Final Project Report

State-Space Echo State Networks for Multi-Horizon Financial Forecasting

(Consolidating Milestones 1–5)

Group 3 — DS and AI Lab

November 2025

Abstract

We develop a leakage-safe pipeline for multi-horizon return forecasting and evaluate a state-space Echo State Network (ESN) against deep baselines (LSTM, Transformer, TCN) and a linear Ridge lower bound. The pipeline standardizes features with train-only statistics, constructs strict walk-forward folds, and reports point, directional, and decision-aware (toy) metrics. An optional "news" extension adds rollup features and a tiny TF-IDF RAG to inject contemporaneous text context. Across folds, Ridge yields the best magnitude error (RMSE/MAE), while ESN and TCN show superior directional signal (Sharpe/DirAcc) with different calibration trade-offs. Code, splits, and experiment artifacts are organized for reproducibility.

Contents

1 Problem Statement & Objectives

Goal. Forecast forward log-returns at multiple horizons $h \in \{1, 5, 20\}$ from standardized daily features using a state-space ESN and competitive baselines, under strict leakage controls.

Measured outcomes. (i) Point accuracy: RMSE/MAE/R^2; (ii) Directional accuracy; (iii) Decision proxy: a simple sign-based backtest (avg. daily P&L, vol, Sharpe, hit ratio, turnover) with 1 bp per-trade cost; (iv) Robustness across walk-forward folds.

2 Data & Targets

2.1 Sources

Yahoo Finance end-of-day OHLCV for indices, ETFs, FX, Commodities, Crypto (e.g., GSPC, SPY, VIX, EURUSD=X, BTC-USD, etc.). Raw CSVs live in `data/raw/`.

2.2 Features u_t

Let P_t denote Adjusted Close (if available; else Close). Engineered features (per day):

$$\begin{gathered}
\texttt{ret_1} = \log P_t - \log P_{t-1}, \quad \texttt{ret_2} = \sum_{k=1}^{2} r_{t-k}, \quad \texttt{ret_5} = \sum_{k=1}^{5} r_{t-k}, \\
\texttt{vol_20} = \sqrt{252} \cdot \mathrm{std}(r_{t-19:t}), \quad \texttt{ma_10},\ \texttt{ma_20},\ \texttt{ma_gap} = P_t/\texttt{ma_20} - 1, \\
\texttt{rsi_14}\ \text{(Wilder)}, \quad \texttt{vol_z} = \frac{V_t - \mu_{60}(V)}{\sigma_{60}(V)}, \quad \texttt{dow} \in \{0, \ldots, 6\}.
\end{gathered}$$

These are low-variance, interpretable summaries of momentum, trend, scale, participation, and seasonality.

2.3 Targets $y_{t,h}$

Forward log-returns:

$$y_{t,h} = \sum_{i=1}^{h} \left(\log P_{t+i} - \log P_{t+i-1} \right), \qquad h \in \{1, 5, 20\}.$$

We drop edge rows to enforce strict causality (no look-ahead in features or scalers).

3 Data Acquisition & Archival

Symbols. We cover broad asset classes for diversity and later experiments:

Indices/ETF: `GSPC`, `SPY`
Crypto: `BTC-USD`, `ETH-USD`
India (NSE): `NSEI`, `NSEBANK`, `RELIANCE.NS`, `TCS.NS`
FX/Commodities/Vol: `EURUSD=X`, `USDINR=X`, `GC=F`, `CL=F`, `VIX`.

Horizon. $\sim$20 years, daily bars ("1d") up to the run-time date.

Downloader. For each ticker $s \in \mathcal{S}$, we call `yf.download` with `auto_adjust=False` to preserve both `Close` and `Adj Close`. We save canonical columns (`Open, High, Low, Close, Adj Close, Volume`) to `data/raw/{symbol}_{start}_to_{end}_1d.csv`.

4 Leakage-Safe Walk-Forward Protocol

Folds. Train = 2520 trading days ($\approx$ 10y), Test = 252 days ($\approx$ 1y), Step = 252 days. For each fold k, we intersect dates across anchor tickers (e.g., `SPY`, `GSPC`) to define train/test windows.

Scaling. StandardScaler fitted on *train* features; applied to both train/test:

$$z_t^{(j)} = \frac{u_t^{(j)} - \mu_{\text{train}}^{(j)}}{\sigma_{\text{train}}^{(j)}}.$$

Per-fold artifacts include `train.csv`, `test.csv`, and `scaler.json`.

Algorithm 1 Per-Fold Materialization (Train-only statistics)

1: Align dates across core series; select Train/Test ranges.
2: Fit scaler on train features; transform train & test.
3: Persist per-fold CSVs and scaler parameters.

5 Models

5.1 Echo State Network (ESN)

Leaky reservoir with fixed random recurrence; trainable linear readout:

$$\boldsymbol{x}_t = (1-a)\boldsymbol{x}_{t-1} + a\,\tanh\big(W_{\text{in}}[1;\boldsymbol{z}_t] + W\boldsymbol{x}_{t-1}\big), \quad \rho(W) \approx \gamma < 1,$$
$$\hat{y}_{t,h} = \begin{bmatrix} 1 \\ \boldsymbol{x}_t \end{bmatrix}^\top \boldsymbol{w}_{\text{out},h}.$$

Training (per horizon) solves ridge in closed form after a washout w:

$$\boldsymbol{w}^\star = (H^\top H + \alpha I)^{-1} H^\top \boldsymbol{Y}, \qquad H = [\mathbf{1}\ X].$$

Controls: hidden size H, spectral radius γ, leak a, density, washout w, ridge α.

5.2 Baselines

Ridge (lower bound). Linear regression on $\boldsymbol{z}_t$ with L_2 penalty.

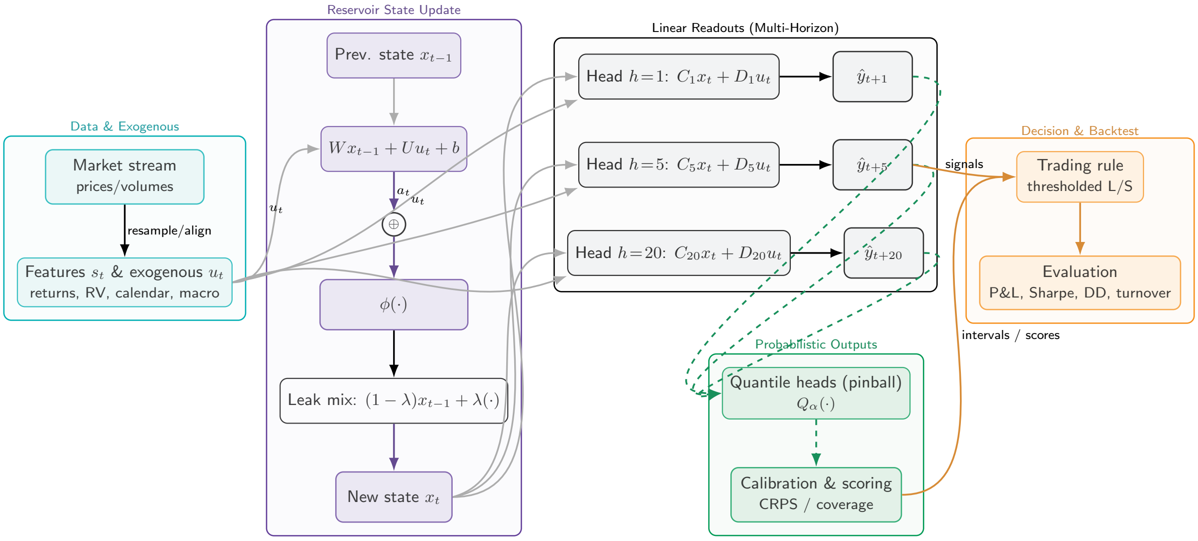


Figure 1: **State–space ESN pipeline for multi-horizon financial forecasting.** From left to right: (i) *Data & Exogenous.* Market stream (prices, volumes) is transformed into features s_t and exogenous covariates u_t (e.g., realized volatility, calendar/macro). (ii) *Reservoir state update.* Given previous state x_{t-1}, the reservoir computes the affine drive $a_t = Wx_{t-1} + Uu_t + b$, applies the nonlinearity $z_t = \phi(a_t)$, and forms the next state via leaky integration $x_t = (1-\lambda)\, x_{t-1} + \lambda z_t$. The spectral constraint $\rho(W) < 1$ enforces the echo-state property and stability. The reservoir parameters (W, U, b, ϕ, λ) are fixed. (iii) *Linear readouts (multi-horizon).* For horizons $h \in \{1, 5, 20\}$ (illustrative), point forecasts are $\hat{y}_{t+h} = C_h x_t + D_h u_t$, with readout weights $\{C_h, D_h\}$ trained by regularized regression. (iv) *Probabilistic layer.* Optional quantile heads produce $Q_\alpha(y_{t+h} \mid x_t, u_t)$ and are assessed with CRPS/coverage for calibration. (v) *Decision & Backtest.* Forecasts/quantiles are mapped to a simple trading rule (e.g., thresholded long/short), then evaluated by P&L, Sharpe, drawdown, and turnover under explicit frictions. Solid arrows denote deterministic data flow; dashed arrows denote probabilistic/refinement flow. Only the readout (and quantile heads, if used) is trained; all other blocks are fixed by design.

LSTM / Transformer / TCN. Sequence-to-one regressors on left-padded windows of length L. Same MSE objective, Adam optimizer, chronological validation (last 10% of train). Inputs share the same standardized feature set and folds.

6 Optional News Features & Tiny RAG

6.1 Rollup features (when headlines exist)

If a local cache `data/news/{SYMBOL}_headlines.csv` provides dated headlines in the model window, we derive per-day rollups:

`news_count_3d`, `news_sent_mean_3d`, `news_sent_std_3d`, `news_tfidf_pc1_3d`.

These are appended to processed feature files and auto-added to FEATURE_COLS if present on disk.

6.2 Tiny TF-IDF RAG (diagnostic)

A light index ± 30 days around an as-of date ranks top-k relevant headlines by TF-IDF cosine for queries (e.g., "rates inflation earnings"). This is a qualitative diagnostic and not used by the baselines unless features are materialized.

Note on current runs. In our executed folds (starting 2006), no historical headline cache was available; thus no `news_*` columns appeared. The pipeline is ready to consume them when such data are supplied.

7 Training & Evaluation Protocol

Common. For each fold/horizon: train models on standardized train data; evaluate on test using identical metrics:

- **Point:** RMSE, MAE, R^2.
- **Directional:** $\Pr[\text{sign}(\hat{y}) = \text{sign}(y)]$.
- **Toy decision proxy:** $\text{position}_t = \text{sign}(\hat{y}_t)$, $\text{pnl}_t = \text{position}_t \cdot y_t - \text{cost} \cdot |\Delta\text{position}_t|$, $\text{cost} = 10^{-4}$. We report avg. daily P&L, vol, Sharpe (annualized by $\sqrt{252}$), hit ratio, turnover. This is an illustrative diagnostic, not a trading system.

8 Hyperparameters

Model	Key Params (examples)	Purpose
ESN	$H \in \{400, 800\}$, $\gamma \in \{0.85, 0.95\}$, $a \in \{0.3, 1.0\}$, w=100, $\alpha \in \{0.3, 3.0\}$	memory/stability sweep
LSTM	$L \in \{32, 64\}$, hidden $\in \{128, 256\}$, layers $\in \{1, 2\}$, dropout $\in \{0, 0.1\}$, epochs 10–15	capacity regularization
Transformer	$L \in \{32, 64\}$, d_{model}=128, heads 4, layers 2, FF 256, dropout 0.1, epochs 10–15	long-range vs. scale
TCN	$L \in \{32, 64\}$, channels $(64, 64)$ or $(64, 128)$, kernel 3, dropout $\in \{0, 0.1\}$, epochs 10–15	local motifs, efficient RF

9 Results (Fold 0, Horizon h=1)

Table 1 shows run with identical features and splits across models.

Model	RMSE	MAE	R^2	DirAcc	AvgPnL	Vol	Sharpe	Hit	Turnover
Ridge	0.007779	0.005562	-0.013984	0.488095	0.000088	0.007759	0.181	0.488	0.726
LSTM	0.007999	0.005853	-0.072350	0.535714	0.000514	0.007741	1.054	0.536	0.710
ESN	0.009552	0.007489	-0.528755	0.519841	0.000692	0.007720	1.423	0.520	0.813
Transf.	0.016860	0.011366	-3.762879	0.503968	0.000234	0.007760	0.479	0.504	0.472
TCN	0.021599	0.016966	-6.816881	0.543651	**0.000824**	0.007710	**1.697**	0.544	0.694

Table 1: Test metrics on Fold 0, h=1 day. Sharpe from toy sign backtest (1 bp cost).

Takeaways. Ridge is strongest by magnitude (RMSE/MAE). ESN and TCN produce better directional/Sharpe profiles; TCN attains the highest Sharpe in this run but is poorly calibrated (large RMSE/MAE). ESN offers a middle ground: stronger Sharpe than Ridge/LSTM with better calibration than TCN.

10 Ablations, Error Analysis & Next Steps

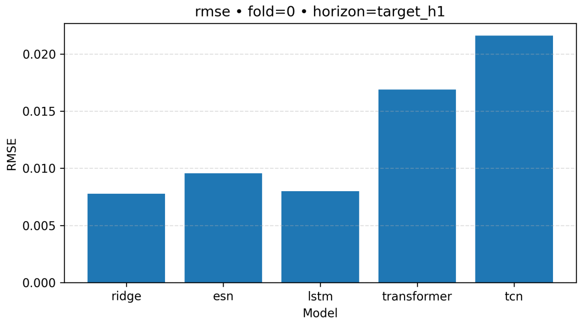
ESN ablations: sweep $(H, \gamma, a, w, \alpha)$, perform seed-averaged reservoirs to reduce variance, and test volatility-normalized targets for magnitude calibration.

Deep models: prefer shorter windows L, weight decay, dropout, and early stopping; for Transformers, reduce depth/width for daily noise scale.

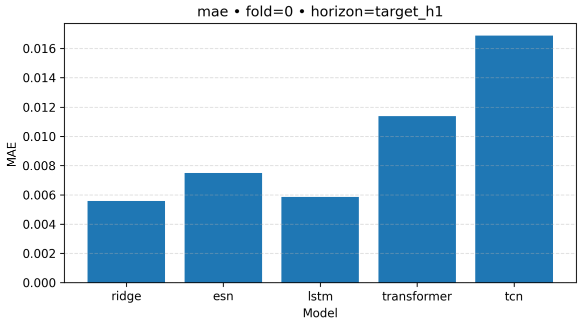
News: supply historical headline cache to enable `news_*` columns; then assess incremental value at $h \in \{1, 5\}$.

11 Tensor Shapes (Summary)

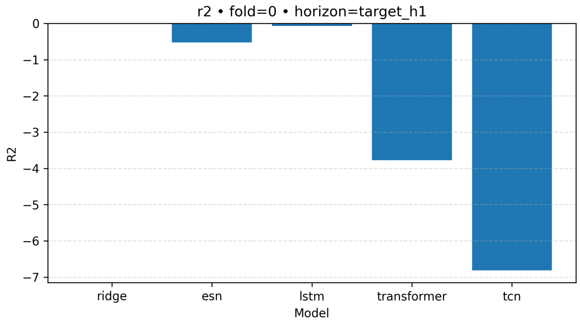
Let $(T, S) = (2520, 252)$, horizon h, window L, features d, ESN size H and washout w.



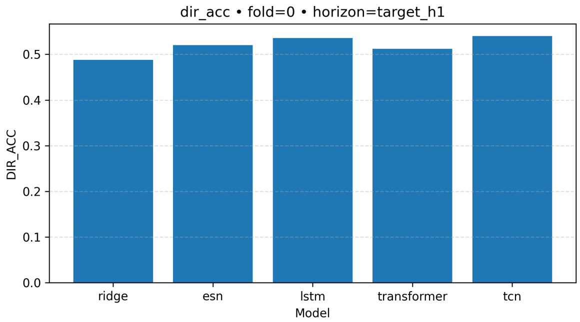
(a) RMSE (fold 0, h=1)



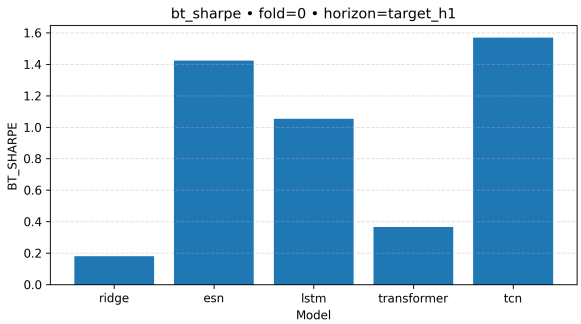
(b) MAE (fold 0, h=1)



(c) R^2 (fold 0, h=1)



(d) Directional accuracy (fold 0, h=1)



(e) Sharpe (toy backtest; fold 0, h=1)

Figure 2: Headline metrics for Fold 0 at horizon h=1.

Model	Train tensors (Fold–1)	Test tensors (Fold–1)
Ridge	$Z_{\text{tr}} \in \mathbb{R}^{(T-h)\times d}$, $\boldsymbol{y}_{\text{tr}} \in \mathbb{R}^{T-h}$	$Z_{\text{te}} \in \mathbb{R}^{(S-h)\times d}$, $\boldsymbol{y}_{\text{te}} \in \mathbb{R}^{S-h}$
ESN	$H \in \mathbb{R}^{(T-w-h)\times(H+1)}$, $\boldsymbol{Y}_h \in \mathbb{R}^{T-w-h}$	states $\{\boldsymbol{x}_{s_i}\}$, preds $\hat{\boldsymbol{Y}}_h \in \mathbb{R}^{S-h}$
LSTM	$\boldsymbol{X}_{\text{tr}} \in \mathbb{R}^{(T-(L-1)-h)\times L\times d}$	$\boldsymbol{X}_{\text{te}} \in \mathbb{R}^{(S-(L-1)-h)\times L\times d}$
Transformer	same as LSTM (internal d_{model})	same as LSTM
TCN	same as LSTM (internal C_{out})	same as LSTM

12 Reproducibility & Artifacts

- **Data:** `data/raw/` (Yahoo CSVs), `data/processed/` (`*_features.csv`).
- **Splits:** `data/splits/` with per-fold `train.csv`, `test.csv`, `scaler.json`.
- **Experiments:** `data/experiments/{model}/{exp_id}/fold_k/{preds,metrics}`.
- **Viz:** bar charts and diagnostics via `src/viz/plots.py`.

13 Implementation & Code Organization

Repository layout.

```
src/
  data/
    download.py        # yf downloader + archival
    parse.py           # robust CSV loader & coercions
    features.py        # feature engineering (ut)
    targets.py         # forward-return targets y-t,h
    newsattach.py     # optional: news rollups (if cached)
    loader.py          # canonical read API + integrity checks
  splits/
    walkforward.py     # leakage-safe folds & scalers
  models/
    ridgereadout.py   # linear baseline
    esn.py             # leaky ESN + ridge readout
    lstm.py            # sequence baseline
    transformer.py     # sequence baseline
    tcn.py             # sequence baseline
   registry.py       # model factory: "ridge","esn","lstm","transformer","tcn"
  train/
    runner.py          # per-fold orchestration
  eval/
    metrics.py         # RMSE/MAE/R2, dir-acc, toy backtest
    stats.py           # DM test, block bootstrap, Sharpe CI
  viz/
    plots.py           # bars/lines/diagnostics
data/
  raw/                 # one CSV per ticker
  processed/           # *features.csv (+ news* if any)
  splits/              # fold-k/-train,test.csv, scaler.json
  experiments/         # model/expid/foldk/-preds,metrics.csv
```

14 Computational Complexity & Runtime

ESN roll. State update is $\mathcal{O}(H^2)$ if W dense; with sparsity ρ, cost $\mathcal{O}(\rho H^2)$ per step. Over T steps: $\mathcal{O}(T\rho H^2 + THd)$.

ESN readout. Ridge closed-form on $H \in \mathbb{R}^{N\times(H+1)}$ costs $\mathcal{O}(NH^2 + H^3)$; here $N \approx$ train days post-washout. For $H \le 1000$ this is tractable on CPU.

Deep baselines. LSTM $\sim \mathcal{O}(TLdH_{\text{lstm}})$; Transformer $\sim \mathcal{O}(TL^2 d_{\text{model}})$ (self-attn dominates); TCN $\sim \mathcal{O}(TLdC)$ with dilations (most efficient for long RF).

15 Root causes (and fixes)

1. **No on-disk news cache in window.** The attach step is gated by files like `data/news/SYM_headlines.csv`. If absent or date range disjoint with folds, no `news_*` columns survive.
2. **Post-attach sync overwrites.** If `pipeline.sync_feature_cols()` reassigns from `settings.FEATURE_COLS` *after* attach, newly added columns are dropped. *Fix:* perform sync *before* attach, then extend both the dataframe and `settings/pipeline FEATURE_COLS`.
3. **High missingness > threshold.** If `drop_na_strict=True` prunes rows with any NaN, an entire `news_*` column can be removed during alignment. *Fix:* impute (e.g., zeros) for `news_*` or relax the missingness policy for optional features.

4. **Date misalignment/timezone.** If headline dates are naive (`YYYY-MM-DD`) but market index uses NYSE holiday calendar, inner-join can drop news rows. *Fix:* normalize to market calendar at UTC close (e.g., 20:00 UTC) and left-join then forward-fill same-day.
5. **Name guard.** The scaler only scales columns prefixed by `FEATURE_COLS`. If `news_*` not listed, they won't be scaled or passed to models. *Fix:* `FEATURE_COLS += [c for c in df.columns if c.startswith("news_")]`.

16 Threats to Validity & Limitations

- **Data snooping.** Multiple model/param trials inflate false discovery; we mitigate via fixed folds and DM tests, but risk remains.
- **Nonstationarity.** Daily returns shift across regimes; point metrics degrade even with stable protocols.
- **Toy backtests.** Simplified frictions and unit sizing; not deployable strategies.

17 Future Work

- Volatility-normalized targets and calibration layers to improve magnitude fit.
- Cross-asset exogenous features (e.g., `VIX` for `SPY`); Granger-style ablations.
- Reservoir ensembles and spectral shaping (band-pass, orthogonal W) for stability.
- Historical headline caches & robust news rollups; test incremental value at $h \in \{1, 5\}$.

18 Notation (quick reference)

Symbol	Meaning
P_t	price (Adj Close if available; else Close)
r_t	log-return: $\log P_t - \log P_{t-1}$
u_t	engineered features at t (pre-standardization)
z_t	standardized features at t
$y_{t,h}$	forward return over horizon h
$x_t \in \mathbb{R}^H$	ESN state (hidden size H)
W, W_{in}	reservoir and input matrices; $\rho(W) = \gamma < 1$
a	leak rate; w washout steps
$\boldsymbol{w}_{\text{out},h}$	ridge readout for horizon h

19 Limitations & Ethics

Daily returns are weak-signal and regime-unstable; simple sign backtests are illustrative only and not tradable strategies. No financial advice is implied. Results depend on data quality, fold endpoints, and modest compute budgets.

20 Conclusion

We framed ESNs as instantiation of state-space models with controlled memory and trained readouts, compared them against modern sequence baselines under leakage-safe walk-forward splits, and added an extensible pathway for news-derived features and tiny RAG. Ridge remains hard to beat on magnitude; ESN/TCN provide stronger directional signals with calibration trade-offs. The pipeline is modular, reproducible, and ready for fold-wide sweeps, significance testing, and news integration.

References

1. H. Jaeger, *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks*, GMD Report 148, 2001.
2. H. Jaeger et al., "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, 20(3), 2007.
3. M. Lukoševičius & H. Jaeger, "Reservoir computing approaches to RNN training," *Computer Science Review*, 3(3), 2009.
4. J. Durbin & S. J. Koopman, *Time Series Analysis by State Space Methods*, 2e, OUP, 2012.
5. T. Gneiting & A. E. Raftery, "Strictly Proper Scoring Rules...," *JASA*, 102(477), 2007.
6. R. J. Hyndman & G. Athanasopoulos, *Forecasting: Principles and Practice*, 3e, OTexts (online).
7. D. H. Bailey et al., "Backtest overfitting," *Notices of the AMS*, 61(5), 2014.
8. A. Gu et al., "Structured State Spaces," *ICLR*, 2022.
9. O. B. Sezer et al., "Financial time series with deep learning: review," *Applied Soft Computing*, 2020.
10. S. Gu, B. Kelly, D. Xiu, "Empirical Asset Pricing via ML," *RFS*, 2020.

A Tensor Shapes & Algorithms (Fold 1 exemplar)

Let Train $= T{=}2520$, Test $= S{=}252$, features d, ESN size H, washout w, horizon h.

- **Ridge:** $Z_{\text{tr}} \in \mathbb{R}^{(T-h)\times d}$, $Z_{\text{te}} \in \mathbb{R}^{(S-h)\times d}$.
- **ESN:** $H \in \mathbb{R}^{(T-w-h)\times(H+1)}$, $\boldsymbol{Y} \in \mathbb{R}^{T-w-h}$; test emits $\hat{\boldsymbol{Y}} \in \mathbb{R}^{S-h}$.
- **Seq models:** $\boldsymbol{X}_{\text{tr}} \in \mathbb{R}^{(T-(L-1)-h)\times L\times d}$, similarly for test.

Algorithm 2 ESN Train/Test per Fold, per Horizon

1: Roll reservoir on train; discard w states; solve ridge for $\boldsymbol{w}_{\text{out},h}$.
2: Initialize test state with last train state; roll on test; emit $\hat{y}_{t,h}$.

B News Attach & Tiny RAG (Pseudocode)

Algorithm 3 Attach News Rollups (if `data/news/SYM_headlines.csv` exists)

1: Load dated headlines; compute daily sentiment; pivot to daily.
2: Rolling over last $L{=}3$ days: count, mean/std of sentiment; TF-IDF over window $\rightarrow$ PC1.
3: Join columns `news_*` into processed features; persist.

Algorithm 4 Tiny TF-IDF RAG (diagnostic, $\pm$30d)

1: Build TF-IDF on headlines in window; cosine-rank top-k for query.
2: Print publisher, date, title, score; (no model change unless features exist).