

Time-series Data: Fundamentals and Concepts

TimeSeries Team

Ngày 6 tháng 9 năm 2025

Mục lục

1	General Information	2
2	Data Generation	4
3	Data Valuation	7
4	Model	10
5	Learning Paradigm	13
6	Evaluation	16
7	Repository	19
8	Kết luận	22

1 General Information

1) Time series là gì? Khác gì *sequence* nói chung?

Sequence data: tập quan sát được sắp xếp theo một **thứ tự có nghĩa** (thời gian, vị trí, tiến trình).

Time series là trường hợp đặc biệt của sequence, trong đó *trục thời gian* chi phối động học: đảo vị trí quan sát sẽ phá vỡ quan hệ trễ, tương quan, xu hướng và mùa vụ.

Hệ quả: mọi thao tác dữ liệu và đánh giá mô hình *phải bảo toàn thứ tự thời gian*; không trộn ngẫu nhiên như bài toán tabular.

2) Hạt mịn (granularity) và cấu trúc một bản ghi

Mỗi bản ghi gồm: **mốc thời gian** (time step), một hay nhiều **kênh/biến đo**, và **granularity** (khoảng lấy mẫu, ví dụ: mỗi giờ).

Granularity chi phối: *resampling*, tạo *lag/lead*, *rolling statistics*, và *calendar features* (giờ, thứ, tháng, kỳ lễ, ca vận hành).

Cần nhất quán timezone/DST; nội suy/điền khuyết sai có thể gây *méo phổ* đối với chuỗi có bước nhảy hay ràng buộc vật lý.

3) Phân loại chuỗi thời gian

Univariate: một kênh theo thời gian.

Multivariate: nhiều kênh đồng thời tại cùng mốc thời gian.

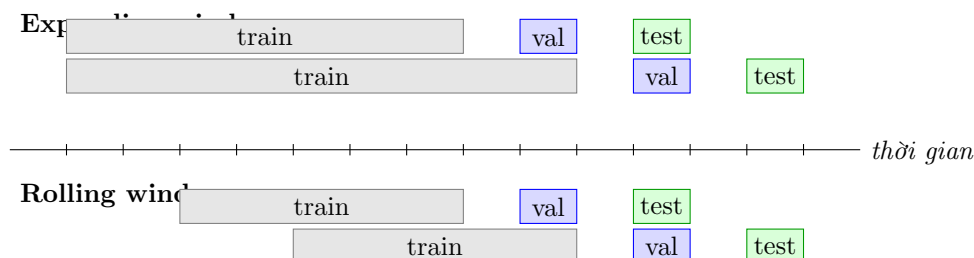
Multiple time series: nhiều chuỗi tương tự từ nhiều đối tượng/điểm đo (ví dụ nhiều trạm, nhiều thiết bị). *Hệ quả mô hình hoá*. Gộp nhiều series để học chung giúp tăng dữ liệu hiệu dụng (giảm overfit) nhưng có nguy cơ *bình quân hoá khác biệt*. Thực hành thường dùng chuẩn hoá theo từng series hoặc phân cụm series trước khi huấn luyện mô hình chung.

4) Chia tập theo thời gian (Data Partitioning)

Mục tiêu là đánh giá mô hình trong bối cảnh *tương lai có thể khác quá khứ*. Hai chiến lược cơ bản:

Expanding window: cửa sổ huấn luyện mở rộng dần \Rightarrow tận dụng lịch sử, giảm phương sai; hợp môi trường *ổn định*, *xu hướng dài hạn rõ*.

Rolling window: cửa sổ kích thước cố định, trượt theo thời gian \Rightarrow thích nghi xu hướng gần đây, chi phí tính toán bị chặn; hợp dữ liệu *phi-dừng* (*non-stationary*).



Hình 1: Minh hoạ chia tập theo thời gian: *expanding* (trên) và *rolling* (dưới).

Quy trình gợi ý.

Dùng *holdout theo thời gian* (train \rightarrow val \rightarrow test) kết hợp *walk-forward CV* (nhiều cửa sổ).

Ấn định *input length* và *forecast horizon* ngay từ bước chia để tránh rò rỉ tương lai khi tạo đặc trưng.

5) Nhiệm vụ điển hình

Forecasting: dự báo giá trị tương lai từ quan sát quá khứ.

Classification: gán nhãn cho toàn bộ chuỗi (kiểu hành vi/thiết bị).

Anomaly Detection: phát hiện điểm hoặc mẫu hình lệch khỏi hành vi mong đợi.

Imputation: điền khuyết các giá trị thiếu.

6) Dữ liệu lịch sử & bối cảnh AI/ML

Ghi chép thiên văn thời Tycho Brahe (thế kỷ XVI) là ví dụ sớm về time series: dù thiếu dữ liệu, độ chính xác vẫn đủ để suy ra mô hình chuyển động thiên thể.

Lĩnh vực dự báo chuỗi đang tăng trưởng mạnh trong AI/ML: ngày càng nhiều phương pháp và bài báo tại các hội nghị top-tier.

7) Checklist thao tác dữ liệu (áp dụng sớm)

Đồng bộ thời gian & granularity: nhất quán mốc, timezone/DST; cân nhắc đặc trưng lịch (giờ, thứ, tháng, lễ, ca).

Thiếu & ngoại lệ: chọn chiến lược imputation phù hợp (MCAR/MAR/MNAR); lưu vết các can thiệp (phục vụ anomaly/forecast).

Không rò rỉ tương lai: mọi feature (rolling mean/std, min/max, encoder) chỉ dùng dữ liệu $\leq t$.

Đặt mục tiêu sớm: xác định horizon (ví dụ y_{t+24}) và cửa sổ đầu vào để thiết kế split/feature.

Chuẩn hoá theo series cho multiple series; tránh series lớn “lấn át” series nhỏ.

Baseline trước, mô hình sau: chạy naive (lag-1, lag- m theo mùa) để có mốc so sánh, sau đó thử Linear/MLP/TCN/Transformer.

2 Data Generation

2.1 Mục tiêu & bối cảnh

Data Generation cho chuỗi thời gian gồm hai nhánh chính:

Data augmentation: áp biến đổi lên chuỗi gốc để tạo thêm mẫu, nhằm *tăng kích thước & đa dạng dữ liệu* \Rightarrow mô hình *robust* hơn, *ít overfit* hơn.

Dataset condensation: nén một *tập huấn luyện lớn* thành một *tập tổng hợp nhỏ* nhưng khi huấn luyện trên tập nhỏ này, mô hình vẫn đạt *hiệu năng tương đương* trên *test unseen*; hữu ích để *giảm chi phí, chia sẻ dữ liệu*, và *bảo vệ riêng tư* (lưu ý *distribution shift*).

2.2 Data Augmentation — dựa trên phân rã

Ý tưởng: tách chuỗi X thành **xu hướng** (T), **mùa vụ** (S), **phần dư** (R), rồi biến đổi từng thành phần và ghép lại.

STL: khi biên độ mùa vụ gần như hằng theo thời gian, dùng mô hình *cộng*

$$X_t = T_t + S_t + R_t.$$

MSTL: khi có *nhiều mùa vụ* (ngày/tuần/...) và/hoặc biên độ tỷ lệ theo mức nền, dùng mô hình *nhân*

$$X_t = T_t \times S_t \times R_t.$$

Quy trình gợi ý:

Chọn cửa sổ theo chu kỳ (ví dụ: 24h, 7 ngày), phân rã để thu S, T, R .

Biến đổi có kiểm soát:

Trend: scale nhẹ ($\pm 5\% - \pm 10\%$), dịch tuyến tính nhỏ.

Season: đổi *pha/biên độ* vừa phải; với mô hình nhân, nhân hệ số gần 1.

Residual: *jitter* nhiều trắng yếu, hoặc tráo đổi *block* rất ngắn.

Ghép lại đúng *dạng* (cộng hoặc nhân), giữ nhân nếu biến đổi bảo toàn ngữ nghĩa nhiệm vụ.

Chọn additive vs multiplicative:

Additive: biên độ mùa vụ xấp xỉ không đổi quanh mức nền.

Multiplicative: biên độ thay đổi tỷ lệ với mức nền (ngày lễ, cao điểm, ...).

Lưu ý an toàn:

Không làm lệch chu kỳ quá mức; hạn chế *dịch pha* trong biên nhỏ (ví dụ $\leq 10\%$ chu kỳ).

Tránh rò rỉ tương lai: mọi thống kê/biến đổi chỉ dùng dữ liệu $\leq t$ (trên *train* hoặc cửa sổ quá khứ).

Nhất quán dạng: phân rã additive thì *cộng* khi ghép; multiplicative thì *nhân*.

2.3 Data Augmentation — các biến đổi cơ bản

Có thể dùng độc lập hoặc kết hợp phân rã:

Jitter / Noise injection: cộng nhiễu nhỏ vào R hoặc trực tiếp vào X (ưu tiên R).

Scaling / Magnitude warp: nhân hệ số nhỏ quanh 1 (tự nhiên với *multiplicative*).

Time warping / Phase shift: giãn/thu thời gian cục bộ hoặc dịch pha mùa vụ vài bước.

Window slicing & cropping: cắt đoạn con có chồng lấn để tăng mẫu mà vẫn giữ cấu trúc.

Permutation nhẹ (block-wise): tráo các *khối ngắn* trong R để đa dạng nhiều.

Mixup/CutMix theo thời gian: trộn hai series cùng miền (ưu tiên trên phần *residualized*); giữ nhân cho forecasting nếu phù hợp.

Masking (hữu ích cho imputation/self-supervised): che ngẫu nhiên một số bước và buộc mô hình dự đoán lại.

Frequency-domain tweaks: can thiệp biên độ các *seasonal harmonics* (FFT/Wavelet) rồi biến đổi ngược.

2.4 Dataset Condensation — nén dữ liệu huấn luyện

Định nghĩa: học ra một *tập nhỏ tổng hợp* để *thay thế* tập lớn thật trong huấn luyện, với điều kiện mô hình huấn luyện trên tập nhỏ đạt *hiệu năng gần tương đương* mô hình huấn luyện trên tập thật khi đánh giá cùng *test unseen*.

Khi nào dùng?

Cần *giảm chi phí* tính toán/IO; cần *chia sẻ* khi dữ liệu gốc bị ràng buộc; mục tiêu *bảo vệ riêng tư*.

Rủi ro & cảnh báo:

Distribution shift: tập ngưng tụ có thể bỏ sót *regime* hiếm.

Memorization/Leakage: mẫu tổng hợp quá giống dữ liệu thật; cần kiểm tra *nearest-neighbor distance/membership inference*.

Protocol đánh giá:

[left=1.2em,itemsep=0.25em,topsep=0.3em]

1. Cố định pipeline (model, HPO, metric).
2. Huấn luyện M_{real} trên $\text{train}_{\text{real}}$, và $M_{\text{synthetic}}$ trên $\text{train}_{\text{condensed}}$.
3. Đánh giá chung trên $\text{test}_{\text{unseen}}$; so sánh MAE/MASE/RelMAE... Chỉ coi *condensation đạt* khi $M_{\text{synthetic}} \approx M_{\text{real}}$.

2.5 Gợi ý chọn “recipe” theo bài toán

Forecasting (uni/multi):

Bắt đầu với STL/MSTL; tăng/giảm nhẹ biên độ S , scale T , jitter R .

Thêm *window slicing* và *masking* (nếu kết hợp self-supervised).

Tránh biến đổi phá vỡ *seasonality/horizon* đã định.

Anomaly Detection:

Bảo toàn *tỉ lệ anomaly*; có thể sinh anomaly bằng *spike/step change* vào R hoặc dịch pha ngắn hạn của S (gắn nhãn rõ ràng).

Imputation:

Dùng *masking có cấu trúc* (block missing) để mô hình học *ngữ cảnh dài*; kết hợp *time warping* nhẹ để tăng robustness.

2.6 Lỗi thường gặp & cách tránh

Nhầm *additive/multiplicative* khi ghép lại \Rightarrow tín hiệu méo; kiểm tra công thức trước khi tổng hợp.

Dịch pha mùa vụ quá lớn \Rightarrow dự báo trượt pha; giới hạn $\leq 10\%$ chu kỳ.

Rò rỉ tương lai khi tính thống kê toàn chuỗi; chỉ dùng dữ liệu $\leq t$ hoặc tính trên *train*.

Condensation kiểm chứng thiếu: phải so trên cùng *test unseen* và đối chiếu hiệu năng với tập thật.

2.7 Tóm tắt thực chiến

Dữ liệu mùa vụ rõ: ưu tiên **STL/MSTL + augmentation** theo thành phần.

Bị giới hạn tài nguyên/nhu cầu chia sẻ: cân nhắc **condensation**, với *protocol so sánh nghiêm* và giám sát *distribution shift*.

3 Data Valuation

3.1 Hai lớp “giá trị dữ liệu”

(A) Thuộc tính thống kê (không xét thứ tự thời gian).

Mục tiêu: mô tả *phân bố* và phát hiện ngoại lệ; không phản ánh cấu trúc thời gian.

Bộ tóm tắt phổ biến: \min , \max , mean , median (Q_2), range , $\mathbf{IQR} = Q_3 - Q_1$.

Quy tắc ngoại lệ (boxplot): điểm nằm ngoài $[Q_1 - 1.5 \cdot \text{IQR}, Q_3 + 1.5 \cdot \text{IQR}]$.

Skewness (độ lệch): dương \Rightarrow đuôi phải dài; âm \Rightarrow đuôi trái dài.

Kurtosis (độ dày đuôi): cao \Rightarrow đuôi nặng/đỉnh nhọn; thấp \Rightarrow đuôi nhẹ/đỉnh phẳng.

(B) Đặc trưng theo thời gian (temporal).

Trend (xu hướng): chuyển động dài hạn lên/xuống.

Seasonality (mùa vụ): mẫu lặp theo chu kỳ cố định.

Remainder/Residual (phần dư): phần không giải thích bởi trend/season.

3.2 Đo “độ mạnh” Trend/Season từ phân rã

Giả sử phân rã (additive) $X = T + S + R$, đặt $R = X - T - S$. Các chỉ số trực giác:

$$\text{TrendStrength} = \max\left(0, 1 - \frac{\text{var}(R)}{\text{var}(X) - \text{var}(S)}\right), \quad \text{SeasonStrength} = \max\left(0, 1 - \frac{\text{var}(R)}{\text{var}(X) - \text{var}(T)}\right).$$

Diễn giải & gợi ý:

$\text{var}(R)$ lớn \Rightarrow phần chưa giải thích còn nhiều (mẫu hình bị bỏ sót).

Trend cao \Rightarrow ưu tiên cửa sổ đầu vào dài, mô hình/đặc trưng ổn định dài hạn.

Season cao \Rightarrow bảo toàn chu kỳ (lag m), baseline seasonal-naive, hoặc đặc trưng hoá các *harmonics*.

3.3 Transitions / Regime changes (chuyển pha hành vi)

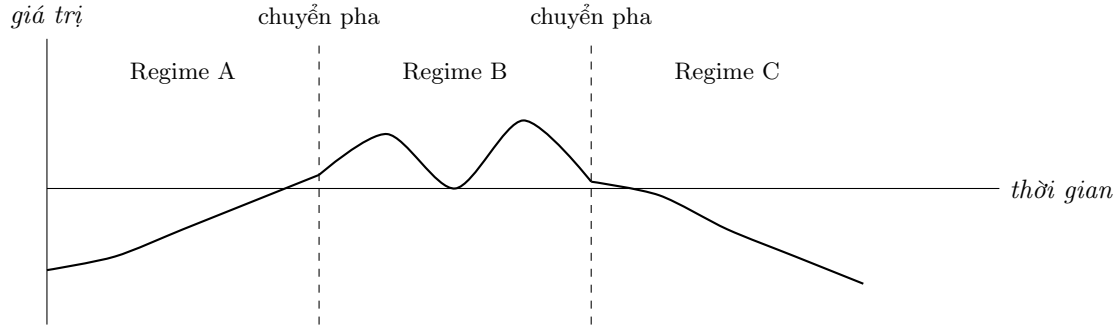
Vì sao quan trọng? Bất được *đổi chế độ* giúp: nhận diện bất thường, phân biệt hành vi chuỗi, và chọn cơ chế đánh giá/huấn luyện hợp bối cảnh (ví dụ rolling).

Gợi ý thao tác:

Gắn nhãn *regime* theo thời đoạn (dựa vào dấu/sườn trend, sức mạnh season) hoặc dùng mô hình ẩn (HMM/Markov).

Tính một độ đo biến thiên chuyển trạng thái (ví dụ độ phân tán/trace của ma trận hiệp phương sai xác suất chuyển).

Khi mức chuyển pha cao \Rightarrow ưu tiên **rolling window** và/hoặc mô hình theo bối cảnh (mixture-of-experts).



Hình 2: Minh hoạ chuyển pha: A (trend lên, mùa vụ yếu) \rightarrow B (phẳng, mùa vụ mạnh) \rightarrow C (trend xuống, nhiễu tăng).

3.4 Biến phương sai & vai trò của $\text{var}(X)$

$\text{var}(X)$ đo độ phân tán quanh trung bình; xuất hiện trực tiếp trong các chỉ số TrendStrength/SeasonStrength. Nếu $\text{var}(X)$ lớn nhưng $\text{var}(R)$ không giảm sau phân rã \Rightarrow phần chưa giải thích còn đáng kể.

3.5 Quy trình “định giá dữ liệu” gợi ý (workflow)

1. **Thống kê mô tả** (không thời gian): min/max/mean/median/range/IQR + skew/kurtosis \Rightarrow so nhóm, tìm ngoại lệ, gợi ý chuẩn hoá.
2. **Phân rã** (STL/MSTL hoặc tương đương) \Rightarrow tính TrendStrength/SeasonStrength; xác định chu kỳ trội.
3. **Phát hiện chuyển pha**: theo dõi chỉ số theo thời đoạn; gán *regime* và đo mức chuyển pha.
4. **Quyết định modeling & split**:
5. Trend/season mạnh, ít chuyển pha \Rightarrow **expanding window**.
6. Nhiều chuyển pha \Rightarrow **rolling window** và ưu tiên đặc trưng “gần hiện tại”.
7. **Chọn baseline & metric**: nếu có mùa \Rightarrow seasonal-naive; dùng metric *scale-free* (MASE/Rel-MAE) để so sánh công bằng giữa nhiều chuỗi khác thang đo.

3.6 Ví dụ ngắn (minh hoạ tư duy)

Dữ liệu bán lẻ theo ngày có đỉnh cuối tuần: IQR tăng mạnh dịp lễ \Rightarrow dùng chuẩn hoá *robust*, tránh xoá nhầm tín hiệu khi lọc ngoại lệ.

Phân rã: SeasonStrength cao, TrendStrength trung bình \Rightarrow baseline seasonal-naive + mô hình có thành phần mùa.

Giai đoạn “Flash sale” làm đổi hành vi (chuyển pha) \Rightarrow đánh giá bằng rolling/walk-forward để phản ánh học từ gần đến xa.

3.7 Lỗi thường gặp & cách tránh

Nhầm giữa “thuộc tính thống kê” và “đặc trưng thời gian”: mean/median/IQR không khẳng định có season/trend.

Diễn giải sai Trend/SeasonStrength: chỉ số cao mang tính tương đối sau phân rã; kiểm tra *residual* đã “trắng” chưa.

Bỏ qua chuyển pha: chỉ đánh giá một lần trên holdout khi dữ liệu nhiều regime \Rightarrow kết luận sai; dùng walk-forward/rolling.

4 Model

4.1 Phân loại theo hình dạng I/O (By I/O Shape)

Short-term: cửa sổ vào/ra ngắn; thường $\text{đầu ra} \leq \text{đầu vào}$; hợp tín hiệu biến thiên nhanh, chu kỳ ngắn.

Long-term: cửa sổ vào/ra dài; đôi khi $\text{đầu ra} \geq \text{đầu vào}$; nhấn mạnh khả năng nắm cấu trúc dài hạn.

4.2 Phân loại theo kiểu đầu ra (By I/O Type)

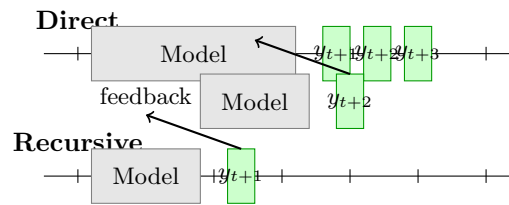
Point/Deterministic: dự báo *điểm* (một số cụ thể); đơn giản, dễ so sánh nhưng không phản ánh bất định.

Probabilistic: dự báo *phân phối/dải* (quantile, interval); hữu ích cho quản trị rủi ro và quyết định.

4.3 Phân loại theo phương thức dự báo (By Method)

Direct: sinh thẳng tất cả bước tương lai yêu cầu \Rightarrow tránh tích lũy sai số; khó cho chân trời rất dài.

Recursive: dự báo từng bước, đưa đầu ra làm đầu vào cho bước kế \Rightarrow linh hoạt nhưng dễ tích lũy sai số.



Hình 3: Direct (trên) vs. Recursive (dưới).

4.4 Phân loại theo kiến trúc (By Architecture)

(A) MLP / Linear & biến thể. *Linear, NLinear, DLinear, N-BEATS, FITS, TSMixer/TimeMixer.*

Ưu: nhanh, đơn giản, baseline mạnh.

Nhược: thiếu “bộ nhớ” trật tự; cần đặc trưng thời gian/phân rã để hiệu quả hơn.

(B) RNN/LSTM/GRU & biến thể. *LSTNet, DA-RNN, SegRNN, xLSTM/xLSTMTime.*

Ưu: có bộ nhớ cho phụ thuộc dài; linh hoạt theo thời gian.

Nhược: huấn luyện chậm; có nguy cơ “quên xa” nếu không xử lý tốt.

Bắt *motif cục bộ* ngắn; có thể học mẫu *trong chu kỳ* và *giữa chu kỳ*.

(C) CNN.

Conv *nhân quả* (không rò rỉ tương lai), *dãn* (receptive field lớn), *residual*; xử lý chuỗi dài hiệu quả.

(D) TCN (Temporal Convolutional Networks).

Hướng chính: sửa *attention* cho phụ thuộc thời gian; nhúng *module TS*; mô hình tương tác đa biến.

Không có “nhà vô địch tuyệt đối” cho LTSF; kết quả phụ thuộc dữ liệu, chân trời và HPO.

(E) Transformer cho chuỗi thời gian.

4.5 Chiến lược kênh (Channel Strategies)

(A) **Channel-Independent.** Mô hình mỗi kênh riêng hoặc *patching* (PatchTST).

Ưu: robust khi các kênh rất khác nhau (thang đo, mẫu hình).

Nhược: bỏ qua tương tác giữa kênh.

Học phụ thuộc giữa kênh bằng attention/mixer; mạnh khi kênh tương quan.

Khi kênh dị biệt, biểu diễn dùng chung có thể làm giảm hiệu năng.

(B) **Channel-Dependent (Cross-Dimension/Mixing).**

Cân bằng *capacity vs robustness* bằng cách gom nhóm kênh tương đồng; cải thiện tổng quát hoá cho kênh mới (zero-shot) bằng prototype/transfer.

Bảng 1: So sánh nhanh chiến lược kênh
ccc

Chiến lược	Ưu điểm	Hạn chế / Khi tránh
Channel-Independent	Robust khi kênh dị biệt; dễ song song hoá	Bỏ qua tương tác giữa kênh
Channel-Dependent	Khai thác tương quan liên kênh; capacity cao	Dễ kém ổn định khi kênh khác biệt mạnh
Channel Clustering	Thoả hiệp capacity–robustness; tốt cho kênh mới	Cần bước phân cụm/đánh giá độ tương đồng ban đầu

(C) **Channel Clustering.**

4.6 Đa tỉ lệ (Multi-Scale)

Kết hợp mẫu hình *global* (thô) và *local* (tinh) qua đa độ phân giải hoặc phân rã; thông tin phong phú hơn \Rightarrow dự báo tốt hơn.

4.7 Foundation Models & TS + Text

Ghép *time series + văn bản* (tin tức/báo cáo) để vừa dự báo vừa giải thích; mô hình chuyên miền có thể ít/không cần fine-tune lại.

4.8 Hướng dẫn chọn mô hình (thực chiến)

[left=1.2em,itemsep=0.25em,topsep=0.3em]

1. Theo dữ liệu/kịch bản:
2. Kênh khác nhau rõ \Rightarrow *channel-independent*/patching hoặc *clustering*.
3. Kênh tương quan rõ \Rightarrow *cross-dimension*/mixer/attention.
4. Mẫu lặp cục bộ mạnh \Rightarrow *CNN/TCN*; chuỗi rất dài \Rightarrow *TCN*.

5. Theo chân trời:

6. Ngắn hạn \Rightarrow Linear/MLP/CNN (nhanh); ưu tiên *Direct* để giảm tích lũy sai số.
7. Dài hạn \Rightarrow Transformer/TCN/Mixer hoặc RNN; chú ý HPO vì không có “nhà vô địch” cố định.
8. **Theo tài nguyên & vòng lặp:** Dùng Linear/DLinear/TSMixer làm baseline nhanh; nâng cấp TCN/Transformer khi còn dư địa.
9. **Theo chiến lược đánh giá:** Nhiều chuyển pha \Rightarrow *rolling window* và đặc trưng “gần hiện tại”; ổn định \Rightarrow *expanding*.

4.9 Lỗi phổ biến & cách tránh

Áp *channel-dependent* khi kênh dị biệt \Rightarrow tụt hiệu năng; kiểm tra tương quan/đồng dạng trước.

Kỳ vọng “một mô hình thắng mọi nơi” cho LTSF; cần benchmark đa họ (Linear/Mixer/TCN/Transformer).

Dùng *Recursive* cho chân trời rất dài mà không kiểm soát tích lũy sai số; cân nhắc *Direct* hoặc hybrid.

5 Learning Paradigm

5.1 Bức tranh toàn cảnh

Bốn “hệ” học cốt lõi cho chuỗi thời gian:

Supervised: học từ cặp (đầu vào, nhãn) để tối ưu loss có giám sát (MAE/MSE/Quantile loss...). Với forecasting, đầu vào là cửa sổ quá khứ $\{x_{t-L+1}, \dots, x_t\}$, nhãn là $\{x_{t+1}, \dots, x_{t+H}\}$.

Semi-Supervised: kết hợp ít nhãn với nhiều mẫu không nhãn (consistency regularization, pseudo-labeling, self-training...).

Unsupervised: khai phá cấu trúc ẩn *không cần nhãn* (nén-khôi phục, mô hình mật độ, phân cụm, phát hiện motif/regime).

Self-Supervised: tự sinh “nhãn” từ dữ liệu theo quy tắc định trước (dịch/che/biến dạng, so khớp augmentation); học *biểu diễn* giàu ngữ cảnh trước khi fine-tune. *Forecasting là supervised hay self-supervised?* Truyền thống xem là **supervised** (có cặp (past, future) rõ ràng); góc nhìn representation-first xem là **self-supervised** vì “nhãn tương lai” sinh tự động bằng quy tắc trượt/shift.

5.2 Self-Supervised cho chuỗi thời gian

Các pretext điển hình:

Masking/Denoising: che điểm hoặc block liên tiếp, buộc mô hình khôi phục \Rightarrow học phụ thuộc dài hạn.

Next-step/Patch prediction: dự đoán bước hoặc mảng (patch) kế tiếp; nhãn lấy trực tiếp từ dữ liệu đã dịch.

Augmentation-invariance: tạo hai “view” (jitter, time-warp nhẹ, crop, mixup theo thời gian), tối ưu để biểu diễn bất biến trước biến đổi. **Thiết kế cặp Positive/Negative (contrastive):**

Positive: hai view của cùng thực thể/cùng thời đoạn nhưng khác augmentation.

Negative: đoạn từ thực thể hoặc bối cảnh khác (khác ngữ nghĩa thời gian).

5.3 Generative Learning cho time series

(A) Autoregressive (AR). Mô hình hoá $p(x_{t+1} | x_{\leq t})$ và lặp cho các bước tương lai; từ ARIMA truyền thống tới neural AR (TCN/Transformer tự hồi quy).

(B) Autoencoder (AE). Nén x thành mã $z = E(x)$, giải nén $\tilde{x} = D(z)$; biến thể: VAE, Seq2Seq, masked-AE. Ứng dụng: giảm chiều, khử nhiễu, phát hiện bất thường (dựa lỗi tái tạo), pretrain biểu diễn.

(C) Diffusion. Thêm nhiễu dần, học mạng “khử nhiễu” để đảo quá trình; ứng dụng: sinh chuỗi đa dạng, in-painting theo thời gian, augmentation xác suất.

5.4 Adversarial Learning (GAN-style)

Hai ứng dụng chính:

Generation: sinh chuỗi mới (uni/multivariate) để mở rộng dữ liệu, mô phỏng kịch bản hiếm.

Imputation: hoàn khuyết block thiếu bằng Generator có điều kiện theo ngữ cảnh; Discriminator đảm bảo tính “thật”. *Lưu ý:* phòng *mode collapse* (đa dạng thấp) bằng regularization (gradient penalty, spectral norm...) và kiểm định bằng thống kê chu kỳ/ACF/PACF lẫn hiệu năng downstream.

5.5 Contrastive Learning

Học nhúng $f(x)$ sao cho *gần* positive và *xa* negative (InfoNCE/NT-Xent; hoặc biến thể không negative như BYOL/Barlow). Ứng dụng: pretrain encoder thời gian cho forecasting, classification, retrieval sự kiện. **Mẹo xây cặp:** *Instance-wise* (cùng thiết bị/series \Rightarrow positive), *Temporal-wise* (cùng series, sát thời gian \Rightarrow positive; xa bối cảnh/khác regime \Rightarrow negative).

5.6 Federated Learning (FL) cho chuỗi thời gian

Dữ liệu phân tán tại nhiều client (trạm cảm biến/thiết bị IoT...). Mục tiêu: *không di chuyển dữ liệu*, chỉ gửi tham số/cập nhật để tổng hợp mô hình toàn cục (FedAvg, FedProx...). **Thách thức:** non-IID mạnh (seasonality, trend khác nhau), granularity/missing khác biệt, *personalization* (tham số/lớp riêng cho client) để vừa chia sẻ tri thức chung vừa phù hợp từng đối tượng.

5.7 Forecasting đặt ở đâu trong bức tranh?

Thực hành ứng dụng: coi là supervised, tối ưu trực tiếp MAE/MSE/Pinball loss.

Chiến lược biểu diễn: self-supervised để pretrain (masking/contrastive/next-patch), rồi fine-tune supervised cho tác vụ chính.

5.8 Cách chọn “paradigm” theo mục tiêu & ràng buộc

[left=1.2em,itemsep=0.25em,topsep=0.3em]

1. **Nhân ít, miền đa dạng, cần tổng quát hoá:** ưu tiên *Self-Supervised/Contrastive* để pretrain encoder; fine-tune mỏng downstream.
2. **Cần sinh dữ liệu/điền khuyết mạnh:** cân nhắc *Diffusion/GAN*; đánh giá bằng downstream + chỉ số cấu trúc thời gian.
3. **Dữ liệu phân tán/nhảy cảm:** *Federated* với tổng hợp chống non-IID, thêm personalization; cân nhắc differential privacy nếu cần.
4. **Hiệu quả triển khai:** bắt đầu từ *supervised* đơn giản (Linear/DLinear/TCN) làm baseline; còn dư địa \Rightarrow bổ sung pretrain SSL hoặc mô hình generative cho augmentation/khôi phục.

5.9 Quy trình khuyến nghị (end-to-end)

1. **Khảo sát & định giá dữ liệu:** missing, regime, season/trend; đặt horizon, granularity, kênh.
2. **Baseline supervised:** chia *expanding/rolling* đúng bối cảnh; chạy naive + Linear/TSMixer/TCN để có mốc.
3. **Pretrain SSL (nếu phù hợp):** chọn pretext (masking/contrastive/next-patch) + augmentation bảo toàn ngữ nghĩa.
4. **Fine-tune supervised:** cho tác vụ chính; cân nhắc dự báo *probabilistic* nếu cần dài.
5. **(Tuỳ chọn) Generative/Adversarial/Diffusion:** cho sinh dữ liệu hoặc imputation; đánh giá bằng downstream + thống kê thời gian.
6. **(Tuỳ chọn) Federated:** khi dữ liệu phân tán/nhảy cảm; thêm cơ chế chống non-IID và personalization.

5.10 Lỗi thường gặp & cách phòng tránh

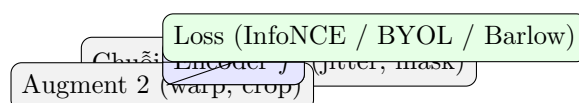
Augmentation phá ngữ nghĩa: dịch pha mùa vụ lớn/xáo trộn chu kỳ \Rightarrow biểu diễn sai; chỉ dùng biến đổi hợp tự nhiên của chuỗi.

Contrastive chọn cặp sai: dễ collapse/học nhầm; positive gắn với *thực thể & thời đoạn*, negative khác ngữ nghĩa.

GAN mode collapse: cần regularization/penalty và kiểm định downstream, không chỉ nhìn đồ thị.

Bỏ qua lợi ích pretrain: coi forecasting chỉ là supervised \Rightarrow bỏ lỡ SSL; nên xem self-supervised như bước hỗ trợ.

FL không xử lý non-IID: hội tụ kém/thiên lệch client; dùng thuật toán tổng hợp phù hợp, personalization, và (nếu cần) privacy.



Hình 4: Sơ đồ tự giám sát: tạo hai view của cùng đoạn, mã hoá bằng encoder chung, tối ưu mục tiêu để hai biểu diễn gần nhau.

6 Evaluation

6.1 Mục tiêu của đánh giá dự báo

Đúng thang đo & công bằng giữa chuỗi: các thước đo phải có ý nghĩa trên từng series và so sánh được giữa nhiều series khác đơn vị/thang đo.

So với baseline hợp lý: mọi kết luận nên đặt cạnh mốc chuẩn (naive hoặc seasonal-naive).

Ổn định theo thời gian & từng chân trời: phản ánh chất lượng ở từng bước dự báo $h = 1, \dots, H$ và qua nhiều cửa sổ walk-forward.

Có ý nghĩa vận hành: nếu ra quyết định bằng dải bất định, cần đánh giá hiệu chỉnh (calibration) và độ sắc (sharpness), không chỉ dự báo điểm.

6.2 Baseline và cách dùng đúng

Naive (không mùa vụ): $\hat{y}_{t+1} = y_t$.

Seasonal-naive: $\hat{y}_{t+1} = y_{t+1-m}$ với chu kỳ m (ngày/tuần/ca...).

Baseline phải được xây *ngoài mẫu* trên tập đánh giá; không “học ké” thông tin từ train.

Khi báo cáo, luôn nêu quan hệ với baseline (thắng/hoà/thua) thay vì chỉ nêu con số tuyệt đối.

6.3 Các họ thước đo và ý nghĩa

MAE (Mean Absolute Error): $MAE = \frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t|$ (≥ 0). Đơn vị-phụ thuộc; không nên cộng gộp thẳng giữa các series khác thang đo.

RMSE/MSE: nhạy ngoại lệ hơn MAE; dùng khi muốn phạt mạnh sai số lớn.

(A) Phụ thuộc thang đo (scale-dependent).

MAPE (Mean Absolute Percentage Error): dễ lệch khi y_t nhỏ/tiệm 0 và có trường hợp không xác định; có thể phạt over-forecast khác under-forecast ở cùng tỷ lệ.

(B) Phần trăm (percentage-based) — cần thận trọng.

RelMAE (Relative MAE):

$$\text{RelMAE} = \frac{\text{MAE}(\text{model})}{\text{MAE}(\text{baseline})}.$$

Diễn giải: < 1 tốt hơn baseline; $= 1$ ngang; > 1 tệ hơn. Nếu $\text{MAE}(\text{baseline}) = 0 \Rightarrow$ không xác định.

RMAE (Relative error trung bình) là biến thể tính “relative error” từng điểm rồi lấy trung bình; diễn giải tương tự.

(C) Tương đối theo baseline (relative-to-baseline).

MASE (Mean Absolute Scaled Error):

$$\text{MASE} = \frac{\frac{1}{N_{\text{test}}} \sum_{t \in \text{test}} |y_t - \hat{y}_t|}{\frac{1}{N_{\text{train}}-1} \sum_{t \in \text{train}} |y_t - y_{t-1}|}.$$

Mẫu số tính trên *train* (xấp xỉ MAE của naive lag-1 trong train), tử số trên *test*; < 1 thắng naive, $= 1$ hoà, > 1 thua. Nếu mẫu số $= 0 \Rightarrow$ không xác định.

(D) Chuẩn hoá theo “bậc nhất” — MASE (scale-free, ổn định).

Pinball loss cho quantile;

Coverage & Width cho các PI (50/80/90/95%): mục tiêu là *calibrated* (coverage gần nominal) và *sharp* (độ rộng nhỏ);

CRPS/WIS: lượng hoá toàn bộ phân phối hoặc nhiều khoảng dự báo.

(E) Phân phối/dải dự báo (probabilistic).

Dùng **Energy score**, **Variogram score** để phản ánh tương quan liên kênh, thay vì đánh từng kênh rồi cộng.

(F) Đa biến (multivariate).

6.4 Gộp kết quả qua nhiều series/horizon một cách công bằng

Macro vs. Micro: Macro = trung bình metric theo từng series (công bằng giữa series); Micro = gộp toàn bộ điểm lỗi (dễ bị series biên độ lớn chi phối).

Theo horizon: báo cáo cho từng $h = 1..H$ và trung bình theo horizon (nhiều mô hình tốt ngắn hạn nhưng tụt về cuối chân trời).

Theo thời đoạn/regime: tách trước/sau sự kiện, mùa cao điểm... để tránh che khuất thực tế.

Theo nhóm series: nhóm theo miền/thang đo (trạm, khu vực), báo cáo macro trong nhóm và macro toàn cục.

6.5 Quy trình backtesting & kiểm soát rò rỉ

Chia theo thời gian (expanding/rolling) và *định nghĩa rõ input length & horizon* trước khi tạo đặc trưng.

Walk-forward qua nhiều cửa sổ (train \rightarrow val \rightarrow test) để giảm phương sai ước lượng.

Cố định baseline và cách tính metric (ví dụ: mẫu số MASE tính trên train; RelMAE dùng naive out-of-sample).

Kiểm tra rò rỉ:

Mọi rolling stats/normalizer chỉ dùng dữ liệu $\leq t$.

Không học tham số chuẩn hoá trên toàn bộ lịch sử rồi áp cho test.

Với metric percentage/relative, bảo đảm mẫu số không lấy từ đoạn chứa tương lai.

6.6 Chẩn đoán lỗi & trực quan hoá

Residual diagnostics: histogram/Q-Q, ACF của residual (còn cấu trúc \Rightarrow mô hình chưa bắt hết), biến phương sai theo thời gian.

Horizon-wise plots: MAE/MASE theo $h = 1..H$ để thấy xu hướng suy giảm.

Calibration: PIT histogram; *coverage vs nominal* cho PI 50/80/95%.

Series-wise league table: xếp hạng theo từng series (thắng/hoà/thua baseline) để phát hiện nhóm khó.

6.7 So sánh mô hình một cách có ý nghĩa

Ưu tiên **scale-free/relative** (MASE/RelMAE) trước, sau đó bổ sung thước đo tuyệt đối (MAE/RMSE) để cảm nhận độ lớn sai số.

Thông kê kiểm định: Diebold–Mariano cho chênh lệch lỗi theo thời điểm; bootstrap theo cửa sổ để ước lượng khoảng tin cậy cho chênh lệch RelMAE/MASE; tường thuật thêm tỷ lệ cửa sổ mô hình thắng.

Phân tích sai lỗi có cấu trúc: xem thiên lệch over/under, nhạy cảm theo regime, và độ bền theo mùa vụ.

6.8 Các “bẫy” thường gặp và cách né

Dùng **MAPE** khi có $y_t = 0$ hoặc rất nhỏ \Rightarrow không xác định/thổi phồng sai số; thay bằng **MASE/RelMAE**.

So **MAE** giữa series khác thang đo \Rightarrow gộp không công bằng; dùng **scale-free/relative**.

Tính **MASE** sai: mẫu số không ở train, hoặc mẫu số = 0 mà vẫn báo số \Rightarrow không hợp lệ.

Baseline “học” từ test: naive/seasonal-naive phải đúng định nghĩa và *out-of-sample*.

Chỉ báo cáo một con số tổng hợp: che khuất vấn đề theo horizon/regime; cần horizon-wise & series-wise.

6.9 Mẫu báo cáo khuyến nghị (gợi ý cấu trúc)

Thiết lập: tập dữ liệu, chu kỳ m , cách chia (expanding/rolling), horizon H , input length L .

Baseline: naive/seasonal-naive (định nghĩa rõ).

Metric: chính = **MASE, RelMAE**; bổ sung **MAE** để cảm nhận tuyệt đối; với dự báo xác suất, thêm **coverage/width** (50/80/95%) và **CRPS/WIS**.

Cách gộp: macro theo series, horizon-wise; bảng league series-wise (thắng/hoà/thua baseline).

Kiểm định: chênh lệch RelMAE/MASE + khoảng tin cậy (bootstrap/walk-forward); (tùy chọn) DM test.

Chẩn đoán: residual ACF, PIT/coverage plots; phân tách theo regime.

Kết luận & rủi ro: nơi mô hình mạnh/yếu, điều kiện vận hành, cảnh báo khi có $y_t \approx 0$ (tránh MAPE).

6.10 Ghi chú thực hành nhanh (cheatsheet)

Có mùa vụ: baseline = **seasonal-naive**; metric = **MASE/RelMAE**.

So sánh nhiều series khác thang đo: **không** dùng MAE đơn lẻ; chuyển sang **scale-free/relative**.

Có số 0: **tránh MAPE**; dùng **MASE/RelMAE**.

Dự báo probabilistic: **coverage & calibration** bắt buộc (kèm CRPS/WIS).

7 Repository

7.1 Mục tiêu của phần “Repository”

Giúp lựa chọn nhanh bộ công cụ phù hợp cho dự báo/biểu diễn chuỗi thời gian theo ba tiêu chí: (i) độ bao phủ phương pháp & metric; (ii) khả năng tái lập; (iii) tính mở rộng cho workflow hiện đại (self-supervised, generative, federated).

7.2 Time Series Library (TSLib)

Tóm tắt. Thư viện tổng hợp nhiều phương pháp dự báo, hữu ích làm “vừa” baseline mạnh để quét nhanh dữ liệu mới.

Khi nên dùng.

Cần bộ mô hình phong phú (Linear/MLP/Transformer...) để khảo sát sớm trước khi HPO nặng.

Muốn tái lập kết quả với giao diện gần nhau, dễ đóng gói pipeline. **Điểm nổi bật.**

Thường được các benchmark khác gọi kèm để mở rộng tập phương pháp có sẵn.

Nhiều *predefined class* cho model/transform, tiện lắp ghép. **Lưu ý.** API từng họ mô hình có thể khác nhỏ (shape, chuẩn hoá, mask) \Rightarrow nên chuẩn hoá adapter khi chạy hàng loạt.

7.3 Time Series Benchmark (TFB)

Tóm tắt. Khung benchmark “một giao diện—nhiều phương pháp”, đi kèm metric thống nhất và báo cáo tái lập.

Vì sao hữu ích.

Unified interface & metrics: dễ cắm model mới, giảm sai lệch do cách tính khác nhau.

Reproducible reports: tự động hoá báo cáo so sánh. **Tình huống dùng.**

Dựng benchmark nội bộ hoặc báo cáo nghiên cứu.

Đào tạo nội bộ: minh hoạ tác động của split/metric/chiến lược.

7.4 GluonTS

Tóm tắt. Thư viện hướng thực dụng với bao phủ rộng về *phương pháp, transform, metric*; mạnh ở *dự báo xác suất* (quantile/distribution).

Điểm mạnh.

Hỗ trợ nhiều mô hình xác suất, thuận tiện nếu nghiệp vụ cần dải dự báo.

Có *DataLoader*/transform sẵn, ráp nhanh pipeline PyTorch/MXNet (tùy phiên bản). **Lưu ý.** Khi gộp nhiều series khác thang đo, ưu tiên metric *scale-free/relative* (MASE/RelMAE) ở bước đánh giá.

7.5 ULTS — Unsupervised Representation Learning for Time Series

Tóm tắt. Kho tập trung học biểu diễn *không giám sát/tự giám sát* cho chuỗi thời gian; phù hợp pretrain encoder rồi fine-tune cho downstream.

Khi nên dùng.

Nhãn khan hiếm, dữ liệu đa miền; cần robust trước shift.

Xây retrieval/semantic search dựa trên embedding thời gian. **Thực chiến.** Thiết kế cặp *positive/negative* đúng ngữ nghĩa thời gian (positive: cùng thực thể/thời đoạn; negative: khác bối cảnh).

7.6 FedProC — Federated cho dự báo chuỗi thời gian

Tóm tắt. Khuôn khổ *federated* cho môi trường đa trạm/thiết bị (thời tiết, năng lượng, IoT) nơi dữ liệu không thể tập trung.

Tại sao cần.

Non-IID giữa client (mùa vụ/biên độ khác nhau), granularity/missing không đồng nhất.

Yêu cầu riêng tư/tuân thủ: dữ liệu ở nguyên tại chỗ, chỉ trao đổi tham số. **Gợi ý sử dụng.**

Khởi động bằng mô hình nền (Linear/TCN/Transformer), dùng FedAvg/FedProx và thêm *personalization*.

Phù hợp kịch bản nhiều trạm (thời tiết/điện lực), có thể tái hiện thực tế phân tán.

7.7 So sánh nhanh & khuyến nghị chọn kho

Bảng 2: Gợi ý chọn repository theo nhu cầu

Nhu cầu	Khuyến nghị
Benchmark tái lập, báo cáo tự động, metric thống nhất	TFB: một interface, nhiều phương pháp; báo cáo tự động
Baseline đa dạng để “quét” nhanh	TSLib: nhiều phương pháp sẵn, thích hợp cho baseline
Pipeline xác suất/quantile và transform phong phú	GluonTS: hỗ trợ distribution/quantile và transform
Học biểu diễn không/tự giám sát	ULTS: tập trung representation learning
Liên minh/federated (nhiều trạm, dữ liệu nhạy cảm)	FedProC: tổng hợp tham số, hỗ trợ personalization

7.8 Lộ trình tích hợp thực chiến (đề xuất)

Chuẩn hoá đánh giá: chọn metric *scale-free/relative* (MASE/RelMAE) làm chuẩn; cố định baseline (naive/seasonal-naive) & walk-forward.

Dựng benchmark: dùng **TFB** để gom **TSLib** + mô hình riêng dưới một interface/metric duy nhất (báo cáo tái lập).

Mở rộng xác suất & transform: nếu cần dải bất định/phép biến đổi phong phú, cắm thêm **GluonTS**.

Pretrain biểu diễn: khi nhân hiểm/đa miền, dùng **ULTS** cho self/unsupervised representation, thiết kế positive/negative hợp ngữ nghĩa.

Federated: khi dữ liệu phân tán/nhạy cảm, chuyển pipeline sang **FedProC**, bật tổng hợp tham số & personalization; giữ nguyên chuẩn metric để so sánh công bằng.

7.9 Cạm bẫy phổ biến & cách né

Trộn API/metric không thống nhất giữa repo \Rightarrow kết luận sai; luôn ép về cùng interface/metric.

Chỉ dùng MAE để gộp nhiều series khác thang đo \Rightarrow thiên lệch; ưu tiên MASE/RelMAE khi tổng hợp.

Contrastive trong ULTS chọn cặp sai \Rightarrow embedding kém nghĩa; positive phải cùng thực thể/thời đoạn, negative khác bối cảnh.

Federated không xử lý non-IID \Rightarrow hội tụ kém; thêm personalization và đánh giá series-wise theo từng client.

7.10 Ghi chú đường link & phạm vi (tra cứu nhanh)

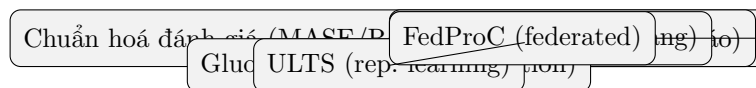
TSLib: thư viện baseline đa dạng cho TS.

TFB: benchmark với interface thống nhất và báo cáo tái lập.

GluonTS: nhiều method/metric/transform, mạnh về dự báo xác suất.

ULTS: repository tập trung representation learning không/tự giám sát cho TS.

FedProC: khuôn khổ federated cho dự báo trong môi trường đa trạm.



Hình 5: Lộ trình tích hợp thực chiến: chuẩn hoá đánh giá \rightarrow benchmark TFB \rightarrow baseline TSLib; mở rộng GluonTS/ULTS; triển khai federated với FedProC.

8 Kết luận

Trong dự báo chuỗi thời gian, chất lượng hệ thống không đến từ một “mô hình thần thánh”, mà từ **chuỗi quyết định đúng**: hiểu dữ liệu \rightarrow tách tập theo thời gian \rightarrow chọn baseline và thước đo công bằng \rightarrow ghép kiến trúc phù hợp bối cảnh \rightarrow đánh giá, giám sát và lặp lại. Khi tôn trọng cấu trúc thời gian, nhất quán granularity/timezone và kiểm soát rò rỉ thông tin, bạn đã đi được nửa chặng đường.

Dữ liệu là trung tâm. Hãy coi **Trend–Season–Residual** là khung tư duy: phân rã để hiểu, tăng cường dữ liệu (augmentation) theo từng thành phần để mở rộng đa dạng mà vẫn giữ ngữ nghĩa. Nếu cần chia sẻ/tiết kiệm chi phí, cân nhắc *dataset condensation* nhưng phải kiểm chứng nghiêm trên cùng *test unseen*. Việc “định giá dữ liệu” trước (thống kê mô tả, sức mạnh trend/season, chuyển pha regime) giúp chọn đúng chiến lược tách tập và baseline ngay từ đầu.

Mô hình không có “vua” tuyệt đối. Dữ liệu kênh rất khác nhau thường hợp *channel-independent/patching*; dữ liệu có tương quan liên kênh nên khai thác *cross-dimension/mixing*; nếu cần cân bằng, dùng *channel clustering*. Bổ sung tư duy *đa tỉ lệ* và đặc trưng theo mùa giúp các kiến trúc Linear/MLP/TCN/Transformer bắt nhịp dữ liệu tốt hơn.

Kết hợp các hệ học. Thực chiến: coi forecasting là *supervised* để tối ưu mục tiêu nghiệp vụ. Khi nhân hiếm/đa miền: *self-/contrastive* để tiền huấn luyện biểu diễn rồi *fine-tune supervised*. Cần sinh dữ liệu/điền khuyết: cân nhắc *diffusion/GAN*; dữ liệu phân tán: *federated* kèm *personalization*.

Đánh giá công bằng và có ý nghĩa. Ưu tiên **MASE/RelMAE** để so sánh nhiều series khác thang đo; luôn đặt cạnh *naive/seasonal-naive* để biết thực sự tốt hơn đâu. Với dự báo xác suất, kiểm tra *calibration* (coverage) song hành *sharpness* (độ hẹp khoảng). Báo cáo nên tách *theo horizon*, *theo regime*, và *theo từng series*.

Lộ trình 10 bước (đề xuất áp dụng)

Cố định granularity/timezone và chân trời H .

Khảo sát dữ liệu: thiếu/ngoại lệ, trend/season strength, chuyển pha.

Định nghĩa baseline (naive/seasonal-naive).

Chọn tách tập *expanding* (ổn định) hoặc *rolling* (nhiều chuyển pha).

Tạo đặc trưng tôn trọng thời gian (lag, rolling, calendar, decomposition/multi-scale).

Chạy baseline nhanh (Linear/DLinear/TSMixer/TCN) để “đọc vị” dữ liệu.

Nâng cấp kiến trúc theo chiến lược kênh (independent/dependent/clustering) và theo chân trời.

Nếu nhân ít/đa miền: *pretrain self-/contrastive*, rồi *fine-tune supervised*.

Đánh giá bằng **MASE/RelMAE** (+ MAE/RMSE để cảm nhận tuyệt đối) và, nếu có, *coverage/CRP-S/WIS* cho dự báo xác suất.

Chuẩn hoá benchmark, lưu cấu hình/HPO/seed, thiết lập giám sát *drift* & tái hiệu chỉnh sau triển khai.

Thông điệp cuối cùng. Đầu tư đúng vào *dữ liệu* và *đánh giá* sẽ mang lại lợi ích bền vững hơn bất kỳ đổi mô hình đơn lẻ nào. Hãy bắt đầu nhỏ—chuẩn hoá baseline, metric và quy trình—rồi mở rộng có kiểm soát bằng những thành phần phù hợp nhất với bối cảnh của bạn.