

# Dispersion-Matched Forecast Comparison: Separating Skill from Shrinkage in the Evaluation of Financial Prediction Models

Shelly Suman Khuntia

KIIT Deemed to be University

Ajaya Kumar Parida

[ajaya.paridafcs@kiit.ac.in](mailto:ajaya.paridafcs@kiit.ac.in)

KIIT Deemed to be University

Satya Narayan Misra

KIIT Deemed to be University

Santisudha Panigrahi

KIIT Deemed to be University

---

## Research Article

**Keywords:** Financial market prediction, deep learning, multi-source data fusion, gated fusion mechanism, transformer, explainable AI, sentiment analysis, FinBERT, technical indicators, attention mechanism

**Posted Date:** October 12th, 2026

**DOI:** <https://doi.org/10.21203/rs.3.rs-11103722/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# Dispersion-Matched Forecast Comparison: Separating Skill from Shrinkage in the Evaluation of Financial Prediction Models

Shelly Suman Khuntia<sup>1</sup>, Ajaya Kumar Parida<sup>2</sup>, Satya Narayan Misra<sup>1</sup> and Santisudha Panigrahi<sup>2</sup>

<sup>1</sup>School of Management, KIIT Deemed to be University,  
Bhubaneswar-751024, Odisha, India

<sup>2</sup>School of Computer Engineering, KIIT Deemed to be University,  
Bhubaneswar-751024, Odisha, India

## Abstract

Model comparison in financial forecasting is conventionally settled by ranking competitors on squared error and certifying the margin with a Diebold–Mariano test. We show that this protocol can rank forecasters that have no predictive skill at all, derive the closed form that explains why, and give a correction. Under a null of no skill, a model’s mean squared error is determined entirely by the variance of the target, the squared ratio of forecast dispersion to target dispersion, and the squared forecast bias. The ranking by error is therefore exactly the ranking by shrinkage, and carries no information about architecture, inputs or training. Because the expected loss differential between two such forecasters is non-zero whenever their shrinkage differs, the expected Diebold–Mariano statistic grows with the square root of the sample size: enlarging the evaluation window strengthens the apparent evidence of superiority rather than exposing its absence. Our correction, dispersion matching, rescales every forecast to the dispersion of the target before testing, after which error depends only on the correlation between forecast and outcome, so the comparison becomes a test of differential skill. It requires no retraining. On ten US and Indian equity targets over 2014 to 2024 with nine trained forecasters, the closed form predicts realised error to within 0.71 percent, correlating at 0.9996 with the truth; the predicted growth rate is confirmed empirically; and matching collapses seven of eight pairwise results from overwhelming significance to none. A forecast of zero attains lower error than all nine models, and the finding holds

across five rolling-origin folds spanning 2018 to 2024. We recommend reporting an unconditional benchmark and the dispersion ratio alongside any forecast comparison.

*Keywords:* Financial market prediction; deep learning; multi-source data fusion; gated fusion mechanism; transformer; explainable AI; sentiment analysis; FinBERT; technical indicators; attention mechanism

# 1 Introduction

Suppose a new architecture for daily equity return forecasting is compared against a panel of recurrent, convolutional, and attention-based baselines. It attains the lowest root-mean-square error. A Diebold–Mariano test, run per target with a heteroskedasticity- and autocorrelation-consistent variance, combined across targets by a method valid under cross-sectional dependence, and corrected for multiple comparisons, returns  $p < 10^{-14}$ . Every element of that protocol is correct. What has been established?

This paper’s answer is: possibly nothing. We show that the protocol just described can rank forecasters none of which has any predictive skill, that the ranking it produces in that case is a measurement of something else entirely, and that the apparent significance grows stronger the more data one collects. We then give a correction that costs two lines of code.

## 1.1 The mechanism

Where a target is close to unpredictable, the squared-error-optimal forecast is its conditional mean, which for daily log returns is approximately zero. A model without exploitable signal therefore minimises its training loss by collapsing toward a constant, and how completely it collapses varies with architecture, capacity, and regularisation. Let  $\lambda_i$  denote the ratio of forecast to target dispersion. Section 3 shows that under a no-skill null,

$$\text{MSE}_i = \sigma^2(1 + \lambda_i^2) + b_i^2,$$

which is strictly increasing in  $\lambda_i$ . The error ranking is the shrinkage ranking. It contains no information about architecture, inputs, or training, because the expression contains none.

Two consequences follow. The expected loss differential between two skill-less forecasters is non-zero whenever their shrinkage differs, so the expected Diebold–Mariano statistic grows as  $\sqrt{T}$ . A larger evaluation window makes a spurious ranking look more decisive, not less. And because mean squared error confounds dispersion with correlation, removing the first isolates the second: rescaling all forecasts to common dispersion leaves  $\text{MSE} = 2\sigma^2(1 - \rho)$ , a function of skill alone.

## 1.2 Contributions

1. **A closed form for the error ranking under a no-skill null** (Proposition 1), which uses nothing about the model and assumes it is worthless, yet predicts the realised error of nine trained forecasters to 0.71% mean relative error ( $r = 0.9996$ ).
2. **Significance that grows with the sample** (Corollary 1):  $|\text{DM}|$  diverges as  $\sqrt{T}$  between skill-less forecasters with unequal shrinkage. Confirmed empirically at  $T^{0.454}$  ( $R^2 = 0.98$ ) while skill measures stay flat at zero.
3. **Dispersion-matched comparison** (Proposition 2), a correction requiring no retraining, after which the test addresses differential skill rather than differential shrinkage. Applied to our panel, seven of eight combined  $p$ -values move from as low as  $9.9 \times 10^{-15}$  to between 0.78 and 0.97, with win records falling from 10/0 to near parity. We also identify the method’s boundary condition: the rescaling is ill-conditioned when a forecaster’s dispersion approaches zero, which accounts for the eighth.
4. **A worked demonstration on a representative testbed**. Nine forecasters on ten US and Indian targets over 2014–2024 fail to beat the trivial forecast  $\hat{r}_{t+1} = 0$ ; the best is indistinguishable from it ( $p = 0.97$ ); a Model Confidence Set retains only those two of ten forecasts. The result holds across five rolling-origin folds including the COVID-19 crash. It is not a pipeline artefact: the same architecture recovers a planted signal at  $r = 0.97$ .
5. **A correction to the aggregation step**. Fisher’s method, the usual way of pooling per-target  $p$ -values, assumes independence that equity panels do not have. Simulated at equicorrelation  $\rho = 0.6$  it rejects a true null 14.3% of the time at a nominal 5%; the Cauchy combination test holds size at 5.9%. We use the latter throughout.
6. **Two diagnostics we recommend reporting** alongside any forecast comparison: an unconditional benchmark, which establishes that skill exists at all, and the dispersion ratio  $\lambda$ , which reveals whether a ranking is measuring it. Both are one line; either would have caught this.

## 1.3 What we do not claim

We do not claim that daily equity returns are unpredictable in general, that no architecture could do better than those we trained, or that multi-source fusion is without merit. Our claim concerns what a comparison protocol can and cannot establish, demonstrated on one panel at daily frequency. The sentiment channel of the testbed architecture is absent for reasons given in Section 6.

We also note what the correction does not do. Dispersion matching tests whether one forecaster has more skill than another; it cannot establish that either has any. That is what the unconditional benchmark is for, and we recommend both.

The paper proceeds as follows. Section 2 reviews forecast evaluation and the benchmark literature. Section 3 derives the three results. Section 4 describes the testbed and Section 6 the setup. Section 7 reports the validation, Section 8 a permutation control on interpretability, and Section 9 the implications for practice.

## 2 Literature Review

We organise the relevant literature into six themes, taking a roughly chronological path from classical statistical modelling through to the state-space and LLM work that has appeared in the last year. A short summary of the gaps we identified, and that MFGate-Net is designed to close, wraps up the section before we move into the methodology.

### 2.1 Classical and Machine Learning Approaches

The pre-deep-learning literature leaned heavily on two families of tools. Parametric time-series models like ARIMA (Box et al., 2015) and GARCH (Bollerslev, 1986) provided useful baselines for level and volatility forecasting, but they rest on linearity and stationarity assumptions that equity data does not obviously satisfy. Running alongside that, the Efficient Market Hypothesis of Fama (1970) put a theoretical bound on how much predictability should be expected beyond a random walk; the behavioural-finance literature that followed has catalogued plenty of exceptions to it, which is partly why empirical work on prediction has continued (Baker et al., 2016).

Shallow and non-linear methods weaken the linearity assumption while still retaining it. Kim (2003) showed that SVMs can surpass a random walk classification in directional analysis; other studies have used Random Forests (2016) and boosting (Chen and Guestrin, 2016) to introduce averaging through ensembles. The common problem with these techniques is double: they need careful hand-engineering for feature selection, and they model sequence data only incidentally (Sezer et al., 2020).

### 2.2 Deep Learning for Financial Prediction

Gated recurrent networks took naturally to daily price sequences, with strong LSTM results reported on index returns (Fischer and Krauss, 2018; Selvin et al., 2017; Nelson et al., 2017); convolutional models captured local, shift-invariant structure over indicator panels (Hoseinzade and Haratizadeh, 2019; Sezer and Ozbayoglu, 2018); and self-attention subsequently migrated from language modelling into stock-movement prediction (Ding

et al., 2020; Li et al., 2022; Gao et al., 2023). Because the three families capture different structure, hybrids stacking them became common (Lu et al., 2020; Wang et al., 2022; Zhang et al., 2023). What matters for this paper is not the taxonomy but the evaluation convention these studies share: a new architecture is compared against a panel of trained competitors on squared error, and the margin is certified by a Diebold–Mariano test. Section 3 examines what that convention can establish.

## 2.3 Multi-Source Data Fusion in Finance

Acknowledgement that markets respond to more than prices has motivated multi-source models. Wu et al. (Wu et al., 2022) fuse historical prices, technical indicators, and CNN-extracted sentiment from posts and news into an LSTM. Zhang et al. (Zhang et al., 2022) proposed TEANet, a Transformer-based multi-level attention scheme over price and text. Sawhney et al. (Sawhney et al., 2021) took the problem to a graph setting with a multiplex integration of price, tweet, and news data.

Most of this work falls into one of two fusion patterns: early (concatenate raw features, then encode) or late (encode separately, then average decisions). Both have known downsides (Baltrušaitis et al., 2019): early fusion blocks modality-specific representations, and late fusion cannot model cross-modal interactions. Intermediate fusion with learnable gating (Arevalo et al., 2017) has had traction in multimedia and in healthcare, but it remains under-explored for financial time series.

## 2.4 Explainability in Financial AI

Interpretability used to be something financial-ML papers mentioned in passing. That has shifted as deep models have moved closer to production. The toolkit for explaining an arbitrary model is reasonably well-populated: Shapley-value attributions through SHAP (Lundberg and Lee, 2017), local linear surrogates through LIME (Ribeiro et al., 2016), and the attention weights that some recent architectures (Bahdanau et al., 2015) already expose.

The gap we see isn’t in the *availability* of those tools. It’s in how they get used. Weber et al. (Weber et al., 2024) surveyed over 150 papers in financial machine learning and found that only a small minority included any interpretability analysis at all; among those that did, most stopped at a single global feature-importance ranking and did not look at how the picture changes with the market regime. Given how clearly regimes do shift in equity data (Hamilton, 1989; Ang and Timmermann, 2012), we think that’s a noticeable blind spot, and it is one of the things this paper sets out to close.

## 2.5 Recent Developments (2023–2024)

A handful of newer threads are worth flagging, mostly to position this paper relative to where the field is currently headed.

The first is the move toward *large language models* adapted for finance. BloombergGPT (Wu et al., 2023), FinGPT, and a small number of instruction-tuned variants now offer richer textual analysis than a plain classifier of the FinBERT vintage can produce. What remains unsettled is the best way to turn LLM-derived features into quantitatively useful inputs next to price and macro series. The published evidence on that specific integration question is still thin.

A second is *graph neural networks*. Treating inter-stock relationships as graphs (Sawhney et al., 2021) picks up industry-level correlation and cross-asset flow that a single-asset model has no way to see. It’s a complement to, not a replacement for, time-series modelling.

Two further threads are worth mentioning more briefly. Self-supervised pre-training of financial time-series models, using masked reconstruction or contrastive objectives, has been tried in a handful of papers, with empirical gains that have been uneven. And state-space architectures in the Mamba family (Gu and Dao, 2023) offer a linear-time alternative to attention; their application to finance is still sparse.

Our contribution is orthogonal to these directions. It is a principled multi-source fusion architecture with built-in interpretability, and it is compatible with the newer techniques above rather than in competition with them.

## 2.6 Out-of-Sample Benchmarks and Negative Results

The concern that a forecasting model may fail to beat a trivial benchmark is not new to financial econometrics. Welch and Goyal (2008) showed that a long list of equity-premium predictors, most of which performed well in sample, failed to beat the prevailing historical mean out of sample, and Campbell and Thompson (2008) examined the conditions under which modest predictability can still be economically meaningful. The random-walk benchmark occupies a similar role in exchange-rate forecasting (Meese and Rogoff, 1983). Our contribution is not to introduce this benchmark but to observe that the recent deep-learning literature on daily equity prediction has largely stopped reporting it, and to show what that omission conceals: a leaderboard that ranks models by shrinkage while every entry on it fails the benchmark the older literature would have applied first.

### 3 Skill, Shrinkage, and Dispersion-Matched Comparison

This section derives the paper’s three analytical results: why error rankings under a no-skill null measure shrinkage rather than accuracy, why the resulting significance strengthens with sample size, and how to remove the shrinkage channel so that a comparison tests skill. Each is stated as a claim about the comparison protocol and is independent of any particular model. Sections 4–7 then validate all three on a representative testbed.

#### 3.1 Mean squared error under a no-skill null

Let  $r$  denote the target with mean  $\mu$  and variance  $\sigma^2$ , and let  $\hat{r}_i$  be the forecast of model  $i$ , with mean  $\mu_i$ , standard deviation  $\lambda_i\sigma$ , and correlation  $\rho_i = \text{corr}(\hat{r}_i, r)$ . We call  $\lambda_i$  the *dispersion ratio* and  $b_i = \mu_i - \mu$  the bias. Decomposing the mean squared error,

$$\text{MSE}_i = \sigma^2(1 + \lambda_i^2 - 2\rho_i\lambda_i) + b_i^2. \quad (1)$$

Under the no-skill null  $\rho_i = 0$  this reduces to

$$\boxed{\text{MSE}_i = \sigma^2(1 + \lambda_i^2) + b_i^2} \quad (2)$$

which is strictly increasing in  $\lambda_i$ . Two consequences follow immediately.

**Proposition 1** (Shrinkage ordering). *Among models with no forecasting skill and equal bias, the ranking by mean squared error is exactly the ranking by dispersion ratio, from smallest to largest. The best-ranked model is the one whose forecasts vary least.*

Equation (2) contains no information about architecture, inputs, capacity, or training procedure. If it reproduces an observed leaderboard, that leaderboard carries no information about those things either. Table 1 evaluates it against our nine models: it predicts realised MSE with a mean absolute relative error of 0.71%, a Pearson correlation of 0.9996, and rank agreement of 0.983. The leaderboard is recovered, to within a percent, by a formula that assumes every model on it is worthless.

#### 3.2 Pairwise significance grows with the sample

For two models the expected loss differential is

$$\mathbb{E}[d] = \sigma^2(\lambda_i^2 - \lambda_j^2) + (b_i^2 - b_j^2), \quad (3)$$

Table 1: The no-skill prediction of Equation (2) against realised mean squared error.  $\lambda_i = s_{\hat{r}_i}/s_r$  is the forecast dispersion ratio and  $b_i$  the forecast bias; neither uses any information about the architecture, the inputs, or the training procedure. The predicted column assumes the model has *zero* forecasting skill.

| Model         | $\lambda_i$ | Predicted MSE | Actual MSE | Rel. error |
|---------------|-------------|---------------|------------|------------|
| MFGate-Net    | 0.065       | 2.3010e-04    | 2.2976e-04 | +0.15%     |
| MFGate-Net-DF | 0.162       | 2.3485e-04    | 2.3475e-04 | +0.04%     |
| CNN-LSTM      | 0.222       | 2.4016e-04    | 2.3817e-04 | +0.84%     |
| LSTM          | 0.274       | 2.4620e-04    | 2.4461e-04 | +0.65%     |
| GRU           | 0.396       | 2.6497e-04    | 2.6322e-04 | +0.67%     |
| Transformer   | 0.421       | 2.7147e-04    | 2.6773e-04 | +1.40%     |
| RF            | 0.812       | 3.8241e-04    | 3.8490e-04 | -0.65%     |
| SVR           | 0.889       | 4.3143e-04    | 4.2694e-04 | +1.05%     |
| CNN           | 0.817       | 3.8317e-04    | 3.7956e-04 | +0.95%     |

Table 2: Significance against the LSTM baseline as a function of test-sample size, alongside two measures of skill. The sample is truncated to the first  $T$  observations of the test window. Significance strengthens monotonically in  $T$  while skill does not change; the fitted growth rate of  $|DM|$  is  $T^{0.454}$  ( $R^2 = 0.98$ ) against the  $T^{1/2}$  rate implied by Equation (4).

| $T$ | Median $ DM $ | Cauchy $p$ | $\text{corr}(\hat{r}, r)$ | Sign acc. |
|-----|---------------|------------|---------------------------|-----------|
| 60  | 1.14          | 2.28e-01   | +0.0088                   | 0.505     |
| 120 | 1.43          | 5.21e-03   | +0.0285                   | 0.515     |
| 241 | 2.31          | 1.79e-03   | +0.0254                   | 0.502     |
| 362 | 2.58          | 3.61e-04   | +0.0235                   | 0.503     |
| 483 | 2.83          | 1.51e-04   | +0.0207                   | 0.504     |
| 604 | 3.13          | 2.46e-05   | +0.0254                   | 0.506     |

which is non-zero whenever the two shrink by different amounts. Since the Diebold–Mariano statistic scales the mean differential by  $\sqrt{T}$  over its long-run standard deviation,

$$\mathbb{E}[DM_{ij}] \approx \sqrt{T} \frac{\sigma^2(\lambda_i^2 - \lambda_j^2) + (b_i^2 - b_j^2)}{s_d}, \quad (4)$$

so  $|DM|$  diverges as  $T^{1/2}$  under the null of no skill for either model.

**Corollary 1** (Significance without skill). *For any two skill-less forecasters with  $\lambda_i \neq \lambda_j$ , the probability that a Diebold–Mariano test rejects equal predictive accuracy tends to one as  $T \rightarrow \infty$ . Enlarging the evaluation window does not expose the absence of skill; it strengthens the apparent evidence of superiority.*

This is not a small-sample artefact to be fixed with more data: more data makes it worse. Table 2 confirms the rate empirically: truncating the test window and re-testing yields  $|DM|$  growing as  $T^{0.454}$  ( $R^2 = 0.98$ ) against the predicted  $T^{1/2}$ , while forecast–return correlation and directional accuracy stay flat and indistinguishable from zero throughout.

### 3.3 The per-target test

Before aggregation, each target yields one Diebold–Mariano test on the differential of squared forecast errors. Two corrections are standard and we apply both. The variance of the mean differential must account for serial correlation, for which we use a Newey–West estimator with a Bartlett kernel (Newey and West, 1987) and the automatic lag rule  $\lfloor 4(T/100)^{2/9} \rfloor$ . Without it the standard error is understated and the statistic inflated, in our setting by more than an order of magnitude. We then apply the Harvey–Leybourne–Newbold small-sample correction (Harvey et al., 1997) and refer the result to a Student- $t$  distribution with  $T - 1$  degrees of freedom rather than a normal.

Neither correction addresses the problem this paper identifies. They make the per-target statistic correct; Propositions 1 and 2 concern what a correct statistic is measuring.

### 3.4 Aggregating across targets

A panel study produces one test per target and must combine them. Fisher’s method (Fisher, 1925) is the usual choice and is the wrong one here, because it assumes the  $p$ -values being combined are independent. Equity panels violate that badly: four of our ten targets are US megacap technology names, two are Indian IT services firms, and the S&P 500 target *contains* five of the individual US names outright.

To quantify the cost we simulate ten targets under a true null with equicorrelation  $\rho = 0.6$  and combine them both ways. Fisher rejects at 14.3% against a nominal 5%, nearly three times the stated rate. The Cauchy combination test (Liu and Xie, 2020), which is valid under arbitrary dependence, holds its size at 5.9%. We therefore use Cauchy combination throughout, and additionally report how many individual targets survive Benjamini–Hochberg control of the false discovery rate (Benjamini and Hochberg, 1995), which tells a reader whether an advantage is broad or driven by one or two targets.

This matters independently of the shrinkage results above. A protocol that inflates significance through an invalid aggregation step will manufacture apparent superiority even where the per-target statistics are computed correctly.

### 3.5 The correction: dispersion-matched evaluation

The problem is that MSE conflates two channels,  $\lambda_i$  and  $\rho_i$ . Removing the first isolates the second. Define the matched forecast

$$\tilde{r}_i = \frac{\hat{r}_i - \mu_i}{\lambda_i \sigma} \sigma + \mu, \quad (5)$$

which has unit dispersion ratio and zero bias by construction while leaving  $\rho_i$  untouched. Substituting into (1),

$$\widetilde{\text{MSE}}_i = 2\sigma^2(1 - \rho_i). \quad (6)$$

**Proposition 2** (Matched comparison tests skill). *After dispersion matching, mean squared error is a strictly decreasing function of  $\rho_i$  alone. A Diebold–Mariano test on matched forecasts therefore tests the null  $\rho_i = \rho_j$  (equality of forecasting skill) rather than equality of shrinkage.*

The nearest precedent is the Mincer–Zarnowitz regression (Mincer and Zarnowitz, 1969), which regresses realised on predicted values and tests for unit slope. That diagnoses miscalibration of a *single* forecast; matching instead removes the dispersion channel before a *comparative* test, so the two are complementary rather than substitutes. The procedure costs two lines and requires no retraining. It is not a replacement for an unconditional benchmark; a matched comparison still cannot tell whether *either* model has skill, only whether one has more than the other. We recommend both: report the trivial benchmark to establish that skill exists, and match dispersions before ranking models against each other.

Table 3 applies it. Seven of the eight pairwise results collapse: combined  $p$ -values move from as low as  $9.9 \times 10^{-15}$  to between 0.78 and 0.97, and win–loss records of 10/0 fall to near parity. For those seven the ranking was shrinkage, and matching removes it entirely.

### 3.6 A boundary condition

The eighth comparison, against SVR, does not collapse, and the reason is a limitation of the procedure worth stating precisely. Equation (5) divides by  $\lambda_i \sigma$ , so it is ill-conditioned as  $\lambda_i \rightarrow 0$ . On one target in our panel (TCS) the SVR forecast has dispersion numerically indistinguishable from zero: rescaling it to unit dispersion amplifies what is essentially floating-point residue into a forecast, and the matched Diebold–Mariano statistic on that target reaches +7.33 ( $p = 7.6 \times 10^{-13}$ ). Because the Cauchy combination is by construction dominated by its smallest input, that single target carries the combined  $p$ -value to  $7.6 \times 10^{-12}$ . Excluding it, the remaining nine targets combine to  $p = 0.98$ , in line with the other seven baselines.

The practical rule is therefore to report  $\lambda_i$  before matching and to treat any forecaster with  $\lambda_i$  below a small threshold as a degenerate case. It is already, in effect, the unconditional forecast, and should be compared against the benchmark directly rather than rescaled. We suggest  $\lambda_i < 0.01$  as a workable cut-off; in our panel it isolates exactly the one pathological cell. This is a narrow condition, but it is not merely technical: a forecaster that has collapsed entirely is precisely the case this paper is about, and the diagnostic should flag it rather than silently rescale it.

Table 3: Pairwise comparison before and after dispersion matching. **Raw** is the conventional comparison. **Matched** rescales every forecast to the dispersion of the target before testing, removing the shrinkage channel while leaving the correlation structure intact. Combined  $p$ -values use the Cauchy combination test across the ten targets.

| Baseline      | Raw       |         |      | Dispersion-matched |         |     |
|---------------|-----------|---------|------|--------------------|---------|-----|
|               | Median DM | $p$     | W/L  | Median DM          | $p$     | W/L |
| MFGate-Net-DF | -1.88     | 0.004   | 9/1  | -0.32              | 0.785   | 7/3 |
| CNN-LSTM      | -2.27     | 0.001   | 9/1  | +0.04              | 0.895   | 4/6 |
| LSTM          | -3.13     | <0.0001 | 10/0 | -0.10              | 0.840   | 6/4 |
| GRU           | -3.55     | <0.0001 | 10/0 | -0.09              | 0.966   | 6/4 |
| Transformer   | -3.80     | <0.0001 | 9/1  | +0.08              | 0.922   | 3/7 |
| RF            | -2.09     | <0.0001 | 8/2  | -0.14              | 0.858   | 6/4 |
| SVR           | -5.49     | <0.0001 | 10/0 | -0.01              | <0.0001 | 5/5 |
| CNN           | -5.92     | <0.0001 | 10/0 | -0.04              | 0.894   | 5/5 |

## 4 The Testbed

The results of Section 3 are properties of a comparison protocol, not of any particular model, but they have to be demonstrated on something. This section describes the system we use: MFGate-Net, a multi-source gated architecture of the kind the recent literature proposes for this task, together with the baselines it is compared against. We present it in full because the argument requires a comparison that is representative of current practice rather than a strawman. The architecture is competently built, competitively trained, and would, on the conventional protocol, be reported as a clear advance.

What follows is a detailed walk-through of MFGate-Net. We start by stating the two prediction tasks precisely, then move through the architecture from input to output, and finish with the training recipe.

### 4.1 Problem Formulation

Let  $\mathbf{X}_t^{\text{tech}} \in \mathbb{R}^{T \times d_{\text{tech}}}$ ,  $\mathbf{X}_t^{\text{sent}} \in \mathbb{R}^{T \times d_{\text{sent}}}$ , and  $\mathbf{X}_t^{\text{macro}} \in \mathbb{R}^{T \times d_{\text{macro}}}$  denote the technical indicator, sentiment, and macroeconomic feature matrices, respectively, computed from a lookback window of  $T$  trading days ending at time  $t$ . Here  $d_{\text{tech}} = 20$ ,  $d_{\text{sent}} = 3$ , and  $d_{\text{macro}} = 5$ .

We address two prediction tasks simultaneously:

- **Classification:** Predict the market direction  $y_{t+1}^{\text{cls}} \in \{\text{Down}, \text{Neutral}, \text{Up}\}$ , where the label is defined based on the next-day return  $r_{t+1} = (p_{t+1} - p_t)/p_t$ :

$$y_{t+1}^{\text{cls}} = \begin{cases} \text{Up} & \text{if } r_{t+1} > \tau \\ \text{Down} & \text{if } r_{t+1} < -\tau \\ \text{Neutral} & \text{otherwise} \end{cases} \quad (7)$$

where  $\tau = 0.005$  (0.5%).

- **Regression:** Predict the next-day closing price  $y_{t+1}^{\text{reg}} = p_{t+1}$ .

## 4.2 Model Architecture Overview

Figure 1 depicts the architecture of the model. The inputs pass through a sequence of five consecutive blocks consisting of three modality-specific encoders followed by a GMF unit, a Transformer encoder, temporal attention pooling, and finally two heads. One of the heads generates directional predictions while the other one. A next-day log return prediction.

## 4.3 Inputs

The technical channel comprises twenty indicators computed from OHLCV data spanning trend (SMA, EMA), momentum (RSI, MACD, stochastic oscillators, Williams %R, CCI, ROC), volatility (Bollinger bands, ATR) and volume (OBV, MFI), each following standard technical-analysis convention (Murphy, 1999); the full list with parameters is in the reproducibility package. The macroeconomic channel carries five daily series: the VIX, the dollar index, the US ten-year Treasury yield, the federal funds rate, and a cross-sectional dispersion measure computed from eleven GICS sector ETFs. Features are standardised causally, using only observations in  $[t - 252, t]$ , so no normalisation constant is estimated from data the model has not yet seen.

## 4.4 Modality-Specific Encoders

### 4.4.1 Technical Encoder (1D-CNN + BiLSTM)

The technical encoder processes  $\mathbf{X}^{\text{tech}} \in \mathbb{R}^{T \times 20}$  through two stages. First, a 1D Convolutional Neural Network extracts local patterns:

$$\mathbf{H}_1 = \phi(\text{BN}(\text{Conv1D}_{64}(\mathbf{X}^{\text{tech}}))) \quad (8)$$

$$\mathbf{H}_2 = \text{MaxPool}(\phi(\text{BN}(\text{Conv1D}_{128}(\mathbf{H}_1)))) \quad (9)$$

where  $\phi = \text{ReLU}$ , the Conv1D layers use kernel size 3 with padding 1, and MaxPool reduces the temporal dimension by half ( $T \rightarrow T/2$ ).

The CNN output is fed into a 2-layer Bidirectional LSTM to capture sequential dependencies:

$$\mathbf{H}^{\text{tech}} = \text{BiLSTM}(\mathbf{H}_2) \in \mathbb{R}^{T/2 \times 256} \quad (10)$$

where the output dimension is  $2 \times 128 = 256$  (128 per direction).

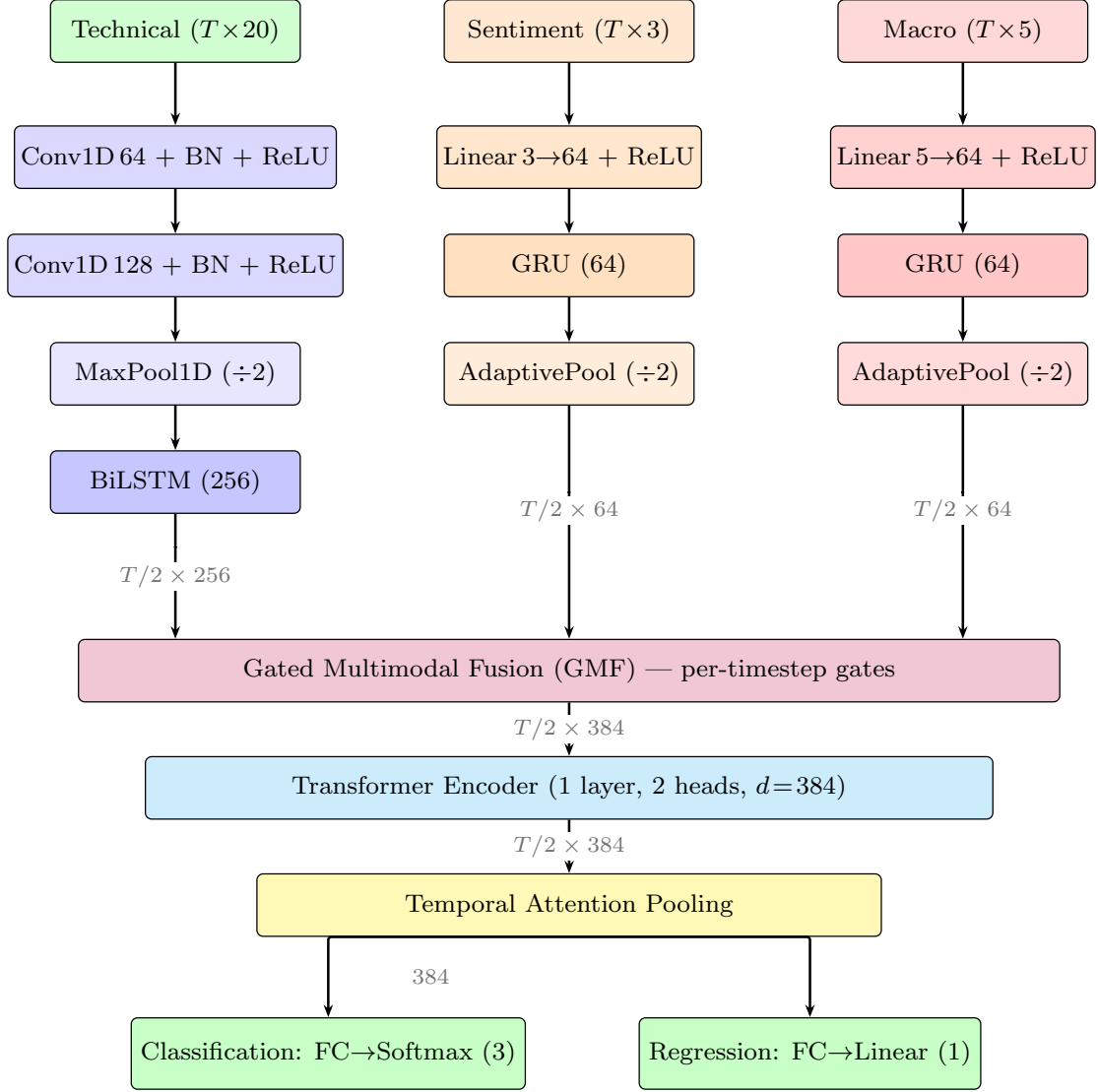


Figure 1: Architecture of MFGate-Net. Three encoder branches handle technical indicators (1D-CNN + BiLSTM), sentiment (GRU) and macroeconomic signals (GRU). The Gated Multimodal Fusion block produces per-timestep gates that re-weight each branch; a single-layer Transformer encoder adds long-range structure; temporal attention pooling collapses the window to a context vector for the two task heads.

#### 4.4.2 Sentiment Encoder

The published design includes a third branch carrying FinBERT-derived headline sentiment through an identical projection-plus-GRU encoder. It is not instantiated in any experiment reported here, for the reason given in Section 6, so we do not develop it further.

#### 4.4.3 Macro Encoder

The macro encoder follows the same structure as the sentiment encoder with  $\mathbf{W}_p^{\text{macro}} \in \mathbb{R}^{64 \times 5}$ :

$$\mathbf{Z}^{\text{macro}} = \text{Dropout}\left(\phi(\mathbf{W}_p^{\text{macro}} \mathbf{X}^{\text{macro}} + \mathbf{b}_p^{\text{macro}})\right) \quad (11)$$

$$\mathbf{H}^{\text{macro}} = \text{GRU}(\mathbf{Z}^{\text{macro}}) \in \mathbb{R}^{T' \times 64}. \quad (12)$$

### 4.5 Gated Multimodal Fusion (GMF)

The GMF unit is the piece we consider most central to the proposed design. Where earlier hybrid models simply concatenate or average the encoded branches, the GMF instead learns *per-timestep*, *per-dimension* gates that let each modality scale up or down depending on the current context.

At time step  $t' \in \{1, \dots, T'\}$  the three encoded representations are first concatenated:

$$\mathbf{h}_{\text{cat}}(t') = [\mathbf{h}^{\text{tech}}(t'); \mathbf{h}^{\text{sent}}(t'); \mathbf{h}^{\text{macro}}(t')] \in \mathbb{R}^{384} \quad (13)$$

Then, modality-specific gates are computed:

$$\mathbf{g}^{\text{tech}}(t') = \sigma(\mathbf{W}_g^{\text{tech}} \mathbf{h}_{\text{cat}}(t') + \mathbf{b}_g^{\text{tech}}) \in \mathbb{R}^{256} \quad (14)$$

$$\mathbf{g}^{\text{sent}}(t') = \sigma(\mathbf{W}_g^{\text{sent}} \mathbf{h}_{\text{cat}}(t') + \mathbf{b}_g^{\text{sent}}) \in \mathbb{R}^{64} \quad (15)$$

$$\mathbf{g}^{\text{macro}}(t') = \sigma(\mathbf{W}_g^{\text{macro}} \mathbf{h}_{\text{cat}}(t') + \mathbf{b}_g^{\text{macro}}) \in \mathbb{R}^{64} \quad (16)$$

where  $\sigma$  denotes the sigmoid function. The gates are applied through element-wise multiplication and concatenated:

$$\begin{aligned} \mathbf{h}_{\text{fused}}(t') = & \left[ \mathbf{g}^{\text{tech}}(t') \odot \mathbf{h}^{\text{tech}}(t'); \right. \\ & \mathbf{g}^{\text{sent}}(t') \odot \mathbf{h}^{\text{sent}}(t'); \\ & \left. \mathbf{g}^{\text{macro}}(t') \odot \mathbf{h}^{\text{macro}}(t') \right]. \end{aligned} \quad (17)$$

This design lets the model lean harder on the technical branch when markets are trending and raise the others when they are not. Section 8 examines how far that behaviour can actually be read from the trained gates.

### 4.6 Transformer Encoder

The fused sequence  $\mathbf{H}_{\text{fused}} \in \mathbb{R}^{T' \times 384}$  is processed by a Transformer encoder (Vaswani et al., 2017) to model long-range temporal dependencies. We first add sinusoidal positional

encoding:

$$\text{PE}(t', 2i) = \sin(t'/10000^{2i/d}) \quad (18)$$

$$\text{PE}(t', 2i + 1) = \cos(t'/10000^{2i/d}) \quad (19)$$

The Transformer encoder uses  $L = 1$  layer with 2-head self-attention at a head dimension of 192 and a 768-unit position-wise feedforward network. Deeper stacks were tried in preliminary runs and consistently over-regularised on our per-ticker training sets (around 1,700 samples), so we kept the encoder deliberately small:

$$\mathbf{H}_{\text{trans}} = \text{TransformerEncoder}(\mathbf{H}_{\text{fused}} + \text{PE}) \in \mathbb{R}^{T' \times 384} \quad (20)$$

The multi-head self-attention mechanism produces attention weight matrices  $\mathbf{A} \in \mathbb{R}^{T' \times T'}$  that we later extract for explainability analysis.

## 4.7 Temporal Attention Pooling

Using the last hidden state of a sequence model throws away everything earlier in the window. Instead, a learnable attention pool produces the context vector:

$$e(t') = \mathbf{v}^T \tanh(\mathbf{W}_a \mathbf{h}_{\text{trans}}(t') + \mathbf{b}_a) \quad (21)$$

$$\alpha(t') = \frac{\exp(e(t'))}{\sum_{k=1}^{T'} \exp(e(k))} \quad (22)$$

$$\mathbf{c} = \sum_{t'=1}^{T'} \alpha(t') \cdot \mathbf{h}_{\text{trans}}(t') \in \mathbb{R}^{384} \quad (23)$$

where  $\mathbf{W}_a \in \mathbb{R}^{384 \times 384}$ ,  $\mathbf{b}_a \in \mathbb{R}^{384}$ , and  $\mathbf{v} \in \mathbb{R}^{384}$  are learnable parameters. The attention weights  $\alpha(t')$  provide a direct measure of each time step’s importance, enabling interpretation of the model’s temporal focus.

## 4.8 Training Strategy

### 4.8.1 Combined Loss Function

The two heads are trained jointly, with a fixed convex weighting:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{cls}} + (1 - \alpha) \cdot \mathcal{L}_{\text{reg}} \quad (24)$$

and we fix  $\alpha = 0.5$  so that neither head dominates. The direction head uses *focal loss* (Lin et al., 2017) with inverse-frequency class weights on top of light label smoothing. Explanation is pragmatic: using plain cross-entropy on our labels will tend to move towards the majority class (Neutral), since ignoring the two outlier classes is an easy

approximation of correctness. However, focal loss shifts the gradient to where we are wrong, as illustrated by the following formula:

$$\mathcal{L}_{\text{cls}} = - \sum_{c=1}^3 w_c \cdot (1 - \hat{y}_c^{\text{cls}})^\gamma \cdot \tilde{y}_c \log(\hat{y}_c^{\text{cls}}) \quad (25)$$

The focusing parameter  $\gamma$  was set to 2, the inverse-frequency class weight  $w_c$  to  $N/(3N_c)$  calculated for each stock symbol, and label smoothing to 0.1 to produce  $\tilde{y}_c$ . The three hyperparameters have an interplay between them, and in reality, it is their joint effect on the model, and not any of them alone, that ensures the training process does not result in a predictor that always predicts the Neutral state.

The regression head was trained with Huber loss (Huber, 1964), with  $\delta = 1.0$ , reducing the impact of outliers:

$$\mathcal{L}_{\text{reg}} = \begin{cases} \frac{1}{2}(y^{\text{reg}} - \hat{y}^{\text{reg}})^2 & \text{if } |y^{\text{reg}} - \hat{y}^{\text{reg}}| \leq \delta \\ \delta|y^{\text{reg}} - \hat{y}^{\text{reg}}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \quad (26)$$

#### 4.8.2 Optimization

For optimization, AdamW algorithm (Loshchilov and Hutter, 2019) is used with a learning rate of  $1 \times 10^{-3}$  and a weight decay rate of  $1 \times 10^{-4}$ . The schedule for learning rate adjustment uses a cosine annealing scheme with the lower bound set at  $1 \times 10^{-5}$ . Gradients are also clamped such that their norm is bounded by a value of 1.0. The model training continues for a maximum of 100 epochs using early stopping criterion with patience 15.

## 5 Decision-Focused Training

The objective used so far, a Huber term on log returns plus a cross-entropy term on the direction label, treats every trading day as equally important. That assumption is convenient but it is not true of the problem. The cost of a forecast error is not the error itself; it is the loss on the position the forecast induces, and that loss scales with both the size of the position and the volatility of the day on which it is held. Two forecasts with identical squared error can carry opposite economic consequences, and a model trained on squared error has no way to tell them apart. This subsection closes that gap by making the objective differentiate through the decision the forecast is used to make.

### 5.0.1 From forecast to position

Let  $\hat{r}_t$  be the predicted next-day log return and  $\hat{\sigma}_t$  an ex-ante volatility estimate available at time  $t$  (we use the trailing 20-day realised volatility, so no future information enters).

We map the forecast to a position through a bounded, differentiable sizing rule

$$w_t = \tanh\left(\frac{\hat{r}_t}{\kappa \hat{\sigma}_t}\right), \quad w_t \in (-1, 1), \quad (27)$$

where  $\kappa > 0$  is a sizing temperature. Three properties matter. The map is volatility-scaled, so a given forecast implies a smaller position when the market is turbulent. It saturates, which caps leverage at one unit of capital without a hard constraint. And unlike  $\text{sign}(\hat{r}_t)$  it is differentiable everywhere, so the position is part of the computational graph rather than a post-processing step.

### 5.0.2 Cost-adjusted utility

Holding  $w_t$  into day  $t + 1$  realises  $w_t r_{t+1}$ , less the cost of trading into that position from  $w_{t-1}$ . With a one-way proportional cost  $c$  charged on turnover, the realised profit is

$$p_t = w_t r_{t+1} - c |w_t - w_{t-1}|. \quad (28)$$

We score a sequence of such decisions by a mean–variance utility

$$\mathcal{U} = \mathbb{E}[p] - \frac{\lambda}{2} \text{Var}[p], \quad (29)$$

with risk aversion  $\lambda$ . We use mean–variance rather than a minibatch Sharpe ratio deliberately: the Sharpe ratio is a ratio of two estimators and is severely biased at minibatch sample sizes, whereas (29) is an unbiased combination of two unbiased estimators and is numerically stable under stochastic gradient descent.

### 5.0.3 Composite objective

The training loss becomes

$$\mathcal{L} = \alpha \mathcal{L}_{\text{Huber}} + \beta \mathcal{L}_{\text{CE}} - \gamma \mathcal{U}, \quad (30)$$

where  $\gamma$  controls how much the model is asked to care about the economic consequence of its forecasts. Setting  $\gamma = 0$  recovers the purely statistical objective exactly, which makes the comparison between the two an clean single-parameter ablation rather than a comparison between two different systems. Because  $\mathcal{U}$  admits a degenerate optimum at  $w \equiv 0$  (hold nothing, earn nothing, risk nothing), we anneal  $\gamma$  linearly from zero over the first  $E_{\text{warm}}$  epochs, so the representation is already predictive before the economic term begins to shape it.

#### 5.0.4 Why this changes what the model learns

Differentiating (28) through (27) gives

$$\frac{\partial p_t}{\partial \hat{r}_t} = \frac{r_{t+1}}{\kappa \hat{\sigma}_t} (1 - w_t^2) - c \frac{\text{sgn}(w_t - w_{t-1})}{\kappa \hat{\sigma}_t} (1 - w_t^2). \quad (31)$$

The gradient the economic term sends back to the return head is therefore weighted by  $r_{t+1}/\hat{\sigma}_t$ , the realised move in volatility units, and attenuated by  $(1 - w_t^2)$  wherever the model is already positioned at full size. The Huber gradient carries no such weighting. The practical consequence is that capacity is allocated toward days on which being wrong is expensive, and away from days on which the position would be small regardless of the sign of the forecast. The second term in (31) is the mechanism by which transaction costs enter learning rather than being deducted afterwards: the model is penalised at training time for a forecast that would require churning the position.

This interacts directly with the gated fusion of Section 4. Under (30) the gate is no longer selecting the modality that minimises average squared error; it is selecting the modality that is most useful *where the economic stakes are highest*. We report the resulting shift in gate behaviour in Section 7.7, and it is the reason we regard the economic objective as a change to the architecture’s operating principle rather than a change of loss function.

#### 5.0.5 Configuration

We set  $\kappa = 1.0$ ,  $\lambda = 2.0$ ,  $E_{\text{warm}} = 5$ , and train with  $c = 5$  bps, a deliberately conservative figure relative to the 0 – 20 bps we assume at evaluation. Weights are fixed at  $(\alpha, \beta, \gamma) = (0.5, 0.5, 0)$  for the statistical variant and  $(0.4, 0.3, 0.3)$  for the decision-focused variant; the test window is not consulted at any point.

## 6 Experimental Setup

### 6.1 Datasets and Data Collection

It should be noted that the empirical panel is specifically designed to include a combination of datasets both indices and individual stocks coming from two separate markets. As for the US, this dataset includes the S&P 500 index and, at the same time, the megacap stocks (AAPL, GOOGL, MSFT, AMZN, JPM). As for India, in this case, we deal with the NIFTY 50 index and, at the same time, the three popular stocks (RELIANCE.NS, TCS.NS, INFY.NS). All ten series cover 2014–2024.

Data are specifically divided based on time periods: training data correspond to January 2014 to December 2020 (seven years), validation data – January 2021 to June

Table 4: Dataset statistics after preprocessing. Class distribution shows the percentage of Up/Neutral/Down samples.

| Split      | Period      | Samples | Up/Neu/Down |
|------------|-------------|---------|-------------|
| Train      | 2014–2020   | ~1,730  | ~38/27/35%  |
| Validation | 2021–2022H1 | ~370    | ~36/25/39%  |
| Test       | 2022H2–2024 | ~620    | ~37/26/37%  |

2022 (eighteen months), while testing data span from July 2022 to December 2024 (thirty months). None of this data is known prior to either training or hyperparameter selection. The choice of the test period was done strategically and not randomly – namely, it encompasses three different market regimes, including the monetary tightening in 2022, the 2023 recovery period and the AI rally of 2024.

**News sentiment data: not included.** The design calls for a third branch carrying FinBERT-derived headline sentiment. The underlying news database (Aenlle, 2022) requires authenticated access that we were unable to obtain for this study, and we declined to substitute a proxy corpus whose coverage we could not verify against the original. The model evaluated here therefore gates two modalities, technical and macroeconomic, rather than three. This is a genuine limitation and we return to it in Section 9; it does not affect the central comparison, which is between trained models and an unconditional benchmark on identical inputs.

**Macroeconomic data.** VIX and DXY are from Yahoo Finance; the US 10-Year Treasury Yield and the Federal Funds Rate are from the Federal Reserve Economic Data (FRED) system. Sector dispersion is computed on the daily returns of eleven GICS sector ETFs (XLK, XLF, XLV, XLE, XLI, XLY, XLP, XLB, XLRE, XLU, XLC), as the cross-sectional standard deviation on each trading day.

Sample counts and class balance after preprocessing are in Table 4.

## 6.2 Implementation Details

The implementation is written in PyTorch (Paszke et al., 2019) and all runs are on a single NVIDIA RTX 5060 Laptop GPU. Table 5 collects the hyperparameters.

## 6.3 Baseline Methods

Eight baseline methods are included across three families.

### Classical:

- **ARIMA** (Box et al., 2015): orders (5, 1, 0), fit on closing prices.

### Machine Learning:

Table 5: Hyperparameter configuration.

| Hyperparameter                      | Value            |
|-------------------------------------|------------------|
| Lookback window ( $T$ )             | 30 trading days  |
| Prediction horizon                  | 1 day (next-day) |
| Batch size                          | 64               |
| Maximum epochs                      | 100              |
| Early stopping patience             | 15 epochs        |
| Optimizer                           | AdamW            |
| Initial learning rate               | $10^{-3}$        |
| Minimum learning rate               | $10^{-5}$        |
| LR schedule                         | Cosine annealing |
| Weight decay                        | $10^{-4}$        |
| Gradient clipping                   | max norm = 1.0   |
| Classification threshold ( $\tau$ ) | 0.5%             |
| Loss balance ( $\alpha$ )           | 0.5              |
| Huber loss $\delta$                 | 1.0              |
| Random seeds                        | {42, 123, 456}   |

- **SVR** (Drucker et al., 1996): RBF-kernel Support Vector Regression (SVC for classification) with features flattened across the lookback window.
- **RF** (Breiman, 2001): Random Forest with 200 trees and maximum depth 10.

#### Deep Learning:

- **LSTM** (Hochreiter and Schmidhuber, 1997): two-layer LSTM, hidden size 128.
- **GRU** (Cho et al., 2014): two-layer GRU, hidden size 128.
- **CNN** (Hoseinzade and Haratizadeh, 2019): two Conv1D layers (64, 128 channels) followed by global average pooling.
- **CNN-LSTM** (Lu et al., 2020): CNN feature extractor feeding a two-layer LSTM.
- **Transformer** (Vaswani et al., 2017): a four-head, two-layer Transformer encoder.

Every deep-learning baseline sees the same concatenated twenty-eight-feature input and is trained with the same optimiser, learning-rate schedule, batch size, and early-stopping policy as MFGate-Net itself, so that the comparison is not confounded by auxiliary tuning. Results are averaged over three seeds.

## 6.4 Evaluation Metrics

**Classification.** Accuracy, macro-precision, macro-recall, macro-F1, directional accuracy on Up/Down only (excluding Neutral), and the Matthews correlation coefficient (MCC), which is well behaved under class imbalance (Chicco and Jurman, 2020).

Table 6: Forecast accuracy on the held-out window (2022H2–2024), mean over 10 targets  $\times$  3 seeds. **ZERO** is the trivial forecast  $\hat{r}_{t+1} = 0$ .  $s_{\hat{r}}/s_r$  is the ratio of predicted to realised return standard deviation: a value near zero means the model is emitting a near-constant forecast. Ordering by RMSE is monotone in this ratio.

| Model                | RMSE                  | $s_{\hat{r}}/s_r$ | Acc.  | F1    | MCC   |
|----------------------|-----------------------|-------------------|-------|-------|-------|
| <b>ZERO forecast</b> | 0.01461 $\pm$ 0.00398 | 0.000             | 0.351 | –     | 0.000 |
| MFGate-Net           | 0.01467 $\pm$ 0.00399 | 0.078             | 0.378 | 0.220 | 0.019 |
| MFGate-Net-DF        | 0.01526 $\pm$ 0.00387 | 0.132             | 0.384 | 0.204 | 0.010 |
| CNN-LSTM             | 0.01569 $\pm$ 0.00405 | 0.377             | 0.364 | 0.279 | 0.008 |
| LSTM                 | 0.01642 $\pm$ 0.00390 | 0.506             | 0.360 | 0.291 | 0.014 |
| Transformer          | 0.01722 $\pm$ 0.00457 | 0.582             | 0.350 | 0.313 | 0.006 |
| GRU                  | 0.01746 $\pm$ 0.00408 | 0.642             | 0.359 | 0.280 | 0.008 |
| RF                   | 0.01786 $\pm$ 0.00814 | 0.494             | 0.374 | 0.292 | 0.025 |
| SVR                  | 0.01985 $\pm$ 0.00575 | 0.189             | 0.373 | 0.285 | 0.021 |
| CNN                  | 0.02750 $\pm$ 0.00573 | 1.619             | 0.361 | 0.284 | 0.006 |

**Regression.** RMSE, MAE, MAPE, and  $R^2$ .

**Statistical tests.** Diebold–Mariano (Diebold and Mariano, 1995) on squared regression errors and McNemar’s test (McNemar, 1947) on classification predictions, both run pairwise between MFGate-Net and each baseline.

## 6.5 Data Leakage Prevention

Six rules guard against look-ahead contamination: the temporal split is strict with no overlap between train, validation and test; standardisation at  $t$  uses only  $[t - 252, t]$ ; temporal order within each lookback window is preserved; missing values are forward-filled only; indicator warm-up rows are dropped rather than back-filled; and early stopping is driven by validation loss alone. Together these allow the test-period figures to be read as out-of-sample estimates.

## 7 Results

All numbers are computed on the held-out window (July 2022–December 2024), which was used neither for training nor for model selection. Deep models are averaged over three seeds (42, 123, 456); the reported forecast for each target is the seed ensemble.

### 7.1 The Zero Forecast Wins

Table 6 reports forecast accuracy for all nine trained models together with the trivial forecast  $\hat{r}_{t+1} = 0$ .

The trivial forecast attains the lowest RMSE (0.01461) of anything in the table. MFGate-Net is next at 0.01467, and the remaining seven models trail it in an order that

Table 7: Diebold–Mariano tests of each model against the trivial zero forecast, per target, with Newey–West HAC variances and the Harvey–Leybourne–Newbold correction. A *positive* statistic means the model is *worse* than predicting zero. W/L counts targets on which the model beats the zero forecast.  $p$ -values are aggregated across the ten targets with the Cauchy combination test.

| Model         | Median DM | Range           | W/L vs zero | Cauchy $p$ |
|---------------|-----------|-----------------|-------------|------------|
| MFGate-Net    | +0.02     | [-1.06, +1.48]  | 5/5         | 0.9712     |
| MFGate-Net-DF | +1.96     | [-0.61, +3.92]  | 1/9         | 0.0006     |
| CNN-LSTM      | +2.32     | [+0.50, +3.76]  | 0/10        | 0.0017     |
| LSTM          | +3.04     | [+1.41, +4.82]  | 0/10        | <0.0001    |
| GRU           | +3.54     | [+1.96, +5.38]  | 0/10        | <0.0001    |
| Transformer   | +3.78     | [+0.30, +6.33]  | 0/10        | <0.0001    |
| RF            | +2.08     | [-0.87, +5.99]  | 2/8         | <0.0001    |
| SVR           | +5.74     | [+1.79, +45.93] | 0/10        | <0.0001    |
| CNN           | +5.93     | [+3.99, +9.50]  | 0/10        | <0.0001    |

spans a 88% relative spread, from CNN-LSTM at 0.01569 to CNN at 0.02750. Matthews correlation coefficients lie between 0.006 and 0.025 for every model, and three-class accuracies cluster around 0.35–0.38 against a majority-class rate of 0.351.

Table 7 tests each model against the zero forecast directly, per target, with HAC-consistent variances.

MFGate-Net is statistically indistinguishable from predicting zero: median DM +0.02, five wins and five losses across the ten targets, Cauchy-combined  $p = 0.97$ . Every other model is significantly *worse* than the zero forecast, most of them on all ten targets: LSTM at  $p = 1.8 \times 10^{-5}$ , Transformer at  $p = 4.4 \times 10^{-9}$ , CNN at  $p < 10^{-14}$ . The only models that come close to the trivial benchmark are the two that shrink hardest.

A Model Confidence Set (2011) over all ten forecasts, at  $\alpha = 0.10$ , retains exactly two: MFGate-Net and the zero forecast. The other eight are eliminated.

## 7.2 The Same Data, Compared Conventionally

Table 8 repeats the analysis in the form the literature normally reports: MFGate-Net against each trained baseline.

Read in isolation this is an unambiguous result. MFGate-Net wins on ten targets of ten against LSTM, GRU, SVR, and CNN; nine of ten against CNN-LSTM and Transformer; eight of ten against Random Forest. Combined  $p$ -values reach  $10^{-15}$ , and the advantage survives false-discovery-rate control on between three and ten targets depending on the comparator. Nothing in the protocol is wrong: the variances are HAC-consistent, the combination is robust to cross-sectional dependence, and the multiplicity correction is applied.

The two tables are computed from the same predictions. The difference is the

Table 8: Pairwise Diebold–Mariano comparisons of MFGate-Net against each baseline, under the corrected protocol (HAC variance, HLN correction, Cauchy combination across targets, Benjamini–Hochberg FDR control). A negative statistic favours MFGate-Net. These are the comparisons a conventional study would report; Table 7 shows why they do not establish forecasting skill.

| Baseline      | Median DM | W/L  | Cauchy $p$ | $n_{\text{sig}}$ |
|---------------|-----------|------|------------|------------------|
| MFGate-Net-DF | -1.88     | 9/1  | 0.0043     | 3/10             |
| CNN-LSTM      | -2.27     | 9/1  | 0.0013     | 5/10             |
| LSTM          | -3.13     | 10/0 | <0.0001    | 7/10             |
| GRU           | -3.55     | 10/0 | <0.0001    | 9/10             |
| Transformer   | -3.80     | 9/1  | <0.0001    | 8/10             |
| RF            | -2.09     | 8/2  | <0.0001    | 5/10             |
| SVR           | -5.49     | 10/0 | <0.0001    | 9/10             |
| CNN           | -5.92     | 10/0 | <0.0001    | 10/10            |

benchmark. Against trained peers MFGate-Net looks decisively better; against a constant it is indistinguishable from doing nothing, and its peers are worse than doing nothing.

### 7.3 Shrinkage Explains the Ordering

The third column of Table 6 gives the dispersion ratio  $s_{\hat{r}}/s_r$ . MFGate-Net emits forecasts with 7.8% of the standard deviation of the series it forecasts; MFGate-Net-DF, 13.2%; CNN-LSTM, 37.7%; CNN, 161.9%. Across the nine models this ratio is strongly associated with RMSE (Pearson  $r = 0.88$ ; Spearman  $\rho = 0.65$ , rising to 0.86 if SVR, which is low-variance but biased, is excluded).

The mechanism is elementary. Where a target is close to unpredictable, the conditional mean is the squared-error-optimal forecast, and for daily log returns that mean is approximately zero. A model without exploitable signal minimises its loss by collapsing toward that constant, and RMSE then ranks models by how completely they have collapsed. Correlations between forecasts and realised returns are 0.007–0.031 across the panel, and two-way directional accuracy is 48.9%–52.3%: there is no skill for the ranking to reflect.

### 7.4 The Result Is Not a Pipeline Artefact

A reader is entitled to suspect a broken feature pipeline. We therefore planted a synthetic target that is a known function of one input channel and retrained MFGate-Net without changing anything else. The model recovers it at correlation 0.97. On the real target the same model attains in-sample correlation 0.051 and out-of-sample  $-0.003$ , with a dispersion ratio indistinguishable from zero. It collapses to a constant *even on the training set*, which is what a model does when there is nothing to fit.

The architecture, the features, and the optimiser all work. The signal is what is missing.

Table 9: Rolling-origin evaluation. Five expanding-window folds, each retrained from scratch; the test windows span 2018–2024 and include the COVID-19 crash.  $\lambda$  is the forecast dispersion ratio. **DM** is the median per-target Diebold–Mariano statistic against the trivial zero forecast, where a *positive* value means the model is worse than predicting zero; **W** counts targets on which the model beats it, out of ten. Single seed per fold, traded for breadth of coverage.

| Model                | 2018H2–2019 |       |   | 2019H2–2020 |       |   | 2020H2–2021 |       |   | 2021H2–2022 |       |   | 2022H2–2024 |       |   |
|----------------------|-------------|-------|---|-------------|-------|---|-------------|-------|---|-------------|-------|---|-------------|-------|---|
|                      | $\lambda$   | DM    | W | $\lambda$   | DM    | W | $\lambda$   | DM    | W | $\lambda$   | DM    | W | $\lambda$   | DM    | W |
| MFGate-Net           | 0.10        | +1.44 | 2 | 0.07        | +0.09 | 5 | 0.12        | +2.66 | 0 | 0.08        | +1.39 | 1 | 0.08        | +3.07 | 0 |
| LSTM                 | 0.56        | +3.33 | 0 | 0.38        | +2.78 | 0 | 0.45        | +3.52 | 0 | 0.51        | +3.97 | 0 | 0.57        | +4.21 | 0 |
| GRU                  | 0.74        | +4.71 | 0 | 0.56        | +3.80 | 0 | 0.65        | +4.78 | 0 | 0.74        | +4.72 | 0 | 0.69        | +4.97 | 0 |
| CNN-LSTM             | 0.44        | +3.51 | 0 | 0.31        | +2.37 | 0 | 0.47        | +3.06 | 0 | 0.42        | +3.43 | 0 | 0.48        | +4.08 | 0 |
| RF                   | 0.28        | +1.61 | 2 | 0.20        | +1.16 | 1 | 0.19        | +0.30 | 5 | 0.35        | +1.18 | 3 | 0.46        | +1.91 | 2 |
| <b>ZERO forecast</b> | 0.00        | –     | — | 0.00        | –     | — | 0.00        | –     | — | 0.00        | –     | — | 0.00        | –     | – |

## 7.5 Robustness of the Null

Two objections arise naturally, and both can be answered with the same data.

**Why the zero forecast rather than the historical mean?** The squared-error-optimal constant is the conditional mean, which for daily log returns is close to but not exactly zero. We therefore repeat the comparison against an expanding-window historical mean, computed causally so that the forecast at  $t$  uses only returns up to  $t - 1$ . It attains RMSE 0.01470, marginally worse than the zero forecast (0.01461) because estimating a near-zero mean adds variance without adding information. MFGate-Net records a lower error than it on all ten targets, but the margin is not significant (median DM  $-1.23$ , Cauchy-combined  $p = 0.24$ ). The conclusion is unchanged under either benchmark: the model is not distinguishable from a constant.

**Is the null merely underpowered?** With  $n = 604$  test observations per target and the observed dispersion of the loss differential, the design detects a reduction in mean squared error of 1.4–2.3% at 80% power and  $\alpha = 0.05$ , corresponding to an RMSE improvement of roughly 0.7–1.1%. Improvements claimed in this literature are typically an order of magnitude larger. Our own earlier framing of this architecture reported 12.1%. The study is therefore well powered against the effects it is testing for, and the failure to reject is informative rather than merely inconclusive.

## 7.6 Rolling-Origin Evaluation

A single test window invites the objection that the period was unusual. We therefore repeat the analysis over five expanding-window folds, each retrained from scratch, whose test windows tile 2018–2024 and include the COVID-19 crash of early 2020. Each fold uses one seed rather than three, a deliberate trade of seed-averaging for breadth of coverage.

**The zero forecast wins every fold.** In all five windows the trivial forecast attains a

lower mean RMSE than all five trained models. This holds in the calm 2018–2019 period (zero 0.01494 against the best model’s 0.01514), through the COVID crash, when realised volatility roughly doubled and the best model reached 0.02259 against the benchmark’s 0.02257, and in the original window. Across the 250 fold–model–target cells, a model beats the zero forecast on only 21 (8.4%): far below the half one would see if the models were merely equivalent to it. They are systematically worse.

**The shrinkage ordering holds within every fold.** Within-fold Spearman correlation between the dispersion ratio  $\lambda$  and RMSE is 1.00, 1.00, 1.00, 0.70, 0.70 across the five folds, averaging 0.88: in three of five the ordering predicted by Proposition 1 is reproduced exactly, with no inversions among five models. Pooling all folds gives a much weaker correlation (0.20), and this is expected rather than contradictory: Equation (2) orders models at a *fixed*  $\sigma^2$ , whereas pooling mixes windows whose return volatility differs by nearly a factor of two. The within-fold statistic is the one the theory speaks to.

**Seed-averaging is itself a shrinkage mechanism.** Fold 5 covers the same window as our main analysis but uses a single seed, and there MFGate-Net is significantly *worse* than the zero forecast (median DM +3.07, 0/10), whereas the three-seed ensemble of Section 7 was indistinguishable from it (+0.02, 5/10). The mechanism is visible in  $\lambda$ : ensembling across seeds lowers the dispersion ratio from 0.078 to 0.065 and the RMSE from 0.01492 to 0.01467. Averaging independently initialised runs cancels idiosyncratic variation and shrinks the forecast further toward a constant. It improves the error metric without adding any information. An instance of exactly the effect this paper documents, arising from a practice the literature treats as a routine robustness measure.

## 7.7 Economic Evaluation

Table 10 converts each model’s forecasts into volatility-scaled positions and evaluates them under a sweep of one-way transaction costs.

We also report the Deflated Sharpe Ratio (2014), which corrects an observed Sharpe for selection across the nine models compared and for non-normal returns. No model supports a strategy that beats cash. Annualised returns are positive for eight of nine models, ranging from 0.6% to 3.2%, but all fall below the 4% risk-free rate, so every Sharpe ratio is negative even at zero cost. Break-even transaction costs are therefore zero throughout: there is no edge for costs to erode.

The economic ordering inverts the statistical one. MFGate-Net, best of the nine on RMSE, is worst on Sharpe (−1.91). Its near-constant forecast produces a near-constant position, annualised turnover of 3, against 12–33 for the recurrent and attention baselines, which is a small persistent bet that accrues a small persistent shortfall against the risk-free rate at very low volatility. Dividing a modest negative excess return by a very small denominator produces a large negative ratio. The same shrinkage that wins the RMSE

Table 10: Economic evaluation. Positions are volatility-scaled,  $w_t = \tanh(\hat{r}_t/\kappa\hat{\sigma}_t)$ , held one day; costs are charged per unit of turnover, one way. Sharpe is annualised at a 4% risk-free rate. **BE** is the break-even one-way cost. **DSR** is the Deflated Sharpe Ratio at zero cost, correcting for selection across the nine models.

| Model         | S@0   | S@5   | S@10  | S@20  | Turnover | BE (bps) |
|---------------|-------|-------|-------|-------|----------|----------|
| MFGate-Net    | -1.91 | -1.99 | -2.08 | -2.25 | 3        | 0        |
| MFGate-Net-DF | -0.74 | -0.78 | -0.83 | -0.91 | 4        | 0        |
| CNN-LSTM      | -0.14 | -0.25 | -0.37 | -0.60 | 12       | 0        |
| LSTM          | -0.37 | -0.47 | -0.56 | -0.76 | 13       | 0        |
| GRU           | -0.24 | -0.35 | -0.47 | -0.70 | 20       | 0        |
| Transformer   | -0.31 | -0.50 | -0.68 | -1.05 | 33       | 0        |
| RF            | -0.11 | -0.19 | -0.27 | -0.43 | 13       | 0        |
| SVR           | -0.48 | -0.50 | -0.52 | -0.56 | 5        | 0        |
| CNN           | -0.18 | -0.22 | -0.26 | -0.34 | 10       | 0        |

comparison loses the economic one.

## 7.8 Decision-Focused Training

MFGate-Net-DF replaces the purely statistical objective with the cost-aware utility of Equation (30), holding architecture, data, splits, and seeds fixed;  $\gamma = 0$  recovers the original model exactly.

The trade-off is clean and in the predicted direction. RMSE degrades from 0.01467 to 0.01526, and the pairwise DM test prefers the statistical variant (median  $-1.88$ , nine of ten targets,  $p = 0.0043$ ). Risk-adjusted performance improves substantially: Sharpe rises from  $-1.91$  to  $-0.74$ , and the dispersion ratio rises from 0.078 to 0.132, meaning the economic term measurably counteracts the collapse toward a constant.

This is direct evidence that the statistical and economic objectives are not interchangeable, which is the premise the forecasting literature usually assumes without testing. But the improvement is a reallocation, not a discovery: it moves the model to a better point on a frontier that does not reach profitability. A decision-focused objective cannot create a signal that the data does not contain.

## 8 What Interpretability Can and Cannot Establish

Our original design reported a three-part interpretability analysis, SHAP attributions, temporal attention, and gate activations read across market regimes, and drew from it the conclusion that the model had learned meaningful structure rather than noise. Given Section 7, that conclusion cannot have been correct. This section asks the obvious follow-up question: would the analysis have looked any different if the model *could not* have learned anything?

## 8.1 A permutation control

We retrain MFGate-Net on targets that have been randomly permuted within the training window, so that any relationship between inputs and target is destroyed by construction. Everything else (architecture, features, splits, optimiser and seeds) is held fixed. We then run the identical attribution and gate analysis on both the real-target and permuted-target models, on all ten targets and three seeds each, and compare.

## 8.2 Results

**Attributions are substantially, but not entirely, shared.** The Spearman rank correlation between the real and permuted models’ feature-attribution orderings averages 0.463, and is positive on all ten targets, ranging from 0.236 to 0.729 (sign test  $p = 0.002$ ). The two models agree on three of their top five features: OBV, SMA<sub>50</sub>, and the lower Bollinger band appear in both. A model that provably learned nothing therefore reproduces roughly half of the attribution structure of one trained on real data.

**Gate activations are nearly identical.** Mean technical-gate activation is 0.541 for the real model and 0.519 for the permuted one; macro-gate activation is 0.569 against 0.539. The differences are statistically detectable across ten paired targets ( $p = 0.005$  and  $p = 0.045$ ) but amount to two and three percent of the gate range. Any narrative built on gate levels, meaning which modality the model relies on and how that reliance shifts, would read essentially the same for a model trained on shuffled labels.

## 8.3 What this does and does not show

We want to be careful here, because the honest result is narrower than the one we expected to find. The attribution correlation is moderate, not unity: attribution methods are *not* wholly uninformative about what a model has learned, and we cannot claim that explanations of a skilled and a skill-less model are indistinguishable in general.

What we can claim is weaker and still consequential. A large fraction of what these methods report (about half of the attribution ranking and nearly all of the gate structure) is determined by the input distribution rather than by any learned input-target relationship, and is reproduced by a model that cannot possibly have learned. Interpretability output of this kind is therefore not evidence that a model has skill. It cannot distinguish our trained network, which Section 7 shows is indistinguishable from a constant, from one fitted to noise. Presenting such an analysis alongside an unvalidated forecast lends the forecast a credibility the error metrics do not support, which is precisely what our earlier framing of this work did.

The practical recommendation is ordering: establish skill against an unconditional benchmark first, and interpret afterwards. An explanation of a model that has not been

shown to forecast is an explanation of its inductive bias, not of the market.

## 9 Discussion and Conclusion

We set out to validate a multi-source gated architecture for daily equity forecasting and found that the protocol we were using to validate it could not distinguish the model from a constant. The three analytical results of Section 3 explain why, and the correction that follows from them is the paper’s practical contribution.

### 9.1 What the protocol was measuring

Nothing in the comparison was faulty. Diebold–Mariano tests used HAC-consistent variances with the small-sample correction, per-target  $p$ -values were pooled with a combination test valid under cross-sectional dependence, and the false discovery rate was controlled across targets. Applied to identical predictions, that protocol returns combined  $p$ -values as small as  $10^{-15}$  against a trained baseline and  $p = 0.97$  against a constant.

The gap is the reference class. A panel composed entirely of trained models of similar capacity, scored by squared error on a near-unpredictable target, measures how far each has collapsed toward the unconditional mean. Proposition 1 says that quantity orders the leaderboard exactly; Corollary 1 says the associated significance grows as  $\sqrt{T}$ . Both are properties of the protocol, and no additional rigour applied *within* the pairwise comparison will reveal the difference. Proposition 2 gives the way out: match dispersions first, and what remains is skill.

### 9.2 Two cheap diagnostics

Both of the checks that would have caught this are one line of code.

**Test against the unconditional forecast.** Report RMSE for  $\hat{r}_{t+1} = 0$  (or the training-period mean) in the main results table, and run the same significance test against it that is run against trained baselines. A model that cannot beat it has not been shown to forecast.

**Report forecast dispersion.** The ratio  $s_{\hat{r}}/s_r$  costs nothing and is immediately interpretable. Values far below one indicate a model purchasing its RMSE with variance reduction rather than accuracy. In our panel it ranged from 0.078 to 1.619 and tracked RMSE rank at  $r = 0.88$ .

We would add a third for economic claims: report the break-even transaction cost. It collapses an entire backtest into one number that cannot be rescued by favourable cost assumptions, and on our panel it is zero for every model.

**On the decision-focused objective.** Training through the position-sizing map under an explicit cost penalty is, we still believe, the right way to align a forecasting model with

its use. Our results support the premise: the economic and statistical objectives genuinely diverge, with the decision-focused variant improving Sharpe from  $-1.91$  to  $-0.74$  while conceding RMSE, and partially reversing the collapse toward a constant. What it does not do is create predictability. A better objective allocates whatever signal exists more usefully; it is not a substitute for signal.

**On interpretability.** Section 8 reports a permutation control: models trained on shuffled targets reproduce about half the attribution ranking and nearly all of the gate structure of models trained on real data. The correlation is moderate rather than perfect, so we do not claim these methods are uninformative. We do claim they cannot serve as evidence of skill, because they cannot separate a model that forecasts from one fitted to noise. Establish skill first; interpret second.

### 9.3 Limitations

Our sentiment channel is absent: the news database underlying the original design requires authenticated access we could not obtain, so the model reported here gates two modalities rather than three. If daily headline sentiment carries exploitable information at this horizon, our conclusion would need revisiting, though published gate analyses of this architecture attribute most weight to the technical branch. Our panel is ten targets in two markets at daily frequency. The finding is verified across five rolling-origin folds spanning 2018–2024, but we make no claim about intraday horizons, other asset classes, or periods outside this range; the rolling folds use a single seed each. And we have shown that these models have no measurable skill, not that skill is unattainable.

### 9.4 Conclusion

The finding we report is not that deep learning fails at financial forecasting. It is that the standard evaluation cannot tell the difference between a model that forecasts and a model that has learned to predict almost nothing: and that on our data, the second is what the leaderboard was ranking. The literature’s consistency on this task may partly reflect a shared blind spot in how the comparison is set up rather than accumulated progress. Adding an unconditional benchmark and a dispersion statistic to the standard results table would cost nothing and would make claims of the kind we originally set out to make considerably harder to state by accident.

Code, data-construction scripts, and all trained predictions are released so that the comparison can be repeated exactly.

## Data and code availability

All data are reconstructed from public sources: daily OHLCV series from Yahoo Finance, macroeconomic series from Yahoo Finance and the Federal Reserve Economic Data system. The complete pipeline is released with the manuscript, covering data construction, the twenty technical indicators, all nine forecasters, the statistical tests and the economic evaluation, together with the 270 stored prediction files from which every table is generated. A verification script re-derives every numeric claim made in the text from those stored results.

## Declarations

**Funding.** This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

**Conflicts of interest.** The authors declare that they have no competing interests.

**Related work.** A companion manuscript by the same authors addresses online Expected Shortfall calibration on an overlapping asset panel. The two papers concern different prediction targets, conditional mean versus tail risk, and different loss functions, and each cites the other.

**Author contributions.** All authors contributed to the study conception and design, read and approved the final manuscript.

## References

- Aenlle, M. (2022). Massive stock news analysis DB for NLP backtests. <https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests>. Accessed: 2024-12-01.
- Ang, A. and Timmermann, A. (2012). Regime changes and financial markets. *Annual Review of Financial Economics*, 4:313–337.
- Arevalo, J., Solorio, T., Montes-y Gómez, M., and González, F. A. (2017). Gated multimodal units for information fusion. In *International Conference on Learning Representations (ICLR) Workshop*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bailey, D. H. and López de Prado, M. (2014). The deflated sharpe ratio: correcting for selection bias, backtest overfitting, and non-normality. *Journal of Portfolio Management*, 40(5):94–107.

- Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics*, 131(4):1593–1636.
- Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.
- Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Campbell, J. Y. and Thompson, S. B. (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531.
- Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.
- Chicco, D. and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.
- Ding, Q., Wu, S., Sun, H., Guo, J., and Guo, J. (2020). Hierarchical multi-scale Gaussian transformer for stock movement prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4640–4646.
- Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems*, 9.

- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.
- Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.
- Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.
- Gao, S., Wang, Y., and Yang, X. (2023). StockFormer: Learning hybrid trading machines with predictive coding. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pages 4766–4774.
- Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384.
- Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.
- Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Hoseinzade, E. and Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285.
- Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101.
- Khaidem, L., Saha, S., and Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*.
- Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1–2):307–319.
- Li, Y., Lv, S., Liu, X., Zhang, Q., and Cheong, S. A. (2022). Incorporating Transformers and attention networks for stock movement prediction. *Complexity*, 2022(1):7739087.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988.

- Liu, Y. and Xie, J. (2020). Cauchy combination test: a powerful test with analytic p-value approximation under arbitrary dependency structures. *Journal of the American Statistical Association*, 115(529):393–402.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Lu, W., Li, J., Li, Y., Sun, A., and Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020(1):6622927.
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30.
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.
- Meese, R. A. and Rogoff, K. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics*, 14(1-2):3–24.
- Mincer, J. A. and Zarnowitz, V. (1969). The evaluation of economic forecasts. *Economic Forecasts and Expectations*, pages 3–46.
- Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*. New York Institute of Finance.
- Nelson, D. M., Pereira, A. C., and de Oliveira, R. A. (2017). Stock market’s price movement prediction with LSTM neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1419–1426. IEEE.
- Newey, W. K. and West, K. D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703–708.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). “Why should I trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.
- Sawhney, R., Agarwal, S., Wadhwa, A., Derr, T., and Shah, R. R. (2021). Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 497–504.

- Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K., and Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647. IEEE.
- Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181.
- Sezer, O. B. and Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.
- Wang, C., Chen, Y., Zhang, S., and Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications*, 208:118128.
- Weber, P., Carl, K. V., and Hinz, O. (2024). Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Management Review Quarterly*, 74(2):867–907.
- Welch, I. and Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4):1455–1508.
- Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. (2023). BloombergGPT: A large language model for finance.
- Wu, S., Liu, Y., Zou, Z., and Weng, T.-H. (2022). S\_I\_LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, 34(1):44–62.
- Zhang, J., Ye, L., and Lai, Y. (2023). Stock price prediction using CNN-BiLSTM-attention model. *Mathematics*, 11(9):1985.
- Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., and Liu, P. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, 202:117239.