# Milestone 2

## State-Space Models for Multi-Horizon Financial Forecasting

Group 3 - DS and AI Lab

October 10, 2025

## 1 Overview and Rationale

**Goal.** Prepare leakage-safe, multi-asset datasets suitable for modeling with state-space models. We: (i) download and archive raw daily OHLCV histories from Yahoo Finance via `yfinance`, (ii) robustly parse heterogeneous CSV formats, (iii) engineer stable, interpretable features $u_t$, (iv) construct forward-return targets at multiple horizons, and (v) produce *walk-forward* train/test splits with per-fold, train-only normalization and a light baseline readout.

**Why these choices?** Financial series are nonstationary and noisy; high-variance modeling choices are brittle. ESN training is readout-only and benefits from (a) leakage-safe preprocessing, (b) features with sensible inductive bias (trend/mean-reversion/volatility), and (c) validation protocols aligned with temporal causality. Walk-forward splits and train-only scalers directly address look-ahead bias.

## 2 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.** ~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`.

## 3 Robust Parsing of Raw CSVs

Yahoo exports appear in two flavors: (A) standard single-header CSV with a `Date` column or indexed date; (B) a multi-row header variant with lines `Price/Ticker/Date`. We implement a robust loader:

1. Try standard parse; coerce dates → `DatetimeIndex`, numeric columns via `to_numeric`.
2. If canonical OHLCV not detected, fall back to scanning for a row starting with "Date", use it as header, then coerce.
3. Keep the canonical OHLCV order; drop all-NaN rows.

## 4 Feature Engineering ($u_t$)

Let $P_t$ denote the adjusted (if available) or close price used for modeling, renamed as PX. We compute:

$$\text{(Log return)} \quad r_t \coloneqq \log P_t - \log P_{t-1}. \tag{1}$$

$$\text{(Lagged cum. returns)} \quad \texttt{ret\_2}_t \coloneqq r_{t-1} + r_{t-2}, \quad \texttt{ret\_5}_t \coloneqq \sum_{k=1}^{5} r_{t-k}. \tag{2}$$

$$\text{(Realized vol, annualized)} \quad \texttt{vol\_20}_t \coloneqq \sqrt{252} \cdot \mathrm{std}\big(r_{t-19:t}\big). \tag{3}$$

$$\text{(Moving avgs)} \quad \texttt{ma\_10}_t \coloneqq \mathrm{mean}\big(P_{t-9:t}\big),\ \texttt{ma\_20}_t \coloneqq \mathrm{mean}\big(P_{t-19:t}\big). \tag{4}$$

$$\text{(MA gap)} \quad \texttt{ma\_gap}_t \coloneqq \frac{P_t}{\texttt{ma\_20}_t} - 1. \tag{5}$$

$$\text{(RSI-14)} \quad \Delta_t = P_t - P_{t-1},\ G_t = \mathrm{mean}(\max(\Delta, 0))_{14},\ L_t = \mathrm{mean}(\max(-\Delta, 0))_{14}, \tag{6}$$

$$\mathrm{RSI}_{14,t} = 100 - \frac{100}{1 + \frac{G_t}{L_t}}. \tag{7}$$

$$\text{(Volume z-score)} \quad \texttt{vol\_z}_t \coloneqq \frac{V_t - \mu_{60}(V)}{\sigma_{60}(V)} \quad \text{(if Volume exists; else NaN)}. \tag{8}$$

$$\text{(Calendar)} \quad \texttt{dow}_t \in \{0, \ldots, 6\} \text{ (day-of-week)}. \tag{9}$$

**Reasoning.** These features summarize short-horizon momentum (returns), local trend (MAs, gaps), risk/scale (volatility, volume), and seasonality (weekday). They are widely interpretable and low-variance, aligning with ESN readout training.

## 5 Target Construction (Multi-Horizon)

We model forward log-returns at horizons $h \in \{1, 5, 20\}$:

$$\texttt{target\_h1}_t = r_{t+1}, \tag{10}$$

$$\texttt{target\_h5}_t = \sum_{k=1}^{5} r_{t+k}, \tag{11}$$

$$\texttt{target\_h20}_t = \sum_{k=1}^{20} r_{t+k}. \tag{12}$$

We drop samples with incomplete windows (warmup/edge NaNs), ensuring strict causality (*no* future leakage).

## 6 Walk-Forward Splits & Scaling

We align two core series (S&P 500 index GSPC and SPY) by intersecting their dates and define rolling folds with

$$\text{Train} = 2520 \text{ trading days } (\approx 10\text{y}), \quad \text{Test} = 252 \text{ days } (\approx 1\text{y}), \quad \text{Step} = 252 \text{ days}.$$

For each fold $k$, we fit a *StandardScaler* on training features only,

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

apply it to train/test, then store `train.csv`, `test.csv`, and `scaler.json` per fold.

**Algorithm 1** Walk-Forward Materialization (per fold $k$)

Compute common index $\mathcal{I} = \text{dates}(GSPC) \cap \text{dates}(SPY)$
**for** $k = 0, 1, \ldots$ **do**
  $\mathcal{T}_k \leftarrow \mathcal{I}[k \cdot 252 : k \cdot 252 + 2520], \quad \mathcal{S}_k \leftarrow \mathcal{I}[k \cdot 252 + 2520 : k \cdot 252 + 2520 + 252]$
  Drop rows with any missing target on $\mathcal{T}_k$ and $\mathcal{S}_k$
  Fit scaler on $\{x_t : t \in \mathcal{T}_k\}$; transform both train/test
  Save `fold_k/train.csv`, `fold_k/test.csv`, `fold_k/scaler.json`
**end for**

**Why walk-forward?** It mimics real-time deployment: models are trained on past data then evaluated on unseen future windows. Rolling the window tests temporal robustness and regime sensitivity. Train-only scaling prevents information leakage via global statistics.

## 7 Light Baseline Readout (for Sanity Checks)

To validate the pipeline, we fit a simple ridge regression on standardized features

$$\hat{y}_{t+h} = \boldsymbol{w}_h^\top \boldsymbol{z}_t + b_h, \quad \boldsymbol{w}_h = \arg\min_{\boldsymbol{w}} \sum_{t \in \mathcal{T}} (\hat{y}_{t+h} - y_{t+h})^2 + \lambda \|\boldsymbol{w}\|_2^2,$$

for $h \in \{1, 5, 20\}$. We report RMSE, MAE, $R^2$, and *directional accuracy* $\Pr[\text{sign}(\hat{y}) = \text{sign}(y)]$ on the test set. As an optional, clearly labeled *toy* diagnostic, we compute a sign-based strategy P&L (long if $\hat{r} > 0$, short otherwise) with per-trade cost (default 1 bp) to gauge whether forecasts carry usable directional signal; this is *not* a trading claim.

## 8 Data Products and Reproducibility

- **Raw data** (`data/raw/`): one CSV per ticker with canonical OHLCV.
- **Processed features** (`data/processed/`): `GSPC_features.csv`, `SPY_features.csv`.
- **Splits** (`data/splits/`): `splits.json` plus per-fold folders with `train.csv`, `test.csv`, `scaler.json`.
- **Diagnostics** (for fold 0): `preds_baseline.csv`, `analysis_summary.json/csv`.

**Determinism.** All transformations are pure functions of training windows, and fold artifacts contain the scaler means/scales for auditability.

## 9 Design Choices & Justifications

1. **Canonical OHLCV + `Adj Close`.** Keeping both `Close` and `Adj Close` preserves flexibility; modeling uses a single price stream PX to avoid mixing adjusted and unadjusted series.
2. **Feature set ($u_t$).** Low-variance, interpretable statistics reflecting momentum (returns), trend (MAs, gaps), risk (vol), liquidity/participation (volume), and seasonality (weekday). These stabilize readout learning and are compatible with ESN latent dynamics.
3. **Multi-horizon targets.** ESNs naturally support multi-task readouts; forecasting $\{1, 5, 20\}$-day returns covers intramonth horizons with distinct use-cases.
4. **Walk-forward protocol.** Temporal splits prevent look-ahead. Rolling windows probe generalization under regime shifts.
5. **Train-only scaling.** Prevents leakage; scaler parameters are saved per fold.
6. **Baseline ridge.** A linear, regularized readout is the correct sanity check before adding nonlinear/stateful models; it provides a lower bound for later ESN readouts.

## 10 Limitations and Next Steps

**Limitations.** (i) Daily frequency ignores intraday structure; (ii) single-asset modeling in the baseline (SPY) omits cross-asset signals, though other downloads enable future multi-asset features; (iii) sign-P&L is a toy diagnostic, not a robust strategy; (iv) no hyperparameter tuning yet.

**Next steps (Milestone 3/4).**

- Implement ESN reservoir and readout(s); verify echo-state property via spectral-radius and leak controls.
- Add exogenous $u_t$ from other tickers/macro (e.g., VIX, rates), possibly via lagged cross-features.
- Expand evaluation: probabilistic metrics (pinball/CRPS), Diebold–Mariano tests across folds.
- Run ablations: reservoir size/sparsity, spectral radius, leak rate, and feature subsets.

## Appendix: Feature Columns (per row)

| | |
|---|---|
| **Bookkeeping** | `Symbol` |
| **Raw (selected)** | `PX` (Adj Close if available, else Close), `Volume` |
| **Features** $u_t$ | `ret_1`, `ret_2`, `ret_5`, `vol_20`, `ma_10`, `ma_20`, `ma_gap`, `rsi_14`, `vol_z`, `dow` |
| **Targets** | `target_h1`, `target_h5`, `target_h20` |
| **Scaled (per fold)** | `z_ret_1`, ..., `z_dow` |

**Ethics/Disclaimer.** This project is academic. Nothing herein constitutes financial advice. Any backtest is illustrative only and includes explicit modeling simplifications.