การคาดการณ์ผลตอบแทนในอนาคตของตราสารทุนหุ้นสามัญโดยการใช้แบบจำลองโครงข่ายประสาทเทียม และแบบจำลอง Long Short-Term Memory

วริศ ปัญญาฉัตรพร

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาการจัดการมหาบัณฑิต
วิทยาลัยการจัดการ มหาวิทยาลัยมหิดล
พ.ศ. 2564

ลิขสิทธิ์ของมหาวิทยาลัยมหิดล

สารนิพนธ์

เรื่อง

การคาดการณ์ผลตอบแทนในอนาคตขอตราสารทุนหุ้นสามัญโดยการใช้แบบจำลองโครงข่ายประสาทเทียม และแบบจำลอง Long Short-Term Memory

ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาการจัดการมหาบัณฑิต

วันที่ 28 ตุลาคม พ.ศ. 2564

วริศ ปัญญาฉัตรพร

นายวริศ ปัญญาฉัตรพร

ผู้วิจัย

ปิยภัสร ธาระวานิช

ผู้ช่วยศาสตราจารย์ปิยภัสร ธาระวานิช

Ph.D.

อาจารย์ที่ปรึกษาสารนิพนธ์

ผู้ช่วยศาสตราจารย์กิตติชัย ราชมหา

Ph.D.

ประธานกรรมการสอบสารนิพนธ์

รองศาสตราจารย์วิชิตา รักธรรม

Ph.D.

คณบดี

วิทยาลัยการจัดการ มหาวิทยาลัยมหิดล

รองศาสตราจารย์ธาตรี จันทรโคลิกา

Ph.D.

กรรมการสอบสารนิพนธ์

กิตติกรรมประกาศ

การวิจัยเรื่อง การคาดการณ์ผลตอบแทนในอนาคตของตราสารทุนหุ้นสามัญโดยการใช้แบบจำลองโครงข่ายประสาทเทียม และแบบจำลอง Long Short-Term Memory (Predicting Stock Return Using ANN and LSTM) สามารถดำเนินการจนประสบความสำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์และสนับสนุนเป็นอย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ปิยภัสร ธาระวานิช และรองศาสตราจารย์ธาตรี จันทรโคลิกา ที่ได้กรุณาให้คำปรึกษา ความรู้ ข้อคิดข้อแนะนำ และปรับปรุงแก้ไขข้อบกพร่องต่างๆ จนกระทั่งการวิจัยครั้งนี้สำเร็จเรียบร้อยด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

สุดท้ายนี้ผู้วิจัยหวังว่างานวิจัยฉบับนี้คงเป็นประโยชน์สำหรับหน่วยงานที่เกี่ยวข้อง และผู้ที่สนใจศึกษาต่อไป

วริศ ปัญญาฉัตรพร

การคาดการณ์ผลตอบแทนในอนาคตของตราสารทุนหุ้นสามัญโดยการใช้แบบจำลอง โครงข่ายประสาทเทียม และแบบจำลอง Long Short-Term Memory
PREDICTING STOCK RETURN USING ANN AND LSTM

วริศ ปัญญาฉัตรพร 6250342

กจ.ม.

คณะกรรมการที่ปรึกษาสารนิพนธ์: ผู้ช่วยศาสตราจารย์ปิยภัสร ธาระวานิช, Ph.D., ผู้ช่วยศาสตราจารย์กิตติชัย ราชมหา, Ph.D., รองศาสตราจารย์ธาตรี จันทรโคลิกา, Ph.D.

บทคัดย่อ

งานวิจัยนี้พยากรณ์ผลตอบแทนของหุ้นสามัญในตลาดหลักทรัพย์แห่งประเทศไทยโดยใช้ระบบ Machine Learning ประกอบกับข้อมูลและปัจจัยต่าง ๆ ในอดีตที่มีผลต่อราคา โดยคาดการณ์ผลตอบแทน 1วัน ,1 เดือน และ 3 เดือน โดยใช้อัลกอริทึม ที่แตกต่างกันของ Machine Learning 2 อัลกอริทึมคือ Artificial neural network (ANN) และ Long Short-Term Memory (LSTM) เพื่อทดสอบว่าอัลกอริทึมใด สามารถสร้างแบบจำลองในการพยากรณ์ผลตอบแทนของหุ้นโดยให้ค่าการพยากรณ์ที่แม่นยำที่สุด

ผลการศึกษาแสดงให้เห็นถึงความสามารถในการพยากรณ์ผลตอบแทนของหุ้นโดยการใช้ข้อมูลของบริษัท ได้แก่ ข้อมูลในงบการเงิน (Financial Statment) ข้อมูลอัตราส่วนทางการเงิน (Financial ratio) ข้อมูลปัจจัยทางเทคนิค(Technical indicator) ข้อมูลเศรษฐศาสตร์มหภาค (MacroEconomic) อัตราแลกเปลี่ยนเงินตรา (Exchange rate) ดัชนีหุ้น (Stock Index) ดัชนีทองคำ (Gold index) และข้อมูลในอดีตของการซื้อขายหุ้น(Historical Data) ประกอบกับการใช้ระบบ Machine Learning ส่งผลให้สามารถคาดการณ์ผลตอบแทนของหุ้นได้แม่นยำที่สุดเพื่อนำไปกำหนดกลยุทธ์ในการลงทุนที่เหมาะสม โดยงานวิจัยนี้แสดงให้เห็นว่าในการคาดการณ์ผลตอบแทนของหุ้นรายวัน นั้นการใช้ระบบ Machine Learning นั่นยังไม่เหมาะสมในการนำมาพยากรณ์ แต่หากเป็นการคาดการณ์ผลตอบแทนของหุ้นรายเดือน และรายไตรมาส แบบจำลอง Artificial neural network สามารถนำมาใช้ในการพยากรณ์ได้แม่นยำที่สุด

คำสำคัญ : โครงข่ายประสาทเทียม/ ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง/ การคาดการณ์ผลตอบแทนหุ้นสามัญ/ แบบจำลอง Artificial neural network/ แบบจำลอง Long Short-Term Memory

35 หน้า

สารบัญ

		หน้า
กิตติกรรมประกาศ		**ข**
บทคัดย่อ		**ค**
สารบัญตาราง		**ช**
สารบัญรูปภาพ		**ซ**
บทที่ 1	**บทนำ**	**1**
บทที่ 2	**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**	**3**
	2.1 ทฤษฎี	3
	2.2 งานศึกษาในอดีต	3
บทที่ 3	**แบบจำลอง**	**5**
	3.1 การใช้คอมพิวเตอร์เรียนรู้ด้วยตนเอง	5
	3.2 Data Normalization	5
	3.3 โครงข่ายประสาทเทียม	6
	3.3.1 การทำงานของโครงข่ายประสาทเทียมหลายชั้น	7
	3.3.2 Activation Function	10
	3.3.2.1 Sigmoid	10
	3.3.2.2 Rectified Linear Unit (ReLU)	11
	3.3.2.3 Hyperbolic Tangent (Tanh)	12
	3.3.3 การเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม	13
	3.3.4 Cost Function	13
	3.3.4.1 Mean Absolute Error (MAE)	14
	3.3.4.2 Mean Square Error (MSE)	14
	3.3.4.3 Root Mean Square Error (RMSE)	15
	3.3.4.4 Hit Rate	15
	3.3.5 Learning Rate	16
	3.3.6 Optimizer	16

สารบัญ (ต่อ)

		หน้า
	3.4 แบบจำลอง Long Short-Term Memory	18
	3.4.1 Forget gate layer	19
	3.4.2 Input gate layer	20
	3.4.3 Output gate layer	21
	3.5 วิธีการ	22
	3.6 Prediction of Stock Return	23
บทที่ 4	**ข้อมูลและตัวแปรที่ใช้ในงานวิจัย**	**24**
	4.1 Stock selection	24
	4.2 ตัวแปร	24
	4.2.1 การจัดเรียงข้อมูลใหม่	25
	4.3 Return Calculation	26
	4.4 การแบ่งข้อมูล	26
	4.5 ผลการวิจัย	27
บทที่ 5	**บทสรุปงานวิจัย**	**31**
บรรณานุกรม		**33**
ประวัติผู้วิจัย		**35**

สารบัญตาราง

ตาราง **หน้า**

4.1 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 วัน (1 Days) สำหรับทุกแบบจำลอง 29

4.2 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 เดือน (1 Month) สำหรับทุกแบบจำลอง 30

4.3 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 3 เดือน (3 Month) สำหรับทุกแบบจำลอง 30

สารบัญรูปภาพ

รูปภาพ		หน้า
3.1	โครงสร้างแบบง่ายของการทำงานของโครงข่ายประสาทเทียม	7
3.2	โครงสร้างแบบง่ายของการทำงานของโครงข่ายประสาทเทียมเพื่ออธิบายการทำงานของโครงข่ายประสาทเทียม 1 โครงข่าย	8
3.3	โครงสร้างการทำงานของโครงข่ายประสาทเทียมเพื่ออธิบายการทำงานของโครงข่ายประสาทเทียม 2 โครงข่าย	9
3.4	แสดงโครงสร้างการทำงานของ RNN	18
3.5	แสดงโครงสร้างการทำงานของ LSTM	19
3.6	แสดงโครงสร้าง Forget gate layer	19
3.7	แสดงโครงสร้าง Input gate layer	20
3.8	แสดงโครงสร้าง Output gate layer	21
4.1	แสดงการกระจายข้อมูลที่มีความถี่น้อยกว่า เข้าสู่ข้อมูลที่มีความถี่รายวัน	25
4.2	แสดงการแบ่งช่วงข้อมูลสำหรับการเรียนรู้, การทดสอบและการคาดการณ์ผลตอบแทนของตราสารทุนของแบบจำลองแต่ละตัว	27

บทที่ 1
บทนำ (Introduction)

ตั้งแต่อดีตจนถึงปัจจุบันการลงทุนในตราสารทุนหุ้นสามัญเป็นการลงทุนที่ค่อนข้างได้รับความนิยมทั้งนักลงทุนสถาบันไปจนถึงนักลงทุนรายย่อย อันเนื่องมาจากผลตอบแทนที่สูงและขั้นตอนที่ง่ายในการลงทุน แต่การที่จะเลือกหลักทรัพย์ในการลงทุน และกำหนดกลยุทธ์ในการลงทุนที่เหมาะสมเพื่อให้ได้ผลตอบแทนที่เหมาะสมกับความคาดหวังของนักลงทุน ซึ่งขึ้นอยู่กับการคาดการณ์ผลตอบแทนของตราสารทุน ในความเป็นจริงนั้นสามารถทำได้ยาก อันเนื่องมาจากปัจจัยหลายประการ อาทิเช่น ความผันผวนของสภาพตลาด ความสามารถในการบริหารของผู้บริหารซึ่งส่งผลโดยตรงต่อผลตอบแทนของตราสารทุน นโยบายของภาครัฐ รวมไปถึงปัจจัยภายนอกประเทศที่ส่งผลโดยตรงต่อความผันผวนของราคาตราสารทุน ซึ่งการคาดการณ์ผลตอบแทนในอนาคตของตราสารทุนจึงเป็นเรื่องที่สำคัญ และช่วยในการตัดสินใจลงทุนในตราสารทุนแต่ละชนิด

การศึกษาการพยากรณ์ผลตอบแทนของหุ้นในอนาคตมีหลากหลายวิธีด้วยกัน โดยการศึกษาครั้งนี้ได้เลือกเทคนิคที่ได้รับความนิยมและมีการประยุกต์ใช้อย่างแพร่หลาย คือ Machine Learning ซึ่งเป็นการทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูลในอดีต และอัลกอริทึมของ Machine Learning ที่น่าสนใจในการศึกษาถึงความสามารถในพยากรณ์ที่แม่นยำ คือ Artificial neural network (ANN) และLong Short-Term Memory (LSTM) โดยเป็นการใช้ข้อมูลปัจจัยที่มีผลต่อผลตอบแทนของหุ้น เป็นข้อมูลเพื่อนำเข้าในระบบ Machine Learning ที่มีการออกแบบให้เหมาะสมกับแบบจำลองเรียนรู้

ดังนั้นเมื่อสามารถสร้างแบบจำลองสำหรับการคาดการณ์ผลตอบแทนในอนาคตได้อย่างแม่นยำ หากอัลกอริทึมใดเหมาะสมกับการพยากรณ์ผลตอบแทนของหุ้นได้แม่นยำที่สุดในช่วงเวลาต่างๆที่กำหนด 1 วัน 1 เดือน และ3 เดือน โดยตัววัดความแม่นยำที่ใช้การการประเมินผลของโมเดล มุ่งมองเฉพาะค่าความใกล้เคียงระหว่างค่าคาดหวังกับค่าที่พยากรณ์ได้ เช่น MAE, RMSE และนอกจากตัววัดดังกล่าวแล้วตัววัดที่เรียกว่า Hit Rate ซึ่งใช้วัดความแม่นยำของทิศทางขึ้น-ลงของข้อมูล ได้ถูกนำมาใช้ร่วมด้วย โดยงานวิจัยนี้จะแสดงให้เห็นว่าโมเดล ANN นั้นแม่นยำกว่าโมเดล LSTM อย่างไรในแต่ละช่วงเวลาของการพยากรณ์เมื่อเปรียบเทียบกับโมเดล LSTM โดยเมื่อได้โมเดลที่มีความแม่นยำในการพยากรณ์มากที่สุดแล้ว สามารถนำมาใช้ในการพยากรณ์หาผลตอบแทนของหุ้นเพื่อเป็นแนวทางในการคัดเลือกหลักทรัพย์ว่าหลักทรัพย์ใดมีความน่าลงทุนและ

กำหนดกลยุทธ์ว่าควรซื้อ-ขายหลักทรัพย์เมื่อใดหรือแม้กระทั้งนำมาใช้ในการวางแผนการลงทุนในอนุพันธ์ต่างๆ ไม่ว่าจะเป็นสัญญาซื้อขายล่วงหน้า (Future) หรือสัญญาสิทธิ (Option) เพื่อดูทิศทางว่าควรซื้อหรือควรขายควรกำหนดกลยุทธ์แบบใดในการลงทุนที่เหมาะสมเพื่อให้ได้ผลตอบแทนที่เหมาะสมกับความคาดหวังของนักลงทุน

โดยทั้ง 2 แบบจำลองในการพยากรณ์ผลตอบแทนแบบรายวันนั้นแม่นยำน้อยกว่าค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้าทั้งระยะสั้น 5 วันและระยะยาว 20 วัน ในส่วนของความแม่นในการพยากรณ์ทิศทางของผลตอบแทนของหุ้นว่าผลตอบแทนจะเพิ่มขึ้น, ลดลง นั้นทั้ง 2 แบบจำลองให้การพยากรณ์ที่แม่นยำกว่าค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้าทั้งระยะสั้น 5 วันและระยะยาว 20 วัน และในส่วนของการพยาการณ์ที่มีช่วงยาวขึ้น คือ รายเดือน และรายไตรมาส พบว่าแบบจำลองที่มีความแม่นยำที่สุดก็คือแบบจำลอง Artificial neural network ซึ่งมีค่าความผิดพลาดต่ำที่สุดทั้งรายเดือนและรายไตรมาส และแม่นยำกว่าในการพยากรณ์ทิศทางของผลตอบแทนของหุ้นว่าผลตอบแทนจะเพิ่มขึ้น, ลดลง มากกว่าแบบจำลองอื่น

โครงสร้างของรายงานการวิจัยนี้ประกอบด้วย บทนำ (Introduction), งานวิจัยที่เกี่ยวข้อง (Literature Review), วิธีการ เทคนิคและรายละเอียดของแบบจำลอง (Methodology), ข้อมูลที่ใช้ในงานวิจัย (Data), ผลการวิจัย (Empirical Results) และส่วนสุดท้ายคือบทสรุปงานวิจัย (Conclusion) ตามลำดับ

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Theories and Literature Review)

2.1 ทฤษฎี (Theories)

Universal Approximation Theorem (Baker & Patil, 1998)

จุดเริ่มต้นของทฤษฎีเกิดจากการที่ The Universal Approximation Theorem บอกว่า Neural Networks สามารถประมาณฟังก์ชันใดก็ได้ เพราะความหมายของทฤษฎีคือเรื่องใดก็ตามที่คำนวณเป็นฟังก์ชันได้ก็สามารถคำนวณโดยโครงข่ายปราสาทเทียมได้ ซึ่งฟังก์ชันคือความสัมพันธ์ระหว่าง Input(x) และoutput(y) โดยที่แต่ละค่าของ Input จะสัมพันธ์ได้เพียงแค่ค่าหนึ่งของ Output เท่านั้น โดยหากเราใช่โครงข่ายปราสาทเทียมเพื่อให้ AI แยกภาพระหว่างสุนัขกับแมวนั้นฟังก์ชันที่ใช้ในการคำนวณก็จะเป็นฟังก์ชันแบบ Linear เนื่องจากการคำนวณไม่ว่าจะมีกี่ตัวแปรก็จะใช้การถ่วงน้ำหนักธรรมดาเพื่อให้ได้ผล แต่ในความเป็นจริงไม่ได้มีแค่ฟังก์ชันที่เป็น Linear แต่ยังมีฟังก์ชัน Non-Linear ที่ซบซ้อนกว่าที่สามารถนำมาคำนวณโดยโครงข่ายปราสาทเทียมได้ ซึ่งทฤษฎี Universal Approximation เป็นจริงได้นั้นเกิดจากการที่โครงข่ายปราสาทเทียมมี Activation function เพื่อช่วยให้โครงข่ายปราสาทเทียมสามารถประมาณฟังก์ชันที่เป็น Non-linear ได้ และมีหลากหลายแบบไว้เลือกใช้ บาง Network ก็มีมากกว่าหนึ่งแบบแล้วแต่ดีไซน์ ซึ่งไม่มีข้อจำกัดใดๆ(Nilson, 2019)

2.2 งานศึกษาในอดีต (Empirical Studies)

Han (2019) ได้ทำการเปรียบเทียบความแม่นยำในการทำนายผลตอบแทนของหุ้น (NASDAQ) ระหว่างอัลกอริทึม Random Forest (RF) กับ Long Short-Term Memory(LSTM) ซึ่งผลที่ได้จากการวิจัยนี้คืออัลกอริทึม LSTM ให้ผลในการพยากรณ์ผลตอบแทนของหุ้นได้ดีกว่า ต่อมาได้มีการวิจัยโดยการ การเพิ่มประสิทธิภาพพอร์ตโฟลิโอด้วยการทำนายผลตอบแทนของหุ้นโดยใช้ Deep Learning และMachine Learning ซึ่งได้มีการนำอัลกอริทึมต่างๆมารวมเข้ากับการทำ Mean–Variance (MV) model ซึ่งให้ผลว่า การทำนายผลตอบแทนของหุ้นจากอัลกอริทึม Random Forest (RF) โดยใช้ MSE, MAE และHit Rate เป็นตัววัดความแม่นยำของการพยากรณ์ ได้แม่นยำที่สุด และมีการนำผลการพยากรณ์จากการใช้อัลกอริทึม Random Forest (RF) จะทำให้การปรับพอร์ตโฟลิโอ

มีประสิทธิภาพมากที่สุดจากบรรดาอัลกอริทึมทั้งหมด เนื่องจากเมื่อใช้ร่วมกันแล้วทำให้ได้ผลตอบแทนรวมจากการลงทุน (Total Return) มากกว่าการใช้อัลกอริทึมอื่นทั้งหมด (Ma, Han, & Wang, 2021)

Patel, Shah, Thakkar, and Kotecha (2015)ได้ทำการพยากรณ์มูลค่าในอนาคตของดัชนีตลาดหุ้นสองตัว ได้แก่ CNX Nifty และ S&P Bombay Stock Exchange (BSE) จากตลาดหุ้นอินเดีย เพื่อทำการทดลอง การทดสอบอ้างอิงจากข้อมูลย้อนหลัง 10 ปีของดัชนีทั้งสอง โดยคาดการณ์ผลตอบแทนล่วงหน้า 1-10 วัน, 15 วัน และ 30 วันโดยบทความนี้เสนอวิธีการใช้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง (Machine Learning) แบบผสมสองขั้นตอน โดยใช้อัลกอริทึม Support Vector Regression (SVR) ในขั้นตอนแรก และใช้อัลกอริทึมArtificial Neural Network (ANN), Random Forest (RF) และ SVR ในขั้นตอนที่สอง ซึ่งทำให้เกิดวิธีการแบบฟิวชันสองขั้นตอนในการใช้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง ดังนี้ SVR-ANN, SVR - RF และ SVR - SVR เพื่อเปรียบเทียบประสิทธิภาพความแม่นยำของการพยากรณ์วิธีแบบขั้นตอนเดียว ANN, RF และ SVR โดยใช้ข้อมูลปัจจัยทางเทคนิค(technical indicator) 10 ตัว เป็นตัวแปรต้น ทั้งนี้ ตัวชี้วัดทางเทคนิค (technical indicator) เป็นตัวเลือกทั่วไปสำหรับนำเข้าเป็นตัวแปรต้นของการใช้ Machine Learning และใช้ Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) และRoot Mean Square Error (RMSE) เป็นตัววัดความแม่นยำของการพยากรณ์ โดยผลออกมาว่าการพยากรณ์แบบฟิวชันสองขั้นตอนนั้นแม่นยำกว่าแบบขั้นตอนเดียว โดยโมเดลที่รวมกันแล้วให้ความแม่นยำมากที่สุดคือ SVR-ANN และผลลัพธ์เมื่อมีการคาดการณ์ล่วงหน้าจำนวนวันมากขึ้น ค่าความผิดพลาดก็จะเพิ่มขึ้นด้วย (Basak, Kar, Saha, Khaidem, & Dey, 2019; Kim, 2003)

Jiemwiriyakul, Sirianuntapiboon, and Lorsubkong (2019) การนำเสนอการคาดการณ์ผลตอบแทนในตราสารทุนในตลาดหลักทรัพย์แห่ง ประเทศไทยโดยการใช้ Artificial neural network ประกอบกับข้อมูลและปัจจัยต่าง ๆ ในอดีตที่มีผลต่อราคา ในตราสารทุนและนำผลตอบแทนที่คาดการณ์ได้มาใช้ในการจัดรูปแบบและปรับพอร์ตโฟลิโอเพื่อ สร้างโอกาสการลงทุนระยะสั้น โดยทฤษฎีที่ใช้ในการจัดพอร์ตโฟลิโอ คือ การเลือกลงทุนที่จุดบนเส้น Efficient Frontier และเลือกน้ำหนักสำหรับการลงทุนโดยใช้ Maximum Sharpe Ratio เพื่อให้ผลตอบแทนของพอร์ตโฟลิโอต่อหนึ่งหน่วยความเสี่ยงสูงที่สุด ซึ่งแสดงให้เห็นถึงการคาดการณ์ผลตอบแทนตราสารทุนที่มีความแม่นยำจากการใช้ Machine Learning โดยเลือกเทคนิคแบบจำลองโครงข่ายประสาทเทียม (Neural Network) ซึ่งสามารถนำผลที่ได้ไปประยุกต์ใช้ประกอบการตัดสินใจในการปรับพอร์ตโฟลิโอให้กับนักลงทุนสถาบันหรือนักลงทุนอื่นๆได้

บทที่ 3
แบบจำลอง (Model)

3.1 การใช้คอมพิวเตอร์เรียนรู้ด้วยตนเอง (Machine Learning)

คือการทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูลแบ่งออกเป็น 2 ประเภทหลักๆคือ

การเรียนรู้โดยมีผู้ช่วยสอน (Supervised Learning) เป็นการให้ระบบคอมพิวเตอร์เรียนรู้โดยนำใส่ข้อมูลตัวแปรต้น (Input) และผลลัพธ์ตัวแปรตาม (Output) จากนั้นให้ระบบคอมพิวเตอร์เรียนรู้แบบจำลองที่เชื่อมโยงระหว่าง Input และ Output เมื่อเรียนรู้เสร็จ ระบบจะพยายามทำนายผลลัพธ์ซึ่งหากผลลัพธ์ที่ทำนายได้นั้นผิด ระบบจะพยายามแก้ไขแบบจำลองที่ใช้ทำนายไปเรื่อยๆตามข้อมูลที่เราป้อนเข้าเพื่อให้เกิดข้อผิดพลาดน้อยที่สุด

การเรียนรู้โดยไม่มีผู้ช่วยสอน (Unsupervised Learning) เป็นการให้ระบบคอมพิวเตอร์เรียนรู้ด้วยการจำแนกข้อมูล วิธีนี้เราจะใส่เพียงข้อมูลนำเข้า (Input) จากนั้นระบบคอมพิวเตอร์จะทำการจำแนกข้อมูล (Clustering) โดยวิธีนี้จะเน้นการใช้งานในรูปแบบการวิเคราะห์ข้อมูล (Analysis)

งานวิจัยนี้จะเลือกใช้ Random Forest (RF), Artificial neural network(ANN) และ Long Short-Term Memory (LSTM) ใน การคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว

3.2 Data Normalization

เป็นเทคนิคส่วนหนึ่งในการจัดเตรียมข้อมูลก่อนการสร้างแบบจำลองที่เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูล (Machine Learning) ซึ่งเป้าหมายของการทำ Data Normalization เป็นการเปลี่ยนข้อมูลตัวแปรต้นที่เป็นตัวเลขให้อยู่ในช่วงความถี่ที่เป็นมาตรฐานเดียวกันทั้งหมดเพื่อเพิ่มความแม่นยำในการพยากรณ์ เช่น ข้อมูลราคาซื้อขายของหุ้นอยู่ในช่วง 200 - 500 หรือ ข้อมูลสินทรัพย์ในงบการเงินอยู่ในช่วง 100,000,000 – 1,000,000,000 ซึ่งจะถูกปรับให้อยู่ในช่วงความถี่มาตรฐานเดียวกัน เช่น ช่วง 0 - 1 เป็นต้น ทั้งนี้ทำให้แบบจำลองเรียนรู้ข้อมูลได้ดีขึ้น การทำ Data Normalization นั้นทำได้หลายวิธี แต่ในงานวิจัยนี้ใช้แบบ Rescaling หรือ Min-Max Normalization ที่จะปรับช่วงข้อมูล ให้เป็นอยู่ในช่วง 0 - 1 โดยพิจารณาจากความห่างของค่าต่ำสุดเป็นสัดส่วนต่อค่าของช่วงข้อมูลทั้งหมด ตามสมการที่ 3.1 และมีการแปลงค่ากลับค่าจริงตามสมการ 3.2

สมการ 3.1 แสดงการทำ Data Normalization ของตัวแปรต้นทุกตัวที่ใช้ในแบบจำลองโครงข่ายประสาทเทียม

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

สมการ 3.2 แสดงการแปลงค่าที่พยากรณ์ได้กลับเป็น Original value หลังจากทำ Data Normalization เพื่อใช้ในแบบจำลองโครงข่ายประสาทเทียม

$$Output\ value = \tilde{y} \times [max(x) - min(x)] + \min(x)$$

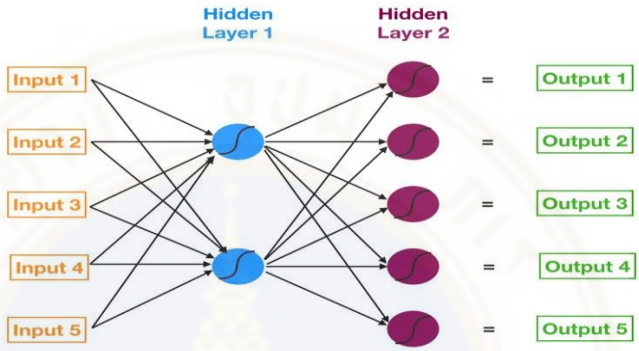
$\tilde{y}$ คือ ค่าที่พยากรณ์ได้จากแบบจำลองโครงข่ายประสาทเทียม

3.3 โครงข่ายประสาทเทียม (Artificial Neural Network, “ANN”)

เป็นระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูล (Machine Learning) จัดอยู่ในประเภทที่เรียนรู้โดยมีผู้ช่วยสอนซึ่งต้องมีข้อมูลมาสอนระบบ (Supervised Learning) เป็นแนวคิดซึ่งจำลองมาจากรูปแบบการประมวลผลของสมองมนุษย์ โดยสมองของมนุษย์นั้นจะมีหน่วยประมวลผลขนาดเล็กอยู่มากมาย เพื่อช่วยให้มนุษย์สามารถ คิด วิเคราะห์ แยกแยะได้อย่างรวดเร็ว แต่โดยหลักการคอมพิวเตอร์ถูกออกแบบมาให้ทำงานตามคำสั่ง ดังนั้นหากต้องการให้คอมพิวเตอร์สามารถเรียนรู้ได้ จึงต้องจำลองการเรียนรู้ของมนุษย์ให้กับคอมพิวเตอร์ด้วยโครงข่ายประสาทเทียม ซึ่งโครงสร้างประกอบด้วย Input Layer, Hidden Layer และ Output Layer ภายในแต่ละ Layer จะประกอบด้วยโหนด (Node) ซึ่งความซับซ้อนของจำนวน Layer และ Node ขึ้นอยู่กับการออกแบบและความเหมาะสมในการทำงานรวมทั้งการทดสอบผล ซึ่งในงานวิจัยนี้ออกแบบให้มี 1 Input Layer (305 Node), 1 Hidden Layer (305 Node) และ1 Output Layer (3 Node) หรือโครงสร้าง 305:305:3 ซึ่ง 1 Input Layer (305 Node) จำนวนโหนดเป็นไปตามจำนวนชุดข้อมูลของตัวแปรทั้งหมดจำนวน 305 ชุด เพื่อให้แบบจำลองเรียนรู้ และ1 Hidden Layer (305 Node) อ้างอิงตามจำนวนโหนดตาม Input Layer และ 1 Output Layer (3 Node) จำนวนโหนดตามการพยากรณ์ผลตอบแทนของหุ้น 3 แบบ 1 วัน 1 เดือน และ3 เดือน เนื่องจากการใช้ Machine Learning สามารถให้แบบจำลองคำนวณในรอบเดียวได้เลยสำหรับผลลัพธ์ทั้ง 3 แบบ

การสร้างแบบจำลองนี้เริ่มต้นจากป้อนข้อมูลตัวแปรต้น (Input Node) หรือปัจจัยต่างๆ ที่มีผลต่อผลตอบแทนของตราสารทุนแต่ละตัว และผลลัพธ์ตัวแปรตาม (Output Node) ซึ่งเป็นเหมือนเฉลยในที่นี้คือผลตอบแทน 1 วัน 1 เดือน และ3 เดือนของหุ้นแต่ละตัวโดยมี โดยตอนคำนวณหาค่าน้ำหนักและค่าความเอนเอียงเรียบร้อยแล้วในแต่ละ Node จะมีการรวบรวมผลด้วย

Activation Function และส่งออกไปเป็น Output ผ่าน Activation Function อีกรอบ และเมื่อป้อนข้อมูลหลายๆชุดให้คอมพิวเตอร์เรียนรู้เพื่อหารูปแบบสร้างเป็นแบบจำลองไว้ใช้ในการพยากรณ์หรือคาดการณ์ผลตอบแทนเมื่อป้อนตัวแปรต้นใหม่ๆเข้าไปแบบจำลองก็จะสามารถคาดการณ์ตัวแปรตามได้ใกล้เคียงค่าจริงที่เกิดขึ้นได้โดยวัดจากฟังก์ชันเปรียบเทียบค่าจริงกับค่าพยากรณ์ หรือฟังก์ชันวัดความคลาดเคลื่อน (Cost Function) ซึ่งลำดับต่อไปจะกล่าวถึงขั้นตอนการทำงานแต่ละส่วนโดยละเอียด



รูปภาพ 3.1 โครงสร้างแบบง่ายของการทำงานของโครงข่ายประสาทเทียม

ที่มา: https://towardsdatascience.com/understanding-neural-networks-19020b758230

3.3.1 การทำงานของโครงข่ายประสาทเทียมหลายชั้น (Multi-Layer Perceptron Process)

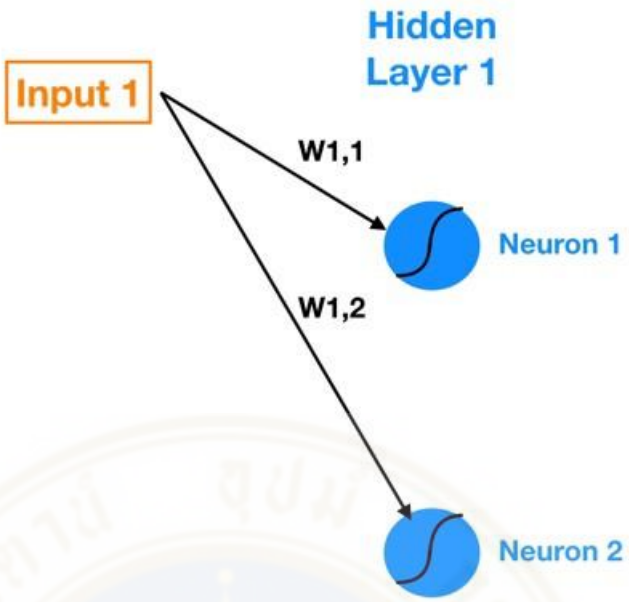
ขั้นตอนการทำงานของโครงข่ายประสาทเทียมอย่างง่าย อธิบายโดยกำหนดให้มี 1 Input Layer, 1 Hidden Layer และ 1 Output Layer มีตัวแปรต้น (Input or X or Feature) 1 ตัว จะคำนวณผ่านฟังก์ชันการวิเคราะห์ถดถอยโลจิสติกส์ (Logistic Regression) ร่วมกับน้ำหนักของตัวแปร X หรือ B1 [Weight (B1)] ที่ Hidden Layer ได้ผลลัพธ์เป็นความน่าจะเป็นของตัวแปรตาม (Predicted Probability) สามารถอธิบายเป็นสมการคณิตศาสตร์ได้ตามสมการ 3.3 โดยจากตัวอย่างใช้ฟังก์ชัน Sigmoid ซึ่งเป็นฟังก์ชันเส้นตรงอย่างง่ายโดยช่วงข้อมูลที่ออกจากฟังก์ชันจะอยู่ในช่วง 0-1 ในการวิเคราะห์ถดถอยโลจิสติกส์ ซึ่งจะกล่าวในรายละเอียดในหัวข้อ Activation Function

รูปภาพ 3.2 โครงสร้างแบบง่ายของการทำงานของโครงข่ายประสาทเทียมเพื่ออธิบายการทำงานของโครงข่ายประสาทเทียม 1 โครงข่าย

ที่มา: https://towardsdatascience.com/understanding-neural-networks-19020b758230

สมการ 3.3 อธิบายการคำนวณผ่านฟังก์ชันการวิเคราะห์ถดถอยโลจิสติกส์ ซึ่งประกอบด้วยน้ำหนักตัวแปรต้น (Weight Bi), ค่าของตัวแปรต้น (X) และค่าความคลาดเคลื่อนของสมการ (B0 or Bias or Logistic regression intercept term) โดยแสดงให้เห็นภาพการทำงานของโครงข่ายประสาทเทียม 1 เส้นโครงข่าย

ในทางปฏิบัติโครงข่ายประสาทเทียมจะทำงานร่วมกันมากกว่า 1 โครงข่าย โดยตัวอย่างจะกำหนดให้มีตัวแปรต้น 5 ตัวแปร (5 X or 5 Features) ส่งข้อมูลผ่าน Input Layer ซึ่งมี 1 Node เข้า Hidden Layer ที่มี 2 Node ตามภาพ 3.3 ซึ่งแต่ละเส้นโครงข่ายสามารถแสดงสมการการคำนวณได้ตาม สมการ 3.4 คือมีน้ำหนักของแต่ละตัวแปรต้นทั้ง 5 ตัว (W1 – W5) คูณกับค่าของตัวแปรต้น (In1- In5) และบวกกับค่าคาดเคลื่อนของสมการ (Bias) เข้าคำนวณผ่านฟังก์ชัน Sigmoid ซึ่งเป็น Activation Functionและเมื่อการคำนวณมากกว่า 1 โครงข่ายจะสามารถอธิบายได้ด้วยสมการเมทริกซ์ตามสมการ 3.5 ซึ่งผลลัพธ์ที่ได้จากสมการจะผ่าน Activation Function ได้เป็นค่าคาดการณ์ (Prediction output)



รูปภาพ 3.3 โครงสร้างการทำงานของโครงข่ายประสาทเทียมเพื่ออธิบายการทำงานของโครงข่ายประสาทเทียม 2 โครงข่าย

ที่มา: https://towardsdatascience.com/understanding-neural-networks-19020b758230

สมการ 3.4 แสดงการคำนวณของโครงข่ายประสาทเทียมที่มีตัวแปรต้น 5 ตัวแปรผ่านการคำนวณด้วย Sigmoid Activation Function

Neuron 1 Activation = Sigmoid(Z1)

Bias_Neuron = Constant Term or Intercept Term in Logistic Regression

ที่มา: https://towardsdatascience.com/understanding-neural-networks-19020b758230

สมการ 3.5 สมการเมทริกซ์แสดงการคำนวณของโครงข่ายประสาทเทียมที่มีตัวแปรต้น 5 ตัวแปร 2 โครงข่าย

$$\begin{bmatrix} W_{1,1} & W_{2,1} & W_{3,1} & W_{4,1} & W_{5,1} \\ W_{1,2} & W_{2,2} & W_{3,2} & W_{4,2} & W_{5,2} \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} + \begin{bmatrix} Bias_1 \\ Bias_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$$

ที่มา: https://towardsdatascience.com/understanding-neural-networks-19020b758230

Wi,n	คือ น้ำหนักของตัวแปรต้นตัวที่ i และโครงข่ายเส้นที่ n
Xi	คือ ค่าของตัวแปรต้นตัวที่ i
Bias n	คือ ค่าคลาดเคลื่อนของโครงข่ายเส้นที่ n
Zn	คือผลลัพธ์จากโครงข่ายเส้นที่ n ที่จะส่งเข้าไปคำนวณใน Activation Function

3.3.2 Activation Function

คือฟังก์ชันที่ใช้ในการรับผลรวมจากการประมวลผลทั้งหมดจากทุก Input Node เข้ามาพิจารณาตามกลไกการคำนวณของ Activation Function นั้นๆแล้วส่งต่อไปเป็น Output ต่อไปซึ่งในงานวิจัยนี้ได้เลือกใช้ Activation Function สองตัว คือ Rectified Linear Unit (ReLU) และ Hyperbolic Tangent (Tanh) สำหรับ Sigmoid Function นั้นจะใช้ในการอธิบายส่วนระเบียบขั้นตอนงานวิจัยเท่านั้น

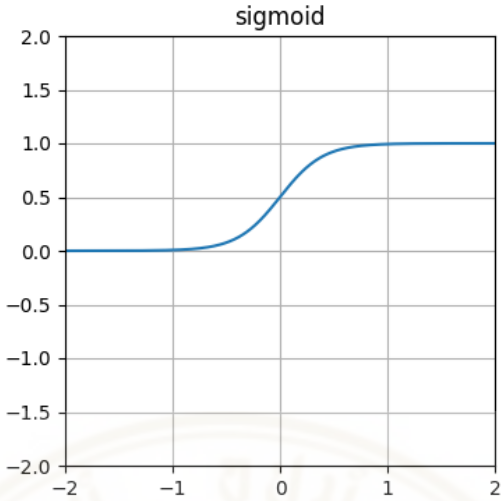
3.3.2.1 Sigmoid

เป็นฟังก์ชันเชิงเส้นอย่างง่ายโดยช่วงข้อมูลที่ออกจากฟังก์ชันจะอยู่ในช่วง 0-1 ตามสมการ 3.6 โดยข้อดีของฟังก์ชันนี้คือเข้าใจได้ง่าย สามารถใช้ได้ในงานวิเคราะห์ความน่าจะเป็น (Probability) หรืองานจำแนกกลุ่ม (Segmentation or Boolean) ข้อเสียคือถ้าตัวแปรต้นมีค่าน้อยกว่า -5 หรือมากกว่า 5 ความชันจะเข้าใกล้ 0 จนเกิดปัญหา Optimizer ไม่ปรับค่าของน้ำหนักของตัวแปรต้นในโครงข่ายประสาทเทียมในขั้นตอนเรียนรู้ของแบบจำลอง (Vanishing Gradient Problem)

สมการ 3.6 Sigmoid Activation Function Equation

$$A = \frac{1}{1 + e^{-x}}$$

ที่มา: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6



กราฟ 3.1 Sigmoid Activation Function

ที่มา: https://drallensmith.github.io/activation.html

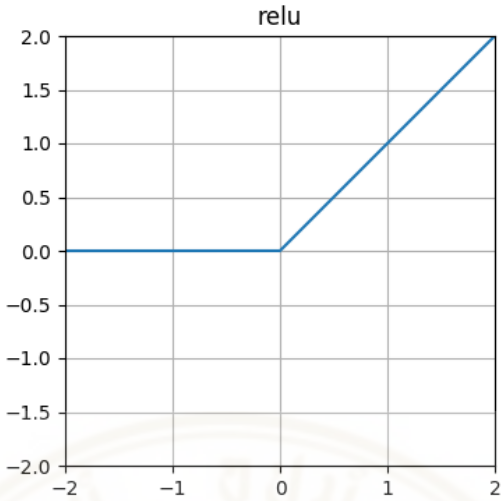
3.3.2.2 Rectified Linear Unit (ReLU)

เป็นฟังก์ชันเส้นตรงที่ถูกปรับแก้ (Rectified) ซึ่งฟังก์ชันนี้เมื่อ Input ตัวแปรต้น X เป็นบวก Slope ของกราฟจะเป็น 1 เสมอตามสมการ 3.7 สำหรับข้อดีของ ReLU คือช่วยให้ขั้นตอนการเรียนรู้ของแบบจำลองผ่าน Optimizer ซึ่งจะกล่าวในหัวข้อถัดไปนั้นทำงานได้เร็วขึ้น อีกทั้งช่วยลดปัญหาการไม่ปรับค่าของน้ำหนักของตัวแปรต้นในโครงข่ายประสาทเทียมในขั้นตอนเรียนรู้ของแบบจำลอง (Vanishing Gradient Problem) แต่ ReLU มีข้อจำกัดคืออาจจะทำให้ข้อมูลออก (Output) ไม่สมดุลคือมี Mean ไม่เท่ากับ 0 เพราะมีแต่ค่าเป็นบวกทำให้ผลการคาดการณ์หาจุดเหมาะสมได้ยาก รวมถึงช่วงข้อมูลออกเป็นได้ตั้งแต่ 0 ถึงไม่จำกัด ส่งผลให้จัดการข้อมูลออกได้ยากกว่า หากเมื่อเทียบกับข้อดีที่มากกว่าแล้วนั้นทำให้ Activation Function นี้เป็นที่นิยมในการใช้งานในแบบจำลองโครงข่ายประสาทเทียม สำหรับตัวแปรต้นก่อนเข้าแบบจำลองของงานวิจัยนี้มีค่าอยู่ในช่วง 0-1 จึงใช้ฟังก์ชันนี้ในทุกโหนดของ Input Layer และ Hidden Layer ในแบบจำลองโครงข่ายประสาทเทียม(Kanoktipsatharporn, 2019a)

สมการ 3.7 ReLU Activation Function Equation

$$A(x) = \max(0, x)$$

ที่มา: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6



กราฟ 3.2 ReLU Activation Function

ที่มา: https://drallensmith.github.io/activation.html

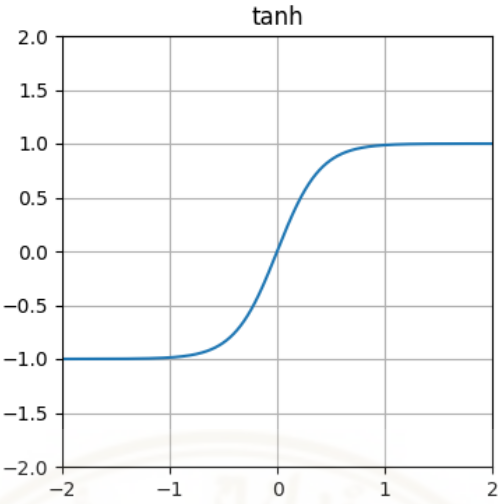
3.3.2.3 Hyperbolic Tangent (Tanh)

เป็นฟังก์ชันที่มีข้อดีในเรื่องของข้อมูลออก (Output) มีความสมดุล มีการกระจายตัวของค่าเฉลี่ย (Mean) เท่ากับ 0 ทำให้การเรียนรู้ของแบบจำลองผ่าน Optimizer ทำได้ง่ายขึ้น โดยช่วงข้อมูลออกอยู่ในช่วง -1 ถึง 1 นิยมใช้งานเพื่อช่วยเป็นการทำให้ข้อมูลอยู่ในรูปอย่างง่าย (Normalization) ฟังก์ชันสามารถอธิบายได้ง่าย ข้อจำกัดของ Tanh อาจจะเกิดปัญหา Vanishing Gradient Problem ตามที่กล่าวไว้ในหัวข้อก่อนหน้าได้ในกรณีที่ข้อมูลเข้ามีค่าน้อยกว่า -3 หรือมากกว่า 3 ซึ่งส่งผลให้ความชันเข้าใกล้ 0 โดยสมการของฟังก์ชันแสดงอยู่ใน สมการที่ 3.8 แสดงให้เห็นการกระจายตัวของข้อมูลบนเส้นฟังก์ชัน Tanh ในส่วน Output Layer ของแบบจำลองโครงข่ายประสาทเทียมใช้ฟังก์ชัน Tanh เนื่องจากความน่าจะเป็นของข้อมูลผลตอบแทนตราสารทุนทั้งหมดในงานวิจัยมีช่วงไม่เกิน -1 ถึง 1(Kanoktipsatharporn, 2019b)

สมการ 3.8 Tanh Activation Function Equation

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 = 2\,sigmoid(2x) - 1$$

ที่มา: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6



กราฟ 3.3 Tanh Activation Function

ที่มา: https://drallensmith.github.io/activation.html

3.3.3 การเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม (Learning process of a neural network)

จากการทำงานของโครงข่ายประสาทเทียมที่กล่าวมาข้างต้น ตั้งแต่ข้อมูลตัวแปรต้นส่งเข้าโครงข่ายประสาทเทียมในแต่ละ Layer เพื่อกำหนดน้ำหนักเริ่มต้นของตัวแปรต้นแต่ละตัว (Weight) รวมทั้งค่าความคลาดเคลื่อนของสมการ (Bias) จนได้ข้อมูลออกจากแต่ละเส้นโครงข่ายรวมกันจนถึง Output Layer ผ่าน Activation Function ได้เป็นค่าคาดการณ์ตัวแปรตาม (Predicted Y) หรือผลตอบแทนของตราสารทุนแต่ละตัวที่คาดการณ์ได้ หลังจากนั้นค่าคาดการณ์ตัวแปรตาม (Predicted Y) หรือผลตอบแทนของตราสารทุนแต่ละตัวที่คาดการณ์ได้นั้น จะถูกนำไปเทียบกับตัวแปรตามหรือค่าผลตอบแทนของตราสารทุนจริงที่ส่งเข้ามาให้แบบจำลองเรียนรู้ผ่านฟังก์ชันวัดความคลาดเคลื่อน (Cost Function) โดยการทำงานจะใช้อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) ในการปรับค่าน้ำหนักของตัวแปรต้นและค่าความคลาดเคลื่อนของสมการ โดยเป้าหมายเพื่อให้ค่าคลาดเคลื่อนที่ได้จากฟังก์ชันวัดความคลาดเคลื่อนมีค่าต่ำที่สุด กลไกการปรับค่าน้ำหนักของตัวแปรต้นและค่าความคลาดเคลื่อนของสมการซึ่งถูกส่งกลับไปที่โครงข่ายประสาทเทียมแต่ละเส้นโครงข่ายเพื่อทำการคำนวณใหม่

3.3.4 Cost Function

เป็นฟังก์ชันวัดความคลาดเคลื่อนซึ่งใช้ในการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม ซึ่งถูกใช้ในขั้นตอนเปรียบเทียบหาค่าคลาดเคลื่อนของตัวแปรตามที่พยากรณ์ได้กับค่าตัวแปร

ตามจริงที่ป้อนเข้ามาให้แบบจำลองเรียนรู้ ซึ่งในงานวิจัยนี้จะเป็นการเปรียบเทียบตัวแปรตามที่เป็นการคาดการณ์ผลตอบแทนรายเดือนของตราสารทุน กับผลตอบแทนรายเดือนของตราสารทุนจริงที่เกิดขึ้น โดยฟังก์ชันที่ใช้คือ Mean Absolute Error (MAE) ซึ่งจะทำงานร่วมกับอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) เพื่อปรับให้ค่าน้ำหนักตัวแปรต้น เพื่อหาค่าฟังก์ชันวัดความคลาดเคลื่อนที่ต่ำที่สุดในทุกๆรอบการเรียนรู้ (epochs) ของแบบจำลองโครงข่ายประสาทเทียม ทั้งนี้ฟังก์ชันวัดความคลาดเคลื่อนยังใช้เป็นผลในการทดสอบแบบจำลองโครงข่ายประสาทเทียมเพื่อเปรียบเทียบและเป็นการวัดผลความแม่นยำในการคาดการณ์ผลตอบแทนของแบบจำลองโครงข่ายประสาทเทียมอีกด้วยสำหรับรายละเอียดของฟังก์ชันวัดความคลาดเคลื่อนที่ใช้ในงานวิจัยมีดังนี้

3.3.4.1 Mean Absolute Error (MAE)

คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์เช่นเดียวกัน ซึ่งเป็นตัววัดหน่วยอิสระ (Unit-free measure) เพื่อวัดค่าสัมบูรณ์ของค่าความคลาดเคลื่อนเฉลี่ยของผลพยากรณ์ ซึ่งใช้วัดค่าคลาดเคลื่อนที่เกิดขึ้นบนข้อมูลที่มีการเปลี่ยนแปลงเพียงเล็กน้อยได้ดี ถูกใช้ในขั้นตอนประเมินความแม่นยำของแบบจำลองเช่นเดียวกัน

สมการ 3.9 Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|(r_t - \hat{r}_t)|$$

n คือ ช่วงของ time series (โดยมีหน่วยตามงานวิจัยคือ รายวัน, เดือน และไตรมาส)

r_t คือ ผลตอบแทนของหุ้น ที่เกิดขึ้นในช่วงทดสอบ t

$\hat{r}_t$ คือ ค่าพยากรณ์ของผลตอบแทนของหุ้น ที่พยากรณ์ได้ในช่วงทดสอบ t

3.3.4.2 Mean Square Error (MSE)

คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์ โดยแสดงเป็นผลเฉลี่ยของค่าคลาดเคลื่อนของทุกๆจุดเวลาที่แบบจำลองทำการพยากรณ์บนหนึ่งช่วงข้อมูล ซึ่งฟังก์ชันใช้ในแบบจำลองโครงข่ายประสาทเทียมเป็น Cost Function ที่ทำงานร่วมกับอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) เพื่อปรับให้ค่าน้ำหนักตัวแปรต้นในขั้นตอนการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียมในงานวิจัยนี้ อีกทั้งจะใช้ในการประเมินความแม่นยำของแบบจำลองในการพยากรณ์ผลตอบแทนตราสารทุนทั้งในขั้นตอนเรียนรู้, ตรวจสอบ และทดสอบแบบจำลอง

สมการ 3.10 Mean Square Error

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(r_t - \hat{r}_t)^2$$

n คือ ช่วงของ time series (โดยมีหน่วยตามงานวิจัยคือ รายวัน, เดือน และไตรมาส)

r_t คือ ผลตอบแทนของหุ้น ที่เกิดขึ้นในช่วงทดสอบ t

$\hat{r}_t$ คือ ค่าพยากรณ์ของผลตอบแทนของหุ้น ที่พยากรณ์ได้ในช่วงทดสอบ t

3.3.4.3 Root Mean Square Error (RMSE)

คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์เช่นเดียวกัน แต่มีการเพิ่ม Square Root ในสมการเพื่อให้สะท้อนค่าเฉลี่ยของค่าคลาดเคลื่อนแต่ละจุดที่มีขนาดใหญ่ RMSE จะให้น้ำหนักของค่าคลาดเคลื่อนดังกล่าวมากกว่า จึงช่วยในการเปรียบเทียบผลได้ดีมากขึ้น ซึ่งฟังก์ชันนี้ใช้ขั้นตอนในการประเมินความแม่นยำของแบบจำลองในการพยากรณ์ผลตอบแทนตราสารทุนทั้งในขั้นตอนเรียนรู้, ตรวจสอบ และทดสอบแบบจำลองซึ่งเป็นตัวประเมินร่วมกับ Mean Square Error (MSE)

สมการ 3.11 Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(r_t - \hat{r}_t)^2}$$

n คือ ช่วงของ time series (โดยมีหน่วยตามงานวิจัยคือ รายวัน, เดือน และไตรมาส)

r_t คือ ผลตอบแทนของหุ้น ที่เกิดขึ้นในช่วงทดสอบ t

$\hat{r}_t$ คือ ค่าพยากรณ์ของผลตอบแทนของหุ้น ที่พยากรณ์ได้ในช่วงทดสอบ t

3.3.4.4 Hit Rate

คือเปอร์เซ็นต์ที่นับจากจำนวนครั้งที่ผลตอบแทนของหุ้นที่พยากรณ์มีทิศทางเดียวกันกับผลตอบแทนจริง ต่อจำนวนครั้งที่ผลตอบแทนของหุ้นทั้งค่าพยากรณ์และค่าผลตอบแทนจริงไม่เป็น 0 ณ จุดเวลาใดๆ โดยมีทั้งตัววัดที่ทิศทางการคาดการณ์นั้นใกล้เคียงกับผลตอบแทนจริงแบบทิศทางผลตอบแทนสูงขึ้น (Hit rate +), ทิศทางในการพยากรณ์นั้นเหมือนกับผลตอบแทนจริงแบบทิศทางผลตอบแทนลดลง (Hit rate -) และวัดทิศทางในการพยากรณ์นั้นเหมือนกับผลตอบแทนจริงทั้งสองทิศทาง (Hit rate) ซึ่งฟังกชันเหล่านี้ใช้ในการประเมินผลคาดการณ์ของแบบจำลองว่ามีความสามารถในการคาดการณ์ในสภาวะตลาดที่ราคาหรือผลตอบแทนของตราสารทุนเป็นช่วงค่าขึ้นและขาลงได้แตกต่างกันหรือไม่

สมการ 3.12 Hit rate ใช้ประเมินความแม่นยำในการคาดการณ์ทิศทางรวมของการขึ้นลงของผลตอบแทน

$$H_R = \frac{count_{t=1}^{n}(r_t \times \hat{r}_t > 0)}{count_{t=1}^{n}(r_t \times \hat{r}_t \neq 0)}$$

สมการ 3.13 Hit rate + ใช้ประเมินความแม่นยำในการคาดการณ์ทิศทางของการเพิ่มขึ้นของผลตอบแทน

$$H_{R+} = \frac{count_{t=1}^{n}(r_t > 0 \; AND \; \hat{r}_t > 0)}{count_{t=1}^{n}(\hat{r}_t > 0)}$$

สมการ 3.14 Hit rate – ใช้ประเมินความแม่นยำในการคาดการณ์ทิศทางของการลดลงของผลตอบแทน

$$H_{R-} = \frac{count_{t=1}^{n}(r_t < 0 \; AND \; \hat{r}_t < 0)}{count_{t=1}^{n}(\hat{r}_t < 0)}$$

n คือ ช่วงของ time series (โดยมีหน่วยตามงานวิจัยคือ รายวัน, เดือน และไตรมาส)

r_t คือ ผลตอบแทนของหุ้น ที่เกิดขึ้นในช่วงทดสอบ t

$\hat{r}_t$ คือ ค่าพยากรณ์ของผลตอบแทนของหุ้น ที่พยากรณ์ได้ในช่วงทดสอบ t

3.3.5 Learning Rate

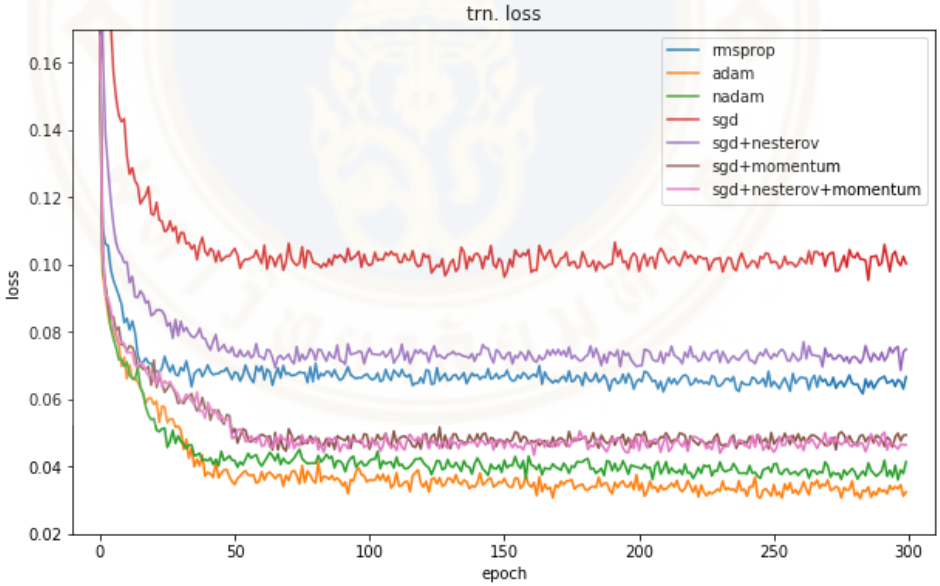
คือ Hyperparameter หรือพารามิเตอร์ต่าง ๆ ที่ผู้ใช้สามารถกำหนดเองได้ก่อนที่โมเดลจะทำการเรียนรู้ เช่น ค่า Learning Rate ที่ควบคุมว่าจะเปลี่ยนแปลงค่า weight ของ Model ใน 1 Step ของ Model Training ค่า Learning ส่งผลต่อ Model Training โดยถ้าปรับไว้มากเกินไปจะทำให้คำตอบของ Model ไม่ลู่เข้าสู่คำตอบจริงหรือคำตอบที่มันควรจะเป็น แต่ถ้าปรับไว้น้อยเกินไปจะทำให้ Model ของเราลู่เข้าสู่คำตอบช้าและใช้เวลาในการ Train นานขึ้น ซึ่งค่า Learning rate ที่เหมาะสมจะถูกปรับโดยการ Optimizer

3.3.6 Optimizer

เป็นอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) ทำหน้าที่เป็นกลไกการปรับปรุงค่าน้ำหนักของตัวแปรต้นต่างๆ รวมถึงค่าคลาดเคลื่อน (Bias) ในขั้นตอนการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียมทำให้ Output หรือผลคาดการณ์ของแบบจำลองที่ได้เข้าใกล้ค่าจริงที่กำหนดให้แบบจำลองใช้เรียนรู้ ซึ่งการทำงานของอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) คือ Gradient Descent ซึ่งเป็นหลักการหาความชันของฟังก์ชันวัดความคลาดเคลื่อน (Cost Function) เพื่อเพิ่มประสิทธิภาพช่วยให้ทราบวิธีการเปลี่ยน Weight และ Learning Rate ของ Neural Network เพื่อลดค่า Loss หรือ Error (Nakwijit, 2020)

สำหรับอัลกอริทึมการเพิ่มประสิทธิภาพที่ใช้ในแบบจำลองของงานวิจัยนี้นั่น เลือกใช้ ADAM (Adaptive Moment Estimation) เป็นอัลกอริทึมการเพิ่มประสิทธิภาพที่สามารถปรับ Learning rates ที่เหมาะสมสำหรับแต่ละน้ำหนักของตัวแปรต้นหรือพารามิเตอร์ในแต่ละครั้งของการเรียนรู้ของแบบจำลองได้และมีความสามารถในการแก้ไขปัญหาของ Optimizer ตัวเดิมๆในอดีต เช่น ปัญหาการ Decaying ของ Gradient Descent จากการใช้ Learning Rate ที่ไม่เหมาะสมกล่าวคือไม่สามารถหาจุดที่ต่ำสุดใน Cost Function ได้ โดยจากการศึกษาเปรียบเทียบจะพบว่า ADAM เป็น Optimizer ที่เหมาะสมที่สุดในการใช้งาน ณ ปัจจุบัน (Floydhub & Keras, 2017) แสดงโดยกราฟ 3.4 ซึ่งเป็นการเปรียบเทียบความสามารถของ Optimizer หลายๆตัวจะเห็นได้ว่า ADAM มีความสามารถในการลดค่าคลาดเคลื่อนของ Cost Function ได้ดีที่สุด

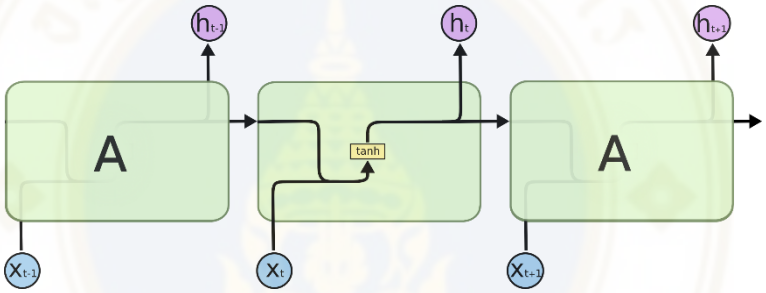
กราฟ 3.4 แสดงผลเปรียบเทียบการทำงานของ Optimizer ตัวต่างๆ ซึ่งจะเห็นได้ว่า ADAM มีความสามารถในการลดค่าคลาดเคลื่อนของ Cost Function ได้ดีที่สุด โดย Loss ตามแกน Y คือค่าคลาดเคลื่อนของ Cost Function ในแต่ละรอบการทำงาน (epochs) ตามแกน X โดย Optimizer ที่นำมาเปรียบเทียบจะเป็น Optimizer ที่นิยมใช้ในอดีต เช่น Stochastic gradient descent (SGD), RMSprop, Nesterov momentum เป็นต้น ซึ่งในที่นี่จะไม่ได้กล่าวถึงรายละเอียดของ Optimizer ที่ไม่ได้ใช้งาน



ที่มา: https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/?source=post_page-----9feb5a87e3b2----------

3.4 แบบจำลอง Long Short-Term Memory (LSTM)

เป็นเทคนิคหนึ่งที่ถูกพัฒนาจาก Recurrent neural network (RNN) ซึ่ง RNN นั้นมีหลักการทำงาน คือ การนำ Output ที่ได้จากการคำนวณจากโหนดก่อนหน้านี้กลับมาใช้เป็นข้อมูล Input ของโหนดถัดไป ซึ่งแต่ละโหนดของ RNN นั้นจะมีข้อมูลที่เข้ามา 2 ส่วน คือ ข้อมูล Input ของโหนดนั้นๆกับ Output ที่ผ่านการคำนวณจากโหนดก่อนหน้า โดยข้อมูลทั้ง 2 ชุดที่เข้ามาในโหนดจะถูกรวมเข้าด้วยกัน ก่อนจะถูกแยกผลลัพธ์ออกเป็น 2 ส่วน คือ ผลลัพธ์ที่ได้จากโหนดนั้นๆ และผลลัพธ์ที่จะถูกนำไปเป็นข้อมูล Input ของโหนดถัดไป โดย RNN ได้พัฒนามาจาก ANN ซึ่งไม่มีการนำ Output ของโหนดก่อนหน้ามาคำนวณหาผลลัพธ์ เทคนิค RNN นั้นเหมาะนำมาใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) หรือข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลอนุกรมเวลา (Time Series), ข้อมูลเสียง, ข้อมูลประเภทข้อความ, ข้อมูลภาพและวิดิทัศน์ เป็นต้น(Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017)



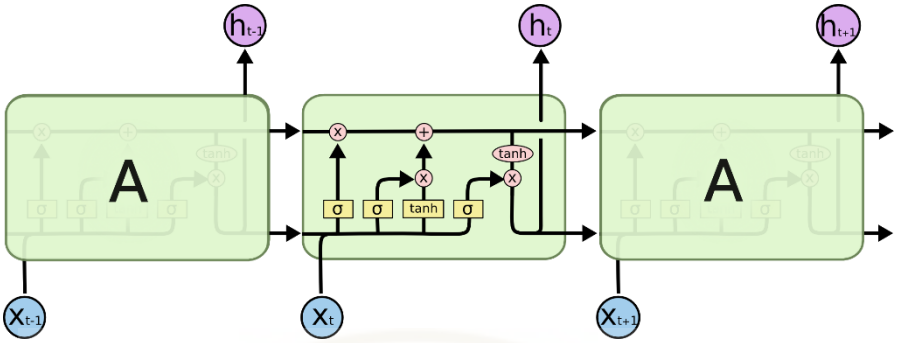
รูปภาพ 3.4 แสดงโครงสร้างการทำงานของ RNN

ที่มา: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

ข้อดีของ RNN คือ สามารถนำข้อมูลก่อนหน้า(ในอดีต) มาใช้ในการทำนายสิ่งที่อาจจะเกิดขึ้นในอนาคตได้ ส่วนข้อเสียของ RNN คือ จะสามารถดูข้อมูลย้อนหลังได้แค่เพียงระยะสั้นๆ เท่านั้น ซึ่งทำให้เกิดปัญหาในการทำ Backpropagation หรือการคำนวณหาค่าความผิดพลาดย้อนหลังของแต่ละโหนดเมื่อสิ้นสุดการทำงาน เพราะการทำ Backpropagation นั้นจะต้องทำย้อนไปหลายขั้นตอนและหลายโหนดจึงทำให้เกิดปัญหา Vanishing Gradient Problem ดังนั้นเพื่อแก้ปัญหาดังกล่างจึงทำให้เกิดเทคนิค LSTM ขึ้น

Long short-term memory (LSTM) เป็นโครงข่ายประเภท RNN รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหลักการทำงานคือ สามารถเก็บ ‘สถานะ’ หรือข้อมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับไปดูจะได้ทราบถึงที่มาของข้อมูลค่าดังกล่าวว่าเดิมเป็นค่าอะไร และจุดเด่นของแบบจำลอง LSTM คือฟังก์ชันพิเศษที่มีหน้าที่เสมือน

ประตู(Gate) ที่คอยควบคุมข้อมูลที่จะเข้าในแต่ละโหนดซึ่งประกอบด้วย Forget gate layer, Input gate layer และ Output gate layer (Jozefowicz, Zaremba, & Sutskever, 2015)

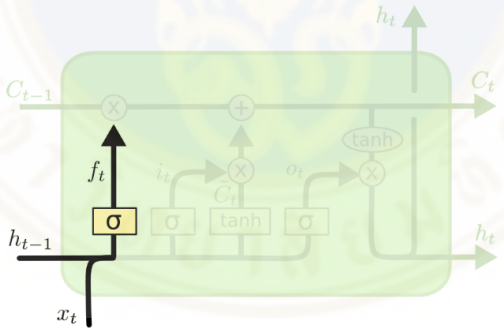


รูปภาพ 3.5 แสดงโครงสร้างการทำงานของ LSTM

ที่มา: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

3.4.1 Forget gate layer

เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน Cell นั้นควรจะถูกเก็บไว้หรือควรจะทิ้งไป ซึ่งข้อมูลที่ถูกตัดสินว่าควรเก็บไว้นั้นจะถูกประเมินจากข้อมูล Input ที่เข้ามาในโหนดนั้นๆ รวมกับผลลัพธ์ที่จะได้จากการคำนวณของโหนดก่อนหน้า ผ่านฟังก์ชัน ReLU ดังสมาการที่ 3.15



รูปภาพ 3.6 แสดงโครงสร้าง Forget gate layer

ที่มา: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

สมการ 3.15 Forget gate layer

$$f_t = ReLU\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

จากสมาการ

f_t คือ $Forget\ gate$

$ReLU$ คือ $A(x) = max(0, x)$

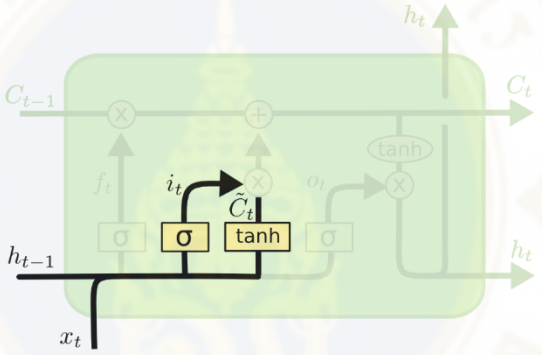
W_f คือ ค่าน้ำหนักของ $Matrics$

h_{t-1} คือ ค่า $Output$ ของ $Cell\ state$ ก่อนหน้า (ที่ $Timestamp\ t-1$)
x_t คือ ค่า $Input$ ที่เข้ามาใน $Cell\ state$ ณ เวลา t และ b_f คือ ค่า Bias

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

3.4.2 Input gate layer

เป็น Gate ที่มีหน้าที่รับข้อมูล Input เข้ามาใหม่แล้วจึงทำการบันทึก หรือ เขียน (Write) ข้อมูลลงไปในแต่ละโหนดโดยมีการทำงานแบ่งออกเป็น 2 ส่วน โดยส่วนแรกคือ ถ้าต้องการ Update cell state เมื่อทำการรับข้อมูล Input เข้ามาแล้วฟังก์ชันที่เป็นตัวควบคุมจะเรียกใช้ Input gate เพื่อเลือกว่าจะให้ Update cell state หรือไม่ต้อง และในส่วนที่สองถ้า Input gate เลือกที่ทำการ Update cell state ฟังก์ชัน tanh ก็จะทำการสร้าง Candidate values ขึ้นมาใน State ดังสมการ 3.16



และ 3.17

รูปภาพ 3.7 แสดงโครงสร้าง Input gate layer

ที่มา: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

สมการ 3.16 Input gate layer

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

สมการ 3.17 Candidate values

$$C_t = Tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

จากสมการ

i_t คือ $Input\ gate$
$ReLU$ คือ $A(x) = max(0, x)$
C_t คือ ค่า candidate ของ Cell state ที่เวลา t
$Tanh$ คือ $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$
W_i, W_c คือ ค่าน้ำหนักของ $Matrics$

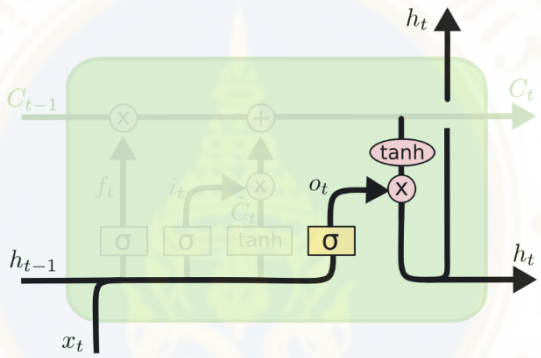
h_{t-1} คือ ค่า $Output$ ของ $Cell\ state$ ก่อนหน้า (ที่ $Timestamp\ t-1$)

x_t คือ ค่า $Input$ ที่เข้ามาใน $Cell\ state$ ณ เวลา t

b_i, b_c คือ ค่า Bias

3.4.3 Output gate layer

เป็น Gate ที่มีหน้าที่เตรียมทำการส่งออกข้อมูล (Output data) โดยข้อมูลที่จะทำการ Output นั้นจะดูจาก Cell state ที่ผ่านกระบวนการคำนวณต่างๆแล้ว โดยฟังก์ชัน ReLU จะเป็นตัวเลือกว่าข้อมูลส่วนไหน Cell state ที่จะถูก Output จากนั้นก็จะนำค่า Cell state เข้าฟังก์ชัน tanh (เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1) แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า Output ที่ได้จาก ReLU gate จากนั้นจะได้ค่า Output ที่ต้องการดังสมาการที่ 3.18 และ 3.19



รูปภาพ 3.8 แสดงโครงสร้าง Output gate layer

ที่มา: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

สมการ 3.18 Output gate layer

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

สมการ 3.19 Output value

$$h_t = o_t \times Tanh(C_t)$$

จากสมาการ

o_t คือ $Output\ gate$

$ReLU$ คือ $A(x) = max(0, x)$

C_t คือ ค่า candidate ของ Cell state ที่เวลา t

$Tanh$ คือ $\tanh(x) = \frac{2}{1+e^{-2x}} - 1$

W_o คือ ค่าน้ำหนักของ $Matrics$

h_{t-1} คือ ค่า $Output$ ของ $Cell\ state$ ก่อนหน้า (ที่ $Timestamp\ t-1$)
x_t คือ ค่า $Input$ ที่เข้ามาใน $Cell\ state$ ณ เวลา t
b_o คือ ค่า Bias
h_t คือ ค่า $Output$

3.5 วิธีการ (Methodology)

งานวิจัยนี้ภาพรวมจะถูกแบ่งออกเป็นสองส่วนย่อยคือ การพยากรณ์ความถี่รายวันของผลตอบแทนรายวัน, รายเดือน และราย 3 เดือนของตราสารทุน แต่ละตัวทั้งหมด 61 แบบจำลองโดยใช้ Machine Learning 2 อัลกอริทึม (ANN และLSTM) ซึ่งการที่ออกแบบด้วยการแยกเป็น 61 แบบจำลองตามจำนวนของหุ้นที่ใช้ในงานวิจัยนี้ เนื่องจากแต่ละตราสารทุนมีปัจจัยที่มีผลต่อการคาดการณ์ผลตอบแทนที่แตกต่างกัน หลังจากนั้นจึงนำผลตอบแทนที่พยากรณ์ได้มาเปรียบเทียบความแม่นยำกันระหว่างแบบจำลองทั้ง 2 อัลกอริทึม ว่าแบบจำลองใดเหมาะสมกับการพยากรณ์ตราสารทุนหุ้นสามัญในตลาดหลักทรัพย์แห่งประเทศไทย

โดยการทำงานแต่ละรอบของการสร้างแบบจำลอง จะเริ่มจากการใช้ข้อมูล 380 วันแรกก่อนในการให้แบบจำลองเรียนรู้และทำการทดสอบแบบจำลองในอีก 100 วันถัดไป (Train set 380 days + Validate set 100 days) เช่น ปัจจุบันอยู่ที่สัปดาห์ ณ วันที่ 5-Jan-2016 จะใช้ข้อมูลย้อนหลัง 380 days (ตั้งแต่สัปดาห์ ณ วันที่ 5-Jan-2009 จนถึง ณ วันที่ 29-Jul-2010) สำหรับให้แบบจำลองเรียนรู้ และจะใช้ข้อมูลย้อนหลัง 100 วันถัดไป (ตั้งแต่สัปดาห์ ณ วันที่ 30-Jul-2010 จน ณ ถึงวันที่ 23-Dec-2010) เพื่อใช้ในการทดสอบแบบจำลองโดยใช้ค่า Cost Function คือ MAE ซึ่งกล่าวในบทก่อนหน้าในการทดสอบว่าแบบจำลองทำงานได้มีประสิทธิภาพในการคาดการณ์ผลตอบแทนได้แม่นยำหรือไม่ หากไม่แม่นยำจะทำการกลับไปปรับแบบจำลองตั้งแต่ขั้นตอนการออกแบบให้เหมาะสม เช่น เพิ่มเส้นโครงข่ายประสาทเทียม, ปรับ Activation Functionให้เหมาะสมกับข้อมูล, เพิ่มจำนวนโหนดในแต่ละ Layer ของโครงข่ายประสาทเทียม เป็นต้น และทำตามขึ้นตอนข้างต้นวนไปทั้งหมด 127 รอบโดยการขยับช่วงข้อมูลไปข้างหน้ารอบละ 20 วัน ซึ่งยังคงโครงสร้างการใช้ข้อมูล 480 สัปดาห์ก่อนหน้านี้ในการให้แบบจำลองเรียนรู้และทำการทดสอบแบบจำลอง (Train set 380 days + Validate set 100 days) ตามรูปภาพ 4.2

เมื่อการพยากรณ์ผลตอบแทนของตราสารทุนแต่ละตัวแล้วจะนำผลที่ได้ไปวัดความแม่นยำของแต่ละแบบจำลอง โดยใช้ RMSE, MAE และ MSE ในการเปรียบเทียบกันว่าแบบจำลองได้มีค่าความแม่นยำสูงที่สุดและวัดผลกับ Benchmark ซึ่งคำนวณจากค่าเฉลี่ยของผลตอบแทนก่อน

หน้า (ทั้งระยะสั้น 5 วัน และระยะยาว 20 วัน (นับตาม Daytrade) และเปรียบเทียบการพยากรณ์ด้วยว่าช่วงการพยากรณ์ใด (1 วัน, 1 เดือน และ 3 เดือน) มีความแม่นยำที่สุด

3.6 Prediction of Stock Return

งานวิจัยนี้พัฒนาแบบจำลอง Neural Network ด้วยโปรแกรมภาษา Python โดยทำงานร่วมกับ Scikit-learn ซึ่งเป็น Machine Learning Library และใช้งานร่วมกับ Keras ซึ่งเป็น High-level Neuron Network API สำหรับการออกแบบ แบบจำลอง ANN และLSTM หลังจากผ่านการปรับแต่งแบบจำลองจากผลการทำ Model Validation ด้วยชุดข้อมูล Validate Set ของทุกแบบจำลองและประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้ในงานวิจัยนี้จึงใช้โครงสร้าง 305:305:3 คือมี Input Layer จำนวน 305 Node ร่วมกับ Hidden Layer 305 Node โดยทั้ง Input layer และ Hidden layer ใช้ ReLU Activation Function เนื่องจากข้อมูลตัวแปรต้นทุกตัวถูกปรับให้ข้อมูลอยู่ในรูปแบบอย่างง่ายหรือทำ Data Normalization มาแล้วทำให้ข้อมูลอยู่ในช่วง 0-1 จึงเหมาะสมในการใช้ ReLU Activation Functionสำหรับส่วน Output Layer จำนวน 3 Node นั้นใช้ Tanh Activation Function ช่วยให้ผลคาดการณ์สอดคล้องกับค่าตัวแปรตามที่ต้องการซึ่งอยู่ในช่วง -1 ถึง 1 เนื่องจากตัวแปรตามหรือผลตอบแทนรายวันของตราสารทุนนั้นมีทั้งค่าบวกและลบ สำหรับการประมวลผลใช้อัลกอลิทึมเพิ่มประสิทธิภาพ ADAM Optimizer ร่วมกับ Cost Function คือ Mean Absolute Error (MAE) โดยประมวลผลที่ 30 epochs ต่อ 1 รอบของการทำการปรับแบบจำลอง (Model calibration)

การพัฒนาแบบจำลองของตราสารทุนแต่ละตราสารมีทั้งหมด 61 แบบจำลองตามจำนวนตราสารทุนที่ใช้ในการคาดการณ์ผลตอบแทนในแต่ละอัลกอริทึม ตามความแตกต่างของปัจจัยมีผลในการคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว อีกทั้งต้องการให้แบบจำลองสามารถทำงานได้อิสระเกิดความเฉพาะในการคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว โดยการทำงานในแบบจำลองแต่ละตัวจะคาดการณ์ผลตอบแทนรายวัน รายเดือน และราย 3 เดือน ของตราสารทุน ซึ่งการคาดการณ์ผลตอบแทนนี้จะทำรายวัน กล่าวคือมีข้อมูลปัจจัยที่มีผลหรือตัวแปรต้นเป็นรายวันย้อนหลัง และป้อนผลตอบแทนรายวัน รายเดือน และราย 3 เดือน ณ วันนั้นๆให้แบบจำลองเรียนรู้ และนำข้อมูลปัจจัยที่มีผลหรือตัวแปรต้น ณ วันที่จะทำการคาดการณ์ ป้อนใส่แบบจำลองเพื่อให้คาดการณ์ผลตอบแทนในแต่ละแบบของวันนั้นออกมา ซึ่งทำพยากรณ์ผลตอบแทนต่อเนื่องทั้งหมด 127 รอบ เพื่อให้แบบจำลองเกิดความแม่นยำมากที่สุด

บทที่ 4
ข้อมูลและตัวแปรที่ใช้ในงานวิจัย (Data and Factors)

4.1 Stock selection

ข้อมูลที่ใช้ในการคาดการณ์ผลตอบแทนตราสารทุนนั้น เลือกใช้ข้อมูล ตราสารทุนที่เป็นองค์ประกอบของดัชนี SET100 จากตลาดหลักทรัพย์แห่งประเทศไทย (Stock Exchange of Thailand) ณ เดือนมกราคม 2009 จำนวน 61 หุ้น รายชื่อตาม ภาคผนวก ก. หลังจากนั้นเพื่อแก้ปัญหาความไม่สมบูรณ์ของข้อมูลและเพื่อสร้างความแม่นยำของแบบจำลองจึงต้องคัดกรองตราสารทุน โดยมีวิธีคัดกรองตราสารทุน คือ ตราสารทุนที่จะนำเข้าไปในแบบจำลองต้องมีข้อมูลเพียงพอในช่วงที่จัดทำแบบจำลอง คือ 05/01/2009 – 22/04/2021 หรือเทียบเท่ากับระยะเวลา 12 ปี 4 เดือนและไม่อยู่ในกลุ่ม Banking Sector เนื่องจากโครงสร้างของงบการเงินทั้งสอง Sector ข้างต้นมีความแตกต่างจาก Sector อื่นๆ ซึ่งไม่เหมาะสมต่อการนำมารวมกันพิจารณาในงานวิจัยนี้

ขอบเขตงานวิจัยนี้เป็นการใช้ข้อมูลตราสารทุนในตลาด SET100 ในช่วงปี 2009 – 2019 ในการเรียนรู้แบบจำลองเพื่อคาดการณ์ผลตอบแทนหุ้น 2020-2021 ซึ่งไม่ครอบคลุมแบบจำลองที่ใช้ในตราสารทุนกลุ่มธนาคาร (Banking Sector) ซึ่งมีข้อมูลปัจจัยและโครงสร้างงบการเงินที่แตกต่างจากธุรกิจในกลุ่มอื่นๆ และข้อจำกัดอื่นๆในงานวิจัยนี้จำแนกเป็นหัวข้อได้ดังนี้

1. ผลตอบแทนในงานวิจัยนี้มิได้คำนึงถึงค่าธรรมเนียมการซื้อขายตราสารทุน
2. รายชื่อตราสารทุนจะเป็น 61 ตัวเดิมตามข้อมูล ณ วันที่ 05/01/2009
3. งานวิจัยนี้ไม่ครอบคลุมตราสารทุนในตลาดอื่นๆนอก SET100

4.2 ตัวแปร (Factors)

การคัดเลือกตัวแปรต้นนั้นแบ่งชนิดของตัวแปรต้นที่จะนำเข้าไปคำนวณในแบบจำลองได้ 9 ประเภท ประกอบด้วย

1) ตัวแปรจากข้อมูลตราสารทุนในตลาด (Stock Trade)
2) ตัวแปรจากงบการเงิน (Financial Statement)
3) ตัวแปรทางเศรษฐศาสตร์มหภาค (Macro Economic)
4) ตัวแปรทางเทคนิคในการซื้อขายหุ้น (Technical Indicator)

5) ตัวแปรจากอัตราส่วนทางการเงิน (Financial ratio)

6) ตัวแปรจากอัตราแลกเปลี่ยนเงินตรา (Exchange rate)

7) ตัวแปรจากดัชนีหุ้น (Stock Index)

8) ตัวแปรจากดัชนีทองคำ (Gold index)

9) ตัวแปรจากตัวเลขผู้ติดเชื้อไวรัสโคโรนา (COVID-19)

โดยมีตัวแปรต้นทั้งหมด 34 ตัวแปรตามรายชื่อในภาคผนวก ข และเพื่อแก้ปัญหาความไม่สมบูรณ์ของข้อมูลและเพื่อสร้างความแม่นยำของแบบจำลองจึงต้องมีการปรับตัวแปรโดยมีวิธีดังนี้

4.2.1 การจัดเรียงข้อมูลใหม่

ซึ่งข้อมูลที่นำเข้าแบบจำลองจะต้องมีความถี่เป็นรายวันเท่านั้น ดังนั้นข้อมูลที่มีความถี่น้อยกว่ารายวัน เช่น ข้อมูลเศรษฐศาสตร์มหภาค, ข้อมูลจากงบการเงิน จะเป็นข้อมูลที่มีความถี่รายปี, รายไตรมาส และรายเดือน ซึ่งจะต้องปรับให้เข้าสู่ความถี่รายวันโดยการเลือกข้อมูลที่ความถี่น้อยกว่าเหล่านั้น กระจายเข้าสู่ข้อมูลที่มีความถี่รายวันโดยเลือกตัวแทนของช่วงความถี่ที่น้อยกว่า ที่เกิดขึ้นก่อนวันนั้นๆของข้อมูล ตามตัวอย่างในภาพที่ 4.1

Daily factor	
DATE	PRICE
15/8/2015	190
16/8/2015	190.5
15/11/2015	189
16/11/2015	187

+

Quartery factor	
QUARTER/YEAR	GDP
2/2015	116.3
3/2015	112.7

+

Monthly factor	
MONTH/YEAR	CPI
7/2015	76.4
10/2015	66.3

→

DATE	PRICE	GDP	CPI
15/8/2015	190	116.3	76.4
16/8/2015	190.5	116.3	76.4
15/11/2015	189	112.7	66.3
16/11/2015	187	112.7	66.3

รูปภาพ 4.1 แสดงการกระจายข้อมูลที่มีความถี่น้อยกว่า เข้าสู่ข้อมูลที่มีความถี่รายวัน

4.3 Return Calculation

การคำนวณผลตอบแทนของตราสารทุนจะแบ่งออกเป็น 3 แบบ คือ รายวัน รายเดือน และราย 3 เดือน ซึ่งจะมาจากส่วนต่างของราคาตราสารทุน ณ ปัจจุบัน กับราคาตราสารทุน ณ ช่วงเวลาก่อนหน้า 1 ช่วงเวลาในอดีต (1 วัน ,1 เดือน และ3 เดือน) ตามสมการที่ 4.1, 4.2 และ 4.3

สมการ 4.1 แสดงการคำนวณผลตอบแทนรายวันของตราสารทุนในข้อมูลที่มีความถี่เป็นวัน

$$\boldsymbol{r_t = \frac{p_t}{p_{t-1}} - 1}$$

สมการ 4.2 แสดงการคำนวณผลตอบแทนรายเดือนของตราสารทุนในข้อมูลที่มีความถี่เป็นวัน

$$\boldsymbol{r_{ij} = \frac{p_{ij}}{p_{i,j-1}} - 1}$$

สมการ 4.3 แสดงการคำนวณผลตอบแทนราย 3 เดือน ของตราสารทุนในข้อมูลที่มีความถี่เป็นวัน

$$\boldsymbol{y_{ij} = \frac{p_{ij}}{p_{i,j-3}} - 1}$$

p_t คือ ราคาของหุ้นรายวัน ณ ช่วงเวลาที่ t (ความถี่เป็นรายวัน)

p_{ij} คือ ราคาเดือนของหุ้น ณ วันที่ i เดือนที่ j

r_t คือ ผลตอบแทนรายวันของหุ้น ณ ช่วงเวลาที่ t

r_{ij} คือ ผลตอบแทนรายเดือนของหุ้น ณ วันที่ i เดือนที่ j

y_{ij} คือ ผลตอบแทนราย 3 เดือน ของหุ้น ณ วันที่ i เดือนที่ j

t คือ ช่วงเวลาที่ t ของ Time series (ความถี่เป็นรายวัน)

i คือ วันที่ i ของ Time series

j คือ เดือนที่ j ของ Time series

4.4 การแบ่งข้อมูล

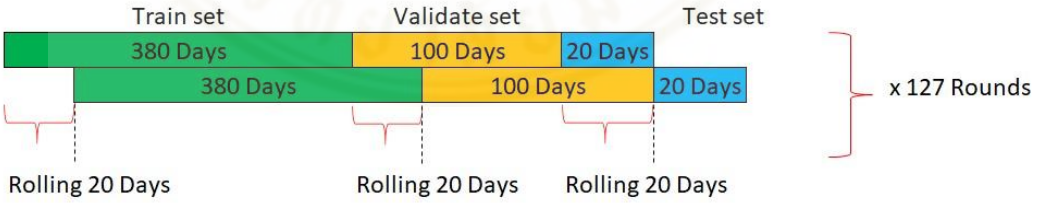
การนำเข้าข้อมูลไปยังแบบจำลองจะแบ่งชุดข้อมูลออกเป็น 3 ชุด ตามภาพที่ 4.2 ดังนี้

1. ชุดพัฒนาแบบจำลอง (Train Set) โดยเป็นชุดข้อมูลที่ใช้เพื่อให้แบบจำลองเรียนรู้

2. ชุดทดสอบแบบจำลอง (Validate Set) ใช้สำหรับทดสอบแบบจำลองที่ระบบได้จัดทำจากข้อมูลชุดที่ 1 เพื่อทดสอบปัญหาเช่น การ Overfitting และ Underfitting ของแบบจำลอง

โดย Overfitting คือ การที่แบบจำลองถูกรบกวนด้วยตัวแปรหลายๆตัว ซึ่งทำให้แบบจำลองได้ผลดีในข้อมูลชุด Train แต่กลับให้ผลที่แย่ในข้อมูลชุดอื่นๆ จึงส่งผลให้แบบจำลองมีความคาดเคลื่อนมาก Underfitting คือ การที่แบบจำลองมีตัวแปรต้นที่ส่งผลในการอธิบายตัวแปรตามมีจำนวนน้อยเกินไปทำให้แบบจำลองมีความสามารถในการพยากรณ์ข้อมูลได้แม่นยำน้อย ซึ่งหากผลการทดสอบแบบจำลองได้ผลไม่ดี กล่าวคือค่าจาก Cost Function มีค่าสูง จะทำการกลับไปปรับแก้ไขโครงสร้างของแบบจำลองโครงข่ายประสาทเทียมให้เหมาะสมและทำการรันทดสอบจนกว่าจะได้ Cost Function ที่มีค่าต่ำที่สุด จึงจะนำแบบจำลองดังกล่าวไปใช้ในการคาดการณ์ผลตอบแทนของตราสารทุนในลำดับต่อไป

3. ชุดคาดการณ์แบบจำลอง (Test Set) 20 วัน ก่อน Rolling ออกรอบละ 20 วัน และเพิ่มเข้าใหม่รอบละ 20 วัน เป็นเหมือนการอัพเดทชุดข้อมูลใหม่ทุกๆ 20 วัน เพื่อใช้สำหรับคาดการณ์ผลตอบแทนรายวัน, รายเดือน และราย 3 เดือน ของตราสารทุนไปพร้อมๆกัน โดยแบบจำลองที่ผ่านการจัดทำจากข้อมูลชุดที่ 1 และทดสอบโดยข้อมูลชุดที่ 2 แล้ว ซึ่งชุดข้อมูลทั้งแบบรายวัน, รายเดือน และรายไตรมาสนั้นจะมีความถี่เป็น 1 วัน เท่ากันทั้งหมด เช่น ผลตอบแทนรายเดือน ณ วันที่ 1 มกราคม คือ ผลตอบแทนหากซื้อหลักทรัพย์ ณ วันที่ 1 มกราคมแล้วไปขายในอีก 1 เดือนข้างหน้า (ขาย ณ วันที่ 1 กุมภาพันธ์) หรือ ผลตอบแทนรายเดือน ณ วันที่ 3 มกราคม คือ ผลตอบแทนหากซื้อหลักทรัพย์ ณ วันที่ 1 มกราคมแล้วไปขายในอีก 1 เดือนข้างหน้า (ขาย ณ วันที่ 3 กุมภาพันธ์) จึงทำให้การคาดการณ์ผลตอบแทนรายวัน, รายเดือน และรายไตรมาส ผ่าน Machine learning ของตราสารทุนไปพร้อมๆกันได้ เนื่องจากข้อมูลในอดีตมีความถี่เป็น 1 วันเท่ากัน โดยมีการแสดงการแบ่งช่วงการเรียนรู้ ตามรูปภาพ 4.2



รูปภาพ 4.2 แสดงการแบ่งช่วงข้อมูลสำหรับการเรียนรู้, การทดสอบและการคาดการณ์ผลตอบแทนของตราสารทุนของแบบจำลองแต่ละตัว

4.5 ผลการวิจัย (Empirical Results)

งานวิจัยนี้นำเสนอการคาดการณ์ผลตอบแทนตราสารทุนโดยใช้ Machine Learning 2 อัลกอริทึม(ANN และLSTM) ที่เรียนรู้จากข้อมูลและปัจจัยที่มีผลกระทบต่อราคาตราสารทุน ผนวกกับการนำผลตอบแทนที่คาดการณ์วัดผลหาค่าความผิดพลาดเพื่อเปรียบเทียบว่าแบบจำลองได้ให้ค่าความผิดพลาดน้อยที่สุดหรือก็คือมีความแม่นยำมากที่สุด

โดยผลจากการวิจัยพบว่าในการพยากรณ์แบบรายวันในทุกแบบจำลองนั้นให้ผลที่แม่นยำน้อยกว่าค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้าทั้งระยะสั้น 5 วันและระยะยาว 20 วัน ตามลำดับ โดยแบบจำลอง ANN มีค่า RMSE ที่ 2.57% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 0.17%, มากกว่าค่าเฉลี่ยระยะยาว 0.15% แต่น้อยกว่าแบบจำลอง LSTM 0.23% และมีค่า Hitrate ที่ 54.14% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 4.54%, มากกว่าค่าเฉลี่ยระยะยาว 4.04% แต่น้อยกว่าแบบจำลอง LSTM 1.3% แบบจำลอง LSTM มีค่า RMSE ที่ 2.8% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 0.4%, มากกว่าค่าเฉลี่ยระยะยาว 0.38%, มากกว่าแบบจำลอง ANN 0.23% และมีค่า Hitrate ที่ 55.44% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 5.84%, มากกว่าค่าเฉลี่ยระยะยาว 5.34% และมากกว่าแบบจำลอง ANN 1.3%

การพยากรณ์แบบรายเดือนให้ผลดังนี้ โดยแบบจำลอง ANN มีค่า RMSE ที่ 7.07% ซึ่งน้อยกว่าค่าเฉลี่ยระยะสั้น 7.16%, น้อยกว่าค่าเฉลี่ยระยะยาว 6.95%, น้อยกว่าแบบจำลอง LSTM 0.18% และมีค่า Hitrate ที่ 78.43% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 28.58%, มากกว่าค่าเฉลี่ยระยะยาว 27.59%, มากกว่าแบบจำลอง LSTM 0.16% แบบจำลอง LSTM มีค่า RMSE ที่ 7.25% ซึ่งน้อยกว่าค่าเฉลี่ยระยะสั้น 6.98%, น้อยกว่าค่าเฉลี่ยระยะยาว 6.77%, มากกว่าแบบจำลอง ANN 0.18% และมีค่า Hitrate ที่ 78.27% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 28.42%, มากกว่าค่าเฉลี่ยระยะยาว 27.43%, น้อยกว่าแบบจำลอง ANN 0.16%

การพยากรณ์แบบรายไตรมาสให้ผลดังนี้ โดยแบบจำลอง ANN มีค่า RMSE ที่ 7.85% ซึ่งน้อยกว่าค่าเฉลี่ยระยะสั้น 18.81%, น้อยกว่าค่าเฉลี่ยระยะยาว 18.95%, น้อยกว่าแบบจำลอง LSTM 0.06% และมีค่า Hitrate ที่ 87.85% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 36.22%, มากกว่าค่าเฉลี่ยระยะยาว 36.42%, มากกว่าแบบจำลอง LSTM 0.10% แบบจำลอง LSTM มีค่า RMSE ที่ 7.91% ซึ่งน้อยกว่าค่าเฉลี่ยระยะสั้น 18.75%, น้อยกว่าค่าเฉลี่ยระยะยาว 18.89%, มากกว่าแบบจำลอง ANN 0.06% และมีค่า Hitrate ที่ 87.75% ซึ่งมากกว่าค่าเฉลี่ยระยะสั้น 36.12%, มากกว่าค่าเฉลี่ยระยะยาว 36.32%, น้อยกว่าแบบจำลอง ANN 0.1%

ซึ่งทั้ง 2 แบบจำลองในการพยากรณ์ผลตอบแทนแบบรายวันนั้นแม่นยำน้อยกว่าค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้าทั้งระยะสั้น 5 วันและระยะยาว 20 วัน แต่ก็น้อยกว่าไม่มากกว่ากันนัก ในส่วนของความแม่นในการพยากรณ์ทิศทางของผลตอบแทนของหุ้นว่าผลตอบแทนจะเพิ่มขึ้น, ลดลง นั้นทั้ง 2 แบบจำลองให้การพยากรณ์ที่แม่นยำกว่าค่าเฉลี่ยของ

ผลตอบแทนช่วงเวลาก่อนหน้าทั้งระยะสั้น 5 วันและระยะยาว 20 วัน แต่ก็แม่นยำไม่มากกว่ากันมากนักเช่นกัน ในส่วนของการพยาการณ์ที่มีช่วงยาวขึ้น คือ รายเดือน และรายไตรมาสพบว่าแบบจำลองที่มีความแม่นยำที่สุดก็คือ Artificial neural network ซึ่งมีค่าความผิดพลาดต่ำที่สุดทั้งรายเดือนและรายไตรมาส หมายความว่าการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ของไทยแบบรายวันทั้ง 2 แบบจำลองอาจจะไม่เหมาะสมหรืออาจเกิดจากการคัดเลือกตัวแปรต้นที่ไม่ส่งผลต่อผลตอบแทนของหุ้นจึงทำให้ในการพยากรณ์รายวันได้ไม่ดี แต่ในทางกลับกันหากเป็นการพยากรณ์ในระยะยาว (รายเดือน รายไตรมาส) แบบจำลอง ANN เหมาะสมกับการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ไทยกว่าแบบจำลอง LSTM และหากสนใจในการคาดการณ์ทิศทางของผลตอบแทนของหุ้นว่าจะเพิ่มขึ้นหรือลดลงนั้นในการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ของไทยทั้ง 3 แบบ คือ แบบรายวัน, รายเดือน และรายไตรมาส แบบจำลองทั้ง 2 มีความเหมาะสมกับการทำนายทิศทางขึ้นลงของหุ้นในตลาดหลักทรัพย์ของไทย ซึ่งผลการวิจัยเป็นไปตามตาราง 4.1-4.3

สำหรับตาราง 4.1-4.3 เป็นการวัดผลแบบจำลองในขั้นตอนการคาดการณ์แบบจำลอง (Model Testing) โดยในขั้นตอนนี้มีค่าคลาดเคลื่อนที่คำนวนจาก RMSE และ MAE เพื่อเปรียบเทียบความแม่นยำในแต่ละแบบจำลองและในแต่ละช่วงผลตอบแทนที่พยากรณ์ โดยมี Benchmark เป็นค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้า 5 วัน และ 20 วัน

ตาราง 4.1 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 วัน (1 Days) สำหรับทุกแบบจำลอง

Model	RMSE	MAE	HitRate	HitRate+	HitRate-
ANN	2.57%	1.85%	54.14%	42.77%	44.99%
LSTM	2.8%	1.99%	55.44%	43.70%	46.20%
Mean 5 day	2.4%	1.66%	49.60%	39.52%	40.94%
Mean 20 day	2.42%	1.67%	50.10%	39.79%	41.39%

ตาราง 4.2 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 เดือน (1 Month) สำหรับทุกแบบจำลอง

Model	RMSE	MAE	HitRate	HitRate+	HitRate-
ANN	7.07%	4.94%	78.43%	76.36%	74.99%
LSTM	7.25%	5.02%	78.27%	76.49%	74.68%
Mean 5 day	14.23%	10.37%	49.85%	49.62%	45.59%
Mean 20 day	14.02%	10.25%	50.84%	50.58%	46.61%

ตาราง 4.3 แสดงผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 3 เดือน (3 Month) สำหรับทุกแบบจำลอง

Model	RMSE	MAE	HitRate	HitRate+	HitRate-
ANN	7.85%	5.3%	87.85%	89.42%	83.68%
LSTM	7.91%	5.34%	87.75%	88.93%	84.02%
Mean 5 day	26.66%	19.28%	51.63%	53.08%	46.30%
Mean 20 day	26.8%	19.33%	51.43%	52.85%	46.09%

บทที่ 5
บทสรุปงานวิจัย (Conclusion)

งานวิจัยนี้นำเสนอการคาดการณ์ผลตอบแทนตราสารทุนโดยใช้ Machine Learning 2 อัลกอริทึม(ANN และLSTM) ที่เรียนรู้จากข้อมูลและปัจจัยที่มีผลกระทบต่อราคาตราสารทุน ผนวกกับการนำผลตอบแทนที่คาดการณ์วัดผลหาค่าความผิดพลาดเพื่อเปรียบเทียบว่าแบบจำลองให้ค่าความผิดพลาดน้อยที่สุดหรือก็คือมีความแม่นยำมากที่สุด โดยผลจากการวิจัยพบว่าในการพยากรณ์แบบรายวันในทุกแบบจำลองนั้นให้ผลที่แม่นยำน้อยกว่าค่าเฉลี่ยของผลตอบแทนช่วงเวลาก่อนหน้าทั้ง 5 วัน และ 20 วันตามลำดับ แต่ในส่วนของการพยาการณ์ที่มีช่วงยาวขึ้น คือ รายเดือน และรายไตรมาสพบว่าแบบจำลองที่มีความแม่นยำที่สุดก็คือแบบจำลอง Artificial neural network ซึ่งมีค่าความผิดพลาดต่ำที่สุดทั้งรายเดือนและรายไตรมาส และหากวัดความแม่นยำในการคาดการณ์ทิศทางของผลตอบแทนของหุ้นว่าผลตอบแทนจะเพิ่มขึ้น, ลดลง หรือว่าไปในทิศทางใดนั้น ผลที่ได้จากการทดลองพบว่าในการพยากรณ์แบบรายวันแบบจำลองที่ให้ค่าความแม่นยำของทิศทางผลตอบแทนของหุ้นแม่นยำที่สุดคือ LSTM แต่ในส่วนของการพยาการณ์ที่มีช่วงยาวขึ้น คือ รายเดือน และรายไตรมาสพบว่าแบบจำลองที่มีความแม่นยำที่สุดก็คือแบบจำลอง Artificial neural network ซึ่งมีค่าความแม่นยำมากที่สุดทั้งรายเดือนและรายไตรมาส

การพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ของไทยแบบรายวันทั้ง 2 แบบจำลองอาจจะไม่เหมาะสมหรืออาจเกิดจากการคัดเลือกตัวแปรต้นที่ไม่ส่งผลต่อผลตอบแทนของหุ้นจึงทำให้ในการพยากรณ์รายวันได้ไม่ดี แต่ในทางกลับกันหากเป็นการพยากรณ์ในระยะยาว(รายเดือน รายไตรมาส) แบบจำลอง ANN เหมาะสมกับการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ไทยกว่าแบบจำลอง LSTM และหากสนใจในการคาดการณ์ทิศทางของผลตอบแทนของหุ้นว่าจะเพิ่มขึ้นหรือลดลงนั้นในการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ของไทยแบบรายวัน แบบจำลองให้พยากรณ์ทิศทางแม่นที่สุดคือ LSTM แต่ถ้าหากเป็นการพยากรณ์แบบรายเดือนและรายไตรมาสนั้นแบบจำลอง ANN จะคาดการณ์ทิศทางของผลตอบแทนหุ้นได้แม่นยำที่สุด

งานวิจัยนี้แสดงให้เห็นถึงการคาดการณ์ผลตอบแทนตราสารทุนผ่านการใช้ Machine Learning โดยเลือกเทคนิคแบบจำลอง 2 อัลกอริทึม คือ ANN และLSTM ซึ่งสามารถนำผลที่ได้ไปประยุกต์ใช้ประกอบการตัดสินใจในการเลือกหลักทรัพย์ที่จะลงทุนหรือกำหนดกลยุทธ์ให้เหมาะสมกับความต้องการของนักลงทุน รวมถึงการนำไปจัดการปรับพอร์ตโฟลิโอให้กับนักลงทุนสถาบัน

หรือนักลงทุนอื่นๆได้ โดยในปัจจุบันแนวโน้มในการใช้ข้อมูลเพื่อตัดสินใจทางธุรกิจ (Data Driven), เทคโนโลยีด้านฮาร์ดแวร์ที่มีศักยภาพการคำนวณที่สูงขึ้นในราคาที่ถูกลงกว่าในอดีต, ข้อมูลปัจจัยต่างๆที่มีผลต่อราคาตราสารทุนมีมากขึ้นทั้งในแง่ขนาด ความเร็วและความหลากหลายข้อมูล รวมถึงแนวโน้มในการวิจัยพัฒนาเทคนิคด้าน Machine Learning และ Deep Learning ที่ได้รับการพัฒนาอย่างรวดเร็วส่งผลให้มีการพัฒนาเทคนิควิธีการใหม่ๆซึ่งช่วยให้งานวิจัยนี้สามารถที่จะนำไปต่อยอดและขยายผลให้เกิดประโยชน์ได้อย่างกว้างขวางเพิ่มศักยภาพความแม่นยำในการคาดการณ์ผลตอบแทนของตราสารทุนได้เป็นอย่างดี เป็นตัวเลือกเพื่อสร้างโอกาสในการลงทุนของนักลงทุนต่างๆได้อย่างมีประสิทธิภาพและมีประสิทธิผล

งานวิจัยนี้ยังสามารถพัฒนาต่อยอดด้วยการนำผลที่ได้ไปจัดทำกลยุทธ์ในการจัดพอร์ตโฟลิโอในรูปแบบต่าง ๆ อาทิเช่น Bayes-Stein shrinkage, Black-Litterman, เป็นต้น สามารถต่อยอดโดยการเพิ่มปัจจัยหรือตัวแปรที่มีผลต่อการพัฒนาแบบจำลองเพื่อให้มีความแม่นยำในการคาดการณ์ผลตอบแทนมากขึ้น หรือนำแบบจำลองไปพัฒนาโดยใช้เทคนิคที่มีระดับสูงขึ้นอย่างเช่น Deep Learning

บรรณานุกรม

Alaloul, W. S., & Qureshi, A. H. (2020). *Data Processing Using Artificial Neural Networks* (D. G. Harkut Ed.).

Baker, M. R., & Patil, R. B. (1998). Universal Approximation Theorem for Interval Neural Networks. *Reliable Computing, 4*, 235–239. doi:https://doi-org.ejournal.mahidol.ac.th/10.1023/A:1009951412412

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *North American Journal of Economics and Finance, 47*(47), 552–567. doi:https://doi.org/10.1016/j.najef.2018.06.013

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32. doi:https://doi.org/10.1023/A:1010933404324

Floydhub, & Keras. (2017). SGD&Adam Which One Is The Best Optimizer. Retrieved from https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/?source=post_page-----9feb5a87e3b2----------

Han, S. (2019). *Stock Prediction with Random Forests and Long Short-term Memory.* (Master Degree). Iowa State University,

Jaroenkitwatcharachai, K. (2018). *Artificial intelligence for forecasting wage.* (Master degree Individual Study). Thammasat University,

Jiemwiriyakul, B., Sirianuntapiboon, P., & Lorsubkong, P. (2019). *Portfolio Return Prediction using Neural Network.* (Master degree Individual Study). Mahidol University

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). *An Empirical Exploration of Recurrent Network Architectures*. Paper presented at the Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research. https://proceedings.mlr.press/v37/jozefowicz15.html

Kanoktipsatharporn, S. (2019a). What is ReLU function why popular deep learning training deep neural network activation function ep.3. Retrieved from https://www.bualabs.com/archives/1355/what-is-relu-function-why-popular-deep-learning-training-deep-neural-network-activation-function-ep-3/

บรรณานุกรม (ต่อ)

Kanoktipsatharporn, S. (2019b). What is Tanh activation function compare different sigmoid function artificial neural network activation function ep.2. Retrieved from https://www.bualabs.com/archives/1324/what-is-tanh-activation-function-compare-different-sigmoid-function-artificial-neural-network-activation-function-ep-2/

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1), 307–319. doi:https://doi.org/10.1016/S0925-2312(03)00372-2

Loh, W. Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery, 1*(1), 14-23. doi:https://doi.org/10.1002/widm.8

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications, 165*, 1-15. doi:https://doi.org/10.1016/j.eswa.2020.113973

Nakwijit, P. (2020). ทำความเข้าใจ Optimizer. Retrieved from https://medium.com/@chameleontk/%E0%B8%97%E0%B8%B3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B9%89%E0%B8%B2%E0%B9%83%E0%B8%88-optimizer-a44455615c32

Nilson, M. (2019). A visual proof that neural nets can compute any function. Retrieved from http://neuralnetworksanddeeplearning.com/chap4.html

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*(4), 2162-2172. doi:https://doi.org/10.1016/j.eswa.2014.10.031

Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222-2232. doi:https://doi.org/10.1109/TNNLS.2016.2582924