การหาแนวโน้มของตลาดหุ้น S&P 500

ปริญญา อบอุ่น
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

บทคัดย่อ

โครงงานนี้นำเสนอระบบหาแนวโน้มของตลาดหุ้นดัชนี S&P 500 โดยใช้เทคนิคการประมวลผลสัญญาณดิจิทัล ได้แก่ เส้นค่าเฉลี่ยเคลื่อนที่ (Simple Moving Average: SMA) เส้นค่าเฉลี่ยเคลื่อนที่แบบเอ็กซ์โพเนนเชียล (Exponential Moving Average: EMA) และโมเดล Auto-Regressive Integrated Moving Average (ARIMA) สำหรับการพยากรณ์ทิศทางตลาด การศึกษานี้ใช้ข้อมูลราคาปิดรายวันของดัชนี S&P 500 ตั้งแต่ปี 2015-2025 จำนวน 2,516 จุด โดยข้อมูลผ่านการทดสอบคุณสมบัติ Stationarity ด้วย Augmented Dickey-Fuller (ADF) และ Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test และหาค่าพารามิเตอร์ที่เหมาะสมด้วยวิธี Grid Search จำนวน 1,000,000 combinations (1000 × 1000)

จากการทดลองพบว่าวิธี EMA Crossover ที่ใช้ EMA(101) ตัดกับ EMA(465) ให้ประสิทธิภาพสูงสุด โดยมีค่า Accuracy 55.68% และ Precision 55.70% รองลงมาคือการใช้ EMA(420) ตัดกับ SMA(484) ที่ให้ค่า Accuracy 55.66% และ Precision 56.19% ส่วนโมเดล ARIMA(2,0,2) ให้ค่า Accuracy 53.56% และ Precision 55.64% ทั้งสามวิธีมีประสิทธิภาพสูงกว่าการทำนายแบบสุ่มอย่างมีนัยสำคัญ (ประมาณ 3.5-5.7% สูงกว่า baseline 50%) ผลการศึกษานี้แสดงให้เห็นถึงความเป็นไปได้ในการประยุกต์ใช้หลักการทางการประมวลผลสัญญาณดิจิทัลเพื่อสร้าง Technical Indicator ที่มีประสิทธิภาพในการวิเคราะห์ตลาดหุ้น

Abstract

This project presents a trend analysis system for the S&P 500 stock market index using digital signal processing techniques, including Simple Moving Average (SMA), Exponential Moving Average (EMA), and Auto-Regressive Integrated Moving Average (ARIMA) model for market direction forecasting. The study utilizes daily closing price data of the S&P 500 index from 2015 to 2025, comprising 2,516 data points. The data underwent stationarity testing using Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests, with optimal parameters determined through Grid Search of 1,000,000 combinations (1000 × 1000).

Experimental results demonstrate that the EMA Crossover strategy using EMA(101) × EMA(465) achieves the highest performance with 55.68% accuracy and 55.70% precision, followed by EMA(420) × SMA(484) with 55.66% accuracy and 56.19% precision. The ARIMA(2,0,2) model yields 53.56% accuracy and 55.64% precision. All three methods significantly outperform random prediction (approximately 3.5-5.7% above the 50% baseline), demonstrating the feasibility of applying digital signal processing principles to develop effective technical indicators for stock market analysis.

1 บทนำ

1.1 ที่มาและความสำคัญ

ในสถานการณ์ปัจจุบันโลกนั้นได้เจอกับการเปลี่ยนแปลงโดยการมีมาตราการจากนโยบายการเงินที่ผ่อนคลายเชิงระบบ (Quantitative Easing) ของธนาคารกลางทั่วโลกทำให้เกิดการขยายตัวของปริมาณเงินในระบบหรือที่เรียกว่าเงินเฟ้อ โดยข้อมูลจาก Federal Reserve แสดงให้เห็นว่าปริมาณเงิน M2 เพิ่มขึ้นอย่างมากมาย

ดังนั้นถ้าอยากให้ความสามารถในการจับจ่ายใช้สอยได้เท่าเดิมเราต้องเอาเงินของเราไปลงทุนเพื่อที่จะทำให้เงินของเรางอกเงย

โครงงานชิ้นนี้จะพัฒนาและวิเคราะห์โดยใช้ความรู้ทางด้าน Digital signal processing เพื่อสร้างตัวชี้วัด (indicator) ที่สามารถตรวจวัดการเปลี่ยนแปลงของตลาด

1.2 วัตถุประสงค์

1. พัฒนาระบบวิเคราะห์แนวโน้มของตลาดหุ้น S&P 500 โดยใช้เทคนิค Moving Average Crossover และ statistical model (ARIMA)
2. สร้าง Technical Indicator ที่มีความแม่นยำมากกว่าการสุ่ม (50%) โดยต้องมากกว่าแบบมีนัยสำคัญ
3. วิเคราะห์คุณสมบัติของ stationarity ของข้อมูล S&P 500 index ด้วย ADF และ KPSS test

1.3 ขอบเขต

โครงงานนี้มีขอบเขตดังต่อไปนี้ :

1. ขอบเขตด้านปฏิบัติ

- ใช้ข้อมูลราคาปิดของกราฟรายวัน (daily) ของ S&P 500 index (^GSPC) จาก Yahoo Finance
- ช่วงเวลา: 1 มกราคม 2015 ถึง 1 มกราคม 2025 (เป็นระยะเวลา 10 ปี)
- จำนวนจุดข้อมูล: 2,516 observations
- ประเภทข้อมูล: ราคาปิด (close price) ในรูปแบบ time series ที่สัญญานเป็น non-stationary

2. ขอบเขตด้านวิธีการ

- Technical analysis: ศึกษาแค่ Moving average indicator (SMA และ EMA)
- Optimization: Grid Search 1000x1000 (1,000,000 combination)
- Evaluation: เน้นความแม่นยำ accuracy ไม่รวม profit/loss

3. ขอบเขตด้านการทดสอบ

- ทดสอบประสิทธิภาพด้วย rolling window prediction (one-step ahead)
- ใช้ entire dataset ไม่มีการแบ่ง train / test

2 ทฤษฎีที่นำเสนอ

2.1 ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

2.1.1 เส้นค่าเฉลี่ยเคลื่อนที่ (Moving average)

นิยาม Moving average (MA) เป็นตัวกรองแบบ Finite Impulse Response (FIR) ที่ใช้ในทางด้าน Digital Signal Processing โดยตัว Moving Average (MA) ทำหน้าที่เป็น Low-pass Filter เพื่อลดความถี่สูงของสัญญานรบกวน (high-frequency noise) และทำให้เห็นแนวโน้มพื้นฐาน (underlying trend) ของอนุกรมเวลา
โดยทำการเฉลี่ยค่าล่าสุดในหน้าต่างขนาด n

$$\mathrm{MA}_w(t) = \left(\frac{1}{w}\right) \sum_{i=t-w+1}^{t} x_i$$

โดยที่:

- $\mathrm{MA}_w(t)$ คือ ค่าเฉลี่ยเคลื่อนที่ (Moving average) ณ เวลาที่ด้วยขนาดหน้าต่าง w
- w คือ ขนาดหน้าต่าง
- x_i คือค่าที่เวลา i
- t คือ เวลาปัจจุบัน

คุณสมบัติทางคณิตศาสตร์ (Mathematical Properties):

1. Linear Time-Invariant(LTI) System: MA เป็นระบบ LTI ที่มีคุณสมบัติ causality และ stability
2. Phase Delay: ทำให้เกิด phase delay เท่ากับ $\frac{w-1}{2}$ samples
3. Smoothing Factor: ระดับการปรับเรียบผกผันกับ w

Implement: ใน rust ระบบใช้ module sma.rs ที่ implement โดยใช้ O(n) สำหรับการคำนวณ

การใช้ MA ช่วยลดความผันผวนของสัญญานและช่วยให้มองเห็นแนวโน้มที่ชัดเจนยิ่งขึ้นแม้ว่าหุ้นนั้นจะแกว่งมากก็ตาม

2.1.2 เส้นค่าเฉลี่ยเคลื่อนที่แบบเอ็กซ์โพเนนเชียล (Exponential moving average)

นิยาม Exponential Moving Average (EMA) เป็นตัวกรองแบบ Infinite Impulse Response (IIR) ที่ให้น้ำหนักแบบ exponential decaying กับข้อมูลในอดีต ทำให้มีการตอบสนอง (response) ที่รวดเร็วกว่า MA

$$\mathrm{EMA}_w(t) = \alpha x_t + \mathrm{EMA}_{\text{t-1}}(1-\alpha)$$

โดยที่:

- $\alpha = \frac{S}{1+\ \mathrm{Days}}$ = smoothing factor
- S = smoothing constant (ใช้ค่า 2 ในโปรเจคนี้)
- Days = จำนวน periods (windows)
- x_t = ค่าปัจจุบัน
- EMA_{t-1} = ค่า EMA ก่อนหน้า

คุณสมบัติทางศาสตร์ (Mathematical Properties):

1. Memory Effect: EMA มี infinite memory แต่ให้น้ำหนักลดแบบ exponential: $w_k = \alpha(1-\alpha)^k$
2. Effective WIndow: หน้าต่างทำให้เกิด phase delay เท่ากับ $\frac{2}{\alpha} - 1$ periods
3. Lag Reduction: มี lag น้อยกว่า SMA เท่ากัย $\frac{w-1}{2}$ periods

Implementation: ema.rs ใช้ interative approach ด้วย time complexity O(n) และ space complexity O(1)

จะเป็นสูตรที่ตอบสนองต่อการเปลี่ยนแปลงของข้อมูลได้เร็วกว่าจะช่วยบอก trend (แนวโน้ม) ของตลาดได้เร็วกว่าแบบเคือ MA

2.1.3 Auto-Regressive Integrated Moving Average

ARIMA คือโมเดลสำหรับวิเคราะห์อนุกรมเวลา โดยรวมองค์ประกอบของการถดถอยอัตโนมัติ (Auto-Regressive) และค่าเฉลี่ยเคลื่อนที่ (Mean Average) เข้าด้วยกัน พร้อมกับการทำให้ข้อมูลเป็นสถิติก่อนด้วยการหาค่าต่าง (differencing) จำนวน d ครั้ง โมเดล ARIMA สามารถเขียนเป็น:

$$Y_t = c + \sum_{i=1}^{p} \Phi_i Y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \Theta_j \varepsilon_{t-j}$$

เลือก parameter โดยการ โดยที่:

- Y_t = ค่าที่สังเกต ณ เวลา t (หลัง differencing)
- c = ค่าคงที่ (drift term)
- Φ_i = สัมประสิทธ์ Auto-Regressive(AR) ที่ lag i ($|\Phi_i| < 1$ สำหรับ stationarity)
- Θ_j = สัมประสิทธ์ Moving-average (MA) ที่ lag j ($|\Theta_j|$ < 1 สำหรับ invertibility)
- ε_t = white noise error term $\sim N(0, \sigma^2)$
- L = lag Operator ($L^k \cdot Y_t = Y_{t-k}$)

การระบุ Parameters (model Identification):

1. Differencing Order (d):
   - ใช้ Augmented Dickey-Fuller (ADF) test
   - KPSS test สำหรับ trend stationarity

2. AR Order (p) ผ่าน PACF:
   - ใช้ OLS regression
   - Cut-off criterion: $|\mathrm{PACF}(K)| < \frac{1.96}{\sqrt{n}}$
3. MA Order (q) ผ่าน ACF:
   - ใช้ FFT-base Compute สำหรับงาน efficiency
   - Autcorrelation function ที่ lag k

2.1.4 การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform)

เป็น algorithm ที่ optimal ที่สุดในเวลานี้สำหรับการคำนวณ Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} \left(x[n] * e^{\frac{-j2\pi kn}{N}} \right)$$

การนำมาประยุกต์:

1. Spectral Density Estimation
2. ACF via Wiener–Khinchin Theorem
3. Periodicity Detection

2.2 การประเมินประสิทธิภาพของระบบ

1. Accuracy (ความถูกต้องโดยรวม):
   $$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{Total}}$$
2. Precision (ความแม่นยำ):
   $$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

2.3 วิธีการแก้ปัญหา

2.3.1 โครงสร้างระบบ (System Architecture)

```
src/module/
├── data
├── eval.rs
├── indicator
├── model
├── mod.rs
├── plot
├── util
└── workflow.rs
```

Design Principles:

1. ประสิทธิภาพของระบบ (Performance)
2. ความแม่นยำ (Accuracy)
3. ความทนทาน (Robustness)
4. การทำซ้ำได้ (Reproducibility):

2.3.1 Algorithm Implementation Details

EMA Crossover Strategy:

- Fast Ema (period สั้น) ตัดบน Slow Ema (period ยาว) = Buy Signal
- Zero-padding สำหรับ boundary conditions
- Time Complexity: O(n) per prediction

ARIMA (Auto-Regressive integrate moving average):

1. Stationnary testing -> ADF/KPSS test
2. Parameter Selection -> PADF/ACF analysis
3. Model fitting -> CSS optimization
4. Rolling Prediction -> One-step ahead forecasts
5. Performance Evaluation -> Directional accuracy

3 ผลการทดลอง

การทดสอบใช้ข้อมูลของหุ้น S&P 500 Index (^GSPC) จาก Yahoo Finance ระหว่างวันที่ 1 มกราคม 2015 ถึง 1 มกราคม 2025 ทั้งหมดคือ 2,516 จุดรายวัน (daily) โดยใช้ตัวแปรคือราคาปิดของวันนั้น

ข้อมูลทั้งหมดที่เป็นความต่างของข้อมูลวันนี้และเมื่อวาน (log return) จะถูกนำไปใช้ใน ARIMA เพื่อการพยากรณ์ว่าจะขึ้นหรือลงวันพรุ่งนี้ที่ (T+1) โดยต้องนำไปทดสอบเพื่อหา parameter คือ ค่า d ว่าเท่ากับเท่าไร ถ้าเป็น stationary จะให้ค่า d = 0 ถ้าไม่ใช่ stationary จะให้ค่า d = 1

ผลการทดสอบ Stationarity (Diff)

- ADF Test (Diff): $t = -41.160 < -3.460$ (ค่าวิกฤต 1%) $\Longrightarrow$ **Stationary**
- KPSS Test (Diff): $0.175 < 0.739$ (ค่าวิกฤต 1%) $\Longrightarrow$ **Level-stationary**

สรุป:ผลการทดสอบ ADF แสดงค่า t-statistic ที่ต่ำกว่าค่าวิกฤตอย่างมีนัยสำคัญ ($p < 0.01$) ทำให้ปฏิเสธสมมติฐานหลักของ unit root ได้ ขณะที่ KPSS test ไม่สามารถปฏิเสธสมมติฐานหลักของ stationarity ได้ แสดงว่าข้อมูลมีคุณสมบัติ stationary เหมาะสมสำหรับการใช้งาน

จากผลดังตารางเราจะให้ค่า d = 0 เป็น stationary จะเรียกว่า ARMA(p,q)

การเลือกพารามิเตอร์ของโมเดล

การวิเคราะห์ Partial Autocorrelation Function (PACF) ด้วยวิธี Levinson-Durbin และ OLS regression ให้ผลสอดคล้องกันที่ $p = 1$ โดยใช้เกณฑ์ cutoff ที่ระดับความเชื่อมั่น 95% ($\pm\frac{1.96}{\sqrt{n}} = \pm 0.0391$)

การหาค่า MA Order (q) ด้วย ACF

การวิเคราะห์ Autocorrelation Function ด้วย FFT-based algorithm แสดงผลดังนี้:

- First-drop criterion: $q = 2$
- Largest significant lag: $q = 10$

จะพิจารณาเลือกใช้ first drop

3.4.1 Grid Search Optimization

จะทำการค้นหาค่าที่ดีที่สุด (accuracy) สูงสุดด้วยการทดสอบ 1000 x 1000 ของ ema/sma period ได้ผลดังนี้

ผลลัพธ์สรุป (Top)

- **Best: EMA(101) × EMA(465)** — Accuracy **55.68%**, Precision **55.70%**.
- **รองลงมา: EMA(420) × SMA(484)** — Accuracy **55.66%**, Precision **56.19%**
- **อ้างอิงเชิงสถิติ: ARIMA(2,0,2)** — Accuracy **53.56%**, Precision **55.64%**.

4 การวิเคราะห์และสรุปผล

1. ประสิทธิภาพของ technical indicator: ด้วยใช้วิธี EMA crossover จะได้ประสิทธิภาพที่ EMA(101) x EMA(465) ได้ผลลัพธ์ที่ดีที่สุดโดยวัดจากความแม่นยำ accuracy คือ 55.68% และ precision ที่ 55.70%
2. ข้อจำกัดของ ARIMA: ตลาดหุ้นเป็น non linear และ deterministic ทำให้ ARIMA ไม่สามารถประมาณค่าได้แม่นยำมาก โดยในการทดลองนี้ใกล้เคียงกับ moving average crossover

โครงงานนี้แสดงให้เห็นการประยุกต์โดยใช้เทคนิคทาง digital signal processing เพื่อสร้าง trend indicator ที่มีประสิทธิภาพเหนือกว่าการสุ่มแบบโยนเหรียญอย่างมีนัยสำคัญที่ประมาณ 6%

5 กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอกราบขอบพระคุณ อาจารย์ ดร.กิตติผล โหราพงษ์ ที่ให้คำแนะนำ คำปรึกษา และข้อเสนอแนะอันมีค่าตลอดเวลาการทำโครงงาน

และขอขอบคุณเพื่อนๆที่คอยให้กำลังใจและสนับสนุน

หากโครงงานฉบับนี้มีข้อบกพร่องประการใด ผู้จัดทำขออภัยมา ณ ที่นี้ และยินดีรับฟังข้อเสนอแนะเพื่อไปปรับปรุงแก้ไขต่อไป

6 เอกสารอ้างอิง

[1] Algorithmic-Oriented Digital Signal Processing for Computer Engineers 01204496, “บท 5: การประมวลผลสัญญาณปรับได้

(Adaptive Signal Processing)," lecture notes, [in Thai].

[2] Anthropic, "Claude AI [Large language model]," Sep. 2025. [Online]. Available: https://www.anthropic.com/claude

[3] P. S. R. Diniz, Signal Processing and Machine Learning Theory. Academic Press, 2024.

[4] G. Gundersen, "Returns and Log Returns," Sep. 2022. [Online]. Available: https://gregorygundersen.com/blog/2022/02/06/log-returns/

[5] OpenAI, "ChatGPT [Large language model]," Sep. 2025. [Online]. Available: https://chatgpt.com

[6] S. Prabhakaran, "Time Series Analysis in Python – A Comprehensive Guide with Examples," MachineLearningPlus (ML+), Feb. 13, 2019. [Online]. Available: https://www.machinelearningplus.com/time-series/time-series-analysis-python/

[7] The Rust Project Developers, The Rustdoc Book. [Online]. Available: https://doc.rust-lang.org/rustdoc/. [Accessed: Sep. 2025].

[8] Machine Learning Plus, "ARIMA Model — Time Series Forecasting in Python," 2025. [Online]. Available: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Accessed: Sep. 2025].

[9] Federal Reserve Bank of St. Louis, "M2 Money Stock (WM2NS)," FRED Economic Data. [Online]. Available: https://fred.stlouisfed.org/series/WM2NS. [Accessed: Sep. 2025].

[10] Wikipedia, "Autoregressive integrated moving average," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average. [Accessed: Sep. 2025].

[11] Wikipedia, "Autoregressive moving-average model," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Autoregressive_moving-average_model. [Accessed: Sep. 2025].

[12] GeeksforGeeks, "ARMA Time Series Model," 2025. [Online]. Available: https://www.geeksforgeeks.org/data-science/arma-time-series-model/. [Accessed: Sep. 2025].

[13] M. Halls-Moore, "Autoregressive Moving Average (ARMA) p, q Models for Time Series Analysis - Part 1," QuantStart. [Online]. Available: https://www.quantstart.com/articles/Autoregressive-Moving-Average-ARMA-p-q-Models-for-Time-Series-Analysis-Part-1/. [Accessed: Sep. 2025].