



โครงการวิศวกรรมคอมพิวเตอร์

เรื่อง

ระบบทำนายราคาและแนวโน้มของหุ้น
Stock Price And Trend Prediction System

โดย

นาย ฌานณโชติณ บัญเซียว รหัสนักศึกษา 6010504694

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

ปีการศึกษา 2563

ระบบทำนายราคาและแนวโน้มของหุ้น
Stock Price And Trend Prediction System

นาย ฌานณโชชน์ บุญเขียว
รหัสนักศึกษา 6010504694

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

ได้รับการพิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษา วันที่ เดือน พ.ศ.
(ผศ. ดร. ยอดเยี่ยม ทิพย์สุวรรณ)

หัวหน้าภาควิชา วันที่ เดือน พ.ศ.
(รศ. ดร. พันธุ์ปิติ เปี่ยมสง่า)

นาย ภูวนัยโชติณ บัญชีเยว ปีการศึกษา 2563

ระบบทำนายราคาและแนวโน้มของหุ้น

วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

ราคาของหุ้น (Stock Price) เป็นตัวเลขที่บ่งบอกถึงมูลค่าของหลักทรัพย์ประเภทหุ้น หากการที่จะเริ่มคิดที่จะซื้อหรือลงทุนหลักทรัพย์ประเภทหุ้น สิ่งที่มีแนวโน้มว่าจะดูกันคือ ราคาของหุ้น ซึ่งราคาจะมีการเพิ่มขึ้นหรือลดลงตลอดเวลา หากสามารถคาดการณ์ว่าราคาหุ้นที่เราจะซื้อ ณ ตอนนี้จะสามารถมีราคาที่เพิ่มขึ้นหรือลดลงในอนาคต จะทำให้สามารถคาดการณ์ได้ว่าหุ้นที่ซื้อไปจะทำไมหรือทำให้ขาดทุนได้ในอนาคต ทำให้ความเสี่ยงในการเล่นหุ้นของมือใหม่ที่ยังไม่รู้ว่าจะเริ่มต้นซื้อหุ้นอย่างไรดีให้สามารถทำกำไรได้ และจึงเป็นที่มาของการสร้างแบบจำลองที่สามารถคาดการณ์ราคาของหุ้นล่วงหน้า เพื่อให้ได้มาซึ่งค่าที่เรียกว่าราคาของหุ้นและเพื่อลดความเสี่ยงที่จะมีโอกาสขาดทุนจากการซื้อหุ้น

ในโครงงานวิจัยนี้ จะทำการนำราคาในอดีตและปัจจุบัน รวมถึงหุ้นหรือปัจจัยอื่นๆ ที่คาดว่าจะส่งผลต่อการเปลี่ยนแปลงของราคาหุ้น มาทำนายราคาหุ้นและแนวโน้มในอนาคตโดยใช้แบบจำลองที่การเรียนรู้ทางสถิติคือ VAR model และ แบบจำลองการเรียนรู้แบบเชิงลึก (DL) ผสมกับแบบจำลองการเรียนรู้ของเครื่องจักร (ML) คือ CNN+LSTM model เพื่อเปรียบเทียบความแม่นยำของการทำนายราคาและเทรนด์ในแต่ละแบบจำลอง โดยการใช้ Root mean square deviation (RMSE) เพื่อบอกความคลาดเคลื่อนเป็นหน่วยตั้งต้นของราคาหุ้น และ Mean absolute percentage error (MAPE) เพื่อบอกความคลาดเคลื่อนเป็นเปอร์เซ็นต์ หากโมเดลใดมีค่าดังกล่าวน้อยกว่า นับว่าการทำนายราคาหุ้นในโมเดลนั้นมีความแม่นยำมากกว่า

ความแม่นยำในการทำงานของโมเดลดังกล่าวขึ้นอยู่กับการใช้พารามิเตอร์และองค์ประกอบ ซึ่งความแม่นยำสามารถเปลี่ยนแปลงได้หากมีการเปลี่ยนองค์ประกอบที่ใช้เทรน VAR Model หรือ เปลี่ยนจำนวนรอบพารามิเตอร์ รวมถึงการเพิ่ม layer ของการเทรนใน CNN+LSTM Model

Channachot Boonkeaw , Academic Year 2020

Stock Market And Trend Prediction System

Bachelor of Engineering, Department of Computer Engineering

Faculty of Engineer, Kasetsart University

Abstract

Stock Price is a number that indicates the value of stock security. Most people who need to invest or trade stock securities have to face problems .We found that beginners will focus on price because price is the first that they see and easy to access. Stock price can increase or decrease all the time. If we can forecast stock prices in the future,We know the stock that we will buy is profitable or not ,resulting in investment risk or trading risk will decrease. Thus the price and trend prediction system was researched to investigate price and trend stock in the future and reduce the investment risk.

This project will bring past and present prices ,including stocks or other factors that affect price stock change to forecast price and trend stock in the future using the model. The system uses Statistical Model (VAR Model) and Machine Learning Model (LSTM) combined with Deep Learning Model (CNN) to compare performance of accuracy between VAR Model and CNN+LSTM Model. By using Root mean square deviation (RMSE) to indicate the deviation as the starting unit of the stock price and Mean absolute percentage error (MAPE) to indicate the deviation as a percentage. If any model has lower RMSE and MAPE values, it is more accurate to predict stock prices.

Precision of models depends on the use of parameters and elements. Accuracy can be changed if the elements that are used to train the VAR Model are changed, or the number of laps, the parameter and increasing the layer of training in the CNN + LSTM Model.

กิตติกรรมประกาศ

ในการทำโครงงานวิจัยนี้ให้สำเร็จลุล่วงได้ เป็นเพราะมาจากความอนุเคราะห์ของอาจารย์ยอดเยี่ยม ทัพย์สุวรรณ ที่ปรึกษาโครงงาน ที่คอยให้คำแนะนำ ข้อเสนอแนะ วิธีการแก้ปัญหา รวมถึงการตรวจสอบข้อมูล และวิธีการต่างๆ ที่ทำให้โครงงานออกมาสำเร็จ ขอกราบขอบพระคุณอาจารย์ยอดเยี่ยมเป็นอย่างสูง ขอขอบคุณภาควิชาที่เอื้ออำนวยสถานที่ทำโครงงาน และขอขอบคุณ machinelearningplus.com ที่ความรู้ของแบบจำลองต่างๆ

นาย ฌานณโชตน์ บุญเขียว

ผู้จัดทำโครงงาน

สารบัญ

บทคัดย่อ	i
Abstract	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญภาพ	vii
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตโครงการ	2
1.4 ระยะเวลาในการดำเนินการ	2
1.5 ประโยชน์ที่ได้รับ	2
1.6 แนวทางการประเมิน	2
1.7 คำนิยามศัพท์เฉพาะ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 Machine Learning	4
2.1.1 Supervised Learning	5
2.1.2 Unsupervised Learning	5
2.1.3 Self-supervised Learning	5
2.1.4 Reinforcement Learning	5
2.2 Deep learning	6
2.3 Regression	7
2.4 Vector Autoregression (VAR)	9
2.5 Convolutional neural network (CNN)	11
2.6 Long Short-Term Memory (LSTM)	16

2.6.1 Forget	17
2.6.2 Write	17
2.6.3 update cell state	18
2.6.4 Read	18
บทที่ 3 วิธีการดำเนินงาน	20
3.1 เครื่องมือและข้อมูลที่ใช้ดำเนินการ	20
3.1.1 ฮาร์ดแวร์ (Hardware)	20
3.1.2 ซอฟต์แวร์ (Software)	20
3.1.3 ข้อมูล (Data)	21
3.1.4 ไลบรารี (Library)	22
3.2 การออกแบบระบบ	23
3.3 พัฒนาระบบ	25
3.3.1 การจัดเตรียมข้อมูล	25
3.3.2 วิเคราะห์ความสัมพันธ์ระหว่างหุ่นและปัจจัย	26
3.3.3 แบ่งข้อมูลสำหรับ การ Train และ Test	27
3.3.4 ตั้งค่าและปรับแต่ง Model และทำการเทรน	28
3.3.5 ทำนายราคา	29
3.3.6 คำนวณและเปรียบเทียบเปอร์เซ็นต์และค่าเงินที่คลาดเคลื่อน	
จากการทำนายราคาระหว่างโมเดลเทียบกับของจริง	29
3.3.7 สร้างแนวโน้มจากราคาที่ทำนายได้เปรียบเทียบกับแนวโน้มของจริง	30
3.3.8 เปรียบเทียบแนวโน้มที่ได้จากการทำนายราคา	
ระหว่างโมเดลเทียบกับของจริง	30
บทที่ 4 ผลการดำเนินโครงการ	31
4.1 ผลการออกแบบ	31
4.2 ผลการทำนายด้วยแบบจำลอง	32

4.2.1 Forecast Price	32
4.2.2 Forecast Price Trend	37
4.2.3 Forecast Price Error	40
4.3 ผลการประเมินแบบจำลอง	43
4.3.1 Forecast Price Error (MAPE)	43
4.3.1 Forecast Price Error (RSME)	44
บทที่ 5 สรุปผลการทดลอง	45
5.1 สรุปผลการพัฒนาโครงงาน	45
5.1.1 ปัจจัยที่ส่งต่อการทำนายใน VAR Model	45
5.1.2 ปัจจัยที่ส่งต่อการทำนายใน CNN+LSTM	45
5.1.3 ผลการประเมินจากการทำนายราคาของ 2 แบบจำลอง	45
5.2 ปัญหา/อุปสรรค และวิธีการดำเนินการแก้ไข	45
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	46
ภาคผนวก	47

สารบัญภาพ

รูปที่ 1.1	สมการการคำนวณ MAPE	3
รูปที่ 1.2	สมการการคำนวณ RMSE	3
รูปที่ 2.1	ประเภทของการฝึกฝนของการเรียนรู้	4
รูปที่ 2.2	โครงสร้างที่แสดง layer ใน deep learning	6
รูปที่ 2.3	กระบวนการทำงานของ deep learning	7
รูปที่ 2.4	ตัวอย่างข้อมูลของ Total marketing budget และ sales	8
รูปที่ 2.5	รูปตัวอย่าง simple linear regression ระหว่าง Sales กับ Total budget	8
รูปที่ 2.6	Equation : Linear regression model, Y is approximate as modeled of X	8
รูปที่ 2.7	รูปสมการทั่วไปใน VAR Model	9
รูปที่ 2.8	เงื่อนไขข้อผิดพลาดของการหา et	10
รูปที่ 2.9	ตัวอย่าง ค่าของ y ณ เวลา t=2	10
รูปที่ 2.10	การหาค่า y ณ เวลา t=1 โดยอิงจาก y,t=2	10
รูปที่ 2.11	สมการของ VAR ที่ประกอบด้วยตัวแปรหลายชนิด	10
รูปที่ 2.12	กระบวนการและขั้นตอนในการคำนวณ Convolutional	11
รูปที่ 2.13	สมการการหาค่า hidden layer	12
รูปที่ 2.14	แผนภาพโครงสร้างของ Neural network	12
รูปที่ 2.15	Pooling layer	13
รูปที่ 2.16	Max Pooling and AVG Pooling	13
รูปที่ 2.17	ตัวอย่างโมเดล ConvNet	14
รูปที่ 2.18	ตัวอย่างการทำ Conv layer and Max pooling	15
รูปที่ 2.19	สมการการหาค่า z	15
รูปที่ 2.20	กระบวนการทำงานของ LSTM	16
รูปที่ 2.21	สมการการคำนวณหาค่า ft ในส่วนของการจดจำข้อมูล	17
รูปที่ 2.22	สมการการคำนวณหาค่า it ในส่วนของการเขียนข้อมูลเพื่ออัปเดตสเตต	17
รูปที่ 2.23	สมการการหาค่าที่จะถูกเขียน	18
รูปที่ 2.24	สมการการอัปเดตข้อมูล	18
รูปที่ 2.25	สมการการอนุญาตการอ่านข้อมูล	18
รูปที่ 2.26	สมการการหาค่า hidden state	19
รูปที่ 3.1	การออกแบบขั้นตอนการทำงาน	24
รูปที่ 3.2	ตารางแสดงข้อมูลของหุ้น	25
รูปที่ 3.3	ค่า correlation ของหุ้น BTS	26
รูปที่ 3.4	ตัวอย่างแสดงว่าข้อมูลไม่เป็น Stationary	26

รูปที่ 3.5 ตัวอย่างแสดงว่าข้อมูลเป็น Stationary	27
รูปที่ 3.6 ข้อมูลในการใช้ Train	27
รูปที่ 3.7 ข้อมูลในการใช้ Test	27
รูปที่ 3.8 สมการที่ใช้การในแบบจำลอง VAR	28
รูปที่ 3.9 ลำดับ layer ในการ CNN+LSTM Model	28
รูปที่ 4.1 ลำดับและรูปแบบของ layer ในการ Train CNN+LSTM Model	31
รูปที่ 4.2 การทำนายราคาของหุ้น BTS โดยใช้ VAR Model	32
รูปที่ 4.3 การทำนายราคาของหุ้น BTS โดยใช้ CNN+LSTM Model	32
รูปที่ 4.4 การทำนายราคาของหุ้น CPF โดยใช้ VAR Model	33
รูปที่ 4.5 การทำนายราคาของหุ้น CPF โดยใช้ CNN+LSTM Model	33
รูปที่ 4.6 การทำนายราคาของหุ้น PTTEP โดยใช้ VAR Model	34
รูปที่ 4.7 การทำนายราคาของหุ้น PTTEP โดยใช้ CNN+LSTM Model	34
รูปที่ 4.8 การทำนายราคาของหุ้น SCC โดยใช้ VAR Model	35
รูปที่ 4.9 การทำนายราคาของหุ้น SCC โดยใช้ CNN+LSTM Model	35
รูปที่ 4.10 การทำนายราคาของหุ้น TVO โดยใช้ VAR Model	36
รูปที่ 4.11 การทำนายราคาของหุ้น TVO โดยใช้ CNN+LSTM Model	36
รูปที่ 4.12 การเปรียบ BTS Trend ที่ได้จากการทำนายเทียบกับของจริง	37
รูปที่ 4.13 การเปรียบ CPF Trend ที่ได้จากการทำนายเทียบกับของจริง	37
รูปที่ 4.14 การเปรียบ PTTEP Trend ที่ได้จากการทำนายเทียบกับของจริง	38
รูปที่ 4.15 การเปรียบ SCC Trend ที่ได้จากการทำนายเทียบกับของจริง	38
รูปที่ 4.16 การเปรียบ TVO Trend ที่ได้จากการทำนายเทียบกับของจริง	39
รูปที่ 4.17 ค่าคลาดเคลื่อน BTS Price ที่ได้จากการทำนายราคา (%)	40
รูปที่ 4.18 ค่าคลาดเคลื่อน CPF Price ที่ได้จากการทำนายราคา (%)	40
รูปที่ 4.19 ค่าคลาดเคลื่อน PTTEP Price ที่ได้จากการทำนายราคา (%)	41
รูปที่ 4.20 ค่าคลาดเคลื่อน SCC Price ที่ได้จากการทำนายราคา (%)	41
รูปที่ 4.21 ค่าคลาดเคลื่อน TVO Price ที่ได้จากการทำนายราคา (%)	42
รูปที่ 4.22 ค่า MAPE ของการทำนายราคา VAR Model และ CNN+LSTM Model ของทุกหุ้น	43
รูปที่ 4.23 ค่า RMSE ของการทำนายราคา VAR Model และ CNN+LSTM Model ของทุกหุ้น	44

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

การที่จะเลือกลงทุนอะไรสักอย่างหนึ่งในชีวิต สิ่งที่คุณพูดถึงกันบ่อยๆคือลงทุนในหุ้น โดยการที่เรานำเงินของเราไปซื้อหุ้น โดยเราจะได้รับผลตอบแทนเป็นเงินปันผลตามกำไรและจำนวนหุ้นที่เราถืออยู่ ซึ่งเรียกได้ว่าเป็นตลาดที่มีการเติบโตตลอดเวลา มีคนเข้าออกวงการนี้เป็นจำนวนมาก บางคนประสบความสำเร็จในการลงทุน แต่บางคนล้มเหลวเพราะไม่มีความรู้ที่มากพอ หรือโชคร้ายเกิดอุบัติเหตุที่คาดไม่ถึง โดยตัวหุ้นที่จะมีการแบ่งออกเป็นหลายประเภท หลายหมวดหมู่ อาจแบ่งได้ตามการเจริญเติบโตของหุ้น แบ่งตามองค์ประกอบและปัจจัยที่ทำให้หุ้นเติบโต แบ่งตามรูปแบบเงินปันผล ซึ่งความหลากหลายนี้นำมาสู่ปัจจัยหลักในการตัดสินใจลงทุนในรูปแบบหุ้น

หากคนทั่วไปที่ไม่มีความรู้ด้านเศรษฐศาสตร์ ไม่รู้จักคำว่าหุ้น ไม่รู้จักการลงทุน สนใจต้องการลงทุนขึ้นมา สิ่งที่จะที่พบเจอคือ อุปสรรค ไม่ว่าจะเป็นด้าน การเรียนรู้ การเริ่มต้นลงทุน การแก้ไขปัญหา เป็นเหตุให้ผู้เล่นหุ้นมือใหม่ไม่สามารถตัดสินใจได้ว่าจะเลือกลงทุนหุ้นตัวใด ลงทุนในรูปแบบไหน อีกทั้งค่าต่างๆที่บ่งบอกถึงตัวหุ้นก็ฟังดูเข้าใจยากไม่เหมาะกับมือใหม่ หากลงทุนแล้วไม่สามารถติดตามการเติบโตของหุ้น ไม่สามารถวิเคราะห์การเติบโตของหุ้นและความเป็นไปได้ของความเสี่ยง อีกทั้งไม่สามารถรู้ราคาที่จะเกิดขึ้นในอนาคต ทำให้เกิดความไม่มั่นใจในตัวหุ้น และเสียผลประโยชน์หลายอย่างกับตัวผู้ถือหุ้น

การคำนวณการเติบโตและทำนายหุ้นล่วงหน้ามีความก้าวหน้าอย่างมากในปัจจุบัน เราสามารถดึงความรู้ของเรื่อง machine learning , statistics และอื่นๆอีกมากมาย มาใช้กับการทำนายความเป็นไปได้ของหุ้นล่วงหน้า อีกทั้งยังสามารถนำไปใช้วิเคราะห์ตัวหุ้น

การทำนายราคาของหุ้นมีความสำคัญในการตัดสินใจที่ลงทุนต่อหรือขายทิ้ง ด้วย model ที่หลากหลายที่จะเข้ามามีบทบาทในการช่วยให้เราสามารถทำนายราคาล่วงหน้าได้ด้วยความรู้ของ Deep learning

ผู้จัดทำจึงเห็นความสำคัญของการศึกษาและพัฒนาระบบการจัดการและทำนายการเติบโตและราคาของหุ้น ซึ่งเป็นการนำความรู้มาประยุกต์ใช้กับชีวิตจริงและเพิ่มประสิทธิภาพของระบบให้มากขึ้น

1.2 วัตถุประสงค์

- 1 เพื่อทำนายราคาของหุ้นที่จะเกิดขึ้นในอนาคต
- 2 เพื่อรับรู้แนวโน้มการเปลี่ยนแปลงของกลุ่มหุ้น
- 3 เพื่อศึกษาและวิเคราะห์การทำงานของ model ที่ใช้เรียนรู้
- 4 เปรียบเทียบประสิทธิภาพระหว่าง model

1.3 ขอบเขตโครงการ

1.3.1 สร้างระบบการทำนายราคาหุ้นและแนวโน้มที่เกิดขึ้นในอนาคต โดยมีการเลือกหุ้นจำนวน 5 ตัว และปัจจัย 5-10 ปัจจัยต่อหุ้น มีการทำนายราคาด้วย VAR model

1.3.2 สร้างระบบการทำนายราคาหุ้นและแนวโน้มที่เกิดขึ้นในอนาคต โดยมีการเลือกหุ้นจำนวน 5 ตัว และมีการทำนายด้วย CNN and LSTM model

1.4 ระยะเวลาการดำเนินการ

สิงหาคม 2563 - มีนาคม 2564 โดยเป็นการทำกิจกรรมต่างๆภายในระยะเวลาที่ดำเนินการ โดยมีกิจกรรมดังนี้

1. ปรึกษากับอาจารย์ที่ปรึกษาเรื่องการทำโครงการและปัญหาที่พบ
2. ศึกษาและพัฒนางานให้ดำเนินการไปอย่างต่อเนื่อง
3. วัดผลและแก้ไขข้อผิดพลาดที่เกิดขึ้นในงานของตัวเอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1 ผู้ใช้สามารถมองเห็นราคาของตัวหุ้นที่กำลังจะเกิดขึ้นได้
- 2 ผู้ใช้สามารถดูแนวโน้มการเติบโตของหุ้นได้
- 3 ผู้ใช้มีความมั่นใจในการลงทุนมากยิ่งขึ้น

1.6 แนวทางการประเมิน

ในการวัดความแม่นยำของแบบจำลองทำโดยการใช้ค่า MAPE และ RSME ซึ่ง MAPE จะให้ความคลาดเคลื่อนของการทำนายเทียบกับของจริงเป็นเปอร์เซ็นต์ ส่วน RSME จะให้ค่าที่ทำนายผิดพลาดออกมาตามหน่วยของการทำนายซึ่งมีความลำเอียงมากกว่า ค่า MAPE และ RMSE หาได้จากรูปดังต่อไปนี้

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \cdot 100$$

รูปที่ 1.1 สมการการคำนวณ MAPE

$$RMSE = \sqrt{\frac{1}{n} * \sum (prediction - actual)^2}$$

รูปที่ 1.2 สมการการคำนวณ RMSE

1.7 คำนิยามศัพท์เฉพาะ

1. การเรียนรู้ด้วยเครื่องจักรกล (Machine learning) คือส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของ AI (Artificial Intelligence) เราอาจพูดได้ว่า AI ใช้ Machine Learning ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ ไม่ได้เกิดจากการเขียนโดยใช้นุชย์ มนุษย์ มีหน้าที่เขียนโปรแกรมให้ AI (เครื่อง) เรียนรู้จากข้อมูลเท่านั้น

2. การเรียนรู้เชิงลึก (Deep learning) คือวิธีการเรียนรู้แบบอัตโนมัติด้วยการ เลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาท (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูล ดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัด หมวดหมู่ข้อมูล (Classify the Data)

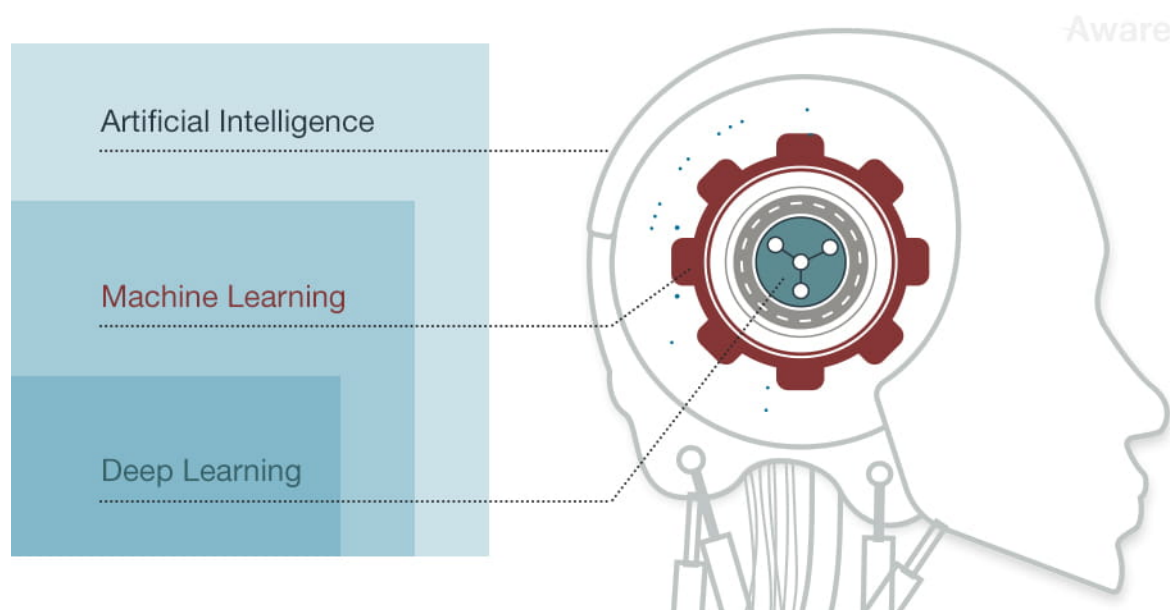
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Machine Learning

Machine Learning คือ ส่วนการเรียนรู้ของเครื่อง ถูกใช้งานเสมือนเป็นสมองของ AI (Artificial Intelligence) เราอาจพูดได้ว่า AI ใช้ Machine Learning ในการสร้างความฉลาด มักจะใช้เรียกโมเดลที่เกิดจากการเรียนรู้ของปัญญาประดิษฐ์ ไม่ได้เกิดจากการเขียนโดยใช้มนุษย์ มนุษย์มีหน้าที่เขียนโปรแกรมให้ AI (เครื่อง) เรียนรู้จากข้อมูลเท่านั้น ที่เหลือเครื่องจัดการเอง

Machine Learning เรียนรู้จากสิ่งที่เราส่งเข้าไปกระตุ้น แล้วจดจำเอาไว้เป็นมันสมอง ส่งผลลัพธ์ออกมาเป็นตัวเลข หรือ code ที่ส่งต่อไปแสดงผล หรือให้เจ้าตัว AI นำไปแสดงการกระทำ Machine Learning เองสามารถเอาไปใช้งานได้หลายรูปแบบ ต้องอาศัยกลไกที่เป็นโปรแกรม หรือเรียกว่า Algorithm ที่มีหลากหลายแบบ โดยมี Data Scientist เป็นผู้ออกแบบ หนึ่งใน Algorithm ที่ได้รับความนิยมสูง คือ Deep Learning ซึ่งถูกออกแบบมาให้ใช้งานได้ง่าย และประยุกต์ใช้ได้หลายลักษณะงาน อย่างไรก็ตาม ในการทำงานจริง Data Scientist จำเป็นต้องออกแบบตัวแปรต่างๆ ทั้งในตัวของ Deep Learning เอง และต้องหา Algorithm อื่นๆ มาเป็นคู่เปรียบเทียบกับ เพื่อมองหา Algorithm ที่เหมาะสมที่สุดในการใช้งานจริง



รูปที่ 2.1 ประเภทของการฝึกฝนของการเรียนรู้

2.1.1 Supervised Learning

การสร้างโมเดลเพื่อแปลงข้อมูล input เป็น target บางอย่าง ตัวอย่างง่ายที่สุดคือ classification กับ regression

- Classification คือมี target เป็นชนิดของข้อมูล เช่น เรียนรู้ว่า email เป็น spam หรือไม่ spam
- Regression คือมี target เป็นตัวเลข เช่น เรียนรู้การประมาณราคาที่ดินจากปัจจัยแวดล้อม

การสร้างโมเดลประเภทนี้ขึ้นมา ต้องมีชุดข้อมูลที่มีทั้ง input และ target ซึ่งจัดหามาโดยมนุษย์ เช่น การสร้าง spam filter ต้องรวบรวมข้อมูล email จำนวนมากและให้คนมาดูว่าอันไหนเป็น spam บ้าง แล้วนำมาสร้างโมเดล spam filter จากข้อมูลเหล่านี้

Supervised Learning เป็น Machine Learning ที่ถูกใช้งานมากที่สุด เข้าใจง่ายที่สุด และทุกคนที่เริ่มเรียน Machine Learning ควรเริ่มจาก Supervised Learning

2.1.2 Unsupervised Learning

การสร้างโมเดลโดยใช้ข้อมูล input เพียงอย่างเดียว ไม่มี target การใช้งานหลักมี 2 อย่างคือ

- Dimensionality reduction การลดมิติของข้อมูล เพื่อลดความซับซ้อนก่อนนำไปใช้ต่อ หรือเพื่อแสดงผลในรูปภาพที่คนอ่านได้
- Clustering การจัดกลุ่มข้อมูลตามคุณลักษณะ เช่น การจัดกลุ่มลูกค้าตามพฤติกรรมการซื้อของ

การสร้างโมเดลประเภทนี้ขึ้นมา ใช้เพียงข้อมูล input อย่างเดียว ไม่ต้องจัดหา target เช่น โมเดลการจัดกลุ่มลูกค้า เราไม่ต้องรู้มาก่อนว่าจะมีกลุ่มอะไรบ้าง

2.1.3 Self-supervised Learning

การสร้างโมเดลด้วยวิธี Supervised Learning แต่ใช้ target แบบที่ไม่ต้องพึ่งคน เช่น

- Autoencoders ใช้ target เหมือนกับ input
- การพยากรณ์อากาศในวันถัดไปด้วยข้อมูลในอดีต ใช้ target เป็น input ในอนาคต

ปกติตำราที่แบ่ง Machine Learning เป็น 3 ประเภทจะไม่มีประเภทนี้ ในกรณีนั้นมักจะจัดรวมอยู่ใน Supervised Learning

2.1.4 Reinforcement Learning

การสอน agent ในสภาพแวดล้อมบางอย่าง ให้เรียนรู้วิธีการตัดสินใจที่ให้ผลลัพธ์ที่ดีที่สุด ลองนึกถึงคอมพิวเตอร์ในเกมส์อะไรสักอย่างที่เราพยายามหาทางเอาชนะเรา

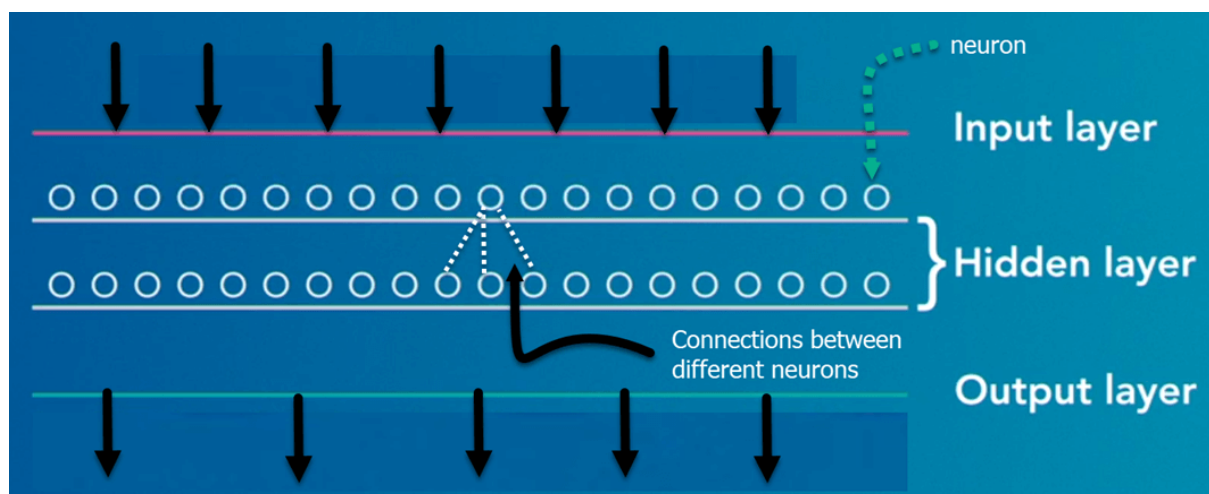
ล่าสุด AlphaGo โปรแกรมแข่งโกะ ซึ่งเป็นผลงานอันยิ่งใหญ่ของ Google DeepMind สามารถเอาชนะ Lee Sedol แชมป์โกะระดับโลก ที่คะแนน 4 ต่อ 1 เกมส์ เมื่อเดือนมีนาคม 2016 โดย AlphaGo ถูกสอนให้เล่นโกะด้วยวิธี Reinforcement Learning

2.2 Deep learning

deep learning คือ ซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท (neurons) ในสมองมนุษย์ ถือเป็นขั้นขั้นของ machine learning

Algorithm ของ deep learning ถูกสร้างขึ้นจากการนำเอา neural network หลายๆ layer มาต่อกัน โดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (Output layer) ส่วน layer ระหว่าง layer แรกสุด และ layer สุดท้าย จะถูกเรียกว่า Hidden layer

คำว่า deep learning มีที่มาจากการใช้ layer ของ neural network หลายอันๆมาต่อกัน (มี Hidden layer มากกว่า 2 layer ก็ถือเป็น deep learning แล้ว) เนื่องจาก layer เหล่านี้เป็นโครงสร้างที่ถูกจัดเก็บแบบเป็นกองซ้อน (stack) จึงเปรียบได้ว่า layer ที่จะนวนเยอะๆ ก็จะทำให้มีโครงสร้างที่ลึก(deep) ยิ่งขึ้นนั่นเอง

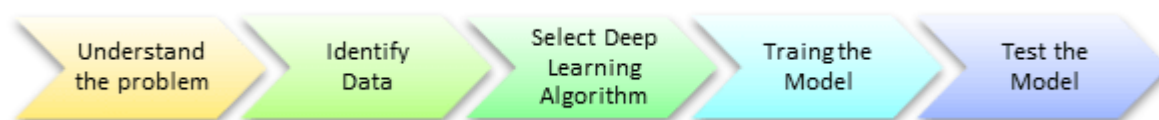


รูปที่ 2.2 โครงสร้างที่แสดง layer ใน deep learning

โดย Hidden layer ของแต่ละ layer จะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท (neural) จำนวนมาก ซึ่งมีหน้าที่ในการประมวลผล รับข้อมูลจาก layer ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยัง layer ที่อยู่ล่างกว่า , ข้อดีของการส่งข้อมูลแบบนี้ก็คือ layer แต่ละ layer สามารถที่จะมี ค่าถ่วงน้ำหนัก (weight), ค่าความเอนเอียงของข้อมูล (bias) และ วิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกันได้

กระบวนการทำงานของ deep learning

model ที่ใช้ deep learning ให้ความแม่นยำ (accuracy) ที่สูงในหลายๆปัญหา ตั้งแต่การตรวจจับวัตถุ (object detection) ไปจนถึงการรู้จำเสียงพูด (speech recognition) โดยที่เราไม่จำเป็นต้องให้ความรู้พื้นฐานใดๆกับมันไว้ล่วงหน้าเลย เพียงแค่ให้ข้อมูลตัวอย่าง (input data) มันก็จะทำการเรียนรู้จากข้อมูลและสังเคราะห์เป็นองค์ความรู้ออกมาได้อย่างอัตโนมัติ อาทิเช่น การใช้ deep learning ในวงการเกม เราไม่จำเป็นต้องบอกมันว่าเล่นยังไง แค่ให้มันเรียนรู้จากผู้เล่นที่เก่งๆเป็นจำนวนมาก มันก็เรียนรู้วิธีการเล่นเกมได้อย่างอัตโนมัติ



รูปที่ 2.3 กระบวนการทำงานของ deep learning

การเรียนรู้ขึ้น เกิดขึ้นจาก 2 เฟส

เฟสแรกคือ การประยุกต์ใช้ การแปลงแบบไม่เป็นเชิงเส้น (nonlinear transformation) กับข้อมูลที่ได้รับ (input) ได้ผลลัพธ์ (output) ออกมาอยู่ในรูปของโมเดลทางสถิติ (statistical model)

เฟสที่สอง คือ การนำโมเดลมาผ่านวิธีการทางคณิตศาสตร์ อย่าง derivative หรือ การดิฟ นั้นเอง

โดยทั้ง 2 เฟสนี้จะถูกทำซ้ำแล้วซ้ำเล่าจนกว่าจะได้มาซึ่งโมเดลที่ความแม่นยำ(accuracy)ในระดับที่น่าพึงพอใจ ซึ่ง การทำซ้ำๆของ 2 เฟสนี้ มีชื่อว่า iteration

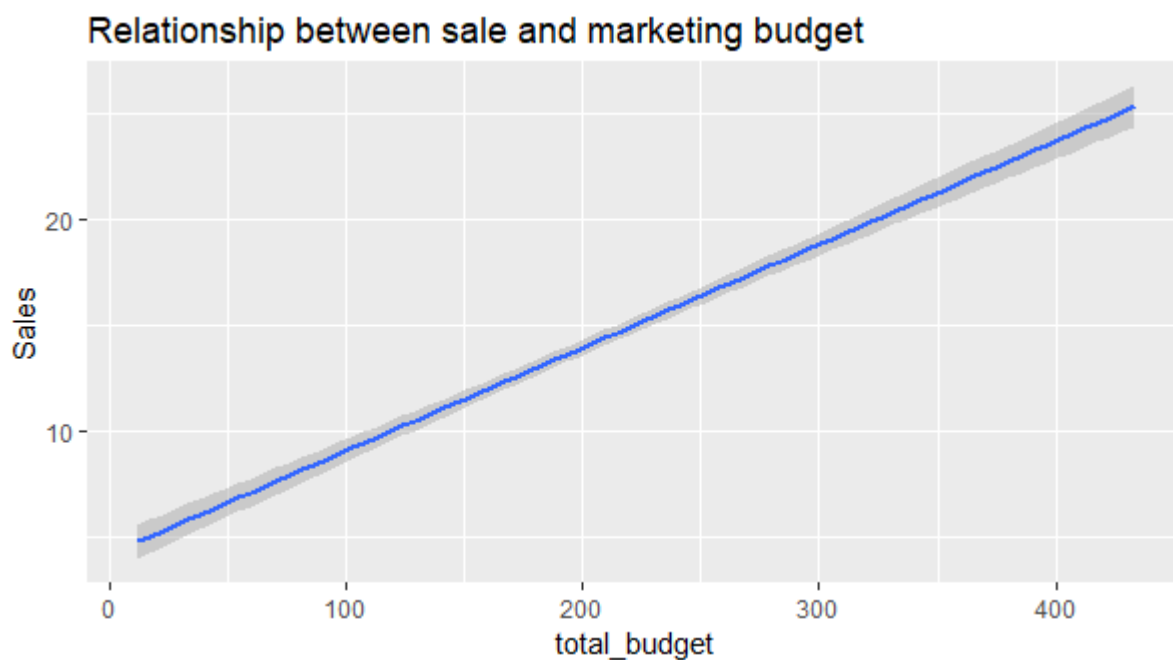
2.3 Regression

Linear regression เป็นโมเดลความสัมพันธ์ระหว่างตัวแปรกับสิ่งที่เราสนใจเช่นปัจจัยต่าง ๆ ทางธุรกิจกับยอดขายในไตรมาสนั้น ๆ หรืออาจจะเป็นส่วนประกอบต่าง ๆ ในสินค้ากับเกรดหรือคุณภาพของสินค้าเป็นต้น โดย linear regression นั้นเป็นโมเดลที่ได้รับความนิยมมากที่สุด เนื่องจากเป็นโมเดลที่เข้าใจง่าย อีกทั้งยังสามารถนำไปใช้งานจริงได้อย่างรวดเร็วอีกด้วย แต่ว่ามีข้อควรระวังอยู่มาก หากไม่เข้าใจหลักการของ linear regression จริง ๆ แล้ว อาจจะนำไปสู่ความผิดพลาดได้ง่าย ๆ เลย

Simple linear regression คือรูปแบบความสัมพันธ์แบบ 1 ตัวแปร เช่นความสัมพันธ์ระหว่างเงินทุนด้านการตลาดกับยอดขาย โดยมีรูปแบบความสัมพันธ์แบบตรงไปตรงมาเช่นเพิ่มเงินทุนด้านการตลาดไป 100 หน่วยจะเพิ่มยอดขายไปอีกประมาณ 9 หน่วยและถ้าเพิ่มเงินทุนด้านการตลาดไป 200 หน่วยจะเพิ่มยอดขายไปอีกประมาณ 14 หน่วย (เงินทุน 1 หน่วยเท่ากับ หนึ่งพันเหรียญ และ 1 หน่วยของยอดขายเท่ากับ หนึ่งพันชิ้น) เป็นต้น

	total_budget	Sales
1	337.1	22.1
2	128.9	10.4
3	132.4	9.3
4	251.3	18.5
5	250.0	12.9
6	132.6	7.2

รูปที่ 2.4 ตัวอย่างข้อมูลของ Total marketing budget และ sales



รูปที่ 2.5 รูปตัวอย่าง simple linear regression ระหว่าง Sales กับ Total budget

ในเชิงคณิตศาสตร์แล้ว ความสัมพันธ์สามารถเขียนอยู่ในรูปแบบสมการดังนี้

$$Y \approx \beta_0 + \beta_1 X$$

รูปที่ 2.6 Equation : Linear regression model, Y is approximate as modeled of X

สมการด้านบนสามารถตีความได้ว่า ผลของ Y จะสามารถประมาณได้จากค่าคงที่ (β_0) รวมกับค่าของตัวแปรที่เราต้องการนำมาวิเคราะห์และความสำคัญของตัวแปรนั้น ๆ (β_1)

ผลของ Y คือค่าที่เราสนใจเป็นหลัก เช่นยอดขาย คุณภาพของสินค้า หรือความต้องการของตลาด โดยตัวแปร Y มีชื่อเรียกที่หลากหลาย ได้แก่ dependent variable, response, หรือ outcome (ในบทความนี้จะเรียกว่า response)

ตัวแปร X คือตัวแปรที่เราต้องการนำมาอธิบายผลของ response Y เช่นตัวแปรที่สามารถอธิบายยอดขายได้ อาจจะเป็นต้นทุนด้านการวิจัย ต้นทุนทางการตลาด ความต้องการของตลาดโดยรวม เป็นต้น โดยตัวแปร X เองก็มีชื่อเรียกหลากหลาย ได้แก่ independent variable, predictor, หรือ explanatory (ในบทความนี้จะเรียกว่า predictor)

ค่าคงที่ β_0 เรียกว่า coefficient หมายถึงค่าที่ response Y จะเป็นเมื่อตัวแปร X มีค่าเป็น 0 หรือไม่มีตัวแปร X นั่นเอง โดยความหมายทางคณิตศาสตร์คือค่าจุดตัดแกน Y (y-intercept)

น้ำหนัก β_1 เป็น parameter ของค่า X หมายถึงค่าที่บอกถึงน้ำหนักของ X ที่ส่งผลต่อ Y ซึ่งยิ่ง β_1 มาก แสดงว่าค่า X สามารถอธิบายผลของ response Y ได้มาก โดย β_1 นั้นมีความหมายทางคณิตศาสตร์คือ slope หรือความชันของสมการเส้นตรง

2.4 Vector Autoregression (VAR)

Vector autoregression (VAR) เป็นแบบจำลองทางสถิติที่ใช้ในการจับความสัมพันธ์ระหว่างปริมาณหลายปริมาณที่เปลี่ยนแปลงตลอดเวลา VAR เป็นรูปแบบกระบวนการสุ่มประเภทหนึ่ง แบบจำลอง VAR ทำให้โมเดลการตอบสนองอัตโนมัติแบบตัวแปรเดียว (univariate) โดยการอนุญาตให้มีอนุกรมเวลาหลายตัวแปร แบบจำลอง VAR มักใช้ในเศรษฐศาสตร์และวิทยาศาสตร์ธรรมชาติ

คำจำกัดความ : แบบจำลอง VAR อธิบายถึงวิวัฒนาการของชุดของตัวแปร k ที่เรียกว่าตัวแปรภายนอกเมื่อเวลาผ่านไป แต่ละช่วงเวลาจะมีหมายเลข $t = 1, \dots, T$ ตัวแปรจะถูกรวบรวมในเวกเตอร์ y_t ซึ่งมีความยาว k (เทียบเท่าเวกเตอร์นี้อาจเรียกว่าเป็นเมทริกซ์ $(k \times 1)$) เวกเตอร์ถูกจำลองเป็นฟังก์ชันเชิงเส้นของค่าก่อนหน้า ส่วนประกอบของเวกเตอร์เรียกว่า y_{it} , t หมายถึงการสังเกตที่เวลา t ของตัวแปร i th ตัวอย่างเช่นหากตัวแปรแรกในแบบจำลองอัตราค่าเช่าบ้านเมื่อเวลาผ่านไป $y_{1,1998}$ จะระบุราคาเช่าบ้านในปี 1998

โมเดล VAR มีลักษณะตามลำดับซึ่งหมายถึงจำนวนช่วงเวลาก่อนหน้านี้ที่โมเดลจะใช้ จากตัวอย่างข้างต้น VAR ลำดับที่ 5 จะจำลองราคาเช่าบ้านในแต่ละปีโดยเป็นการรวมเชิงเส้นของราคาเช่าบ้านในช่วงห้าปีที่ผ่านมา ความล่าช้าคือค่าของตัวแปรในช่วงเวลาก่อนหน้า ดังนั้นโดยทั่วไป VAR ลำดับ p หมายถึงรูปแบบ VAR ซึ่งรวมถึงความล่าช้าของช่วงเวลา p สุดท้าย VAR ลำดับ p th จะแสดงเป็น "VAR (p)" และบางครั้งเรียกว่า "VAR ที่มี p ล่าช้า" โมเดล VAR ลำดับ p th เขียนเป็น

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \dots + A_p y_{t-p} + e_t,$$

รูปที่ 2.7 รูปสมการทั่วไปใน VAR Model

ตัวแปรของรูปแบบ $y_t - i$ ระบุว่าค่าของตัวแปร i ช่วงเวลาก่อนหน้านี้และเรียกว่า "ith lag" ของ y_t ตัวแปร c คือเวกเตอร์ k ของค่าคงที่ที่ทำหน้าที่เป็นตัวสัดกั๊กกันของแบบจำลอง A_i เป็นเมทริกซ์ไม่แปรผันตาม

เวลา $(k \times k)$ และ e_t คือเวกเตอร์ k ของเงื่อนไขความผิดพลาด เงื่อนไขข้อผิดพลาดต้องเป็นไปตามเงื่อนไขสามประการ:

$$E(e_t) = 0$$

$$E(e_t e_t') = \Omega$$

$$E(e_t e_{t-k}') = 0$$

รูปที่ 2.8 เงื่อนไขข้อผิดพลาดของการหา e_t

VAR ที่มี p (lag order) สามารถเขียน VAR ได้เสมอโดยมี lag order เพียงครั้งเดียวโดยการกำหนดตัวแปรตามใหม่อย่างเหมาะสม การแปลงจะมีค่าเท่ากับการซ้อนกันของ lag order ของตัวแปร VAR (p) ในตัวแปรตาม VAR (1) ใหม่และการต่อท้ายข้อมูลประจำตัวเพื่อเติมเต็มจำนวนสมการ

VAR(2) model

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + e_t$$

รูปที่ 2.9 ตัวอย่าง ค่าของ y ณ เวลา $t=2$

จะเขียน VAR(1) ได้เป็น

$$\begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} A_1 & A_2 \\ I & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} e_t \\ 0 \end{bmatrix},$$

รูปที่ 2.10 การหาค่า y ณ เวลา $t=1$ โดยอิงจาก $y, t=2$

(VAR) ขยายแนวคิดของการตอบสนองอัตโนมัติแบบไม่แปรผันไปยังการถดถอยของอนุกรมเวลา โดยที่ค่าล่าช้าของทั้งหมด k ซีรีส์จะปรากฏเป็นตัวถอยหลัง ใส่ให้แตกต่างกันในรูปแบบ VAR เราจะถอยหลังเวกเตอร์ของตัวแปรอนุกรมเวลาบนเวกเตอร์ที่ล่าช้าของตัวแปรเหล่านี้ สำหรับ AR (p) แบบจำลองลำดับ lag order แสดงโดย p ดังนั้น VAR (p) แบบจำลองของสองตัวแปร X_t และ Y_t ($k=2$) จะให้สมการดังนี้

$$\begin{aligned} Y_t &= \beta_{10} + \beta_{11}Y_{t-1} + \cdots + \beta_{1p}Y_{t-p} + \gamma_{11}X_{t-1} + \cdots + \gamma_{1p}X_{t-p} + u_{1t}, \\ X_t &= \beta_{20} + \beta_{21}Y_{t-1} + \cdots + \beta_{2p}Y_{t-p} + \gamma_{21}X_{t-1} + \cdots + \gamma_{2p}X_{t-p} + u_{2t}. \end{aligned}$$

รูปที่ 2.11 สมการของ VAR ที่ประกอบด้วยตัวแปรหลายชนิด

β s and γ s สามารถประมาณได้โดยใช้ OLS ในแต่ละสมการ สมมติฐานสำหรับ VAR คือสมมติฐานอนุกรมเวลา

เป็นเรื่องง่ายที่จะประมาณแบบจำลอง VAR ใน R วิธีการที่เป็นไปได้คือเพียงแค่อ้างอิง `lm()` สำหรับการประมาณสมการแต่ละสมการ นอกจากนี้ Rpackage `vars` ยังมีเครื่องมือมาตรฐานสำหรับการประมาณค่าการทดสอบวินิจฉัยและการทำนายโดยใช้แบบจำลองประเภทนี้

2.5 Convolutional neural network (CNN)

Convolution (สัญลักษณ์ $*$) = การดำเนินการทางคณิตศาสตร์รูปแบบหนึ่ง ระหว่างสองฟังก์ชัน $f * g$ เพื่อดูการเปลี่ยนแปลงของฟังก์ชัน (f) เมื่อมีฟังก์ชัน (g) เข้ามา

Convolution ถูกนำมาใช้อย่างกว้างขวางทั้งในคณิตศาสตร์, สถิติ, ประมวลผลสัญญาณ (Signal processing) รวมถึง การประมวลผลภาพ (Computer vision)

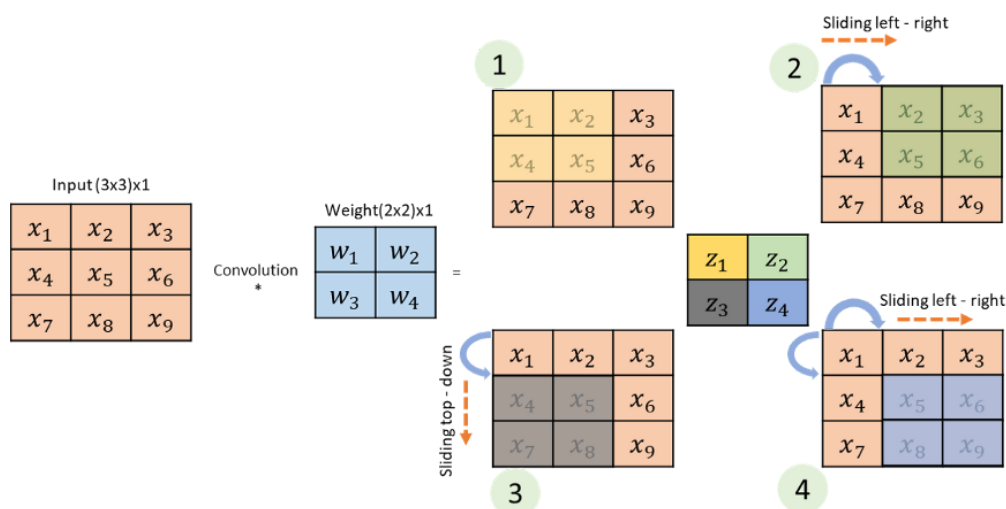
ในยุคของ Classic machine learning สำหรับการประมวลผลภาพนั้น convolution ถูกนำมาใช้อย่างมากในกระบวนการจำแนกลักษณะเด่นของรูปภาพ (Feature extraction)

ในยุคของ Deep learning หรือ Neural network สำหรับการประมวลผลภาพนั้น convolution ถูกนำมาใช้เช่นกัน จึงถูกเรียกว่า Convolutional neural network (ConvNet) โดยนำการทำ Convolution มาใช้ในรูปแบบโมเดลที่สามารถเรียนรู้และเลือกใช้ลักษณะเด่น (Feature extraction) ของรูปภาพได้ด้วยตัวเอง

ข้อดีคือ โมเดลจะเรียนรู้และเลือกลักษณะเด่นของภาพได้ดีกว่าเราทำเอง ดังนั้นจะทำให้ได้การประมวลผลที่แม่นยำมากกว่า

ตัวอย่าง คำนวณ Convolutional

กำหนด Input ขนาด $3 \times 3 \times 1$ และ Weight ขนาด $2 \times 2 \times 1$



รูปที่ 2.12 กระบวนการและขั้นตอนในการคำนวณ Convolutional

ขั้นตอนการคำนวณคือ หาผลรวมของการคูณระหว่าง Input กับ weight โดยใช้ weight ชุดเดิมแล้วสแกน ไปทั้ง Input จากซ้ายไปขวา และบนลงล่าง (1)-(4) ผลลัพธ์ (Z) ที่ได้คือ

$$Z_1 = w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5$$

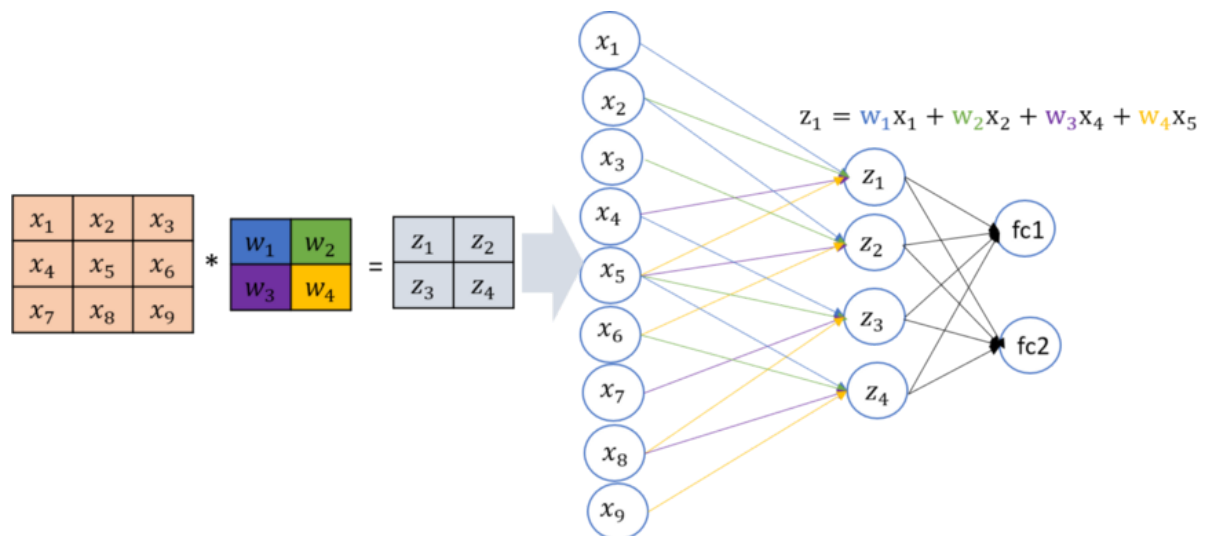
$$Z_2 = w_1x_2 + w_2x_3 + w_3x_5 + w_4x_6$$

$$Z_3 = w_1x_4 + w_2x_5 + w_3x_7 + w_4x_8$$

$$Z_4 = w_1x_5 + w_2x_6 + w_3x_8 + w_4x_9$$

รูปที่ 2.13 สมการการหาค่า hidden layer

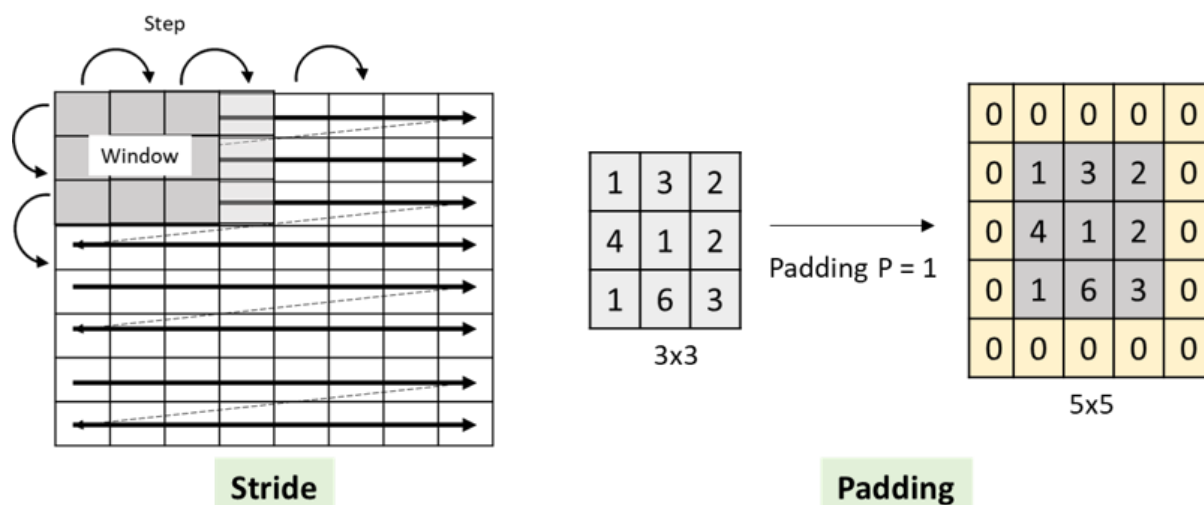
เขียนในรูปแบบ Neural network จะได้ว่า



รูปที่ 2.14 แผนภาพโครงสร้างของ Neural network

ConvNet ประกอบไปด้วย layer 2 แบบคือ Convolutional layer และ Pooling layer

Convolutional layer : มีพารามิเตอร์ที่ต้องกำหนด (Hyper parameters) คือ ขนาด weight window, การปรับขนาดอินพุต (Padding) และ สเตปการสแกน (Stride) ตัวอย่าง กำหนด Input ขนาด 3x3



รูปที่ 2.15 Pooling layer

ตั้งค่า Padding = 1 หมายความว่า เราต้องการขยาย Input จาก 3x3 เป็น 5x5 โดยการขยายใช้วิธีเติมค่า 0 ล้อมรอบ Input เดิม เป้าหมายของการกำหนด padding คือ การให้ความสำคัญกับข้อมูลตรงขอบของ Input

ตั้งค่า Stride = 1 หมายความว่า เราจะให้ weight สแกน Input ทีละ 1 step จากซ้ายไปขวา และบนลงล่าง เป้าหมายของ Stride คือ กำหนดความละเอียดของการสแกน ยิ่งสแกนน้อย คือ ต้องการความละเอียดมาก

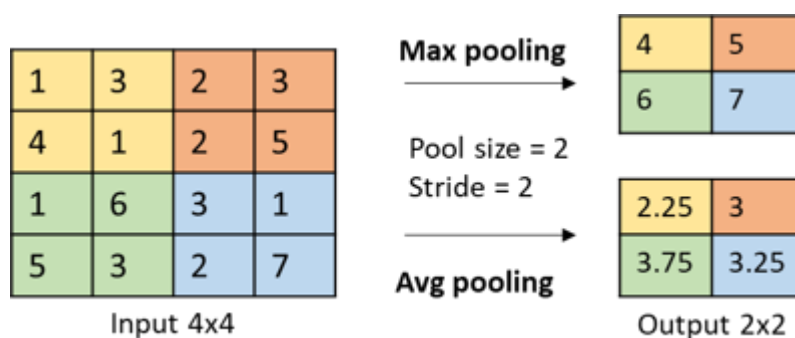
Pooling layer : การปรับขนาดข้อมูล แนวคิดคือ แทนที่เราจะใช้ข้อมูลภาพทั้งหมด เราจะแบ่งเป็นพื้นที่แล้วเลือกบางค่าจากพื้นที่นั้น (เหมือนกับลักษณะการมองของคน ที่บางครั้งมองภาพคร่าวๆ ก็บอกได้ว่าคืออะไร)

โดยทั่วไป Pooling layer มีสองแบบที่เราสามารถเลือกใช้ได้ คือ

Max pooling : เลือกค่าสูงสุดจาก Pool size

Average pooling: ค่าเฉลี่ยของ Pool size

พารามิเตอร์ที่ต้องกำหนด (Hyper parameters) คือ สเตปการสแกน (Stride) และ ขนาดของ Pool ตัวอย่างเช่น มี Input ขนาด 4x4, กำหนดขนาด Pool size = 2 และ stride = 2



รูปที่ 2.16 Max Pooling and AVG Pooling

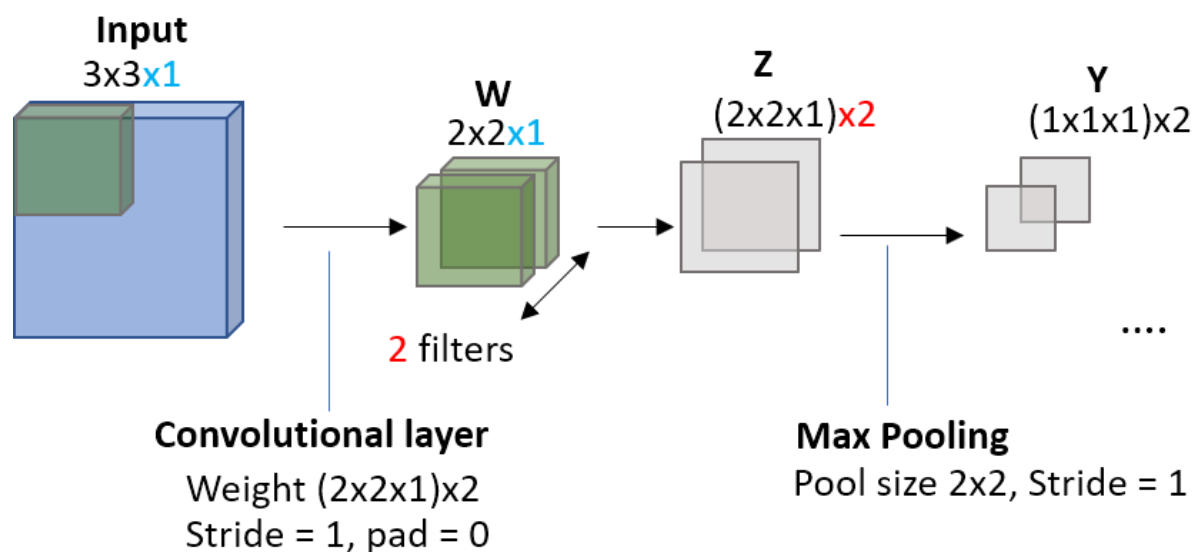
ตัวอย่าง Pool สีเหลือง ผลลัพธ์จะได้ว่า

Max pooling : เลือกค่าสูงสุดใน pool นั้นคือ 4

Average pooling: หาค่าเฉลี่ยใน pool นั้นคือ $(1+3+4+1)/4 = 2.25$

จุดประสงค์ของการทำ Pooling คือ เพื่อลดขนาดของข้อมูล ทำให้โมเดลไม่จำเป็นต้องเรียนรู้ค่า weight จำนวนมากเกินไป

ตัวอย่างโมเดล ConvNet



รูปที่ 2.17 ตัวอย่างโมเดล ConvNet

โมเดลประกอบไปด้วย Convolutional layer และ Pooling layer โดยที่ Input มีขนาด $3 \times 3 \times 1$ และ Output มีขนาด $(1 \times 1 \times 1) \times 2$ หรือ $1 \times 1 \times 2$

Convolutional layer: ค่าพารามิเตอร์ที่กำหนดคือ

Weight ขนาด $2 \times 2 \times 1$ และกำหนดให้มีจำนวน 2 ชุด ในบางครั้งเรียกว่า จำนวนชุดของ weight ว่า จำนวน Filter

กำหนด Stride = 1 และ Padding = 0

Input ของ layer นี้มีขนาด $3 \times 3 \times 1$ และ Output มีขนาด $(2 \times 2 \times 1) \times 2$ หรือ $2 \times 2 \times 2$

ข้อสังเกตคือ ความลึกของ Input กับ ความลึกของ weight จะมีค่าเท่ากันเสมอ (สีน้ำเงิน) และความลึกของ Output จะมีค่าเท่ากับจำนวน Filter (สีแดง)

เนื่องจากความลึกของ weight จะขึ้นอยู่กับ Input ดังนั้นโดยทั่วไป เราจะไม่เขียนความลึกของ weight จะเขียนเพียงแค่ว่า กว้าง \times ยาว และ จำนวน Filter

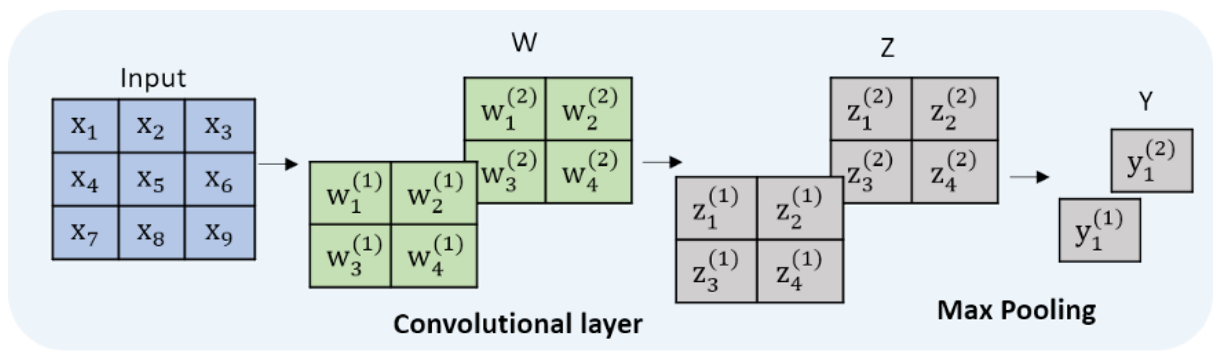
Max pooling layer: ค่าพารามิเตอร์ที่กำหนดคือ

กำหนด Pool size = 2×2 และ Stride = 1

Input ของ layer นี้มีขนาด $2 \times 2 \times 2$ และ Output มีขนาด $(1 \times 1 \times 1) \times 2$ หรือ $1 \times 1 \times 2$

ข้อสังเกตคือ การทำ pooling ไม่มีผลต่อความลึกของ Input และ Output

จากโมเดลข้างต้น เมื่อเขียนในรูปของตัวแปร จะได้ดังต่อไปนี้



รูปที่ 2.18 ตัวอย่างการทำ Conv layer and Max pooling

โดยตัวอย่างคำนวณค่า z_1 จะทำได้ดังนี้

$$z_1^{(1)} = (1)(1) + (0)(2) + (0)(4) + (1)(5) = 6$$

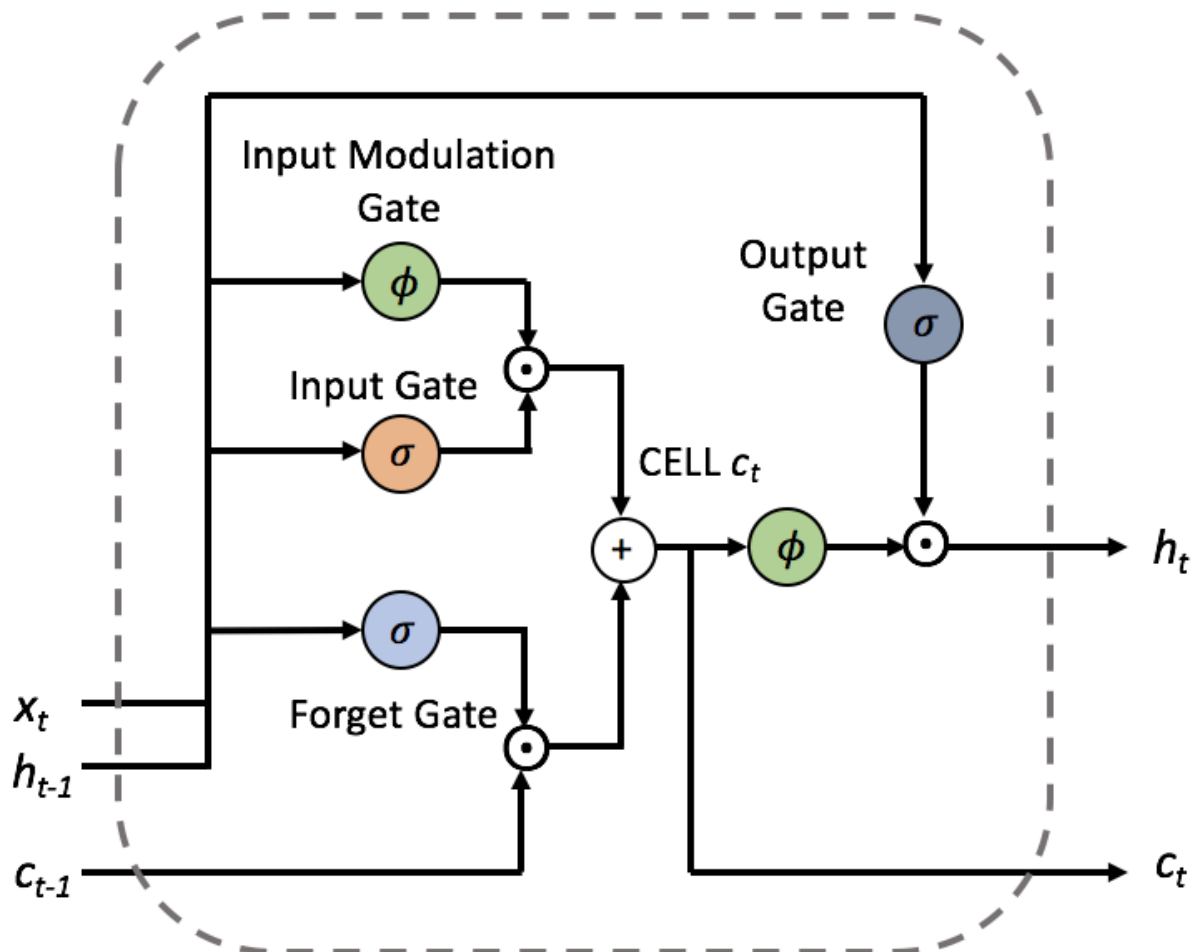
$$z_1^{(2)} = (1)(1) + (0)(2) + (1)(4) + (0)(5) = 5$$

รูปที่ 2.19 สมการการหาค่า z

2.6 Long Short-Term Memory (LSTM)

เพื่อแก้ปัญหาของ RNN ที่มีต่อ sequence ยาวๆ ของข้อมูล ก็เลยมีการเสนอการใช้ Long Short-Term Memory หรือ LSTM นี้ขึ้นมาโดย Sepp Hochreiter และ Juergen Schmidhuber

LSTM ก็เหมือน RNN แบบหนึ่ง ที่มีรายละเอียดเยอะกว่าหน่อย (ก็ไม่ค่อยน้อยเท่าไร) เพื่อการง่ายในการอธิบายต่อไป ขอแสดงรูปการทำงานทำงานของ LSTM ให้ดูก่อน



รูปที่ 2.20 กระบวนการทำงานของ LSTM

หากมอง RNN เป็นเหมือน Neural Network ที่มี memory ธรรมดาๆ อยู่ข้างในเพื่อบันทึก hidden state ก่อนหน้าแล้วละก็ LSTM ก็มี memory ภายในอยู่เช่นกัน แต่ที่เหนือกว่า RNN ก็คือ memory ที่ว่านี้สามารถมีตัวบอกได้ด้วยว่า เมื่อไหร่ที่ควรที่จะ write, forget (delete) หรือ อนุญาตให้ read ได้

ฟังดูคล้ายๆ memory ที่อยู่ในคอมพิวเตอร์ทั่วไป ที่มันสามารถ read, write หรือ forget (delete) ได้ โดยที่ในคอมพิวเตอร์นั้นเราก็จะมีตัวคล้ายๆ กับบิต 1/0 ที่คอยสั่งว่า เมื่อไหร่ที่มันควรจะให้ read หรือ write หรือ forget ในส่วนของ LSTM ก็เช่นกัน ต่างกันที่ตัวที่ควบคุมกับ read, write หรือ forget นั้นไม่ใช่บิต 1/0 แต่เป็นค่า analog ที่มีขึ้นเพื่อควบคุมความสามารถตรงนี้

การทำงานของ LSTM

ก่อนจะเริ่มเข้าสู่การทำงานของ LSTM มีตัวแปรที่ควรรู้กัน คือ

- Cell state เป็นตัวเก็บ state ของ memory cell ใน LSTM
- Gate เป็นตัวที่ควบคุมการไหลของข้อมูล ซึ่งก็คือ ค่า analog ที่คอยควบคุมว่าเมื่อไหร่ควรจะ read, write หรือ forget ซึ่งมันก็เหมือนกับประตูที่จะดูว่า เมื่อไหร่ควรเปิดให้ข้อมูลไหลเข้า ไหลออก หรือ ไหลหายไปเลย (forget)

วิธีการทำงาน

2.6.1 Forget

การ forget ก็เหมือนการล้าง cell state เดิมออกไป คือ จงลืมมันไปซะ เตรียมเคลียร์พื้นที่รับข้อมูลใหม่ แต่คนที่จะตัดสินใจว่าจะลบดี หรือไม่ลบดี ก็จะเป็นหน้าที่ของสิ่งที่เรียกว่า forget gate (อย่างที่บอกว่า gate จะเป็นตัวควบคุมการไหลไปมาของข้อมูลข้างใน) ซึ่งเป็นเหมือนตัวตัดสินใจ ถ้า forget gate ให้ค่าเป็น 0 ก็จงลบ cell state เดิมออกไปซะ แต่ถ้า forget gate ให้ค่าเป็น 1 ก็ยินดีด้วย เรายังยังเก็บ cell state นี้ต่อไป

การสร้าง forget gate นี้ เราจะดู input data ที่เข้ามา ประกอบกับ hidden state ก่อนหน้า (ตามสูตรของ RNN) ประกอบการตัดสินใจ โดยจะใช้ sigmoid function เป็นตัวตัดสินใจ ดังสมการ

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

รูปที่ 2.21 สมการการคำนวณหาค่า f_t ในส่วนของการจดจำข้อมูล

2.6.2 Write

เมื่อมี input data ใหม่เข้ามา เราก็ต้องมานั่งคิดว่า 1) เราอัปเดต cell state ของเราด้วย input data ใหม่ดีมั๊ย และ 2) ถ้าเราจะอัปเดตเราจะอัปเดตด้วยค่าอะไรดี มากล่าวกันทีละอย่างเลย

เรื่องแรก เราจะอัปเดต cell state ของเราดีมั๊ย อันนี้จะควบคุมโดยสิ่งที่เรียกว่า input gate ตรงนี้เราก็จะยังคงใช้ sigmoid function เป็นตัวตัดสินใจว่าจะอนุญาตให้อัปเดตหรือไม่ แน่นอนว่าการคำนวณนี้ใช้ค่า input data ที่เข้ามา กับ hidden state ก่อนหน้านั้น

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

รูปที่ 2.22 สมการการคำนวณหาค่า i_t ในส่วนของการอนุญาตกรเขียนข้อมูลเพื่ออัปเดตสเตต

มาถึงปัญหาที่สอง คือ แล้วถ้าเราจะอัปเดตจริงๆ เราจะอัปเดตด้วยค่าอะไรดี ซึ่งรอบนี้จะใช้สิ่งที่เรียกว่า Input modulation gate เป็นตัวจัดการ โดยสมการก็จะคล้ายๆ กับ input gate แต่จะใช้เป็น tanh function แทน ซึ่งค่าที่ได้นั้น จะมองว่าเป็น cell state candidate ก็ได้

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

รูปที่ 2.23 สมการการหาค่าที่จะถูกเขียน

2.6.3 อัปเดต cell state

ก่อนจะไปถึงตัวที่บอกว่าจะให้ read ขอค้นด้วยการอัปเดตค่า cell state ก่อน ตอนนี้เราได้ข้อมูลจาก forget gate, input gate และ input modulation gate แล้ว ซึ่งก็เพียงพอต่อการอัปเดต cell state ได้แล้ว ทีนี้เราก้รวมทุกสิ่งทุกอย่างเข้าด้วยกันเลย ดังนี้

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

รูปที่ 2.24 สมการการอัปเดตข้อมูล

มาดูทางฝั่งขวามือของสมการกัน เริ่มจากส่วนแรกของสมการ จะเห็นว่า ถ้า forget gate บอกว่า ให้ลบ cell state เดิมทิ้ง (f_t มีค่าเป็น 0) เราก้จะไม่เอาค่า c_{t-1} มาประกอบการอัปเดต cell state เลย คือ ลืมมันไปซะ แต่ถ้า f_t มีค่าเป็น 1 เราก้จะยังคงค่า c_{t-1} เอาไว้ประกอบการพิจารณาการอัปเดตอยู่

มาในส่วนหลังของสมการบ้าง ส่วนนี้จะเป็นส่วนของการอัปเดต cell state จากข้อมูลใหม่ ตอนนี้เรามีค่าที่จะอัปเดตเตรียมไว้แล้วจาก input modulation gate หรือ g_t ทีนี้เราก้จะมาดูกันว่า ค่า g_t ที่เตรียมไว้เนี่ย จะได้ใช้ หรือไม่ได้ใช้ อันนี้ก็จะใช้ output จาก input gate หรือ i_t มาตัดสิน ถ้า i_t เป็น 1 ก็ลุยเลย ใช้ค่า g_t อัปเดตไปเลย แต่ถ้า i_t เป็น 0 ก็เสียใจด้วย เราจะไม่ใช้ g_t ที่ dimension นั้นอัปเดตนะ

จากค่าทั้งหมดที่ได้มา เราก้จะได้ค่า c_t ตัวใหม่เป็นที่เรียบร้อย

2.6.4 Read

จากเรื่อง RNN ตัวเดิม สิ่งที่เราต้องผลิตออกไปก็คือ hidden state ณ เวลาที่ t หรือ h_t ออกไป ซึ่งเมื่อตอนที่เวลา $t+1$ ตัว LSTM นี้จะได้เอาค่า h_t นี้ไปคำนวณด้วย (ตามสูตร) ดังนั้นคำว่า read จริงๆ ในที่นี้แล้ว ถ้าบอกให้ถูก มันก็คือ การที่จะอนุญาตให้คนข้างนอกมา read ตัว h_t ได้หรือเปล่า (จริงๆ ก็เหมือนตั้ง permission อย่างนึงแหละ ว่าจะมาเห็นได้มัย) หรือว่าจะเก็บเอาไว้ในใจเจ๊ยยๆ ไม่บอกใคร ไม่ส่งค่า h_t ต่อ ตรงนี้เราก้จะมี output gate มาช่วยตัดสิน โดยจะยังคงใช้สูตรเดิมกับ forget gate และ input gate อยู่ก็คือใช้ sigmoid function กับค่า hidden state ตัวก่อนหน้า กับ input data ที่เข้ามาในตอนนั้น

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

รูปที่ 2.25 สมการการอนุญาตการอ่านข้อมูล

ค่าที่เราจะ output เป็นค่า h_t สำหรับ sequence ถัดไป ก็จะเป็นว่า

$$h_t = o_t \odot \tanh(c_t)$$

รูปที่ 2.26 สมการการหาค่า hidden state

จะเห็นว่า ถ้า output gate ให้ตัว o_t มีค่าเป็น 0 แล้วละก็ ค่าของ h_t ก็จะมีค่าเป็น 0 คือไม่ส่งค่าใดๆ ออกไป ในขณะเดียวกัน ถ้า o_t มีค่าเป็น 1 เราก็คำนวณค่า h_t และส่งออกไปข้างนอก หรือก็คือ อนุญาตให้คนอื่นๆ สามารถดูค่า h_t

บทที่ 3

วิธีการดำเนินงาน

3.1 เครื่องมือและข้อมูลที่ใช้ดำเนินการ

3.1.1 ฮาร์ดแวร์ (Hardware)

คอมพิวเตอร์ (notebook) Acer Nitro 5 ,ชุดประมวลผล Intel(R) Core(TM) i5-7300HQ
CPU @ 2.50GHz 2.50 GHz, หน่วยความจำ (RAM) 16.0 GB, ชุดประมวลผลหน้าจอ GeForce
GTX 1050

3.1.2 ซอฟต์แวร์ (Software)

3.1.2.1 Jupyter notebook

Jupyter Notebook คือ เว็บไซต์ที่ประกอบด้วย ช่อง ๆ cell เรียงต่อกันลงไป โดยแต่ละ cell สามารถเป็นเนื้อหา static content ต่าง ๆ เช่น ข้อความ รูปภาพ กราฟ วิดีโอ เสียง หรือ เป็นโค้ดโปรแกรมคอมพิวเตอร์ ภาษา Python ที่สามารถรันคำสั่งประมวลผล แสดงผลลัพธ์ออกมาได้จริง ๆ

3.1.2.2 Python 3.6.1

ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เราเขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreter คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่เรต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น

3.1.2.3 Microsoft excel

Microsoft Excel (ไมโครซอฟท์ เอ็กเซล) เป็นโปรแกรมประเภท สเปรดชีต (spreadsheet) หรือตารางคำนวณอิเล็กทรอนิกส์ที่ใช้เก็บบันทึกข้อมูลในลักษณะต่างๆ ซึ่ง

ส่วนใหญ่มักเก็บข้อมูลประเภทการคำนวณ โดยจะเก็บข้อมูลลงในตารางสี่เหลี่ยมที่เรียกว่า เซล (Cell) ที่สามารถนำเอาเซลมาอ้างอิงใส่ในสูตร เพื่อให้โปรแกรมคำนวณหาผลลัพธ์จากข้อมูลที่บันทึกไว้ได้

3.1.3 ข้อมูล (Data)

3.1.3.1 หุ่น

ประกอบไปด้วยหุ่นดังนี้

- 1.BTS.BK มีปัจจัยและหุ่นที่เกี่ยวข้องที่นำมารวมทำนาย จำนวน 10 ปัจจัย มีดังนี้
BEM.BK ,EGCO.BK ,AOT.BK ,III.BK ,JWD.BK ,CK.BK ,UNIQ.BK ,SI=F
,PL=F ,HG=F
- 2.CPF.BK มีปัจจัยและหุ่นที่เกี่ยวข้องที่นำมารวมทำนาย จำนวน 9 ปัจจัย มีดังนี้
CC=F ,KC=F ,SB=F ,ZL=F ,GF=F ,LE=F ,HE=F ,ZM=F ,ZR=F
- 3.PTTEP.BK มีปัจจัยและหุ่นที่เกี่ยวข้องที่นำมารวมทำนาย จำนวน 6 ปัจจัย มีดังนี้
CL=F ,RB=F ,HO=F ,NG=F ,ESSO.BK ,BCP.BK
- 4.SCC.BK มีปัจจัยและหุ่นที่เกี่ยวข้องที่นำมารวมทำนาย จำนวน 8 ปัจจัย มีดังนี้
TNPC.BK ,TOA.BK ,DRT.BK ,STEC.BK ,TPOLY.BK ,SI=F ,PL=F ,HG=F
- 5.TVO.BK มีปัจจัยและหุ่นที่เกี่ยวข้องที่นำมารวมทำนาย จำนวน 7 ปัจจัย มีดังนี้
LST.BK ,VPO.BK ,UVAN.BK ,UPOIC.BK ,ZM=F ,ZL=F ,ZS=F

ประกอบไปด้วย 9 column

1. Date
วัน เดือน ปี
2. Name
ชื่อย่อหรือสัญลักษณ์หุ่น
3. Open Price
ราคาซื้อขายตอนเปิด
4. Close Price
ราคาซื้อขายตอนปิด

5. Adjusted Close Price

ราคาที่หักค่าใช้จ่ายต่างๆของผู้ถือหุ้น ที่สามารถนำไปคำนวณ return ได้

6. Volumn

จำนวนหรือปริมาณหุ้น

3.1.4 ไลบรารี (Library)

3.1.4.1 Tensorflow

TensorFlow เป็นไลบรารีสำหรับใช้พัฒนา machine learning เป็น Open source (เขียนด้วย Python) ที่พัฒนาโดยกูเกิล โดยกูเกิลได้ปล่อย TensorFlow ซึ่งเป็นไลบรารีสำหรับใช้พัฒนา machine learning ที่ใช้งานกันในกูเกิลเอง ให้กลายเป็นโปรแกรม Open source

3.1.4.2 numpy

เป็นส่วนขยายของภาษาไพทอน เพื่อจัดการเมทริกซ์หรืออาร์เรย์หลายมิติรวมถึงฟังก์ชันทางคณิตศาสตร์ที่ทำงานบนอาร์เรย์เหล่านี้ นอกจากนี้คลังซอฟต์แวร์โอเพนซอร์สและเสรีนี้มีฟังก์ชันหลายอย่างที่ทำให้ทำได้ โดยเฉพาะอย่างยิ่งในการสร้างอาร์เรย์โดยตรงจากไฟล์ หรือเพื่อบันทึกอาร์เรย์ในไฟล์ เพื่อจัดการเวกเตอร์ เมทริกซ์ และพหุนาม

3.1.4.3 Pandas

Pandas คือ Library หนึ่งของ Python ที่มีความสามารถสำหรับจัดเตรียมทำความสะอาด เก็บกวาดข้อมูลไว้สำหรับการทำ Data Visualization และ สร้าง Model ต่อไป โดยเจ้า Pandas นี้ถูกสร้างขึ้นบนพื้นฐานของ Library Numpy ครบ สิ่งที่ทำให้ Pandas ถูกนำไปใช้อย่างแพร่หลายนั้น เพราะมี Object ที่สารพัดประโยชน์มากๆ เรียกว่า DataFrame

3.1.4.4 Keras

เคราส เป็นไลบรารีโอเพนซอร์สของภาษาไพทอนสำหรับการพัฒนาโครงข่ายประสาทเทียม สามารถทำงานบน TensorFlow, Microsoft Cognitive Toolkit, R,

Theano, หรือ PlaidML ได้ เคราสถูกออกแบบมาให้ผู้ใช้สามารถพัฒนาโปรแกรมด้วยการเรียนรู้เชิงลึกได้อย่างรวดเร็ว จึงใช้งานง่าย มีฟังก์ชันให้เลือกหลากหลาย

3.1.4.5 plotly

Plotly เป็นบริการรายงานข้อมูลกราฟแบบออนไลน์ (<https://plot.ly/>) และยังสามารถออกโมดูล Plotly เป็นโมดูลหนึ่งที่เป็นที่นิยมกันในการสร้างกราฟรายงานข้อมูลทั้งแบบ 2 มิติ (รูปแท่ง , รูปวงกลม และอื่น ๆ) และ แบบ 3 มิติ ในภาษาไพทอน นิยมใช้งานกับ IPython Notebook นอกจากนั้นยังสามารถบันทึกกราฟข้อมูลขึ้นไว้กับ Plotly account ทำให้สามารถดูกราฟแบบออนไลน์ได้และยังสามารถใช้งานแบบออฟไลน์ได้

3.1.4.6 Jupyter Dash

Dash เป็น Plotly ของโอเพนซอร์ส python กรอบสำหรับการสร้างสแต็คเต็มรูปแบบการใช้งานเว็บโดยใช้การวิเคราะห์ด้วย python (ไม่ต้องใช้ JavaScript) Dash DevTools นักพัฒนาสามารถทำการออกแบบแอปพลิเคชันได้อย่างรวดเร็วโดยใช้ Integrated Development Environment (IDE) และ JupyterDash ทำให้คุณสมบัติเหล่านี้และอีกมากมายพร้อมใช้งานจาก Jupyter

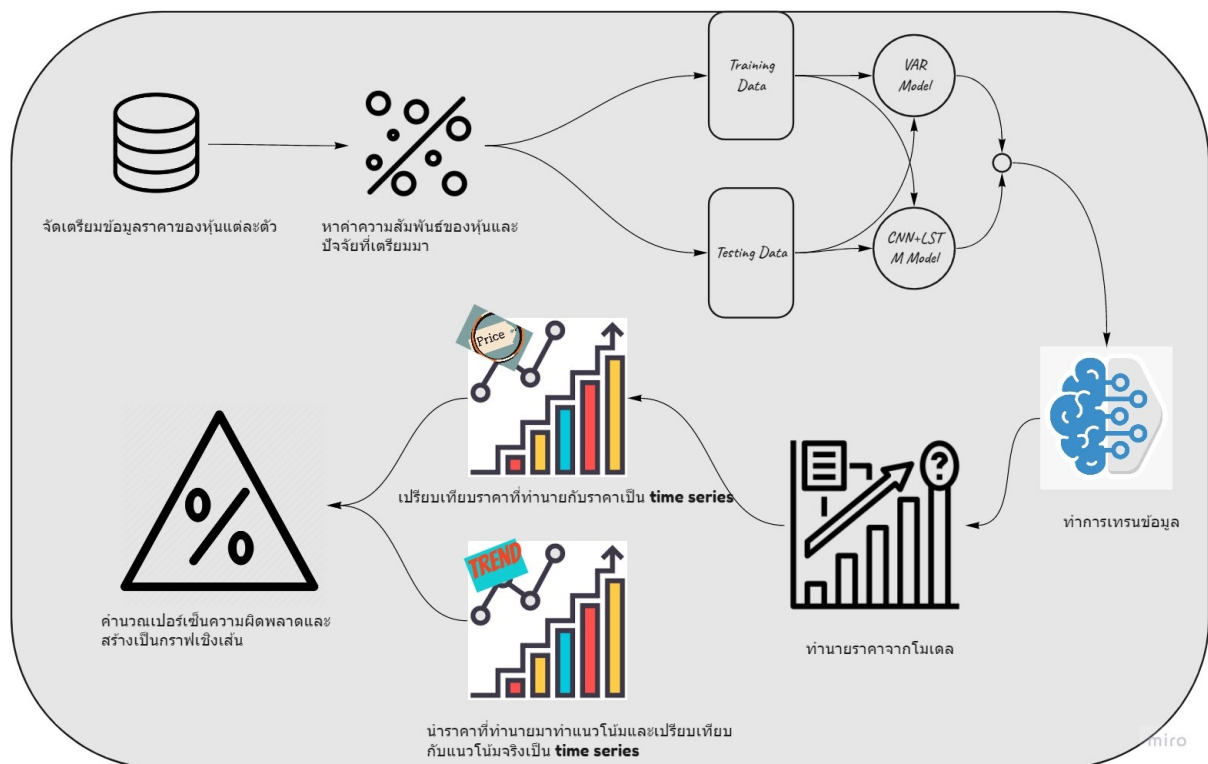
3.2 การออกแบบระบบ

การสร้างระบบวิเคราะห์ราคาหุ้นจะถูกออกแบบด้วยการใช้ข้อมูลที่มีมาก่อนอยู่แล้วมาใช้ฝึกฝน และเข้าระบบสมการตามโมเดลที่วางแผนไว้ การสร้างระบบจำเป็นต้องวิเคราะห์ปัจจัยต่างๆที่มีต่อราคาของหุ้น และมีตัวชี้วัดที่บอกได้ว่าระบบมีประสิทธิภาพเพียงใด และมีการวิเคราะห์รูปแบบโมเดลที่เลือกใช้ ค่าต่างๆที่ถูกตั้งไว้ในโมเดลที่เราต้องการนำมาเทรน

ออกแบบการทำงานของระบบ

- เตรียมข้อมูลราคาและจัดทำแนวโน้มของราคา
- ความสัมพันธ์ของปัจจัยที่ส่งผลต่อหุ้นและหุ้นที่เลือกมาต้องมีความเข้ากัน
- แบ่งข้อมูลระหว่างข้อมูลที่เอาไว้ train and test
- กำหนดค่าต่างๆในกระบวนการระหว่างการเทรนของ model

- ทำการเทรนข้อมูลโดยใช้แบบจำลองที่เตรียมไว้
- ทำนายราคา
- คำนวณและเปรียบเทียบเปอร์เซ็นต์และค่าเงินที่คลาดเคลื่อนจากการทำนายราคา
- สร้างแนวโน้มจากราคาที่ทำนายและเปรียบเทียบกับของจริง



รูปที่ 3.1 การออกแบบขั้นตอนการทำงาน

3.3 พัฒนาระบบ

การออกแบบถูกแบ่งออกเป็น 2 ส่วน คือ ระบบที่พัฒนาด้วยการวิเคราะห์การถดถอยพหุคูณที่ประกอบด้วย VAR model และ ระบบที่พัฒนาด้วยการเรียนรู้ของเครื่องจักรและโครงข่ายประสาทเทียมที่ประกอบด้วย LSTM+CNN

3.3.1 การจัดเตรียมข้อมูล

จัดเตรียมในรูปแบบตารางที่แสดงถึง Adj close price ของหุ้นที่ต้องการทำนาย(สีแดง) และหุ้นหรือปัจจัยอื่นๆ โดยช่วงเวลาที่นำมาใช้คือ 04/01/2010 - 28/08/2020 จะถูกจัดเก็บเป็นไฟล์ CSV

	AOT.BK	BEM.BK	CK.BK	EGCO.BK	HG=F	III.BK	JWD.BK	PL=F	SI=F	UNIQ.BK	BTS.BK
Date											
2010-01-04	3.245882	1.677729	4.719489	51.251045	3.3880	9.071856	8.132977	1517.300049	17.440001	5.188159	1.981645
2010-01-05	3.266824	1.677729	4.643977	51.568375	3.3960	9.071856	8.132977	1530.800049	17.781000	5.151363	1.981645
2010-01-06	3.308706	1.677729	4.681734	51.092346	3.4775	9.071856	8.132977	1552.199951	18.163000	5.151363	1.948617
2010-01-07	3.350588	1.698967	4.643977	51.092346	3.4115	9.071856	8.132977	1553.000000	18.333000	5.151363	1.948617
2010-01-08	3.350588	1.677729	4.681734	51.092346	3.3880	9.071856	8.132977	1564.599976	18.458000	5.188159	1.948617
...
2020-08-24	56.421200	9.250000	17.799999	220.557617	2.9225	4.840000	7.250000	920.799988	26.587999	4.820000	10.239175
2020-08-25	56.669750	9.250000	17.700001	220.557617	2.9310	4.760000	7.350000	930.599976	26.254000	4.840000	10.239175
2020-08-26	56.669750	9.250000	17.900000	220.557617	2.9595	4.780000	7.600000	936.400024	27.430000	4.880000	10.534535
2020-08-27	56.421200	9.300000	18.700001	214.753464	2.9650	4.720000	7.500000	925.599976	27.010000	5.050000	10.337628
2020-08-28	56.421200	9.300000	19.100000	216.688187	2.9950	4.520000	7.600000	937.500000	27.610001	5.000000	10.436083

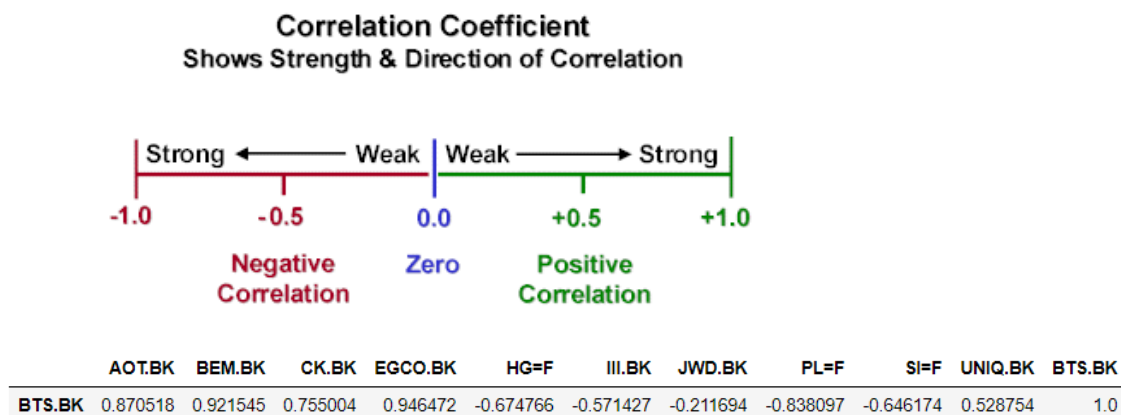
2604 rows × 11 columns

รูปที่ 3.2 ตารางแสดงข้อมูลของหุ้น

3.3.2 วิเคราะห์ความสัมพันธ์ระหว่างหุ้นและปัจจัย

3.3.2.1 หาค่า correlation ระหว่างตัวหุ้น

ปัจจัยที่มีค่าความสัมพันธ์สหสัมพันธ์ที่มากกว่า 0.5 หรือ น้อยกว่า -0.5 แสดงถึงความเกี่ยวข้องของราคาระหว่างกัน



รูปที่ 3.3 ค่า correlation ของหุ้น BTS

3.3.2.2 ตรวจสอบความเป็น stationary ของข้อมูล

ทำการตัด trend และ seasonal ของข้อมูลออกเพื่อที่จะได้นำไปเข้าสู่กระบวนการเทรนในโมเดลโดยเพื่อให้ความคลาดเคลื่อนน้อยลง ด้วยวิธีการทำการ difference ไปเรื่อยๆจนกว่าข้อมูลจะกลายเป็น Stationary

```
Augmented Dickey-Fuller Test on "GF=F"
-----
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level      = 0.05
Test Statistic         = -2.0638
No. Lags Chosen        = 1
Critical value 1%      = -3.433
Critical value 5%      = -2.863
Critical value 10%     = -2.567
=> P-Value = 0.2593. Weak evidence to reject the Null Hypothesis.
=> Series is Non-Stationary.
```

รูปที่ 3.4 ตัวอย่างแสดงว่าข้อมูลไม่เป็น Stationary

Augmented Dickey-Fuller Test on "GF=F"

```
-----
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level      = 0.05
Test Statistic          = -46.0372
No. Lags Chosen         = 0
Critical value 1%       = -3.433
Critical value 5%       = -2.863
Critical value 10%      = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.
```

รูปที่ 3.5 ตัวอย่างแสดงว่าข้อมูลเป็น Stationary

3.3.3 แบ่งข้อมูลสำหรับ การ Train และ Test

สำหรับข้อมูลที่ใช้ในการเทรนจะอยู่ในช่วงเวลา 04/01/2010 - 31/12/2018

Date	AOT.BK	BEM.BK	CK.BK	EGCO.BK	HG=F	III.BK	JWD.BK	PL=F	SI=F	UNIQ.BK	BTS.BK
2010-01-04	3.245882	1.677729	4.719489	51.251045	3.3880	9.071856	8.132977	1517.300049	17.440001	5.188159	1.981645
2010-01-05	3.266824	1.677729	4.643977	51.568375	3.3960	9.071856	8.132977	1530.800049	17.781000	5.151363	1.981645
2010-01-06	3.308706	1.677729	4.681734	51.092346	3.4775	9.071856	8.132977	1552.199951	18.163000	5.151363	1.948617
2010-01-07	3.350588	1.698967	4.643977	51.092346	3.4115	9.071856	8.132977	1553.000000	18.333000	5.151363	1.948617
2010-01-08	3.350588	1.677729	4.681734	51.092346	3.3880	9.071856	8.132977	1564.599976	18.458000	5.188159	1.948617
...
2018-12-24	63.712601	9.867889	24.392187	233.731750	2.7085	4.689181	6.074696	799.500000	15.004000	8.296088	8.781550
2018-12-25	61.507168	9.626030	23.818254	233.731750	2.7085	4.614154	5.894705	799.500000	15.004000	7.955736	8.599550
2018-12-26	62.732407	9.190682	23.818254	231.861877	2.7085	4.632911	5.894705	799.500000	15.004000	7.743016	8.599550
2018-12-27	61.997261	9.335797	23.722597	228.122177	2.6675	4.670424	6.074696	795.500000	15.191000	7.743016	8.554049
2018-12-28	62.977455	9.384170	23.913908	231.861877	2.6790	4.670424	6.299685	789.599976	15.326000	8.168457	8.690550

รูปที่ 3.6 ข้อมูลในการใช้ Train

สำหรับข้อมูลที่ใช้ในการทดสอบจะอยู่ในช่วงเวลา 01/01/2019 - 28/08/2020

Date	AOT.BK	BEM.BK	CK.BK	EGCO.BK	HG=F	III.BK	JWD.BK	PL=F	SI=F	UNIQ.BK	BTS.BK
2019-01-02	62.977455	9.626030	23.722597	229.992020	2.6250	4.632911	6.299685	799.099976	15.542000	7.870647	8.645051
2019-01-03	62.242313	9.577659	23.913908	231.861877	2.5705	4.689181	6.299685	794.500000	15.706000	7.828104	8.645051
2019-01-04	63.222504	9.432543	23.913908	236.536530	2.6515	4.689181	6.299685	822.000000	15.695000	7.955736	8.645051
2019-01-07	62.977455	9.529286	24.631327	230.926987	2.6410	4.736073	6.299685	818.400024	15.669000	8.211000	8.645051
2019-01-08	63.467552	9.529286	24.153048	230.926987	2.6605	4.970532	6.299685	816.500000	15.626000	8.296088	8.645051
...
2020-08-24	56.421200	9.250000	17.799999	220.557617	2.9225	4.840000	7.250000	920.799988	26.587999	4.820000	10.239175
2020-08-25	56.669750	9.250000	17.700001	220.557617	2.9310	4.760000	7.350000	930.599976	26.254000	4.840000	10.239175
2020-08-26	56.669750	9.250000	17.900000	220.557617	2.9595	4.780000	7.600000	936.400024	27.430000	4.880000	10.534535
2020-08-27	56.421200	9.300000	18.700001	214.753464	2.9650	4.720000	7.500000	925.599976	27.010000	5.050000	10.337628
2020-08-28	56.421200	9.300000	19.100000	216.688187	2.9950	4.520000	7.600000	937.500000	27.610001	5.000000	10.436083

รูปที่ 3.7 ข้อมูลในการใช้ Test

3.3.4 ตั้งค่าและปรับแต่ง Model และทำการเทรน

3.3.4.1 นำเข้ากระบวนการ fitting โดยใช้ VAR model

เป็นการเทรนข้อมูลโดยมีค่าของปัจจัยหลายตัวที่ถูกนำมาคำนวณเพื่อให้ค่าต่อไปของหุ้น กล่าวคือ จะใช้ ข้อมูลปัจจัยทุกตัว ณ เวลา t มาคำนวณ เพื่อให้ได้ ข้อมูลของของหุ้น ณ เวลา t+1 และยังมีการนำข้อมูลปัจจัยด้วยกันเองมาคำนวณเพื่อหาค่าของข้อมูลของแต่ละปัจจัย ณ เวลา t+1 จะมีการทำไปตลอดจนข้อมูลจะหมดหรือตามที่กำหนดไว้

$$\begin{aligned} Y_{1,t} &= \alpha_1 + \beta_{11,1}Y_{1,t-1} + \beta_{12,1}Y_{2,t-1} + \beta_{13,1}Y_{3,t-1} + \beta_{11,2}Y_{1,t-2} + \beta_{12,2}Y_{2,t-2} + \beta_{13,2}Y_{3,t-2} + \epsilon_{1,t} \\ Y_{2,t} &= \alpha_2 + \beta_{21,1}Y_{1,t-1} + \beta_{22,1}Y_{2,t-1} + \beta_{23,1}Y_{3,t-1} + \beta_{21,2}Y_{1,t-2} + \beta_{22,2}Y_{2,t-2} + \beta_{23,2}Y_{3,t-2} + \epsilon_{2,t} \\ Y_{3,t} &= \alpha_3 + \beta_{31,1}Y_{1,t-1} + \beta_{32,1}Y_{2,t-1} + \beta_{33,1}Y_{3,t-1} + \beta_{31,2}Y_{1,t-2} + \beta_{32,2}Y_{2,t-2} + \beta_{33,2}Y_{3,t-2} + \epsilon_{3,t} \end{aligned}$$

รูปที่ 3.8 สมการที่ใช้การในแบบจำลอง VAR

3.3.4.2 นำเข้ากระบวนการ fitting โดยใช้ CNN+LSTM model

เราจะนำข้อมูลของตัวหุ้นหลักเพียงตัวเดียวนำเข้าไปเทรนโดย layer ที่ถูกใช้ในการเทรนครั้งนี้ เป็นดังนี้

CNN 1D → MAX_pooling → LSTM → LSTM → Dense → Dense

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 2197, 64)	192
max_pooling1d_2 (MaxPooling1D)	(None, 1098, 64)	0
lstm_2 (LSTM)	(None, 1098, 64)	33024
lstm_3 (LSTM)	(None, 32)	12416
dense (Dense)	(None, 16)	528
dense_1 (Dense)	(None, 7)	119

รูปที่ 3.9 ลำดับ layer ในการ CNN+LSTM Model

3.3.5 ทำนายราคา

เราจะนำ ราคาในช่วงเวลา 04-01-2010 - 31-12-2018 มาทำนายราคา ราคาอีก 7 วันข้างหน้า โดย จะทำการเลื่อนกรอบระยะเวลาไปที่ละ 7 วัน ดังนี้

ข้อมูลที่นำเข้า คือ 04-01-2010 - 31-12-2018 จะทำนายราคาวันที่ 01-01-2019 - 07-01-2019

ในการทำนายรอบต่อไปคือ 11-01-2010 - 07-01-2019 จะทำนายราคาวันที่ 08-01-2019 - 14-01-2019

ทำแบบนี้ไปจนครบทุกข้อมูล คือ 01-01-2019 - 28-08-2020

หลังจากนั้น จะทำการทำนายล่วงหน้า 7 วัน คือทำนายวันที่ 29-08-2020 - 04-09-2020 โดยใช้กรอบที่ขนาดอินพุตเท่ากับข้อมูลของการทดสอบ

3.3.6 คำนวนและเปรียบเทียบเปอร์เซ็นต์และค่าเงินที่คลาดเคลื่อนจากการทำนายราคาระหว่างโมเดลเทียบกับของจริง

ในการวัดผลจะสูตรคำนวณ 2 อย่าง คือ

MAPE บ่งบอกความคลาดเคลื่อนเฉลี่ยเป็นเปอร์เซ็นต์จากทำนายราคา y คือ ราคาจริง y_{hat} คือราคาที่ทำนายได้

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \cdot 100$$

RMSE บ่งบอกความคลาดเคลื่อนเฉลี่ยเป็นค่าตัวเลขที่ละเอียดก็คือค่าเงินที่ทำนายคลาดเคลื่อนจากความเป็นจริง จากทำนาย ราคา y คือ ราคาจริง y_{hat} คือราคาที่ทำนายได้

$$RMSE = \sqrt{\frac{1}{n} * \sum (prediction - actual)^2}$$

จากนั้น จะคำนวณค่าผิดพลาดในแต่ละวัน โดยใช้สูตรเดียวกันโดยไม่นำการหาค่าเฉลี่ยมาคำนวณ และนำไปสร้างกราฟแสดงค่าความคลาดเคลื่อน

3.3.7 สร้างแนวโน้มจากราคาที่ทำนายได้เปรียบเทียบกับแนวโน้มของจริง

สร้างแนวโน้มโดยการนำราคา 7 วันก่อน รวมถึงวันปัจจุบัน มาทำการหาค่าเฉลี่ย ดังนี้

ต้องการหาเทรน 07-01-2019 จะนำวันที่ 1 ถึง 7 มาหาค่าเฉลี่ย จากนั้นจะได้ค่าแนวโน้มออกมา ทำตั้งแต่ 07-01-2019 จนถึง 28-08-2020 รวมถึง วันที่ได้จากการทำนาย คือ 29-08-2020 - 04-09-2020

จากนั้นนำแนวโน้มที่ได้จากการทำนายแต่ละแบบจำลองมาเปรียบเทียบกับแนวโน้มของจริง

3.3.8 เปรียบเทียบแนวโน้มที่ได้จากการทำนายราคาระหว่างโมเดลเทียบกับของจริง

จากการนำราคาที่ทำนายมาสร้างแนวโน้มเพื่อให้สามารถดูการเพิ่มขึ้นและลดลงของราคาได้ง่ายขึ้น โดยการนำแนวโน้มจากการทำนายราคาของ VAR Model และ CNN+LSTM Model มาเปรียบเทียบกับแนวโน้มของราคาจริง

บทที่ 4

ผลการดำเนินโครงการ

4.1 ผลการออกแบบ

4.1.1 VAR Model หลังจากการสร้างแบบจำลองได้มีการปรับปัจจัยหรือองค์ประกอบที่จะนำมาคำนวณเพื่อทำนายราคาของหุ้นหลักทั้ง 5 ตัว สามารถจำแนกได้ว่าหุ้นแต่ละตัวมีปัจจัยที่ส่งผลต่อราคาหุ้นต่างกัน โดยหุ้นแต่ละตัวมีปัจจัยที่ส่งผลต่อราคาของหุ้นหลักดังนี้

BTS = [BEM.BK EGCO.BK AOT.BK III.BK JWD.BK CK.BK UNIQ.BK SI=F PL=F HG=F]

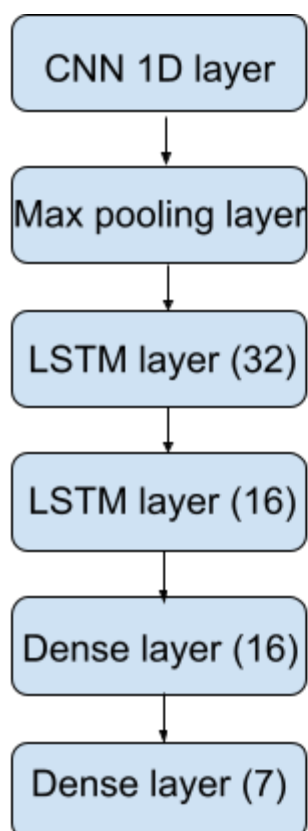
CPF = [CC=F KC=F SB=F ZL=F GF=F LE=F HE=F ZM=F ZR=F]

PTTEP = [CL=F RB=F HO=F NG=F ESSO.BK BCP.BK]

SCC = [TNPC.BK TOA.BK DRT.BK STEC.BK TPOLY.BK SI=F PL=F HG=F]

TVO = [LST.BK VPO.BK UVAN.BK UPOIC.BK ZM=F ZL=F ZS=F]

4.1.2 CNN+LSTM Model มีการปรับปรุงจำนวนชั้นของ layer และ จำนวนรอบที่ใช้ในการเทรน ดังนี้



รูปที่ 4.1 ลำดับและรูปแบบของ layer ในการ Train CNN+LSTM Model

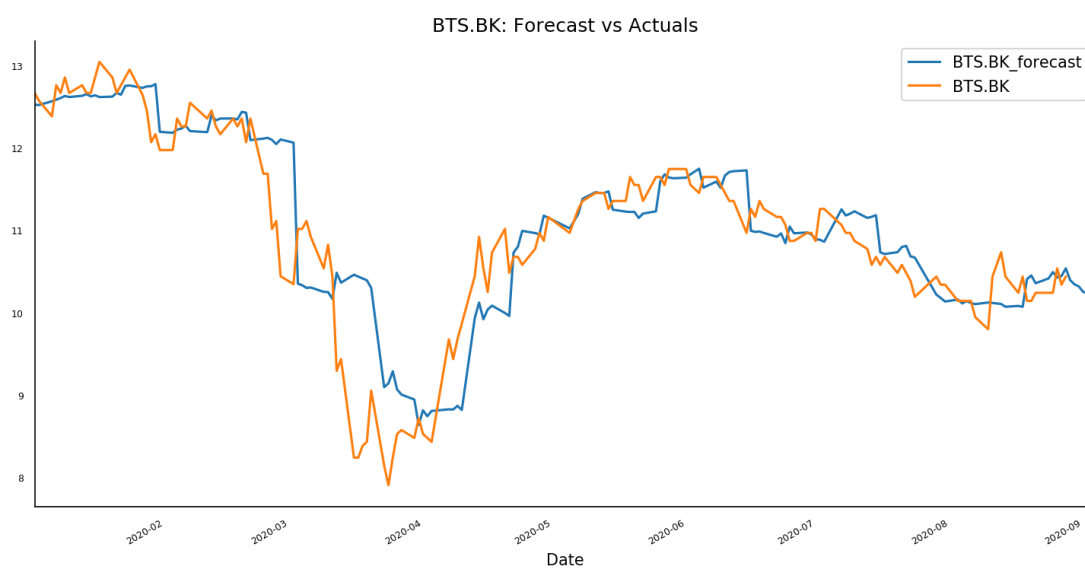
4.2 ผลการทำนายด้วยแบบจำลอง

4.2.1 Forecast Price

กราฟดังกล่าวคือกราฟที่ได้จากการนำข้อมูลหุ้นไปทำนาย และนำมาเปรียบเทียบกับข้อมูลจริงและมีการทำนาย 7 วันล่วงหน้า ก่อนที่ของจริงจะเกิดขึ้น

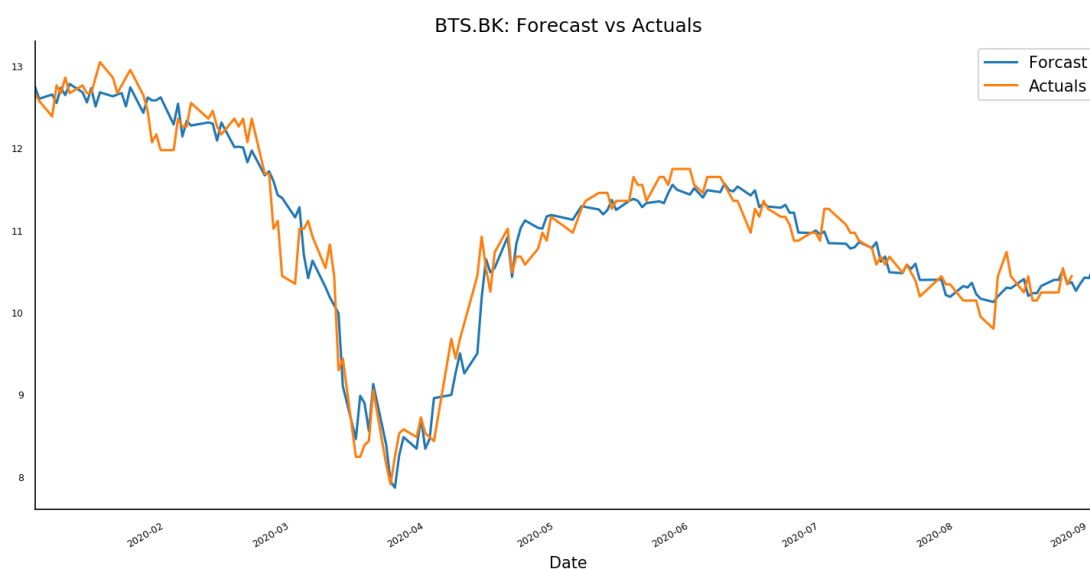
4.2.1.1 BTS Stock

VAR Model



รูปที่ 4.2 การทำนายราคาของหุ้น BTS โดยใช้ VAR Model

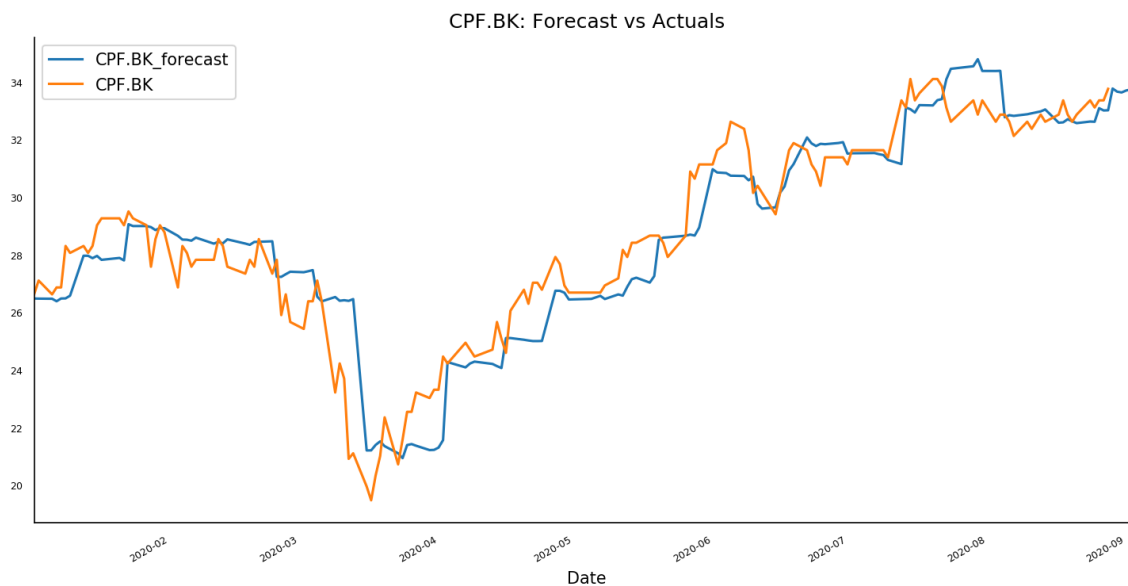
CNN+LSTM Model



รูปที่ 4.3 การทำนายราคาของหุ้น BTS โดยใช้ CNN+LSTM Model

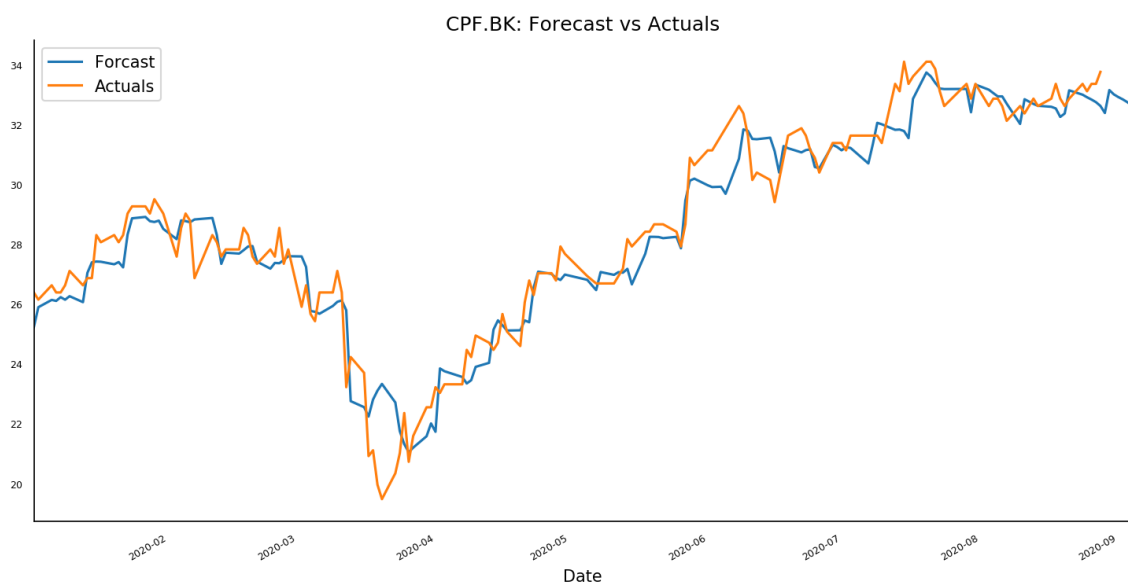
4.2.1.2 CPF Stock

VAR model



รูปที่ 4.4 การทำนายราคาของหุ้น CPF โดยใช้ VAR Model

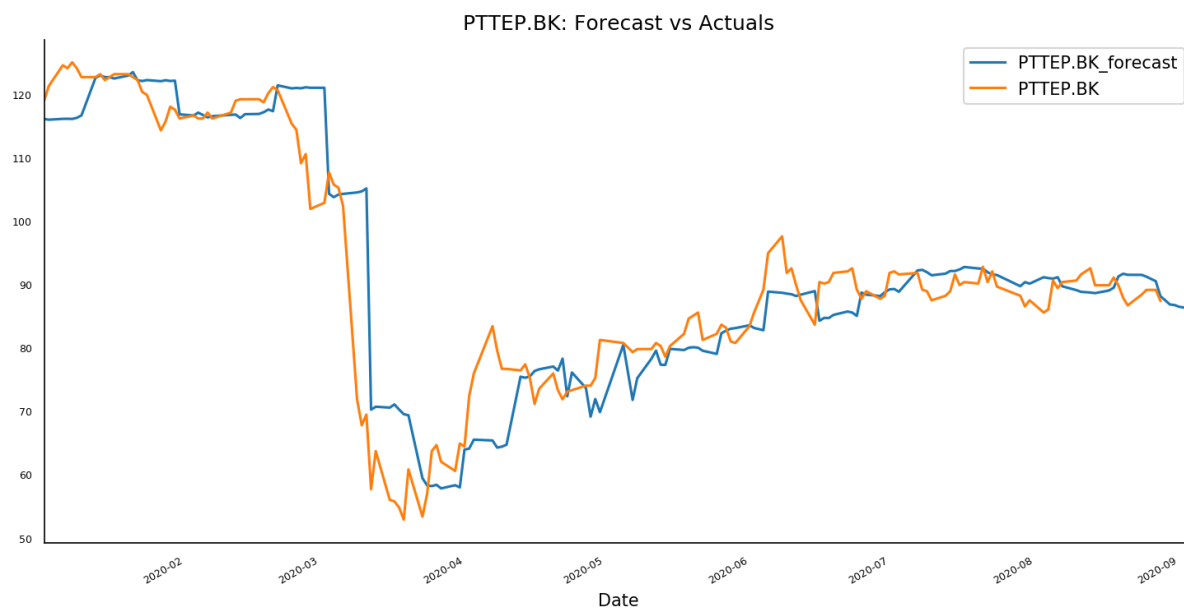
CNN+LSTM Model



รูปที่ 4.5 การทำนายราคาของหุ้น CPF โดยใช้ CNN+LSTM Model

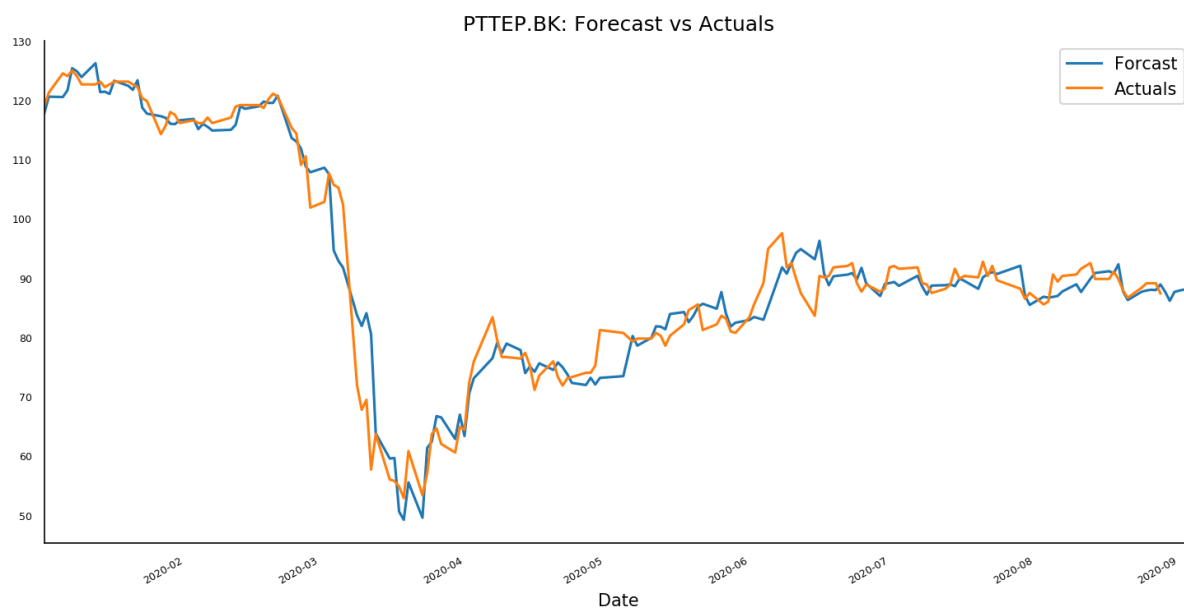
4.2.1.3 PTTEP Stock

VAR Model



รูปที่ 4.6 การทำนายราคาของหุ้น PTTEP โดยใช้ VAR Model

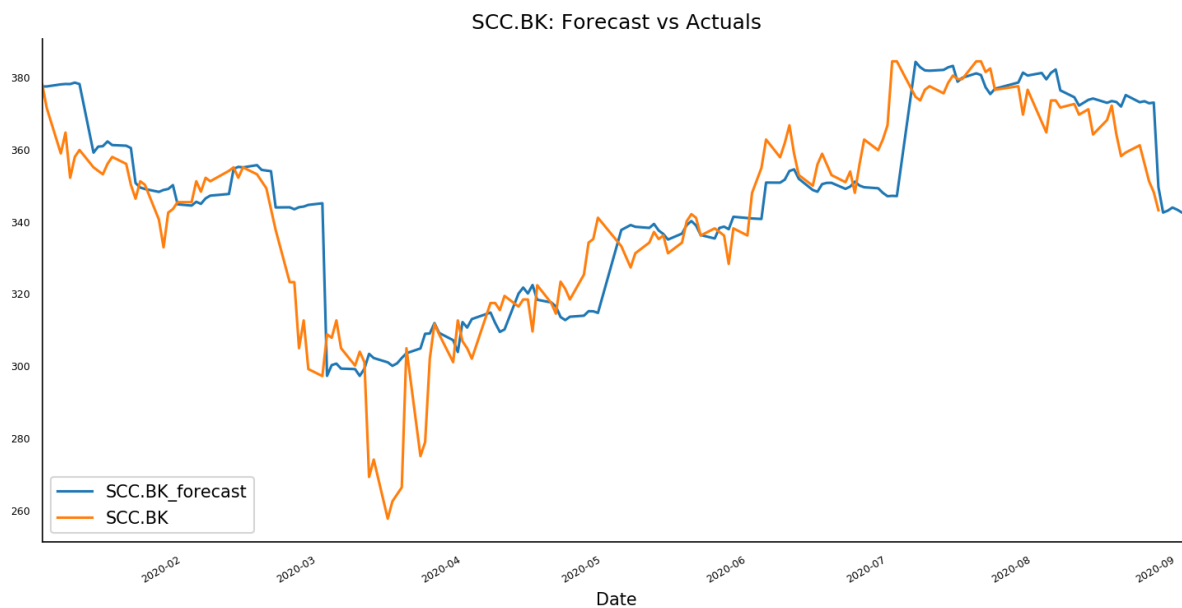
CNN+LSTM Model



รูปที่ 4.7 การทำนายราคาของหุ้น PTTEP โดยใช้ CNN+LSTM Model

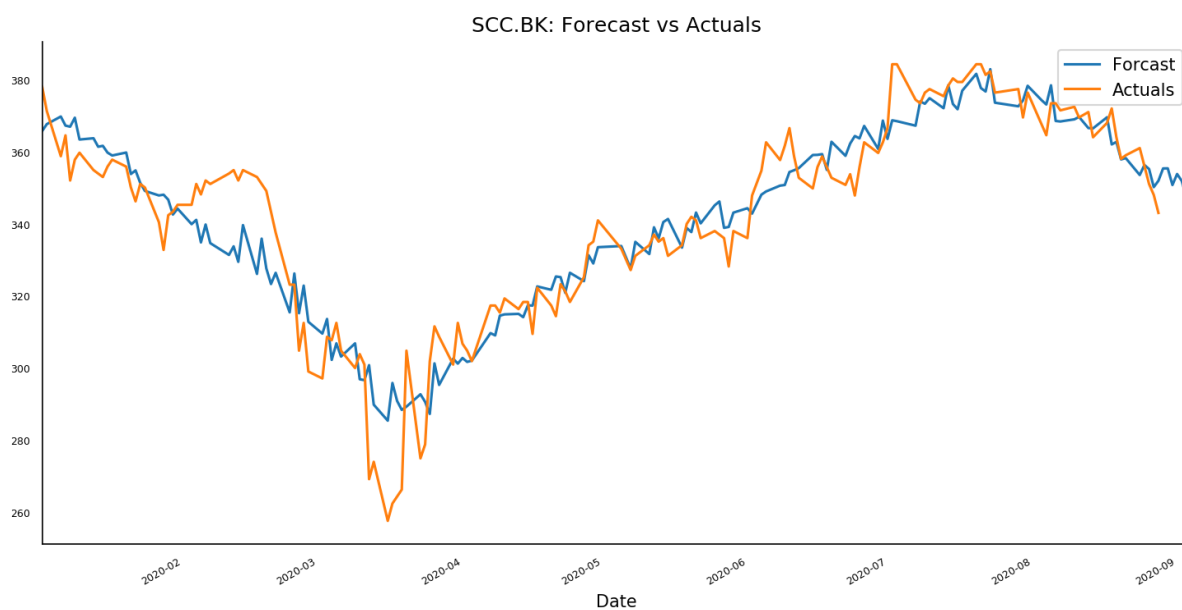
4.2.1.4 SCC Stock

VAR Model



รูปที่ 4.8 การทำนายราคาของหุ้น SCC โดยใช้ VAR Model

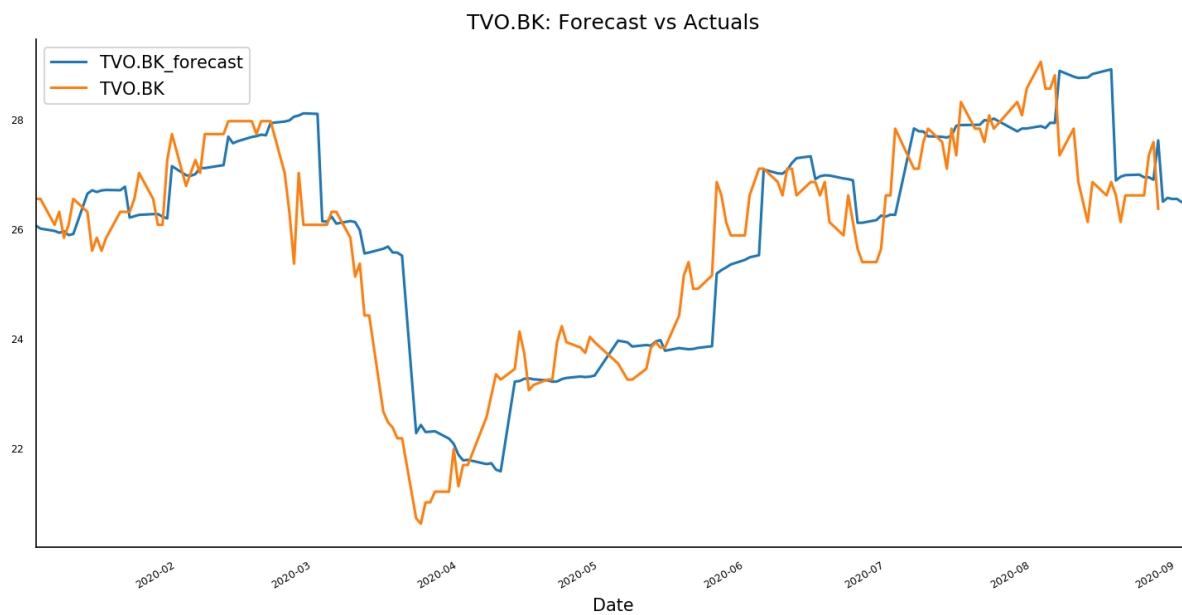
CNN+LSTM Model



รูปที่ 4.9 การทำนายราคาของหุ้น SCC โดยใช้ CNN+LSTM Model

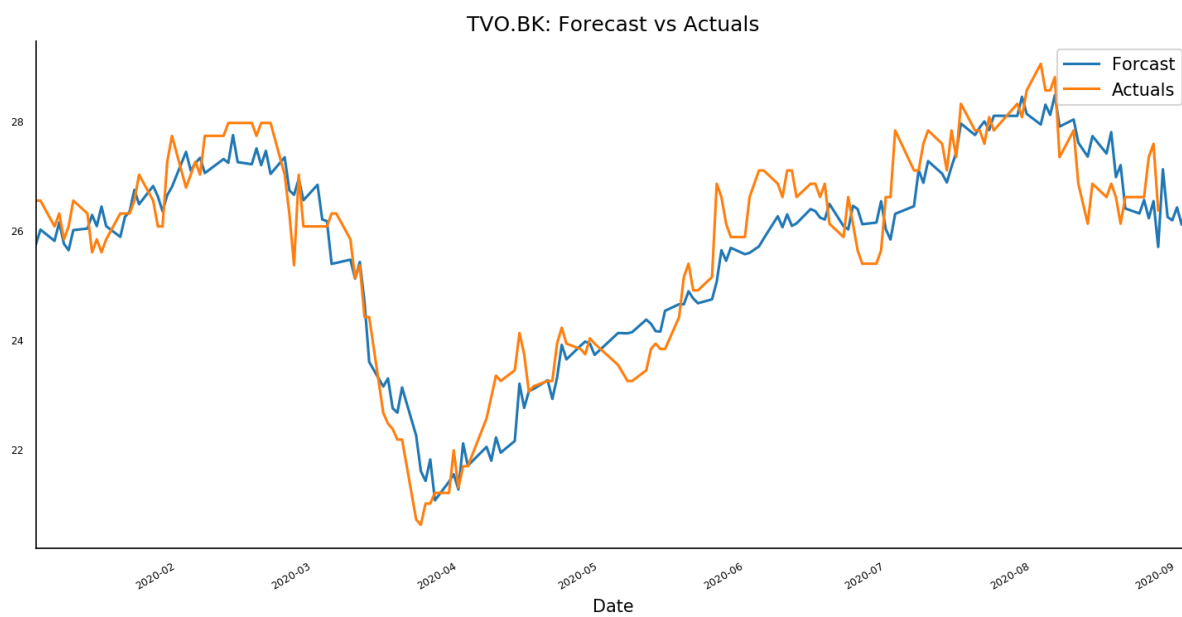
4.2.1.5 TVO Stock

VAR Model



รูปที่ 4.10 การทำนายราคาของหุ้น TVO โดยใช้ VAR Model

CNN+LSTM Model

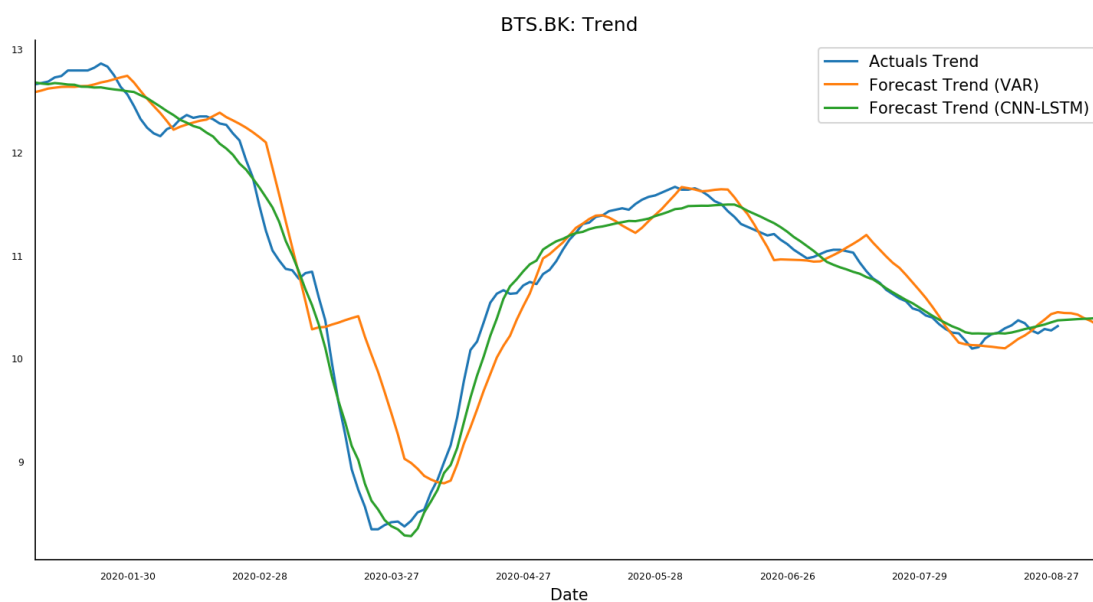


รูปที่ 4.11 การทำนายราคาของหุ้น TVO โดยใช้ CNN+LSTM Model

4.2.2 Forecast Price Trend

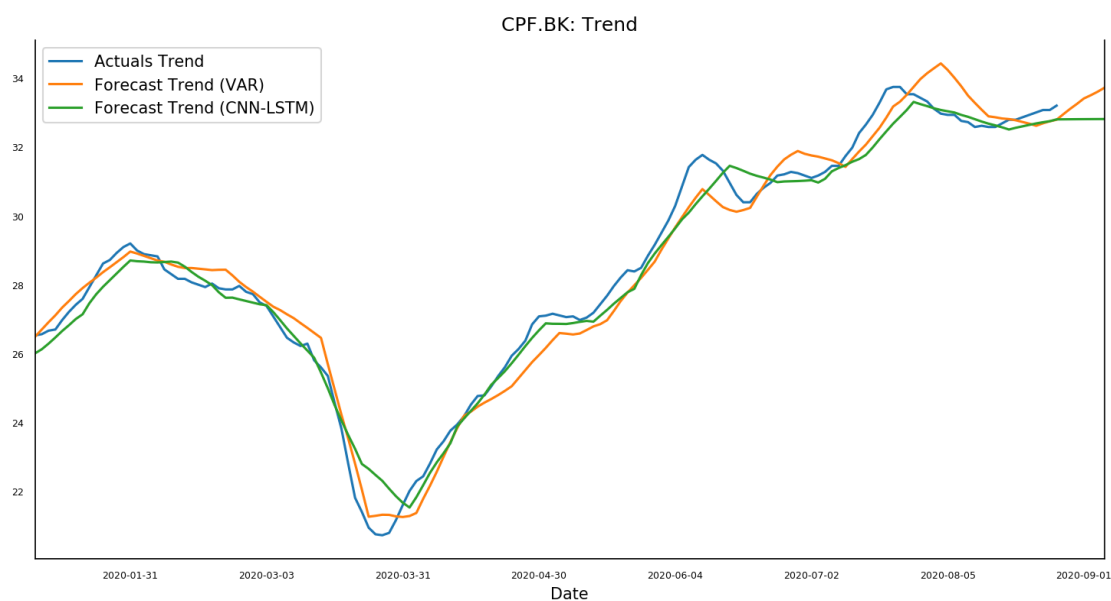
เป็นการนำราคาของหุ้นตามความเป็นจริง ,VAR model และ CNN+LSTM model มาพิจารณาและทำเป็นกราฟที่แสดงแนวโน้มการขึ้นลงของราคาเพื่อสังเกตว่าราคาที่เรานำมาทำการเปลี่ยนแปลงแนวโน้มใกล้เคียงกับของจริงเพียงใด

4.2.2.1 BTS Stock



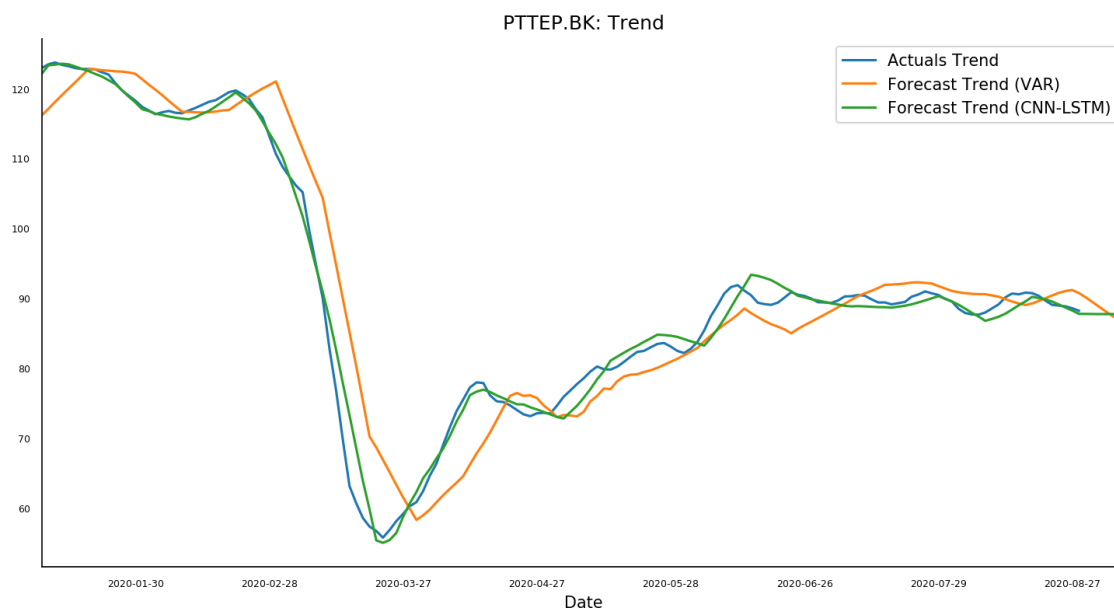
รูปที่ 4.12 การเปรียบเทียบ BTS Trend ที่ได้จากการทำนายเทียบกับของจริง

4.2.2.2 CPF Stock



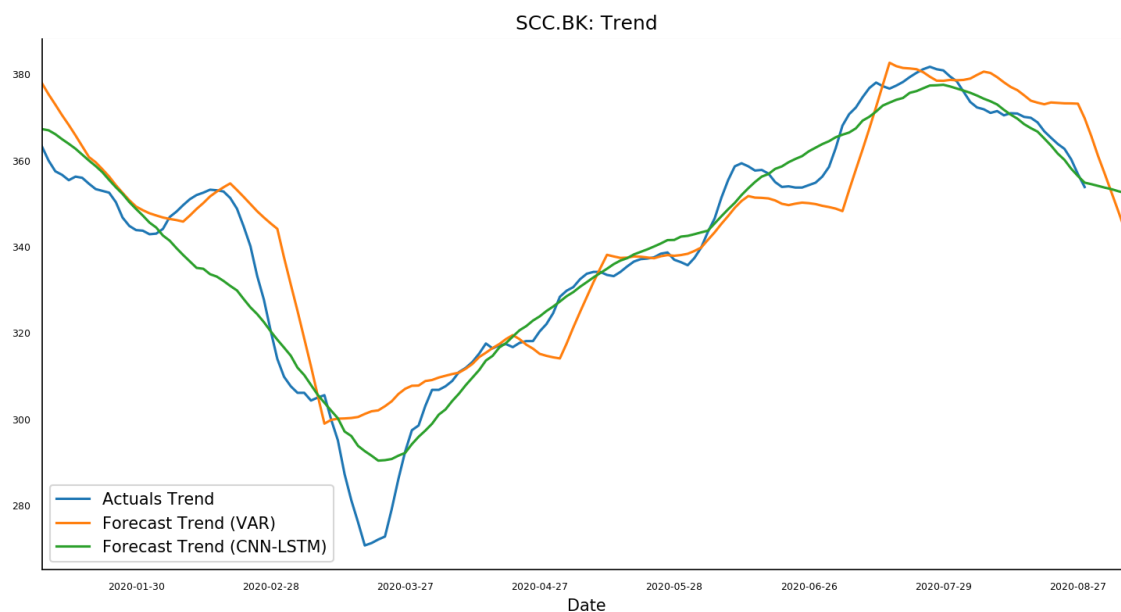
รูปที่ 4.13 การเปรียบเทียบ CPF Trend ที่ได้จากการทำนายเทียบกับของจริง

4.2.2.3 PTTEP Stock



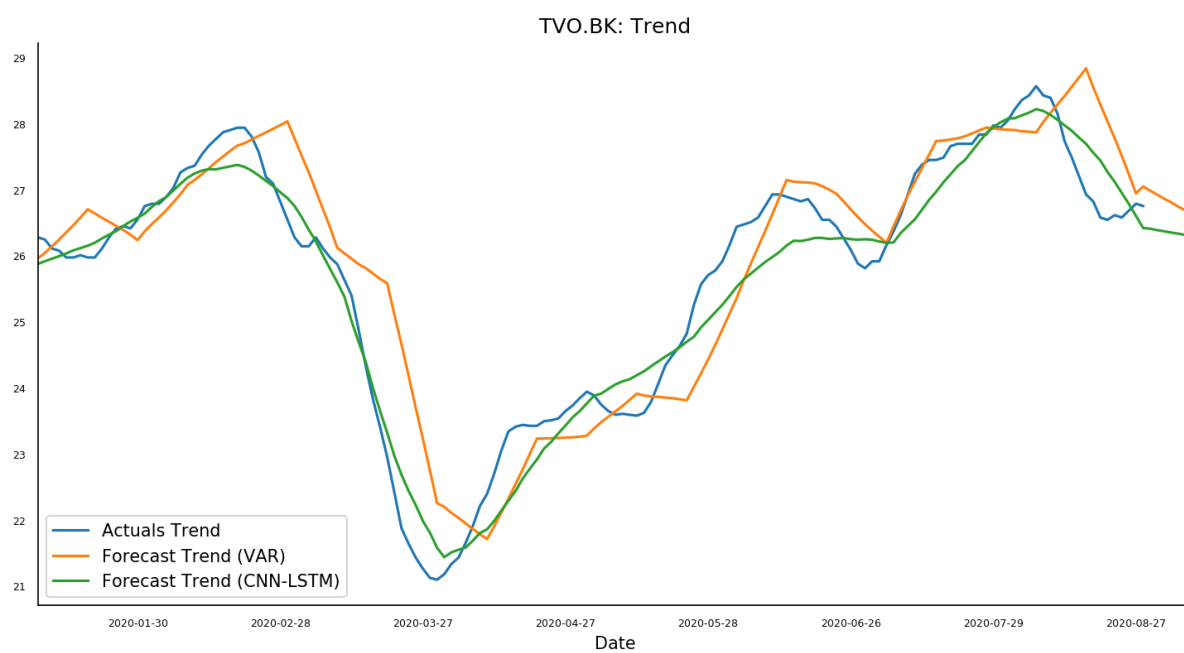
รูปที่ 4.14 การเปรียบ PTTEP Trend ที่ได้จากการทำนายเทียบกับของจริง

4.2.2.4 SCC Stock



รูปที่ 4.15 การเปรียบ SCC Trend ที่ได้จากการทำนายเทียบกับของจริง

4.2.2.5 TVO Stock

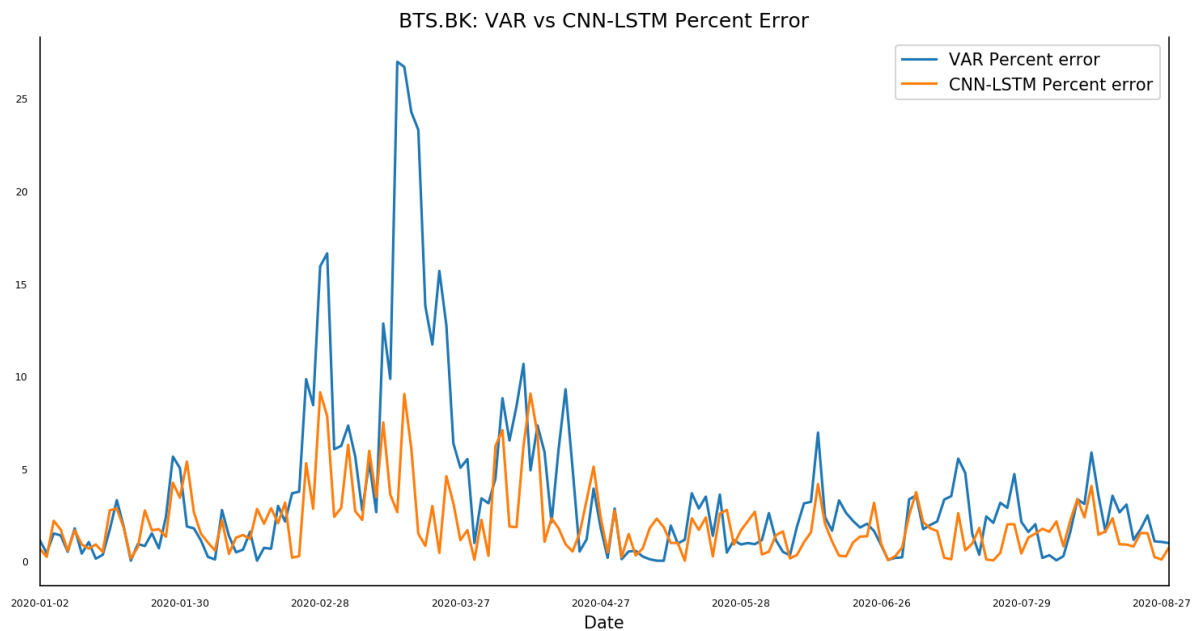


รูปที่ 4.16 การเปรียบเทียบ TVO Trend ที่ได้จากการทำนายเทียบกับของจริง

4.2.3 Forecast Price Error

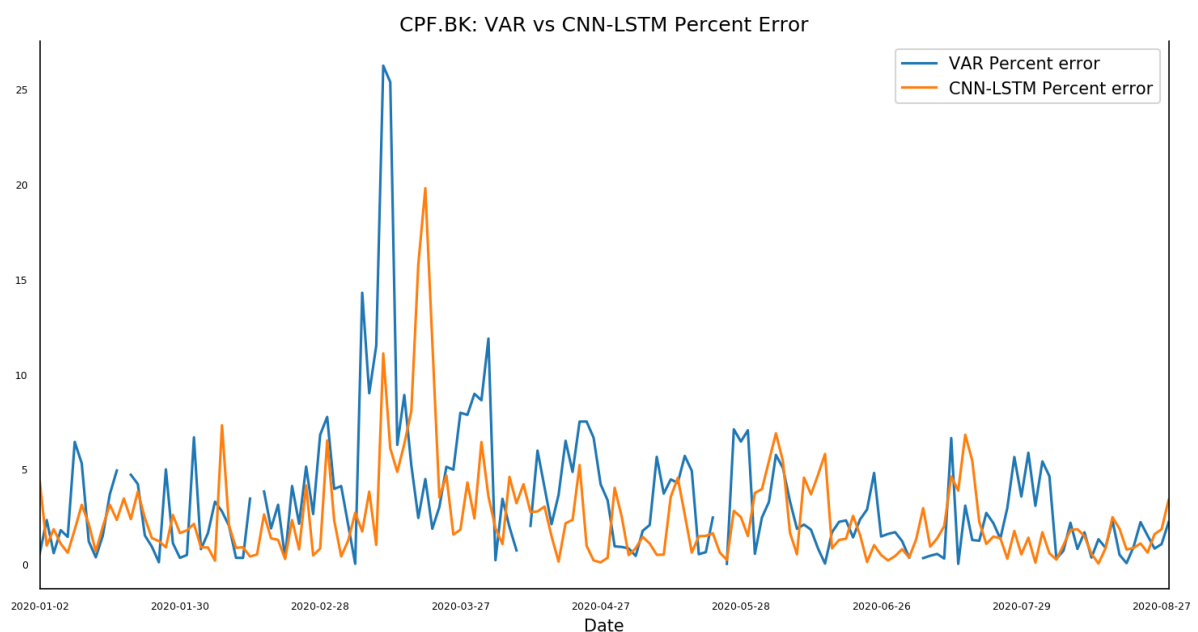
นำความคลาดเคลื่อนที่คำนวณจากการนำแบบจำลองแต่ละแบบเทียบกับของจริงมา และนำมาค่าที่ได้จากคำนวณทั้ง 2 โมเดลมาสร้างเป็นกราฟเพื่อเปรียบเทียบกัน

4.2.3.1 BTS Stock



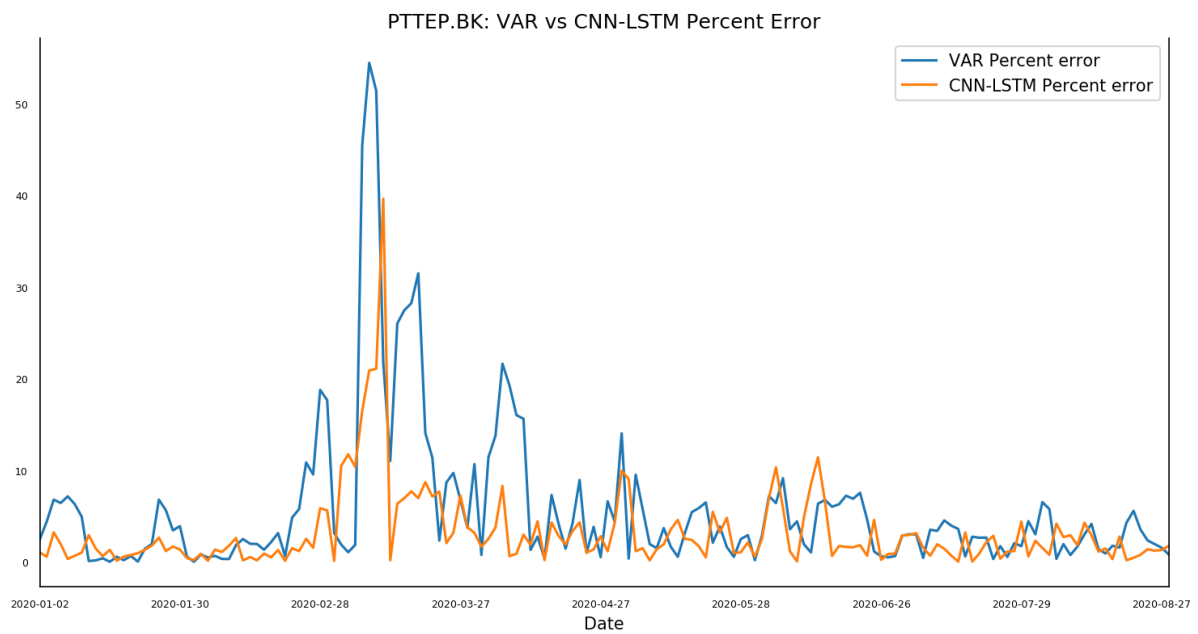
รูปที่ 4.17 ค่าคลาดเคลื่อน BTS Price ที่ได้จากการทำนายราคา (%)

4.2.3.2 CPF Stock



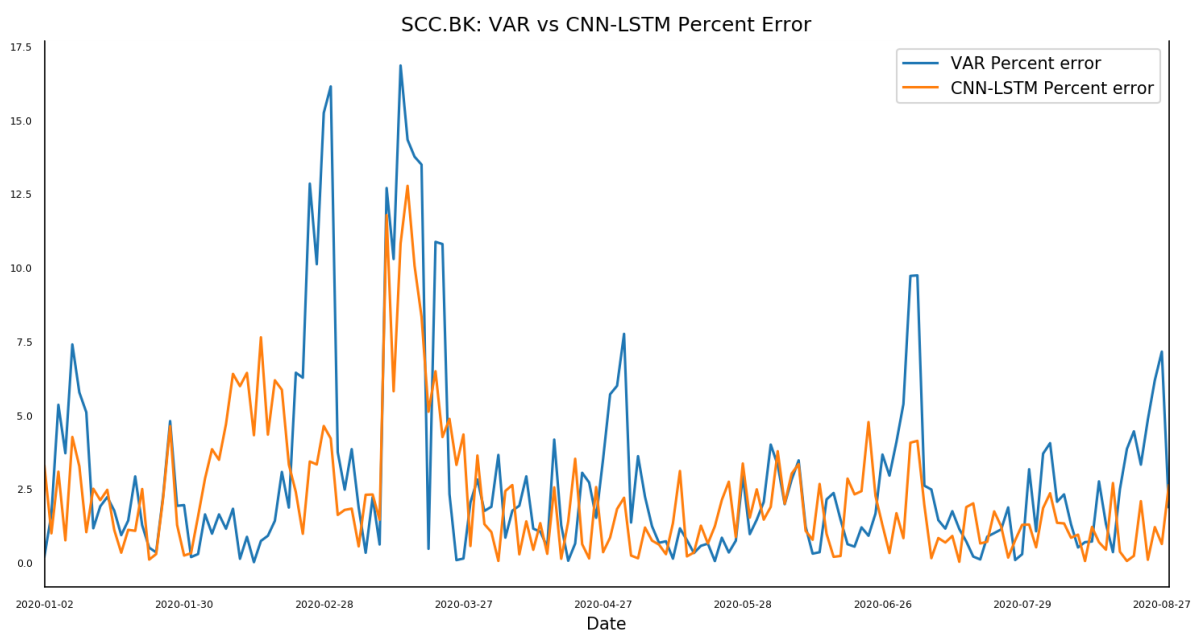
รูปที่ 4.18 ค่าคลาดเคลื่อน CPF Price ที่ได้จากการทำนายราคา (%)

4.2.3.3 PTTEP Stock



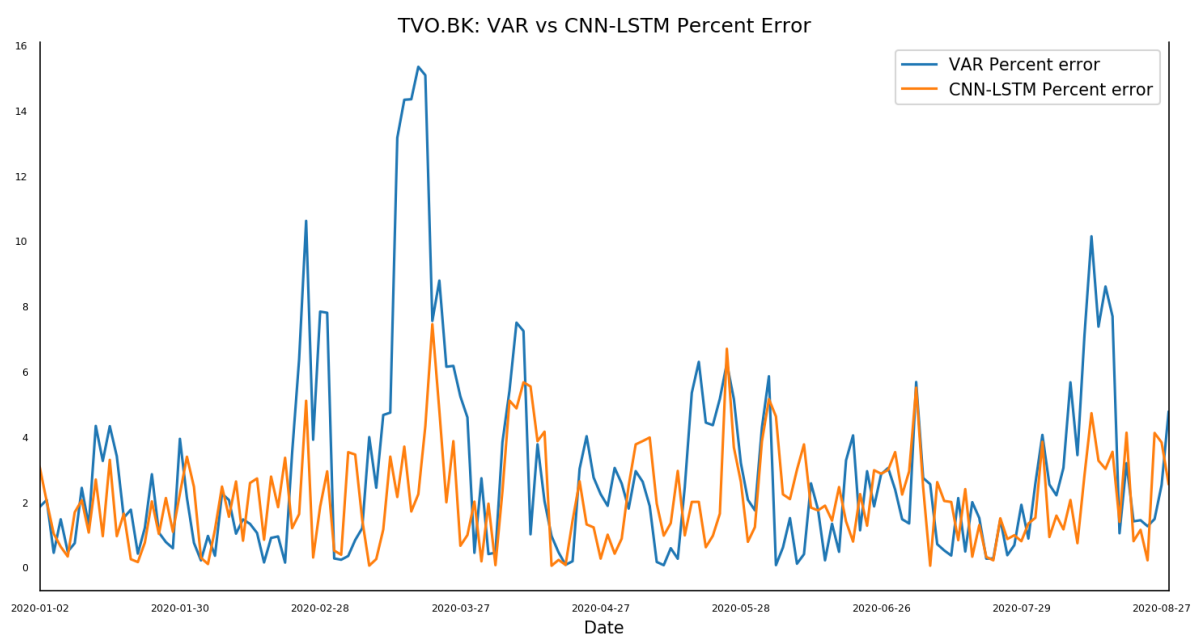
รูปที่ 4.19 ค่าคลาดเคลื่อน PTTEP Price ที่ได้จากการทำนายราคา (%)

4.2.3.4 SCC Stock



รูปที่ 4.20 ค่าคลาดเคลื่อน SCC Price ที่ได้จากการทำนายราคา (%)

4.2.3.5 TVO Stock



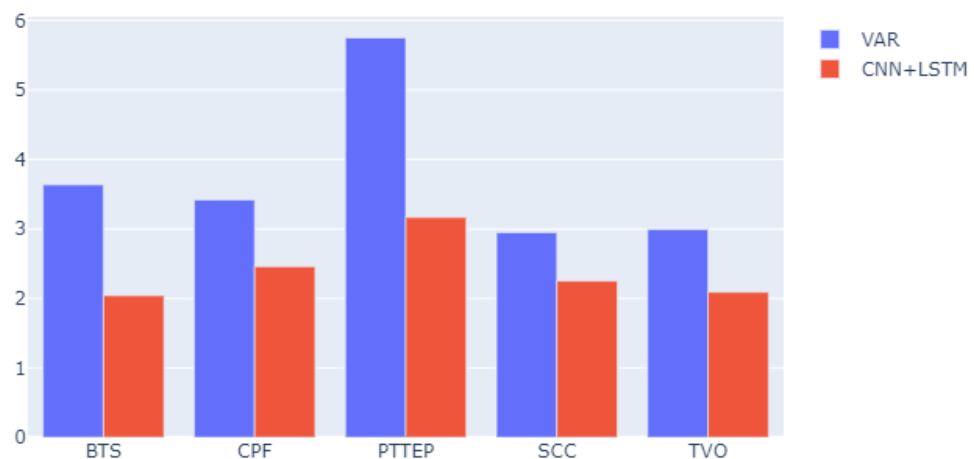
รูปที่ 4.21 ค่าคลาดเคลื่อน TVO Price ที่ได้จากการทำนายราคา (%)

4.3 ผลการประเมินแบบจำลอง

4.3.1 Mean absolute percentage error (MAPE)

นำค่าความคลาดเคลื่อนเป็นเปอร์เซ็นต์ของราคาหุ้นของ 2 แบบจำลอง แต่ละตัวมาเปรียบเทียบกัน ได้ดังนี้

Error Price (%)



รูปที่ 4.22 ค่า MAPE ของการทำนายราคา VAR Model และ CNN+LSTM Model ของทุกหุ้น

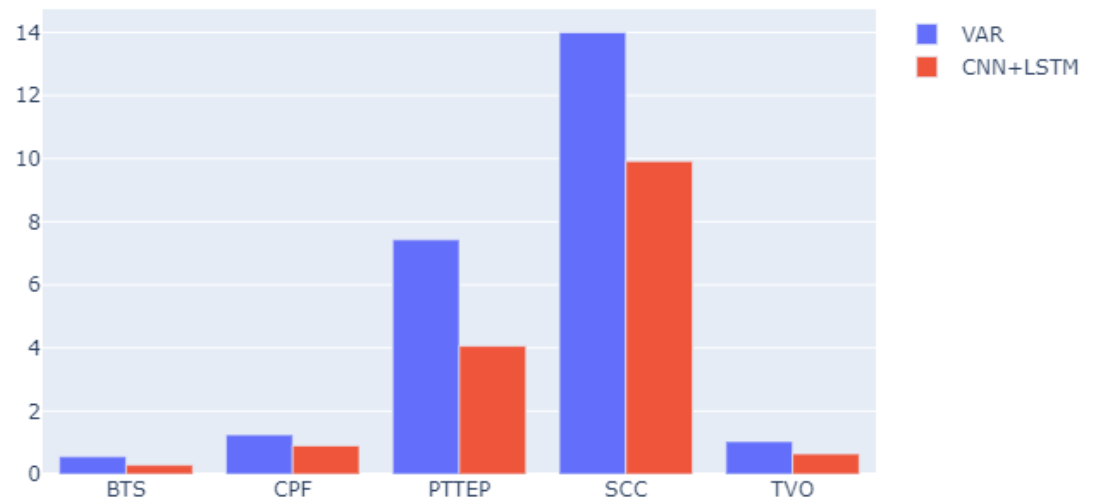
จากแผนภูมิแท่งที่แสดงข้างต้นจะเห็น แบบจำลอง VAR มีความผิดพลาดเฉลี่ยตลอดช่วงการทำนายมากกว่าแบบจำลอง CNN+LSTM โดยระบุเป็นตัวเลขได้ดังนี้

BTS Stock	CNN+LSTM Model < VAR Model	อยู่ 1.594565 %
CPF Stock	CNN+LSTM Model < VAR Model	อยู่ 0.959333 %
PTTEP Stock	CNN+LSTM Model < VAR Model	อยู่ 2.585593 %
SCC Stock	CNN+LSTM Model < VAR Model	อยู่ 0.699377 %
TVO Stock	CNN+LSTM Model < VAR Model	อยู่ 0.904901 %

4.3.2 Root mean square deviation (RMSE)

นำค่าความคลาดเคลื่อนที่เป็นหน่วยของราคาของราคาหุ้นของ 2 แบบจำลอง แต่ละตัวมาเปรียบเทียบกัน ได้ดังนี้

Error Price (บาท)



รูปที่ 4.23 ค่า RMSE ของการทำนายราคา VAR Model และ CNN+LSTM Model ของทุกหุ้น

จากแผนภูมิแท่งที่แสดงข้างต้นจะเห็น แบบจำลอง VAR มีความผิดพลาดเฉลี่ยตลอดช่วงการทำนายมากกว่าแบบจำลอง CNN+LSTM โดยระบุเป็นตัวเลขได้ดังนี้

BTS Stock CNN+LSTM Model < VAR Model อยู่ 0.268028 บาท

CPF Stock CNN+LSTM Model < VAR Model อยู่ 0.348631 บาท

PTTEP Stock CNN+LSTM Model < VAR Model อยู่ 3.369692 บาท

SCC Stock CNN+LSTM Model < VAR Model อยู่ 4.080627 บาท

TVO Stock CNN+LSTM Model < VAR Model อยู่ 0.388907 บาท

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการพัฒนาโครงการ

จากการทดลอง แสดงให้เห็นว่าการใช้ค่าความคลาดเคลื่อนเฉลี่ย (MAPE) ค่าความคลาดเคลื่อนเป็นหน่วยบาท (RMSE) เป็นตัววัดผล ทำให้แบบจำลองการทำนายราคาและเทรนด์ของหุ้นที่สร้างขึ้นด้วยการใช้แบบจำลองการเรียนรู้ของเครื่องจักรผสมผสานกับการเรียนรู้เชิงลึกให้ผลดีและแม่นยำกว่าแบบจำลองเชิงสถิติ

5.1.1 ปัจจัยที่ส่งต่อการทำนายใน VAR Model

ปัจจัยหลักที่ส่งผลกระทบต่อความแม่นยำของแบบจำลอง VAR คือ องค์ประกอบที่นำมาทำนายร่วมกับหุ้น หากองค์ประกอบไม่มีความสัมพันธ์เกี่ยวข้องหรือเกี่ยวข้องน้อยจะทำให้การทำนายที่ออกมาขึ้นอยู่กับตัวหุ้นของตนเองสูง ทำให้ประสิทธิภาพในการทำนายในช่วงเวลาที่มีการเปลี่ยนแนวโน้มกระทันหันมีผลความแม่นยำต่ำ

5.1.2 ปัจจัยที่ส่งต่อการทำนายใน CNN+LSTM

ปัจจัยหลักที่ส่งผลกระทบต่อความแม่นยำของแบบจำลอง CNN+LSTM คือ ราคาในอดีตและรูปแบบและกระบวนการฝึกฝน โดยหากลดจำนวนของ layer ที่ใช้เทรน เช่น ฝึกฝนโดยใช้ LSTM layer 2 รอบ ลดเหลือ 1 รอบ ทำให้ประสิทธิภาพการทำนายลดลง รวมถึงจำนวนข้อมูลและจำนวนรอบที่ใช้ในการฝึกฝน ทั้งนี้ หากมากเกินไป จะทำให้ข้อมูลออกมา overfitting จนการทำนายล่วงหน้า 7 วัน ออกมาแย่กว่าการคาดการณ์ของมนุษย์

5.1.3 ผลการประเมินจากการทำนายราคาของ 2 แบบจำลอง

กรณีราคาของหุ้นช่วงใดช่วงหนึ่งเปลี่ยนแปลงอย่างช้าๆ และไม่เกิดการกระโดดของราคาบ่อยจนเกินไป VAR Model จะให้ผลดีกว่า CNN+LSTM Model

กรณีราคาของหุ้นช่วงใดช่วงหนึ่งที่มีการเปลี่ยนแปลงอย่างรวดเร็วและพุ่งหรือถ่วงกระโดดบ่อย CNN+LSTM Model จะให้ผลดีกว่า VAR Model

5.2 ปัญหา/อุปสรรค และวิธีการดำเนินการแก้ไข

5.2.1 การเรียนรู้เชิงลึกผสมผสานกับการเรียนรู้ของเครื่องจักรนั้นเป็นหัวข้อที่แตกต่างกัน การนำมาผสมกันจึงทำให้เกิดยากต่อการเรียนรู้ จึงสอบถามอาจารย์และค่อยๆ เรียนรู้

5.2.2 ข้อมูลของตลาดหุ้นที่สามารถดาวน์โหลดได้ฟรีและได้รับข้อมูลอย่างถูกมีน้อย จึงทำให้ไม่สามารถทำนายหุ้นบางตัวได้เพราะไม่สามารถนำข้อมูลที่มีอยู่มาใช้ได้ จึงตัดปัจจัยบางอย่างออกไปและนำของต่างประเทศมาใช้แทน

5.2.3 แม้ว่าการทำนายราคามีผลออกมาอย่างแม่นยำแต่ในความเป็นจริงอาจจะมีเหตุการณ์บางอย่างที่ทำให้ราคาหุ้นผิดแปลกไป ทำให้ไม่สามารถทำนายช่วงเวลานั้นได้ ในกรณีเป็นไปได้ยากที่จะทำการแก้ไข

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

5.3.1 พัฒนาระบบที่จะทำนายค่าต่างๆได้มากขึ้น

5.3.2 เพิ่มจำนวนของตัวหุ้นที่จะทำนายให้มากขึ้นโดยการเพิ่มของต่างประเทศเข้ามา

5.3.3 ปรับแต่งแบบจำลองให้มีความแม่นยำมากยิ่งขึ้น

ภาคผนวก

[1] Witchapong Daroontham. “forecast ราคาหุ้นแบบง่ายๆ ด้วย Deep Learning — LSTM”. สืบค้นเมื่อ 15 สิงหาคม 2563

จาก <https://medium.com/datawiz-th/มาลอง-forecast-ราคาหุ้นแบบง่ายๆ-ด้วย-deep-learning-lstm-python-305c480db223>

[2] David M. Q. Nelson , Adriano C. M. Pereira , Renato A. de Oliveira. “Stock market's price movement prediction with LSTM neural networks”. สืบค้นเมื่อ 16 สิงหาคม 2563

จาก <https://ieeexplore.ieee.org/abstract/document/7966019>

[3] Patcharaphon Sophonpaisal. “Classification (การจำแนกประเภทข้อมูล)” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <https://medium.com/tni-university/classification-74def5f0d363>

[4] Nut Chukamphaeng. “มาทำความรู้จักกับ Reinforcement Learning แบบเบากันเถอะ” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <https://medium.com/@nutorbitx/มาทำความรู้จักกับ-reinforcement-learning-แบบเบากันเถอะ-d36e71237b8>

[5] TITIPATA. “Reinforcement learning คืออะไร” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <https://tupleblog.github.io/reinforcement-learning/>

[6] CORALINE CO. LTD. “5 Algorithms เบื้องต้นที่ช่วยให้ธุรกิจเติบโตได้” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก https://medium.com/@info_46914/5-algorithms-เบื้องต้นที่ช่วยให้ธุรกิจเติบโตได้-f24c5fcc7a9f

[7] Coraline Team. “สถิติเบื้องต้นง่ายๆ ที่จะช่วยให้คุณเข้าใจการวิเคราะห์มากขึ้น” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <https://www.coraline.co.th/single-post/Basic-Statistic-2>

[8] Nittaya Phuangrach. “การประยุกต์สมการถดถอยพหุพหุสของในงานวิจัย ทางด้านทันตสาธารณสุข” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <http://thaidj.org/index.php/JHS/article/view/392>

[9] ภัควัฒย์ จันทศิริภาส. “การประยุกต์ใช้การเรียนรู้แบบเสริมกำลังกับการวางแผนทางการเงิน” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <https://cuir.car.chula.ac.th/handle/123456789/64782>

[10] นายศุภชัย มุกดาสนิท. “การเปรียบเทียบเทคนิคการลดมิติสำหรับการจำแนกเอกสารบนโครงข่ายประสาท” . สืบค้นเมื่อ 27 สิงหาคม 2563

จาก <http://cmuir.cmu.ac.th/handle/6653943832/34595>

[11] HD COE. “Convolutional Neural Network คืออะไร” . สืบค้นเมื่อ 25 กุมภาพันธ์ 2564

จาก <https://medium.com/@hadee2531earvesdrop/convolutional-neural-network-คืออะไร-42c45f7ec16b>

[12] Stock, J. H., and M. W. Watson. 2015. Introduction to Econometrics, Third Update, Global Edition. Pearson Education Limited. “Vector Autoregressions” . สืบค้นเมื่อ 25 กุมภาพันธ์ 2564

จาก <https://www.econometrics-with-r.org/16-1-vector-autoregressions.html>

ประวัตินิสิต

ชื่อ-สกุล	ฉานณโชชน์ บุญเขียว
ชื่อโครงการ	ระบบทำนายราคาและแนวโน้มของหุ้น
ชื่อโครงการภาษาอังกฤษ	Price and Trend Prediction Systems
ภาควิชา	คอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ประวัติส่วนตัว	เกิดเมื่อ 7 กรกฎาคม 2541 (อายุ 22 ปี) ที่อยู่ นวชน อพาร์ทเมนต์ 86/20-22 ซ.งามวงศ์วาน 46 ถ.งามวงศ์วาน แขวงลาดยาว กทม. 10900
ประวัติการศึกษา	โรงเรียนบริหารแจ่มใสวิทยา 1 ปีการศึกษา 2556 โรงเรียนกรรณสูตศึกษาลัย ปีการศึกษา 2559