การคาดการณ์ผลตอบแทนในอนาคตของตราสารทุนหุ้นสามัญโดยการใช้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง

Predicting Stock Return Using Machine Learning

วิศรุต แก้วมหา[1] และวริศ ปัญญาฉัตรพร[2]

Witsarut Kaewmaha[1] and Varis Punyachatporn[2]

บริษัท เมืองไทยประกันภัย จำกัด (มหาชน)[1], วิทยาลัยการจัดการ มหาวิทยาลัยมหิดล[2]

Muang Thai Insurance Public Company Limited[1], College of Management Mahidol University[2]

E-mail: witsarut.kmh@gmail.com[1], varis116@hotmail.com[2]

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อประเมินความสามารถพยากรณ์ของแบบจำลองคอมพิวเตอร์เรียนรู้ได้ด้วยตอนเองต่อผลตอบแทนของตราสารทุน โดยใช้ข้อมูลราคารายวันของตราสารทุนในตลาดหลักทรัพย์แห่งประเทศไทย ข้อมูลในงบการเงิน ข้อมูลอัตราส่วนทางการเงิน ข้อมูลปัจจัยทางเทคนิคข้อมูล เศรษฐศาสตร์มหภาค อัตราแลกเปลี่ยน ดัชนีหลักทรัพย์ และดัชนีทองคำ ข้อมูลช่วง 2009 -2021 ผลการศึกษาแสดงการพยากรณ์ผลตอบแทนด้วยแบบจำลองคอมพิวเตอร์เรียนรู้ได้ด้วยตอนเองคาดการณ์ผลตอบแทนด้วยแบบจำลองที่มีความแม่นยำที่สุดก็คือแบบจำลอง Random Forest ซึ่งมีค่าความผิดพลาดต่ำที่สุดในทุกช่วงของการคาดการ์ผลตอบแทน (ราย 1 วัน 1 เดือน และ 3 เดือน)

คำสำคัญ: ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตอนเอง ตราสารทุน ผลตอบแทน

Abstract

The objective of this research was to determine the predictive capability of machine learning models on the returns of equity instruments. This research used daily price information of equity securities in the Stock Exchange of Thailand, financial statements data, financial ratio data, technical analysis data, macroeconomic data, exchange rate, stock index, and gold indices during 2009 to 2021. The results show that the most accurate model estimation of a machine learning model is the Random Forest model, which has the lowest deviation across all ranges of estimation window forecasts (1-day, 1-month, and 3-month).

Keywords: Machine Learning, stock, returns

บทนำ

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), Random Forest(RF) และLong Short-Term Memory(LSTM) โดยเป็นการใช้ข้อมูลปัจจัยที่มีผลต่อผลตอบแทนของหุ้น เป็นข้อมูลเพื่อนำเข้าในระบบ Machine Learning ที่มีการออกแบบให้เหมาะสมกับแบบจำลองเรียนรู้

ดังนั้นเมื่อสามารถสร้างแบบจำลองสำหรับการคาดการณ์ผลตอบแทนในอนาคตได้อย่างแม่นยำ หากอัลกอริทึมใดเหมาะสมกับการพยากรณ์ผลตอบแทนของหุ้นได้แม่นยำที่สุดในช่วงเวลาต่างๆที่กำหนด (1 วัน 1 เดือน และ3 เดือน) เพื่อนำโมเดลที่ได้มาใช้คาดการผลตอบแทนของหุ้นเพื่อเป็นแนวทางในการคัดเลือกหลักทรัพย์และกำหนดกลยุทธ์ในการลงทุนที่เหมาะสมเพื่อให้ได้ผลตอบแทนที่เหมาะสมกับความคาดหวังของนักลงทุน

งานวิจัยที่เกี่ยวข้อง

1. งานศึกษาในอดีต (Empirical Studies)

Patel, Shah, Thakkar, and Kotecha (2015)ได้ทำการพยากรณ์มูลค่าในอนาคตของดัชนีตลาดหุ้นสองตัว ได้แก่ CNX Nifty และ S&P Bombay Stock Exchange (BSE) จากตลาดหุ้นอินเดียเพื่อทำการทดลอง การทดสอบอ้างอิงจากข้อมูลย้อนหลัง 10 ปีของดัชนีทั้งสอง โดยจะการคาดการณ์ผลตอบแทนล่วงหน้า 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 ซึ่งทำให้เกิดวิธีการแบบฟิวชันสองขั้นตอนในการใช้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองดังนี้ SVR-ANN, SVR - RF และ SVR - SVR เพื่อเปรียบเทียบประสิทธิภาพความแม่นยำของการพยากรณ์ของโมเดลเหล่านี้ในสถานการณ์เดียวเดียวกันกับที่การวิธีแบบขั้นตอนเดียว ANN, RF และ SVR โดยใช้ข้อมูลปัจจัย

ทางเทคนิค(technical indicator) 10 ตัว เป็นตัวแปรต้นสำหรับแบบจำลองในการทำนายแต่ละแบบ ตัวชี้วัดทางเทคนิค (technical indicator) เป็นตัวเลือกทั่วไปสำหรับนำเข้าเป็นตัวแปรอินพุตของวิธีการใช้ Machine Learning เช่น(Basak, Kar, Saha, Khaidem, & Dey, 2019; Kim, 2003)

Ma, Han, and Wang (2021) ได้ศึกษาการคาดการณ์ผลตอบแทนของ ดัชนี CSI 100 ของตลาดหุ้นจีน โดยใช้ Machine Learning และ Deep Learning ในการทำการพยากรณ์ ซึ่งการใช้แบบจำลองทั้งสองนั้นคาดการณ์ผลตอบแทนได้ดีกว่าแบบจำลองอนุกรมเวลา นอกจากนี้ วิธีการ Machine Learning ทั้งหมดถือว่ามีประสิทธิภาพเหนือกว่ากลยุทธ์ Buy-and-Hold ใน Trading simulation (Nevasalmi, 2020)

Banerjee (2019) ได้ทำการทดลองนำอัตราส่วนทางการเงิน(Financial ratios) มาทำนายผลตอบแทนของหุ้นของบริษัท 30 แห่งที่จดทะเบียนในตลาดการเงินดูไบและตลาดหลักทรัพย์อาบูดาบี ผลคืออัตราส่วนทางการเงินสามารถช่วยให้นักลงทุนคาดการณ์ผลตอบแทนของหุ้นในปีถัดไปได้

วิธีการ เทคนิคและรายละเอียดของแบบจำลอง (Methodology)

1. การใช้คอมพิวเตอร์เรียนรู้ด้วยตนเอง (Machine Learning) คือการทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูลแบ่งออกเป็น 2 ประเภทหลักๆคือ

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

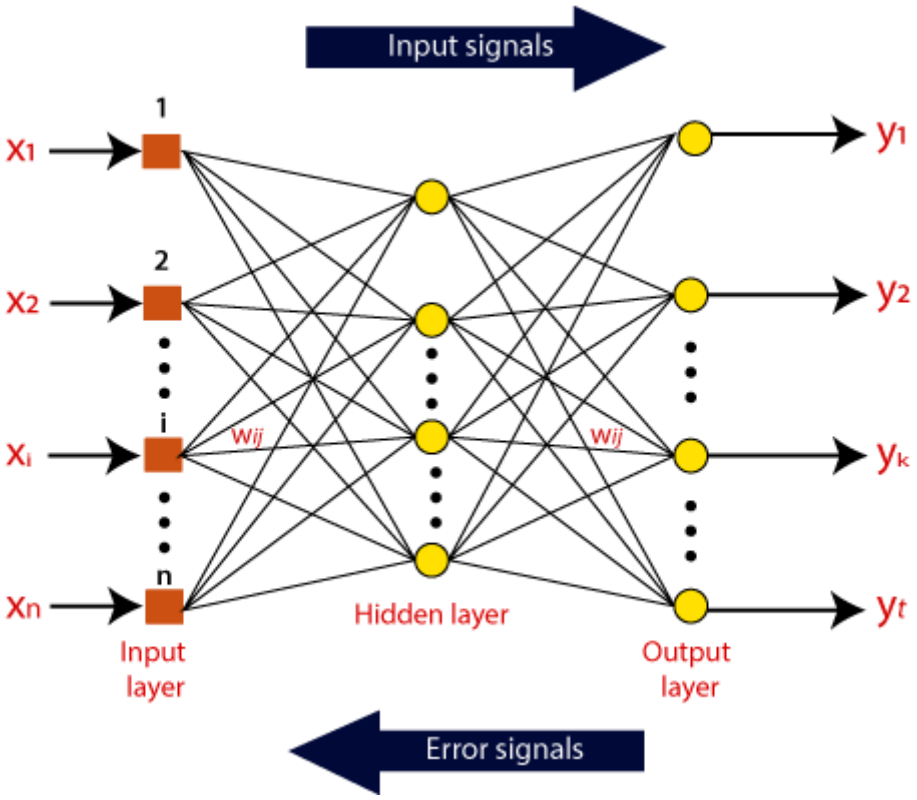
การเรียนรู้โดยไม่มีผู้ช่วยสอน (Unsupervised Learning) เป็นการให้ระบบคอมพิวเตอร์เรียนรู้ด้วยการจำแนกข้อมูล วิธีนี้เราจะใส่เพียงข้อมูลนำเข้า (Input) จากนั้นระบบคอมพิวเตอร์จะทำการจำแนกข้อมูล (Clustering) โดยวิธีนี้จะเน้นการใช้งานในรูปแบบการวิเคราะห์ข้อมูล (Analysis)

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

2. โครงข่ายประสาทเทียม (Artificial Neural Network, “ANN”) เป็นการทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเองโดยใช้ข้อมูล (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

การสร้างแบบจำลองนี้เริ่มต้นจากป้อนข้อมูลตัวแปรต้น (Input Node) หรือปัจจัยต่างๆที่มีผลต่อผลตอบแทนของตราสารทุนแต่ละตัว และผลลัพธ์ตัวแปรตาม (Output Node) ซึ่งเป็นเหมือนเฉลยในที่นี้คือผลตอบแทน 1 วัน 1 เดือน และ3 เดือนของหุ้นแต่ละตัว เมื่อป้อนข้อมูลหลายๆชุดให้คอมพิวเตอร์เรียนรู้เพื่อหารูปแบบสร้างเป็นแบบจำลองไว้ใช้ในการพยากรณ์หรือคาดการณ์ผลตอบแทนเมื่อป้อนตัวแปรต้นใหม่ๆเข้าไปแบบจำลองก็จะสามารถคาดการณ์ตัวแปรตามได้ใกล้เคียงค่าจริงที่เกิดขึ้นได้โดยวัดจากฟังก์ชันเปรียบเทียบค่าจริงกับค่าพยากรณ์ หรือฟังก์ชันวัดความคลาดเคลื่อน (Cost Function)



รูปที่ 1 โครงสร้างแบบง่ายของการทำงานของโครงข่ายประสาทเทียม

ที่มา: https://www.javatpoint.com/artificial-neural-network)

3. 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 [Weight (X)] ที่ Hidden

Layer ได้ผลลัพธ์เป็นความน่าจะเป็นของตัวแปรตาม (Predicted Probability) สามารถอธิบายเป็นสมการคณิตศาสตร์ได้ตามสมการที่ 3.1

สมการที่ 3.1 อธิบายการคำนวณผ่านฟังก์ชันการวิเคราะห์ถดถอยโลจิสติกส์ ซึ่งประกอบด้วยน้ำหนักตัวแปรต้น (W or Weight), ค่าของตัวแปรต้น (Input or X or Feature) และค่าความคลาดเคลื่อนของสมการ (B or Bias or Logistic regression intercept term) โดยแสดงให้เห็นภาพการทำงานของโครงข่ายประสาทเทียม 1 เส้นโครงข่าย

$$Activation\ Function\left[\sum(\text{Weights} \times \text{Inputs})\right] + \text{Bias} = \text{Output}$$

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

3.1.1 Sigmoid เป็นฟังก์ชันเส้นตรงอย่างง่ายโดยช่วงข้อมูลที่ออกจากฟังก์ชันจะอยู่ในช่วง 0-1 ตามสมการ 2 และรูปที่ 1 โดยข้อดีของฟังก์ชันนี้คือเข้าใจได้ง่าย สามารถใช้ได้ในงานวิเคราห์ความน่าจะเป็น (Probability) หรืองานจำแนกกลุ่ม (Segmentation or Boolean) ข้อเสียคือถ้าตัวแปรต้นมีค่าน้อยกว่า -5 หรือมากกว่า 5 ความชันจะเข้าใกล้ 0 จนเกิดปัญหา Optimizer ไม่ปรับค่าของน้ำหนักของตัวแปรต้นในโครงข่ายประสาทเทียมในขั้นตอนเรียนรู้ของแบบจำลอง (Vanishing Gradient Problem)

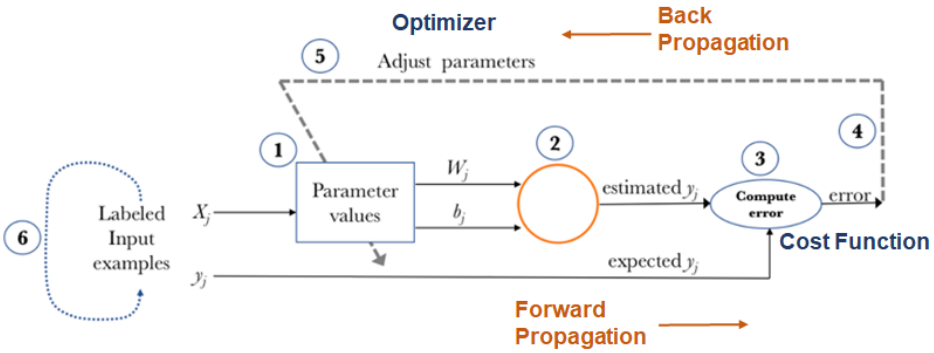
3.1.2 Rectified Linear Unit (ReLU) เป็นฟังก์ชันเส้นตรงที่ถูกปรับแก้ (Rectified) ซึ่งฟังก์ชันนี้เมื่อ Input ตัวแปรต้น X เป็นบวก Slope ของกราฟจะเป็น 1 เสมอตามสมการ 3.3 และรูปที่ 2 สำหรับข้อดีของ ReLU คือช่วยให้ขั้นตอนการเรียนรู้ของแบบจำลองผ่าน Optimizer ซึ่งจะกล่าวในหัวข้อถัดไปนั้นทำงานได้เร็วขึ้น อีกทั้งช่วยลดปัญหาการไม่ปรับค่าของน้ำหนักของตัวแปรต้นในโครงข่ายประสาทเทียมในขั้นตอนเรียนรู้ของแบบจำลอง (Vanishing Gradient Problem) แต่ ReLU มีข้อจำกัดคืออาจจะทำให้ข้อมูลออก (Output) ไม่สมดุลทำให้ผลการคาดการณ์หาจุดเหมาะสมได้ยาก รวมถึงช่วงข้อมูลออกเป็นได้ตั้งแต่ 0 ถึงไม่จำกัด ส่งผลให้จัดการข้อมูลออกได้ยากกว่า หากเมื่อเทียบกับข้อดีที่มากกว่าแล้วนั้นทำให้ Activation Function นี้เป็นที่นิยมในการใช้งานในแบบจำลองโครงข่ายประสาทเทียม สำหรับตัวแปรต้นก่อนเข้าแบบจำลองของงานวิจัยนี้มีค่าอยู่ในช่วง 0-1 จึงใช้ฟังก์ชันนี้ในทุกโหนดของ Input Layer และ Hidden Layer ในแบบจำลองโครงข่ายประสาทเทียม

3.1.3 Hyperbolic Tangent (Tanh) เป็นฟังก์ชันที่มีข้อดีในเรื่องของข้อมูลออก (Output) มีความสมดุล มีการกระจายตัวของค่าเฉลี่ย (Mean) เท่ากับ 0 ทำให้การการเรียนรู้ของแบบจำลองผ่าน Optimizer ทำได้ง่ายขึ้น โดยช่วงข้อมูลออกอยู่ในช่วง -1 ถึง 1 นิยมใช้งานเพื่อช่วยเป็นการทำให้ข้อมูล

อยู่ในรูปอย่างง่าย (Normalization) ฟังก์ชันสามารถอธิบายได้ง่าย ข้อจำกัดของ Tanh อาจจะเกิดปัญหา Vanishing Gradient Problem ตามที่กล่าวไว้ในหัวข้อก่อนหน้าได้ในกรณีที่ข้อมูลเข้ามีค่าน้อยกว่า -3 หรือมากกว่า 3 ซึ่งส่งผลให้ความชันเข้าใกล้ 0 โดยสมการของฟังก์ชันแสดงอยู่ใน สมการที่ 3.4 และกราฟ 3.3 แสดงให้เห็นการกระจายตัวของข้อมูลบนเส้นฟังก์ชัน Tanh ในส่วน Output Layer ของแบบจำลองโครงข่ายประสาทเทียมใช้ฟังก์ชัน Tanh เนื่องจากความน่าจะเป็นของข้อมูลผลตอบแทนตราสารทุนทั้งหมดในงานวิจัยมีช่วงไม่เกิน -1 ถึง 1

4. การเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม (Learning process of a neural network)

จากการทำงานของโครงข่ายประสาทเทียมที่กล่าวมาข้างต้น ตั้งแต่ข้อมูลตัวแปรต้นส่งเข้าโครงข่ายประสาทเทียมในแต่ละ Layer เพื่อกำหนดน้ำหนักเริ่มต้นของตัวแปรต้นแต่ละตัว (Weight) รวมทั้งค่าความคลาดเคลื่อนของสมการ (B0 or Bias or Logistic regression intercept term) จนได้ข้อมูลออกจากแต่ละเส้นโครงข่ายรวมกันจนถึง Output Layer ผ่าน Activation Function ได้เป็นค่าคาดการณ์ตัวแปรตาม (Predicted Y) หรือผลตอบแทนของตราสารทุนแต่ละตัวที่คาดการณ์ได้ กระบวนการนี้เรียกว่า Forward Propagation Process หลังจากนั้นค่าคาดการณ์ตัวแปรตาม (Predicted Y) หรือผลตอบแทนของตราสารทุนแต่ละตัวที่คาดการณ์ได้นั้น จะถูกนำไปเทียบกับตัวแปรตาม หรือค่าผลตอบแทนของตราสารทุนจริงที่ส่งเข้ามาให้แบบจำลองเรียนรู้ผ่านฟังก์ชันวัดความคลาดเคลื่อน (Cost Function) โดยการทำงานจะใช้อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) ในการปรับค่าน้ำหนักของตัวแปรต้นและค่าความคลาดเคลื่อนของสมการ โดยเป้าหมายเพื่อให้ค่าคลาดเคลื่อนที่ได้จากฟังก์ชันวัดความคลาดเคลื่อนมีค่าต่ำที่สุด กลไกการปรับค่าน้ำหนักของตัวแปรต้นและค่าความคลาดเคลื่อนของสมการซึ่งถูกส่งกลับไปที่โครงข่ายประสาทเทียมแต่ละเส้นโครงข่ายเพื่อทำการคำนวณใหม่นั้นเรียกว่า Back Propagation Process



รูปที่ 2 ขั้นตอนการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม เรียงตามลำดับตั้งแต่ข้อมูลตัวแปรต้นเข้าจนถึงการส่งค่ากลับเพื่อปรับน้ำหนักตัวแปรเพื่อให้เกิดการเรียนรู้ของแบบจำลองเพื่อการพยากรณ์ค่าตัวแปรตามให้ใกล้เคียงค่าจริงมากขึ้น

4.1 Cost Function เป็นฟังก์ชันวัดความคลาดเคลื่อนซึ่งใช้ในการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียม ซึ่งถูกใช้ในขั้นตอนเปรียบเทียบหาค่าคลาดเคลื่อนของตัวแปรตามที่พยากรณ์ได้กับค่าตัวแปรตามจริงที่ป้อนเข้ามาให้แบบจำลองเรียนรู้ ซึ่งในงานวิจัยนี้จะเป็นการเปรียบเทียบตัวแปรตามที่เป็นการคาดการณ์ผลตอบแทนรายเดือนของตราสารทุน กับผลตอบแทนรายเดือนของตราสารทุนจริงที่เกิดขึ้น โดยฟังก์ชันที่ใช้คือ Mean Square Error (MSE) ซึ่งจะทำงานร่วมกับอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) เพื่อปรับให้ค่าน้ำหนักตัวแปรต้น เพื่อหาค่าฟังก์ชันวัดความคลาดเคลื่อนที่ต่ำที่สุดในทุกๆรอบการเรียนรู้ (epochs) ของแบบจำลองโครงข่ายประสาทเทียม ทั้งนี้ฟังก์ชันวัดความคลาดเคลื่อนยังใช้เป็นผลในการทดสอบแบบจำลองโครงข่ายประสาทเทียมเพื่อเปรียบเทียบและเป็นการวัดผลความแม่นยำในการคาดการณ์ผลตอบแทนของแบบจำลองโครงข่ายประสาทเทียมอีกด้วยสำหรับรายละเอียดของฟังก์ชันวัดความคลาดเคลื่อนที่ใช้ในงานวิจัยมีดังนี้

4.1.1 Mean Square Error (MSE) คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์ โดยแสดงเป็นผลเฉลี่ยของค่าคลาดเคลื่อนของทุกๆจุดเวลาที่แบบจำลองทำการพยากรณ์บนหนึ่งช่วงข้อมูล ซึ่งฟังก์ชันใช้ในแบบจำลองโครงข่ายประสาทเทียมเป็น Cost Function ที่ทำงานร่วมกับอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) เพื่อปรับให้ค่าน้ำหนักตัวแปรต้นในขั้นตอนการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียมในงานวิจัยนี้ อีกทั้งจะใช้ในการประเมินความแม่นยำของแบบจำลองในการพยากรณ์ผลตอบแทนตราสารทุนทั้งในขั้นตอนเรียนรู้, ตรวจสอบ และทดสอบแบบจำลอง

4.1.2 Root Mean Square Error (RMSE) คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์เช่นเดียวกัน แต่มีการเพิ่ม Square Root ในสมการเพื่อให้สะท้อนค่าเฉลี่ยของค่าคลาดเคลื่อนแต่ละจุดที่มีขนาดใหญ่ RMSE จะให้น้ำหนักของค่าคลาดเคลื่อนดังกล่าวมากกว่าจึงช่วยในการเปรียบเทียบผลได้ดีมากขึ้น ซึ่งฟังก์ชันนี้ใช้ขั้นตอนในการประเมินความแม่นยำของแบบจำลองในการพยากรณ์ผลตอบแทนตราสารทุนทั้งในขั้นตอนเรียนรู้, ตรวจสอบ และทดสอบแบบจำลองซึ่งเป็นตัวประเมินร่วมกับ Mean Square Error (MSE)

4.1.3 Mean Absolute Error (MAE) คือฟังก์ชันเปรียบเทียบความแตกต่างระหว่างค่าจริงกับค่าคาดเคลื่อนของตัวพยากรณ์เช่นเดียวกัน ซึ่งเป็นตัววัดหน่วยอิสระ (Unit-free measure) เพื่อวัดค่าสัมบูรณ์ของค่าความคลาดเคลื่อนเฉลี่ยของผลพยากรณ์ ซึ่งใช้วัดค่าคลาดเคลื่อนที่เกิดขึ้นบนข้อมูลที่มีการเปลี่ยนแปลงเพียงเล็กน้อยได้ดี ถูกใช้ในขั้นตอนประเมินความแม่นยำของแบบจำลองเช่นเดียวกัน

4.2 Optimizer เป็นอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) ทำหน้าที่เป็นกลไกการปรับปรุงค่าน้ำหนักของตัวแปรต้นต่าง ๆ รวมถึงค่าคลาดเคลื่อน (Bias) ในขั้นตอนการเรียนรู้ของแบบจำลองโครงข่ายประสาทเทียมทำให้ Output หรือผลคาดการณ์ของแบบจำลองที่ได้เข้าใกล้ค่าจริงที่กำหนดให้แบบจำลองใช้เรียนรู้ ซึ่งการทำงานของอัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) คือ Gradient Descent ซึ่งเป็นหลักการหามุมของฟังก์ชันวัดความคลาดเคลื่อน (Cost Function) (θ Theta) ที่ต่ำที่สุด ซึ่งภายในฟังก์ชันวัดความคลาดเคลื่อน

ตามที่กล่าวข้างต้นเป็นการใช้น้ำหนักของตัวแปรตาม (Weight) คูณกับตัวแปรตามผ่าน Activation Function จนได้ค่าคาดการณ์ตัวแปรตามมาเปรียบเทียบหาความต่างกับค่าจริงนั้น ค่า Theta ที่ดีที่สุดจะมาจากค่า Theta ก่อนหน้าซึ่งถูกปรับลดด้วย Learning rate คูณผลฟังก์ชัน Mean Square Error ของค่า Theta โดยปรับซ้ำหลายรอบจนค่าน้ำหนักของตัวแปรต้นที่ได้มาเทียบใน ฟังก์ชัน Mean Square Error ส่งผลให้ได้ค่า Theta ที่ต่ำที่สุด ตามสมการ 3.13

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 แสดงโดยกราฟ 3.4 ซึ่งเป็นการเปรียบเทียบความสามารถของ Optimizer หลายๆตัวจะเห็นได้ว่า ADAM มีความสามารถในการลดค่าคลาดเคลื่อนของ Cost Function ได้ดีที่สุด

5. แบบจำลอง Random Forest (RF) เป็นหนึ่งในกลุ่มของโมเดลที่เรียกว่า Ensemble learning ที่มีหลักการคือการเทรนโมเดลที่เหมือนกันหลายๆ ครั้ง (หลาย Instance) บนข้อมูลชุดเดียวกัน โดยแต่ละครั้งของการเทรนจะเลือกส่วนของข้อมูลที่เทรนไม่เหมือนกัน แล้วเอาการตัดสินใจของโมเดลเหล่านั้นมาโหวตกันว่า Class ไหนถูกเลือกมากที่สุด

5.1 กระบวนการทำงาน (Process) โมเดลทำงานโดยการรวมการตัดสินใจของผู้ตัดสินใจจำนวนมากเข้าด้วยกันมักจะให้ผลการตัดสินใจที่แม่นยำมากกว่าการพึงพาการตัดสินใจจากแหล่งเดียว การเรียนรู้แบบ Ensemble นี้จะทำงานได้ดีบนเงื่อนไขที่ว่า โมเดลผู้ทำนายแต่ละตัวจะต้องเรียนรู้อย่างเป็นอิสระต่อกันให้มากที่สุด เหมือนว่าคนแต่ละคนจะต้องตัดสินใจด้วยตนเองให้มากที่สุดโดยไม่ได้รับข้อมูลจากคนอื่นหรือนำเอาข้อมูลจากคนอื่นมาเป็นส่วนในการตัดสินใจ ตัวอย่างวิธีการคือกำหนดจำนวนการสร้าง Decision Tree โดยกำหนดจำนวน คือ 1,000 ต้น เพื่อสุ่มตัวอย่างข้อมูล โดยการสุ่มข้อมูลตัวอย่าง (Bootstrapping หรือการสร้างต้นไม้หลายๆต้นไม่ให้ซ้ำกัน) จาก Data set ที่เป็นตัวแปรนำเข้า ให้ได้ข้อมูลออกมา 1,000 ชุดที่ไม่เหมือนกัน ตามจำนวน Decision Tree ใน Random Forest เพื่อคำนวณหาผลลัพธ์เป็นข้อมูลออก (Output) ตามที่ได้ทำการให้โมเดลเรียนรู้

6. 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 ชุดที่เข้า

มาในโหนดจะถูกรวมเข้าด้วยกัน ก่อนจะถูกแยกผลลัพธ์ออกเป็น 2 ส่วน คือ ผลลัพธ์ที่ได้จากโหนดนั้น ๆ และผลลัพธ์ที่จะถูกนำไปเป็นข้อมูล Input ของโหนดถัดไป เทคนิค RNN นั้นเหมาะนำมาใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (Sequence) หรือข้อมูลที่มีความต่อเนื่อง เช่น ข้อมูลอนุกรมเวลา (Time Series), ข้อมูลเสียง, ข้อมูลประเภทข้อความ, ข้อมูลภาพและวิดิทัศน์ เป็นต้น(Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017)

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

Long short-term memory (LSTM) เป็นโครงข่ายประเภท RNN รูปแบบหนึ่งที่ถูกพัฒนาขึ้นมาให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหลักการทำงานคือ สามารถเก็บ ‘สถานะ’ หรือขึ้นมูลของแต่ละโหนดเอาไว้เพื่อที่เวลาย้อนกลับไปดูจะได้ทราบถึงที่มาของข้อมูลค่าดังกล่าวว่าเดิมเป็นค่าอะไร และจุดเด่นของแบบจำลอง LSTM คือฟังก์ชันพิเศษที่มีหน้าที่เสมือนประตู(Gate) ที่คอยความคุมข้อมูลที่จะเข้าในแต่ละโหนด ซึ่งประกอบด้วย Forget gate layer, Input gate layer และ Output gate layer(Jozefowicz, Zaremba, & Sutskever, 2015)

6.1 Forget gate layer เป็น Gate ที่มีหน้าที่ในการกำหนดว่าข้อมูลที่เข้ามาใน Cell นั้นควรจะถูกเก็บไว้หรือควรจะทิ้งไป ซึ้งข้อมูลที่ถูกตัดสินว่าควรเก็บไว้นั้นจะถูกประเมินจากข้อมูล Input ที่เข้ามาในโหนดนั้นๆ รวมกับผลลัพธ์ที่จะได้จากการคำนวณของโหนดก่อนหน้า ผ่านฟังก์ชัน ReLUผล ลัพธ์ที่ได้จาก Forget gate layer จะอยู่ระหว่างค่า 0 และ 1 ซึ่งถ้าได้ค่าเป็น 0 นั้น หมายถึงให้ลบค่า Cell state เดิมออก แต่ถ้าได้ค่าเป็น 1 นั้นหมายถึงให้เก็บค่า Cell state นี้ต่อไป

6.2 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 เลือกที่จำทำการ Update cell state ฟังก์ชัน tanh ก็จะทำการสร้าง Candidate values ขึ้นมาใน State

6.3 Output gate layer เป็น Gate ที่มีหน้าที่เตรียมทำการส่งออกข้อมูล (Output data) โดยข้อมูลที่จะทำการ Output นั้นจะดูจาก Cell state ที่ผ่านกระบวนการคำนวณต่างๆแล้ว โดยฟังก์ชัน ReLU จะเป็นตัวเลิกว่าข้อมูลส่วนไหน Cell state ที่จะถูก Output จากนั้นก็จะนำค่า Cell state เข้าฟังก์ชัน

tanh (เพื่อหาว่าจะได้ค่าออกมาเป็น 1 หรือ -1) แล้วนำค่าที่ได้จากฟังก์ชัน tanh มาทำการคำนวณกับค่า Output ที่ได้จาก ReLU gate จากนั้นจะได้ค่า Output ที่ต้องการ

ข้อมูลที่ใช้ในงานวิจัย (Data)

1. Stock selection

ข้อมูลที่ใช้ในการคาดการณ์ผลตอบแทนของพอร์ตโฟลิโอตราสารทุนนั้น เลือกใช้ข้อมูลตราสารทุนที่เป็นองค์ประกอบของดัชนี SET100 จากตลาดหลักทรัพย์แห่งประเทศไทย (Stock Exchange of Thailand) ณ เดือนมกราคม 2009 รายชื่อตาม ตารางที่ 1 หลังจากนั้นเพื่อแก้ปัญหาความไม่สมบูรณ์ของข้อมูลและเพื่อสร้างความแม่นยำของแบบจำลองจึงต้องคัดกรองตราสารทุน โดยมีวิธีคัดกรองตราสารทุน คือ ตราสารทุนที่จะนำเข้าไปในแบบจำลองต้องมีข้อมูลเพียงพอในช่วงที่จัดทำแบบจำลอง คือ 05/01/2009 – 22/04/2021 หรือเทียบเท่ากับระยะเวลา 12 ปี 4 เดือนและไม่อยู่ในกลุ่ม Banking Sector เนื่องจากโครงสร้างของงบการเงินทั้งสอง Sector ข้างต้นมีความแตกต่างจาก Sector อื่นๆ ซึ่งไม่เหมาะสมต่อการนำมารวมกันพิจารณาในงานวิจัยนี้

2. Factor Selection 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)

เพื่อแก้ปัญหาความไม่สมบูรณ์ของข้อมูลและเพื่อสร้างความแม่นยำของแบบจำลองจึงต้องมีการปรับตัวแปรโดยมีวิธีดังนี้

2.1 การจัดเรียงข้อมูลใหม่ ซึ่งข้อมูลที่นำเข้าแบบจำลองจะต้องมีความถี่เป็นรายวันเท่านั้น ดังนั้นข้อมูลที่มีความถี่น้อยกว่ารายวัน เช่น ข้อมูลเศรษฐศาสตร์มหภาค, ข้อมูลจากงบการเงิน จะเป็นข้อมูลที่มีความถี่รายปี, รายไตรมาส และรายเดือน ซึ่งจะต้องปรับให้เข้าสู่ความถี่รายวันโดยการเลือกข้อมูลที่ความถี่น้อย

กว่าเหล่านั้น กระจายเข้าสู่ข้อมูลที่มีความถี่รายวันโดยเลือกตัวแทนของช่วงความถี่ที่น้อยกว่า ที่เกิดขึ้นก่อนวันนั้นๆของข้อมูล

3. 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 แต่ในงานวิจัยนี้ใช้สูตรการปรับความถี่ของข้อมูลโดยพิจารณาจากความห่างของค่าต่ำสุดเป็นสัดส่วนต่อค่าของช่วงข้อมูลทั้งหมด ตามสมการที่ 3

สมการ 3 แสดงการทำ Data Normalization ของตัวแปรต้นทุกตัวที่ใช้ในแบบจำลองโครงข่ายประสาทเทียม

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

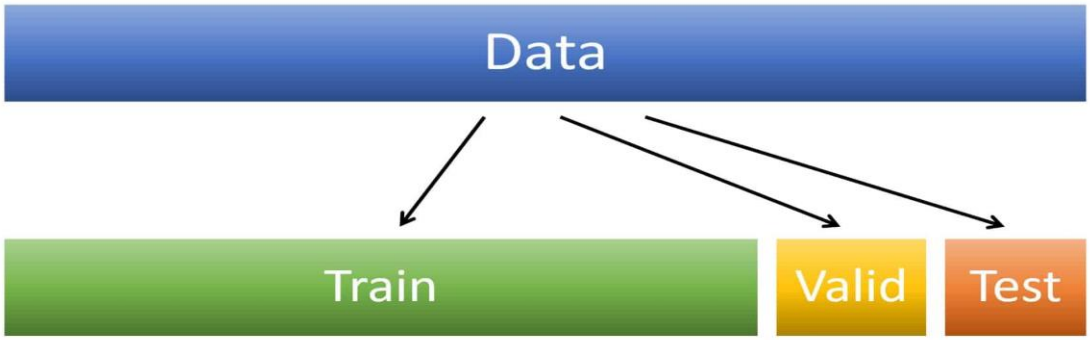
4. การแบ่งข้อมูล

การนำเข้าข้อมูลไปยังแบบจำลองจะแบ่งชุดข้อมูลออกเป็น 3 ชุด ตามรูปที่ 3 ดังนี้

4.1 ชุดพัฒนาแบบจำลอง (Train Set) โดยเป็นชุดข้อมูลที่ใช้เพื่อให้แบบจำลองเรียนรู้

4.2 ชุดทดสอบแบบจำลอง (Validate Set) ใช้สำหรับทดสอบแบบจำลองที่ระบบได้จัดทำจากข้อมูลชุดที่ 1 เพื่อทดสอบปัญหาเช่น การ Overfitting และ Underfitting ของแบบจำลองโดย Overfitting คือ การที่แบบจำลองถูกรบกวนด้วยตัวแปรหลายๆตัว ซึ่งทำให้แบบจำลองได้ผลดีในข้อมูลชุด Train แต่กลับให้ผลที่แย่ในข้อมูลชุดอื่นๆ จึงส่งผลให้แบบจำลองมีความคาดเคลื่อนมาก Underfitting คือ การที่แบบจำลองมีตัวแปรต้นที่ส่งผลในการอธิบายตัวแปรตามมีจำนวนน้อยเกินไปทำให้แบบจำลองมีความสามารถในการพยากรณ์ข้อมูลได้แม่นยำน้อย ซึ่งหากผลการทดสอบแบบจำลองได้ผลไม่ดี กล่าวคือค่าจาก Cost Function มีค่าสูง จะทำการกลับไปปรับแก้ไขโครงสร้างของแบบจำลองโครงข่ายประสาทเทียมให้เหมาะสมและทำการรันทดสอบจนกว่าค่าที่ได้จะอยู่ในเกณฑ์ที่ดีจึงจะนำแบบจำลองดังกล่าวไปใช้ในการคาดการณ์ผลตอบแทนของตราสารทุนในลำดับต่อไป

4.3 ชุดคาดการณ์แบบจำลอง (Test Set) 20 วัน ก่อน Rolling ออกรอบละ 20 วัน และเพิ่มเข้าใหม่รอบละ 20 วัน ใช้สำหรับคาดการณ์ผลตอบแทนรายวัน, รายเดือน และราย 3 เดือน ของตราสารทุนโดยแบบจำลองที่ผ่านการจัดทำจากข้อมูลชุดที่ 1 และทดสอบโดยข้อมูลชุดที่ 2 แล้ว



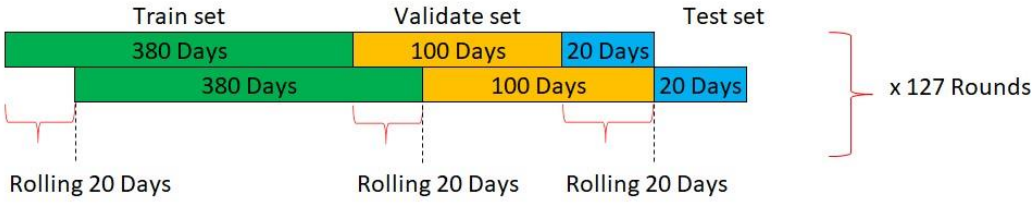
รูปที่ 3 การแบ่งช่วงข้อมูลสำหรับการเรียนรู้ การทดสอบและการคาดการณ์ผลตอบแทนของตราสารทุนของแบบจำลองแต่ละตัว

ผลการวิจัย

งานวิจัยนี้ภาพรวมจะถูกแบ่งออกเป็นสองส่วนย่อยคือ การพยากรณ์ความถี่รายวันของผลตอบแทนรายวัน, รายเดือน และราย 3 เดือนของตราสารทุน แต่ละตัวทั้งหมด 61 แบบจำลองโดยใช้ Machine Learning 3 อัลกอริทึม (ANN, RF และLSTM) ซึ่งการที่ออกแบบด้วยการแยกเป็น 61 แบบจำลองเนื่องจากแต่ละตราสารทุนมีปัจจัยที่มีผลต่อการคาดการณ์ผลตอบแทนที่แตกต่างกัน หลังจากนั้นจึงนำผลตอบแทนที่พยากรณ์ได้มาเปรียบเทียบความแม่นยำกันระหว่างแบบจำลองทั้ง 3 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 ทั้ง RMSE, MAE และ MSE ซึ่งกล่าวในบทก่อนหน้าในการทดสอบว่าแบบจำลองทำงานได้มีประสิทธิภาพในการคาดการณ์ผลตอบแทนได้แม่นยำหรือไม่ หากไม่แม่นยำจะทำการกลับไปปรับแบบจำลองตั้งแต่ขั้นตอนการออกแบบให้เหมาะสม เช่น เพิ่มเส้นโครงข่ายประสาทเทียม, ปรับ Activation Functionให้เหมาะสมกับข้อมูล, เพิ่มจำนวนโหนดในแต่ละ Layer ของโครงข่ายประสาทเทียม เป็นต้น และทำตามขั้นตอนข้างต้นวนไปทั้งหมด 127 รอบโดยการขยับช่วงข้อมูลไปข้างหน้ารอบละ 20 วัน ซึ่งยังคงโครงสร้างการใช้ข้อมูล 480 สัปดาห์ก่อนหน้านี้ในการให้แบบจำลองเรียนรู้และทำการทดสอบแบบจำลอง (Train set 380 days + Validate set 100 days) ตามรูปที่ 4

เมื่อการพยากรณ์ผลตอบแทนของตราสารทุนแต่ละตัวแล้วจะนำผลที่ได้ไปวัดความแม่นยำของแต่ละแบบจำลอง โดยใช้ RMSE, MAE และ MSE ในการเปรียบเทียบกันว่าแบบจำลองได้มีค่าความแม่นยำสูงที่สุด และเปรียบเทียบการพยากรณ์ด้วยว่าช่วงการพยากรณ์ใด (1 วัน, 1 เดือน และ 3 เดือน) มีความแม่นยำที่สุด



รูปที่ 4 การขยับช่วงข้อมูลในแต่ละรอบการปรับแบบจำลองในการคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว

1. 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 มาแล้ว ทำให้ข้อมูลอยู่ในช่วง 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) และในส่วนของ RF กำหนดจำนวนการสร้าง Decision Tree โดยกำหนดจำนวน คือ 305 ต้น เพื่อสุ่มตัวอย่างข้อมูล โดยการสุ่มข้อมูลตัวอย่าง (Bootstrapping หรือการสร้างต้นไม้หลายๆต้นไม่ให้ซ้ำกัน) จาก Date set ที่เป็นข้อมูลตัวแปรต้น (Input data) ให้ได้ข้อมูลออกมา 305 ชุด ที่ไม่เหมือนกัน ตามจำนวน Decision Tree ใน Random Forest

การพัฒนาแบบจำลองของตราสารทุนแต่ละตราสารมีทั้งหมด 61 แบบจำลองตามจำนวนตราสารทุนที่ใช้ในการคาดการณ์ผลตอบแทนในแต่ละอัลกอริทึม ตามความแตกต่างของปัจจัยมีผลในการคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว อีกทั้งต้องการให้แบบจำลองสามารถทำงานได้อิสระเกิดความเฉพาะใน

การคาดการณ์ผลตอบแทนของตราสารทุนแต่ละตัว โดยการทำงานในแบบจำลองแต่ละตัวจะคาดการณ์ผลตอบแทนรายวัน รายเดือน และราย 3 เดือน ของตราสารทุน ซึ่งการคาดการณ์ผลตอบแทนนี้จะทำรายวัน กล่าวคือมีข้อมูลปัจจัยที่มีผลหรือตัวแปรต้นเป็นรายวันย้อนหลัง และป้อนผลตอบแทนรายวัน รายเดือน และราย 3 เดือน ณ วันนั้นๆให้แบบจำลองเรียนรู้ และนำข้อมูลปัจจัยที่มีผลหรือตัวแปรต้น ณ วันที่จะทำการคาดการณ์ ป้อนใส่แบบจำลองเพื่อให้คาดการณ์ผลตอบแทนในแต่ละแบบของวันนั้นออกมา ซึ่งทำพยากรณ์ผลตอบแทนต่อเนื่องทั้งหมด 127 รอบ เพื่อให้แบบจำลองเกิดความแม่นยำมากที่สุด

สำหรับตารางที่ 1–3 เป็นการวัดผลแบบจำลองในขั้นตอนการคาดการณ์แบบจำลอง (Model Testing) โดยในขั้นตอนนี้มีค่าคลาดเคลื่อนที่คำนวนจาก MSE, RMSE และ MAE เพื่อเปรียบเทียบความแม่นยำในแต่ละแบบจำลองและในแต่ละช่วงผลตอบแทนที่พยากรณ์

ตารางที่ 1 ผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 วัน (1 Days) สำหรับทุกแบบจำลอง

Model	MSE	RMSE	MAE
ANN	7.01%	2.57%	1.85%
LSTM	8.34%	2.8%	1.99%
RF	6.61%	2.5%	1.8%

ตารางที่ 2 ผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 1 เดือน (1 Month) สำหรับทุกแบบจำลอง

Model	MSE	RMSE	MAE
ANN	53.48%	7.07%	4.94%
LSTM	56.53%	7.25%	5.02%
RF	43.83%	6.37%	4.51%

ตารางที่ 3 ผลค่าคลาดเคลื่อนจากผลการทดสอบแบบจำลองหรือช่วงการใช้แบบจำลองคาดการณ์ผลตอบแทนของตราสารทุน (Model Testing) แบบ 3 เดือน (3 Month) สำหรับทุกแบบจำลอง

Model	MSE	RMSE	MAE
ANN	68.06%	7.85%	5.3%
LSTM	68.91%	7.91%	5.34%
RF	55.54%	7.07%	4.85%

สรุปงานวิจัย

งานวิจัยนี้นำเสนอการคาดการณ์ผลตอบแทนตราสารทุนโดยใช้ Machine Learning 3 อัลกอริทึม (ANN, RF และLSTM) ที่เรียนรู้จากข้อมูลและปัจจัยที่มีผลกระทบต่อราคาตราสารทุน ผนวกกับการนำผลตอบแทนที่คาดการณ์วัดผลหาค่าความผิดพลาดเพื่อเปรียบเทียบว่าแบบจำลองได้ให้ค่าความผิดพลาดน้อยที่สุดหรือก็คือมีความแม่นยำมากที่สุด โดยผลจากการวิจัยพบว่าแบบจำลองที่มีความแม่นยำที่สุดก็คือแบบจำลอง Random Forest ซึ่งมีค่าความผิดพลาดต่ำที่สุดในทุกช่วงของการคาดการ์ผลตอบแทน (1 วัน, 1 เดือน และ 3 เดือน) หมายความว่าแบบจำลอง RF เหมาะสมกับการพยากรณ์ตราสารทุนในตลาดหลักทรัพย์ไทยกว่าแบบจำลอง ANN และLSTM

งานวิจัยนี้แสดงให้เห็นถึงการคาดการณ์ผลตอบแทนตราสารทุนที่มีความแม่นยำจากการใช้ Machine Learning โดยเลือกเทคนิคแบบจำลอง 3 อัลกอริทึม คือ ANN, RF และLSTM ซึ่งสามารถนำผลที่ได้ไปประยุกต์ใช้ประกอบการตัดสินใจในการเลือกหลักทรัพย์ที่จะลงทุนหรือกำหนดกลยุทธ์ให้เหมาะสมกับความต้องการของนักลงทุน รวมถึงการนำไปจัดการปรับพอร์ตโฟลิโอให้กับนักลงทุนสถาบันหรือนักลงทุนอื่นๆได้ 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.).

Banerjee, A. (2019). Predicting Stock Return of UAE Listed Companies Using Financial Ratios. *Accounting and Finance Research, 8*(2). doi:https://doi.org/10.5430/afr.v8n2p214

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

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

javatpoint. What is Artificial Neural Network. *Artificial Neural Network Tutorial.* Retrieved from https://www.javatpoint.com/artificial-neural-network

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

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

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

Nevasalmi, L. (2020). Forecasting multinomial stock returns using machine learning methods. *Journal of Finance and Data Science, 6*(1), 86-106. doi:https://doi.org/10.1016/j.jfds.2020.09.001

Olah, C. (2015). Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

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