LSTM for Competition Forecasting: An Intuitive Guide and Playbook

Calvin L.

1 Overview: What is an LSTM?

An LSTM (Long Short-Term Memory) is a recurrent neural network cell designed for sequential data. Intuitively, it is a learnable memory system that:

- *Remembers* useful information across many time steps.
- *Forgets* stale or distracting information.
- *Exposes* just enough information at each step to make a prediction.

It overcomes the vanishing/exploding gradient issues of vanilla RNNs by maintaining a *cell state* that carries information forward with minimal distortion.

2 Core Mechanics

2.1 Cell State (Long-Term Memory)

A persistent pathway that allows gradients and information to flow over long horizons with limited decay.

2.2 Gates (Learned Switches)

- **Forget gate:** decides what to drop from memory.
- **Input gate:** decides what new information to write.
- **Output gate:** decides what to reveal as the step output.

2.3 Why This Helps

- Enables learning *long-range dependencies* (e.g., weekly/seasonal effects, regime shifts).
- More stable training than plain RNNs due to controlled information flow.
- Flexible: can model nonlinear interactions and integrate many covariates.

3 When to Use LSTM (vs. ARIMA/ETS)

- Use LSTM when data are **time-ordered** and relationships are **nonlinear** or depend on **multiple lags** and exogenous drivers (weather, promos, holidays).
- Prefer classical baselines when the series is **short**, **stationary**, or well-captured by linear dynamics (ARIMA/ETS can be strong and fast).
- In practice, benchmark both and consider **ensembling**.

4 Competition Playbook

4.1 Data Preparation

- Scale features (z-score or min–max).
- Engineer calendar features (day-of-week, month, holiday flags) and rolling statistics (means, standard deviations).
- Create lag features for the target (e.g., $t-1$, $t-7$, $t-28$).

4.2 Windowing and Horizon

- Choose an input window W (e.g., 32, 64, 128 steps; or a few seasonal cycles).
- Choose an output horizon H (one-step or multi-step); use *direct* (predict all H at once) or *recursive* (roll forward).

4.3 Baselines First

- Seasonal naïve / last value, moving average, ETS/ARIMA.
- A strong tree model on lagged features (e.g., XGBoost) for a nonlinear baseline.

4.4 LSTM Baseline Recipe

- 1–2 LSTM (or GRU) layers with 64–128 units; small dense head.
- Dropout 0.1–0.3; gradient clipping (e.g., norm $= 1.0$).
- Start univariate (target only) with $W \approx$ one to two short cycles; expand to multivariate if exogenous signals matter.

4.5 Training and Validation

- Use **walk-forward** (rolling) validation to mimic real deployment.
- Early stopping on the competition metric (e.g., MAE, RMSE, SMAPE).
- Reduce learning rate on plateau; monitor overfitting.

4.6 Regularisation and Stability

- Dropout in recurrent and dense layers; weight decay if available.
- Careful batch sizes; ensure shuffling does not break temporal order.

4.7 Metrics and Reporting

- Match the competition metric; report both validation and test-period results.
- Provide diagnostic plots (predictions vs. truth; residual distribution; error by horizon).

4.8 Reproducibility

- Fix seeds; log model configs; save checkpoints; script the full pipeline.

5 Patterns and Extensions

5.1 Multivariate Forecasting

Concatenate target lags with exogenous features per time step; improves accuracy when drivers (e.g., weather, promotions) are predictive.

5.2 Seq2Seq vs. Direct/Recursive

- **Direct:** one model outputs all H steps; stable but more parameters.
- **Recursive:** predict next step and feed it back; lighter but can accumulate error.
- **Seq2Seq:** encoder–decoder for complex multi-step mapping; consider only if richer signals justify the complexity.

5.3 Classification and Anomaly Detection

Use LSTM to classify windows (e.g., up/down moves, anomaly flags) when the target is categorical or when early-event detection matters.

5.4 Ensembling

- Average multiple LSTMs (different seeds/windows) and blend with ARIMA/ETS/XGBoost.
- Often yields more robust leaderboard performance.

6 Pitfalls and Cautions

- **Data leakage:** keep future information out of training windows and scalers (fit scalers on training only).
- **Too little data:** LSTMs overfit easily on short series; prefer simpler baselines or GRU.
- **Validation mismatch:** use time-aware splits; random k-folds are invalid for sequences.
- **Over-complexity:** seq2seq/multivariate models can underperform if exogenous features are weak or noisy.

7 Doc-by-Doc Takeaways

7.1 Understanding LSTM (tutorial)

- **Key points:** Rationale for gates; cell state as a stable memory path; intuitive diagrams.
- **Use for us:** Clear language and figures to explain LSTM choices in the report and to guide window/gate intuition.

7.2 LSTM Intro (intuitive explainer)

- **Key points:** Step-by-step cell update; common variants (peephole, coupled gates, GRU).
- **Use for us:** Quick reference for explaining alternatives; try GRU as a lighter baseline.

7.3 Original LSTM Paper (Hochreiter–Schmidhuber)

- **Key points:** Constant-error flow; multiplicative gates; long-lag learning.
- **Use for us:** Authoritative citation to justify LSTM on long-range dependencies.

7.4 Stock Price Prediction Using ML and LSTM-Based DL (NIFTY 50)

- **Key points:** LSTM outperformed several ML baselines; best model was *univariate* with ˜one week of history; heavier seq2seq/multivariate did not help.
- **Use for us:** Start simple (univariate, short lookback) with walk-forward validation; scale up only if exogenous features are strong.

7.5 LSTM-based Stock Prediction Modeling and Analysis (VTI)

- **Key points:** For very short horizons (1–10 days), simple baselines matched or beat LSTM.
- **Use for us:** Include strong naïve/MA/linear/tree baselines; do not assume LSTM dominance on short-term targets.

7.6 LSTM for Financial Time Series (OMX30)

- **Key points:** Ensemble of LSTMs improved movement classification; trading backtests showed higher returns and lower volatility than simple comparators.
- **Use for us:** Consider an LSTM *ensemble* when the objective is directional accuracy or trading metrics.

8 Minimal Checklist

1. Define forecast horizon and metric; lock strong baselines.
2. Build sliding windows; fit scalers on training only.
3. Train a simple LSTM/GRU baseline with walk-forward validation.
4. Tune window size, hidden units, dropout, learning rate.
5. Add exogenous features if predictive; monitor leakage.
6. Ensemble with statistical and tree models; report gains.
7. Produce diagnostics and a clear methodology write-up.