

1) Bối cảnh, mục tiêu & tiêu chuẩn khoa học

Bối cảnh Việt Nam: Tính đến giữa 2025, ~16% dân số tham gia chứng khoán. Nhà đầu tư cá nhân nhiều, nhưng thiên về cảm xúc, dẫn tới hiệu suất kém bền vững. Dự án đặt mục tiêu xây dựng một nền tảng “**đầu tư khoa học**” cho newbie: hiểu rõ thứ mình mua, quản trị tiền và rủi ro, hướng tới lợi nhuận ổn định.

Mục tiêu kỹ thuật:

- Dự báo **xác suất Long trong 20 phiên tới** cho từng mã dựa trên **cửa sổ 60 phiên** (ban đầu 90 phiên, sau tối ưu về 60).
- Không chỉ dự báo đơn mã, mà **nổi tới danh mục**: chọn top-k mã, rồi **phân bổ vốn tối ưu rủi ro** bằng Monte Carlo 3-stage.
- Tránh look-ahead** và các dạng rò rỉ nhãn (label leakage) tinh vi trong dữ liệu tài chính.
- Backtest trung thực**: tái cân bằng mỗi 20 phiên, phí giao dịch, TP/SL theo từng mã, và chuẩn mực CRN (Common Random Numbers) trong mô phỏng.

Tiêu chuẩn khoa học:

- Chia tập **theo thời gian** (2015–2021 train, 2022 val, 2023–2024 test), **embargo** theo horizon để tách nhãn không chồng lấn.
- Thống kê, trực quan, kiểm tra dị thường (giá = 0, return vô hạn, cú sốc >30–40%).
- Báo cáo **ROC-AUC, PR-AUC, Brier score** (xác suất), và **CAGR, Sharpe, MDD** (danh mục).

Kết quả cuối cùng (out-sample) sau tối ưu: **CAGR≈28.8% | Vol≈21.9% | Sharpe≈1.32 | MDD≈-14% | Win rate≈61% | 131 giao dịch**. Đây là mức hiệu năng **đáng tin cậy** khi đã tính phí, TP/SL, và mô phỏng 10.000 kịch bản Monte Carlo.

2) Dữ liệu & tiền xử lý: làm sạch kỹ để tránh “rác vào–rác ra”

Nguồn & phạm vi

- 24 mã HOSE thanh khoản cao (ban đầu 25, loại VHM vì nhiều điểm dị thường).
- Khoảng thời gian **2015-02-09** → **2024-12-03** (sau lọc sạch).
- Khoảng **55k dòng gốc** → còn **~53k dòng sạch** sau loại NaN/inf, close=0, & outlier cực đoan.

Các trường sử dụng

- Giá/khối lượng & chỉ báo kỹ thuật:
 - `close`, `volume`;
 - `r1_calc` (log return ngày), `rsi14_calc`, `vol_rel20_calc` (khối lượng tương đối 20), `volat20_calc` (σ rolling 20);
 - `volume_log1p` (ổn định phân phối).
- **Vĩ mô**: `interbank_week` (đầu tuần), `cpi_yoy` (tháng), `usd_vnd` (tháng).
- **Sentiment**: `sentiment_value` $\in \{-1, 0, 1\}$ (PhoBERT fine-tune trên tiêu đề CafeF; lan 3 phiên sau tin).
- **Market regime**: biến 0/1 từ EMA-60 của log return tổng hợp để báo bull/bear nhẹ (để ổn định mô hình).

Làm sạch & kiểm soát dị thường

- **close=0** (ví dụ POW/VIB 2018-07-23) → loại; return vô hạn (`inf`, `-1` với simple return) → loại.
- Kiểm tra $|\text{return}| > 20\text{--}30\text{--}40\%$ theo ticker; kết luận: phần lớn do ngày lỗi/điều chỉnh → loại ở mức vừa đủ để giữ cấu trúc thật (không “sanitize” quá tay).
- Sau khi xóa dòng dị thường, **tính lại toàn bộ chỉ báo kỹ thuật** để không lệ thuộc vào chuỗi đã hỏng → tránh trôi bias.

Nhận xét & quyết định

- Tránh dùng nhãn/logics phụ thuộc các dòng đã xoá mà **không cập nhật lại đặc trưng** (dẫn leakage vi tế).
 - Mặc dù sentiment có thiên lệch dương (nhiều 1), ta **không sửa tay**; chỉ dùng như nguồn thông tin ngắn hạn đúng như hành vi thị trường.
-

3) Gán nhãn (labeling) chuẩn xác & chống look-ahead

Định nghĩa nhãn

- Tại thời điểm t , $y(t)=1$ nếu **tổng log return** từ $t+1$ đến $t+20 > 0$, ngược lại **0**.
- Dùng **log return** cho nhãn (tính chất cộng), nhưng vẫn giữ **simple return** cho một số kiểm tra chéo.
- **Embargo/Purge**: khi split theo mốc $2021 \rightarrow 2022 \rightarrow 2023$, ta chèn khoảng trống $\geq \text{Horizon}$ (20 phiên) để nhãn train không “đụng” với val/test.

Kiểm tra phân phối nhãn

- Tỷ lệ $\sim 54/46$ (1/0) tổng thể; riêng năm 2022 **bearish** (val), tỷ lệ 1 thấp hơn rõ.
 - Nhận xét: **khác biệt phân phối theo năm** là đúng “regime shift”; mô hình phải “tổng quát hoá” qua chế độ này.
-

4) Chia tập, chuẩn hoá & dựng chuỗi theo ticker–time

Chia tập theo **mốc thời gian** (không theo thứ tự dòng, vì dữ liệu sắp theo ticker):

- Train: 2015–2021
- Val: 2022

- Test: 2023–2024

Embargo: chứa 20 phiên giữa các mốc để triệt tiêu chồng lấn nhãn.

Chuẩn hoá

- **StandardScaler** fit trên **train** (theo **từng feature**, không trộn train/val/test).
- Không leak: biến đổi được áp dụng “out-of-sample” cho val/test.
- Thêm **volume_log1p** để ổn định. Macro/sentiment để nguyên (mức thay đổi thừa), vẫn đưa vào chuẩn hoá cùng pipeline.

Dựng chuỗi input

- Cửa sổ **W=60** time steps \times **F** features (kỹ thuật + macro + sentiment + regime).
- **Slide 1 ngày** \rightarrow Các sample **chồng lấn** mạnh (89/90 trước đây; rút về 59/60); ta **đánh giá theo mốc ngày (không shuffle)** và dùng **embargo** để hạn chế “over-optimism”.

Nhận xét:

- Sau khi kiểm tra lại số lượng ticker trong mỗi split, xác nhận **mỗi split đều có đủ 24 ticker**.
- Thống kê nhãn theo ticker tại val cho thấy vài mã (VIC, VIB, NVL...) có tỷ lệ 1 thấp (thị trường giảm). Điều này **giải thích vì sao độ khó val cao**.

5) Khảo sát mô hình: LSTM \rightarrow GRU \rightarrow Transformer (và vì sao chọn Transformer)

LSTM/GRU baseline

- LSTM 1-2 tầng, dropout, early stopping.

- Kết quả: ROC-AUC $\sim 0.60\text{--}0.62$ (val), test dao động xuống $\sim 0.48\text{--}0.56$; Brier $\sim 0.24\text{--}0.25$.
- GRU tương tự LSTM, đôi khi val nhỉnh hơn nhưng test thiếu ổn định (đặc biệt khi regime thay đổi).

Transformer Encoder

- **Positional Encoding** (sin-cos), 2–3 **encoder blocks** với Multi-Head Attention (num_heads=4), FFN (128–256), dropout 0.1–0.2.
- Lợi điểm:
 - **Tự chú ý (self-attention)** học “mối quan hệ” giữa các thời điểm trong cửa sổ 60 phiên **không bị ràng buộc tuần tự** như RNN.
 - **Tổng quát hoá better** khi kết hợp **macro** (thay đổi chậm) + **sentiment** (ngắn hạn) + **technical** (dao động liên tục).
 - Xử lý **tương tác không tuyến tính** giữa các nhóm tín hiệu.

Tiến trình tăng dần

- **+ Market regime** → val AUC cải thiện nhẹ, test ổn định hơn (giảm overreaction trong bear).
- **+ Macro (interbank, CPI YoY, USD/VND)** → val AUC tăng rõ (0.70+), test tăng vừa phải (0.56+).
- **+ Sentiment** (PhoBERT, lan 3 phiên) → val AUC ~ 0.76 , PR-AUC ~ 0.69 ; test PR-AUC ~ 0.63 , ROC-AUC $\sim 0.56\text{--}0.63$ tùy W.
- **Giảm W từ 90→60**: test ROC-AUC **tăng tới $\sim 0.63\text{--}0.65$** , Brier giảm. Giải thích: ít chông lún hơn, tập trung đúng “hơi thở” 1–3 tháng.

Kết luận mô hình

- **Transformer + Tech + Macro + Sentiment, W=60** cho khả năng phân tách xác suất tốt nhất và ổn định nhất.
 - Ta lưu file chuẩn: `transformer_tech_macro_sent_w60_deep.keras`.
-

6) Đánh giá xác suất & calibration

Dự báo của mô hình là $\hat{p}_{\text{long}}(t)$. Ta kiểm tra:

- **ROC-AUC, PR-AUC**: phân biệt tốt/không tốt.
- **Brier score**: chất lượng xác suất (càng thấp càng tốt).
- **Độ ổn định threshold**: sử dụng **val** để chọn ngưỡng $\hat{p} \geq 0.55 \rightarrow$ chọn danh mục.

Gợi ý tương lai: **Isotonic/Platt calibration** để \hat{p}_{long} “sát biên” hơn, cải thiện cả thresholding lẫn “tilt” bước Monte Carlo.

7) Từ xác suất \rightarrow danh mục: hai con đường & vì sao chọn Monte Carlo 3-stage

Con đường A (đơn giản): Top-k theo \hat{p}_{long} , **equal weight**.

- Ưu điểm: dễ triển khai, turnover thấp.
- Nhược điểm: **không nắm bắt rủi ro chéo** (tương quan nhóm ngành, biến động từng mã) \Rightarrow hiệu suất chưa tối ưu.

Con đường B (đề xuất): Top-k theo $\hat{p}_{\text{long}} \rightarrow$ **tối ưu trọng số bằng Monte Carlo 3-stage**.

- Giữ được **động lực ngắn hạn** (block bootstrap) và **tương quan thực** giữa mã.
 - **Tilt** kích bản dựa trên \hat{p}_{long} (nhưng chỉ vừa phải) để **không overfit** theo dự báo.
 - Áp dụng **CRN** (Common Random Numbers) trong tối ưu để so sánh công bằng giữa các bộ trọng số, bề mặt mục tiêu “mượt” hơn.
-

8) Monte Carlo 3-stage – Thiết kế chi tiết (điểm nhấn quan trọng)

Đầu vào mỗi kỳ tái cân bằng (mốc d_0):

- Lấy **Top-k** (mặc định $k=7$) mã có $\hat{p}_{\text{long}}(d_0)$ cao nhất, và **vượt ngưỡng** (ví dụ 0.55).
- Trích **ma trận log return $60 \times k$** trước d_0 : **R_hist** (đồng bộ ngày; winsorize nhẹ 1% tail).
- Tính **drift** hằng ngày cho từng mã:
 - **mu_hist** (trung bình gần nhất)
 - **mu_tilt** $\propto \hat{p}_{\text{long}} / \text{sum}(\hat{p}_{\text{long}})$ (chuẩn hoá)
 - **drift** = $0.5 \cdot \text{mu_hist} + 0.5 \cdot \text{mu_tilt} \cdot \text{scale}$ (scale nhỏ, chúng tôi dùng **tilt_scale** ≈ 0.002 để **không ép** kịch bản chạy quá “đẹp” theo mô hình).

Sinh kịch bản:

- **Block bootstrap** đa biến, **block=5 phiên**, **HOLD=20 phiên** \Rightarrow ghép 4 block.
- Giữ nguyên tương quan giữa cột (mã) trong mỗi block.
- **Thêm drift** (dịch trung bình ngày), **clip $\pm 10\%$ /ngày** để tránh kịch bản phi thực tế.
- **Số kịch bản: 10.000** cho đánh giá cuối; tối ưu thì có thể 500–1.000 để chạy nhanh, sau đó xác nhận lại bằng 10.000.

CRN (Common Random Numbers):

- **Rút sẵn chỉ số block** cho toàn bộ kịch bản (seed cố định).
- Khi so sánh hai bộ trọng số khác nhau, **dùng cùng một bộ kịch bản** \rightarrow **so sánh công bằng** (variance giảm).

3-Stage tối ưu:

- **Stage A (quét lưới thô):** trọng số bước 5%, trần CAP/mã (ví dụ 0.3). Đánh giá ~ 2.000 kịch bản.

- Chỉ giữ ứng viên có **Sharpe trung vị** cao và **MDD trung vị $\leq 25\%$** .
- Lấy Top 20 ứng viên.
- **Stage B (zoom-in)**: tạo **200 biến thể** quanh mỗi ứng viên (tổng ~ 4.000). Đánh giá 5.000 kịch bản.
 - Lọc nghiêm **MDD trung vị & MDD phân vị 75%**.
 - Lấy Top 5 ứng viên **đa dạng** (tránh 5 cái quá giống).
- **Stage C (tinh chỉnh cục bộ)**: tham lam/gradient-like quanh 5 ứng viên cuối; đánh giá trên **10.000 kịch bản** (seed độc lập).
 - Chọn bộ trọng số cuối **max tổng hợp** (Sharpe trung vị, “Sharpe kịch bản xấu”, “MDD kịch bản xấu”).

Nhận xét:

- Cách làm **thực dụng và bền vững** hơn Markowitz cổ điển (vốn nhạy với ước lượng μ , Σ), **đặc biệt với dữ liệu Việt Nam** ít chiều và nhiễu cao.
- Phù hợp với nhà đầu tư cá nhân vì **trực giác rõ**: dự báo ra xác suất \rightarrow “ngiên” nhẹ kỳ vọng \rightarrow bootstrap từ lịch sử gần nhất \rightarrow tìm trọng số **ổn định qua nhiều kịch bản**.

9) Backtest “sát thực tế”

Quy tắc vận hành

- Tái cân bằng mỗi **20 phiên** (≈ 1 tháng).
- Chọn **Top-k** (mặc định 7) tại d_0 **vượt ngưỡng** $\hat{p} \geq 0.55$; nếu mã kỳ trước vẫn thuộc top \rightarrow **giữ liên tục** (tránh phí vòng vo).
- **Phân bổ**: dùng trọng số tối ưu Monte Carlo 3-stage; có trần **CAP per asset** (0.30) để tránh tập trung.
- **TP/SL theo từng mã** trong 20 phiên: TP=+15%, SL=-20%; khi chạm, **out khỏi vị thế** (tiền về “ví chờ” kỳ sau).
- **Phí giao dịch**: 0.1%/vòng (vào + ra) ở thời điểm tái cân bằng.

Thước đo

- **CAGR, Volatility (ann), Sharpe, Max Drawdown, Win rate, Tổng số lệnh, TP/SL hit, đường Equity/Drawdown, histogram daily returns, heatmap trọng số** theo 19 kỳ.

Kết quả cuối (Monte Carlo 10k kịch bản)

- **Total trades = 131 | TP hits = 2 | SL hits = 0**
- **Win rate $\approx 61\%$**
- **CAGR $\approx 28.79\%$**
- **Vol $\approx 21.88\%$**
- **Sharpe ≈ 1.32**
- **MDD $\approx -13.95\%$**

Nhận xét & diễn giải

- **Sharpe >1 với MDD $<-15\%$ là bức tranh cân đối** cho danh mục 7 mã tái cân bằng hàng tháng (có phí/TP/SL).
- Số lệnh đủ dày (131) để tin vào thống kê. **Không “đẹp quá mức”** như khi chỉ equal weight/top-k/không phí.
- **TP ít và SL = 0** gợi ý: phân bổ theo Monte Carlo đã **tránh các vị thế rủi ro**; nhiều lệnh kết thúc với lợi nhuận nhỏ—vừa thay vì nhắm TP “dày”.

10) Vì sao Transformer + Monte Carlo 3-stage “ăn điểm” so với baseline?

1. **Xác suất tốt hơn:** Transformer học tương tác *khoảng cách động* giữa tín hiệu kỹ thuật, vĩ mô thừa mẫu, và sentiment ngắn hạn. $W=60$ vừa đủ: **ít chông lán**, vừa “ôm” đủ 2–3 tháng nhịp thị trường.
2. **Danh mục ổn định hơn:** Monte Carlo 3-stage **không phụ thuộc chính xác μ/Σ** (thường ước lượng kém với dữ liệu ít), mà “lấy sức mạnh” từ **động lực lịch sử gần nhất + độ nghiêng nhẹ theo xác suất**. Cộng với **CRN** giúp so sánh trọng số công

bằng và giảm nhiều.

3. **Tính thực dụng:** Có ngưỡng vào, **TP/SL**, **phí**, **CAP** theo mã, **giữ vị thế cũ** khi vẫn nằm trong top → đúng tư duy vận hành thật và hạn chế turnover.
-

11) Những “bẫy” đã tránh & bài học rút ra

- **Look-ahead/label leakage:**
 - Không dùng thông tin tương lai khi gán nhãn/chuẩn hoá.
 - **Embargo** ≥ 20 phiên giữa train–val–test; chia theo **mốc thời gian**, không theo chỉ số dòng.
 - Sau khi xoá dòng lỗi (close=0, inf), **tính lại chỉ báo**.
- **Over-optimism do sample overlap:**
 - W=60 thay vì 90 giảm bớt chồng lấn; đánh giá theo ngày, không shuffle.
- **Kết quả quá “đẹp” ở val nhưng gãy ở test:**
 - Bổ sung **macro + regime** giúp ổn định;
 - **Sentiment** giúp “tránh sự kiện ngắn hạn”, cải thiện cả PR-AUC lẫn Brier.
- **Tối ưu trọng số “mỏng manh”:**
 - Không dựa vào MVO thuần; dùng **block bootstrap** và **CRN**; 3-stage kiểm soát **MDD median** và **MDD tail**.