองที่ต้องการศึกษานั้นมี 2 ส่วนที่แตกต่างกัน ได้แก่

1. RNN-LSTM ใช้ Simple RNN layer ในการหา Predicted value และ LSTM layer ในการหา Predicted residuals
2. LSTM-RNN ใช้ LSTM layer ในการหา Predicted value และ Simple RNN layer ในการหา Predicted residuals
3. RNN-GRU ใช้ Simple RNN layer ในการหา Predicted value และ GRU layer ในการหา Predicted residuals
4. GRU-RNN ใช้ GRU layer ในการหา Predicted value และ Simple RNN layer ในการหา Predicted residuals
5. LSTM-GRU ใช้ LSTM layer ในการหา Predicted value และ GRU layer ในการหา Predicted residuals
6. GRU-LSTM ใช้ GRU layer ในการหา Predicted value และ LSTM layer ในการหา Predicted residuals

การกำหนดการตั้งค่าสำหรับไฮเปอร์พารามิเตอร์จุนนิ่ง

หลังจากทำการกำหนดขั้นในแต่ละแบบจำลองครบแล้ว ผู้วิจัยจะทำการเขียนโปรแกรมที่ทำให้แบบจำลองการเรียนรู้เชิงลึกสามารถทำการไฮเปอร์พารามิเตอร์จุนนิ่งได้ โดยมีการตั้งค่าสำหรับการทำไฮเปอร์พารามิเตอร์จุนนิ่งดังนี้

1. สำหรับแบบจำลองการเรียนรู้เชิงลึกแบบผสมในรูปแบบ Stacked layers ได้กำหนดการจุนไฮเปอร์พารามิเตอร์ทั้งหมด 5 ตัว ได้แก่ จำนวนโนหนดในชั้นซ่อนที่สอง อัตราการเรียนรู้ ค่าแอลเมบ์ดาของ L2 Regularization และ Beta_1 (โมเมนตัม) และแบบผสมแบบ Cascaded neural network ทำการกำหนดการจุนไฮเปอร์พารามิเตอร์ 4 ตัวแปร คือ ได้แก่ จำนวนโนหนดในชั้นซ่อนแรก อัตราการเรียนรู้ ค่าแอลเมบ์ดาของ L2 Regularization และ Beta_1 (โมเมนตัม) ซึ่งการกำหนดการตั้งค่าเพื่อใช้ในการทำไฮเปอร์พารามิเตอร์จุนนิ่งดังนี้
 - จำนวนโนหนดในชั้นซ่อน กำหนดไว้ตั้งแต่ 1 ถึง 100 หน่วย
 - อัตราการเรียนรู้ กำหนดไว้ตั้งแต่ 0.001 ถึง 0.1
 - ค่าแอลเมบ์ดาของ L2 Regularization ค่าที่กำหนดไว้คือ 0.001, 0.005, 0.01, 0.05 และ 0.1
 - และ Beta_1 (โมเมนตัม) ค่าที่กำหนดไว้คือ 0.7, 0.8 และ 0.9
2. กำหนด optimizer ที่ใช้ในงานวิจัยนี้คือ Adam เหตุผลเนื่องจากจะทำให้แบบจำลองถูเข้าสู่ค่าที่ถูกต้องได้ไวยิ่งขึ้นลดการสั่นของการปรับค่าน้ำหนักจากขั้นตอน Gradient descent รวมถึงเป็น optimizer ที่มีการลด Learning Rate ลงเมื่อ gradient เข้าใกล้จุดต่ำสุดซึ่งทำให้แบบจำลองสามารถทำการเรียนรู้ได้มากขึ้นส่งผลให้แบบจำลองประสิทธิภาพมากขึ้น
3. Loss function ที่ผู้วิจัยเลือกใช้คือ RMSE เนื่องจากเป็นตัววัดประสิทธิภาพที่มีความไว กับค่าที่มีความผิดปกติสูงหรือความผันผวนสูง จากข้อมูลของผู้วิจัยที่ทำการศึกษาราคา ปิดของหุ้นที่มีความผันผวนสูงในแต่ละวัน ตัววัด RMSE จึงเหมาะสมที่จะนำมาใช้เพื่อรับมือกับราคากลางที่เกิดการเปลี่ยนแปลงราคาที่มีการเพิ่มขึ้นและลดลงมากกว่าปกติ
4. ตัววัดประสิทธิภาพที่จะแสดงผลเวลาดูประสิทธิภาพของแบบจำลองทั้งหมด คือ ค่า RMSE

กระบวนการทำไฮเปอร์พารามิเตอร์จุนนิ่ง

ในขั้นตอนนี้ได้ทำการกำหนด search engine สำหรับการหาค่าพารามิเตอร์ที่เหมาะสมที่สุด ในแต่ละแบบจำลองซึ่ง search engine ที่ผู้วิจัยเลือกใช้คือ randomized search เนื่องจาก ทรัพยากรคอมพิวเตอร์ของผู้วิจัยมีจำกัด การเลือกใช้ randomized search ทำให้เกิดการเลือกค่าพารามิเตอร์ที่มีความหลากหลายกว่า Grid search ในจำนวนรอบที่จำกัด ในเบื้องต้นผู้วิจัยจะทำการกำหนดการตั้งค่าของ randomized search ไว้ดังนี้

1. n_iter อยู่ที่ 100 ชุด เพราะฉะนั้นจะเกิดการทดลองชุดไฮเปอร์พารามิเตอร์ 100 ชุดที่แตกต่างกันในแต่ละแบบจำลองการเรียนรู้เชิงลึกแบบผสม
2. กำหนด objective ในการ search คือ “val_loss” เนื่องจาก loss function ของแบบจำลองแต่ละชนิดคือ RMSE เพราะฉะนั้นผู้วิจัยจะทำการเลือกชุดไฮเปอร์พารามิเตอร์ที่ให้ค่า val_loss ต่ำที่สุด ความหมายก็คือไฮเปอร์พารามิเตอร์ชุดที่พยากรณ์ข้อมูลตรวจสอบได้ดีที่สุดในจำนวน 100 ชุดไฮเปอร์พารามิเตอร์ที่ผู้วิจัยได้ทดสอบ
3. กำหนดเงื่อนไขในการค้นหาของ Randomized search ดังนี้
 - EPOCH ใน การเรียนรู้ในแต่ละชุดไฮเปอร์พารามิเตอร์ใช้จำนวน 100 รอบ
 - batch_size กำหนดให้เท่ากับ 64 ตัวอย่าง
 - Call_back โดยการใช้เทคนิค Early stopping เข้ามาร่วมในการค้นหา ด้วยการใช้ Early stopping ทำให้แต่ละชุดไฮเปอร์พารามิเตอร์นั้นได้ค่า val_loss ในรอบที่ได้ทำการเรียนรู้ที่ต่ำที่สุดออกจากเปรียบเทียบกันเพื่อให้ได้ชุดไฮเปอร์พารามิเตอร์ที่มีประสิทธิภาพในการพยากรณ์สูงที่สุด.

โดยการตั้งค่าของแต่ละแบบจำลองการเรียนรู้เชิงลึกแบบผสมในการค้นหาชุดไฮเปอร์พารามิเตอร์ที่ให้ค่า val_loss ต่ำที่สุดในแต่ละแบบจำลอง

นำชุดไอกิริมีเตอร์ที่เหมาะสมที่สุดมาทำการฝึกแบบจำลองอีกรั้ง

หลังจากผ่านกระบวนการไอกิริมีเตอร์จนนิ่งผู้วิจัยจะได้ชุดของไอกิริมีเตอร์ที่ทำให้ค่า “val_loss” ต่ำที่สุดในแต่ละแบบจำลองการเรียนรู้เชิงลึกแบบสม จากนั้นผู้วิจัยจะนำชุดไอกิริมีเตอร์เหล่านี้ไปทำการฝึกเรียนรู้ในแบบจำลองอีกรั้งหนึ่ง เพื่อให้ประสิทธิภาพในการพยากรณ์สูงขึ้น เนื่องจากตอนทำผู้วิจัยได้เรียนรู้เป็นจำนวนเพียงแค่ 100 EPOCH ซึ่งอาจไม่เพียงพอหากอยากรู้ว่าไอกิริมีเตอร์ที่มีประสิทธิภาพในการพยากรณ์สูงที่สุด ดังนั้นผู้วิจัยจึงเริ่มกระบวนการฝึกแบบจำลองอีกรั้งแต่จะเปลี่ยนการตั้งค่าการฝึกของแบบจำลองดังนี้

- EPOCH ใน การเรียนรู้เท่ากับ 150
- batch_size กำหนดให้ค่าเท่ากับ 64
- ใช้เทคนิค call back จาก Early stopping เมื่อันเดิม เพื่อให้ผลลัพธ์สุดท้ายผู้วิจัยจะได้แบบจำลองการเรียนรู้เชิงลึกที่มีประสิทธิภาพในการพยากรณ์สูงที่สุดเพื่อไปวัดประสิทธิภาพในขั้นตอนสุดท้ายต่อไป

สุดท้ายผู้วิจัยจะได้แบบจำลองที่มีประสิทธิภาพในการพยากรณ์สูงที่สุดในทั้ง 12 แบบจำลอง การเรียนรู้เชิงลึกแบบสมที่งานวิจัยนี้ต้องการจะศึกษา เพื่อใช้สำหรับการวัดประสิทธิภาพการพยากรณ์ราคาปิดหุ้นในอีก 7 วัน และ 30 วันข้างหน้าและเปรียบเทียบประสิทธิภาพของแบบจำลองที่แตกต่างกัน ซึ่งจะพูดถึงในหัวข้อถัดไป

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.4 การวัดประสิทธิภาพการพยากรณ์

ทางผู้วิจัยจะทำการวัดประสิทธิภาพการพยากรณ์ข้อมูลที่ต้องการจะศึกษา โดยเครื่องมือที่จะนำมาใช้ในการวิเคราะห์มีทั้งหมด 3 ตัวชี้วัด ดังนี้

- ค่ารากที่สองของค่าเฉลีุ่ความคลาดเคลื่อนกำลังสอง (Root Mean Square Error: RMSE) เป็นการวัดความคลาดเคลื่อนเพื่อเปรียบเทียบความแตกต่างระหว่างค่าจากการพยากรณ์ และค่าจริงเฉลี่ยกำลังสอง สามารถคำนวณได้ดังสมการ (23)

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{N}} \quad (23)$$

โดย Y_t คือ ค่าจริง ณ เวลาที่ t

\hat{Y}_t คือ ค่าพยากรณ์ ณ เวลาที่ t

N คือ จำนวนข้อมูลที่นำมาพิจารณา

- ค่าเฉลี่ยของเบอร์เซ็นต์ความผิดพลาดสัมบูรณ์ (Mean Absolute Percentage Error: MAPE) เป็นวิธีวัดความแม่นยำจากการคำนวณเบอร์เซ็นต์ความผิดพลาดในการพยากรณ์ สามารถคำนวณได้ดังสมการ (24)

$$MAPE = \frac{\sum_{t=1}^n |(Y_t - \hat{Y}_t)/Y_t|}{N} \times 100\% \quad (24)$$

โดย Y_t คือ ค่าจริง ณ เวลาที่ t

\hat{Y}_t คือ ค่าพยากรณ์ ณ เวลาที่ t

N คือ จำนวนข้อมูลที่นำมาพิจารณา

- ระยะเวลาในการฝึกฝนแบบจำลอง ผู้วิจัยยังต้องการที่จะเปรียบเทียบเวลาที่ใช้ในการฝึกฝนแบบจำลองในหุ่นแต่ละตัว ดังนั้นผู้วิจัยจะทำการบันทึกเวลาในการฝึกฝน ซึ่งการวัดผลด้วยระยะเวลาในการฝึกฝนแบบจำลอง สามารถสะท้อนต้นทุนด้านทรัพยากรและพลังงาน ซึ่งมีผลต่อการตัดสินใจว่าควรใช้แบบจำลองนี้หรือไม่

ถ้าหาก RMSE และ MAPE ต่ำแสดงว่าค่าพยากรณ์มีความใกล้เคียงกับคาดการณ์ ซึ่งหมายถึงแบบจำลองมีประสิทธิภาพในการพยากรณ์สูง นอกจากนี้หากระยะเวลาในการฝึกฝนแบบจำลองน้อยแสดงว่าต้นทุนการใช้ทรัพยากรคอมพิวเตอร์และเวลาค่อนข้างน้อย

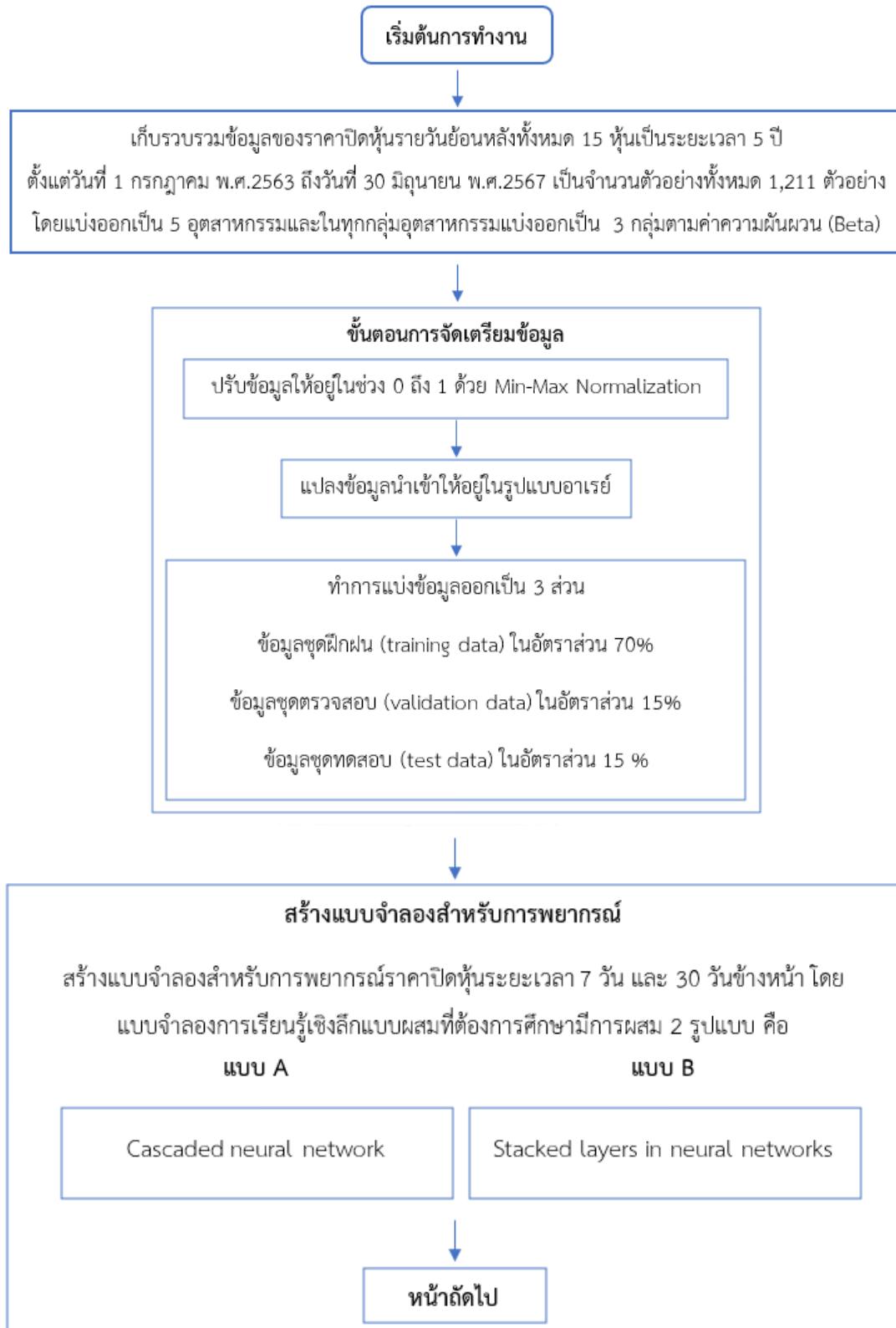
3.5 สรุปผลและวิเคราะห์ผลงานวิจัย

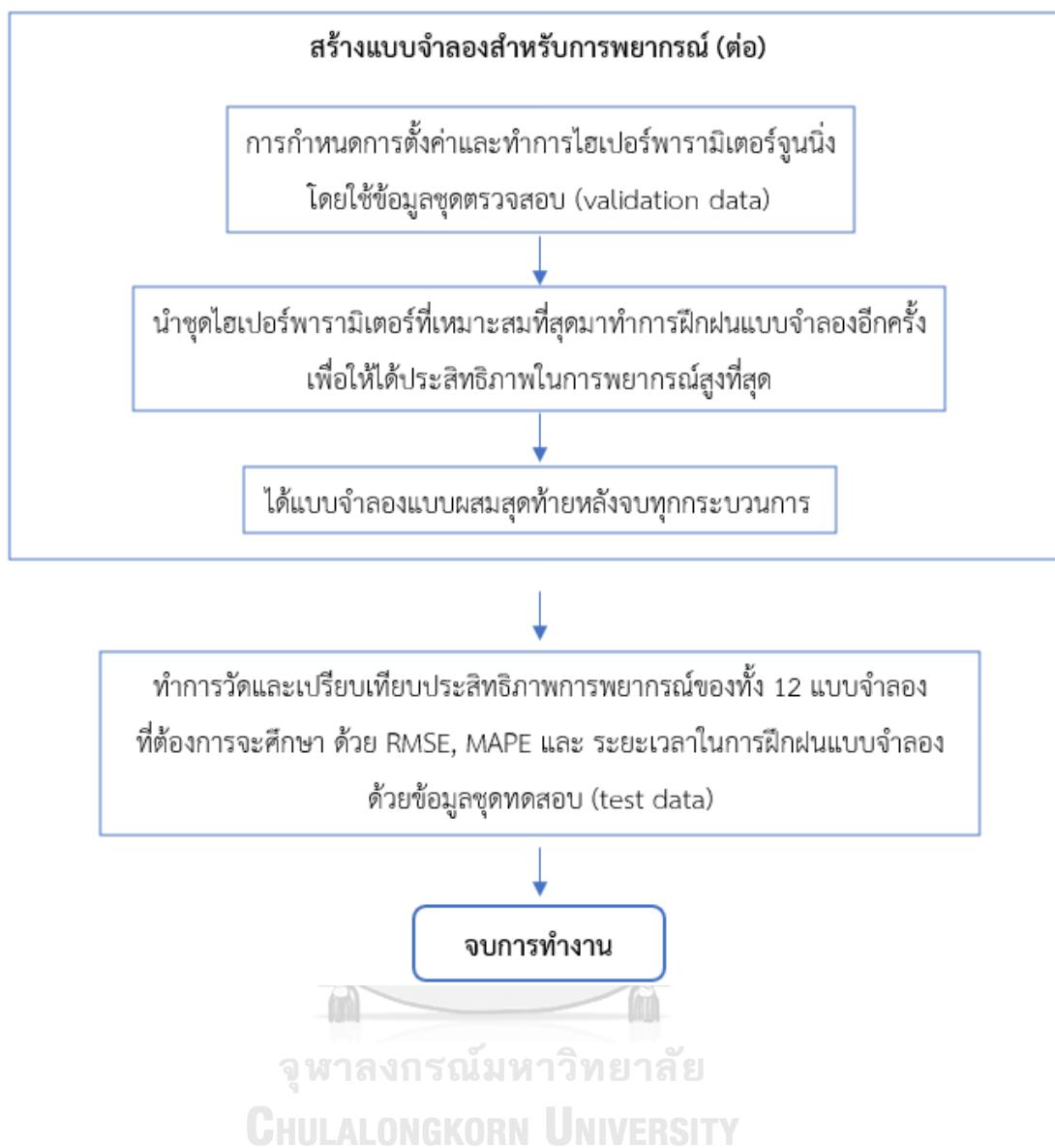
ผู้วิจัยจะทำการนำเสนอผ่านกราฟเปรียบเทียบค่าจริง-ค่าพยากรณ์ประกอบกับการใช้ตารางสรุปผล โดยในตารางจะประกอบไปด้วยแบบจำลองที่เราใช้กับข้อมูลทุกชุด พร้อมรายงานผล RMSE, MAPE และระยะเวลาในการฝึกฝนแบบจำลอง ของทุกแบบจำลอง เพื่อตรวจสอบประสิทธิภาพแต่ละวิธีที่ใช้กับข้อมูลราคาปิดหันอุตสาหกรรมและความผันผวนที่แตกต่างกัน นอกจากนี้ เพื่อดูว่าในภาพรวมรูปแบบการใช้ Cascaded neural network มีประสิทธิภาพที่แตกต่างกับการใช้ Stacked layers หรือไม่ และการสลับลำดับของแบบจำลองที่ใช้ใน Cascaded neural network และ Stacked layers ส่งผลกระทบต่อประสิทธิภาพของแบบจำลองการเรียนรู้เชิงลึกแบบสมหรือไม่ ผู้วิจัยจึงนำ Wilcoxon's Signed Rank test ซึ่งเป็นสถิติแบบอันอันดับ (non-parametric test) ที่ใช้เปรียบเทียบค่ามัธยฐานของตัวอย่างที่มีการจับคู่กัน (paired samples) เพื่อตรวจสอบว่ามีความแตกต่างอย่างมีนัยสำคัญหรือไม่ โดยใช้กับข้อมูลที่มีการแจกแจงที่ไม่จำเป็นต้องเป็นแบบปกติ (non-normal distribution) มาทำการทดสอบความแตกต่างของประสิทธิภาพ เพื่อประกอบการวิเคราะห์และเปรียบเทียบ โดยมีการตั้งสมมติฐานว่างและสมมติฐานทางเลือกดังนี้

สมมติฐานว่าง (H_0): มัธยฐานของความแตกต่างระหว่างค่าที่จับคู่กันเท่ากับศูนย์ หรือไม่มีความแตกต่างอย่างมีนัยสำคัญระหว่างสองกลุ่ม

สมมติฐานทางเลือก (H_1): มัธยฐานของความแตกต่างระหว่างค่าที่จับคู่กัน ไม่เท่ากับ ศูนย์ หรือมีความแตกต่างอย่างมีนัยสำคัญระหว่างสองกลุ่ม

3.6 แผนผังการดำเนินงานวิจัย





บทที่ 4

ผลการวิจัย

บทนี้จะกล่าวถึงผลการวิจัยของการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบผสม เพื่อวัดประสิทธิภาพและระยะเวลาที่ใช้ในการสร้างแบบจำลองของ Hybrid Deep Learning Models ในรูปแบบ Stacked layer และ Cascaded neural network ในการพยากรณ์ระยะสั้น 7 วัน และระยะยาว 30 วัน ในส่วนของขั้นตอนการวัดประสิทธิภาพแบบจำลองกับข้อมูลชุดทดสอบของหุ้นแต่ละตัวเราจะทำการรายงานผลโดยแบ่งตามกลุ่มความผันผวน หรือ Beta โดยแบ่งออกเป็นห้องหมวด 3 กลุ่ม ดังนี้

- กลุ่มของหุ้นที่มีความผันผวนต่ำ (Beta ต่ำ) ได้แก่ KBANK, STANLY, PTT, BDMS, TRUE
- กลุ่มของหุ้นที่มีความผันผวนปานกลาง (Beta ปานกลาง) ได้แก่ MTC, AMC, BCP, MINT, HANA
- กลุ่มของหุ้นที่มีความผันผวนสูง (Beta สูง) เช่น XPG, PTTGC, TOP, RCL, SMT

ซึ่งหุ้นทุกตัวในแต่ละกลุ่มความผันผวนจะถูกนำมาแสดงผลการวิจัยของแบบจำลองการเรียนรู้เชิงลึกแบบผสมที่สนใจวิจัยโดยแบ่งออกเป็น 2 ส่วนดังนี้

1. ผลการวิจัยในการสร้างตัวแบบจำลองการเรียนรู้เชิงลึกเพื่อใช้วัดประสิทธิภาพโมเดลจากการเรียนรู้ข้อมูลชุดฝึกฝน (training set) เพื่อปรับปรุงประสิทธิภาพของแบบจำลองโดยค้นหาค่าไฮเปอร์พารามิเตอร์ที่เหมาะสมที่สุด ซึ่งให้ค่าความคลาดเคลื่อนเฉลี่ยกำลังสอง (RMSE) ต่ำที่สุดจากการประเมินด้วยข้อมูลชุดตรวจสอบ (validation set)
2. ผลการวิจัยในขั้นตอนการวัดประสิทธิภาพตัวแบบจำลองกับข้อมูลชุดทดสอบ (test set) ด้วย RMSE, MAPE และระยะเวลาในการฝึกฝนแบบจำลอง เพื่อวัดความแม่นยำ ประสิทธิภาพและความคุ้มค่าของแบบจำลองในขั้นสุดท้าย

4.1 ผลการวิจัยของหุ้นที่มีความผันผวนต่ำ (Low Beta)

ผลการวิจัยในการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบสมเพื่อใช้วัดประสิทธิภาพโมเดลจากการเรียนรู้ข้อมูลชุดฝึกฝน (training set) และหาชุดไฮเปอร์พารามิเตอร์ที่ให้ค่า RMSE ต่ำที่สุดจากข้อมูลชุดตรวจสอบ (validation set) โดยค้นหาด้วยวิธี RANDOMSEARCH จำนวน 100 ชุดไฮเปอร์พารามิเตอร์ จากการกำหนดขอบเขตของไฮเปอร์พารามิเตอร์การผสมในรูปแบบ Stacked layers ได้กำหนดขอบเขตไฮเปอร์พารามิเตอร์ทั้งหมด 5 ตัว ได้แก่ จำนวนโนนดในชั้นช่อนแรกจำนวนโนนดในชั้นช่อนที่สอง อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenต้ม) และแบบผสมแบบ Cascaded neural network ทำการกำหนดการจุนไฮเปอร์พารามิเตอร์ 4 ตัวแปร คือ ได้แก่ จำนวนโนนดในชั้นช่อนแรก อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenต้ม) ซึ่งการกำหนดการตั้งค่าเพื่อใช้ในการทำไฮเปอร์พารามิเตอร์จุนนึงดังนี้ จำนวนโนนดในชั้นช่อนกำหนดไว้ตั้งแต่ 1 ถึง 100 หน่วย อัตราการเรียนรู้กำหนดไว้ตั้งแต่ 0.001 ถึง 0.1 ค่าแลมบ์ดาของ L2 Regularization ค่าที่กำหนดไว้คือ 0.001, 0.005, 0.01, 0.05 และ 0.1 และ Beta_1 (ไมemenต้ม) ค่าที่กำหนดไว้คือ 0.7, 0.8 และ 0.9 โดยใช้ RMSE ที่ได้จาก Validation set เป็นเกณฑ์ในการตัดสินใจ ซึ่งจะเลือกตัวแบบที่ดีที่สุดจากชุดไฮเปอร์ที่ให้ค่า RMSE ต่ำที่สุด

เมื่อได้แบบจำลองการเรียนรู้เชิงลึกแบบสม ด้วยวิธี Cascaded Neural Network และ Stacked layers ของหุ้นในกลุ่มความผันผวนต่ำทั้งในระยะสั้น 7 วัน และระยะยาว 30 วัน จากชุดไฮเปอร์พารามิเตอร์ที่ดีที่สุด ผู้วิจัยจึงนำตัวแบบการเรียนรู้เชิงลึกดังกล่าวไปวัดประสิทธิภาพกับ Test set ต่อไป โดยมีตัววัดประสิทธิ์ตัวแบบ คือ RMSE และ MAPE โดยชุด Hyperparameter ที่เหมาะสมที่สุดและผลลัพธ์จากการวิจัยของหุ้นที่มีความผันผวนต่ำ (Low Beta) แสดงดังตารางผนวกดังนี้

KBANK	แสดงตั้งตารางผนวกที่ 1 - 6
STANLY	แสดงตั้งตารางผนวกที่ 7 - 12
PTT	แสดงตั้งตารางผนวกที่ 13 - 18
BDMS	แสดงตั้งตารางผนวกที่ 19 - 24
TRUE	แสดงตั้งตารางผนวกที่ 25 - 30

4.2 ผลการวิจัยของหุ้นที่มีความผันผวนปานกลาง (Medium beta)

ผลการวิจัยในการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบสมเพื่อใช้วัดประสิทธิภาพโมเดลจากการเรียนรู้ข้อมูลชุดฝึกฝน (training set) และหาชุดไฮเปอร์พารามิเตอร์ที่ให้ค่า RMSE ต่ำที่สุดจากข้อมูลชุดตรวจสอบ (validation set) โดยค้นหาด้วยวิธี RANDOMSEARCH จำนวน 100 ชุดไฮเปอร์พารามิเตอร์ จากการกำหนดขอบเขตของไฮเปอร์พารามิเตอร์การผสมในรูปแบบ Stacked layers ได้กำหนดขอบเขตไฮเปอร์พารามิเตอร์ทั้งหมด 5 ตัว ได้แก่ จำนวนโนนดในชั้นช่อนแรกจำนวนโนนดในชั้นช่อนที่สอง อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenตัม) และแบบผสมแบบ Cascaded neural network ทำการกำหนดการจุนไฮเปอร์พารามิเตอร์ 4 ตัวแปร คือ ได้แก่ จำนวนโนนดในชั้นช่อนแรก อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenตัม) ซึ่งการกำหนดการตั้งค่าเพื่อใช้ในการทำไฮเปอร์พารามิเตอร์จุนนึงดังนี้ จำนวนโนนดในชั้นช่อนกำหนดไว้ตั้งแต่ 1 ถึง 100 หน่วย อัตราการเรียนรู้กำหนดไว้ตั้งแต่ 0.001 ถึง 0.1 ค่าแลมบ์ดาของ L2 Regularization ค่าที่กำหนดไว้คือ 0.001, 0.005, 0.01, 0.05 และ 0.1 และ Beta_1 (ไมemenตัม) ค่าที่กำหนดไว้คือ 0.7, 0.8 และ 0.9 โดยใช้ RMSE ที่ได้จาก validation set เป็นเกณฑ์ในการตัดสินใจ ซึ่งจะเลือกตัวแบบที่ดีที่สุดจากชุดไฮเปอร์ที่ให้ค่า RMSE ต่ำที่สุด

เมื่อได้แบบจำลองการเรียนรู้เชิงลึกแบบสม ด้วยวิธี Cascaded Neural Network และ Stacked layers ของหุ้นในกลุ่มความผันผวนปานกลางทั้งในระยะสั้น 7 วัน และระยะยาว 30 วัน จากชุดไฮเปอร์พารามิเตอร์ที่ดีที่สุด ผู้วิจัยจึงนำตัวแบบการเรียนรู้เชิงลึกดังกล่าวไปวัดประสิทธิภาพ กับ Test set ต่อไป โดยมีตัววัดประสิทธิ์ตัวแบบ คือ RMSE และ MAPE โดยชุด Hyperparameter ที่เหมาะสมที่สุดและผลลัพธ์จากการวิจัยของหุ้นที่มีความผันผวนปานกลาง (Medium Beta) แสดงดังตารางนحوต่อไปนี้

MTC แสดงดังตารางนحوที่ 31 - 36

AMC แสดงดังตารางนحوที่ 37 - 42

BCP แสดงดังตารางนحوที่ 43 - 48

MINT แสดงดังตารางนحوที่ 49 - 54

HANA แสดงดังตารางนحوที่ 55 - 60

4.3 ผลการวิจัยของหุ้นที่มีความผันผวนสูง (High beta)

ผลการวิจัยในการสร้างแบบจำลองการเรียนรู้เชิงลึกแบบสมเพื่อใช้วัดประสิทธิภาพโมเดลจากการเรียนรู้ข้อมูลชุดฝึกฝน (training set) และหาชุดไฮเปอร์พารามิเตอร์ที่ให้ค่า RMSE ต่ำที่สุดจากข้อมูลชุดตรวจสอบ (validation set) โดยค้นหาด้วยวิธี RANDOMSEARCH จำนวน 100 ชุดไฮเปอร์พารามิเตอร์ จากการกำหนดขอบเขตของไฮเปอร์พารามิเตอร์การผสมในรูปแบบ Stacked layers ได้กำหนดขอบเขตไฮเปอร์พารามิเตอร์ทั้งหมด 5 ตัว ได้แก่ จำนวนโนนดในชั้นช่อนแรกจำนวนโนนดในชั้นช่อนที่สอง อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenต้ม) และแบบผสมแบบ Cascaded neural network ทำการกำหนดการจุนไฮเปอร์พารามิเตอร์ 4 ตัวแปร คือ ได้แก่ จำนวนโนนดในชั้นช่อนแรก อัตราการเรียนรู้ ค่าแลมบ์ดาของ L2 Regularization และ Beta_1 (ไมemenต้ม) ซึ่งการกำหนดการตั้งค่าเพื่อใช้ในการทำไฮเปอร์พารามิเตอร์จุนนึงดังนี้ จำนวนโนนดในชั้นช่อนกำหนดไว้ตั้งแต่ 1 ถึง 100 หน่วย อัตราการเรียนรู้กำหนดไว้ตั้งแต่ 0.001 ถึง 0.1 ค่าแลมบ์ดาของ L2 Regularization ค่าที่กำหนดไว้คือ 0.001, 0.005, 0.01, 0.05 และ 0.1 (ไมemenต้ม) ค่าที่กำหนดไว้คือ 0.7, 0.8 และ 0.9 โดยใช้ RMSE ที่ได้จาก validation set เป็นเกณฑ์ในการตัดสินใจ ซึ่งจะเลือกตัวแบบที่ดีที่สุดจากชุดไฮเปอร์ที่ให้ค่า RMSE ต่ำที่สุด

เมื่อได้แบบจำลองการเรียนรู้เชิงลึกแบบสม ด้วยวิธี Cascaded Neural Network และ Stacked layers ของหุ้นในกลุ่มความผันผวนสูงทั้งในระยะสั้น 7 วัน และระยะยาว 30 วัน จากชุดไฮเปอร์พารามิเตอร์ที่ดีที่สุด ผู้วิจัยจึงนำตัวแบบการเรียนรู้เชิงลึกดังกล่าวไปวัดประสิทธิภาพกับ Test set ต่อไป โดยมีตัววัดประสิทธิ์ตัวแบบ คือ RMSE และ MAPE โดยชุด Hyperparameter ที่เหมาะสมที่สุดและผลลัพธ์จากการวิจัยของหุ้นที่มีความผันผวนสูง (High Beta) แสดงดังตารางผนวกดังนี้

XPG แสดงตั้งตารางผนวกที่ 61 - 66

PTTGC แสดงตั้งตารางผนวกที่ 67 - 72

TOP แสดงตั้งตารางผนวกที่ 73 - 78

RCL แสดงตั้งตารางผนวกที่ 79 - 84

SMT แสดงตั้งตารางผนวกที่ 85 - 90

จากผลการวิจัยข้างต้นสามารถนำมาสรุปได้ดังตารางที่ 4 – 7

ตารางที่ 4 แบบจำลองและวิธีการทดสอบที่มีประสิทธิภาพที่สุดในการพยากรณ์ระยะสั้น 7 วัน ของแต่ละหุ้นโดยวัดจาก MAPE และ RMSE

Beta	Stock	Best Model by MAPE			Best Model by RMSE		
		Model	Hybrid Method	MAPE	Model	Hybrid Method	RMSE
High	XPG	RNN – GRU	Stacked	6.043	RNN – GRU	Stacked	0.093
	PTTGC	RNN – GRU	Stacked	2.963	RNN – GRU	Stacked	1.345
	TOP	LSTM – RNN	Cascaded	2.468	LSTM – RNN	Cascaded	1.553
	RCL	LSTM – RNN	Stacked	6.723	LSTM – RNN	Stacked	2.102
	SMT	GRU – RNN	Cascaded	4.452	GRU – RNN	Cascaded	0.206
Medium	MTC	LSTM – GRU	Cascaded	2.845	LSTM – GRU	Cascaded	1.566
	AMC	GRU – RNN	Stacked	2.539	GRU – RNN	Stacked	0.099
	BCP	GRU – RNN	Stacked	2.672	GRU – RNN	Stacked	1.382
	MINT	LSTM – RNN	Stacked	2.220	RNN – GRU	Stacked	0.846
	HANA	RNN – GRU	Stacked	5.118	RNN – LSTM	Cascaded	3.154
Low	KBANK	GRU – RNN	Stacked	1.804	RNN – LSTM	Cascaded	2.979
	STANLY	RNN – GRU	Cascaded	2.493	LSTM – RNN	Stacked	7.174
	PTT	RNN – GRU	Stacked	1.717	RNN – GRU	Stacked	0.697
	BDMS	GRU – RNN	Stacked	1.695	GRU – LSTM	Cascaded	0.366
	TRUE	LSTM – RNN	Stacked	4.280	LSTM – RNN	Stacked	0.362

จากการที่ 4 ในการพยากรณ์ระยะสั้น 7 วันของหุ้นทุกรายดับความผันผวนโดยพิจารณาจาก MAPE (%) ในการวัดประสิทธิภาพแบบจำลองและวิธีการทดสอบที่ดีที่สุด ดังนี้ แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น XPG คือ RNN – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 6.043 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น PTTGC คือ RNN – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.963 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น TOP คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.468 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น RCL คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 6.723 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น SMT คือ GRU – RNN โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.452 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น MTC คือ LSTM – GRU โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.845 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น AMC คือ GRU – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.539 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น BCP คือ GRU – RNN โดยใช้วิธีการ

ผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.672 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น MINT คือ LSTM – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.220 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น HANA คือ RNN – GRU โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 5.118 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น KBANK คือ GRU – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 1.804 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น STANLY คือ RNN – GRU โดยใช้วิธีการผลมแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.493 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น PTT คือ RNN – GRU โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 1.717 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น BDMS คือ GRU – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 1.695 และแบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น TRUE คือ LSTM – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.280

ในการพยากรณ์ระยะสั้น 7 วันของหุ้นที่ทุกระดับความผันผวน โดยพิจารณาจาก RMSE ในการวัดประสิทธิภาพแบบจำลองและวิธีการผลมที่ดีที่สุด แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น XPG คือ RNN – GRU โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.093 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น PTTGC คือ RNN – GRU โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 1.345 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น TOP คือ LSTM – RNN โดยใช้วิธีการผลมแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 1.553 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น RCL คือ LSTM – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 2.102 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น SMT คือ GRU – RNN โดยใช้วิธีการผลมแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.206 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น MTC คือ LSTM – GRU โดยใช้วิธีการผลมแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 1.566 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น AMC คือ GRU – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.099 แบบจำลองเชิงลึกแบบผลมที่มีประสิทธิภาพมากที่สุดของหุ้น BCP คือ GRU – RNN โดยใช้วิธีการผลมแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ

1.382 แบบจำลองเชิงลึกแบบผสานที่มีประสิทธิภาพมากที่สุดของหุ้น MINT คือ RNN – GRU โดยใช้ วิธีการผสานแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.846 แบบจำลองเชิงลึกแบบผสานที่มี ประสิทธิภาพมากที่สุดของหุ้น HANA คือ RNN – LSTM โดยใช้ วิธีการผสานแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 3.154 แบบจำลองเชิงลึกแบบผสานที่มีประสิทธิภาพมากที่สุด ของหุ้น KBANK คือ RNN – LSTM โดยใช้ วิธีการผสานแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 2.979 แบบจำลองเชิงลึกแบบผสานที่มีประสิทธิภาพมากที่สุดของหุ้น STANLY คือ LSTM – RNN โดยใช้ วิธีการผสานแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 7.174 แบบจำลองเชิงลึกแบบผสานที่มีประสิทธิภาพมากที่สุดของหุ้น PTT คือ RNN – GRU โดยใช้ วิธีการ ผสานแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.697 แบบจำลองเชิงลึกแบบผสานที่มี ประสิทธิภาพมากที่สุดของหุ้น BDMS คือ GRU – LSTM โดยใช้ วิธีการผสานแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.366 แบบจำลองเชิงลึกแบบผสานที่มีประสิทธิภาพมากที่สุด ของหุ้น TRUE คือ LSTM – RNN โดยใช้ วิธีการผสานแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุด เท่ากับ 0.362

ตารางที่ 5 แบบจำลองและวิธีการผสานที่มีประสิทธิภาพที่สุดในการพยากรณ์ระยะยาว 30 วัน ของแต่ละหุ้นโดยวัดจาก MAPE และ RMSE

Beta	Stock	Best Model by MAPE			Best Model by RMSE		
		Model	Hybrid Method	MAPE	Model	Hybrid Method	RMSE
High	XPG	LSTM – GRU	Stacked	11.973	LSTM – GRU	Stacked	0.176
	PTTGC	LSTM – RNN	Stacked	5.657	RNN – LSTM	Stacked	2.459
	TOP	GRU – RNN	Stacked	4.638	GRU – RNN	Stacked	3.089
	RCL	GRU – RNN	Stacked	11.685	RNN – GRU	Stacked	3.585
	SMT	GRU – RNN	Stacked	8.510	GRU – RNN	Stacked	0.391
Medium	MTC	LSTM – RNN	Cascaded	4.874	LSTM – RNN	Cascaded	2.582
	AMC	GRU – LSTM	Stacked	3.754	LSTM – GRU	Stacked	0.144
	BCP	LSTM – RNN	Stacked	4.981	LSTM – RNN	Stacked	2.422
	MINT	RNN – LSTM	Cascaded	4.265	RNN – LSTM	Cascaded	1.518
	HANA	LSTM – GRU	Stacked	9.215	LSTM – RNN	Cascaded	5.312
Low	KBANK	RNN – GRU	Stacked	3.184	RNN – GRU	Stacked	5.350
	STANLY	LSTM – GRU	Stacked	4.539	GRU – LSTM	Stacked	11.983
	PTT	GRU – LSTM	Cascaded	3.520	GRU – LSTM	Cascaded	1.364
	BDMS	RNN – LSTM	Cascaded	2.577	RNN – LSTM	Cascaded	0.868
	TRUE	LSTM – RNN	Stacked	10.032	LSTM – RNN	Stacked	0.815

จากตารางที่ 5 ในการพยากรณ์ระยะเวลา 30 วันของหั้นทุกระดับความผันผวนโดยพิจารณาจาก MAPE ในการวัดประสิทธิภาพแบบจำลองและวิธีการผลสมที่ดีที่สุด แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น XPG คือ LSTM – GRU โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 11.973 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น PTTGC คือ LSTM – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 5.657 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น TOP คือ GRU – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.638 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น RCL คือ GRU – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 11.685 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น SMT คือ GRU – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 8.510 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น MTC คือ LSTM – RNN LSTM – GRU โดยใช้วิธีการผลสมแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.874 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น AMC คือ GRU – LSTM โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 3.754 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น BCP คือ LSTM – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.981 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น MINT คือ RNN – LSTM โดยใช้วิธีการผลสมแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.265 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น HANA คือ LSTM – GRU โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 9.215 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น KBANK คือ RNN – GRU โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 3.184 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น STANLY คือ LSTM – GRU โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 4.539 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น PTT คือ GRU – LSTM โดยใช้วิธีการผลสมแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 3.520 แบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น BDMS คือ RNN – LSTM โดยใช้วิธีการผลสมแบบ Cascaded Neural Network ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 2.577 และแบบจำลองเชิงลึกแบบผลสมที่มีประสิทธิภาพมากที่สุดของหั้น TRUE คือ LSTM – RNN โดยใช้วิธีการผลสมแบบ Stacked layers ให้ค่า MAPE (%) ต่ำที่สุดเท่ากับ 10.032

ในการพยากรณ์ระยะยาว 30 วันของหุ้นทุกระดับความผันผวนโดยพิจารณาจาก RMSE ใน การวัดประสิทธิภาพแบบจำลองและวิธีการทดสอบที่ดีที่สุด แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น XPG คือ LSTM – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.176 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น PTTGC คือ RNN – LSTM โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 2.459 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น TOP คือ GRU – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 3.089 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น RCL คือ RNN – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 3.585 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น SMT คือ GRU – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.391 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น MTC คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 2.582 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น AMC คือ LSTM – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.144 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น BCP คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 2.422 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น MINT คือ RNN – LSTM โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 1.518 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น HANA คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 5.312 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น KBANK คือ RNN – GRU โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 5.350 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น STANLY คือ GRU – LSTM โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 11.983 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น PTT คือ GRU – LSTM โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 1.364 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น BDMS คือ RNN – LSTM โดยใช้วิธีการทดสอบแบบ Cascaded Neural Network ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.868 แบบจำลองเชิงลึกแบบทดสอบที่มีประสิทธิภาพมากที่สุดของหุ้น TRUE คือ LSTM – RNN โดยใช้วิธีการทดสอบแบบ Stacked layers ให้ค่า RMSE ต่ำที่สุดเท่ากับ 0.815

ตารางที่ 6 Hybrid Deep Learning Model ที่ดีที่สุดสำหรับพยากรณ์ระยะสั้น 7 วันโดยรัดจากระยะเวลาในการฝึกแบบจำลอง

Beta	Stock	Best Model by Runtime		Runtime	MAPE (%)	Range of MAPE (Min - Max)
		Model	Hybrid Method			
High	XPG	LSTM – GRU	Stacked	1,710	6.732	6.043 - 8.308
	PTTGC	GRU – RNN	Stacked	1,123	3.083	2.963 - 4.274
	TOP	RNN – LSTM	Stacked	1,443	2.973	2.468 - 3.316
	RCL	GRU – RNN	Stacked	564	7.125	6.723 - 9.600
	SMT	RNN – LSTM	Stacked	3,333	4.879	4.452 - 8.708
Medium	MTC	RNN – LSTM	Stacked	2,824	3.370	2.845 - 4.468
	AMC	LSTM – GRU	Cascaded	2,929	4.472	2.539 - 4.973
	BCP	RNN – GRU	Stacked	1,918	2.876	2.672 - 3.846
	MINT	RNN – LSTM	Stacked	2,277	2.370	2.220 - 2.551
	HANA	LSTM – RNN	Cascaded	589	6.237	5.118 - 6.923
Low	KBANK	LSTM – GRU	Cascaded	1,957	2.398	1.804 - 2.398
	STANLY	RNN – LSTM	Stacked	2,987	4.140	2.493 - 4.140
	PTT	LSTM – GRU	Stacked	1,257	2.199	1.717 - 2.577
	BDMS	GRU – RNN	Stacked	1,727	1.695	1.695 - 1.908
	TRUE	RNN – LSTM	Stacked	2,933	5.842	4.280 - 6.893

จากการวัด Runtime ในการรัดประสิทธิภาพแบบจำลองและวิธีการผสมที่ดีที่สุด ดังนี้แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น XPG คือ LSTM – GRU โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,710 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 6.732 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 6.043 - 8.308 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น PTTGC คือ GRU – RNN โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,123 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 3.083 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.963 - 4.274 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น TOP คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,443 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.973 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.468 - 3.316 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น RCL คือ GRU – RNN โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 564 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 7.125 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 6.723 - 9.600 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น SMT คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ

3,333 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 4.879 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.452 - 8.708 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น MTC คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 2,824 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 3.370 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.845 - 4.468 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น AMC คือ LSTM – GRU โดยใช้วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 2,929 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 4.472 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.539 - 4.973 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น BCP คือ RNN – GRU โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,918 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.876 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.672 - 3.846 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น MINT คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 2,277 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.370 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.220 - 2.551 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น HANA คือ LSTM – RNN โดยใช้วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 589 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 6.237 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 5.118 - 6.923 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น KBANK คือ LSTM – GRU โดยใช้วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,957 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.398 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 1.804 - 2.398 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น STANLY คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 2,987 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 4.140 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.493 - 4.140 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น PTT คือ LSTM – GRU โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,257 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.199 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 1.717 - 2.577 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น BDMS คือ GRU – RNN โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,727 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 1.695 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 1.695 - 1.908 และแบบจำลองเชิงลึก

แบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น TRUE คือ RNN – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 2,933 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 5.842 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.280 - 6.893

ตารางที่ 7 Hybrid Deep Learning Model ที่ดีที่สุดสำหรับพยากรณ์ระยะยาว 30 วัน โดยวัดจากระยะเวลาในการฝึกฝนแบบจำลอง

Beta	Stock	Best Model by Runtime		Runtime	MAPE (%)	Range of MAPE (Min - Max)
		Model	Hybrid Method			
High	XPG	RNN – GRU	Stacked	1,380	14.272	11.973 - 18.684
	PTTGC	GRU – LSTM	Stacked	1,135	5.762	5.657 - 7.966
	TOP	GRU – LSTM	Stacked	560	5.354	4.638 - 5.584
	RCL	LSTM – RNN	Stacked	1,220	12.000	11.685 - 13.558
	SMT	RNN – GRU	Stacked	559	9.770	8.510 - 15.438
Medium	MTC	RNN – GRU	Stacked	555	6.035	4.874 - 6.833
	AMC	LSTM – GRU	Stacked	1,129	3.799	2.801 - 4.859
	BCP	LSTM – GRU	Cascaded	1,481	7.626	4.981 - 7.626
	MINT	LSTM – GRU	Stacked	958	5.391	4.265 - 5.391
	HANA	LSTM – RNN	Stacked	3,102	9.355	9.215 - 10.627
Low	KBANK	GRU – RNN	Cascaded	1,503	3.648	3.184 - 4.433
	STANLY	RNN – GRU	Stacked	636	5.245	4.539 - 5.706
	PTT	GRU – LSTM	Cascaded	760	3.520	3.520 - 4.917
	BDMS	GRU – LSTM	Stacked	1,748	2.589	2.577 - 3.028
	TRUE	GRU – RNN	Stacked	1,108	11.259	10.032 - 14.363

จากดังตารางที่ 7 ใน การพยากรณ์ระยะยาว 30 วันของหุ้นทุกระดับความผันผวนโดยพิจารณาจาก Runtime ใน การวัดประสิทธิภาพแบบจำลองและวิธีการผสมที่ดีที่สุด ดังนี้แบบจำลอง เชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น XPG คือ RNN – GRU โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,380 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 14.272 โดยช่วงของ MAPE (%) จาก การพยากรณ์ของข้อมูลชุดนี้ คือ 11.973 - 18.684 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น PTTGC คือ GRU – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,135 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 5.762 โดยช่วงของ MAPE (%) จาก การพยากรณ์ของข้อมูลชุดนี้ คือ 5.657 - 7.966 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น TOP คือ GRU – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 560 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 5.354 โดยช่วงของ MAPE (%) จาก การพยากรณ์ของข้อมูลชุดนี้ คือ 4.638 - 5.584 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น RCL คือ LSTM – RNN โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า

Runtime ต่ำที่สุดเท่ากับ 1,220 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 12.000 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 11.685 - 13.558 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น SMT คือ RNN – GRU โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 559 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 9.770 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 8.510 - 15.438 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น MTC คือ RNN – GRU โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 555 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 6.035 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.874 - 6.833 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น AMC คือ LSTM – GRU โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,129 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 3.799 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.801 - 4.859 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น BCP คือ LSTM – GRU โดยใช้ วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,481 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 7.626 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.981 - 7.626 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น MINT คือ LSTM – GRU โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 958 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 5.391 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.265 - 5.391 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น HANA คือ LSTM – RNN โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 3,102 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 9.355 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 9.215 - 10.627 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น KBANK คือ GRU – RNN โดยใช้ วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,503 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 3.648 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 3.184 - 4.433 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น STANLY คือ RNN – GRU โดยใช้ วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 636 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 5.245 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 4.539 - 5.706 แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น PTT คือ GRU – LSTM โดยใช้ วิธีการผสมแบบ Cascaded Neural Network ให้ค่า Runtime ต่ำที่สุดเท่ากับ 760 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 3.520 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 3.520 - 4.917

แบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น BDMS คือ GRU – LSTM โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,748 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 2.589 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 2.577 - 3.028 และแบบจำลองเชิงลึกแบบผสมที่มีประสิทธิภาพมากที่สุดของหุ้น TRUE คือ GRU – RNN โดยใช้วิธีการผสมแบบ Stacked layers ให้ค่า Runtime ต่ำที่สุดเท่ากับ 1,108 วินาที ซึ่งให้ค่า MAPE (%) เท่ากับ 11.259 โดยช่วงของ MAPE (%) จากการพยากรณ์ของข้อมูลชุดนี้ คือ 10.032 - 14.363



บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอการเปรียบเทียบประสิทธิภาพและระยะเวลาที่ใช้ในการสร้าง Hybrid Deep Learning Models ในรูปแบบ Stacked layer และ Cascaded neural network สำหรับการพยากรณ์ในระยะสั้น 7 วัน และระยะยาว 30 วัน โดยแบ่งการศึกษาออกเป็น 3 ส่วนหลัก

ตารางที่ 8 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพของการลับแบบจำลองที่ใช้การผสมแบบ Stacked layer

Combination	Metric	P-value	
		7-Day	30-Day
RNN/LSTM	MAPE (%)	0.2078	0.4543
	RMSE	0.3028	0.8469
RNN/GRU	MAPE (%)	0.8904	0.3028
	RMSE	0.7615	0.6788
LSTM/GRU	MAPE (%)	0.2933	0.4887
	RMSE	0.3591	0.0901

ตารางที่ 9 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพของการลับแบบจำลองที่ใช้การผสมแบบ Cascaded neural network

Combination	Metric	P-value	
		7-Day	30-Day
RNN/LSTM	MAPE (%)	0.1205	0.2078
	RMSE	0.4212	0.5245
RNN/GRU	MAPE (%)	0.2524	0.5995
	RMSE	0.1673	0.5995
LSTM/GRU	MAPE (%)	0.8469	0.1514
	RMSE	0.5245	0.0637

ในส่วนที่ 1 ทำการทดสอบทางสถิติด้วย Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพจากการลับแบบจำลองก่อน-หลังระหว่างแบบจำลอง RNN/ LSTM, RNN/GRU, และ LSTM/GRU ในการผสมแบบจำลองด้วยวิธี Cascaded neural network และ Stacked layer ดังตารางที่ 8 - 9 พบร่วมกัน P-value ของ MAPE (%) และ RMSE

ในการพยากรณ์ระยะสั้น 7 วันและระยะยาว 30 วัน มีค่ามากกว่า 0.05 ในทุก Hybrid Models จึงสรุปได้ว่าค่ามัธยฐานของ Evaluation Metric MAPE(%) และ RMSE ไม่แตกต่างกันอย่างมีนัยสำคัญหรือสรุปได้ว่า การสลับลำดับของแบบจำลองที่ใช้ใน Cascaded neural network และ Stacked layers ไม่ส่งผลต่อประสิทธิภาพของ Hybrid Deep Learning Models

ตารางที่ 10 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network

Combination	Metric	P-value	
		7-Day	30-Day
RNN-LSTM	MAPE (%)	0.0479*	0.8040
	RMSE	0.0946	0.8904
LSTM-RNN	MAPE (%)	0.0637	0.3591
	RMSE	0.0353*	0.1688
RNN-GRU	MAPE (%)	0.1514	0.3028
	RMSE	0.2524	0.3303
GRU-RNN	MAPE (%)	0.2293	0.5614
	RMSE	0.0353*	0.9780
LSTM-GRU	MAPE (%)	0.8647	0.6788
	RMSE	0.4887	0.8469
GRU-LSTM	MAPE (%)	0.7197	0.8040
	RMSE	0.5614	0.7982

* แตกต่างกันอย่างมีนัยสำคัญ ที่ระดับนัยสำคัญ 0.05

ส่วนที่ 2 ทำการทดสอบทางสถิติด้วย Wilcoxon's Signed Rank เพื่อเปรียบเทียบประสิทธิภาพระหว่าง การสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers กับวิธี Cascaded Neural Network ในภาพรวม ดังตารางที่ 10

ในการพยากรณ์ระยะสั้น 7 วัน หากพิจารณาด้วย MAPE (%) จะพบว่าในกรณีที่ Hybrid model ถูกสร้างโดย RNN-LSTM จะให้ P-value ที่น้อยกว่าระดับนัยสำคัญที่ 0.05 แสดงว่ากรณีนี้ค่ามัธยฐานของ MAPE (%) ของวิธี Cascaded neural network แตกต่างกับ Stacked layers อย่างมีนัยสำคัญ หรืออีกนัยหนึ่งการสร้าง Hybrid Model ด้วย Cascaded neural network มีประสิทธิภาพที่แตกต่างกับการใช้ Stacked layers โดยที่ Cascaded neural network จะให้ประสิทธิภาพที่ดีกว่าเมื่อเทียบกับ Stacked layers หากสร้าง Hybrid model ด้วย RNN-LSTM ส่วนกรณีที่ Hybrid model ถูกสร้างโดยคู่อื่น ๆ จะให้ P-value ที่มากกว่าระดับนัยสำคัญที่ 0.05 แสดงว่ากรณีนี้ค่ามัธยฐานของ MAPE (%) ของวิธี Cascaded neural network ไม่แตกต่างกับ

Stacked layers อย่างมีนัยสำคัญ หรือการสร้างแบบจำลองผสมด้วย Cascaded neural network มีประสิทธิภาพที่ไม่แตกต่างกับการใช้ Stacked layers และ หากพิจารณาด้วย RMSE จะพบว่าในกรณีที่ Hybrid model ถูกสร้างโดย LSTM-RNN และ GRU-RNN จะให้ P-value ที่น้อยกว่าระดับนัยสำคัญที่ 0.05 แสดงว่ากรณีนี้ค่ามัธยฐานของ RMSE ของวิธี Cascaded neural network แตกต่างกับ Stacked layers อย่างมีนัยสำคัญ หรืออีกนัยหนึ่งการสร้าง Hybrid Model ด้วย Cascaded neural network มีประสิทธิภาพที่แตกต่างกับการใช้ Stacked layers แต่ในกรณีนี้พบว่าการสร้าง Hybrid Model ด้วย Stacked layers จะให้ประสิทธิภาพที่ดีกว่าเมื่อเทียบกับ Cascaded neural network หาก LSTM-RNN และ GRU-RNN ส่วนกรณีที่ Hybrid model ถูกสร้างโดยคู่อื่น ๆ จะให้ P-value ที่มากกว่าระดับนัยสำคัญที่ 0.05 แสดงว่ากรณีนี้ค่ามัธยฐานของ MAPE (%) ของวิธี Cascaded neural network ไม่แตกต่างกับ Stacked layers อย่างมีนัยสำคัญ หรือการสร้างแบบจำลองผสมด้วย Cascaded neural network มีประสิทธิภาพที่ไม่แตกต่างกับการใช้ Stacked layers ในส่วนของการพยากรณ์ระยะยาว 30 วัน P-value ของ Evaluation Metric MAPE (%) และ RMSE, Hybrid Deep Learning Model มีค่ามากกว่า 0.05 ในทุก Hybrid Models จึงสามารถสรุปได้ว่าค่ามัธยฐานของ Evaluation Metric ไม่แตกต่างกันอย่างมีนัยสำคัญ หรืออีกนัยหนึ่ง การใช้วิธี Cascaded neural network มีประสิทธิภาพที่ไม่แตกต่างกับการใช้ Stacked layers

ตารางที่ 11 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ในกลุ่มความผันผวนสูง

Combination	Metric	P-value	
		7-Day	30-Day
RNN-LSTM	MAPE (%)	0.8125	0.8125
	RMSE	1.0000	0.4375
LSTM-RNN	MAPE (%)	0.8125	0.8125
	RMSE	0.6250	0.3125
RNN-GRU	MAPE (%)	0.0625	0.6250
	RMSE	0.1250	0.4375
GRU-RNN	MAPE (%)	0.6250	0.1875
	RMSE	1.0000	0.4375
LSTM-GRU	MAPE (%)	0.6250	0.8125
	RMSE	1.0000	0.6250
GRU-LSTM	MAPE (%)	0.4375	1.0000
	RMSE	0.1875	1.0000

ตารางที่ 12 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ในกลุ่มความผันผวนปานกลาง

Combination	Metric	P-value	
		7-Day	30-Day
RNN-LSTM	MAPE (%)	0.3125	1.0000
	RMSE	0.1250	1.0000
LSTM-RNN	MAPE (%)	0.0625	0.6250
	RMSE	0.0625	0.8125
RNN-GRU	MAPE (%)	0.0625	0.1250
	RMSE	0.0625	0.1875
GRU-RNN	MAPE (%)	0.3125	0.4375
	RMSE	0.3125	0.4375
LSTM-GRU	MAPE (%)	1.0000	0.4375
	RMSE	1.0000	0.8125
GRU-LSTM	MAPE (%)	0.8125	1.0000
	RMSE	0.8125	0.8125

ตารางที่ 13 P-value จากการทดสอบ Wilcoxon's Signed Rank เพื่อการเปรียบเทียบประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network ในกลุ่มความผันผวนต่ำ

Combination	Metric	P-value	
		7-Day	30-Day
RNN-LSTM	MAPE (%)	0.1250	0.3125
	RMSE	0.1875	0.6250
LSTM-RNN	MAPE (%)	0.8125	0.1875
	RMSE	0.3125	0.3125
RNN-GRU	MAPE (%)	0.3125	0.3125
	RMSE	0.1875	0.4375
GRU-RNN	MAPE (%)	0.1250	1.0000
	RMSE	0.0625	1.0000
LSTM-GRU	MAPE (%)	0.8125	0.4375
	RMSE	0.3125	0.6250
GRU-LSTM	MAPE (%)	0.4375	0.6250
	RMSE	0.1875	1.0000

ขณะเดียวกัน ผู้วิจัยได้พิจารณาผลการพยากรณ์จำแนกตามระดับความผันผวนสูง ปานกลาง และต่ำ ดังตารางที่ 11 - 13 พบว่าในการพยากรณ์ระยะสั้น 7 วันและระยะยาว 30 วัน มีค่า P-value มากกว่า 0.05 ในทุก Hybrid Models จึงสรุปได้ว่าค่ามัธยฐานของ Evaluation Metric MAPE (%) และ RMSE ประสิทธิภาพของการใช้วิธีการผสมทั้งสองรูปแบบตามระดับความผันผวนสูง ปานกลาง และต่ำ ไม่แตกต่างกันอย่างมีนัยสำคัญ

ค่ามัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) และ RMSE ในภาพรวม แสดงดังตารางที่ 14 - 15 ตามลำดับ

ตารางที่ 14 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลอง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	3.846	3.126	5.669	5.321
LSTM-RNN	3.300	3.626	4.981	5.276
RNN-GRU	2.963	3.316	6.035	5.584
GRU-RNN	2.980	3.150	5.484	5.193
LSTM-GRU	3.501	3.377	5.391	5.533
GRU-LSTM	3.280	3.316	5.354	5.105

ตารางที่ 15 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลอง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	1.363	1.434	2.459	2.733
LSTM-RNN	1.429	1.499	2.422	2.487
RNN-GRU	1.345	1.535	2.735	2.586
GRU-RNN	1.382	1.396	2.771	2.734
LSTM-GRU	1.586	1.487	2.639	2.620
GRU-LSTM	1.473	1.555	2.439	2.649

ค่ามัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) และ RMSE ตามระดับความผันผวนสูง
ปานกลาง และต่ำ แสดงดังตารางที่ 16 – 21 ตามลำดับ

ตารางที่ 16 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนสูง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	4.879	4.578	10.566	12.861
LSTM-RNN	6.667	5.695	12.000	10.815
RNN-GRU	4.758	5.349	9.770	11.427
GRU-RNN	5.579	4.452	8.510	11.117
LSTM-GRU	5.260	4.653	12.301	10.101
GRU-LSTM	5.360	4.781	11.973	10.918

ตารางที่ 17 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนสูง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	1.363	1.434	2.459	3.029
LSTM-RNN	1.600	1.553	2.502	2.896
RNN-GRU	1.345	1.535	2.735	2.633
GRU-RNN	1.424	1.396	3.089	2.962
LSTM-GRU	1.549	1.676	2.505	2.811
GRU-LSTM	1.586	1.487	2.735	3.606

ตารางที่ 18 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวน
ปานกลาง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	3.846	3.305	5.178	5.321
LSTM-RNN	3.300	3.626	4.981	4.874
RNN-GRU	2.922	3.646	6.035	4.904
GRU-RNN	2.672	3.311	5.484	5.169
LSTM-GRU	2.852	3.086	5.004	5.001
GRU-LSTM	3.619	3.540	5.391	5.072

ตารางที่ 19 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนปานกลาง

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	1.763	1.626	2.527	2.733
LSTM-RNN	1.429	1.499	2.422	2.487
RNN-GRU	1.460	1.570	3.100	2.586
GRU-RNN	1.382	1.570	2.771	2.734
LSTM-GRU	1.473	1.555	2.439	2.649
GRU-LSTM	1.630	1.566	2.639	2.620

ตารางที่ 20 มัธยฐานของการวัดประสิทธิภาพด้วย MAPE (%) ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนต่ำ

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	3.846	3.305	5.178	5.321
LSTM-RNN	3.300	3.626	4.981	4.874
RNN-GRU	2.922	3.646	6.035	4.904
GRU-RNN	2.672	3.311	5.484	5.169
LSTM-GRU	2.852	3.086	5.004	5.001
GRU-LSTM	3.619	3.540	5.391	5.072

ตารางที่ 21 มัธยฐานของการวัดประสิทธิภาพด้วย RMSE ระหว่างวิธี Stacked layers กับวิธี Cascaded Neural Network ในทุก Combination ของแบบจำลองตามกลุ่มระดับความผันผวนต่ำ

Combination	7-Day		30-Day	
	Stack	Cascaded	Stack	Cascaded
RNN-LSTM	0.987	0.802	1.632	1.598
LSTM-RNN	0.831	0.785	1.818	1.599
RNN-GRU	0.697	0.774	1.565	1.468
GRU-RNN	0.703	0.784	1.809	1.659
LSTM-GRU	0.788	0.799	1.682	1.364
GRU-LSTM	0.858	0.790	1.802	1.646

จากผลการวิเคราะห์ข้างต้นจะพบว่าประสิทธิภาพ Hybrid Deep Learning Models ระหว่าง Stacked layer และ Cascaded neural network มีหลายกรณีไม่แตกต่างกันอย่างมีนัยสำคัญ ดังนั้นในส่วนที่ 3 จะทำการเปรียบเทียบระยะเวลาที่ใช้ในการสร้าง Hybrid Deep Learning Model ในทุกหุ้นพร้อมระบุแบบจำลองและวิธีการทดสอบที่ใช้ระยะเวลาในการฝึกฝนแบบจำลองน้อยที่สุด เพื่อแสดงให้เห็นถึงแบบจำลองและรูปแบบการทดสอบแบบจำลองที่เหมาะสมและคุ้มค่าที่สุดกับการนำไปใช้จริง

ในส่วนที่ 3 จากการเปรียบเทียบระยะเวลาที่ใช้ในการฝึกฝนแบบจำลอง ดังตารางที่ 6 - 7 ในบทที่ 4 พบว่า ในการพยากรณ์ระยะสั้น 7 วัน และในระยะยาว 30 วัน วิธีการสร้าง Hybrid Model ที่ใช้ระยะเวลาในการฝึกฝนแบบจำลองน้อยที่สุดในกลุ่มหุ้นที่มีความผันผวนสูง คือวิธี Stacked layers ซึ่งให้ประสิทธิภาพไม่ต่างจากแบบจำลองที่ดีที่สุดมากนัก ในส่วนของกลุ่มความผันผวนปานกลางและความผันผวนต่ำ วิธีการสร้าง Hybrid Model ที่ใช้เวลาในการฝึกฝนแบบจำลองน้อยที่สุด คือวิธี Stacked layers เช่นเดียวกัน แต่อาจมีบางกรณีที่ Cascaded neural network นั้นใช้เวลาประมาณผลที่เร็วกว่า เนื่องจากการใช้ Stacked layers ข้อมูลต้องโหลดผ่านทุกเลเยอร์โดยตรง ทำให้มีการคำนวณเชิงลึกส่งผลให้ใช้เวลามากในบางกรณี แต่ในส่วนของ Cascaded Neural Network เป็นการพยากรณ์เป็นสองชั้ntonแยกกัน DL1 ไม่จำเป็นต้องเรียนรู้รายละเอียดของค่าความคลาดเคลื่อนเนื่องจาก DL2 จะทำหน้าที่ปรับค่าที่เหลือแทน หาก DL1 สามารถพยากรณ์ค่าได้แม่นยำเพียงพอ ค่า Residuals ที่เหลือจะมีขนาดเล็กมาก ทำให้ DL2 ต้องทำงานน้อยลง เนื่องจากเรียนรู้เฉพาะ Residuals ที่มีขนาดเล็กและรูปแบบไม่ซับซ้อน อาจทำให้คำนวณได้เร็วขึ้นในบางกรณี

โดยสรุปแล้วแม้ว่าการสร้าง Hybrid Deep Learning model ด้วยวิธี Cascaded Neural Network อาจจะมีข้อดีในบางกรณี แต่การสร้างด้วยวิธี Stacked layers เป็นทางเลือกที่เหมาะสมกว่าสำหรับการสร้าง Hybrid Deep Learning Models สำหรับการพยากรณ์ เนื่องจากให้ประสิทธิภาพที่ไม่ต่างกันอย่างมีนัยสำคัญ ใช้ระยะเวลาในการฝึกฝนและทรัพยากรในการสร้าง Hybrid Deep Learning Model น้อยกว่า Cascaded Neural Network เมื่อนำไปประยุกต์ใช้จริงในการพยากรณ์ราคาหุ้นหรือแนวโน้มทางธุรกิจในภาพรวม รูปแบบการทดสอบแบบจำลองที่สามารถให้ประสิทธิภาพและความเสถียรในการพยากรณ์ ความคุ้มค่า และมีความเหมาะสมในการนำมาใช้งานได้ในหลายสถานการณ์

5.2 ข้อเสนอแนะ

งานวิจัยนี้ผู้ที่สนใจจากจะนำไปศึกษาต่อได้อีกในเรื่องต่อไปนี้

1. ในงานวิจัยนี้เลือกใช้แบบจำลองการเรียนรู้เชิงลึก (Deep Learning) ที่ใช้ร่วมกับข้อมูลประเภทอนุกรมเวลามาใช้ในการผสมแบบจำลอง (Hybrid Model) เพียง 3 ตัวเท่านั้น ได้แก่ RNN, LSTM และ GRU ซึ่งในความเป็นจริงแล้วยังมีแบบจำลองอื่น ๆ ที่น่าสนใจที่จะนำมาใช้ในการผสมแบบจำลอง ในการศึกษาครั้งต่อไปอาจนำ CNN (1D Convnets) นำมากสักด้วยคุณลักษณะเด่นก่อนที่จะนำมาผสมผสานด้วย RNN, LSTM หรือ GRU ช่วยพยากรณ์ ซึ่งอาจเพิ่มประสิทธิภาพการพยากรณ์ได้
2. ในงานวิจัยนี้ผู้วิจัยได้เลือกใช้การแปลงข้อมูลก่อนนำไปสร้างแบบจำลอง เพื่อให้แบบจำลองสามารถเรียนรู้ข้อมูลได้ดีขึ้น ด้วยวิธี Min-Max normalization ผู้ที่สนใจใช้วิธีอื่นในการแปลงข้อมูล เช่น Z-score standardization หรือ Log Transformation เพื่อเปรียบเทียบการแปลงข้อมูลที่แตกต่างกันส่งผลต่อความแม่นยำและเสถียรภาพของแบบจำลอง
3. ในงานวิจัยนี้เลือกใช้วิธีการผสมผสานแบบจำลองการเรียนรู้เชิงลึกเพียง 2 วิธีการเท่านั้น ได้แก่ Stacked layers และ Cascaded Neural Network (Additive combination) ซึ่งในการศึกษาครั้งต่อไปอาจนำวิธีการผสมผสานรูปแบบอื่นมาใช้เพื่อเปรียบเทียบกับวิธีนี้ได้ยกตัวอย่างเช่น การผสมผสานแบบจำลองการเรียนรู้เชิงลึก Linear Regression Combination เป็นอีกหนึ่งวิธีที่น่าสนใจและได้รับความนิยมในการนำมาใช้ในการผสมผสานโดยเดล หรือวิธีการผสมผสานเชิงสถิติและการเรียนรู้เชิงลึกแบบอื่น ๆ เพื่อค้นหาแนวทางที่เหมาะสมและมีประสิทธิภาพสูงสุดสำหรับข้อมูลหุ้นหรือข้อมูลธุรกิจที่มีลักษณะแตกต่างกัน
4. โปรแกรมค้นหาชุดพารามิเตอร์ที่เหมาะสมที่สุด (Search engine) ผู้วิจัยเลือกใช้ Randomized search เนื่องจากทรัพยากรคอมพิวเตอร์ของผู้วิจัยมีจำกัด ในการศึกษาครั้งต่อไปอาจเลือกใช้ Engine search ตัวอื่นที่มีประสิทธิภาพในการค้นหาชุดไอก่อน ที่มีคุณภาพมากกว่านี้ เช่น Hyper Band, Bayesian Optimization เป็นต้น ซึ่งอาจช่วยให้ค้นหาชุดพารามิเตอร์ที่เหมาะสมได้แม่นยำและรวดเร็วยิ่งขึ้น

บรรณานุกรม

- Adebiyi, A., Adewumi, A., & Ayo, C. (2014). Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics*, 2014, 1-7. <https://doi.org/10.1155/2014/614342>
- Al-Thelaya, K., El-Alfy, E.-S., & Mohammed, S. (2018). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. <https://doi.org/10.1109/IACS.2018.8355458>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9, 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hossain, M., Karim, R., Thulasiram, R., Bruce, N., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction. <https://doi.org/10.1109/SSCI.2018.8628641>
- Khairalla, M. A. E., Xu-Ning, & Al-Jallad, N. T. (2017). Hybrid Forecasting Scheme for Financial Time-Series Data using Neural Network and Statistical Methods. *International Journal of Advanced Computer Science and Applications*, 8.
- Liu, Y., Wang, Z., & Zheng, B. (2019). Application of Regularized GRU-LSTM Model in Stock Price Prediction. <https://doi.org/10.1109/ICCC47050.2019.9064035>
- Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33, 1-13. <https://doi.org/10.1007/s00521-020-05532-z>
- Pandian, S. (2025). Time series analysis: Definition, components, methods, and applications. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/>
- Pirani, M., Thakkar, P., Jivrani, P., Bohara, M., & Garg, D. (2022). A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting. <https://doi.org/10.1109/ICDCECE53908.2022.9793213>

- Salalli, B. (2023, 09 Oct, 2023). *Stacked RNNs in NLP*.
<https://www.geeksforgeeks.org/stacked-rnns-in-nlp/>
- Wijaya, Y. B., Kom, S., & Napitupulu, T. A. (2010, 2-3 Dec. 2010). Stock Price Prediction: Comparison of Arima and Artificial Neural Network Methods - An Indonesia Stock's Case. 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies,
- Zhang, P. (2003). Zhang, G.P.: Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing* 50, 159-175. *Neurocomputing*, 50, 159-175.
[https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- กุลวนิช, น. (2564). การเพิ่มเติมແມ່ນຍໍ່ໃນການພຍາກຮົນຂອ້ມູນຕາມເວລາດ້ວຍຕົວແບບຜສນ ARIMA-ANN ແລະ ການວິເຄາະທີ່ກາຣົດດອຍ. *ວາරສານວິທະຍາສາສົຽນແລະ ເຖິງໂຄໂລຢີ*, 28(12), 12.
<https://doi.org/10.14456/tstj.2020.167>
- ຈົງສຶກສົງ, ກ. (2564). ການເປີຍບັນດາທີ່ໃນການພຍາກຮົນຈຳລອງອາຮີມ່າ, ໂຄງໝ່າຍ ປະສາທເຫີມ ແລະ ຕົວແບບຜສນ ຈຸ່າລັງກຮົນມໍາໄວທະຍາລັຍ].
- ເຄຣວົງ, ຮ. (2566). ປະສິທິກາພຂອງແບບຈຳລອງໂຄຮັງສ້າງຜສນຮ່ວງແບບຈຳລອງໜ່ວຍຄວາມຈຳຮະຍະ ສັນແບບຍາວແລະ ແບບຈຳລອງໂຄຮັງຂ່າຍປະຕູວັກກັບສໍາຫັບການພຍາກຮົນຮະຍະຍາວຂອງຈຳລັງ ຫຼຸ້ນ ຈຸ່າລັງກຮົນມໍາໄວທະຍາລັຍ]. <https://cuir.car.chula.ac.th/handle/123456789/85693>
- ບຸ້ມູນານະ, ທ. (2558). ການເປີຍບັນດາທີ່ໃນການພຍາກຮົນດ້ວຍຕົວແບບອນຸກຮົມເວລາແບບຜສນ ຈຸ່າລັງກຮົນມໍາໄວທະຍາລັຍ].
- ປະກອບຜລ, ອ. (2552). ໂຄງໝ່າຍປະສາທເຫີມ. *ວາරສານ ມະກ.ວິຊາການ*, 12(24), 15.
- ກູສືເຈີຍາ, ຮ. (2564). ການພຍາກຮົນປະມານນັ້ນພະຍະສັນໃນບົຣິເວັນພື້ນທີ່ສະນາບິນສຸວະຮົນຄູນມີດ້ວຍ ໂຄງໝ່າຍຮະບບປະສາທແບບຍັອນກລັບ ຈຸ່າລັງກຮົນມໍາໄວທະຍາລັຍ].

ภาคผนวก

กลุ่มหุ้นที่มีความผันผวนต่ำ (Low Beta)

ตารางผนวกที่ 1 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะล้วน 7 วันของ KBANK

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	49	82	0.003	0.001	0.700
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	56	34	0.004	0.001	0.800
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	34	56	0.004	0.001	0.800

ตารางผนวกที่ 2 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะล้วน 7 วันของ KBANK

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	12	0.018	0.001	0.900
	Predicted error	24	0.036	0.100	0.700
LSTM – RNN	Predicted value	12	0.018	0.001	0.900
	Predicted error	12	0.018	0.001	0.900
RNN – GRU	Predicted value	95	0.038	0.001	0.900
	Predicted error	70	0.003	0.001	0.800
GRU – RNN	Predicted value	12	0.018	0.001	0.900
	Predicted error	48	0.010	0.001	0.700
LSTM – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	70	0.003	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	23	0.018	0.001	0.800

ตารางผนวกที่ 3 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ KBANK

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	56	34	0.004	0.001	0.800
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	19	37	0.010	0.001	0.700
LSTM – GRU	37	19	0.010	0.001	0.700
GRU – LSTM	82	49	0.003	0.001	0.700

ตารางผนวกที่ 4 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ KBANK

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	31	0.055	0.100	0.800
LSTM – RNN	Predicted value	12	0.018	0.001	0.900
	Predicted error	9	0.036	0.100	0.800
RNN – GRU	Predicted value	95	0.038	0.001	0.900
	Predicted error	91	0.013	0.001	0.900
GRU – RNN	Predicted value	48	0.010	0.001	0.700
	Predicted error	1	0.082	0.001	0.700
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	69	0.031	0.001	0.900
	Predicted error	4	0.099	0.100	0.800

ตารางผนวกที่ 5 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ KBANK

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	1.9494	3.1561	2,652	1.8129	2.9788	4,901
LSTM – RNN	1.8382	3.0402	3,060	1.9720	3.2347	5,636
RNN – GRU	1.9671	3.1064	2,983	1.8318	3.0211	5,671
GRU – RNN	1.8038	3.0113	3,287	2.2994	3.5882	2,345
LSTM – GRU	2.2612	3.6403	3,307	2.3982	3.7543	1,957
GRU – LSTM	2.3151	3.7269	3,312	2.1067	3.3801	5,661

ตารางผนวกที่ 6 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ KBANK

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	3.2246	5.4691	3,036	3.3366	5.6439	3,086
LSTM – RNN	3.4146	5.7026	3,075	4.4328	7.0728	5,631
RNN – GRU	3.1839	5.3500	2,944	3.6764	6.1113	5,597
GRU – RNN	3.2812	5.5065	3,286	3.6482	6.0329	1,503
LSTM – GRU	3.6401	6.0677	3,311	3.6579	6.0606	1,909
GRU – LSTM	3.4262	5.7761	3,314	3.5278	5.7482	1,890

ตารางผนวกที่ 7 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ STANLY

Model	Hyperparameter Tuning				
	Node 1st layer	Node 2nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 8 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ STANLY

Model	Hyperparameter Tuning				
	Node 1st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	52	0.001	0.100	0.800
LSTM – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	12	0.018	0.001	0.900
RNN – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	52	0.001	0.100	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	75	0.001	0.100	0.800
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	24	0.036	0.100	0.700
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	75	0.001	0.100	0.800

ตารางผนวกที่ 9 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ STANLY

Model	Hyperparameter Tuning				
	Node 1st layer	Node 2nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	34	56	0.004	0.001	0.800

ตารางผนวกที่ 10 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ STANLY

Model	Hyperparameter Tuning				
	Node 1st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	59	0.007	0.100	0.800
LSTM – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	36	0.002	0.005	0.900
RNN – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	70	0.003	0.001	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	36	0.025	0.010	0.900
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	13	0.032	0.100	0.700
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	38	0.009	0.100	0.800

ตารางผนวกที่ 11 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ STANLY

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	4.1403	10.7647	2,987	2.5377	7.2964	6,302
LSTM – RNN	2.5050	7.1736	3,215	2.5469	7.2259	6,664
RNN – GRU	3.5616	9.4508	3,191	2.4932	7.2240	6,792
GRU – RNN	2.6685	7.3875	3,789	2.9875	8.1614	6,860
LSTM – GRU	2.6669	7.6374	3,923	2.7906	7.8910	6,901
GRU – LSTM	3.2798	8.8837	3,924	2.9726	8.1456	6,856

ตารางผนวกที่ 12 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ STANLY

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.7060	15.1538	3,053	4.5979	12.6518	6,722
LSTM – RNN	4.6109	12.1406	3,204	5.4190	14.5926	4,323
RNN – GRU	5.2451	14.1268	636	4.5470	12.2943	4,772
GRU – RNN	4.7844	13.2811	3,839	4.6309	12.8128	4,780
LSTM – GRU	4.5394	12.4734	3,931	4.8768	13.6031	4,718
GRU – LSTM	4.7058	11.9829	3,891	4.8011	13.2294	4,441

ตารางผนวกที่ 13 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ PTT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	82	49	0.003	0.001	0.700
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 14 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ PTT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
LSTM – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	91	0.013	0.001	0.900
RNN – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	1	0.082	0.001	0.700
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	56	0.062	0.005	0.900
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	24	0.036	0.100	0.700

ตารางผนวกที่ 15 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ PTT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	78	99	0.001	0.010	0.900
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	63	73	0.002	0.005	0.800
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	34	56	0.004	0.001	0.800

ตารางผนวกที่ 16 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ PTT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	28	0.076	0.050	0.900
LSTM – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
RNN – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	9	0.036	0.100	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	4	0.100	0.099	0.800
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	9	0.036	0.100	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	31	0.055	0.100	0.800

ตารางผนวกที่ 17 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ PTT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	2.5772	0.9871	2,246	1.9916	0.8015	3,686
LSTM – RNN	2.1198	0.8306	2,447	1.9421	0.7852	3,184
RNN – GRU	1.7172	0.6971	2,679	1.9135	0.7741	3,991
GRU – RNN	1.7721	0.7028	1,374	1.9561	0.7840	3,229
LSTM – GRU	2.1985	0.8577	1,257	1.9725	0.7903	3,236
GRU – LSTM	1.9795	0.7884	1,475	2.0145	0.7989	3,739

ตารางผนวกที่ 18 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ PTT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	4.3411	1.6323	2,397	4.0894	1.5981	2,737
LSTM – RNN	4.9167	1.8180	2,427	4.1987	1.5989	2,123
RNN – GRU	4.1318	1.5645	2,073	3.8424	1.4682	945
GRU – RNN	4.8787	1.8089	1,374	4.3641	1.6587	2,949
LSTM – GRU	4.8494	1.8016	1,255	4.3489	1.6463	2,127
GRU – LSTM	4.4747	1.6821	1,483	3.5200	1.3636	760

ตารางผนวกที่ 19 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ BDMS

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	49	82	0.003	0.001	0.700
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	82	49	0.003	0.001	0.700

ตารางผนวกที่ 20 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ BDMS

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	12	0.018	0.001	0.900
	Predicted error	56	0.013	0.100	0.900
LSTM – RNN	Predicted value	93	0.028	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
RNN – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	47	0.098	0.050	0.900
GRU – RNN	Predicted value	93	0.028	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	70	0.003	0.001	0.800
GRU – LSTM	Predicted value	91	0.013	0.001	0.900
	Predicted error	88	0.005	0.050	0.800

ตารางผนวกที่ 21 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ BDMS

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	91	38	0.004	0.010	0.900
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	91	38	0.004	0.010	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	90	57	0.049	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 22 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ BDMS

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	44	0.066	0.100	0.800
LSTM – RNN	Predicted value	93	0.028	0.001	0.900
	Predicted error	59	0.007	0.100	0.800
RNN – GRU	Predicted value	87	0.083	0.005	0.900
	Predicted error	17	0.004	0.001	0.800
GRU – RNN	Predicted value	12	0.018	0.001	0.900
	Predicted error	52	0.006	0.005	0.700
LSTM – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	48	0.010	0.001	0.700
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	45	0.027	0.005	0.900

ตารางผนวกที่ 23 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ BDMS

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	1.7610	0.6303	2,944	1.7786	0.6403	5,673
LSTM – RNN	1.8116	0.6523	4,438	1.7637	0.6390	6,586
RNN – GRU	1.9079	0.6600	4,447	1.7224	0.6279	6,648
GRU – RNN	1.6947	0.6242	1,727	1.7610	0.6432	6,695
LSTM – GRU	1.7494	0.6402	4,474	1.8192	0.6634	6,737
GRU – LSTM	1.7905	0.6550	1,750	1.7302	0.3664	6,230

ตารางผนวกที่ 24 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ BDMS

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	2.7363	0.9436	2,974	2.5774	0.8679	6,458
LSTM – RNN	2.7892	0.9109	3,129	3.0175	1.0670	5,077
RNN – GRU	2.8489	1.0138	3,097	2.6523	0.9234	3,626
GRU – RNN	2.7712	0.9189	4,451	2.7704	0.9538	2,100
LSTM – GRU	3.0283	1.0613	4,532	2.6963	0.9189	5,388
GRU – LSTM	2.5889	0.8926	1,748	2.5868	0.8977	5,322

ตารางผนวกที่ 25 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ TRUE

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	70	48	0.012	0.050	0.800
LSTM – RNN	99	78	0.001	0.010	0.800
RNN – GRU	86	47	0.019	0.050	0.900
GRU – RNN	99	78	0.001	0.010	0.800
LSTM – GRU	92	69	0.059	0.050	0.900
GRU – LSTM	34	56	0.004	0.010	0.800

ตารางผนวกที่ 26 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ TRUE

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	12	0.018	0.001	0.900
	Predicted error	63	0.076	0.001	0.800
LSTM – RNN	Predicted value	12	0.018	0.001	0.900
	Predicted error	91	0.013	0.001	0.900
RNN – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	96	0.077	0.005	0.800
GRU – RNN	Predicted value	12	0.012	0.001	0.900
	Predicted error	21	0.020	0.005	0.900
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	53	0.083	0.050	0.800
GRU – LSTM	Predicted value	69	0.009	0.001	0.800
	Predicted error	67	0.043	0.001	0.700

ตารางผนวกที่ 27 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ TRUE

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	91	38	0.004	0.010	0.900
LSTM – RNN	68	98	0.035	0.050	0.900
RNN – GRU	82	76	0.072	0.005	0.800
GRU – RNN	10	14	0.022	0.050	0.800
LSTM – GRU	84	57	0.092	0.001	0.800
GRU – LSTM	37	70	0.051	0.050	0.900

ตารางผนวกที่ 28 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ TRUE

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	98	0.035	0.005	0.800
	Predicted error	31	0.055	0.100	0.800
LSTM – RNN	Predicted value	48	0.078	0.001	0.800
	Predicted error	34	0.049	0.100	0.900
RNN – GRU	Predicted value	48	0.010	0.001	0.700
	Predicted error	21	0.017	0.010	0.700
GRU – RNN	Predicted value	93	0.033	0.001	0.700
	Predicted error	26	0.017	0.010	0.800
LSTM – GRU	Predicted value	93	0.030	0.001	0.700
	Predicted error	89	0.049	0.050	0.700
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	36	0.100	0.005	0.700

ตารางผนวกที่ 29 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะ
ลั้น 7 วันของ TRUE

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.8421	0.4608	2,933	5.1837	0.4564	5,756
LSTM – RNN	4.2803	0.3624	3,201	5.5560	0.4983	5,875
RNN – GRU	6.8932	0.5674	3,407	4.7448	0.4214	5,891
GRU – RNN	6.7303	0.5845	4,149	6.7013	0.6053	5,560
LSTM – GRU	6.7839	0.5251	4,174	6.1280	0.5383	5,516
GRU – LSTM	4.4679	0.3911	4,156	4.5781	0.3832	6,063

ตารางผนวกที่ 30 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะ
ยาว 30 วันของ TRUE

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	10.2661	0.9033	3,127	10.3866	0.9225	5,543
LSTM – RNN	10.0316	0.8148	3,291	11.3825	0.9900	5,644
RNN – GRU	14.3632	1.2839	3,199	12.1734	1.1002	5,662
GRU – RNN	11.2585	1.0177	1,108	11.6529	1.0663	5,691
LSTM – GRU	12.3605	1.1044	4,158	10.6110	0.9174	4,091
GRU – LSTM	11.5754	0.9913	1,123	11.2899	1.0128	3,786

กลุ่มหุ้นที่มีความผันผวนปานกลาง (Medium Beta)

ตารางผนวกที่ 31 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะลั้น 7 วันของ MTC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	65	67	0.014	0.005	0.900
GRU – LSTM	47	86	0.019	0.001	0.900

ตารางผนวกที่ 32 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะลั้น 7 วันของ MTC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	36	0.002	0.005	0.900
	Predicted error	8	0.090	0.010	0.800
LSTM – RNN	Predicted value	95	0.038	0.001	0.900
	Predicted error	36	0.002	0.005	0.900
RNN – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	23	0.018	0.005	0.800
GRU – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	36	0.098	0.010	0.700
LSTM – GRU	Predicted value	91	0.013	0.001	0.900
	Predicted error	12	0.018	0.001	0.900
GRU – LSTM	Predicted value	69	0.009	0.001	0.800
	Predicted error	56	0.013	0.100	0.900

ตารางผนวกที่ 33 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ MTC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	31	47	0.089	0.001	0.900
RNN – GRU	60	57	0.013	0.005	0.900
GRU – RNN	57	90	0.049	0.001	0.900
LSTM – GRU	66	57	0.061	0.001	0.900
GRU – LSTM	47	86	0.019	0.001	0.900

ตารางผนวกที่ 34 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ MTC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	60	0.058	0.001	0.700
	Predicted error	31	0.056	0.050	0.700
LSTM – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	88	0.005	0.050	0.800
RNN – GRU	Predicted value	36	0.002	0.005	0.900
	Predicted error	25	0.022	0.100	0.700
GRU – RNN	Predicted value	24	0.054	0.001	0.800
	Predicted error	59	0.007	0.010	0.700
LSTM – GRU	Predicted value	36	0.002	0.005	0.900
	Predicted error	4	0.099	0.100	0.800
GRU – LSTM	Predicted value	48	0.010	0.001	0.700
	Predicted error	1	0.082	0.001	0.700

ตารางผนวกที่ 35 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ MTC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	3.3698	1.7634	2,824	3.0661	1.6263	6,933
LSTM – RNN	3.5634	1.8390	2,848	3.6259	1.8822	7,629
RNN – GRU	3.3463	1.7543	2,929	3.6460	1.9130	7,868
GRU – RNN	3.8330	1.9383	3,403	3.3111	1.7510	7,800
LSTM – GRU	4.4677	2.2519	3,546	2.8446	1.5655	3,220
GRU – LSTM	4.2654	2.1661	3,491	3.0860	1.6147	3,134

ตารางผนวกที่ 36 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ MTC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	6.6337	3.3610	2,838	5.3205	2.8489	7,296
LSTM – RNN	5.1869	2.7656	2,933	4.8741	2.5816	7,741
RNN – GRU	6.0346	3.0996	555	4.9044	2.5863	7,868
GRU – RNN	6.8331	3.4417	3,419	5.1692	2.7344	7,707
LSTM – GRU	5.0381	2.6387	3,533	5.0720	2.6198	3,238
GRU – LSTM	6.2794	3.1859	3,444	5.0006	2.6494	3,020

ตารางผนวกที่ 37 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ AMC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	58	54	0.013	0.005	0.900
LSTM – RNN	57	66	0.061	0.001	0.900
RNN – GRU	10	58	0.010	0.005	0.700
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	65	67	0.014	0.005	0.900
GRU – LSTM	47	86	0.019	0.001	0.900

ตารางผนวกที่ 38 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ AMC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	35	0.039	0.001	0.800
	Predicted error	75	0.001	0.100	0.800
LSTM – RNN	Predicted value	93	0.028	0.001	0.900
	Predicted error	75	0.001	0.100	0.800
RNN – GRU	Predicted value	35	0.039	0.001	0.800
	Predicted error	75	0.001	0.100	0.800
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	75	0.001	0.100	0.800
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	79	0.025	0.100	0.900
GRU – LSTM	Predicted value	69	0.009	0.001	0.800
	Predicted error	52	0.001	0.100	0.088

ตารางผนวกที่ 39 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ AMC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	49	82	0.003	0.001	0.700
LSTM – RNN	84	89	0.081	0.001	0.900
RNN – GRU	10	58	0.010	0.005	0.700
GRU – RNN	57	90	0.049	0.001	0.900
LSTM – GRU	70	37	0.051	0.005	0.900
GRU – LSTM	54	58	0.013	0.005	0.900

ตารางผนวกที่ 40 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ AMC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	7	0.039	0.005	0.800
	Predicted error	6	0.085	0.010	0.900
LSTM –	Predicted value	17	0.004	0.001	0.800
RNN	Predicted error	1	0.082	0.001	0.700
RNN – GRU	Predicted value	91	0.013	0.001	0.900
	Predicted error	14	0.092	0.010	0.700
GRU – RNN	Predicted value	69	0.031	0.001	0.900
	Predicted error	31	0.055	0.100	0.800
LSTM – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	12	0.069	0.050	0.800
GRU – LSTM	Predicted value	12	0.018	0.001	0.900
	Predicted error	56	0.045	0.050	0.800

ตารางผนวกที่ 41 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ AMC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	3.8988	0.1420	3,078	3.7731	0.1369	6,152
LSTM – RNN	3.3003	0.1202	3,315	4.4007	0.1603	6,583
RNN – GRU	2.9221	0.1044	3,295	3.9084	0.1438	6,678
GRU – RNN	2.5387	0.0990	3,882	4.5557	0.1668	5,612
LSTM – GRU	3.6192	0.1307	3,908	4.4716	0.1674	2,929
GRU – LSTM	2.8011	0.1059	3,898	4.9732	0.1811	6,716

ตารางผนวกที่ 42 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ AMC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	4.6704	0.1715	3,232	4.6484	0.1747	6,362
LSTM – RNN	3.9055	0.1503	3,303	4.4227	0.1659	5,255
RNN – GRU	4.6647	0.1712	3,282	4.5002	0.1657	5,813
GRU – RNN	4.8585	0.1817	2,165	4.4401	0.1618	5,865
LSTM – GRU	3.7987	0.1435	1,129	4.3019	0.1587	5,835
GRU – LSTM	3.7543	0.1444	3,899	4.0711	0.1493	5,735

ตารางผนวกที่ 43 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ BCP

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	63	73	0.002	0.005	0.800
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	78	99	0.001	0.010	0.800
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 44 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ BCP

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	36	0.002	0.005	0.900
LSTM – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	70	0.003	0.001	0.800
RNN – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	9	0.036	0.100	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	70	0.003	0.001	0.800
LSTM – GRU	Predicted value	69	0.009	0.001	0.800
	Predicted error	49	0.028	0.010	0.700
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	61	0.029	0.005	0.900

ตารางผนวกที่ 45 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ BCP

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	78	99	0.001	0.010	0.800
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	56	34	0.004	0.001	0.800
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	38	91	0.004	0.010	0.900

ตารางผนวกที่ 46 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ BCP

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	69	0.031	0.001	0.900
	Predicted error	38	0.058	0.100	0.900
LSTM – RNN	Predicted value	61	0.029	0.005	0.900
	Predicted error	17	0.004	0.001	0.800
RNN – GRU	Predicted value	91	0.013	0.001	0.900
	Predicted error	68	0.090	0.100	0.800
GRU – RNN	Predicted value	69	0.009	0.001	0.800
	Predicted error	100	0.031	0.050	0.900
LSTM – GRU	Predicted value	48	0.010	0.001	0.700
	Predicted error	70	0.003	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	76	0.015	0.100	0.800

ตารางผนวกที่ 47 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ BCP

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	3.8461	1.9321	3,209	3.3051	1.6542	5,987
LSTM – RNN	2.8070	1.4290	2,491	2.9646	1.4993	5,707
RNN – GRU	2.8758	1.4601	1,918	3.0499	1.5697	6,321
GRU – RNN	2.6715	1.3819	2,555	3.1499	1.5697	6,423
LSTM – GRU	3.1836	1.6296	3,008	3.5398	1.7878	3,089
GRU – LSTM	2.8517	1.4729	3,175	3.0681	1.5548	6,452

ตารางผนวกที่ 48 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ BCP

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.1778	2.5265	1,923	5.4791	2.7330	3,302
LSTM – RNN	4.9808	2.4219	2,541	4.9971	2.4871	3,279
RNN – GRU	6.3920	3.1746	1,916	6.0223	3.0122	3,383
GRU – RNN	5.4836	2.7713	2,524	6.8635	3.4492	1,996
LSTM – GRU	5.6188	2.8377	3,025	7.6262	3.6648	1,481
GRU – LSTM	5.0038	2.4391	3,164	6.0946	3.0480	3,064

ตารางผนวกที่ 49 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ MINT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 50 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ MINT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	69	0.009	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
LSTM – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	12	0.018	0.001	0.900
RNN – GRU	Predicted value	36	0.002	0.005	0.900
	Predicted error	17	0.004	0.001	0.800
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	12	0.018	0.001	0.900
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	38	0.058	0.100	0.900

ตารางผนวกที่ 51 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ MINT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	91	38	0.004	0.010	0.900
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	91	38	0.004	0.010	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	90	57	0.049	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 52 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ MINT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	25	0.022	0.100	0.700
LSTM – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	70	0.003	0.001	0.800
RNN – GRU	Predicted value	69	0.009	0.001	0.800
	Predicted error	28	0.076	0.050	0.900
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	8	0.090	0.010	0.800
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	56	0.045	0.050	0.800

ตารางผนวกที่ 53 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ MINT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	2.3695	0.8703	2,277	2.2955	0.8719	5,202
LSTM – RNN	2.2204	0.8541	2,704	2.3991	0.9039	8,789
RNN – GRU	2.2395	0.8462	2,699	2.5513	0.9498	8,920
GRU – RNN	2.2627	0.8584	2,914	2.2655	0.8663	4,103
LSTM – GRU	2.3343	0.8880	3,257	2.4498	0.9285	4,254
GRU – LSTM	2.2897	0.8578	3,230	2.2961	0.8676	4,233

ตารางผนวกที่ 54 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ MINT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	4.5165	1.5592	3,703	4.2649	1.5177	9,297
LSTM – RNN	4.5114	1.5802	2,702	4.8345	1.7428	9,272
RNN – GRU	4.4857	1.5441	2,687	4.6119	1.6200	9,151
GRU – RNN	4.7459	1.6408	3,136	4.3984	1.5620	9,251
LSTM – GRU	5.3908	1.8760	958	4.5482	1.6003	5,566
GRU – LSTM	4.5328	1.5790	3,196	4.3707	1.5417	9,136

ตารางผนวกที่ 55 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ HANA

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	73	63	0.002	0.005	0.800
RNN – GRU	56	34	0.004	0.001	0.800
GRU – RNN	82	49	0.003	0.001	0.700
LSTM – GRU	58	54	0.013	0.005	0.900
GRU – LSTM	67	65	0.014	0.005	0.900

ตารางผนวกที่ 56 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ HANA

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	12	0.018	0.001	0.900
	Predicted error	75	0.001	0.100	0.800
LSTM – RNN	Predicted value	69	0.009	0.001	0.800
	Predicted error	36	0.002	0.005	0.900
RNN – GRU	Predicted value	69	0.031	0.001	0.001
	Predicted error	52	0.001	0.100	0.100
GRU – RNN	Predicted value	95	0.038	0.001	0.900
	Predicted error	38	0.058	0.100	0.900
LSTM – GRU	Predicted value	48	0.010	0.001	0.700
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	82	0.034	0.100	0.700

ตารางผนวกที่ 57 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ HANA

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	70	48	0.012	0.005	0.800
LSTM – RNN	73	63	0.002	0.005	0.800
RNN – GRU	65	67	0.014	0.005	0.900
GRU – RNN	99	78	0.001	0.010	0.800
LSTM – GRU	65	67	0.014	0.005	0.900
GRU – LSTM	57	60	0.013	0.005	0.900

ตารางผนวกที่ 58 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ HANA

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	67	0.043	0.001	0.700
	Predicted error	57	0.075	0.005	0.700
LSTM – RNN	Predicted value	69	0.009	0.001	0.800
	Predicted error	33	0.040	0.100	0.900
RNN – GRU	Predicted value	91	0.013	0.001	0.900
	Predicted error	100	0.031	0.050	0.900
GRU – RNN	Predicted value	63	0.076	0.001	0.800
	Predicted error	25	0.022	0.100	0.700
LSTM – GRU	Predicted value	69	0.009	0.001	0.800
	Predicted error	100	0.031	0.050	0.900
GRU – LSTM	Predicted value	93	0.033	0.001	0.700
	Predicted error	59	0.007	0.010	0.700

ตารางผนวกที่ 59 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ HANA

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.4384	3.1931	2,641	5.5864	3.1536	1,798
LSTM – RNN	5.8273	3.4422	3,150	6.2373	3.4974	589
RNN – GRU	5.1180	3.1687	3,258	5.8922	3.3065	3,072
GRU – RNN	5.5586	3.2510	3,214	6.9233	3.7956	2,115
LSTM – GRU	6.4037	3.7183	2,626	6.0784	3.5942	3,085
GRU – LSTM	6.4809	3.6746	3,239	6.3066	3.5154	1,078

ตารางผนวกที่ 60 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ HANA

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	9.3656	5.5300	3,157	9.6486	5.6373	4,529
LSTM – RNN	9.3550	5.5880	3,102	9.2788	5.3123	4,504
RNN – GRU	10.6271	6.2771	3,256	9.8364	5.7436	4,667
GRU – RNN	9.9489	5.9346	3,216	9.5841	5.6383	4,782
LSTM – GRU	9.2151	5.4442	3,257	9.7057	5.5922	4,040
GRU – LSTM	9.5004	5.4594	3,212	9.5943	5.5936	3,996

กลุ่มหุ้นที่มีความผันผวนสูง (High Beta)

ตารางผนวกที่ 61 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะลั้น 7 วันของ XPG

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	63	73	0.002	0.005	0.800
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	49	82	0.003	0.001	0.700
GRU – RNN	99	78	0.001	0.010	0.800
LSTM – GRU	49	82	0.003	0.001	0.700
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 62 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะลั้น 7 วันของ XPG

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	56	0.045	0.050	0.800
LSTM – RNN	Predicted value	48	0.010	0.001	0.700
	Predicted error	69	0.031	0.001	0.900
RNN – GRU	Predicted value	93	0.028	0.001	0.900
	Predicted error	93	0.028	0.001	0.900
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	69	0.009	0.001	0.800
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	13	0.032	0.100	0.700
GRU – LSTM	Predicted value	91	0.013	0.001	0.900
	Predicted error	91	0.013	0.001	0.900

ตารางผนวกที่ 63 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะยาว 30 วันของ XPG

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	70	37	0.051	0.005	0.900
LSTM – RNN	69	92	0.059	0.001	0.900
RNN – GRU	80	60	0.024	0.050	0.700
GRU – RNN	35	96	0.015	0.100	0.800
LSTM – GRU	53	53	0.066	0.050	0.900
GRU – LSTM	14	40	0.058	0.005	0.800

ตารางผนวกที่ 64 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะยาว 30 วันของ XPG

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	38	0.009	0.100	0.800
	Predicted error	38	0.009	0.100	0.800
LSTM – RNN	Predicted value	59	0.007	0.010	0.700
	Predicted error	4	0.099	0.100	0.800
RNN – GRU	Predicted value	52	0.006	0.005	0.700
	Predicted error	24	0.036	0.100	0.700
GRU – RNN	Predicted value	99	0.076	0.010	0.900
	Predicted error	13	0.032	0.100	0.700
LSTM – GRU	Predicted value	93	0.028	0.001	0.900
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	52	0.006	0.005	0.700
	Predicted error	13	0.032	0.100	0.700

ตารางผนวกที่ 65 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ XPG

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	6.9960	0.1061	6,220	6.7846	0.1050	5,841
LSTM – RNN	6.6665	0.1027	6,273	7.9861	0.1182	6,339
RNN – GRU	6.0430	0.0931	6,321	7.5216	0.1123	6,438
GRU – RNN	7.1803	0.1090	6,344	7.1033	0.1102	6,470
LSTM – GRU	6.7318	0.1017	1,710	7.3130	0.1124	6,575
GRU – LSTM	6.7735	0.1028	5,683	8.3081	0.1214	6,497

ตารางผนวกที่ 66 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ XPG

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	18.2027	0.2642	6,261	12.9145	0.1923	6,260
LSTM – RNN	18.6838	0.2880	6,293	14.2526	0.2086	3,578
RNN – GRU	14.2722	0.2087	1,380	12.6453	0.1873	3,387
GRU – RNN	11.9889	0.1779	2,347	12.9822	0.1938	3,483
LSTM – GRU	11.9734	0.1763	1,643	13.7404	0.2074	3,951
GRU – LSTM	13.2100	0.1923	1,750	15.5110	0.2259	3,836

ตารางผนวกที่ 67 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ PTTGC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	47	86	0.019	0.001	0.900

ตารางผนวกที่ 68 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ PTTGC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	91	0.013	0.001	0.900
	Predicted error	69	0.068	0.050	0.800
LSTM – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	47	0.098	0.050	0.900
RNN – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	31	0.055	0.100	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	36	0.100	0.005	0.700
LSTM – GRU	Predicted value	69	0.009	0.001	0.800
	Predicted error	59	0.015	0.100	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	89	0.049	0.050	0.700

ตารางผนวกที่ 69 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ PTTGC

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	78	99	0.001	0.010	0.800
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	60	57	0.013	0.005	0.900
GRU – RNN	7	35	0.019	0.010	0.800
LSTM – GRU	60	57	0.013	0.005	0.900
GRU – LSTM	73	63	0.002	0.005	0.800

ตารางผนวกที่ 70 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ PTTGC

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	66	0.024	0.005	0.800
	Predicted error	1	0.082	0.001	0.700
LSTM – RNN	Predicted value	21	0.020	0.005	0.900
	Predicted error	87	0.090	0.005	0.800
RNN – GRU	Predicted value	93	0.030	0.001	0.700
	Predicted error	69	0.009	0.001	0.800
GRU – RNN	Predicted value	66	0.024	0.005	0.800
	Predicted error	93	0.028	0.001	0.700
LSTM – GRU	Predicted value	69	0.031	0.001	0.900
	Predicted error	56	0.013	0.100	0.900
GRU – LSTM	Predicted value	21	0.020	0.005	0.900
	Predicted error	31	0.055	0.100	0.800

ตารางผนวกที่ 71 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ PTTGC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	3.0184	1.3633	2,165	3.1260	1.4340	4,681
LSTM – RNN	3.5678	1.6087	3,930	4.2739	1.8860	4,871
RNN – GRU	2.9633	1.3445	3,909	3.3162	1.5349	4,956
GRU – RNN	3.0832	1.4236	1,123	3.0531	1.3960	4,971
LSTM – GRU	3.5007	1.5857	3,379	3.3771	1.4869	4,908
GRU – LSTM	3.5567	15488	3,382	3.6699	1.6758	6,350

ตารางผนวกที่ 72 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ PTTGC

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.6690	2.4594	2,078	6.7277	3.0294	6,555
LSTM – RNN	5.6571	2.5019	2,171	6.4709	2.8958	6,665
RNN – GRU	6.3392	2.7348	3,893	6.1791	2.6328	6,735
GRU – RNN	7.4682	3.3350	3,357	6.6147	2.9624	6,709
LSTM – GRU	6.1816	2.7353	3,381	7.9658	3.6062	6,505
GRU – LSTM	5.7616	2.5050	1,135	6.2937	2.8109	6,353

ตารางผนวกที่ 73 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ TOP

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	82	49	0.003	0.001	0.700
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	47	55	0.003	0.001	0.900

ตารางผนวกที่ 74 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ TOP

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
LSTM – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	91	0.013	0.001	0.900
RNN – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	1	0.082	0.001	0.700
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	56	0.062	0.005	0.900
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	24	0.036	0.100	0.700

ตารางผนวกที่ 75 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ TOP

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	78	99	0.001	0.010	0.900
LSTM – RNN	82	49	0.003	0.001	0.700
RNN – GRU	63	73	0.002	0.005	0.800
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	55	47	0.003	0.001	0.900
GRU – LSTM	34	56	0.004	0.001	0.800

ตารางผนวกที่ 76 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ TOP

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	28	0.076	0.050	0.900
LSTM – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
RNN – GRU	Predicted value	12	0.018	0.001	0.900
	Predicted error	9	0.036	0.100	0.800
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	4	0.100	0.099	0.800
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	9	0.036	0.100	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	31	0.055	0.100	0.800

ตารางผนวกที่ 77 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ TOP

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	2.9731	1.9129	1,443	2.7433	1.7643	6,280
LSTM – RNN	2.5212	1.6000	1,835	2.4676	1.5530	6,500
RNN – GRU	2.8899	1.8615	1,613	2.9445	1.8570	6,625
GRU – RNN	2.9800	1.9124	2,281	2.9725	1.9155	6,696
LSTM – GRU	2.8746	1.8391	2,661	2.8948	1.8416	6,712
GRU – LSTM	3.1732	2.0008	2,572	3.3156	2.0710	6,672

ตารางผนวกที่ 78 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ TOP

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	5.1717	3.4402	1,668	4.7210	3.1316	6,578
LSTM – RNN	4.7481	3.1422	1,829	5.2758	3.3787	965
RNN – GRU	5.0539	3.3927	1,370	5.5839	3.7229	2,698
GRU – RNN	4.6382	3.0889	2,624	5.1934	3.4720	2,705
LSTM – GRU	5.2058	3.4840	2,643	5.5327	3.6377	1,469
GRU – LSTM	5.3536	3.5949	560	5.1053	3.3008	4,116

ตารางผนวกที่ 79 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะสั้น 7 วันของ RCL

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	56	34	0.004	0.001	0.800
LSTM – RNN	47	55	0.003	0.001	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	37	19	0.010	0.001	0.700
GRU – LSTM	57	60	0.013	0.005	0.900

ตารางผนวกที่ 80 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะสั้น 7 วันของ RCL

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	87	0.047	0.001	0.800
	Predicted error	36	0.002	0.005	0.900
LSTM – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	12	0.018	0.001	0.900
RNN – GRU	Predicted value	69	0.031	0.001	0.900
	Predicted error	36	0.002	0.005	0.900
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	75	.001	0.100	0.800
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	17	0.004	0.001	0.800
GRU – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	60	0.058	0.001	0.700

ตารางผนวกที่ 81 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ RCL

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	49	82	0.003	0.001	0.700
LSTM – RNN	34	56	0.004	0.001	0.800
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	34	56	0.004	0.001	0.800
LSTM – GRU	56	34	0.004	0.001	0.800
GRU – LSTM	82	49	0.003	0.001	0.700

ตารางผนวกที่ 82 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ RCL

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	93	0.030	0.001	0.700
	Predicted error	24	0.054	0.001	0.800
LSTM – RNN	Predicted value	69	0.009	0.001	0.800
	Predicted error	13	0.032	0.100	0.700
RNN – GRU	Predicted value	93	0.028	0.001	0.900
	Predicted error	1	0.082	0.001	0.700
GRU – RNN	Predicted value	17	0.004	0.001	0.800
	Predicted error	1	0.082	0.001	0.700
LSTM – GRU	Predicted value	17	0.004	0.001	0.800
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	17	0.004	0.001	0.800
	Predicted error	4	0.099	0.100	0.800

ตารางผนวกที่ 83 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ RCL

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	6.9894	2.1804	1,511	9.6003	2.8371	4,539
LSTM – RNN	6.7232	2.1024	1,601	8.4259	2.4773	6,285
RNN – GRU	7.2107	2.2293	1,345	8.0017	2.3502	6,414
GRU – RNN	7.1248	2.1804	564	8.4785	2.5306	5,830
LSTM – GRU	7.7196	2.3883	3,344	8.6443	2.5302	6,497
GRU – LSTM	7.4058	2.2750	3,336	7.7919	2.3846	7,094

ตารางผนวกที่ 84 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ RCL

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	12.1050	3.6180	1,573	12.8726	3.9280	6,511
LSTM – RNN	11.9997	3.7207	1,220	13.5584	4.0773	6,448
RNN – GRU	11.9164	3.5851	1,337	12.7501	3.9008	4,629
GRU – RNN	11.6848	3.6886	3,255	12.7823	3.8616	2,870
LSTM – GRU	12.4333	3.8278	3,341	12.2300	3.7736	4,459
GRU – LSTM	13.0670	3.8879	3,324	12.8363	3.9048	4,658

ตารางผนวกที่ 85 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะลั่น 7 วันของ SMT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	55	47	0.003	0.001	0.900
LSTM – RNN	14	74	0.051	0.005	0.900
RNN – GRU	55	47	0.003	0.001	0.900
GRU – RNN	47	55	0.003	0.001	0.900
LSTM – GRU	86	47	0.019	0.001	0.900
GRU – LSTM	82	49	0.003	0.001	0.700

ตารางผนวกที่ 86 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะลั่น 7 วันของ SMT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	93	0.028	0.001	0.900
	Predicted error	70	0.003	0.001	0.800
LSTM – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	6	0.016	0.010	0.700
RNN – GRU	Predicted value	87	0.047	0.001	0.800
	Predicted error	15	0.071	0.005	0.900
GRU – RNN	Predicted value	70	0.003	0.001	0.800
	Predicted error	44	0.023	0.010	0.900
LSTM – GRU	Predicted value	95	0.038	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
GRU – LSTM	Predicted value	70	0.003	0.001	0.800
	Predicted error	26	0.020	0.010	0.700

ตารางผนวกที่ 87 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Stacked layers ในการพยากรณ์ระยะเวลา 30 วันของ SMT

Model	Hyperparameter Tuning				
	Node 1 st layer	Node 2 nd layer	Learning rate	L2(lambda)	Beta_1 (Momentum)
RNN – LSTM	56	34	0.004	0.001	0.800
LSTM – RNN	68	98	0.035	0.001	0.900
RNN – GRU	56	34	0.004	0.001	0.800
GRU – RNN	14	74	0.051	0.005	0.900
LSTM – GRU	14	55	0.077	0.001	0.900
GRU – LSTM	47	86	0.019	0.001	0.900

ตารางผนวกที่ 88 แสดงชุด Hyperparameter tuning ที่ดีที่สุดในการสร้าง Hybrid Deep Learning Model ด้วยวิธี Cascaded Neural network ในการพยากรณ์ระยะเวลา 30 วันของ SMT

Model	Hyperparameter Tuning				
	Node 1 st layer	Learning rate	L2(lambda)	Beta_1 (Momentum)	
RNN – LSTM	Predicted value	60	0.058	0.001	0.700
	Predicted error	89	0.049	0.050	0.700
LSTM – RNN	Predicted value	91	0.013	0.001	0.900
	Predicted error	17	0.004	0.001	0.800
RNN – GRU	Predicted value	69	0.031	0.001	0.900
	Predicted error	4	0.099	0.100	0.800
GRU – RNN	Predicted value	69	0.031	0.001	0.900
	Predicted error	4	0.099	0.100	0.800
LSTM – GRU	Predicted value	70	0.003	0.001	0.800
	Predicted error	1	0.082	0.001	0.700
GRU – LSTM	Predicted value	91	0.013	0.001	0.900
	Predicted error	4	0.099	0.100	0.800

ตารางผนวกที่ 89 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะสั้น 7 วันของ SMT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	4.8793	0.2211	3,333	4.5784	0.2115	5,666
LSTM – RNN	8.7084	0.3503	3,377	5.6952	0.2518	5,902
RNN – GRU	4.7584	0.2121	3,361	5.3486	0.2371	5,993
GRU – RNN	5.5786	0.2391	4,174	4.4518	0.2063	6,109
LSTM – GRU	5.3599	0.2413	4,220	4.7806	0.2231	5,956
GRU – LSTM	5.2597	0.2346	4,182	4.6529	0.2156	5,925

ตารางผนวกที่ 90 แสดงประสิทธิภาพของ Hybrid Deep Learning Model ในการพยากรณ์ระยะยาว 30 วันของ SMT

Model	Stacked layers			Cascaded		
	MAPE (%)	RMSE	Runtime (Seconds)	MAPE (%)	RMSE	Runtime (Seconds)
RNN – LSTM	10.5656	0.4471	3,342	12.8612	0.5291	5,733
LSTM – RNN	15.0327	0.5849	3,340	10.8154	0.4577	4,016
RNN – GRU	9.7700	0.4244	559	11.4272	0.4556	4,197
GRU – RNN	8.5098	0.3914	4,170	11.1171	0.4537	4,446
LSTM – GRU	15.4376	0.6143	4,215	10.9181	0.4618	4,410
GRU – LSTM	12.3012	0.5157	4,147	10.1014	0.4376	5,580

สามารถเข้าถึงตัวอย่างชุดคำสั่งและรหัส (Code) ที่ใช้ในวิทยานิพนธ์นี้ได้ที่ GitHub:

https://github.com/Kittikun0851/Thesis_Coding.git

ประวัติผู้เขียน

ชื่อ-สกุล กิตติคุณ ทัดประดิษฐ์
วัน เดือน ปี เกิด 3 พฤศจิกายน 2543
วุฒิการศึกษา เศรษฐศาสตรบัณฑิต มหาวิทยาลัยธรรมศาสตร์
ที่อยู่ปัจจุบัน 72 ซอยเฉลิมพระเกียรติ ร.9 ซอย 30 แยก 1 แขวงหนองบอน เขตประเวศ^{กรุงเทพมหานคร 10250}



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY