

The Utility of Technical Indicators in LSTM-Based Stock Market Prediction

Jace Baptista-Allan
jacedba@uw.edu

Laís Omori
lomori@uw.edu

1 Introduction

The stock market is notorious for its extreme difficulty to predict. Traders rely heavily on technical indicators to capture market trends, which embeds in them the idea that these engineered features are essential parameters for stock market prediction. For this reason, algorithmic traders commonly feed deep learning models hand-engineered technical indicators (such as moving-average convergence/divergence (MACD), the relative strength index (RSI), Bollinger Bands, and so on), alongside raw price and data [Fischer and Krauss, 2018, Sezer et al., 2020]. This practice comes from the assumption that explicitly computed indicators supply predictive patterns that a sufficiently expressive sequence model cannot extract from the raw series itself. However, indicators are deterministic functions of past prices and volume; and Long short-term memory networks [Hochreiter and Schmidhuber, 1997] are a default choice for modeling financial time series. If an LSTM can already learn those transformations internally, the indicators could be redundant and may even hurt by inflating input dimensionality without adding signal.

So, we question whether explicit technical indicators provide predictive value over OHLCV for an LSTM directional classifier, and whether the answer depends on the asset class.

To answer this, we run a controlled input ablation. Three branches, B1 (raw OHLCV returns), B2 (indicators only), and B3 (combined), are trained with an identical LSTM architecture; only the input feature dimension changes ($F \in \{5, 13, 18\}$). Holding capacity, labels, splits, and the training recipe fixed means any difference in test performance is attributable to input information content rather than modeling power. We repeat the ablation across seven instruments spanning equities, cryptocurrency, and CME currency futures, yielding 21 trained models, and we rank branches by the Matthews Correlation Coefficient (MCC) [Matthews, 1975, Chicco and Jurman, 2020], which is robust to the class imbalance inherent in our three-class labels.

1.1 Contributions

Our contributions are:

- An ablation isolating the marginal value of technical indicators for LSTM directional forecasting, with per-ticker models aggregated to the asset-class level.
- A leakage-audited pipeline using volatility-adjusted Triple Barrier labeling [López de Prado, 2018], chronological splits with a boundary embargo, and train-only feature scaling.
- Empirical evidence that the raw OHLCV baseline is hard to beat with only indicators

2 Related Work

Recurrent models (LSTMs in particular) are widely applied to financial sequence prediction; Fischer and Krauss [2018] report LSTMs outperforming memory-free classifiers on large equity, and the survey of Sezer et al. [2020] documents the dominance of deep recurrent architectures in the 2005–2019 literature. Much of this work feeds engineered indicators to the networks without isolating their marginal contribution, and the evidence on whether the indicators actually help is mixed. On the supportive side, Dhokane and Agarwal [2024] augment OHLC with RSI, MACD, and Bollinger Bands and report lower price-forecasting error than OHLC alone, while Kah Him and Ching Pang [2023] add moving averages, EMA, RSI, and MACD as LSTM inputs and observe higher directional-classification accuracy. On the skeptical side, Saaïd [2026] report a $\sim 35\%$ increase in RMSE when a large indicator collection (RSI, stochastic oscillator, SMA, EMA, ADX, MACD, ATR, Bollinger Bands, etc.) is combined with OHLCV versus OHLCV alone, consistent with the broader observation that adding correlated indicators degrades generalization when they merely echo information already present in returns [Sezer et al., 2020]. These conflicting results motivate a controlled test: we hold the architecture fixed and vary only the input feature set, so the indicator contribution is measured directly, with an imbalance-aware metric (MCC) rather than RMSE on a regression target. Outside FX, our findings align more with the skeptical side.

To evaluate the market trend, The Triple Barrier Method [López de Prado, 2018] assigns labels based on which barrier, profit-take, stop-loss, a price path hits first within a fixed time-frame. Unlike standard next-day up/down labels, this produces three classes (Up, Down, Neutral) that are scaled to each asset’s volatility, making them more robust to noise. We adopt this approach and apply a boundary embargo to prevent forward-looking labels from leaking across train/validation/test splits.

To compare the performance results of the predicted labels, The Matthews Correlation Coefficient uses all cells of the confusion matrix and is markedly more informative under imbalance [Matthews, 1975, Chicco and Jurman, 2020]; we use it as our primary ranking metric and reserve macro-F1 for model selection. For significance we use McNemar’s exact test [McNemar, 1947, Dietterich, 1998], the standard paired test for comparing classifiers on a shared test set.

3 Methods

3.1 Data

We collect ten years (2014–2024) of daily OHLCV bars from Yahoo Finance for seven instruments grouped into three asset classes (Table 1). For FX we use CME currency *futures* (6E=F, 6B=F) rather than spot pairs because spot FX has no centralized volume on Yahoo (returning zero/NaN), which would degenerate the volume-based features in the FX asset class; the futures carry real exchange volume (median $\sim 118\text{k}$ and $\sim 60\text{k}$ contracts, respectively).

Table 1: Instrument universe and usable sample counts after indicator warm-up and differencing. One model is trained per (branch, ticker).

Asset class	Tickers	Rows / ticker
Equity	SPY, AAPL, MSFT	~ 2456
Crypto	BTC-USD, ETH-USD	3333 / 2184
FX (CME futures)	6E=F, 6B=F	2450 / 2446

3.2 Feature branches

The three branches share the same target and architecture and differ only in input features (Table 2). Raw price and volume levels are non-stationary; their mean and variance drift over time, and an LSTM fed absolute values will tend to learn a “one-day-lag” identity ($\hat{C}_t \approx C_{t-1}$) that may look accurate superficially, but encodes no actual directional signal. So, we mapped the five OHLCV columns to stationary, scale-invariant quantities:

$$\begin{aligned} \text{ret_close}_t &= \ln(C_t/C_{t-1}), & \text{ret_open}_t &= \ln(O_t/C_{t-1}), \\ \text{ret_high}_t &= \ln(H_t/C_t), & \text{ret_low}_t &= \ln(L_t/C_t), \\ \text{ret_volume}_t &= \ln(V_t/V_{t-1}), \end{aligned} \quad (1)$$

yielding the $F=5$ feature vector for B1. The rationale is threefold. (i) *Log returns* (ret_close, ret_volume) convert multiplicative price/volume growth into an additive, roughly mean-zero series that is comparable across price levels and across tickers, so the same scaler and model apply to a \$5 future and a \$400 ETF alike. (ii) *Intraday ratios* (ret_high, ret_low, relative to the same-day close) encode the bar’s high/low range, a proxy for realized intraday volatility, without embedding the absolute price. (iii) *Open-to-prev-close* (ret_open) captures the overnight gap. Because each transform differences against C_{t-1} or C_t , the first row of every series is undefined and is dropped before windowing; non-positive volume ratios (zero-volume days) are masked to keep the logarithm defined. B2 comprises 13 technical indicators computed with the `ta` library [Padiyal, 2018]: RSI, Stochastic %K/%D, MACD (line/signal/histogram, each divided by C_t to stationarize), ADX with $\pm$ DI, Bollinger %b and width, ATR (divided by C_t), and a rolling-mean-normalized OBV. Bounded oscillators are used directly; price-unit indicators are normalized as noted. B3 concatenates B1 and B2.

Table 2: Ablation branches. Only the input dimension F differs; the LSTM is identical across branches.

Branch	Inputs	F
B1	Raw OHLCV (stationary returns)	5
B2	Technical indicators only	13
B3	Combined (B1 + B2)	18

3.3 Labels: volatility-adjusted Triple Barrier Method

Targets are assigned by the Triple Barrier Method (TBM) [López de Prado, 2018], which labels each timestep by which of three barriers a forward price path touches first (Figure 1). For each t we set a barrier half-width that scales with *local* volatility,

$$\theta_t = k \sigma_t, \quad \sigma_t = \sqrt{\frac{1}{W-1} \sum_{i=0}^{W-1} (r_{t-i} - \bar{r}_t)^2}, \quad (2)$$

where $r_t = \ln(C_t/C_{t-1})$ is the log return, σ_t is its $W=20$ -day rolling sample standard deviation, and k is the width multiplier. The upper and lower price barriers are placed symmetrically in log space,

$$\text{Upper}_t = C_t e^{+\theta_t}, \quad \text{Lower}_t = C_t e^{-\theta_t}, \quad (3)$$

so a touch is equivalent to the cumulative forward log return exceeding $\pm\theta_t$ (i.e., a $\pm k\sigma_t$ move). The forward close path $C_{t+1}, \dots, C_{t+H}$ over horizon H trading days (the vertical *time* barrier) is scanned for the first touch:

$$y_t = \begin{cases} +1 & \text{if Upper}_t \text{ is touched first (up),} \\ -1 & \text{if Lower}_t \text{ is touched first (down),} \\ 0 & \text{if neither barrier is touched within } H \text{ (neutral).} \end{cases} \quad (4)$$

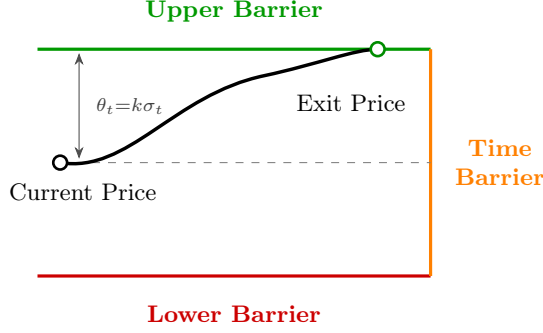


Figure 1: Triple Barrier labeling for one timestep. From the current price C_t , symmetric upper/lower barriers are set at $C_t e^{\pm\theta_t}$ with $\theta_t = k\sigma_t$; a vertical time barrier closes the window after H days. Here, the label is +1 (up).

Tying θ_t to σ_t rather than using a fixed percentage barrier makes the label difficulty self-normalizing across assets and over time: a $k\sigma$ move represents the same statistical surprise for a placid FX future as for volatile crypto. A fixed barrier would over-trigger the directional classes in high-volatility regimes and collapse them to neutral in calm ones, confounding the asset-class comparison that is the point of this study.

The two hyperparameters jointly control the class balance and are coupled: larger k or smaller H both make a touch less likely and inflate the neutral share. Holding $W=20$ fixed, we swept each while measuring the pooled (all-ticker) class distribution against the design target of no class more skewed than 40/40/20 (down/up/neutral), i.e. neutral $\approx 20\%$ (Table 3). At the original $H=30$, almost every path touches a barrier and neutral collapses to $\sim 1\%$, which class weighting cannot rescue. Shortening to $H=7$ restores a usable three-class balance (0.35/0.23/0.42, per-ticker neutral in $[0.19, 0.29]$), and among the width multipliers $k=1.5$ lands closest to the target while keeping the directional classes in the majority—appropriate since the study targets directional breakouts, not ranging regimes. We therefore fix $k=1.5$, $H=7$, $W=20$.

Table 3: TBM class-balance calibration (pooled over all seven tickers). Top: horizon sweep at $k=1.5$; bottom: width sweep at $H=7$. The chosen setting ($k=1.5$, $H=7$) is in **bold**; it is the configuration closest to the 40/40/20 guideline with directional classes still dominant.

Setting	Down	Neutral	Up
<i>Horizon H (at $k = 1.5$)</i>			
$H = 30$	0.44	0.01	0.55
$H = 15$	0.42	0.06	0.52
$H = 10$	0.39	0.13	0.48
$H = 7$	0.35	0.23	0.42
$H = 5$	0.30	0.35	0.36
<i>Width k (at $H = 7$)</i>			
$k = 1.0$	0.42	0.08	0.50
$k = 1.25$	0.39	0.15	0.46
$k = 1.5$	0.35	0.23	0.42
$k = 1.75$	0.31	0.31	0.38
$k = 2.0$	0.28	0.39	0.34

3.4 Splits, embargo, and scaling

Each ticker is split chronologically by row count into 70% train, 15% validation, and 15% test, with no shuffling. Because TBM labels look forward up to H rows, we apply a López de Prado embargo of $H = 7$ rows at each split boundary so a label never resolves using prices from the next split. A **StandardScaler** is fit on training rows only and applied to validation and test. Sliding windows are built *within* each split (no cross-boundary windows); window i spans rows $[i, i+w-1]$ and predicts the label at its last row. The first 60 rows of each series are discarded (the MACD signal, an EMA of an EMA, is slowest to converge).

3.5 Architecture

All branches use an identical two-layer stacked LSTM (Table 4). The first layer has H units and returns the full sequence; the second has $H/2$ units and returns its final hidden state, which passes through dropout, a 64-unit ReLU dense layer, batch normalization [Ioffe and Szegedy, 2015], and a 3-way linear head producing class logits. Regularization uses inter-layer dropout [Srivastava et al., 2014]; Keras-style recurrent dropout is omitted because PyTorch’s `nn.LSTM` does not support it natively. Figure 2 details the shared computation graph; only the input width F differs across the three branches.

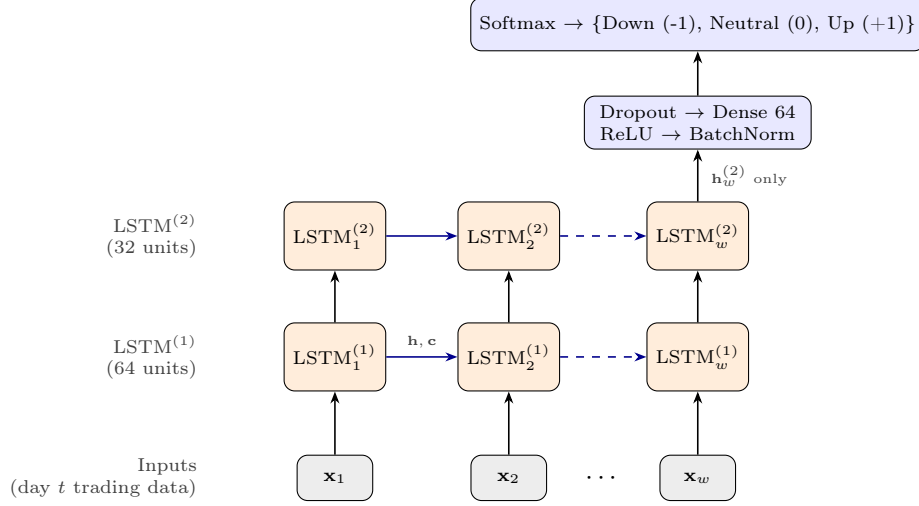


Figure 2: Shared two-layer stacked LSTM classifier, identical across branches B1/B2/B3; only the input width $F \in \{5, 13, 18\}$ changes, isolating input information content from model capacity. The network is unrolled through time: each box is the same layer reused at every timestep, solid blue arrows are the recurrent carry ($\mathbf{h}, \mathbf{c}$), and dashed arrows abbreviate the remaining timesteps up to $w=30$. Only the final layer-2 state $\mathbf{h}_w^{(2)}$ feeds the classification head.

3.6 Training

Models are trained with Adam [Kingma and Ba, 2014] (initial $\text{lr} = 10^{-3}$) minimizing class-weighted cross-entropy, with weights $w_c = N/(3N_c)$ computed on the training partition to counter label imbalance without resampling (which would violate temporal order). The learning rate is halved on a validation-loss plateau (patience 5). Training stops early on validation loss (with minimum 10 epochs). We train one model per (branch, ticker) pair, totaling $3 \times 7 = 21$ models, rather than pooling tickers, so that windows do not span instrument boundaries.

Table 4: Architecture and configuration (held same for all 21 runs; selected by the sweep in Section 4.1).

Setting	Value
LSTM layer 1 / layer 2 units	$H = 64$ / $H/2 = 32$
Lookback window w	30 days
Dropout	0.1
Dense units	64 (ReLU) $\rightarrow$ BatchNorm
Output	3-way softmax
Optimizer	Adam, $\text{lr} = 10^{-3}$
LR schedule	$\times 0.5$ on val-loss plateau (patience 5)
Early stopping	val loss, patience 10
Checkpoint criterion	best val macro-F1
Batch size / max epochs	64 / 100
Loss	class-weighted cross-entropy
Seed	42 (single)
Framework	PyTorch (CUDA)

3.7 Evaluation

Our ranking metric is the multiclass MCC, which incorporates all nine cells of the 3×3 confusion matrix and is robust to imbalance and to class collapse [Matthews, 1975, Chicco and Jurman, 2020]. For a K -class confusion matrix C (here $K = 3$), the generalized coefficient is

$$\text{MCC} = \frac{c s - \sum_k p_k t_k}{\sqrt{(s^2 - \sum_k p_k^2)(s^2 - \sum_k t_k^2)}}, \quad (5)$$

where $s = \sum_{i,j} C_{ij}$ is the total number of samples, $c = \sum_k C_{kk}$ the number correctly classified, $t_k = \sum_i C_{ik}$ the number of true occurrences of class k , and $p_k = \sum_i C_{ki}$ the number of times class k was predicted. Our rationale for this is that because Equation (5) uses both the predicted and the true marginals (p_k and t_k), it cannot be inflated by exploiting a dominant class (accuracy and macro-F1 can remain deceptively high, even if a classifier collapses into the dominant class).

MCC ranges from $[-1, +1]$, where $+1$ is perfect agreement, 0 is no better than random, and negative values indicate predictions anti-correlated with the truth (worse than chance). This is why most of our scores cluster just above 0 : daily directional prediction is close to the no-skill boundary, and the ablation is read from *relative* MCC differences between branches rather than their absolute level. Macro-F1 is reported as a secondary metric and is the criterion used for model selection; test cross-entropy, one-vs-rest AUC, and expected calibration error are reported but not used for ranking.¹

3.8 Estimating significance

Pairwise branch differences are assessed with McNemar’s exact (binomial) test on the paired correct/incorrect outcomes of two branches on the same test windows [McNemar, 1947, Dietterich, 1998]: with discordant counts b_{01} (A right, B wrong) and b_{10} (A wrong, B right), $n = b_{01} + b_{10}$, and $m = \min(b_{01}, b_{10})$, the two-sided p -value is

$$p = \min(1, 2 F_{\text{Binom}}(m \mid n, 0.5)), \quad (6)$$

significant at $\alpha = 0.05$.

4 Experiments and Results

4.1 Architecture selection

We ran a one-time sweep over hidden units $\{32, 64, 128\}$, window $\{20, 30, 45\}$, dropout $\{0.1, 0.2, 0.3\}$, and dense units $\{32, 64\}$ on the highest-dimensional branch (B3) and the most-studied ticker (SPY), ranking by validation macro-F1. The winning configuration, $H=64$, $w=30$, dropout 0.1 , dense 64 (validation macro-F1 0.4624), was frozen for all 21 runs. The landscape is flat (the top ten configurations fall within ~ 0.04 macro-F1) so the ablation conclusions are unlikely to be an artifact of architecture choice, which is the reason for fixing a single shared model as an experimental control.

4.2 Primary ablation: MCC by asset class

Table 5 reports asset-class mean MCC and macro-F1 ($\pm$ standard deviation across the tickers in each class), and Figure 3 visualizes the same results. On equities and crypto, raw OHLCV (B1) achieves the highest mean MCC (0.096 and 0.039); adding indicators is at best neutral and often slightly harmful (equity B3 0.067 ; crypto B2 0.001 , B3 -0.009). FX is the sole asset class where a feature-augmented branch leads, with B3 highest at 0.032 . Absolute MCC values are low throughout (mostly 0 – 0.2 , occasionally negative), which is

¹It’s worth noting that selecting on macro-F1 while ranking on MCC was a deliberate choice in our procedure, NOT an inconsistency. Macro-F1 is a smoother, per-class-balanced signal well suited to guiding the sweep and early stopping, whereas the harder-to-game MCC is a more appropriate arbiter. Reporting both also lets us check whether they agree (Table 5), which strengthens the conclusions drawn from their agreement.

expected for daily directional prediction of liquid markets near the efficient-market no-skill boundary [Fama, 1970]; the contribution is the *relative* ablation pattern rather than the absolute level.

Table 5: Asset-class mean test MCC and macro-F1 (mean $\pm$ std across tickers in the class). Best MCC per asset class in **bold**. Std is cross-ticker dispersion over n tickers.

Asset class (n)	Branch	MCC	Macro-F1
Equity (3)	B1	0.096 ± 0.087	0.396 ± 0.053
	B2	0.093 ± 0.105	0.377 ± 0.100
	B3	0.067 ± 0.076	0.366 ± 0.061
Crypto (2)	B1	0.039 ± 0.005	0.298 ± 0.021
	B2	0.001 ± 0.026	0.322 ± 0.014
	B3	-0.009 ± 0.012	0.294 ± 0.001
FX (2)	B1	0.009 ± 0.045	0.338 ± 0.029
	B2	-0.005 ± 0.115	0.324 ± 0.051
	B3	0.032 ± 0.119	0.347 ± 0.069

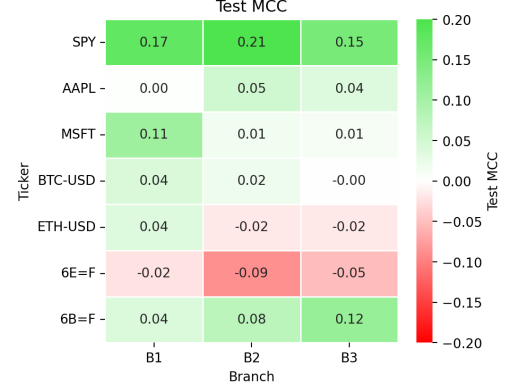


Figure 3: Summary test-MCC heatmap (branch $\times$ tickers).

The per-ticker breakdown (Table 6) shows the aggregates are driven by a few instruments. The FX edge is almost entirely the GBP future (6B=F, B3 = +0.116), while the EUR future (6E=F) is at or below chance for all branches. SPY is the only equity where B2 leads (0.212 vs. B1 0.175), but this is not statistically robust (Section 4.4).

Table 6: Per-ticker test MCC for all 21 runs. Best branch per ticker in **bold**. No run exhibited class collapse.

Asset class	Ticker	B1	B2	B3
Equity	SPY	0.175	0.212	0.153
	AAPL	0.003	0.053	0.036
	MSFT	0.109	0.014	0.011
Crypto	BTC-USD	0.043	0.019	-0.001
	ETH-USD	0.036	-0.018	-0.018
FX	6E=F	-0.023	-0.086	-0.052
	6B=F	0.041	0.076	0.116

4.3 Failure-mode check

A central risk for imbalanced three-class problems is collapse onto the neutral class. The row-normalized confusion matrices (Figure 4) confirm that all branches predict all three classes. No run collapsed, which validates the TBM horizon recalibration and the class-weighted loss. Errors concentrate in the expected place: directional classes (up/down) are harder than neutral, and confusion is mostly between adjacent barrier labels rather than the more catastrophic up/down confusion.

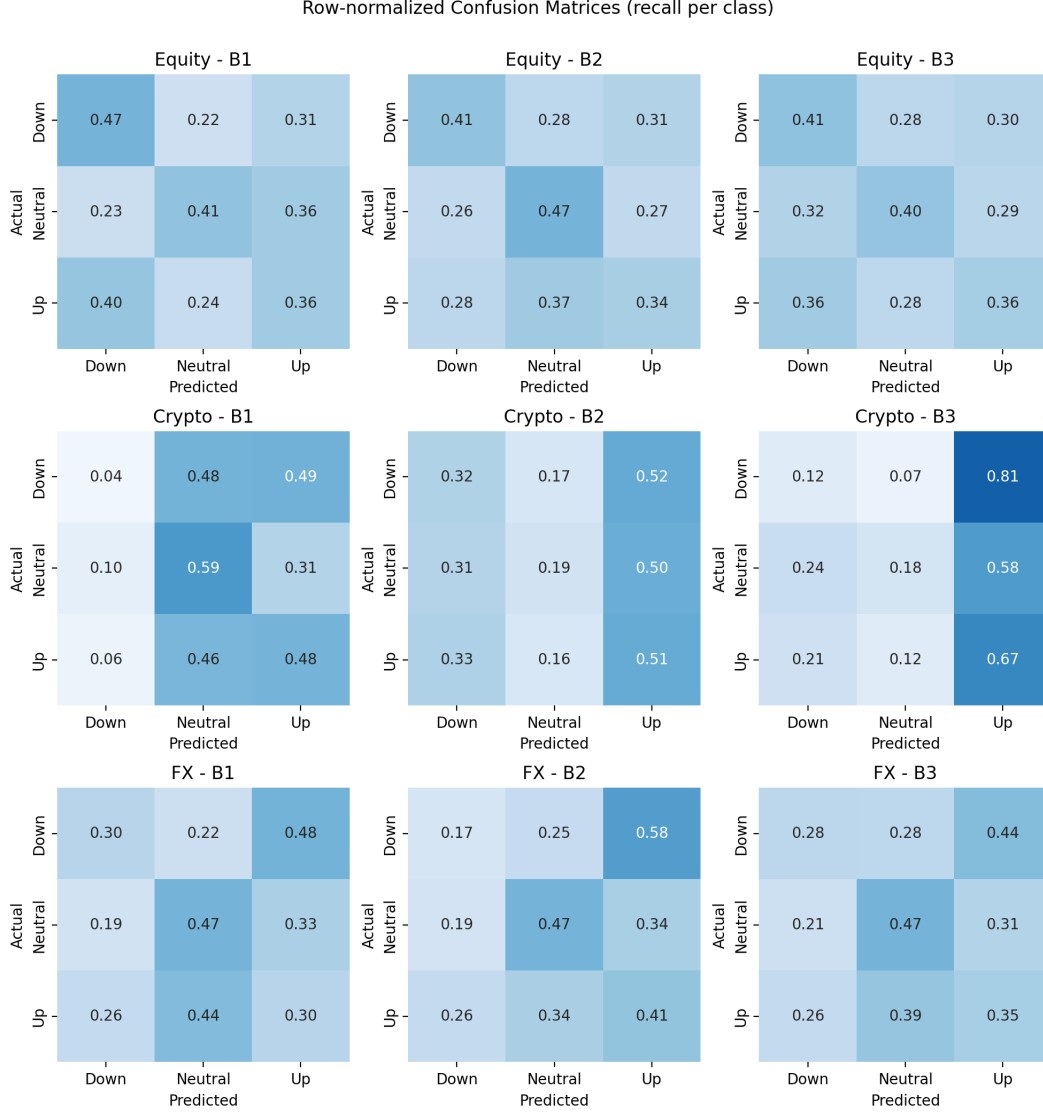


Figure 4: Row-normalized confusion matrices (recall per class) by asset class (rows) and branch (columns). No class collapse; the diagonal is populated for every configuration.

4.4 Statistical significance

Across the 21 pairwise McNemar tests ($7 \text{ tickers} \times 3 \text{ branch pairs}$), only 3 reach significance at $\alpha = 0.05$ (Table 7). Two favor raw OHLCV on MSFT (B1 over both B2 and B3), and one is an intra-augmented comparison on SPY (B2 over B3). Critically, no FX comparison is statistically significant, so the apparent FX advantage of B3 is not statistically supported on these test windows. The dominant outcome is that the three branches are statistically indistinguishable on most instruments.

Table 7: Full McNemar exact-test results for all 21 branch pairs (7 tickers $\times$ 3 pairs). b_{01} counts windows where the first branch in the pair is correct and the second is wrong; b_{10} is the reverse; $n = b_{01} + b_{10}$. **Bold** p -values are significant at $\alpha = 0.05$ (3 of 21).

Ticker	Pair	b_{01}	b_{10}	n	p
SPY	B1 vs. B2	29	43	72	0.1249
SPY	B1 vs. B3	37	32	69	0.6305
SPY	B2 vs. B3	46	27	73	0.0344
AAPL	B1 vs. B2	64	65	129	1.0000
AAPL	B1 vs. B3	54	60	114	0.6398
AAPL	B2 vs. B3	37	42	79	0.6530
MSFT	B1 vs. B2	73	42	115	0.0049
MSFT	B1 vs. B3	61	35	96	0.0103
MSFT	B2 vs. B3	24	29	53	0.5831
BTC-USD	B1 vs. B2	119	96	215	0.1333
BTC-USD	B1 vs. B3	114	91	205	0.1242
BTC-USD	B2 vs. B3	88	88	176	1.0000
ETH-USD	B1 vs. B2	43	33	76	0.3019
ETH-USD	B1 vs. B3	41	32	73	0.3492
ETH-USD	B2 vs. B3	31	32	63	1.0000
6E=F	B1 vs. B2	63	58	121	0.7163
6E=F	B1 vs. B3	47	42	89	0.6718
6E=F	B2 vs. B3	34	34	68	1.0000
6B=F	B1 vs. B2	57	62	119	0.7140
6B=F	B1 vs. B3	51	65	116	0.2273
6B=F	B2 vs. B3	19	28	47	0.2430

4.5 Analysis

Q1: Do indicators add value over OHLCV (B3 vs. B1)? No for equity ($\Delta\text{MCC} = -0.029$) and crypto (-0.048); a marginal $+0.023$ for FX. The LSTM seems to recover equivalent signal from raw returns, so adding indicators mostly adds input dimensions without information (supporting the redundancy view).

Q2: Can indicators alone match raw OHLCV (B2 vs. B1)? No: $B2 \leq B1$ in mean MCC for every asset class (equity -0.003 , crypto -0.038 , FX -0.014). Stripping the raw return series discards timing information that the indicators do not fully encode.

Q3: Does the ablation pattern differ by asset class? Partly. Equity and crypto favor B1; FX is the only asset class where a feature-augmented branch (B3) leads on mean MCC, with a per-ticker peak at 6B=F B3 ($+0.116$). That pattern is not statistically robust: 6E=F is near or below chance for all branches, and no FX McNemar comparison is significant.

5 Discussion and Conclusion

5.1 Interpretations

The headline is a largely **null ablation**: for an LSTM with this labeling and evaluation setup, raw stationary OHLCV is a strong baseline that explicit technical indicators do not reliably improve. Outside FX, indicators are at best neutral and sometimes mildly harmful, consistent with the view that an LSTM can internally approximate the deterministic transformations the indicators encode, so that supplying them adds dimensionality without information. The one asset-class-specific pattern we observe is B3 leading on FX, driven mainly by 6B=F, but that difference is weak and not statistically supported, so we treat it as an exploratory observation rather than a confirmed finding.

5.2 Limitations

Several caveats bound our conclusions. (i) Per-ticker test sets are small ($n_{\text{test}} \in [280, 450]$ windows), making individual MCC estimates noisy and explaining the negative values on some runs. (ii) Each asset class contains only 2–3 hand-picked tickers, so class-level means and their dispersion are unstable and not a random sample of the universe. (iii) We evaluate on a single chronological test window; a single market-regime shift over that period could dominate results. (iv) There is no external state-of-the-art baseline. The branches are internal controls, so we make only relative claims, not absolute-performance ones.

Two methodological points proved decisive. First, label design matters just as much as model design: the initial $H = 30$ horizon on TBM collapsed the neutral 0 class to $\sim 1\%$ of occurrences, and would have made our three-class metrics meaningless; recalibrating to $H = 7$ was prerequisite for interpretable results. Second, metric choice changes the story; SPY’s B2 looks best by macro-F1 and MCC yet is not significant under McNemar, demonstrating why we treat paired significance as the final arbiter.

6 Future Works

The most valuable extensions are:

- Testing with financial sentiment encoders on daily headlines (also available through `yfinance`) as input
- Multi-seed replication with aligned selection criteria to obtain real confidence intervals
- Walk-forward validation across several expanding windows to test stability across market periods
- A within-B2 feature-family ablation (remove volatility, then momentum, then trend) to localize any FX signal
- Pooling more instruments per asset class to tighten class-level estimates.

7 Conclusion

We asked whether explicit technical indicators add marginal value to an LSTM directional classifier and whether the answer is asset-class-dependent. Across 21 controlled runs in three asset classes, raw OHLCV matched or beat indicator-augmented inputs everywhere except FX, where a weak, non-significant B3 advantage appeared; only 3 of 21 pairwise differences were significant. For this setup, an LSTM on stationary returns is a baseline that indicators do not dependably improve. This is a result that contests the assumed value of engineered features.

References

- Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, 2020.
- R. M. Dhokane and Sohit Agarwal. Stock price prediction using technical indicators and OHLC features. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3):1169–1176, 2024. <https://ijisae.org/index.php/IJISAE/article/view/5396>.
- Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
- Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.
- Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.
- Kah Him and Ching Pang. Technical indicators as LSTM input features for stock directional movement prediction. *INTI Journal*, 2023. <https://iuojs.intimal.edu.my/index.php/intijournal/article/view/132>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Marcos López de Prado. *Advances in Financial Machine Learning*. John Wiley & Sons, 2018.
- Brian W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.
- Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
- Darío López Padial. Technical analysis library in Python (ta). <https://github.com/bukosabino/ta>, 2018. Accessed 2026.
- Saaïd. On the redundancy of technical indicators for deep stock forecasting. Preprints.org, 2026. <https://www.preprints.org/manuscript/202602.0692>.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181, 2020.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.