

วิชา การเรียนรู้เชิงลึก (Deep Learning)  
รหัสวิชา 01204466

จัดทำโดย  
นายก้องภพ ไพรeras  
รหัสนิสิต 6610505268

นายปฐมพงศ์ บวรเจริญพันธุ์  
รหัสนิสิต 6610505462

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์

อาจารย์ผู้สอน  
อาจารย์ ภาณุ รัตนวนพันธุ์

มหาวิทยาลัยเกษตรศาสตร์  
ภาคต้น ปีการศึกษา 2568

# รายงานผลการดำเนินงานโครงการ

หัวข้อ: การพยากรณ์ราคาหุ้นด้วยข้อมูลเชิงเวลาแบบหลายตัวแปรโดยใช้สถาปัตยกรรม Transformer

## 1. บทนำ

การเคลื่อนไหวของราคาหุ้นเป็นกระบวนการที่มีความซับซ้อนและยากต่อการคาดการณ์ เนื่องจากได้รับอิทธิพลจากหลายปัจจัย ทั้งปัจจัยทางเศรษฐกิจ ภาวะตลาด และพฤติกรรมของนักลงทุน โครงการนี้จึงมีวัตถุประสงค์เพื่อพัฒนาแบบจำลองเชิงลึก (Deep Learning Model) ที่สามารถเรียนรู้รูปแบบการเปลี่ยนแปลงของข้อมูลราคาหุ้นในอดีต เพื่อใช้ในการพยากรณ์แนวโน้มในอนาคต โดยเฉพาะอย่างยิ่งการใช้สถาปัตยกรรม Transformer ซึ่งเป็นเทคโนโลยีที่ได้รับความนิยมสูงในปัจจุบัน เนื่องจากสามารถเรียนรู้ความสัมพันธ์ระยะยาวของข้อมูลเชิงเวลาได้อย่างมีประสิทธิภาพ

## 2. เหตุผลและความน่าสนใจของหัวข้อ

หัวข้อการพยากรณ์ราคาหุ้นเป็นเรื่องที่มีความสำคัญและได้รับความสนใจอย่างแพร่หลายในวงการวิศวกรรมข้อมูลและการเงิน เนื่องจากสามารถนำไปประยุกต์ใช้ในระบบแนะนำการลงทุน การบริหารความเสี่ยง และระบบเทรดอัตโนมัติ การใช้เทคโนโลยี Deep Learning เข้ามาช่วยในการวิเคราะห์ข้อมูลช่วยให้การเรียนรู้รูปแบบที่ซับซ้อนและไม่เชิงเส้น (non-linear) มีความแม่นยำและยืดหยุ่นมากกว่าวิธีการแบบดั้งเดิม

โครงการนี้จึงมีความน่าสนใจตรงที่สามารถนำแนวคิดของ Transformer-based Sequential Modeling มาประยุกต์ใช้กับข้อมูลทางการเงินที่มีลักษณะเป็นข้อมูลเชิงเวลา (time-series data) หลายมิติ ซึ่งสะท้อนพฤติกรรมของตลาดได้อย่างครบถ้วนมากกว่าเดิม

### 3. เหตุผลที่เลือกใช้ Deep Learning และการเปรียบเทียบกับวิธีอื่น

#### 3.1 วิธีดั้งเดิม

วิธีแบบดั้งเดิม เช่น ARIMA, Linear Regression หรือ GARCH มักหมายความว่ากับข้อมูลที่มีความสัมพันธ์เชิงเส้นและการแปรผันคงที่ แต่ในข้อมูลราคาหุ้นจริงซึ่งมีความผันผวนสูงและมีความไม่แน่นอน การใช้วิธีเชิงสถิติไม่สามารถจับความสัมพันธ์ระหว่างตัวแปรได้ครบถ้วน

#### 3.2 วิธี Deep Learning

โครงการนี้เลือกใช้สถาปัตยกรรม Transformer เนื่องจากสามารถเรียนรู้ความสัมพันธ์ระยะยาวในข้อมูล (long-term dependency) ได้โดยไม่ต้องพึงคำด้วยเวลาแบบ RNN และยังสามารถจัดการกับข้อมูลหลายตัวแปร (multivariate data) ได้ในคราวเดียว

ข้อดีของ Deep Learning:

- สามารถเรียนรู้ความสัมพันธ์ที่ไม่เชิงเส้นระหว่างตัวแปรได้ดี
- รองรับข้อมูลปริมาณมากและมีมิติสูง
- มีศักยภาพในการ generalize ไปยังหุ้นหลายตัวได้พร้อมกัน

ข้อจำกัด:

- ต้องใช้ทรัพยากรคอมพิวเตอร์สูง
- ต้องมีการเตรียมข้อมูล (data preprocessing) ที่รอบคอบ
- มีความซับซ้อนในการปรับค่าพารามิเตอร์

## 4. สถาปัตยกรรมของโมเดล Deep Learning

โครงงานนี้ใช้สถาปัตยกรรม Transformer Encoder Network สำหรับการพยากรณ์ข้อมูลเชิงเวลา โดยมีโครงสร้างดังนี้

### 4.1 สรุปองค์ประกอบของสถาปัตยกรรม

ส่วนประกอบ	รายละเอียด
Input Layer	รับข้อมูล 37 ตัวแปร/ไฟเจอร์ (เช่น log-return, RSI, MACD, Volume features และ Cyclical time features) ย้อนหลัง 90 วัน (90 time steps)
Embedding Layer	แปลงมิติของไฟเจอร์จาก 37 เป็น 128 โดยใช้ nn.Linear
Positional Encoding	เพิ่มข้อมูลลำดับเวลา (Positional Encoding) เข้ากับ input embedding เพื่อให้โมเดลทราบลำดับของข้อมูล
Transformer Encoder (3 ชั้น)	ใช้กลไก Multi-Head Attention (8 Heads) จำนวน 3 ชั้น เพื่อจับความสัมพันธ์เชิงเวลา (long-term dependency) ระหว่างวัน
Feedforward Network	มีจำนวนโหนดซ่อน 256 และใช้ ReLU เป็น activation function ภายในแต่ละชั้นของ Encoder

Output Layer	รับผลลัพธ์จาก token วันล่าสุด และคาดการณ์ log-return ของหุ้นในอีก 5 วันข้างหน้า (Output dimension = 5)
--------------	---

#### 4.2 ภาพรวมโครงสร้าง (อธิบายเชิงโครงสร้าง)

Input (Batch, 90 Day, 37 Feature)

↓

Linear Projection (Features → 128)

↓

Positional Encoding

↓

Transformer Encoder × 3 Layers

↓

Feedforward (ReLU)

↓

Linear Output (128 → 5)

↓

Predicted 5-day log-return

## 5. การอธิบายโค้ดและโครงสร้างระบบ

โครงงานนี้พัฒนาโดยใช้ PyTorch ซึ่งแบ่งการทำงานหลักออกเป็น 3 ส่วนสำคัญดังนี้

### 5.1 ส่วนการเตรียมข้อมูล (Data Preprocessing)

ส่วนนี้รับผิดชอบในการเตรียมข้อมูลดิบที่ได้มาจากการ API (yfinance) ให้พร้อมสำหรับการฝึกสอนโมเดล โดยมีขั้นตอนการประมวลผลที่ซับซ้อนดังนี้:

- ดึงข้อมูลราคาหุ้นจาก Yahoo Finance API (yfinance)
- การสร้างฟีเจอร์เชิงลึก (Feature Engineering) ด้วยฟังก์ชัน `process_stock()` ซึ่งประกอบด้วย:
  - ฟีเจอร์จากราคาและปริมาณการซื้อขาย (Price and Volume Features): คำนวณ log-return (`log_ret`, `high_ret`, `low_ret`) และค่าการเปลี่ยนแปลงของปริมาณการซื้อขายในรูปแบบล็อก (`vol_ret`, `vol_log`)
  - ตัวชี้วัดทางเทคนิค (Technical Indicators): ใช้ไลบรารี `pandas_ta` ในการสร้างตัวชี้วัดหลักๆ เช่น ค่าเฉลี่ยเคลื่อนที่ (SMA, EMA, HMA), RSI, Stochastics, CMF, MACD, และ ATR
  - Bollinger Bands Features: คำนวณค่ากลาง (`mid`), ส่วนเบี่ยงเบนมาตรฐาน (`std`), ความกว้าง (`width`), และเปอร์เซ็นต์ตำแหน่งของราคากลาง (`bb_percent`)
  - ฟีเจอร์เชิงสัมพัทธ์และการปรับค่าให้เป็นบรรทัดฐาน (Relative/Normalized Features): แปลงค่าตัวชี้วัดที่ไม่ได้มีการปรับค่า เช่น SMA, EMA, Bollinger Band Mid ให้เป็นค่าสัมพัทธ์เทียบกับราคากลาง (`sma_5_rel = SMA_5 / Close - 1`) เพื่อให้ข้อมูลมีความคงที่ (stationary)
  - ฟีเจอร์เชิงโมเมนตัม (Momentum Features): คำนวณผลรวมของ `log-return` ย้อนหลัง 5, 10, และ 20 วัน (`mom_5`, `mom_10`, `mom_20`)
  - ฟีเจอร์เชิงเวลาแบบวงกลม (Cyclical Time Features): แปลงวันในสัปดาห์ (`dayofweek`) ให้เป็นฟีเจอร์แบบ Sin/Cos (`dow_sin`, `dow_cos`) เพื่อให้โมเดลสามารถเรียนรู้ความสัมพันธ์ของเวลาในรูปแบบวงกลมได้

- การทำความสะอาดข้อมูลและการลบคอลัมน์ที่ไม่จำเป็น
  - การจัดการค่าสูญหาย (NaN): มีการเรียกใช้ `df.dropna(inplace=True)` ในขั้นตอนสุดท้ายของ `process_stock()` เพื่อลบเฉพาะที่มีค่าว่างทั้งหมดที่เกิดจากการคำนวนตัวชี้วัดทางเทคนิค
  - การคัดเลือกฟีเจอร์ (Feature Selection): มีการลบคอลัมน์ข้อมูลดิบที่ไม่คงที่ (`non-stationary`) เช่น Open, High, Low, Close, Volume และตัวชี้วัดที่ยังไม่ได้แปลงค่า (เช่น SMA\_5 ดั้งเดิม) ออกจาก DataFrame เพื่อให้โมเดลเรียนรู้เฉพาะฟีเจอร์ที่ผ่านการแปลงแล้ว (รวมทั้งหมด 37 ฟีเจอร์)
- การทำ Normalization และการสร้าง Sequence
  - Normalization: ใช้ฟังก์ชัน `dataframe_normalizer()` ในการปรับขนาดข้อมูล (Scaling) โดยใช้ `StandardScaler` กับทุกคอลัมน์ของข้อมูล เพื่อให้ฟีเจอร์มีค่าเฉลี่ยเป็น 0 และส่วนเบี่ยงเบนมาตรฐานเป็น 1
  - การสร้าง Sequence และการแบ่งชุดข้อมูล: ใช้ฟังก์ชัน `create_and_split_sequences()` เพื่อแบ่งข้อมูลที่ผ่านการ Normalization แล้ว ให้เป็นลำดับเวลา (sequence) โดยใช้ Lookback window 90 วัน (Input X) และ Forecast horizon 5 วัน (Target Y) จากนั้นจึงทำการแบ่งชุดข้อมูลออกเป็น Training set (80%), Validation set (10%), และ Test set ตามอัตราส่วนที่กำหนด และรวมชุด Training และ Validation จากทั้งหลายตัวเข้าเป็นชุดข้อมูลเดียวกันโดยใช้ `np.concatenate()`.

## 5.2 ส่วนการสร้างโมเดล (Model Definition)

- นิยามโมเดลเป็นคลาส `nn.Module` โดยใช้สถาปัตยกรรม Transformer Encoder สำหรับข้อมูลเชิงเวลาแบบหลายตัวแปร ภายในประกอบด้วย Multi-Head Attention, Feedforward Layer, LayerNorm, และ Dropout พร้อม Residual Connection และ Positional Encoding เพื่อเรียนรู้ลำดับเวลา ใช้ ReLU เป็น activation function ภายใน Feedforward Layer และ Linear head สำหรับคาดการณ์ 5 วันล่วงหน้า

## 5.3 ส่วนการฝึกสอนโมเดล (Model Training)

- ใช้ AdamW optimizer ร่วมกับ OneCycleLR scheduler และ SmoothL1Loss (Huber Loss) เพื่อลดผลกระทบจาก outliers เทคนิคด้วย Automatic Mixed Precision (AMP) และ Early Stopping เพื่อเพิ่มประสิทธิภาพการเรียนรู้ แบ่งข้อมูลเป็น train, validation และ test set ตามสัดส่วนที่เหมาะสม

## 6. วิธีการฝึกสอนและข้อมูลที่ใช้

### 6.1 Dataset

- แหล่งข้อมูล: Yahoo Finance API
- ช่วงเวลา: 1 มกราคม 2018 – ปัจจุบัน
- จำนวนหุ้นที่ใช้: 50 ตัว ในดัชนีหลักของสหรัฐอเมริกา (เช่น AAPL, MSFT, NVDA, TSLA, GOOGL, AMZN, JPM, V, ฯลฯ)
- ข้อมูลที่ใช้:
  - ราคาเปิด (Open), สูงสุด (High), ต่ำสุด (Low), ปิด (Close), ปริมาณการซื้อขาย (Volume)
  - 37 ตัวแปร (ประกอบด้วย *log-return*, *Volume features*, *RSI*, *MACD*, *Bollinger Band*, *Momentum* และ *Cyclical time features*)
- ระยะเวลาการย้อนดู (Lookback window): 90 วัน
- ระยะเวลาที่พยากรณ์ (Forecast horizon): 5 วัน

### 6.2 Hyperparameters

พารามิเตอร์	ค่า
Optimizer	AdamW
Learning Rate Scheduler	ReduceLROnPlateau (mode = 'min', factor = 0.5, patience = 10)
Learning Rate เริ่มต้น	$1 \times 10^{-4}$
Batch Size	128
Epochs	40

Loss Function	MSELoss (Mean Squared Error)
Device	GPU (Google Colab T4)

## 7. การประเมินผลและผลการทดลอง

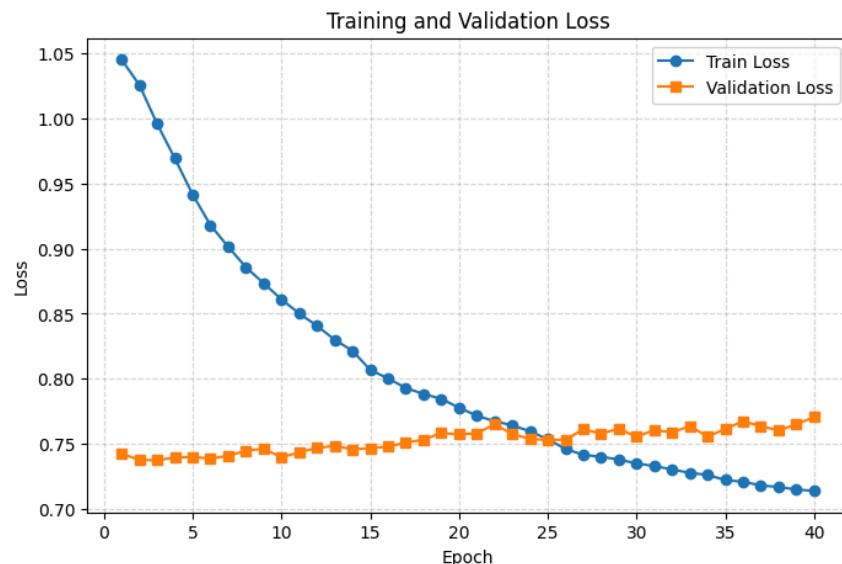
### 7.1 การประเมินด้วย Loss

จากการทดลอง training และ validation พบร่วมกันว่า

- Loss ของชุดฝึก (train) = 0.713939
- Loss ของชุดตรวจสอบ (validation) = 0.770610
- การที่ Train Loss ลดลง ในขณะที่ Val Loss เพิ่มขึ้น (divergence) เป็นสัญญาณคลาสสิกของ Overfitting เล็กน้อย

### 7.2 การประเมินด้วยกราฟ

- กราฟ Loss มีลักษณะราบและคงที่ แสดงถึงความเสถียรของการเรียนรู้



กราฟการพยากรณ์ 5 วัน พบร่วมกับแนวโน้มของผลลัพธ์ที่โมเดลทำนาย (predicted) สอดคล้องกับทิศทางเฉลี่ยของข้อมูลจริง (true) แม้ยังไม่สามารถจับความผันผวนสูงสุดได้ทั้งหมด

 Test MSE Results on reconstructed close prices

AAPL: 4.237372

MSFT: 9.118809

NVDA: 0.134818

TSLA: 3.412017

GOOGL: 5.768840

AMZN: 13.141598

META: 75.847274

NFLX: 270.511695

AMD: 3.010355

INTC: 9.682481

JPM: 11.496308

V: 14.369947

MA: 34.386661

ORCL: 9.631925

XOM: 5.043098

COST: 27.322198

WMT: 0.729547

PG: 2.607862

JNJ: 5.173348

ADBE: 84.588963

BAC: 0.925882

KO: 0.507113

PEP: 5.126472

CSCO: 1.338298

CRM: 26.151292

T: 0.161673

MCD: 8.954828

HON: 12.531485

IBM: 14.522351

QCOM: 5.385689

UPS: 10.960292

PM: 3.650359

CVX: 10.925041

BA: 226.507680

SBUX: 3.577445

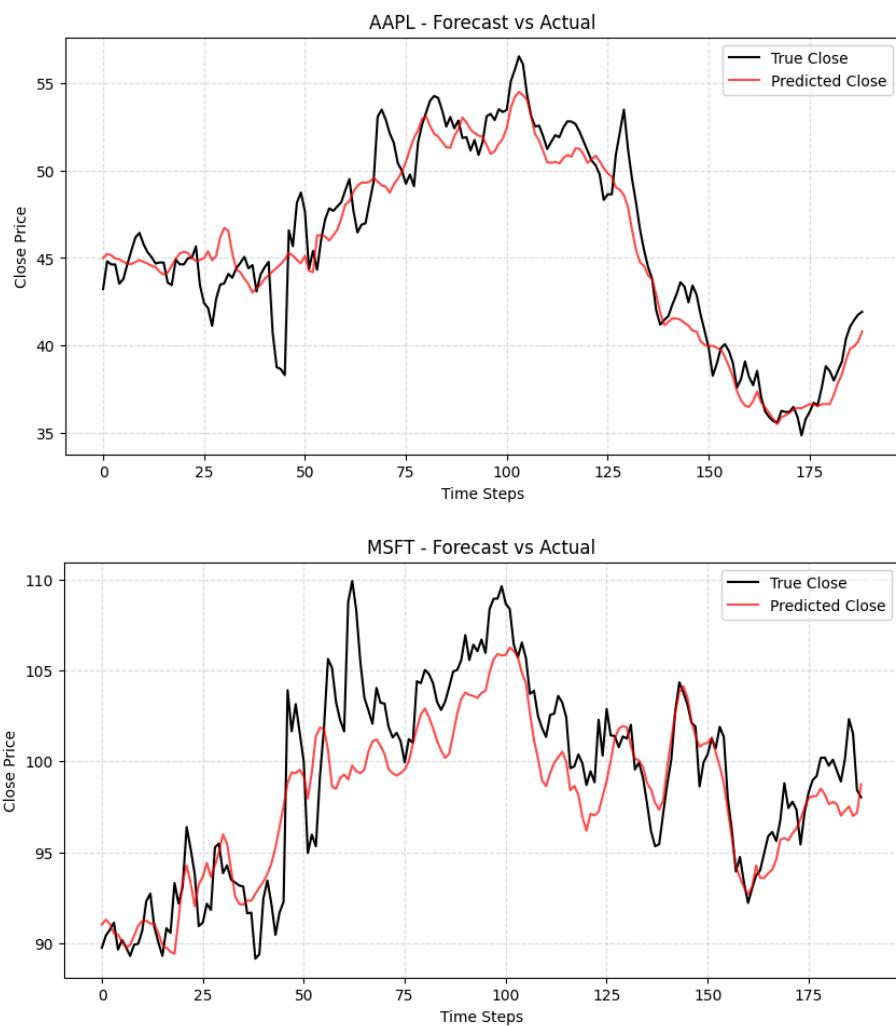
GS: 73.904233

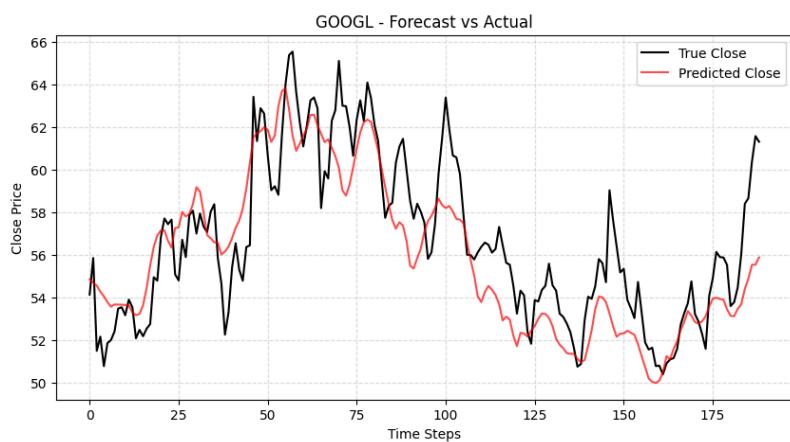
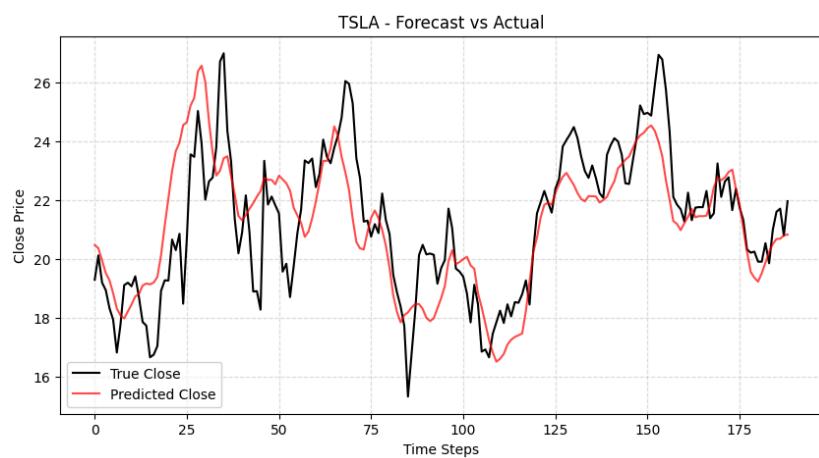
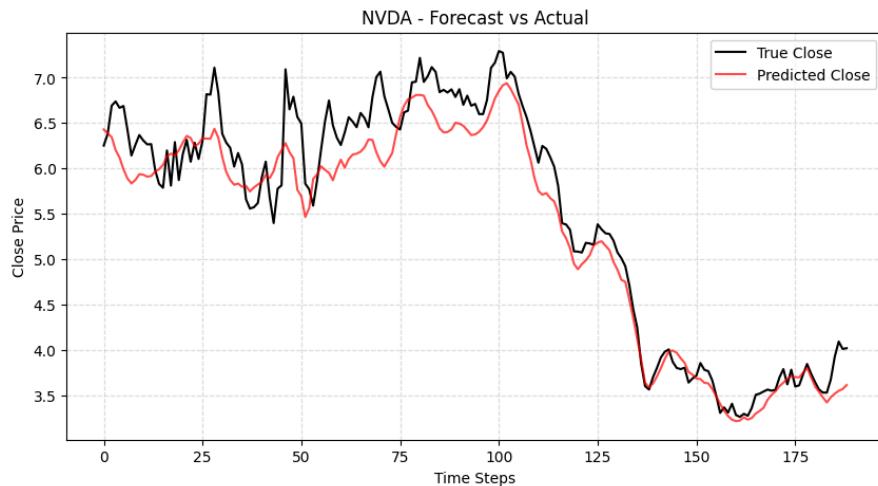
CAT: 20.366396

AXP: 17.222128

MDT: 3.861965

AMGN: 27.982163  
BLK: 202.867997  
SPGI: 38.964114  
TXN: 18.354885  
NEE: 1.402991  
MMM: 21.602198  
UNH: 218.624503  
LIN: 12.012116  
RTX: 6.870896  
LOW: 7.523004  
BKNG: 5174.270416  
Overall: 135.345420





### 7.3 Metric

ใช้ Mean Absolute Error (MSE) เป็น Metric หลัก ในการประเมินประสิทธิภาพบนชุดทดสอบ โดยมีการสร้างราคาปิด (Close Price) ย้อนกลับจากค่า *log-return* ที่ทำนายไว้ (Inverse Transform) เพื่อคำนวณความคลาดเคลื่อนระหว่างราคาปิดจริงกับราคาปิดที่ทำนาย

## 8. สรุปผลการดำเนินงาน

โครงการนี้ประสบความสำเร็จในการสร้างระบบพยากรณ์ราคาหุ้นที่สามารถรับข้อมูลจากหุ้นหลายตัวและหลายตัวแปรพร้อมกัน โดยใช้โมเดล Deep Learning สถาปัตยกรรม Transformer ซึ่งมีประสิทธิภาพสูงในการเรียนรู้ข้อมูลเชิงเวลา

ผลการทดลองแสดงให้เห็นว่าโมเดลมีความเสถียร และสามารถจับแนวโน้มราคาหุ้นในระยะสั้นได้อย่างแม่นยำ ซึ่งสามารถนำไปต่อยอดสู่ระบบวิเคราะห์ทางการเงินอัตโนมัติได้ในอนาคต

## 9. งานที่เกี่ยวข้องและเอกสารอ้างอิง

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)* (pp. 6000–6010).
2. Fischer, T., & Krauss, C. (2018). *Deep learning with long short-term memory networks for financial market predictions*. *European Journal of Operational Research*, 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
3. Zhang, X., Li, Y., & Pan, L. (2020). *Stock movement prediction with spatial-temporal graph transformer*. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020)*, 2825–2832. <https://doi.org/10.1145/3340531.3412109>
4. Yahoo Inc. (n.d.). *Yahoo Finance API documentation*. Retrieved from <https://finance.yahoo.com>
5. Pandas-TA. (n.d.). *Technical Analysis Library for Python*. Retrieved from <https://github.com/twopirllc/pandas-ta>

## 10. การแบ่งงานและสัดส่วนของผู้จัดทำ

ลำดับ	ชื่อผู้จัดทำ	หน้าที่ที่รับผิดชอบ	สัดส่วนของงาน ทั้งหมด
1	ก้องภพ ไพราระ	<ul style="list-style-type: none"> <li>- จัดเตรียมและประมวลผลข้อมูล (Preprocessing)</li> <li>- สร้างและคำนวณตัวชี้วัดทางเทคนิค (Technical Indicators)</li> <li>- จัดรูปแบบข้อมูลให้อยู่ในโครงสร้างที่เหมาะสมต่อการนำเข้าโมเดล (Formatting Data)</li> </ul>	50%
2	ปฐมพงศ์ บวรเจริญพันธุ์	<ul style="list-style-type: none"> <li>- ออกรูปแบบและพัฒนาโมเดล Deep Learning (Transformer)</li> <li>- เขียนส่วนการฝึกสอนโมเดล (Training Loop) และปรับพารามิเตอร์</li> <li>- พัฒนาและทดสอบส่วนการอนุมานผล (Inference / Evaluation)</li> </ul>	50%